

AWS Capstone Project 1: How to Deploy a Multi-Tier PHP Website Using the AWS EC2 & RDS Services

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.

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

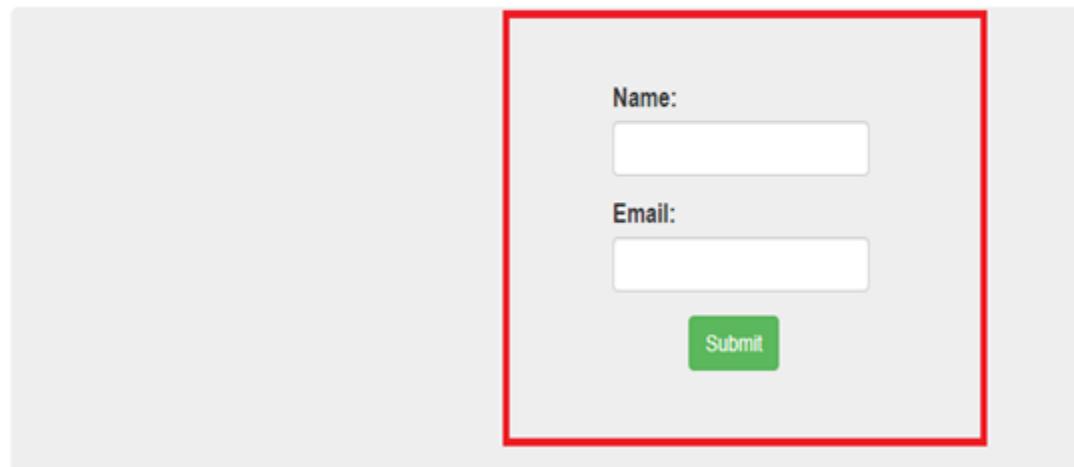
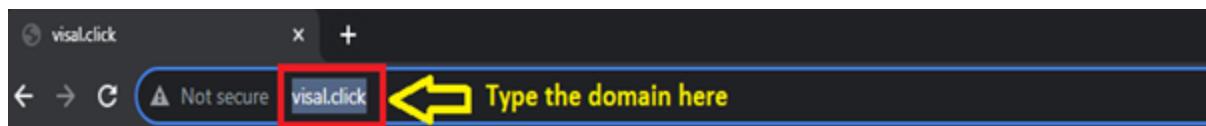
6. Allow traffic from EC2 to RDS instance

7. Allow all-traffic to EC2 instance

Check the Git Hub Repository for this Assignment to Copy the Commands & Code:

GitHub - visaltyagi/AWS-Capstone-Project-1: How to Deploy a Multi-Tier PHP Website Using the AWS...

How to Deploy a Multi-Tier PHP Website Using the AWS EC2 & MySQL RDS - visaltyagi/AWS-Capstone-Project-1
github.com



AWS Capstone Project 1: How to Deploy a Multi-Tier PHP Website Using the AWS EC2 & RDS Services

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”.

The screenshot shows the AWS Management Console Home page. At the top, there's a navigation bar with 'Services' (with a grid icon), a search bar containing 'Search' with a magnifying glass icon, and a keyboard shortcut '[Alt+S]'. Below the navigation bar, the title 'Console Home' is displayed with an 'Info' link. Underneath, a section titled 'Recently visited' has an 'Info' link. A red box highlights the 'EC2' service icon, which is a blue square with a white computer monitor and keyboard. A red arrow points from the text 'Click Here' to this highlighted icon. To the right of the 'EC2' icon, the list of recently visited services includes: Billing (green square with a document icon), CloudWatch (red square with a magnifying glass icon), Route 53 (blue square with a shield icon), RDS (blue square with a database icon), AWS Auto Scaling (pink square with a plus sign icon), AWS Cost Explorer (green square with a magnifying glass icon), Amazon EventBridge (purple square with a gear icon), Simple Notification Service (pink square with a mail icon), FSx (green square with a storage icon), Directory Service (red square with a person icon), AWS Compute Optimizer (red square with a gear icon), and VPC (purple square with a network icon).

