

# **Project - 1: Deploying a Two-Tier Website Using AWS EC2 & RDS**

## **Description:**

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

## **GitHub Repository:**

<https://github.com/visaltyagi/AWS-PROJECT1/>

## **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 To Solve:**

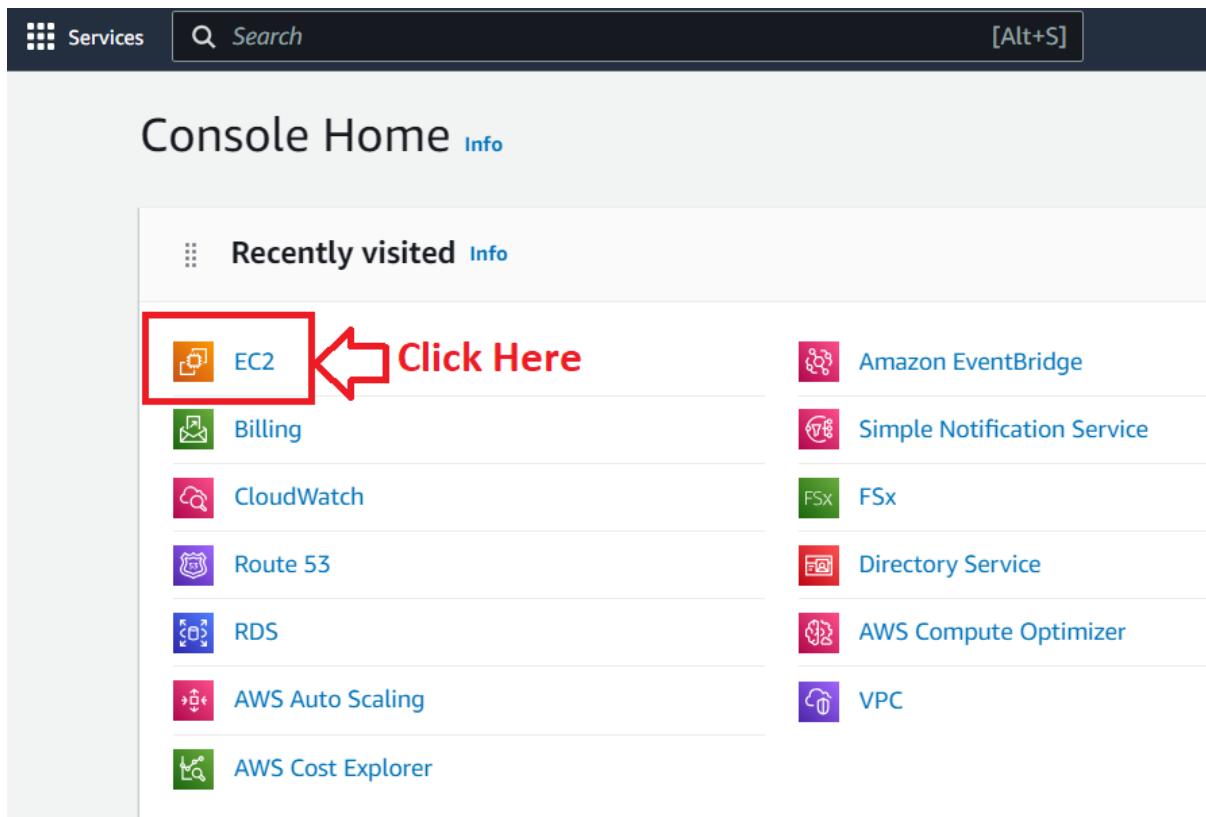
1. Launch an EC2 Instance

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

## **Problem Solution**

### **1. Create an EC2 Instance and Install Apache2 Web Server Over Here**

**Step 1: Login into your AWS Management Console. Click on the “EC2”.**



**Step 2: Go to the “Instances (running)”.**

The screenshot shows the EC2 Dashboard. On the left, there's a sidebar with links: 'New EC2 Experience' (with a feedback button), 'EC2 Dashboard', 'EC2 Global View', 'Events', and a expanded 'Instances' section with 'Instances', 'Instance Types', and 'Launch Templates'. The main area is titled 'Resources' and contains a message: 'You are using the following Amazon EC2 resources in the Asia Pacific'. Below this, there's a table of resources:

Resource Type	Count	Action
Instances (running)	0	Auto Scaling Groups
Elastic IPs	0	Instances
Load balancers	0	Placement groups
Snapshots	0	Volumes

A red box highlights the 'Instances (running)' row, and a red arrow points to the 'Instances' column with the text 'Click here' overlaid in red.

**Step 3: Click on the “Launch instances”.**

Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
cca0c53fe9a32e	Terminated	t2.micro	-	No alarms	+ ap-south-1a	-
8664e9e78b6df6	Terminated	t2.micro	-	No alarms	+ ap-south-1a	-
4bbdcce36b756b	Terminated	t2.micro	-	No alarms	+ ap-south-1a	-
1c54252d727b1a	Terminated	t2.micro	-	No alarms	+ ap-south-1a	-
4b17b8c89c597f	Terminated	t2.micro	-	No alarms	+ ap-south-1a	-

**Step 4: Write the Name as “AWS\_EC2\_Project1” in the “Name” under the “Name and tags” section.**

EC2 > Instances > Launch an instance

### Launch an instance Info

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

**1. Write Your EC2 Instance Name Here**

**Name and tags Info**

Name

AWS\_EC2\_Project1 |

Add additional tags

**Step 5: Choose the “Application and OS Images (Amazon Machine Image)” as “Ubuntu”.**

## ▼ Application and OS Images (Amazon Machine Image) [Info](#)

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

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

Recents      Quick Start      Choose this AMI

Browse more AMIs  
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type ami-0f5ee92e2d63afc18 (64-bit (x86)) / ami-077053fb4029de92f (64-bit (Arm)) Virtualization: hvm ENA enabled: true Root device type: ebs	Free tier eligible
--	--------------------

### Description

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-05-16

### Architecture

64-bit (x86)

### AMI ID

ami-0f5ee92e2d63afc18

Verified provider

## Step 6: Choose the “Instance Type” as “t2.micro”.

▼ Instance type [Info](#)      Choose t2.micro here

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: 0.0724 USD per Hour On-Demand SUSE base pricing: 0.0124 USD per Hour		

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

**Step 7: Choose your previously created login pair (“MY-WEB-SERVER”) or you can create a new “Key pair (login)” using the “Create new key pair” option.**

▼ Key pair (login) [Info](#)

**Choose key pair here**

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

Key pair name - required

MY-WEB-SERVER

[Create new key pair](#)

**Step 8: Click on the “Edit” in the “Network Settings”.**

▼ Network settings [Info](#)

Network [Info](#)

vpc-04c3e6ef2b9c7b60f

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

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

**Edit**

**Click  
Here**

**Step 9: Choose the “Firewall (security groups)” as “Create security group”. While write the “Security Group Name and Description” as the “AWS\_EC2\_Project1”.**

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

**1.**

Select existing security group

Security group name - required

AWS\_EC2\_Project1

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and \_-:/()#,@[]+=;&;!\$\*

Description - required [Info](#)

AWS\_EC2\_Project1

**2. Write security group  
name & description over  
here**

**Step 10: Now, we will give the “Inbound Security Group Rules” one by one.**

**a. First, we will choose the “Type” as “SSH” & the “Source Type” as “Anywhere”.**

#### Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0) Remove

Type <a href="#">Info</a> 1.	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>
ssh	TCP	22
Source type <a href="#">Info</a> 2.	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
Anywhere	Add CIDR, prefix list or security 0.0.0.0/0 X	e.g. SSH for admin desktop

**Click on the “Add security group rule”.**

Anywhere	Add CIDR, prefix list or security 0.0.0.0/0 X	e.g. SSH for admin
----------	--	--------------------

**⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend security group rules to allow access from known IP addresses only.**

**Add security group rule**

**← Click Here**

**b. Second, we will choose the “Type” as “HTTP” & the “Source Type” as “Anywhere”.**

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0) Remove

Type <a href="#">Info</a> HTTP <b>1.</b> ▾	Protocol <a href="#">Info</a> TCP	Port range <a href="#">Info</a> 80
Source type <a href="#">Info</a> Anywhere <b>2.</b> ▾	Source <a href="#">Info</a> <input type="text"/> Add CIDR, prefix list or security 0.0.0.0/0 X	Description - optional <a href="#">Info</a> e.g. SSH for admin desktop

**Click on the “Add security group rule”.**

Anywhere ▾	<input type="text"/> Add CIDR, prefix list or security 0.0.0.0/0 X	e.g. SSH for admin
------------	---	--------------------

**⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend security group rules to allow access from known IP addresses only.**

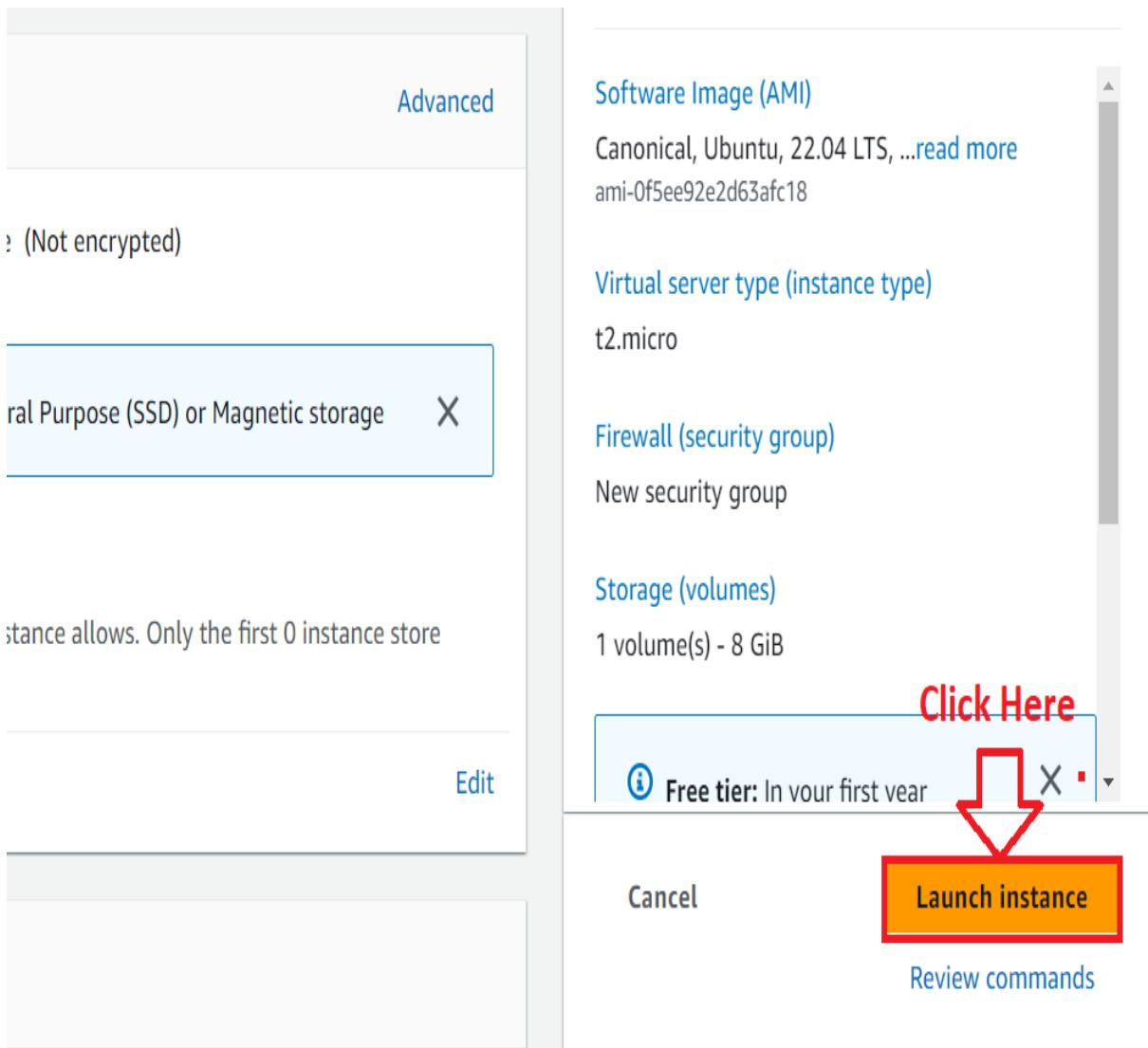
**Add security group rule** ← Click Here

**c. Third, we will choose the “Type” as “MySQL/Aurora” & “Source Type” as “Anywhere”.**

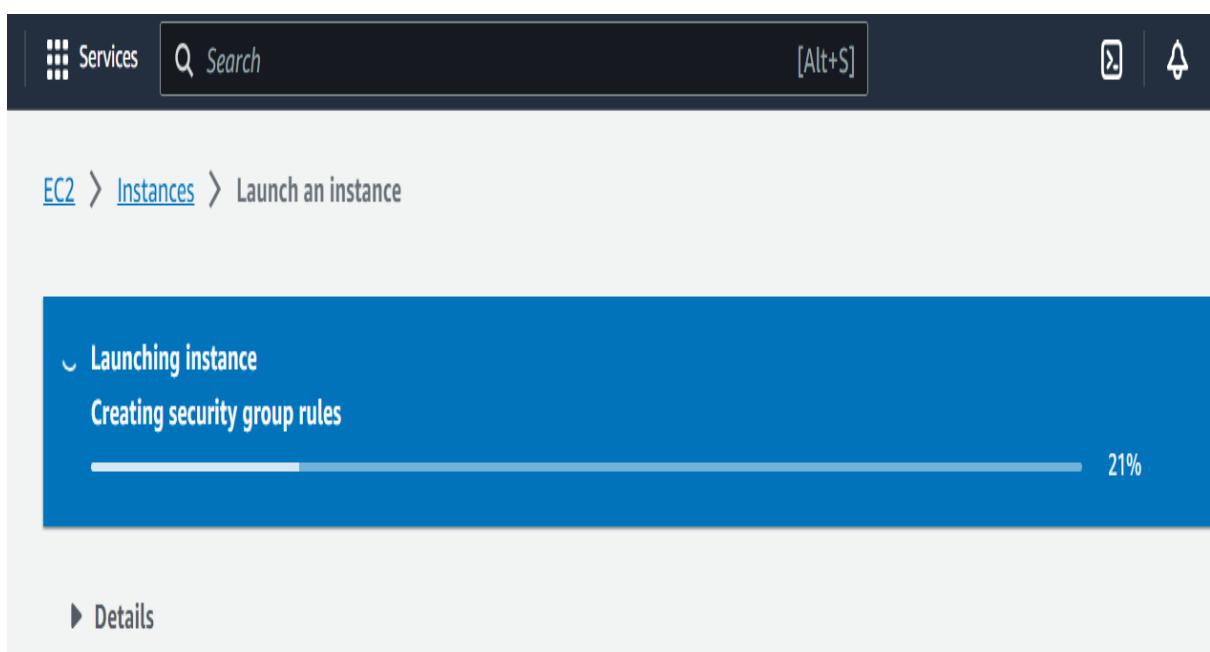
▼ Security group rule 3 (TCP, 3306, 0.0.0.0/0) Remove

Type <a href="#">Info</a> <b>1.</b> MySQL/Aurora ▾	Protocol <a href="#">Info</a> TCP	Port range <a href="#">Info</a> 3306
Source type <a href="#">Info</a> <b>2.</b> Anywhere ▾	Source <a href="#">Info</a> <input type="text"/> Add CIDR, prefix list or security 0.0.0.0/0 X	Description - optional <a href="#">Info</a> e.g. SSH for admin desktop

**Step 11: Click on the “Launch Instance”.**



### Step 12: The instance will be started launching.



**Step 13:** The instance [AWS\_EC2\_Project1] will be launched successfully.  
Click on the “Hyperlink”.

The screenshot shows the AWS EC2 "Launch an instance" success page. At the top, the breadcrumb navigation is "EC2 > Instances > Launch an instance". Below it, a green success message box contains the text "Success" with a checkmark icon and "Successfully initiated launch of instance (i-0824471f793e5106d)". A red arrow labeled "Click Here" points to the instance ID "i-0824471f793e5106d". Below the message box, there is a link "Launch log". In the bottom left corner of the main content area, there is a "Next Steps" section.

**Step 14:** First, the instance will be in the “Pending” state.

The screenshot shows the AWS EC2 "Instances" page with one instance listed. The instance details are: Name: AWS\_EC2\_Proj..., Instance ID: i-0824471f793e5106d, Instance state: Pending, Instance type: t2.micro, Status check: - (No alarms). A red box highlights the "Pending" state, and a red arrow labeled "Pending state" points to it. The "Clear filters" button is also visible.

**Step 15:** It will be in the “Running State” after sometime.

The screenshot shows the AWS EC2 "Instances" page with the same instance listed. The instance details are: Name: AWS\_EC2\_Proj..., Instance ID: i-0824471f793e5106d, Instance state: Running, Instance type: t2.micro, Status check: - (No alarms). A red box highlights the "Running" state, and a red arrow labeled "Running State" points to it. The "Clear filters" button is also visible.

**Step 16: Select the instance [AWS\_EC2\_Project1] & click on the “Connect”.**

The screenshot shows the AWS EC2 Instances page. At the top, there is a search bar with placeholder text "Find instance by attribute or tag (case-sensitive)" and a "Clear filters" button. Below the search bar, there is a table header with columns: Name, Instance ID, Instance state, Instance type, and Status check. A red box labeled "1." highlights the checkbox next to the instance name "AWS\_EC2\_Proj...". Another red box labeled "2. Click Here" highlights the "Connect" button at the top right of the table.

**Step 17: Click on the “Connect” in the “EC2 Instance Connect”. We will be doing all the operation from the “AWS EC2 Instance Connect” option.**

The screenshot shows the "EC2 Instance Connect" configuration page. At the top, there are tabs: "EC2 Instance Connect" (highlighted with a red box and labeled "1."), "Session Manager", "SSH client", and "EC2 serial console". Below the tabs, there is a section for "Instance ID" with a dropdown menu showing "i-0824471f793e5106d (AWS\_EC2\_Project1)". Under "Connection Type", there are two options: "Connect using EC2 Instance Connect" (selected, highlighted with a blue box) and "Connect using EC2 Instance Connect Endpoint". In the "Public IP address" section, the value "13.127.16.2" is listed. Under "User name", there is a text input field containing "ubuntu". A note at the bottom states: "Note: In most cases, the default user name, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name." A red box labeled "2. Click Here" points to the "Connect" button at the bottom right of the page.

## **Step 18: You will be successfully connected to your machine.**

```
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status
```

```
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update
```

```
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*copyright.
```

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.
```

```
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.
```

```
ubuntu@ip-172-31-42-115:~$
```

i-0824471f793e5106d (AWS\_EC2\_Project1)

Public IPs: 13.127.16.2 Private IPs: 172.31.42.115

## **Step 19: First, we will update the "AWS\_EC2\_Project1" machine using this command: "sudo apt update".**

```
ubuntu@ip-172-31-42-115:~$ sudo apt update  
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease  
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]  
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]  
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]  
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]  
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]  
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]  
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]  
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]  
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
```

i-0824471f793e5106d (AWS\_EC2\_Project1)

Public IPs: 13.127.16.2 Private IPs: 172.31.42.115

The “Ubuntu machine [AWS\_EC2\_Project1]” will be successfully updated.

```
Fetched 27.4 MB in 5s (5428 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
129 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-42-115:~$
```

i-0824471f793e5106d (AWS\_EC2\_Project1)

PublicIPs: 13.127.16.2 PrivateIPs: 172.31.42.115

**Step 20: Now, we will install the “apache2” web server over here to run the “PHP” website. Use this command (`sudo apt install apache2 -y`) to install the Apache web server.**

```
ubuntu@ip-172-31-42-115:~$ sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libapr
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser bzip2-doc
```

 Apache 2 Installation  
Command

```
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 → /.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.
```

```
Progress: [ 96%] [#########################################]
```

i-0824471f793e5106d (AWS\_EC2\_Project1)

PublicIPs: 13.127.16.2 PrivateIPs: 172.31.42.115

## Step 21: The apache2 server will be successfully installed.

```
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.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ubuntu@ip-172-31-42-115:~$
```

i-0824471f793e5106d (AWS\_EC2\_Project1)

PublicIPs: 13.127.16.2 PrivateIPs: 172.31.42.115

To check that Apache2 web server has been successfully installed or not, copy the “Public IP Address (13.127.16.2)” & paste it into the browser address bar. You will notice that “Apache2 Web Page” will be successfully shown.

A screenshot of a web browser window. The address bar contains the URL "13.127.16.2". A red box highlights the URL, and a red arrow points from the text "Paste the Public IP Address Over Here" to the URL field. The browser interface includes various icons and links at the top. The main content area displays the Apache2 Default Page. It features the Ubuntu logo, the text "Apache2 Default Page", the word "Ubuntu", and a red button with the text "It works!". Below this, there is explanatory text about the default welcome page and a "Configuration Overview" section with detailed configuration information.

Not secure | 13.127.16.2 Paste the Public IP Address Over Here

Blog... Resources 15 Best Black Friday... 7 Best Gutenberg T... How to create a sit... Page not found – cr... free image sitemap... Search Results f

**Apache2 Default Page**

**Ubuntu**

**It works!**

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

**Configuration Overview**

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

## 2. Deploy the PHP Website on EC2 Server

**Step 1:** First, we will remove the **index.html** file & create a new **index.php** file. Go to the “**html**” directory using the command: “**cd /var/www/html**”.

You will go inside into the **html** directory.

```
No containers need to be restarted.
```

```
No user sessions are running outdated binaries.
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

```
ubuntu@ip-172-31-42-115:~$ cd /var/www/html
```

```
ubuntu@ip-172-31-42-115:/var/www/html$
```



**html directory**

i-0824471f793e5106d (AWS\_EC2\_Project1)

PublicIPs: 13.127.16.2 PrivateIPs: 172.31.42.115

**Step 2:** Run the command “**ls**” & you will notice that “**index.html**” file will be present over here.

```
No user sessions are running outdated binaries.
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

```
ubuntu@ip-172-31-42-115:~$ cd /var/www/html
```

```
ubuntu@ip-172-31-42-115:/var/www/html$ ls
```

```
index.html
```

```
ubuntu@ip-172-31-42-115:/var/www/html$
```



**index.html file**

i-0824471f793e5106d (AWS\_EC2\_Project1)

PublicIP: 13.127.16.2 PrivateIP: 172.31.42.115

**Step 3:** Remove the **index.html** file using the command:- “**sudo rm index.html**”. Do “**ls**” & you will notice that there is no “**index.html**” file

**present over here. Press “enter” from the keyboard after typing the command.**

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ubuntu@ip-172-31-42-115:~$ cd /var/www/html  
ubuntu@ip-172-31-42-115:/var/www/html$ ls  
index.html  
ubuntu@ip-172-31-42-115:/var/www/html$ sudo rm index.html  
ubuntu@ip-172-31-42-115:/var/www/html$ ls  
ubuntu@ip-172-31-42-115:/var/www/html$
```

i-0824471f793e5106d (AWS\_EC2\_Project1)

PublicIPs: 13.127.16.2 PrivateIPs: 172.31.42.115

**Step 4: Create an “index.php” file using the command: - “sudo nano index.php”. Press “Enter” from keyboard after typing the command.**

```
No user sessions are running outdated binaries.
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ubuntu@ip-172-31-42-115:~$ cd /var/www/html  
ubuntu@ip-172-31-42-115:/var/www/html$ ls  
index.html  
ubuntu@ip-172-31-42-115:/var/www/html$ sudo rm index.html  
ubuntu@ip-172-31-42-115:/var/www/html$ ls  
ubuntu@ip-172-31-42-115:/var/www/html$ sudo nano index.php
```

i-0824471f793e5106d (AWS\_EC2\_Project1)

PublicIPs: 13.127.16.2 PrivateIPs: 172.31.42.115

**Step 5: A file Editor will be opened & paste the “PHP Website Code” over here.**

```
GNU nano 6.2

[ N

^G Help          ^O Write Out      ^W Where Is      ^K Cut           ^T Execute
^X Exit          ^R Read File      ^\ Replace       ^U Paste         ^J Justify
```

i-0824471f793e5106d (AWS\_EC2\_Project1)  
Public IPs: 13.127.16.2 Private IPs: 172.31.42.115

**Step 6:** This is a sample PHP web page code which we will paste it into the “Nano Editor”. Copy this PHP web page code.

**Step 7: Paste the above code into the “Nano Editor”.**

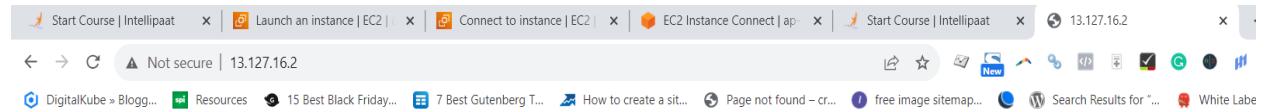
**Step 8: Do “CTRL+S” to save & do “CTRL+X” from exiting the editor. You will be successfully exited from the “Nano Editor”.**

```
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.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ubuntu@ip-172-31-42-115:~$ cd /var/www/html  
ubuntu@ip-172-31-42-115:/var/www/html$ ls  
index.html  
ubuntu@ip-172-31-42-115:/var/www/html$ sudo rm index.html  
ubuntu@ip-172-31-42-115:/var/www/html$ ls  
ubuntu@ip-172-31-42-115:/var/www/html$ sudo nano index.php  
ubuntu@ip-172-31-42-115:/var/www/html$ █
```

i-0824471f793e5106d (AWS\_EC2\_Project1)

Public IPs: 13.127.16.2 Private IPs: 172.31.42.115

**Step 9: Paste the “Public IP Address (13.127.16.2)” in the “Browser Address Bar” & press “enter” from the keyboard. A “PHP Web Page” will be shown.**

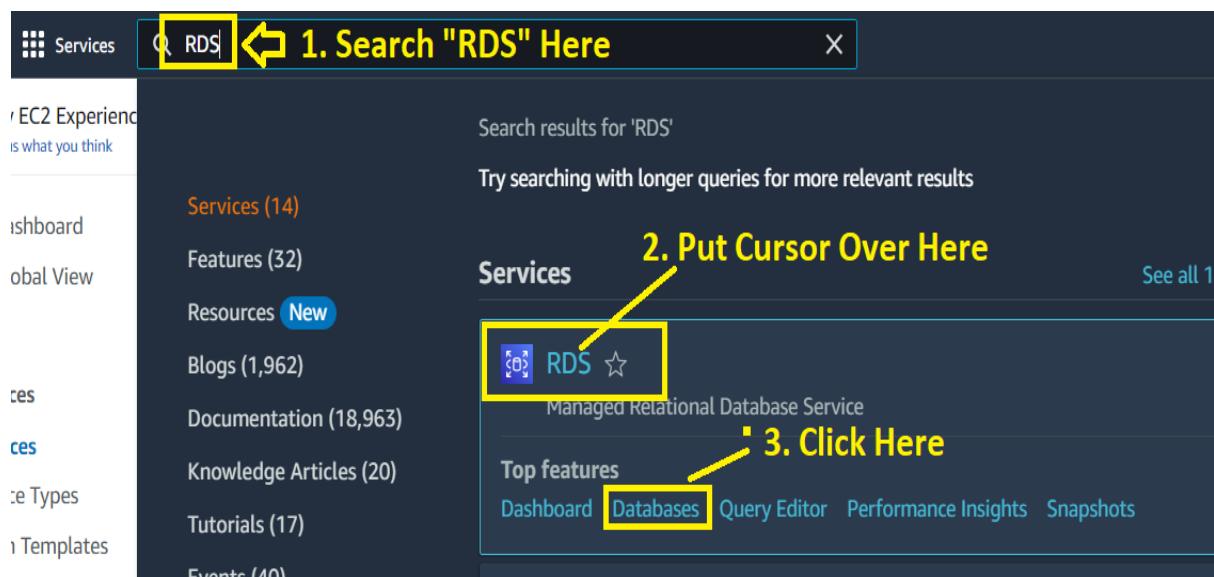
A screenshot of a simple PHP web page. The page contains a form with two text input fields: "Name:" and "Email:". Below the fields is a green "Submit" button. The background is white, and the overall appearance is minimalist.

```
connect_error) { die("Connection failed: " . $conn->connect_error); } if(isset($_POST['firstname']) && isset($_POST['email'])){ $sql = "INSERT INTO data (firstname,email) VALUES ('".$_firstname."', '".$_email."')"; echo "New record created successfully"; } else { echo "Error: " . $sql . " " . $conn->error; } $conn->close(); } ?>
```

### **3. Create a MySQL Database & Set Up Connection Over EC2 Server with Proper Testing**

#### **A. Create a MySQL Database**

**Step 1: Go to the “Services” section & Search the “RDS” over Here. Put the cursor over “RDS” & click on the “Databases”.**



## Step 2: Click on the “Create database”.

The screenshot shows the 'Databases' section of the Amazon RDS console. At the top, there's a blue header bar with information about Aurora I/O-Optimized. Below it, a modal window provides instructions on creating a Blue/Green deployment, with a 'Click Here' link and a red arrow pointing to the 'Create database' button. The main table lists databases, with a search bar and various filters at the top. A message at the bottom states 'No instances found'.

Step 3: In the “Create Database” option, choose the “Standard create” as “Choose a database creation method” & the “Engine Options” as “MySQL”.

The screenshot shows the 'Create database' wizard. The first step, 'Choose a database creation method', has 'Standard create' selected (highlighted with a red box and a red arrow). The second step, 'Engine options', shows three options: 'Aurora (MySQL Compatible)', 'Aurora (PostgreSQL Compatible)', and 'MySQL'. 'MySQL' is selected (highlighted with a red box and a red arrow).

## Step 4: While leave the “SQL Version (MySQL 8.0.33)” as it is.

Edition

MySQL Community



### Known issues/limitations

Review the [Known issues/limitations](#) to learn about potential compatibility issues with specific database versions.

▼ Hide filters

#### Show versions that support the Multi-AZ DB cluster [Info](#)

Create a Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.

#### Show versions that support the Amazon RDS Optimized Writes [Info](#)

Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Engine Version

MySQL 8.0.33



Remain as it is

## Step 5: Choose the “Templates” as the “Free Tier”.

### Templates

Choose a sample template to meet your use case.

#### Production

Use defaults for high availability and fast, consistent performance.

#### Dev/Test

This instance is intended for development use outside of a production environment.

Choose this option



#### Free tier

Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.

[Info](#)

## Step 6: Rename the “DB Instance Identifier” as the “application-database”.

Settings **Write DB Instance Identifier Name Here**

DB instance identifier [Info](#)

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

application-database

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

## Step 7: In the “Credential Settings”, write the “Master username” as “admin” & the “Master Password” as “intel123”.

### ▼ 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.

1. Choose username here

#### Manage master credentials in AWS Secrets Manager

Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

i If you manage the master user credentials in Secrets Manager, some RDS features aren't supported.

[Learn more](#) 

#### Auto generate a password

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

Master password [Info](#)

\*\*\*\*\*

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

2. Fill Password as  
"intel123"

Confirm master password [Info](#)

\*\*\*\*\*

**Step 8: Choose the “DB instance class” as the “Burstable classes (includes t classes) as the “db.t3.micro” in the “Instance Configuration” section.**

#### Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.



Amazon RDS Optimized Writes - new [Info](#)

Show instance classes that support Amazon RDS Optimized Writes

DB instance class [Info](#)

- Standard classes (includes m classes)
- Memory optimized classes (includes r and x classes)
- Burstable classes (includes t classes)

Remain as it is

db.t3.micro

2 vCPUs 1 GiB RAM Network: 2,085 Mbps

Include previous generation classes

**Step 9: While leave the “Storage” Configuration as it is. Disable the “Storage autoscaling” by unselecting the “Enable storage autoscaling”.**

#### Storage

1. No need to change here

Storage type [Info](#)

General Purpose SSD (gp2)

Baseline performance determined by volume size

Allocated storage [Info](#)

20

GiB

The minimum value is 20 GiB and the maximum value is 6,144 GiB

After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization. Your instance will remain available as the storage-optimization operation completes.  
[Learn more](#)

#### ▼ Storage autoscaling

Storage autoscaling [Info](#)

Provides dynamic scaling support for your database's storage based on your application's needs.

Enable storage autoscaling

Enabling this feature will allow the storage to increase after the specified threshold is exceeded.

Deselect by clicking here

**Step 10: In the “Connectivity” section, choose the “Compute resource” as “Connect to an EC2 compute resource”. Choose your “EC2 Instance (AWS\_EC2\_Project1)” here.**

c

**Connectivity [Info](#)**

**1. Choose this option**

Compute resource  
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource  
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource  
Set up a connection to an EC2 compute resource for this database.

**EC2 instance [Info](#)**  
Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

i-0824471f793e5106d  
AWS\_EC2\_Project1

**2. Select EC2 Instance Here**

**Some VPC settings can't be changed when a compute resource is added**  
Adding an EC2 compute resource automatically selects the VPC, DB subnet group, and public access settings for this database. To allow the EC2 instance to access the database, a VPC security group rds-ec2-X is added to the database and another called ec2-rds-X to the EC2 instance. You can remove the new security group for the database only by removing the compute resource.

**Step 11: Choose the “DB subnet group” as the “Automatic setup”.**

DB subnet group [Info](#)

Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

Choose existing  
Choose existing DB subnet group

Automatic setup  
RDS creates a new subnet group for you or reuses an existing subnet group

DB subnet group name

rds-ec2-db-subnet-group-1

Existing DB subnet group reused.

**1. Choose this option**

Public access [Info](#)

**2. Remain as it is**

Yes

RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

No

RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

**Step 12: Choose the “VPC security group (firewall)” as the “Choose existing”, while the “Additional VPC security group” as the “AWS\_EC2\_Project1”.**

VPC security group (firewall) [Info](#)

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

Choose existing  
Choose existing VPC security groups

Create new  
Create new VPC security group

Additional VPC security group

Choose one or more options ▾

AWS\_EC2\_Project1 X

1. ■

2. ■

i Amazon RDS will add a new VPC security group *rds-ec2-2* to allow connectivity with your compute resource.

**Step 13: While leave the "Database authentication options" as by default the "Password authentication".**

Database authentication

Database authentication options [Info](#)

Password authentication  
Authenticates using database passwords.

Password and IAM database authentication  
Authenticates using the database password and user credentials through AWS IAM users and roles.

Password and Kerberos authentication  
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

← Remain this option as "choosen"

**Step 14: Go to the “Additional Configuration” & put the “initial database name” as “intel”. Disable the other features such as “Backup”, “Encryption” & “Maintenance”. Because they are the chargeable options.**

#### Database options

Initial database name [Info](#)

1. Choose Database Name as "intel"

If you do not specify a database name, Amazon RDS does not create a database.

#### DB parameter group [Info](#)

#### Option group [Info](#)

#### Backup

Enable automated backups  
Creates a point-in-time snapshot of your database

2. Disable by Deselecting these options

#### Encryption

Enable encryption  
Choose to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service console. [Info](#)

**Step 15: Click on the “Create database”.**

#### Estimated monthly costs

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

[Learn more about AWS Free Tier.](#)

When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the [Amazon RDS Pricing page](#).

 You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

**Click Here**

Cancel

**Create database**

**Step 16: The database [application-database] will be started creating. It will take some time to setup.**

The screenshot shows the AWS RDS Databases page. At the top, there is a blue info box with the text: "Consider creating a Blue/Green Deployment to minimize downtime during upgrades" and "You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases." Below the info box, the main table has a header row with columns: DB identifier, Status, Role, Engine, Region & AZ, Size, Actions, CPU, and Current activity. A single row is visible below the header, showing the database 'application-database' with a status of 'Creating'. This row is highlighted with a red box.

**Step 17: After sometime, the database “Status” will be shown as “Available”.**

The screenshot shows the AWS RDS Databases page again. The 'Status' column header is highlighted with a red box. In the row for 'application-database', the status is now 'Available', which is also highlighted with a red box. The rest of the table structure remains the same as in the previous screenshot.

**Step 19: Click on “application-database” hyperlink. You will get all the database information.**

The screenshot shows the AWS RDS Database details page for 'application-database'. At the top, there are two green success messages: "Successfully set up a connection between application-database and EC2 instance i-0824471f793e5106d" and "Successfully created database application-database". Below the messages, the breadcrumb navigation shows 'RDS > Databases > application-database'. The main content area has a 'Summary' section with various database metrics. At the bottom, there are tabs for 'Connectivity & security', 'Monitoring', 'Logs & events', 'Configuration', 'Maintenance & backups', and 'Tags'. The 'Connectivity & security' tab is currently selected.