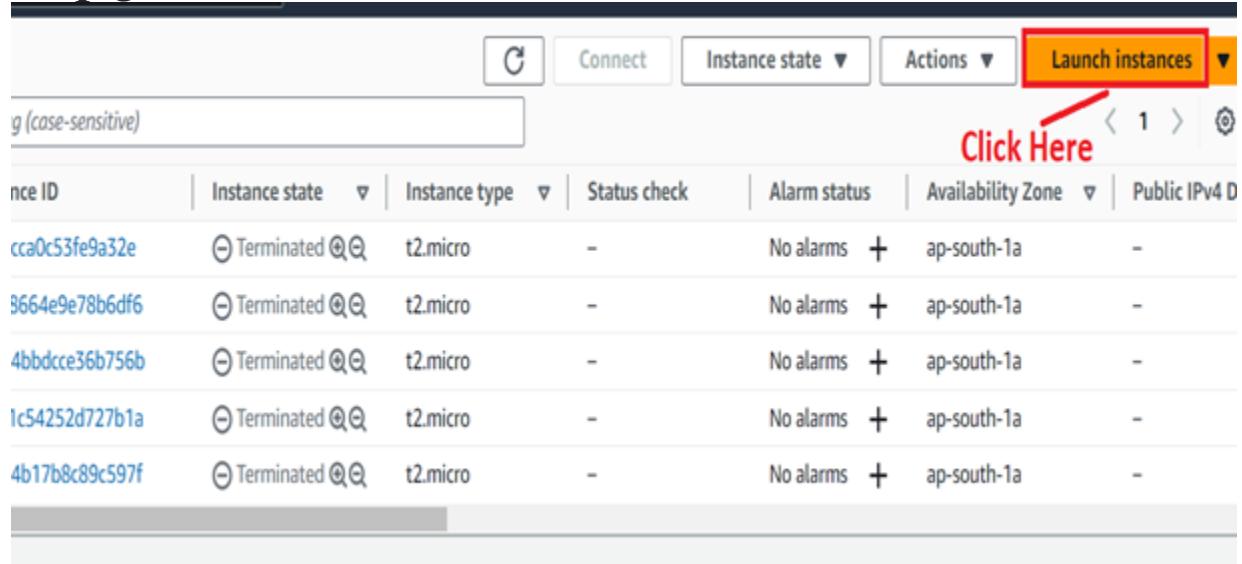
Go to the “EC2” Service

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

The screenshot shows the EC2 Dashboard. On the left, there's a sidebar with a 'New EC2 Experience' feedback prompt ('Tell us what you think') and a close button 'X'. Below it, the sidebar lists: 'EC2 Dashboard', 'EC2 Global View', 'Events', and a '▼ Instances' section with 'Instances', 'Instance Types', and 'Launch Templates'. The main area is titled 'Resources' with a 'Click here' label above it. It displays a message: 'You are using the following Amazon EC2 resources in the Asia Pacific'. Below this, a table shows resource counts: 'Instances (running)' with 0, 'Auto Scaling Groups' with 0, 'Elastic IPs' with 0, 'Instances' with 0, 'Load balancers' with 0, 'Placement groups' with 0, 'Snapshots' with 0, and 'Volumes' with 0. The 'Instances (running)' row is highlighted with a red box and a red arrow pointing from the 'Click here' label to its value.

Go to the “Instances (running)”

Step 3: Click on the “Launch instances”.