Connectivity & security	Monitoring	Logs & events	Configuration	Maintenance & backups	Tags
Connectivity & security		<b>Copy this Endpoint</b>			
Endpoint & port		Networking		Security	
Endpoint	application-database.calupg2c7sxe.ap-south-1.rds.amazonaws.com	Availability Zone	ap-south-1a	VPC security groups	AWS_EC2_Project1 (sg-04d792595bc10f95d)
Port	3306	VPC	vpc-04c3e6ef2b9c7b60f	Active	rds-ec2-2 (sg-065ff9e0f5f70e541)
		Subnet group	rds-ec2-db-subnet-group-1	Active	
		Subnets	<a href="#">subnet-0cc15942d40e40f23</a> <a href="#">subnet-06f63da6a038b0fd3</a> <a href="#">subnet-0c6de8b75819dbb74</a>	Publicly accessible	
		Network type	IPv4	Certificate authority	<a href="#">Info</a> rds-ca-2019
				Certificate authority date	August 22, 2024, 22:38 (UTC+05:30)

## B. Set Up Database Connection with EC2 Server & Test the Created Database Over Here

**Step 1:** First, we have to install “MySQL” over “EC2” server using this command:- “`sudo apt install mysql-server -y`”. MySQL server will be successfully installed.

```
sudo apt install mysql-server -y
[...]
... Done
... Done
[...]
packages will be installed:
[...]
libclone-perl libencode-locale-perl libevent-pthreads-2.1
html-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-m
ite23 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 :
er-8.0 mysql-server-8.0
[...]
c-sharedcache-perl libbusiness-isbn-perl libwww-perl mailx tinyca
s will be installed:
[...]
libclone-perl libencode-locale-perl libevent-pthreads-2.1
html-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-m
ite23 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 :
er mysql-server-8.0 mysql-server-core-8.0
[...]
alled, 0 to remove and 0 not upgraded.
chives.

  MB of additional disk space will be used.
[...]
```

```
aws | Services | Search [Alt+S]
Setting up libhttp-message-perl (6.36-1) ...
Setting up mysql-server-8.0 (8.0.34-0ubuntu0.22.04.1) ...
update-alternatives: using /etc/mysql/mysql.cnf to provide /etc/mysql/my.cnf (my.cnf) in auto mode
Renaming removed key_buffer and myisam-recover options (if present)
mysqld will log errors to /var/log/mysql/error.log
mysqld is running as pid 2692
Created symlink /etc/systemd/system/multi-user.target.wants/mysql.service → /lib/systemd/system/mysql.service.
Setting up libcgi-pm-perl (4.54-1) ...
Setting up libhtml-template-perl (2.97-1.1) ...
Setting up mysql-server (8.0.34-0ubuntu0.22.04.1) ...
Setting up libcgi-fast-perl (1:2.15-1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
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.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-42-115:~$ █
```

i-0824471f793e5106d (AWS\_EC2\_Project1)  
PublicIPs: 3.110.85.213 PrivateIPs: 172.31.42.115

**Step 2: First, you have to change the directory. Go to the html directory using the command:- cd /var/www/html.**

```
No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-42-115:~$ sudo mysql -h application-database.calupg2c7sxe.ap-south-1.rds.amazonaws.com -u admin -p123
mysql: [Warning] Using a password on the command line interface can be insecure.
ERROR 1045 (28000): Access denied for user 'admin'@'172.31.42.115' (using password: YES)
ubuntu@ip-172-31-42-115:~$ █ cd /var/www/html
ubuntu@ip-172-31-42-115:/var/www/html$ mysql --version
mysql Ver 8.0.34-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu))
```

**Step 3: Now, we will set up the MySQL database connection with the EC2 server using this command: - “*sudo mysql -h application-database.calupg2c7sxe.ap-south-1.rds.amazonaws.com -u admin -pintel123*”.**

**It will be successfully connected with the EC2 server.**

```
ubuntu@ip-172-31-42-115:/var/www/html$ mysql --version
mysql  Ver 8.0.34-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu))
ubuntu@ip-172-31-42-115:/var/www/html$ sudo mysql -h application-database.calupg2c7sxe.ap-south-1.rds.amazonaws.com -u admin -pintel123
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 8.0.33 Source distribution

Copyright (c) 2000, 2023, 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> |
```

↑  
**Type this command &  
press enter from  
keyboard**

**Step 4: Now, type the command “show databases”, it will show all the created databases here. We have created a database named with “intel”. You will notice this database after typing the above query.**

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> show databases; | 1. Type this command
+-----+
| Database      |
+-----+
| information_schema |
| intel          | 2. Database shown
| mysql          |
| performance_schema |
| sys            |
+-----+
5 rows in set (0.00 sec)

mysql> |
```

i-0824471f793e5106d (AWS\_EC2\_Project1)

Public IPs: 3.110.85.213 Private IPs: 172.31.42.115

**Step 5: Now, go to the “intel” database. Write the command (“use intel”) here. Now, we will be inside the “intel” database.**

```
mysql> use intel
Database changed
```

i-0824471f793e5106d (AWS\_EC2\_Project1)

PublicIPs: 3.110.85.213 PrivateIPs: 172.31.42.115

**Step 6: Now, we will create a table named as “data” with provided fields according to the “PHP source code”. In the PHP file, only the “firstname & email” has been given.**

**Use the following query to create a table: “create table data (firstname varchar(20), email varchar(20));”.**

**The table will be successfully created.**

```
mysql> create table data (firstname varchar(20), email varchar(20));
Query OK, 0 rows affected (0.02 sec)

mysql> show table;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'show table' at line 1
mysql> show tables;
+-----+
| Tables_in_intel |
+-----+
| data           |
+-----+
1 row in set (0.00 sec)

mysql>
```

**Step 7: Now, we will insert the value into the data table using the following query: insert into data values ('AWS', 'support@aws.com');**

**Your query will be successfully executed.**

```
mysql> insert into data values ('AWS', 'support@aws.com');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> |
```

i-0824471f793e5106d (AWS\_EC2\_Project1)

PublicIPs: 3.110.85.213 PrivateIPs: 172.31.42.115

**Step 8: If you want to check that data is successfully inserted or not, type this query: select \* from data;**

**You will notice that your query has been successfully inserted.**

```
mysql> select * from data;  
+-----+-----+  
| firstname | email      |  
+-----+-----+  
| AWS       | support@aws.com |  
+-----+-----+  
1 row in set (0.00 sec)
```

```
mysql> |
```

i-0824471f793e5106d (AWS\_EC2\_Project1)

PublicIPs: 3.110.85.213 PrivateIPs: 172.31.42.115

**Step 9: Now, we are exiting from the database using the “exit” command.**

```
mysql> exit  
Bye  
ubuntu@ip-172-31-42-115:/var/www/html$
```

i-0824471f793e5106d (AWS\_EC2\_Project1)

PublicIPs: 3.110.85.213 PrivateIPs: 172.31.42.115

### **C. Configure the Correct Database Settings in PHP file & Insert Dependencies to Remove the Error from PHP Web Page**

#### **1. Configure the Right Database Settings in the PHP File**

**Step 1: Open the “index.php” file using the following command:- “sudo nano index.php”.**

```
mysql> exit  
Bye  
ubuntu@ip-172-31-42-115:/var/www/html$ sudo nano index.php
```

i-0824471f793e5106d (AWS\_EC2\_Project1)

PublicIPs: 3.110.85.213 PrivateIPs: 172.31.42.115

**Step 2: The “index.php” file will be opened. Scroll down & do these two settings for setting up the right database connection:**

**servername = “Real Database Endpoint (application-database.calupg2c7sxe.ap-south-1.rds.amazonaws.com)”**

**username = “admin” instead of “intel”.**

```
GNU nano 6.2
</div>
</td><td colspan="4">
<td colspan="4">
<tr>
<table>
<div>
<div>
<?php
$firstname=$_POST['firstname'];
$email=$_POST['email'];
servername = "application-database.calupg2c7sxe.ap-south-1.rds.amazonaws.com";
$username = "admin";
$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);
}

```

**Change servername & username here**

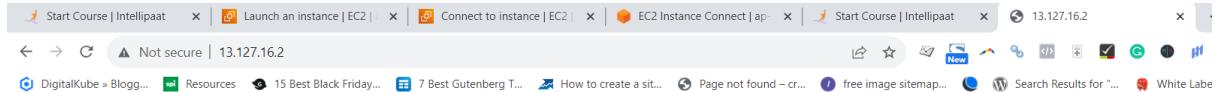
**Step 3: Do CTL+S to save & do CTRL+X to exit from the index.php file.**

```
Last login: Mon Oct 2 11:22:22 2023 from 13.233.177.5
ubuntu@ip-172-31-42-115:~$ cd /var/wwwhtml
-bash: cd: /var/wwwhtml: No such file or directory
ubuntu@ip-172-31-42-115:~$ cd /var/www/html
ubuntu@ip-172-31-42-115:/var/www/html$ sudo nano index.php
ubuntu@ip-172-31-42-115:/var/www/html$
```

i-0824471f793e5106d (AWS\_EC2\_Project1)

Public IPs: 3.110.85.213 Private IPs: 172.31.42.115

**Step 4: When we open the web page by typing the public IP address [13.127.16.2] in the browser address bar. You will notice the database connection error over here.**



```
connect_error) { die("Connection failed: " . $conn->connect_error); } if(isset($_POST['firstname']) && isset($_POST['email'])){ $sql = "INSERT INTO data (firstname,email) VALUES ('".$_firstname."','".$."_email."')"  
echo "New record created successfully"; } else { echo "Error: " . $sql . "  
" . $conn->error; } $conn->close(); } ?>
```

**Now, we will remove this error in our second step.**

## **2. Install the PHP MySQL Database Dependencies to Remove the Error from WebPage**

**Step 1: Copy the below-given php repository command & paste it into the machine & press “enter” from the keyboard.**

**Command:- “sudo add-apt-repository -y ppa:ondrej/php”.**

**The “PHP repository” will be successfully added.**

```
ubuntu@ip-172-31-42-115:/var/www/html$ sudo nano index.php  
ubuntu@ip-172-31-42-115:/var/www/html$ sudo add-apt-repository -y ppa:ondrej/php  
PPA publishes dbgsym, you may need to include 'main/debug' component  
Repository: 'deb https://ppa.launchpadcontent.net/ondrej/php/ubuntu/ jammy main'  
Description:  
Co-installable PHP versions: PHP 5.6, PHP 7.x, PHP 8.x and most requested extensions are included. Only Su  
r Supported Ubuntu Releases (https://wiki.ubuntu.com/Releases) are provided. Don't ask for end-of-life PHP  
  
Debian oldstable and stable packages are provided as well: https://deb.sury.org/#debian-dpa  
  
You can get more information about the packages at https://deb.sury.org  
  
IMPORTANT: The <foo>-backports is now required on older Ubuntu releases.  
  
BUGS&FEATURES: This PPA now has a issue tracker:  
https://deb.sury.org/#bug-reporting
```

```
Adding repository.  
Adding deb entry to /etc/apt/sources.list.d/ondrej-ubuntu-php-jammy.list  
Adding disabled deb-src entry to /etc/apt/sources.list.d/ondrej-ubuntu-php-jammy.list  
Adding key to /etc/apt/trusted.gpg.d/ondrej-ubuntu-php.gpg with fingerprint 14AA40EC0831756D7F66C4F4EA0AAE5267A6C  
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease  
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]  
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]  
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]  
Get:5 https://ppa.launchpadcontent.net/ondrej/php/ubuntu jammy InRelease [23.9 kB]  
Get:6 https://ppa.launchpadcontent.net/ondrej/php/ubuntu jammy/main amd64 Packages [115 kB]  
Get:7 https://ppa.launchpadcontent.net/ondrej/php/ubuntu jammy/main Translation-en [35.6 kB]  
Fetched 512 kB in 2s (205 kB/s)  
Reading package lists... Done  
ubuntu@ip-172-31-42-115:/var/www/html$
```

i-0824471f793e5106d (AWS\_EC2\_Project1)

PublicIPs: 3.110.85.213 PrivateIPs: 172.31.42.115

**Step 2: Now, install the mysql-client over the “EC2 Machine [AWS\_EC2\_Project1]” using this command:- “sudo apt install php5.6 mysql-client php5.6-mysqli”. Press “enter” from the keyboard to execute the command.**

```
Reading package lists... Done  
ubuntu@ip-172-31-42-115:/var/www/html$ sudo apt install php5.6 mysql-client php5.6-mysqli  
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:  
 libapache2-mod-php5.6 libpcre3 php-common php5.6-cli php5.6-common php5.6-json php5.6-opcache php5.6-readline  
Suggested packages:  
 php-pear  
The following NEW packages will be installed:
```

i-0824471f793e5106d (AWS\_EC2\_Project1)

PublicIPs: 3.110.85.213 PrivateIPs: 172.31.42.115

```
Unpacking php5.6-readline (5.6.40-68+ubuntu22.04.1+deb.sury.org+1) ...
Selecting previously unselected package php5.6-cli.
Preparing to unpack .../5-php5.6-cli_5.6.40-68+ubuntu22.04.1+deb.sury.org+1_amd64.deb ...
Unpacking php5.6-cli (5.6.40-68+ubuntu22.04.1+deb.sury.org+1) ...
Selecting previously unselected package libapache2-mod-php5.6.
Preparing to unpack .../6-libapache2-mod-php5.6_5.6.40-68+ubuntu22.04.1+deb.sury.org+1_amd64.deb ...
Unpacking libapache2-mod-php5.6 (5.6.40-68+ubuntu22.04.1+deb.sury.org+1) ...
Selecting previously unselected package mysql-client.
Preparing to unpack .../7-mysql-client_8.0.34-0ubuntu0.22.04.1_all.deb ...
Unpacking mysql-client (8.0.34-0ubuntu0.22.04.1) ...
Selecting previously unselected package php5.6.
Preparing to unpack .../8-php5.6_5.6.40-68+ubuntu22.04.1+deb.sury.org+1_all.deb ...
Unpacking php5.6 (5.6.40-68+ubuntu22.04.1+deb.sury.org+1) ...
Selecting previously unselected package php5.6-mysql.
Preparing to unpack .../9-php5.6-mysql_5.6.40-68+ubuntu22.04.1+deb.sury.org+1_amd64.deb ...
Unpacking php5.6-mysql (5.6.40-68+ubuntu22.04.1+deb.sury.org+1) ...
Setting up php-common (2:93+ubuntu22.04.1+deb.sury.org+2) ...
Created symlink /etc/systemd/system/timers.target.wants/phpsessionclean.timer → /lib/systemd/system/phpsessionclean.timer.
Setting up php5.6-common (5.6.40-68+ubuntu22.04.1+deb.sury.org+1) ...
[...]
Progress: [ 60%] [#########################################.....
```

i-0824471f793e5106d (AWS\_EC2\_Project1)  
PublicIPs: 3.110.85.213 PrivateIPs: 172.31.42.115

**Step 3: The PHP & MySQL dependencies will be successfully installed.**  
**Check your web page by refreshing the button.**

```
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Processing triggers for php5.6-cli (5.6.40-68+ubuntu22.04.1+deb.sury.org+1) ...
Processing triggers for libapache2-mod-php5.6 (5.6.40-68+ubuntu22.04.1+deb.sury.org+1) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

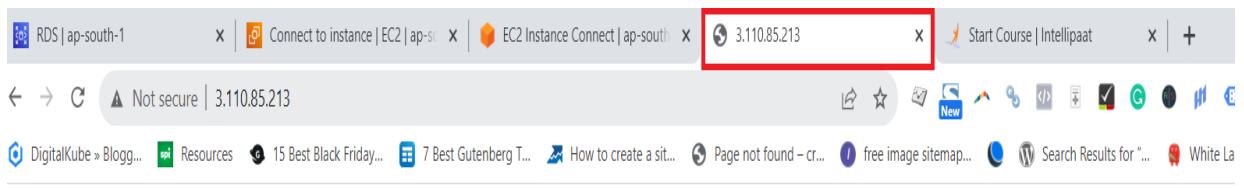
Restarting services...
systemctl restart packagekit.service polkit.service
Service restarts being deferred:
systemctl restart networkd-dispatcher.service
systemctl restart unattended-upgrades.service

No containers need to be restarted.

No user sessions are running outdated binaries.

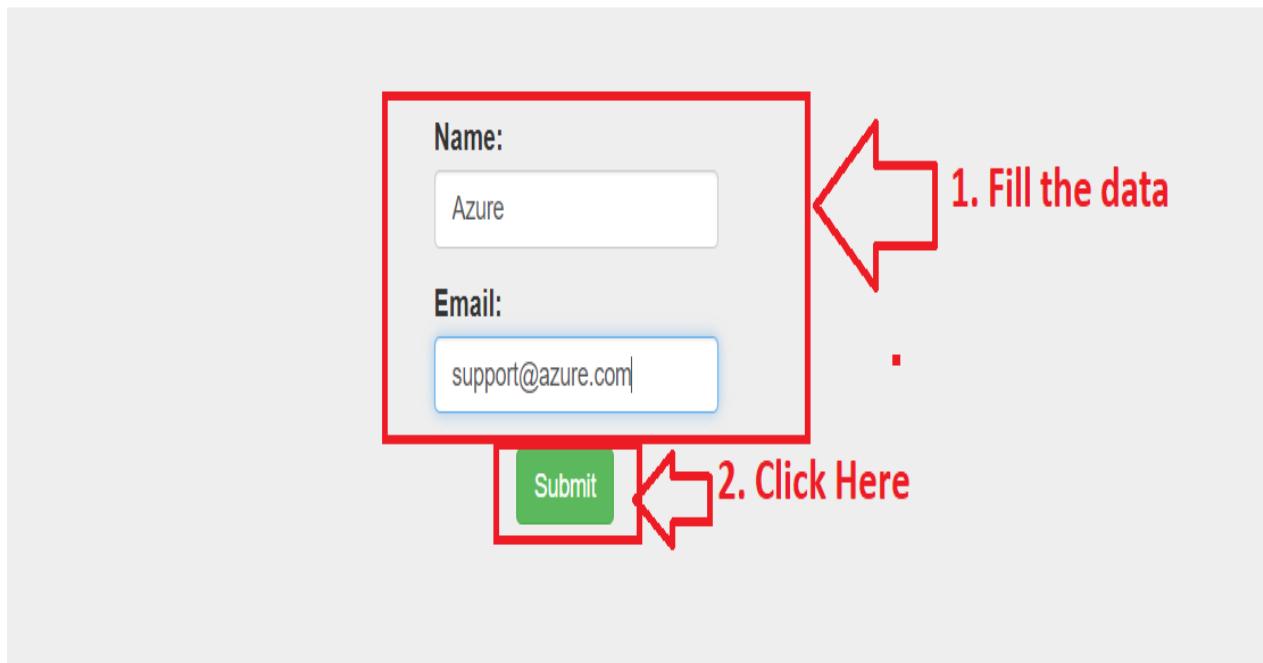
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-42-115:/var/www/html$
```

i-0824471f793e5106d (AWS\_EC2\_Project1)  
PublicIPs: 3.110.85.213 PrivateIPs: 172.31.42.115



## D. Test the Database Connection with PHP Web Page Again

**Step 1:** Insert the “Name” and “Email” through the web page. Click on the “Submit” button.



**Step 2:** The record will be successfully created.

A screenshot of a web form. It contains two input fields: one for 'Name' and one for 'Email', both with placeholder text. Below the fields is a green 'Submit' button.

New record created successfully

**Step 3: Go to the “EC2” Machine & connect machine with database using the following command: sudo mysql -h application-database.calupg2c7sxe.ap-south-1.rds.amazonaws.com -u admin -pintel123**

**The database will be successfully connected.**

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ubuntu@ip-172-31-42-115:/var/www/html$ sudo mysql -h application-database.calupg2c7sxe.ap-south-1.rds.amazonaws.com -u admin -pintel123  
mysql: [Warning] Using a password on the command line interface can be insecure.  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 38  
Server version: 8.0.33 Source distribution  
  
Copyright (c) 2000, 2023, 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> |
```

**Step 4: Now, go to the “intel” database. Write the command (“use intel;”) here. Now, we will inside the “intel” database.**

```
mysql> use intel
Database changed
mysql> [REDACTED]
```

i-0824471f793e5106d (AWS\_EC2\_Project1)

Public IPs: 3.110.85.213 Private IPs: 172.31.42.115

**Step 5: Type the “select \* from data” query & you will notice that other record will be successfully inserted.**

```
Database changed
mysql> select * from data;
+-----+-----+
| firstname | email      |
+-----+-----+
| AWS       | support@aws.com |
| Azure     | support@azure.com |
+-----+-----+
2 rows in set (0.00 sec)

mysql> [REDACTED]
```

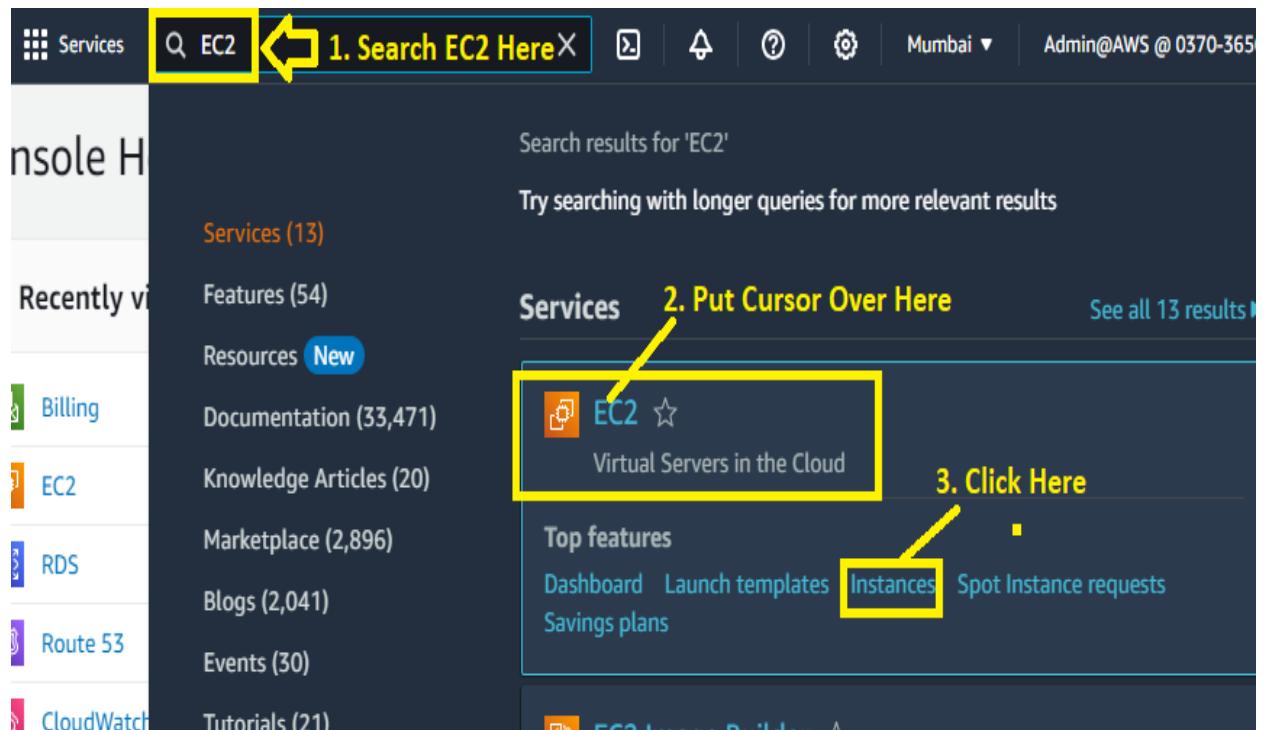
i-0824471f793e5106d (AWS\_EC2\_Project1)

Public IPs: 3.110.85.213 Private IPs: 172.31.42.115

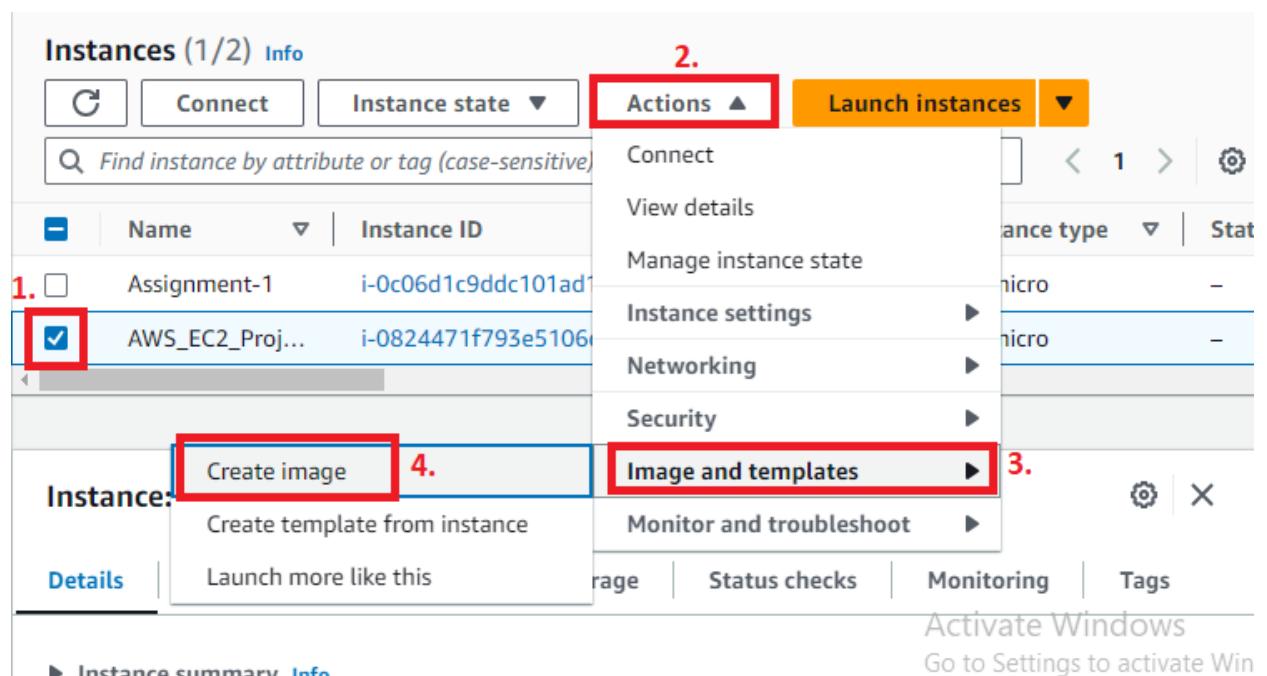
***So, our PHP website has been successfully configured with MySQL database.***

## 4. Create the AMI (Amazon Machine Image) for Launch Template

Step 1: Go to the “Services” section & search the “EC2”. Put cursor over “EC2” & click on the “Instances”.



Step 2: Select the Instance (“AWS\_EC2\_Project1”) & Go to the “Actions>Image and templates>Create image”.



### Step 3: Write “Image Name & Description” as “AWS\_EC2\_Project1\_Images”.

Create image [Info](#)

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from an existing instance or volume, or you can create one directly from a snapshot.

Instance ID  
i-0824471f793e5106d (AWS\_EC2\_Project1)

**Image name**  
AWS\_EC2\_Project1\_Images  
Maximum 127 characters. Can't be modified after creation.

**Image description - optional**  
AWS\_EC2\_Project1\_Images  
Maximum 255 characters

**Write Image Name & Description Here**

**Step 4: While, leave the other settings as it is. While in the “Tags-optional” section, choose the “Tag image and snapshots together”. Click on the “Create image”.**

Storage type	Device	Snapshot	Size	Volume type	IOPS
EBS	/dev/...	Create new snapshot fr...	8	EBS General Purpose S...	1

Add volume

**During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.**

**Tags - optional**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources.

Tag image and snapshots together  
Tag the image and the snapshots with the same tag.

**1.**

Tag image and snapshots together  
Tag the image and the snapshots with the same tag.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

search and filter your resources or track your AWS costs.

- Tag image and snapshots separately  
Tag the image and the snapshots with different tags.

2. Click Here

Activate Windows  
[Create image](#)  
Cancel  
Go to Settings to activate Windows.

**Step 5: The “AMI” will be start creating. Click on the “Hyperlink” in the green strip.**

Currently creating AMI [ami-00307ae5e3b18418c](#) from instance i-0824471f793e5106d. Check that the AMI status is 'Available' before related to this AMI.

[Click Here](#)

Instances (2) <a href="#">Info</a>		<a href="#">C</a>	<a href="#">Connect</a>	<a href="#">Instance settings</a>
		<a href="#">Find instance by attribute or tag (case-sensitive)</a>		
	Name	Instance ID	Instance state	Instance type
<input type="checkbox"/>	Assignment-1	i-0c06d1c9ddc101ad1	<a href="#">Stopped</a>	t2.micro
<input type="checkbox"/>	AWS_EC2_Proj...	i-0824471f793e5106d	<a href="#">Running</a>	t2.micro

**Step 6: In starting, your “AMI Status” will be “Pending”. It will take some time (3-5 Minutes) to show as “Available” status.**

Amazon Machine Images (AMIs) (1) [Info](#)

Owned by me [▼](#) [Find AMI by attribute or tag](#)

AMI ID = ami-00307ae5e3b18418c [X](#) [Clear filters](#)

Source	Owner	Visibility	Status	Creation date
037036564217/AWS_EC2_Project1_Im...	037036564217	Private	Pending	2023/10/04 1...

Status as "Pending"

**Step 7: Now, the “AMI Status” is showing in the “Available” State.**

The screenshot shows the AWS Lambda console with the heading "Amazon Machine Images (AMIs) (1) Info". There are buttons for "C" (Create), "Recycle Bin", "EC2 Image Builder", "Actions", and "Launch". A search bar says "Find AMI by attribute or tag". Below it, a filter bar shows "Owned by me" and "AMI ID = ami-00307ae5e3b18418c". A "Clear filters" button is also present. The main table has columns: "Source", "Owner", "Visibility", "Status", and "Creation date". A single row is shown with "Source" as "037036564217/AWS\_EC2\_Project1\_Im...", "Owner" as "037036564217", "Visibility" as "Private", "Status" as "Available" (with a green checkmark icon), and "Creation date" as "2023/10/04 14:20 GMT+5:30". A red box highlights the "Available" status, and a red arrow points from the text "Available Status" above to this box.

**Step 8: Scroll to the left & click on the “Edit Icon”.**

**Choose your “Name” as “AWS\_EC2\_Project1\_AMI”. While the “Name” is a tag for “AMI”.**

The “Name” will be reflect after the AMI ID like this: **ami-00307ae5e3b18418c (AWS\_EC2\_Project1\_AMI)**.

The screenshot shows the AWS Lambda console with the heading "Amazon Machine Images (AMIs) (1/1) Info". There are buttons for "C" (Create), "Recycle Bin", "EC2 Image Builder", and "Actions". A search bar says "Find AMI by attribute or tag". Below it, a filter bar shows "Owned by me" and "AMI ID = ami-00307ae5e3b18418c". A "Clear filters" button is also present. The main table has columns: "Name", "AMI ID", "AMI name", and "Source". A single row is shown with "Name" as "ami-00307ae5e3b18418c", "AMI ID" as "ami-00307ae5e3b18418c", "AMI name" as "AWS\_EC2\_Project1\_Images", and "Source" as "037036564217/AWS\_EC2\_Project1\_Im...". A red box highlights the "Name" column header, and a red arrow points from the text "Click Here" above to the edit icon in the "Name" column of the first row.

The screenshot shows the AWS Lambda console with the heading "Amazon Machine Images (AMIs) (1/1) Info". There are buttons for "C" (Create), "Recycle Bin", "EC2 Image Builder", and "Actions". A search bar says "Find AMI by attribute or tag". Below it, a filter bar shows "Owned by me" and "AMI ID = ami-00307ae5e3b18418c". A "Clear filters" button is also present. The main table has columns: "Name", "AMI ID", "AMI name", and "Source". A single row is shown with "Name" as "AWS\_EC2\_Proj...", "AMI ID" as "ami-00307ae5e3b18418c", "AMI name" as "AWS\_EC2\_Project1\_Images", and "Source" as "037036564217/AWS\_EC2\_Project1\_Im...". A red box highlights the "Name" column of the first row. Below the table, the text "Name after AMI Id" is written, with a red arrow pointing from the "Name" text to the "Name" column of the table. At the bottom, a box highlights the "AMI ID" field with the value "ami-00307ae5e3b18418c (AWS\_EC2\_Project1\_AMI)".

## 5. Create the Launch Template for Auto Scaling Groups

Step 1: Go to the “Launch Template”.

The screenshot shows the AWS EC2 Instances page. On the left sidebar, under the 'Instances' section, the 'Launch Templates' option is highlighted with a red box and labeled 'Click Here'. The main content area displays 'Amazon Machine Images (AMIs) (1)' with a single entry: 'AMI ID = ami-00307ae5e3b18418c' and 'Name = AWS\_EC2\_Proj...'. Below this is a section titled 'Select an AMI'.

Step 2: Click on the “Create launch template”.

The screenshot shows the 'EC2 launch templates' landing page. It features a large title 'EC2 launch templates' and a subtitle 'Streamline, simplify and standardize instance launches'. Below the title, there is a description of the benefits of using launch templates. On the right side, there is a call-to-action button labeled 'Create launch template' with a yellow arrow pointing to it, and the text 'Click here' above it.

Step 3: Fill these options in the “Launch template name and description” in the “Create launch template”.

**Launch template name** - AWS\_EC2\_Project\_Template

**Template version description** - A launch template for EC2 Project

**Enable the option “Provide guidance to help me set up a template that I can use with EC2 Auto Scaling” by clicking in the “Auto Scaling guidance” option.**

## Launch template name and description

Launch template name - required

AWS\_EC2\_Project\_Template

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

Template version description

A launch template for EC2 Project

Max 255 chars

1. Write Template Name Here

2. Write Template Description Here

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

3. Enable this option by clicking

► Template tags

**Step 4: In the “Application and OS Images (Amazon Machine Image)- required”, choose the “My AMI” option. Select your created AMI here.**

### ▼ Application and OS Images (Amazon Machine Image) - required [Info](#)

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

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

1.

Recents

My AMIs

Quick Start

Owned by me

2.

Shared with me



Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI) 3.

AWS\_EC2\_Project1\_Images

ami-00307ae5e3b18418c

2023-10-04T08:50:41.000Z Virtualization: hvm ENA enabled: true Root device type: ebs

#### Amazon Machine Image (AMI)

AWS\_EC2\_Project1\_Images  
ami-00307ae5e3b18418c  
2023-10-04T08:50:41.000Z Virtualization: hvm ENA enabled: true Root device type: ebs

#### Description

AWS\_EC2\_Project1\_Images

#### Architecture

x86\_64

#### AMI ID

ami-00307ae5e3b18418c

**Step 5: Choose the “instance type” as the “t2.micro”, while key pair as yours key pair name “MY-WEB-SERVER”.**

Instance type

1. Choose t2.micro here

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: 0.0724 USD per Hour	
On-Demand SUSE base pricing: 0.0124 USD per Hour	

All generations

Compare instance types

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 launch the instance.

2. Choose your key pair name

Key pair name

MY-WEB-SERVER

Create new key pair

**Step 6: While leave the “Subnet” as it is. No need to choose here. While in the “Firewall (security groups)”, choose the “Select existing security group”. While choose your created security group (AWS\_EC2\_Project1) here for this project.**

▼ Network settings [Info](#)

Subnet [Info](#)

Don't include in launch template [▼](#) [Create new subnet](#)

When you specify a subnet, a network interface is automatically added to your template.

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.

1.  Select existing security group  Create security group

Security groups [Info](#)

Select security groups [▼](#)

AWS\_EC2\_Project1 sg-04d792595bc10f95d VPC: vpc-04c3e6ef2b9c7b60f

2. Choose your security group here

▼ Advanced network configuration

### Step 7: Click on the “Create launch template”.

le it in the launch template.

Firewall (security group)  
AWS\_EC2\_Project1

Storage (volumes)  
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instances

[Click Here](#)

[Cancel](#) [Create launch template](#) [Activate Windows](#)

Step 8: Your launch template (AWS\_EC2\_Project\_Template(It-0798f90e53f8c8c27)) will be successfully created. Click on the hyperlink.

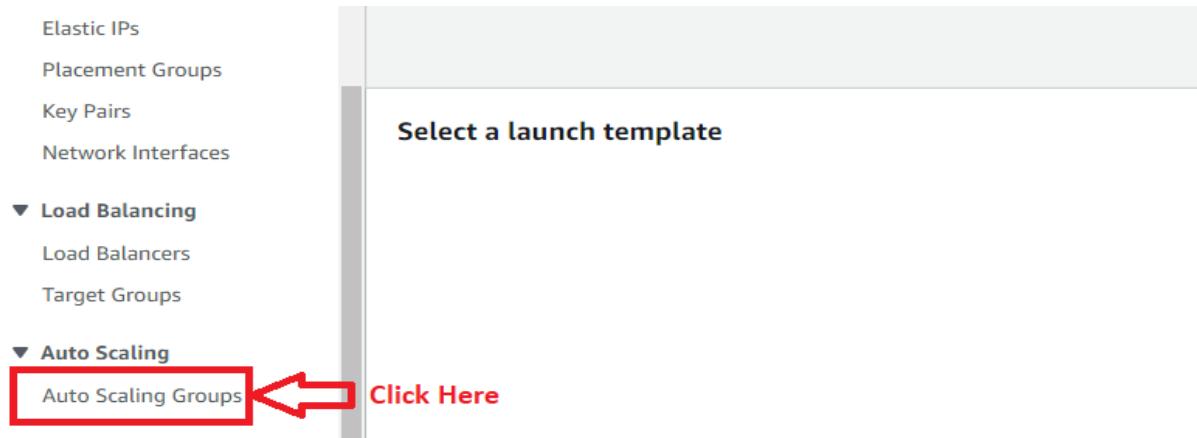
The screenshot shows the AWS EC2 Launch Templates creation interface. At the top, there's a navigation bar with 'Services' and a search bar. Below it, the path 'EC2 > Launch templates > Create launch template' is shown. A green success message box contains the text 'Success' with a checkmark icon and 'Successfully created AWS EC2 Project Template(lt-0798f90e53f8c8c27)'. A yellow arrow points from the text 'Click Here' to the hyperlink in the message. Below the message, there's a link to 'Actions log' and a 'Next step' button.

Your launch template description will be shown after clicking on the hyperlink.

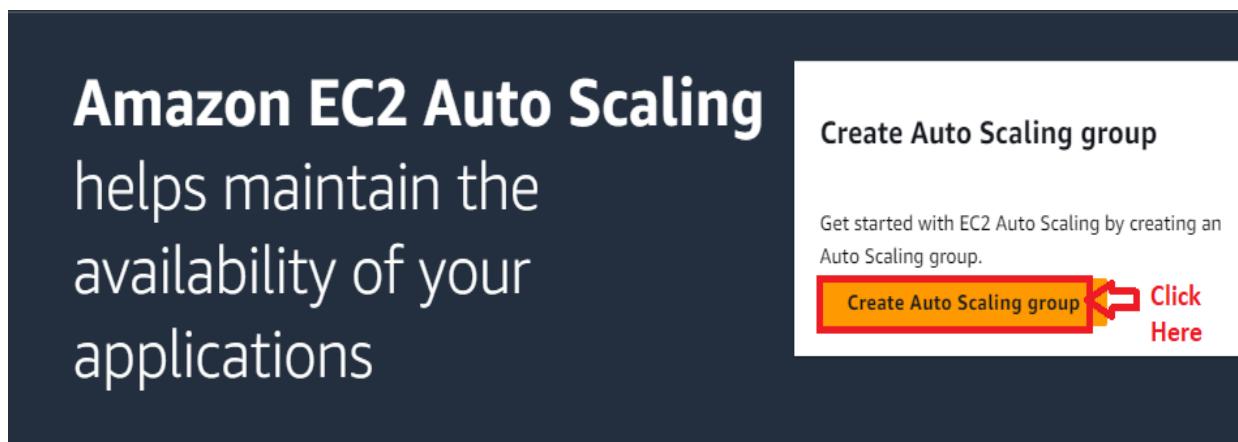
The screenshot shows the AWS EC2 Launch Template details page for 'AWS\_EC2\_Project\_Template (lt-0798f90e53f8c8c27)'. The template name is highlighted with a red box. On the right, there are 'Actions' and 'Delete template' buttons. Below the template name, a 'Launch template details' section is shown with fields: Launch template ID (lt-0798f90e53f8c8c27), Launch template name (AWS\_EC2\_Project\_Template), Default version (1), and Owner (arn:aws:iam::037036564217:user/Admin@AWS). The 'Details' tab is selected. At the bottom, a 'Launch template version details' section is shown with a dropdown for Version (1 (Default)), Description (A launch template for EC2 Project), Date created (2023-10-04T10:05:50.000Z), and Created by (arn:aws:iam::037036564217:user/Admin@AWS). The 'Delete template version' button is also visible.

## 6. Create an Auto Scaling Group for this PHP website

Step 1: Go to the left side & click on the “Auto Scaling Groups” in the “Auto Scaling”.