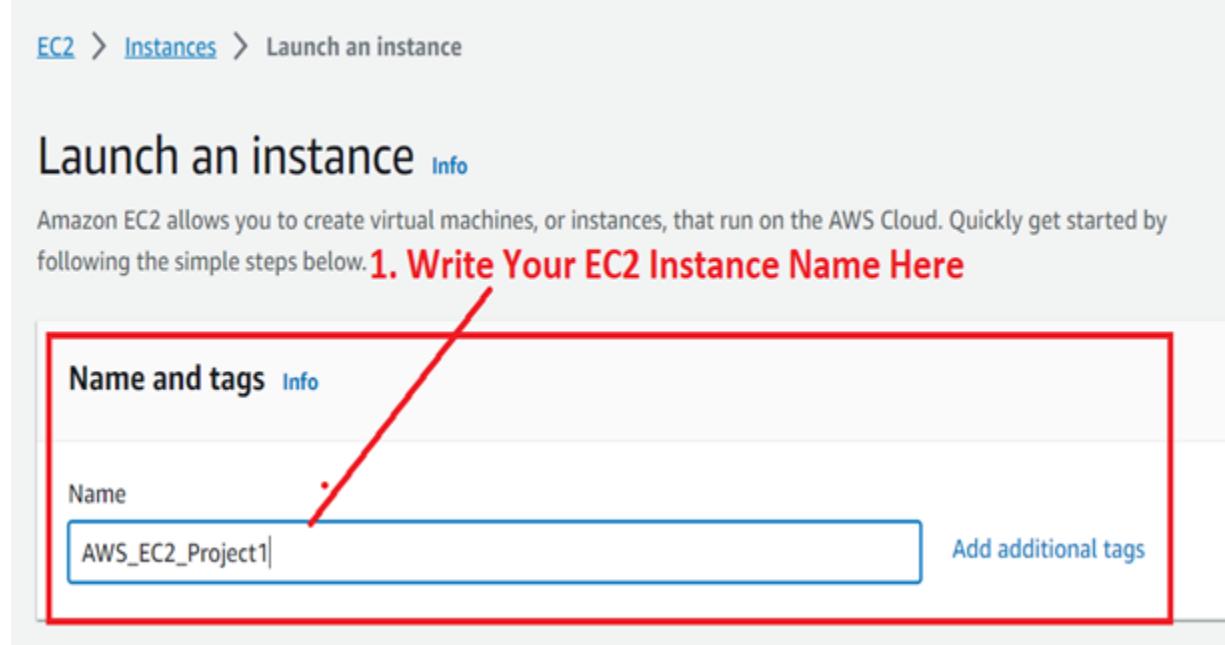


Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
cca0c53fe9a32e	Terminated	t2.micro	-	No alarms	+ ap-south-1a	-
9664e9e78b6df6	Terminated	t2.micro	-	No alarms	+ ap-south-1a	-
4bbdcce36b756b	Terminated	t2.micro	-	No alarms	+ ap-south-1a	-
1c54252d727b1a	Terminated	t2.micro	-	No alarms	+ ap-south-1a	-
4b17b8c89c597f	Terminated	t2.micro	-	No alarms	+ ap-south-1a	-

Launch a New Instance

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 : Add additional tags

Write the Instance Name

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

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: x86_64
AMI ID: ami-0f5ee92e2d63afc18
Verified provider

Select the “Ubuntu” AMI

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

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

Additional costs apply for AMIs with pre-installed software

All generations

Compare instance types

Choose the “t2.micro” instance type

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.

Choose key pair here

Key pair name - required

MY-WEB-SERVER

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.

Create new key pair

Select the “Key Pair” Name here

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 the Network Settings



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
 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](#)
 2. Write security group name & description over here



Write the Security Group Name and Description

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

Create the SSH Rule

Click on the “Add security group rule”.

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

⚠ 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

Add a new security group rule

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)

Type [Info](#) **Protocol** [Info](#) **Port range** [Info](#)

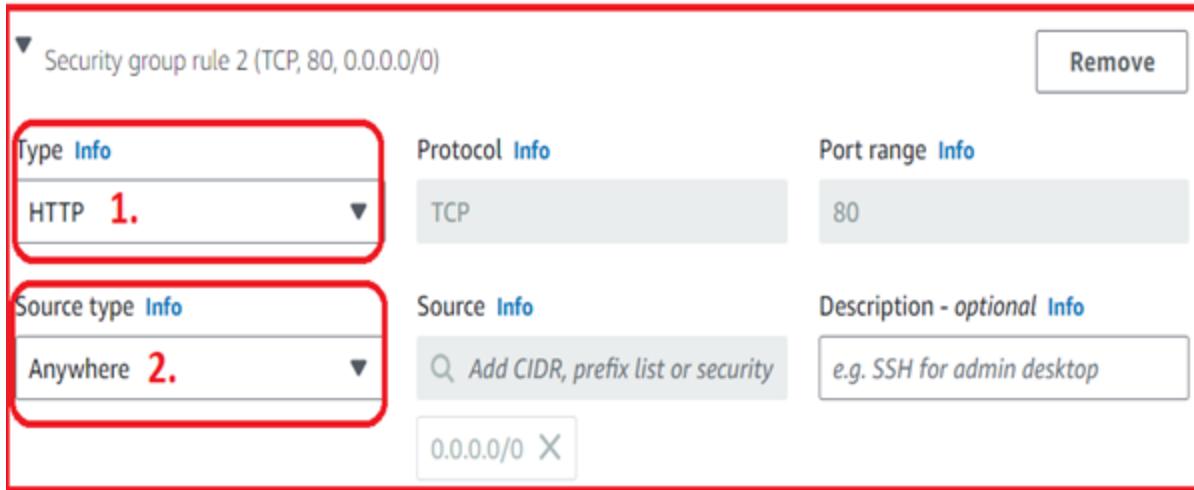
HTTP **1.** TCP 80

Source type [Info](#) **Source** [Info](#) **Description - optional** [Info](#)