Step 2: Click on the “Create Auto Scaling group”.



Step 3: Choose the “Auto Scaling group name” as “AWS\_EC2\_Project1\_ASG”.

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1  
Choose launch template or configuration

Step 2  
Choose instance launch options

Step 3 - optional  
Configure advanced options

Step 4 - optional  
Configure group size and scaling policies

Choose launch template or configuration Info

Specify a launch template that contains settings common to all EC2 instances that are launched in this group. If you currently use launch configurations, you might consider migrating to launch templates.

Name	Write Auto Scaling Group Name Here
Auto Scaling group name	Enter a name to identify the group.
AWS_EC2_Project1_ASG	
Must be unique to this account in the current Region and no more than 255 characters.	

**Step 4: Choose the “Launch Template” as “AWS\_EC2\_Project\_Template”. All the “launch template” configuration will be automatically fetched.**

Launch template [Info](#) [Switch to launch configuration](#)

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.

AWS\_EC2\_Project\_Template  

Create a launch template 

Version  
Default (1)  

**Choose launch template here**

Create a launch template version 

Description A launch template for EC2 Project	Launch template <a href="#">AWS_EC2_Project_Template</a>  lt-0798f90e53f8c8c27	Instance type t2.micro
AMI ID ami-00307ae5e3b18418c	Security groups -	Request Spot Instances No
Key pair name MY-WEB-SERVER	Security group IDs <a href="#">sg-04d792595bc10f95d</a> 	

**Step 5: Click on “Next”.**

Date created  
Wed Oct 04 2023 15:35:50  
GMT+0530 (India Standard Time)

**Click Here**



Cancel **Next**  Go to Settings to

**Step 6: In the “Choose instance launch options”, the “Launch template” features will be shown in “Instance type requirements”.**

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1  
Choose launch template or configuration

Step 2  
Choose instance launch options

Step 3 - optional  
Configure advanced options

Step 4 - optional  
Configure group size and scaling policies

Step 5 - optional  
Add notifications

**Choose instance launch options**

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

**Instance type requirements**

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

Override launch template

Launch template	Version	Description
AWS_EC2_Project_Template	Default	A launch template for EC2 Project
lt-0798f90e53f8c8c27		
Instance type	t2.micro	

**Step 7: In the “Networks”, leave the “VPC” settings as it is. While in the “Availability zones and subnets”, choose the “ap-south-1a zone with the default subnet” & the “ap-south-1b” zone with created subnet (Zone\_B).**

**Network**

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across 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-04c3e6ef2b9c7b60f    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

ap-south-1a | subnet-0012fe787fe8d6766    172.31.32.0/20    Default

ap-south-1b | subnet-00cf17266e1b37cef (Zone\_B)    172.31.16.0/20

Create a subnet

Cancel Skip to review Previous Next

**Click on the “Next”.**

**Step 8: Choose the “No load balancer” option in “Load balancing” option. Click on the “Next”.**

**Configure advanced options - optional** Info

Choose a load balancer to distribute incoming traffic for your application across instances to make it more reliable and easily scalable. You can also set options that give you more control over health check replacements and monitoring.

**Load balancing** Info **Choose this option**

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.

**Health check grace period** Info  
This time period delays the first health check until your instances finish initializing. It doesn't prevent an instance from terminating when placed into a non-running state.  
300 seconds

**Additional settings**

**Monitoring** Info  
 Enable group metrics collection within CloudWatch

**Default instance warmup** Info  
The amount of time that CloudWatch metrics for new instances do not contribute to the group's aggregated instance metrics, as their usage data is not reliable yet.  
 Enable default instance warmup

**Click Here** 

**Next** 

Cancel Skip to review Previous

**Step 9: In the “group size” choose the following metrics:**

**Desired capacity – 2**

**Minimum capacity – 1**

**Maximum capacity – 3**

### Configure group size and scaling policies - *optional* Info

Set the desired, minimum, and maximum capacity of your Auto Scaling group. You can optionally add a scaling policy to dynamically scale the number of instances in the group.

#### Group size - *optional* Info

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity	<input type="text" value="2"/>
Minimum capacity	<input type="text" value="1"/>
Maximum capacity	<input type="text" value="3"/> <input type="button" value="▼"/>

**Choose "Desired" & "Maximum" Capacity**

**Step 10: Choose the “Scaling policies – optional” as “None”. Click on the “Next”.**

#### Scaling policies - *optional*

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. Info

Target tracking scaling policy

Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

None

**1. Choose this option**

#### Instance scale-in protection - *optional*

##### Instance scale-in protection

If protect from scale in is enabled, newly launched instances will be protected from scale in by default.

Enable instance scale-in protection

**2. Click Here**



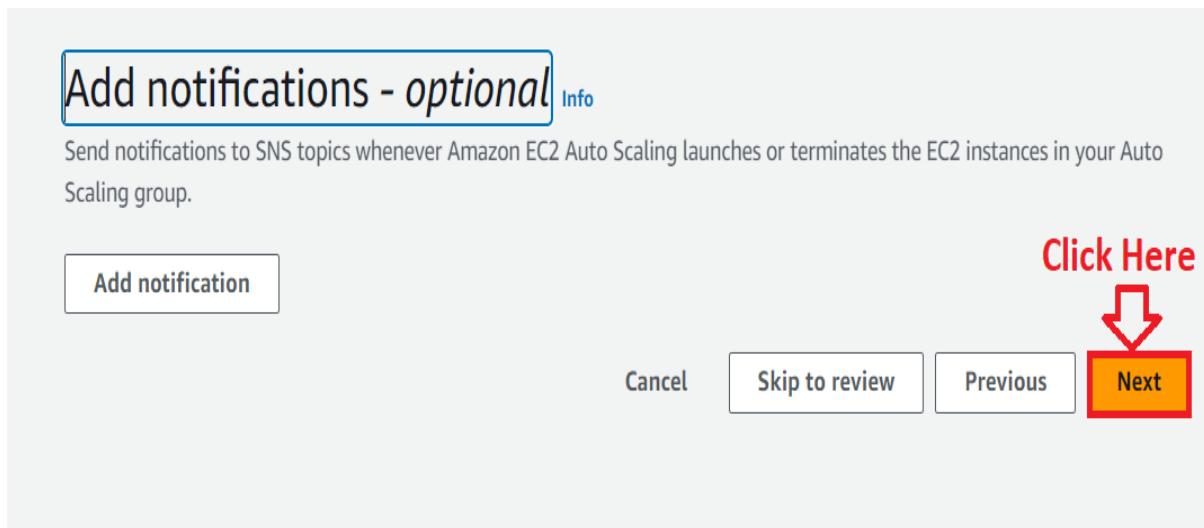
Cancel

Skip to review

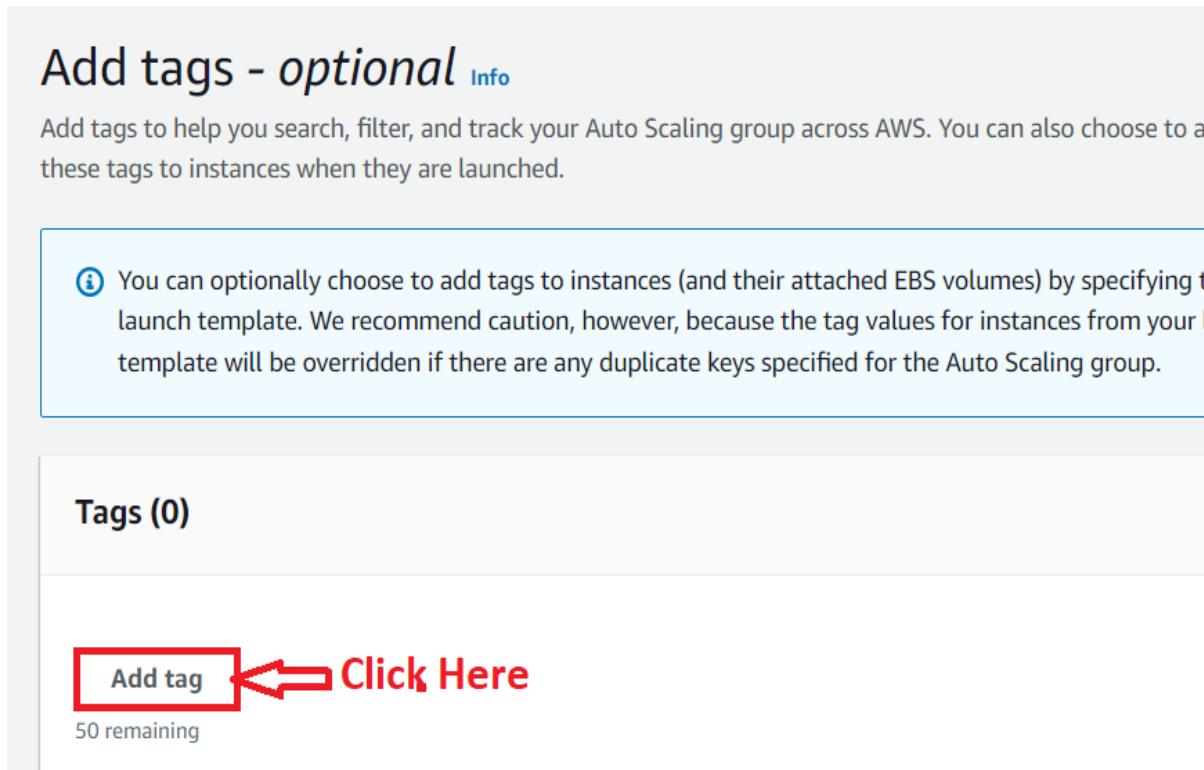
Previous

**Next**

**Step 11: Click on the “Next” in the “Add notifications-optional”.**



**Step 12: Click on the “Add tag”.**



**Step 13: Choose the following options here:**

**Key – Name**

**Value-optional: - asg-autoscaling-grp**

**Tag new instances – Remain enable**

**Click on “Next”.**

## Add tags - optional Info

Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched.

ⓘ You can optionally choose to add tags to instances (and their attached EBS volumes) by specifying tags in your launch template. We recommend caution, however, because the tag values for instances from your launch template will be overridden if there are any duplicate keys specified for the Auto Scaling group. X

Tags (1)

1.

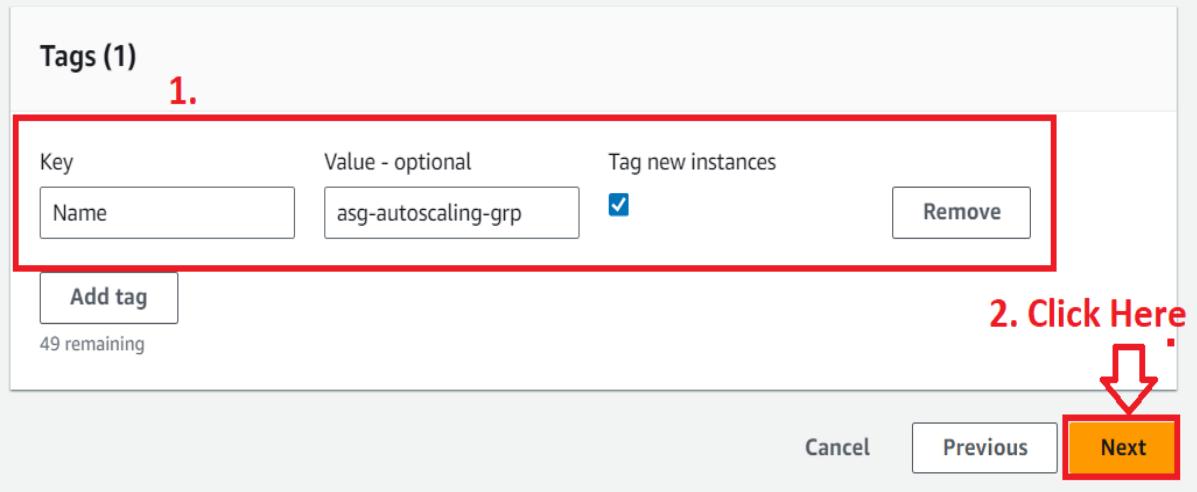
Key	Value - optional	Tag new instances	Remove
Name	asg-autoscaling-grp	<input checked="" type="checkbox"/>	

**Add tag**

49 remaining

2. Click Here

Cancel Previous **Next**



**Step 14:** In the “Review” section, scroll down and click on the “Create auto scaling group”.

No notifications

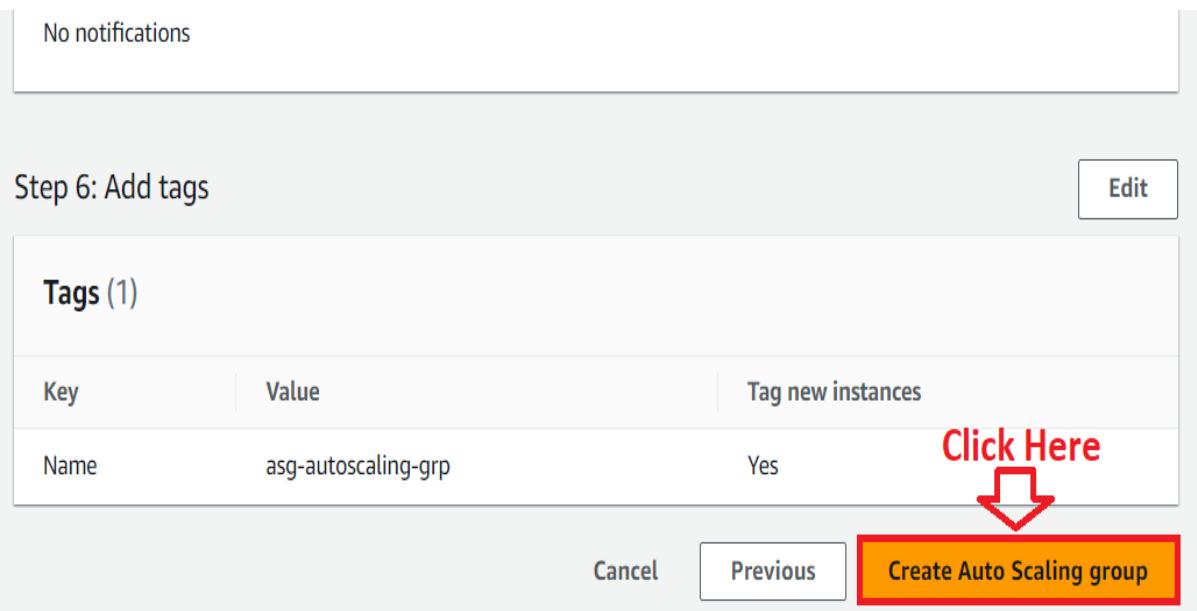
Step 6: Add tags Edit

Tags (1)

Key	Value	Tag new instances
Name	asg-autoscaling-grp	Yes

**Click Here**

Cancel Previous **Create Auto Scaling group**



**Step 15:** Your “Auto Scaling Group” will be created successfully. Click on the “AWS\_EC2\_Project1\_ASG”.

The screenshot shows the AWS EC2 Auto Scaling groups page. At the top, there is a breadcrumb navigation: EC2 > Auto Scaling groups. Below the navigation, a table lists one Auto Scaling group: "Auto Scaling groups (1) Info". A red arrow points from the text "Click Here" to the "Info" link. The table has columns for Name, Launch template/configuration, and Instances. The first row shows "Name" as "AWS\_EC2\_Project1\_ASG", "Launch template/configuration" as "AWS\_EC2\_Project\_Template | Version De 0", and "Instances" as "0".

**Step 16:** Your “Auto Scaling Group” details will be shown here & go to the “Activity” section.

The screenshot shows the AWS EC2 Auto Scaling group details page for "AWS\_EC2\_Project1\_ASG". The breadcrumb navigation is EC2 > Auto Scaling groups > AWS\_EC2\_Project1\_ASG. Below the navigation, there is a navigation bar with tabs: Details, Activity, Automatic scaling, Instance management, Monitoring, and Instance refresh. A red box highlights the "Activity" tab, and a red arrow points from the text "Click Here" to the tab name. The main content area is titled "Group details" and contains a table with the following data:

Auto Scaling group name	Desired capacity	Status
AWS_EC2_Project1_ASG	2	Updating capacity
Date created	Minimum capacity	
Fri Oct 06 2023 06:48:28 GMT+0530 (India Standard Time)	1	
	Maximum capacity	
	3	

**Step 17:** The “EC2 Auto Scaling Group” will be successfully launched your instances.

The screenshot shows the AWS EC2 Auto Scaling group activity history page for "AWS\_EC2\_Project1\_ASG". The breadcrumb navigation is EC2 > Auto Scaling groups > AWS\_EC2\_Project1\_ASG > Activity history (2). Below the navigation, there is a search bar labeled "Filter activity history" and a red arrow pointing to it with the text "Instances". The main content area is titled "Activity history (2)" and contains a table with the following data:

Status	Description	Cause	Start time
Successful	Launching a new EC2 instance: i-05aad03887b7ab07	At 2023-10-06T01:18:28Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2023-10-06T01:18:33Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.	2023 October 06, 06:48:35 AM +05:30
Successful	Launching a new EC2 instance: i-06954abe8b725bb84	At 2023-10-06T01:18:28Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2023-10-06T01:18:33Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.	2023 October 06, 06:48:35 AM +05:30

**Step 18: Click on the “EC2”.**

The screenshot shows the AWS EC2 Activity notifications page. At the top, there is a breadcrumb navigation: EC2 > Auto Scaling groups > AWS\_EC2\_Project1\_ASG. A red box highlights the 'EC2' link, and a red arrow points to it with the text 'Click Here'. Below the breadcrumb, the title 'AWS\_EC2\_Project1\_ASG' is displayed. A navigation bar at the top includes tabs for Details, Activity (which is underlined in blue), Automatic scaling, Instance management, Monitoring, and Instance refresh. Under the 'Activity' tab, the section 'Activity notifications (0)' is shown, along with a search bar labeled 'Filter notifications' and a 'Send to' button. To the right, there is a dropdown menu and a link 'On instance action'. A note at the bottom states 'No notifications are currently specified'.

**Step 19: Go to the “Instances (running)”.**

The screenshot shows the AWS EC2 Resources page. At the top, there is a 'Resources' tab and an 'EC2' link. A red box highlights the 'Resources' tab, and a red arrow points to it with the text 'Click Here'. Below the tabs, a message says 'You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region'. A grid of resource counts is shown:

Instances (running)	3	Auto Scaling Groups	1
Elastic IPs	0	Instances	4
Load balancers	0	Placement groups	0
Snapshots	1	Volumes	4

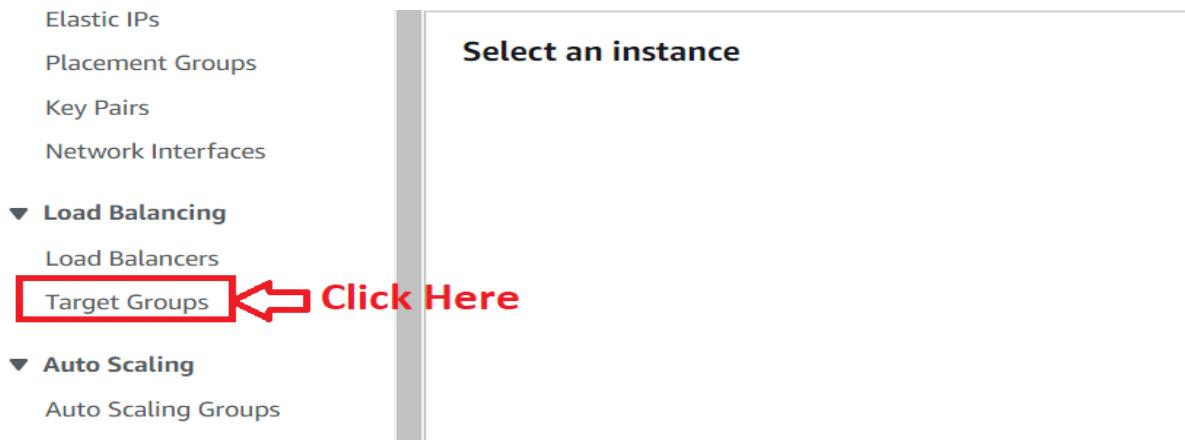
**Step 20: You will notice that your Auto Scaling Group Instances are successfully running.**

The screenshot shows the AWS EC2 Instances page. At the top, there is a header 'Instances (3) Info' with a 'Find instance by attribute or tag (case-sensitive)' search bar, a 'Connect' button, and a 'Instance state' dropdown. A red box highlights the 'Info' link, and a red arrow points to it with the text 'Auto Scaling Instances'. Below the header, a table lists three instances:

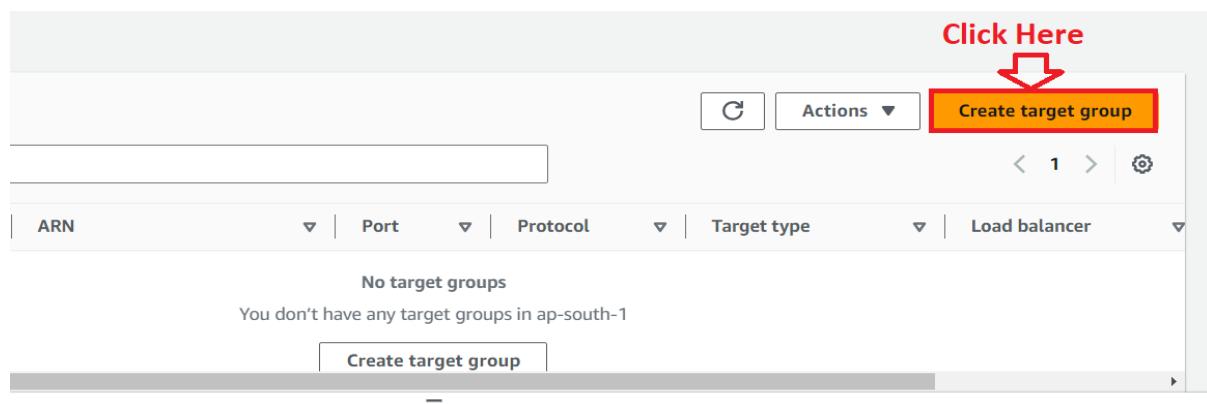
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	asg-autoscalin...	i-06954abe8b725bb84	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	No alarms
<input type="checkbox"/>	AWS_EC2_Proj...	i-0824471f793e5106d	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	No alarms
<input type="checkbox"/>	asg-autoscalin...	i-05aad03887b7ab07	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	No alarms

## 7. Create a Target Group for Load Balancer

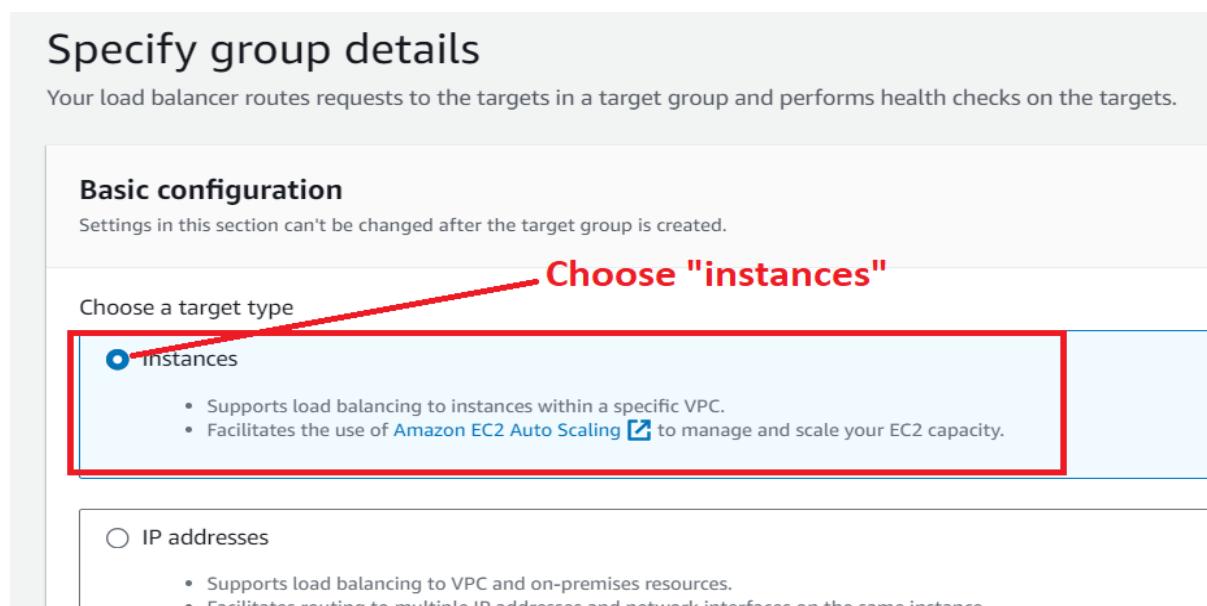
Step 1: In the left side, scroll down. Click on the “Target Groups” in the “Load Balancing”.



Step 2: Click on the “Create target group”.



Step 3: Choose the “instances” as “Choose a target type”.



## Step 4: Choose the “Target Group Name” as the “EC2-Project-Target-Group”. Leave the other settings as by default.

Target group name  
EC2-Project-Target-Group

Protocol Port  
HTTP : 80  
1-65535

IP address type  
Only targets with the indicated IP address type can be included in this target group.  
 IPv4  
Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.  
 IPv6  
Each target you register must have an assigned primary IPv6 address. This is configured on the instances default network interface (eth0). [Learn more](#)

VPC  
Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.  
-  
vpc-04c3e6ef2b9c7b60f  
IPv4: 172.31.0.0/16

Protocol version  
 HTTP1

**1. Choose Target Group Name Here**

**2. Leave as it is, No change here**

## Step 5: Click on the “Next”.

**Attributes**

**Info** Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

► **Tags - optional**

Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

**Click Here**

**Cancel** **Next**

**Step 6: Choose the third instance (AWS\_EC2\_Project1) as an “Available Instance”, because others are “Auto Scaling Groups” instances, you can delete those & they will come again as it is. While “AWS\_EC2\_Project1” is the main instance. From which we do the traffic flow to other instances.**

**Click on the “Include as pending below”.**

Available instances (1/3)

Instance ID	Name	State	Security groups
i-06954abe8b725bb84	asg-autoscaling-grp	Running	AWS_EC2_Project1
i-05aadc03887b7ab07	asg-autoscaling-grp	Running	AWS_EC2_Project1
<input checked="" type="checkbox"/> i-0824471f793e5106d	AWS_EC2_Project1	Running	ec2-rds-2, AWS_EC2_Project1

1 selected

Ports for the selected instances  
Ports for routing traffic to the selected instances.  
80  
1-65535 (separate multiple ports with commas)

**1. Choose this Instance**

**2. Click Here**

**Include as pending below**

**Step 7: In the “Review targets”, your instance will be added. Click on the “Create target group”.**

Targets (1)

Remove	Health status	Instance ID	Name	Port	State	Security groups	Zone
X	Pending	i-0824471f793e5106d	AWS_EC2_Project1	80	Running	ec2-rds-2, AWS_EC2_Project1	ap-south1

1 pending

**Click Here**

**Create target group**

**Step 8: Your target group (EC2-Project-Target-Group) will be successfully created.**

⌚ Successfully created target group: EC2-Project-Target-Group

EC2 > Target groups

Target groups (1) [Info](#)

Filter target groups

Name	ARN	Port	Protocol	Target type
EC2-Project-Target-Group	arn:aws:elasticloadbalancing:ap-south-1:037036564217:targetgroup/EC2-Project-Target-Group/70a42334c12f2a25	80	HTTP	Instance

**Step 9: Click on the “EC2-Project\_Target-Group”. You will notice that “One Healthy” target has been registered.**

**Details**

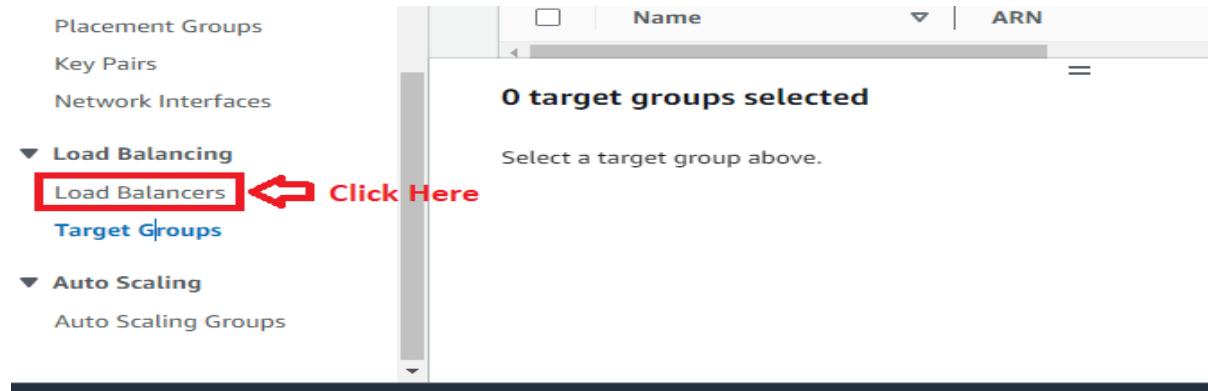
arn:aws:elasticloadbalancing:ap-south-1:037036564217:targetgroup/EC2-Project-Target-Group/70a42334c12f2a25

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1
IP address type IPv4	Load balancer <a href="#">None associated</a>	
Total targets 1	Healthy ⌚ 0	Unhealthy ✖ 0
		Unused 🕒 1

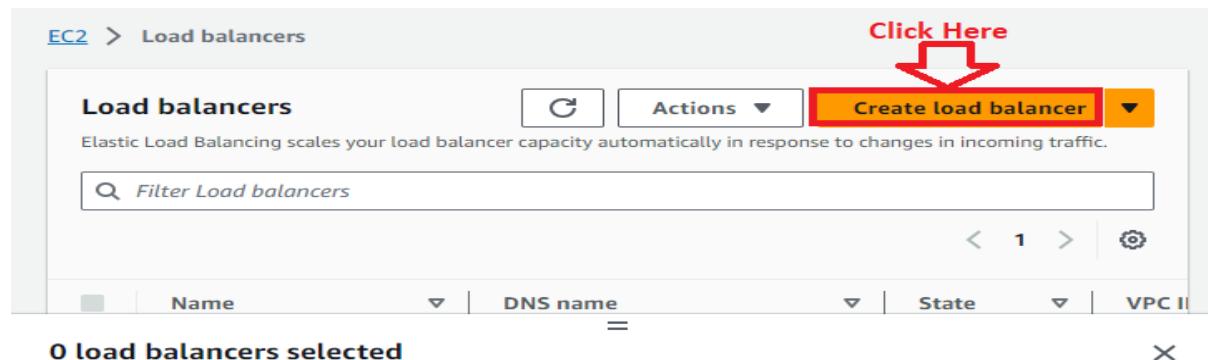
*In the “Load Balancer” section, you will notice that “None associated”, now we will create a load balancer. Which will receive the traffic & send the request to the targets one by one. If target is unhealthy, load balancer will not send the traffic.*

## 8. Create a Load Balancer for Balancing the Traffic

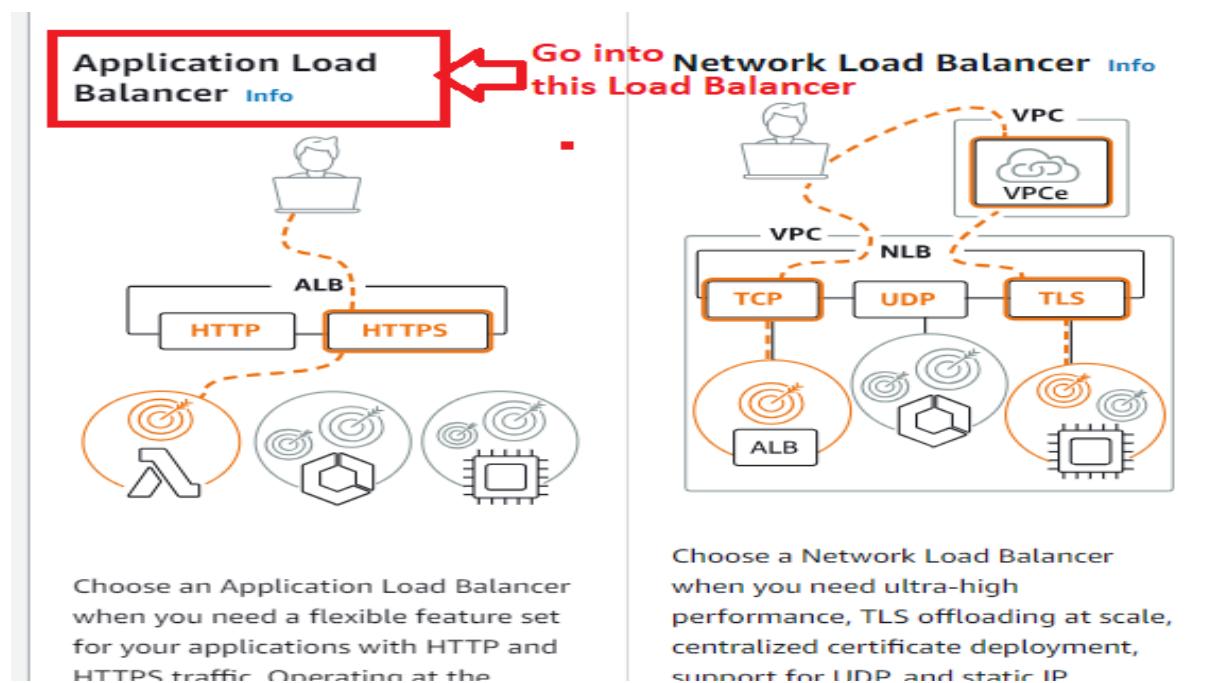
Step 1: Go to the left side & click on the “Load Balancers” in the “Load Balancing” option.



Step 2: Click on the “Create load balancer”.



Step 3: Go to the “Application Load Balancer” & click on the “Create”.



Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Create

Click here

Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

Create

**Step 4:** In the “Basic Configuration”, choose the “Load balancer name” as the “AWS-EC2-Project-Load-Balancer”. While leave the “scheme address” as the “Internet-facing” & the “IP address type” as “IPv4”.

#### Basic configuration

##### 1. Write Load Balancer Name here

###### Load balancer name

Name must be unique within your AWS account and can't be changed after the load balancer is created.

AWS-EC2-Project-Load-Balancer

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

###### Scheme | Info

Scheme can't be changed after the load balancer is created.

###### Internet-facing

An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

###### Internal

An internal load balancer routes requests from clients to targets using private IP addresses.

###### IP address type | Info

Select the type of IP addresses that your subnets use.

###### IPv4

Recommended for internal load balancers.

###### Dualstack

Includes IPv4 and IPv6 addresses.

2. Remain Options as it is

Activat

Cancel

## Step 5: In the “Network Mapping”, choose the “ap-south-1a” & “ap-south-1b” zones with its subnets.

Mappings | Info

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability balancer or the VPC are not available for selection.

1.  ap-south-1a (aps1-az1)

2. Subnet  
subnet-0012fe787fe8d6766

IPv4 address  
Assigned by AWS

3.  ap-south-1b (aps1-az3)

4. Subnet  
subnet-00cf17266e1b37cef Zone\_B

IPv4 address  
Assigned by AWS

## Step 6: Choose the “Security groups” as created in the beginning of EC2 Creation. We have created the “AWS\_EC2\_Project1” & choose this security group here.

Security groups | Info

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can create a new security group [Create](#).

**Choose your security group here**

Security groups

Select up to 5 security groups

AWS\_EC2\_Project1 X  
sg-04d792595bc10f95d VPC: vpc-04c3e6ef2b9c7b60f

**Step 7: In “Listeners and routing”, choose your “Target Group (EC2-Project-Target Group)”.**

The screenshot shows the 'Handler' section of a Lambda function configuration. It includes fields for 'Handler' (set to 'lambda\_function.lambda\_handler'), 'File' (set to 'lambda\_function'), and 'Runtime' (set to 'python3.8'). A red box highlights the 'Handler' field, and a red arrow points to it with the text 'Choose "Handler" Here'.

**Step 8: Leave other settings as it is. Click on “Create load balancer”.**

The screenshot shows a confirmation message: "Your load balancer has been successfully created. You can view and edit them after creating the load balancer." Below this, there is a 'Create load balancer' button highlighted with a red box and a red arrow pointing to it with the text 'Click Here'.

**Step 9: Your load balancer will be successfully created. Click on the “Hyperlink” to view the load balancer details.**

The screenshot shows the Lambda function details page. A green banner at the top states: "Successfully created load balancer: AWS-EC2-Project-Load-Balancer". A red arrow points to this message with the text 'Click Here'. Below the banner, a note says: "Note: It might take a few minutes for your load balancer to be fully set up and ready to route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks." At the bottom of the page, there is a 'Suggested next steps' section.

**Step 10: You will notice that “Load Balancer” will be successfully created & in “Active” state.**

The screenshot shows the AWS Load Balancers console. At the top, there is a breadcrumb navigation: EC2 > Load balancers. Below the navigation, a header bar contains "Load balancers (1)", a "Create load balancer" button, and an "Actions" dropdown. A message states: "Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic." A search bar labeled "Filter Load balancers" is present. Below the header is a table with columns: Name, DNS name, State, and VPC ID. A single row is listed, highlighted with a red box. The row details are: Name: AWS-EC2-Project-Load-Balancer, DNS name: AWS-EC2-Project-Load-Bal..., State: Active (with a green checkmark), and VPC ID: vpc-0... .

## 9. Attach the Load Balancer to Auto Scaling Group

**Step 1: Go to the left side & click on the “Auto Scaling Groups” in the “Auto Scaling”.**

The screenshot shows the AWS Auto Scaling Groups console. On the left sidebar, there are several navigation items: Placement Groups, Key Pairs, Network Interfaces, Load Balancing (with a red arrow pointing to it), Load Balancers, Target Groups, Auto Scaling (with a red box around it), and Auto Scaling Groups. The main content area displays a table with columns: Name, DNS name, and State. One row is listed: Name: AWS-EC2-Project-Load-Balancer, DNS name: AWS-EC2-Project-Load-Bal..., State: Active. Below the table, a message says: "0 load balancers selected" and "Select a load balancer above."

**Step 2: Click on the “AWS\_EC2\_Project1\_ASG”.**

EC2 > Auto Scaling groups

Auto Scaling groups (1) [Info](#)

[Create Auto Scaling group](#)

Search your Auto Scaling groups

Name	Launch template/configuration	Instances
AWS_EC2_Project1_ASG	AWS_EC2_Project_Template   Version De 2	2

Step 3: Go to the “Details” section & scroll down.