Anywhere **2.** Add CIDR, prefix list or security e.g. SSH for admin desktop

0.0.0.0/0 X

Remove

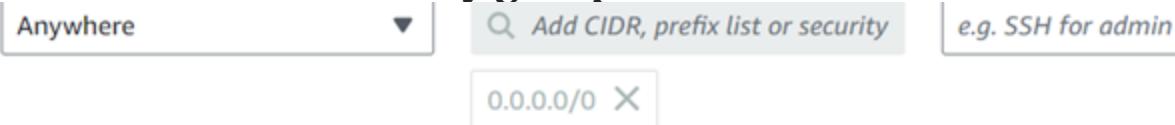


Choose the HTTP Security Group Rule

Click on the “Add security group rule”.

Anywhere ▾ Add CIDR, prefix list or security e.g. SSH for admin

0.0.0.0/0 X



⚠ 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

Add another security group rule

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

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

Type Info 1. MySQL/Aurora	Protocol Info TCP	Port range Info 3306
Source type Info 2. Anywhere	Source Info Add CIDR, prefix list or security 0.0.0.0/0 X	Description - optional Info e.g. SSH for admin desktop

Choose the RDS MySQL Rule

Step 11: Click on the “Launch Instance”.

Advanced

(Not encrypted)

General Purpose (SSD) or Magnetic storage X

Instance allows. Only the first 0 instance store

Edit

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...[read more](#)
ami-0f5ee92e2d63afc18

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

Click Here

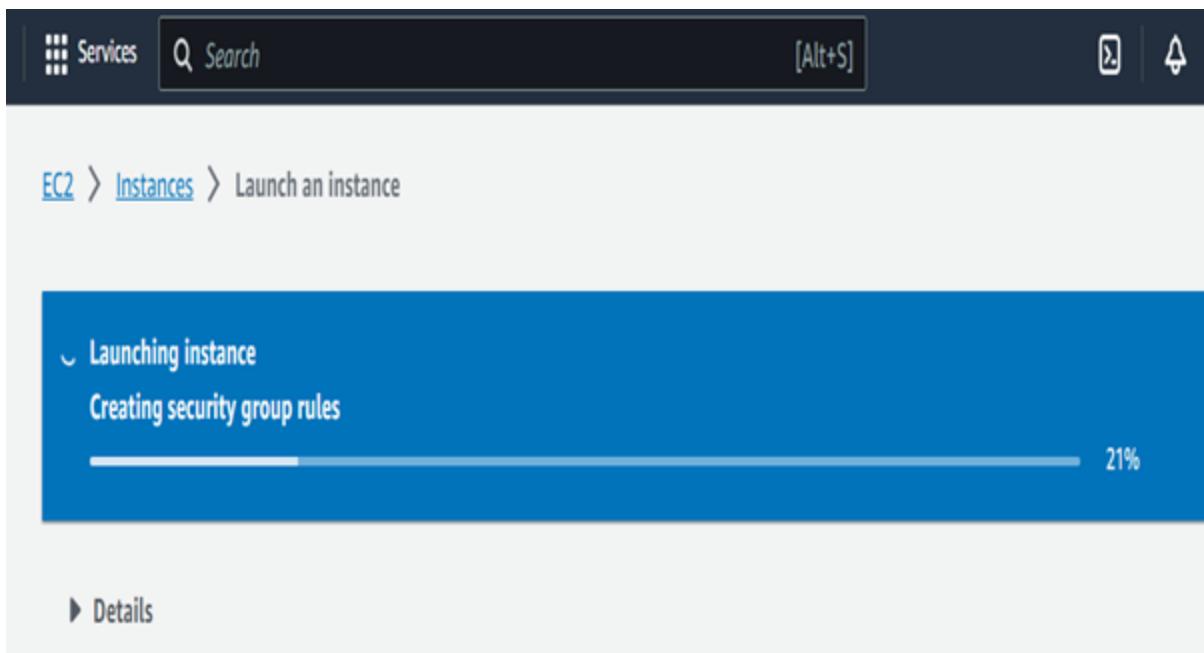
Free tier: In your first year X ·

Cancel Launch instance

Review commands

Launch the Instance

Step 12: The instance will be started launching.



Instance Started Launching

Step 13: The instance [AWS_EC2_Project1] will be launched successfully. Click on the “Hyperlink”.

A screenshot of the AWS EC2 Instances launch log. The path 'EC2 > Instances > Launch an instance' is at the top. Below it, a green success message box contains the text: 'Success' with a checkmark icon, followed by 'Successfully initiated launch of instance (i-0824471f793e5106d)'. A red box highlights the instance ID 'i-0824471f793e5106d', and a red arrow points to it from the text 'Click Here' above. Below the message box, there's a 'Launch log' link. At the bottom, a 'Next Steps' section contains the text 'Click on the Hyperlink'.

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

Instances (1) Info		C	Connect	Instance state ▾	Actions ▾
<input type="text"/> Find instance by attribute or tag (case-sensitive)					
Instance ID = i-0824471f793e5106d X	Clear filters	Pending state			
<input type="checkbox"/> Name ▾ Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm sta	
<input type="checkbox"/> AWS_EC2_Proj... i-0824471f793e5106d	(Pending)	t2.micro	-	No alarms	

Instance in the Pending State

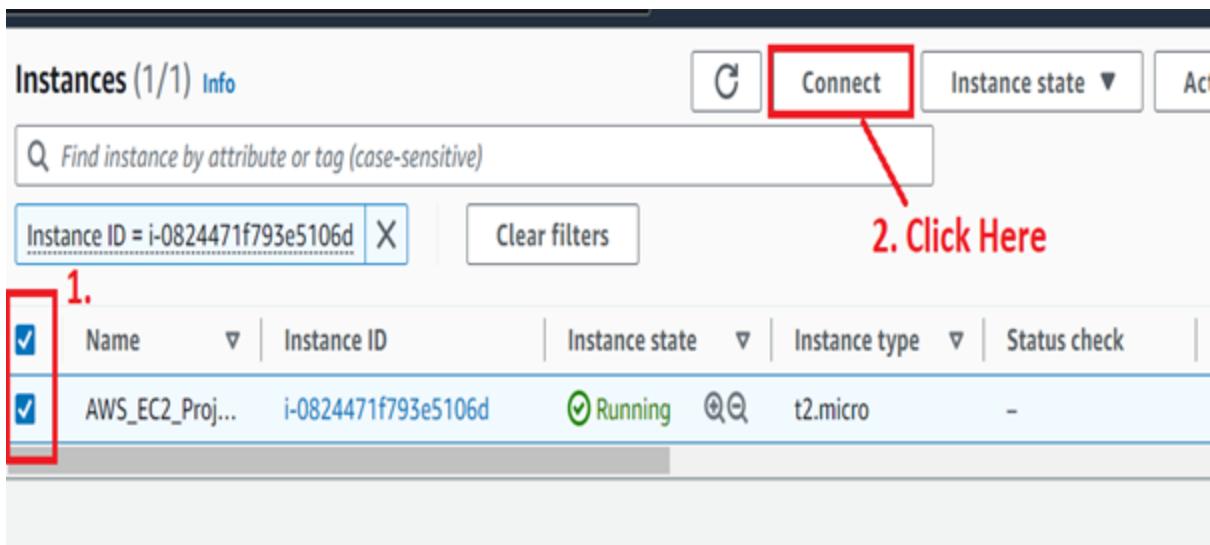
Step 15: It will be in the “Running State” after sometime.

Instances (1) Info		C	Connect	Instance state ▾	
<input type="text"/> Find instance by attribute or tag (case-sensitive)					
Instance ID = i-0824471f793e5106d X	Clear filters	Running State			
<input type="checkbox"/> Name ▾ Instance ID	Instance state ▾	Instance type ▾	Status check		
<input type="checkbox"/> AWS_EC2_Proj... i-0824471f793e5106d	(Running)	t2.micro	-		

Running Instance

Step 16: Select the instance

[AWS_EC2_Project1] & click on the “Connect”.