EC2 > Auto Scaling groups > AWS\_EC2\_Project1\_ASG

## AWS\_EC2\_Project1\_ASG

Click Here

< Details Activity Automatic scaling Instance management More >

Group details		Edit
Auto Scaling group name	AWS_EC2_Project1_ASG	Desired capacity
		2
Date created	Fri Oct 06 2023 06:48:28 GMT+0530 (India Standard Time)	Minimum capacity
		1
		Maximum capacity
		3

Step 4: In the “Load balancing”, click on the “Edit”.

**Instance type requirements**

Your Auto Scaling group adheres to the launch template for purchase option and instance type.

**Load balancing**

Load balancer target groups      Classic Load Balancers

**Edit**

**Click here**

**Step 5: Choose the “Application, Network or Gateway Load Balancer target groups” option & select your target group (EC2-Project-Target\_Group) here.**

Click on the “Update”.

**Load balancing - optional**

**Load balancers**      **1. Choose this option**

Application, Network or Gateway Load Balancer target groups

Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups

EC2-Project-Target-Group | HTTP      X

Application Load Balancer: AWS-EC2-Project-Load-Balancer

Classic Load Balancers

Create and attach new load balancers

Add a new load balancer

**2. Select the target group here**

Cancel      **3. Click Here**

Update

**Step 6: Your load balancer will be successfully attached.**

The screenshot shows the AWS Auto Scaling console. At the top, a green banner displays the message "Auto Scaling group updated successfully". Below this, the "Load balancing" section is highlighted with a red box. It contains two tabs: "Load balancer target groups" (selected) which shows "EC2-Project-Target-Group", and "Classic Load Balancers" (disabled). There are "Edit" buttons for both sections. Below the load balancing section is another "Edit" button under the "Health checks" section.

## 10. Test the Load Balancer Will Working Fine or Not

**Step 1: Go to the left side & click on the “Load Balancers” in the “Load Balancing” option.**

The screenshot shows the AWS Lambda console. On the left sidebar, under the "Load Balancing" section, the "Load Balancers" option is selected and highlighted with a red box, with a red arrow pointing to it labeled "Click Here". Other options like "Target Groups" are also visible. The main area shows a table header with columns "Name" and "ARN". Below the header, a message says "0 target groups selected" and "Select a target group above.".

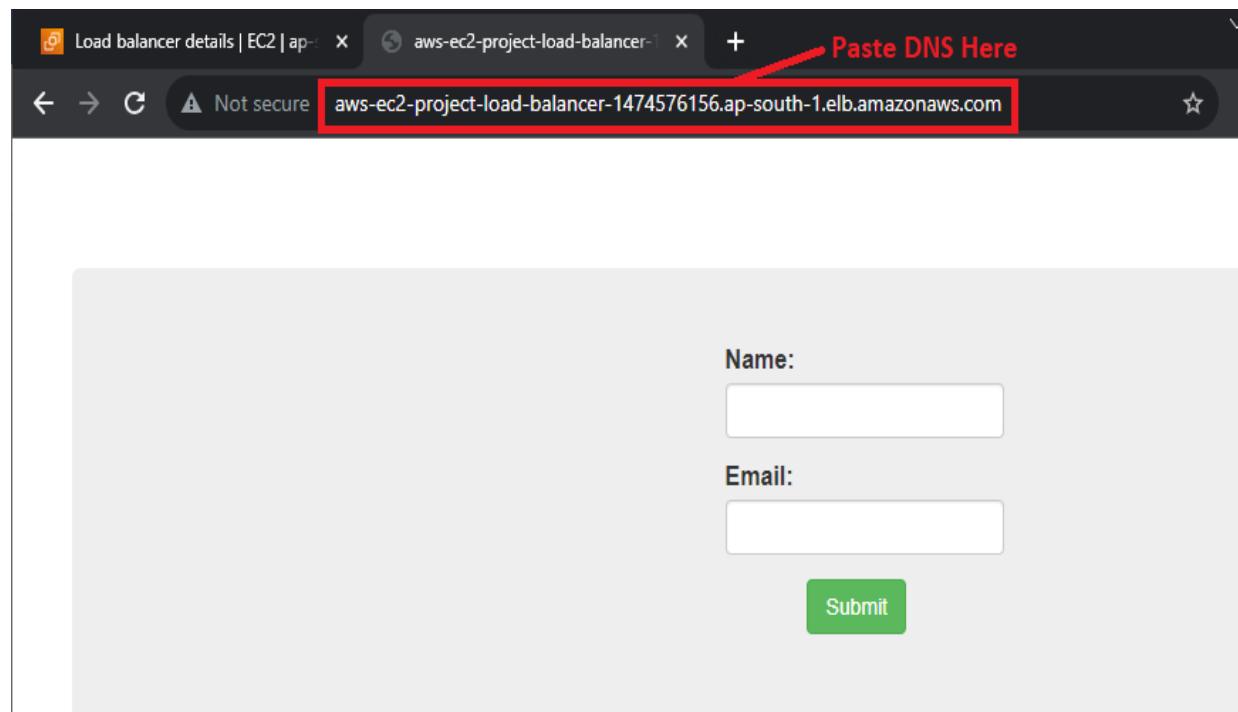
**Step 2: Click on the “Name (AWS-EC2-Project-Load-Balancer)”.**

The screenshot shows the AWS Lambda console displaying a list of load balancers. The title bar says "Load balancers (1)". The main area shows a table with columns: "Name", "DNS name", "State", and "VPC ID". A single row is present, showing "AWS-EC2-Project-Load-Balancer" in the Name column, "AWS-EC2-Project-Load-Bal..." in the DNS name column, "Active" in the State column, and "vpc-0..." in the VPC ID column. A red box highlights the "Name" column header, and a red arrow points from the "Name" header to the "AWS-EC2-Project-Load-Balancer" entry, with the text "Click Here" overlaid on the arrow.

**Step 3: Copy the “DNS Name (AWS-EC2-Project-Load-Balancer-1657334538.ap-south-1.elb.amazonaws.com)”.**

Load balancer type	Status
Application	<span>Active</span>
Scheme	Hosted zone
Internet-facing	ZP97RAFLXTNZK
VPC	IP address type
vpc-04c3e6ef2b9c7b60f	IPv4
Availability Zones	Date created
subnet-0012fe787fe8d6766  ap-south-1a (aps1-az1)	October 6, 2023, 16:50 (UTC+05:30)
subnet-00cf17266e1b37cef  ap-south-1b (aps1-az3)	
Load balancer ARN	<b>Copy DNS From Here</b>
arn:aws:elasticloadbalancing:ap-south-1:037036564217:loadbalancer/app/AWS-EC2-Project-Load-Balancer/6bd68a365672496e	DNS name AWS-EC2-Project-Load-Balancer-1474576156.ap-south-1.elb.amazonaws.com (A Record)

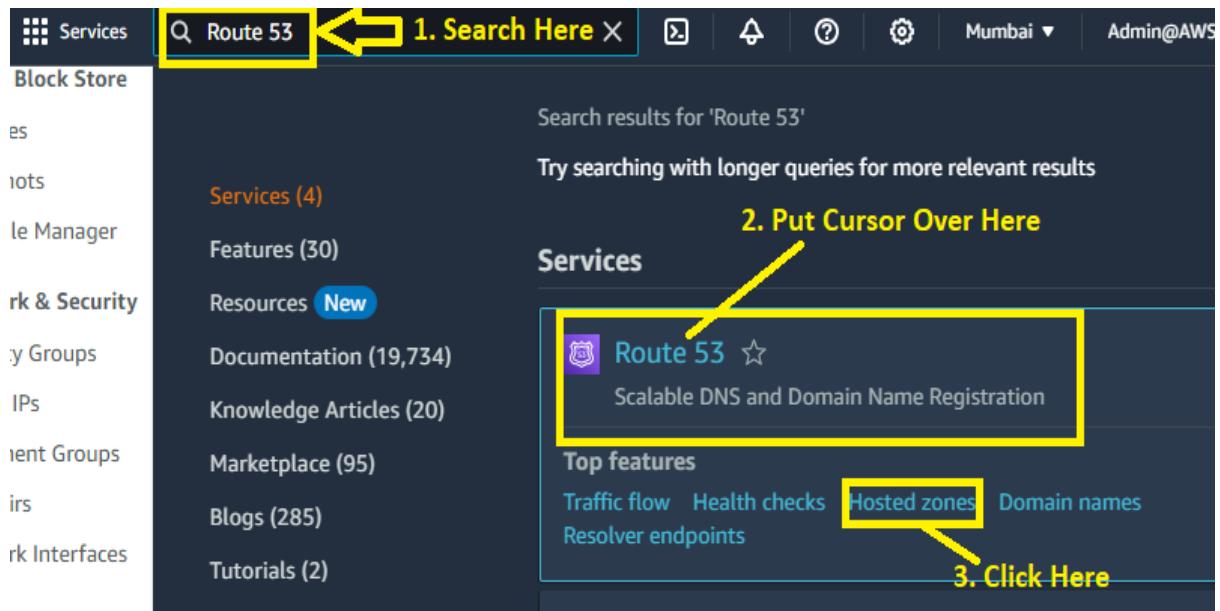
**Step 4: Paste the “DNS Name” into the “Browser Address Bar”. When you refresh your DNS multiple times, same page will have shown here.**



***This means the load balancer is working fine.***

## 11. Route Traffic from Load Balancer to A Specific Domain (Optional)

Step 1: Go to the “Services” section & search the “Route 53”. Click on the “Hosted Zones”.



Step 2: We have a domain (`visal.click`) with the hosted zones.

Click on the “`visal.click`” under Hosted zone name.

The screenshot shows the AWS Route 53 Hosted zones list. The URL in the browser is "Route 53 > Hosted zones". The page title is "Hosted zones (1)". A red arrow labeled "Click here" points to the "visal.click" entry in the list. The list table has columns: Hosted zone name, Type, Create..., and Rec. The "visal.click" entry is highlighted with a red box. The "Type" column shows "Public" and the "Create..." column shows "Route 53".

Hosted zone name	Type	Create...	Rec
visal.click	Public	Route 53	2

Step 3: Click on the “Create record”.

**Records (2) [Info](#)**

Automatic mode is the current search behavior optimized for best filter results. [To change modes go to settings.](#)

[Create record](#) Click Here

[Delete record](#) [Import zone file](#) [Create record](#)

[Filter records by property or v](#) [Type](#) [Routing policy](#) [Alias](#)

[<>](#) [1](#) [>](#) [⚙️](#)

<input type="checkbox"/>	<a href="#">Record ...</a>	<a href="#">Type</a>	<a href="#">Routin...</a>	<a href="#">Differ...</a>	<a href="#">Alias</a>
<input type="checkbox"/>	visal.click	NS	Simple	-	No
<input type="checkbox"/>	visal.click	SOA	Simple	-	No

**Step 4: Enable the “Alias” option. Choose the following options here:**

**Choose endpoint:** - Alias to Application and Classic Load Balancer

**Choose Region:** - Asia Pacific (Mumbai)

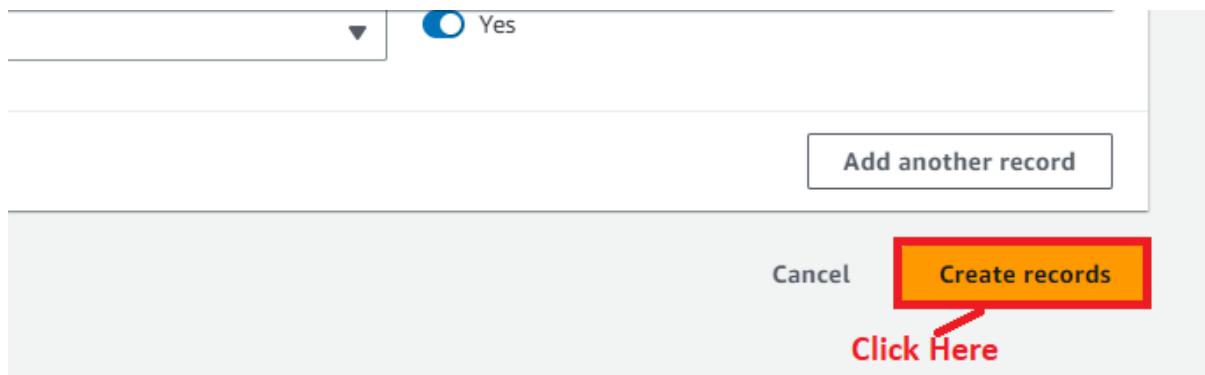
**Choose load balancer:** - dualstack.AWS-EC2-Project-Load-Balancer-1474576156.ap-south-1.elb.amazonaws.com

▼ Record 1 [Delete](#)

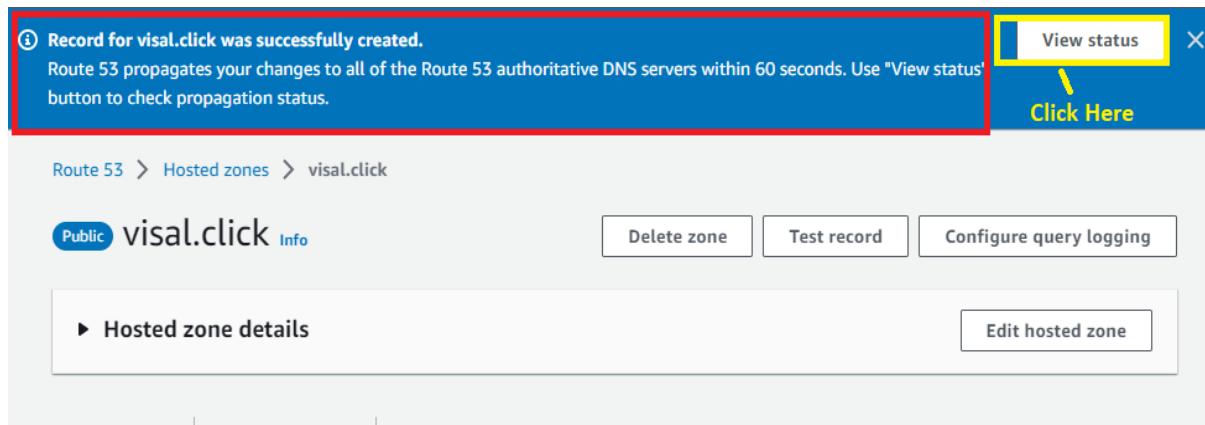
<b>Record name</b> <a href="#">Info</a>	<b>Record type</b> <a href="#">Info</a>
<input type="text"/> subdomain visal.click	A – Routes traffic to an IPv4 address and some AWS r... ▾
Keep blank to create a record for the root domain.	
<input checked="" type="radio"/> <b>Alias</b> <span style="color: red;">1. Enable this option</span>	<span style="color: red;">2. Choose these options</span>
<b>Route traffic to</b> <a href="#">Info</a> <div style="border: 2px solid red; padding: 5px;"> <input type="text"/> Alias to Application and Classic Load Balancer           <input type="text"/> Asia Pacific (Mumbai)           <input type="text" value="dualstack.AWS-EC2-Project-Load-Balancer-1474576156.ap-south-1.elb.amazonaws.com"/> <span style="float: right;"><a href="#">X</a></span> </div>	
Alias hosted zone ID: ZP97RAFLXTNZK	
<b>Routing policy</b> <a href="#">Info</a>	<b>3.</b> <input checked="" type="radio"/> Yes <b>Evaluate target health</b>
<input type="text"/> Simple routing	

## Routing policy – Simple routing

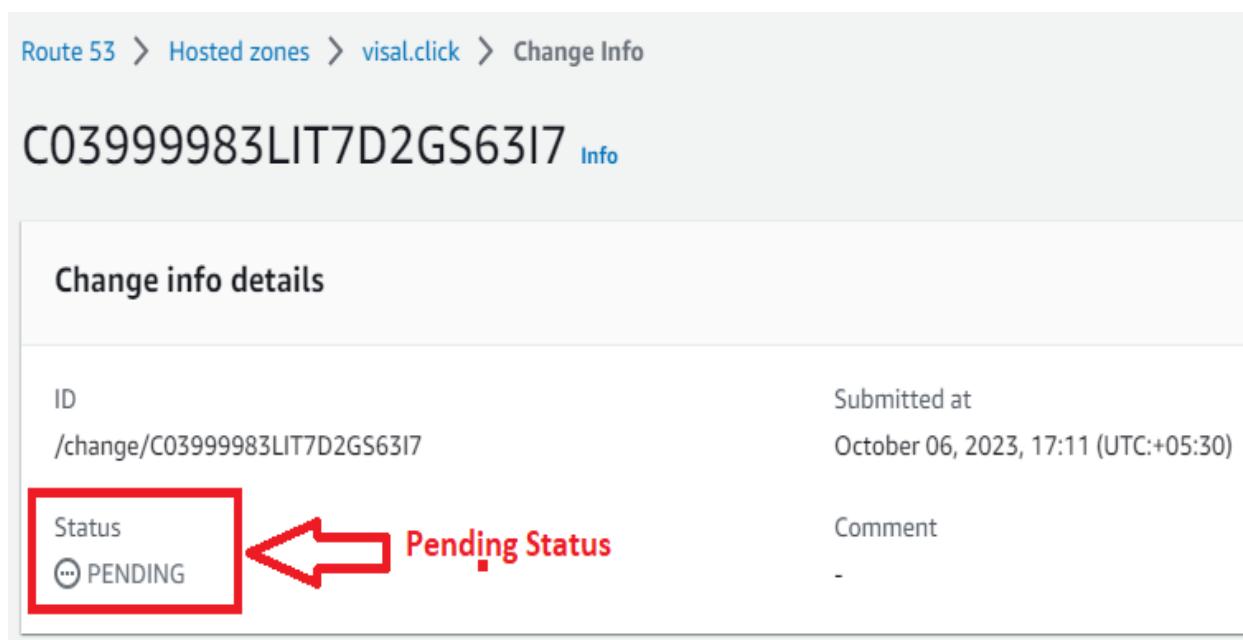
Click on the “Create records”.