Select the Instance for Connection

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.

1. EC2 Instance Connect | Session Manager | SSH client | EC2 serial console

Instance ID
 i-0824471f793e5106d (AWS_EC2_Project1)

Connection Type

Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address
 13.127.16.2

User name
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ubuntu.

2. Click Here

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.

Cancel **Connect**



Connect to the AWS_EC2_Project1 Instance

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)
```

```
PublicIPs: 13.127.16.2 PrivateIPs: 172.31.42.115
```

EC2 Instance Connected Successfully

Step 19: First, we will update the “AWS_EC2_Project1” machine using the below-given 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)

PublicIPs: 13.127.16.2 PrivateIPs: 172.31.42.115

Update the “AWS_EC2_Project1” Machine

The “Ubuntu machine [AWS_EC2_Project1]” will be **successfully updated**.

Step 20: Now, we will install the “apache2” web server over here to run the “PHP” website. Use the below-given command to install the Apache web server:

```
sudo apt install apache2 -y
```

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**

Apache2 Server Installation Start

```
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

Apache2 Server is Installing

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

Apache2 Server Successfully Installed

To check that the “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.



Apache2 Default Page

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 below-given 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.
```

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

```
buntu@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

Go to the “html” directory

Step 2: Run the below-given

**command & you will notice that “index.html” file will
be present over here.**

```
ls
```

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

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

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

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

```
index.html
```

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



index.html file

i-0824471f793e5106d (AWS_EC2_Project1)

PublicID: 13.127.16.2 PrivateID: 172.31.42.115

The Index.html file shown here

Step 3: Remove the index.html file using the below-given command:

```
sudo rm index.html
```

Run the below-given command to check that index.html file will be present or not:

```
ls
```

You will notice that there is no “index.html” file present over here. Press “enter” from the keyboard after typing the above-given 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

Remove the “index.html” file

Step 4: Create an “index.php” file using the below-given command:

```
sudo nano index.php
```

Press “enter” from the 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

Create the index.php file

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

```
GNU nano 6.2

[ N ] i-0824471f793e5106d (AWS_EC2_Project1)
Public IPs: 13.127.16.2 Private IPs: 172.31.42.115
```

The file editor opened

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

The index.php file

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

Paste the 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”.

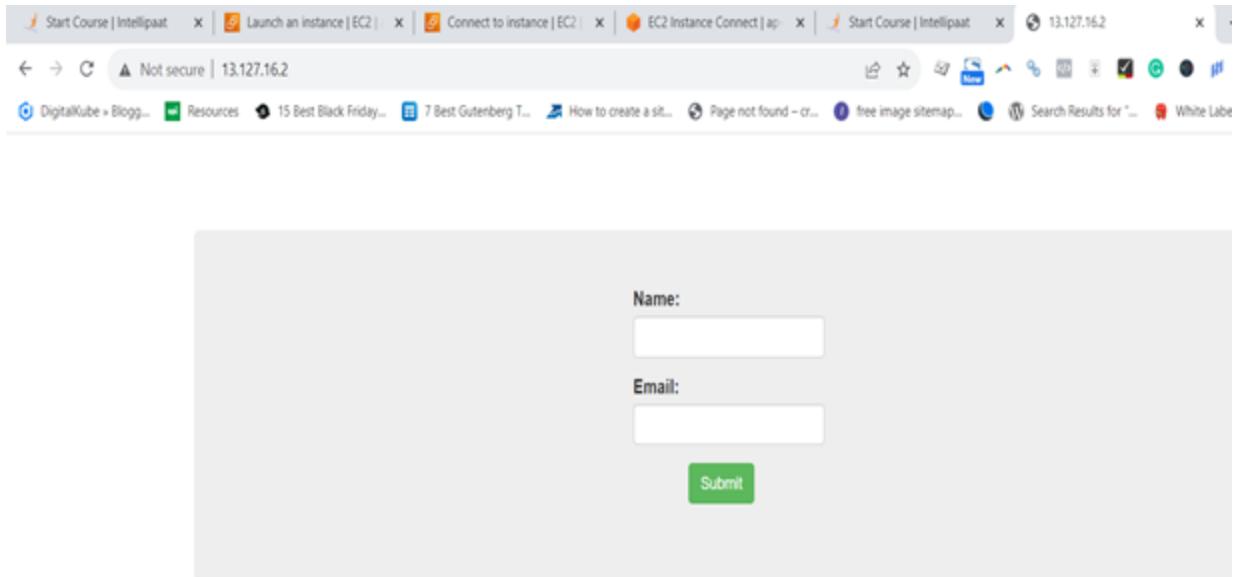
```
Running kernel seems to be up-to-date.  
No services need to be restarted.  
No containers need to be restarted.  
No user sessions are running outdated binaries.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ubuntu@ip-172-31-42-115:~$ cd /var/www/html  
ubuntu@ip-172-31-42-115:/var/www/html$ 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

Save the index.php file

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.



The screenshot shows a web browser window with multiple tabs open at the top. The active tab is titled "Start Course | Intellipaat" and has the URL "13.127.16.2". The page content is a simple form with two input fields: "Name:" and "Email:", each with a corresponding text input box below it. A green "Submit" button is located at the bottom right of the form area.