**Step 5: The record will be successfully created. Click on the “View status”. It will take 60 seconds to be active.**



**Step 6: First, it will be in the “Pending State”.**



**Step 7: After the 60 seconds, it will show in the “INSYNC” mode. This means, your records will be successfully created & the load balancer traffic will be successfully routed to the domain name.**

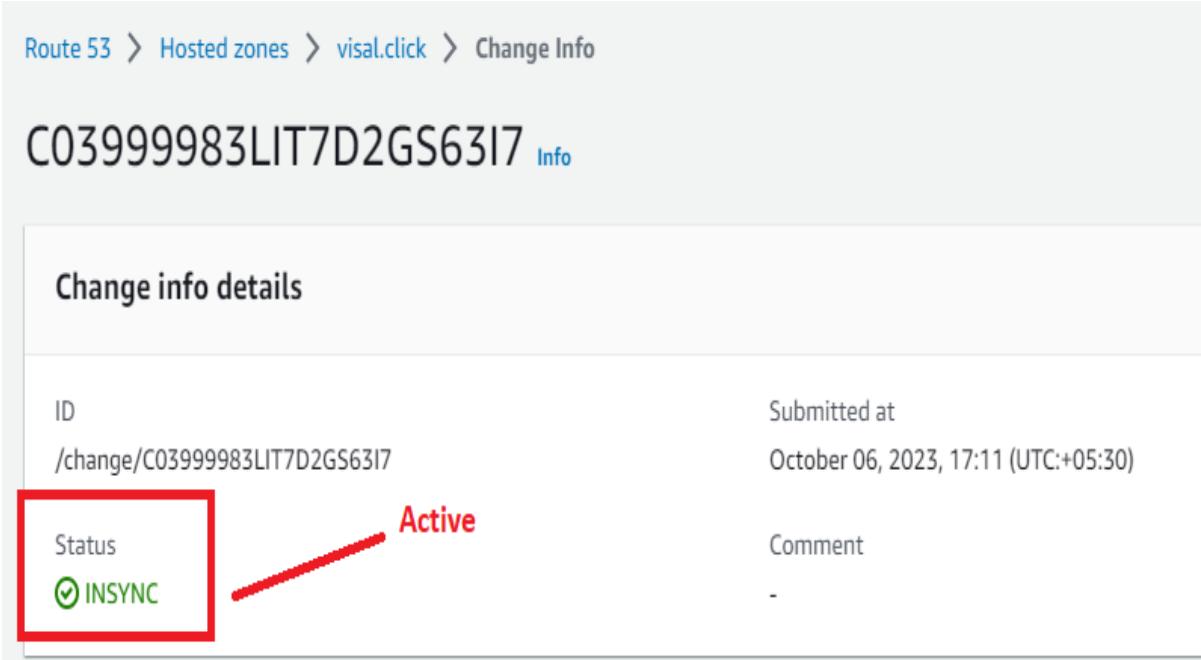
Route 53 > Hosted zones > visal.click > Change Info

C03999983LIT7D2GS63I7 [Info](#)

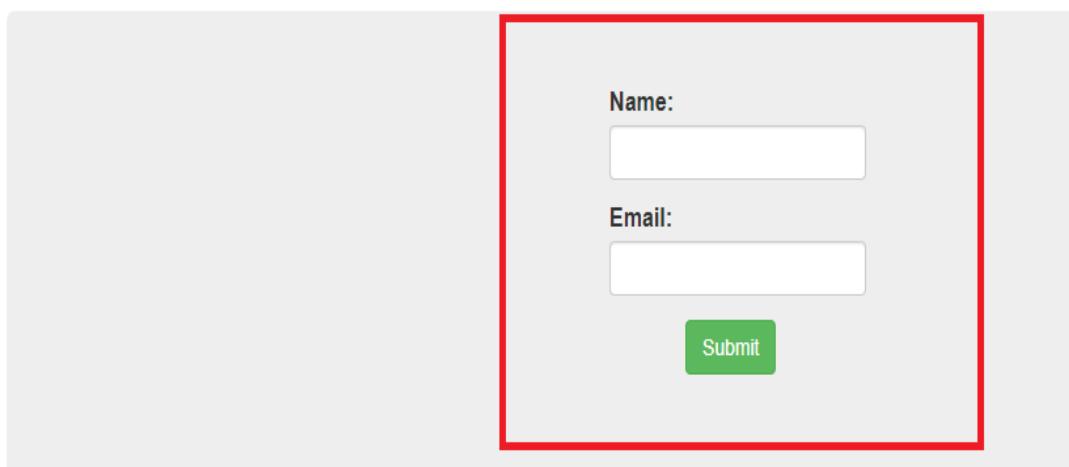
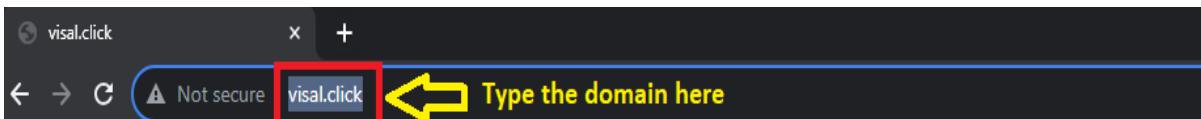
Change info details

ID	Submitted at
/change/C03999983LIT7D2GS63I7	October 06, 2023, 17:11 (UTC:+05:30)
Status	Comment
<input checked="" type="checkbox"/> INSYNC	-

**Active**



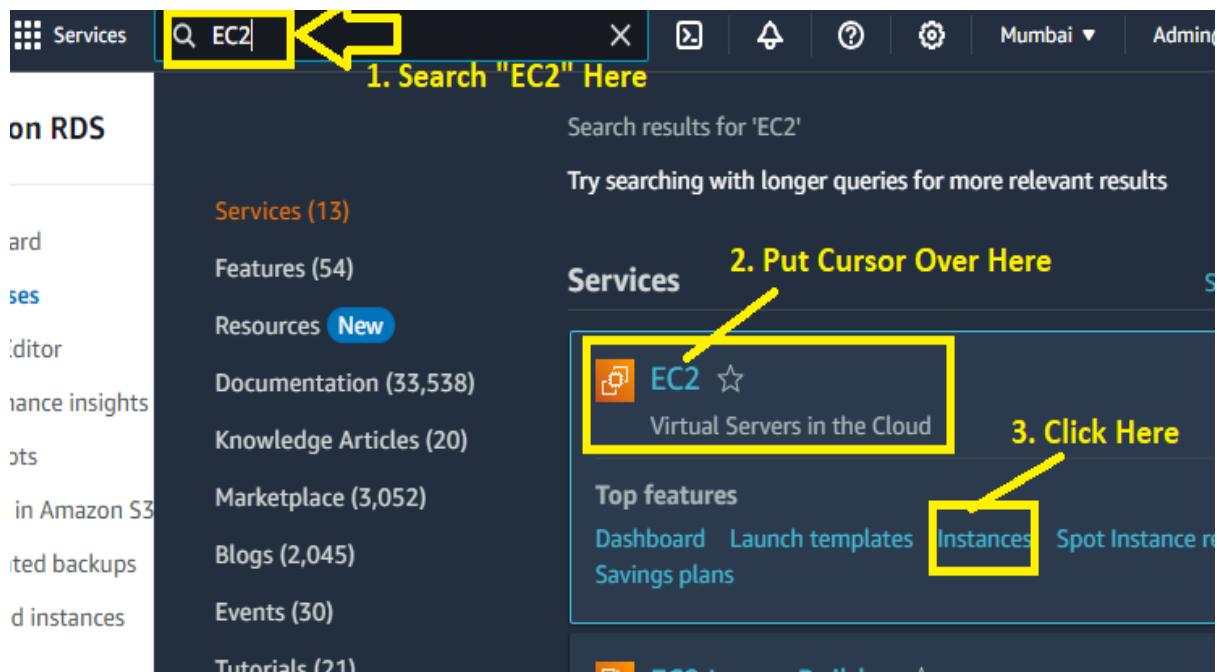
**Step 8: Type the domain in the separate browser address bar. You will notice that same PHP website will be visible on the registered domain (visal.click).**



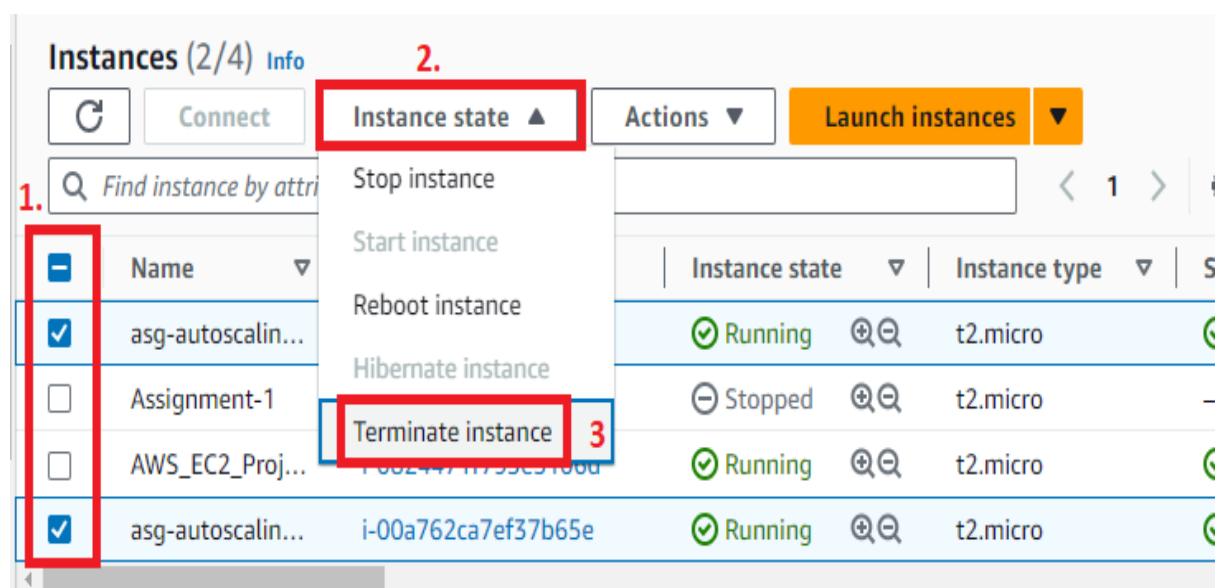
## 12. Test the Auto Scaling is Working Properly or Not

Now, we will test that auto scaling group is working fine or not. We will delete the auto scaling groups & test that website is working properly or not.

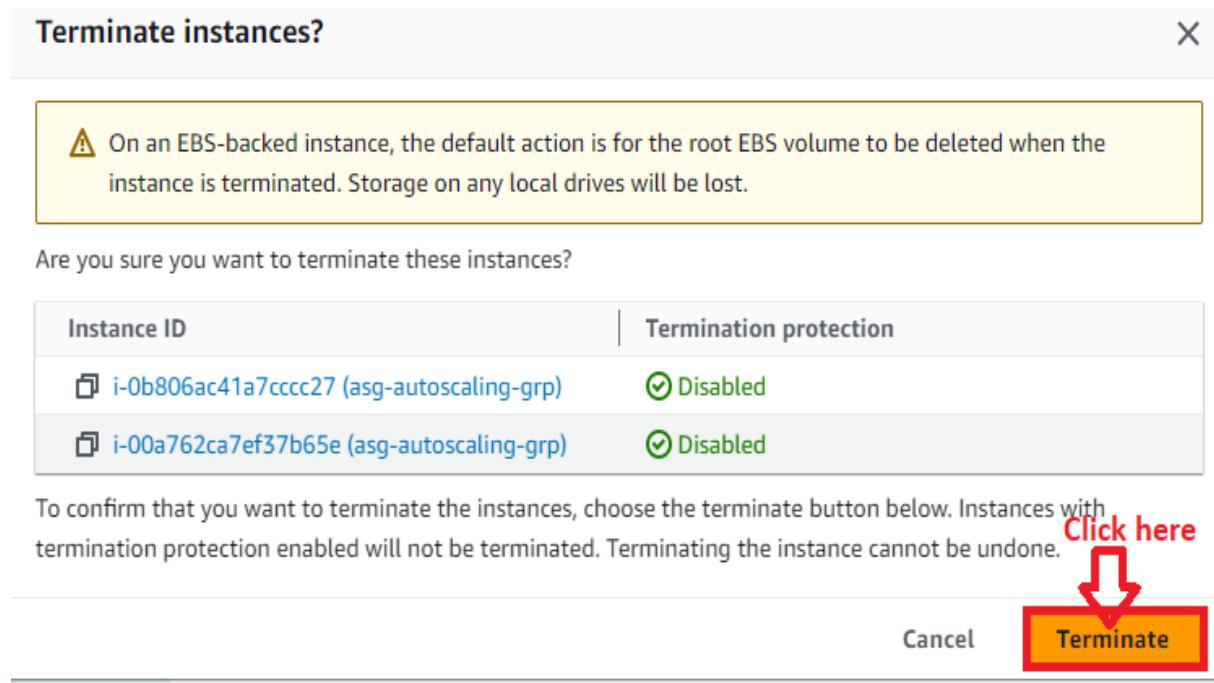
**Step 1:** Go to the “Services” section & search the “EC2”. Put cursor over the “EC2” & click on the “Instances”.



**Step 2:** We will select both the “asg-autoscaling-grp” Instances & click on the “Instance State”. Terminate the instance by clicking on the Terminate instance.



### Step 3: Click on the “Terminate”.



### Step 4: Your instance will be successfully terminated.

Instances (4) <a href="#">Info</a>						
<input type="checkbox"/>	Name	1. <a href="#">▼</a>	Instance ID	Instance state	Instance type	Status
<input type="checkbox"/>	asg-autoscalin...	<a href="#">i-0b806ac41a7cccc27</a>	<a href="#">Terminated</a>	t2.micro	-	
<input type="checkbox"/>	Assignment-1	<a href="#">i-0c06d1c9ddc101ad1</a>	<a href="#">Stopped</a>	t2.micro	-	
<input type="checkbox"/>	AWS_EC2_Proj... <a href="#">2</a>	<a href="#">i-0824471f793e5106d</a>	<a href="#">Running</a>	t2.micro	<a href="#">2/2</a>	
<input type="checkbox"/>	asg-autoscalin...	<a href="#">i-00a762ca7ef37b65e</a>	<a href="#">Terminated</a>	t2.micro	-	

Step 5: When we wait for some time & refresh the instances. You will notice that the two new instances (asg-autoscaling-grp) is successfully created & in the “Running” state.

Instances (6) <a href="#">Info</a>					
<input type="checkbox"/>	Name	1.	Instance ID	Instance state	Actions ▾
<input type="checkbox"/>	asg-autoscalin...	i-0aade793c1ff842e1	<span>✓ Running</span>	<span>Q Q</span>	t2.micro <span>Init</span>
<input type="checkbox"/>	Assignment-1	i-0c06d1c9ddc101ad1	<span>✗ Stopped</span>	<span>Q Q</span>	t2.micro
<input type="checkbox"/>	AWS_EC2_Proj...	i-0824471f793e5106d	<span>✓ Running</span>	<span>Q Q</span>	t2.micro <span>✓ 2/2</span>
<input type="checkbox"/>	asg-autoscalin...	i-00a762ca7ef37b65e	<span>✗ Terminated</span>	<span>Q Q</span>	t2.micro
<input type="checkbox"/>	asg-autoscalin...	i-05fb535b584884a89	<span>✓ Running</span>	<span>Q Q</span>	t2.micro <span>✓ 2/2</span>

**Step 6: Go to the left side & click on the “Auto Scaling Groups” in the “Auto Scaling”.**

Placement Groups  
Key Pairs  
Network Interfaces  
▼ Load Balancing  
Load Balancers  
Target Groups  
▼ Auto Scaling  
**Auto Scaling Groups**

Instances 6 Key  
Load balancers 1 Plan  
Security groups 15 Sna  
Volumes 4

**Launch instance**

**Step 7: Click on the “AWS\_EC2\_Project1\_ASG”.**

Create Auto Scaling group

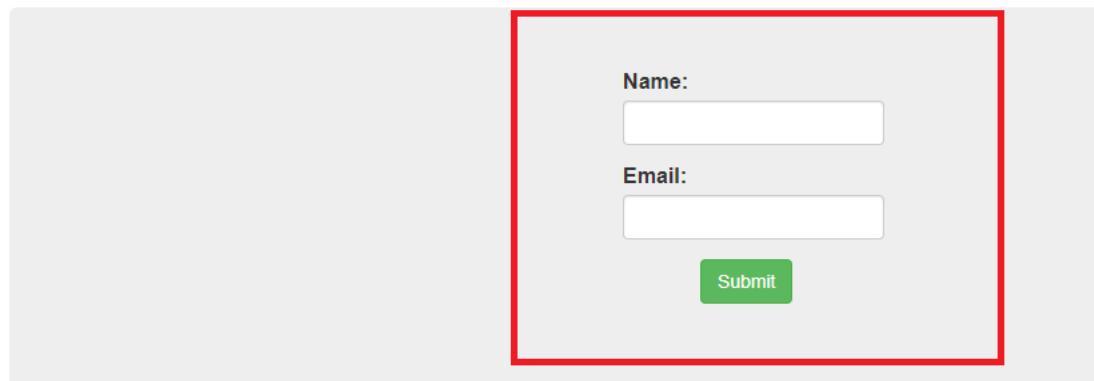
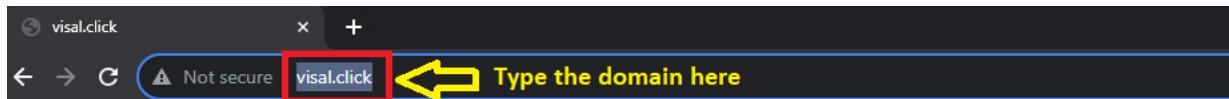
Search your Auto Scaling groups **Click Here**

<input type="checkbox"/>	Name	Launch template/configuration
<input type="checkbox"/>	<b>AWS_EC2_Project1_ASG</b>	AWS_EC2_Project_Template   Version De

**Step 8: Go to the “Activity” & you will notice that instance has been successfully terminated & again created.**

Status	Description	Cause
Successful	Launching a new EC2 instance: i-Oaade793c1ff842e1	At 2023-10-09T06:28:08Z an instance was launched in response to an unhealthy instance needing to be replaced.
Successful	Terminating EC2 instance: i-0b806ac41a7cccc27	At 2023-10-09T06:28:08Z an instance was taken out of service in response to an EC2 health check indicating it has been terminated or stopped.
Successful	Launching a new EC2 instance: i-05fb535b584884a89	At 2023-10-09T06:26:05Z an instance was launched in response to an unhealthy instance needing to be replaced.
Successful	Terminating EC2 instance: i-00a762ca7ef37b65e	At 2023-10-09T06:26:05Z an instance was taken out of service in response to an EC2 health check indicating it has been terminated or stopped.

**Step 9: When we refresh our domain (visal.click), it is showing that our PHP website is up & running. No downtime for us here.**



### **13. Test the Database is Working Properly or Not**

We have checked that our website is working fine with EC2 Server, Elastic Load Balancer & Auto Scaling Groups. No problem in page loading & routing to domain.

Now, we will test our database is successfully storing the data or not.

**Step 1: We will fill the following entries in the name:**

**Name – Google**

**Email – [support@google.com](mailto:support@google.com)**

**Click on the “Submit”.**

Name:  
Google

Email:  
support@google.com

Submit

1. Fill Name Here
2. Fill Email Here
3. Click on Submit.

**Step 2: A message “New record created successfully” has been shown after submitting the details.**

Name:

Email:

Submit

Success Message

New record created successfully

**Step 3: Now, we will go to our EC2 instance (AWS\_EC2\_Project1), where we have created the website. Now, we will connect the website to the database again & check that information in the database is showing or not.**

**Go to the “EC2” Machine & connect the machine with the database using the following command: sudo mysql -h application-**

**database.calupg2c7sxe.ap-south-1.rds.amazonaws.com -u admin -pintel123**

**The database will be successfully connected.**

```
No VM guests are running outdated hypervisor (aemu) binaries on this host.  
ubuntu@ip-172-31-42-115:/var/www/html$ sudo mysql -h application-database.calupg2c7sxe.ap-south-1.rds.amazonaws.com -u admin -pintel123  
mysql: [Warning] Using a password on the command line interface can be insecure.  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 38  
Server version: 8.0.33 Source distribution  
  
Copyright (c) 2000, 2023, 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> |
```

**Step 4: Now, go to the “intel” database. Write the command (“use intel”) here. Now, we are inside the “intel” database.**

```
mysql> use intel  
Database changed  
mysql> |
```

**i-0824471f793e5106d (AWS\_EC2\_Project1)**

Public IPs: 3.110.85.213 Private IPs: 172.31.42.115

**Step 5: When we run the “select \* from data;” command, it is showing our filled data.**

```
mysql> select * from data;
+-----+-----+
| firstname | email      |
+-----+-----+
| AWS       | support@aws.com |
| Azure     | support@azure.com |
| AWS       | support@aws.com |
| Google    | support@google.com |
+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql>
```

i-0824471f793e5106d (AWS\_EC2\_Project1)

PublicIPs: 35.154.6.135 PrivateIPs: 172.31.42.115

***This means, our website & database is successfully connecting and working fine with Auto Scaling Groups.***