```
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(); } ?>
```

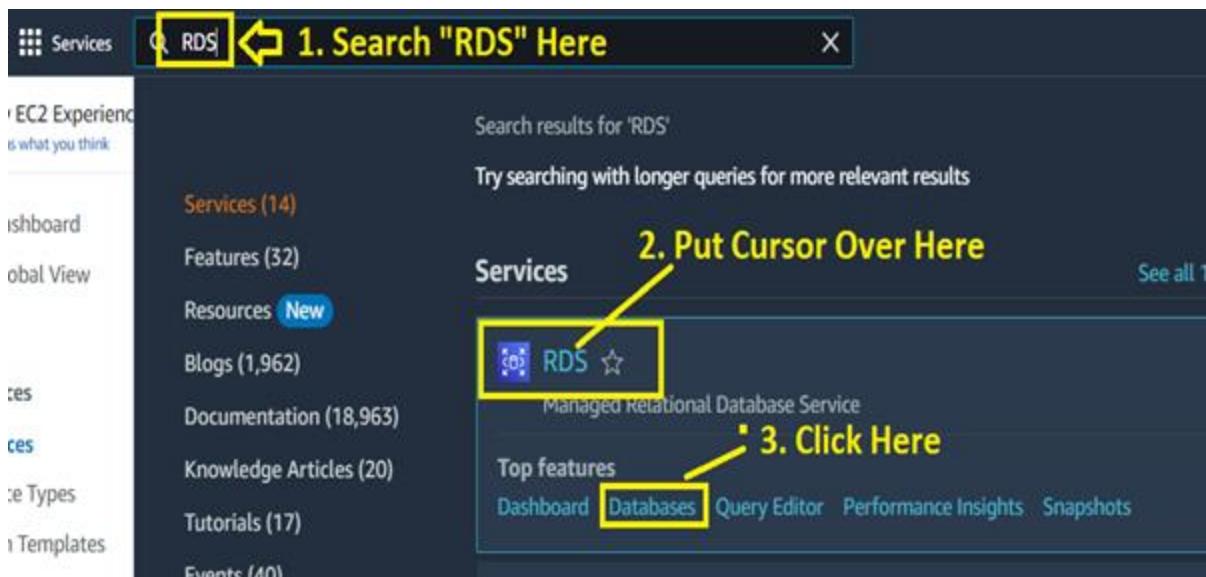
PHP Website Implemented

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

Follow these steps here:

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”.**



Go to the “Databases” section

Step 2: Click on the “Create database”.

The screenshot shows the 'Databases' section of the RDS management console. A red box highlights the 'Create database' button in the top right corner of the main content area. A red arrow points downwards from this button towards the bottom right corner of the page.

Create a new database

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

Create database

Choose a database creation method Info

1. Choose this option

Standard create

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

Easy create

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

Engine options

2. Choose this Engine Option



Engine type Info

Aurora (MySQL Compatible)



Aurora (PostgreSQL Compatible)



MySQL



Choose the Database Creation Method & Engine Type

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'

Leave the By Default SQL Version

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

Templates

Choose a sample template to meet your use case.

Choose this option

Production

Use defaults for high availability and fast, consistent performance.

Dev/Test

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

Free tier

Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
[Info](#)

Choose the “Free tier” template

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.

Write the “DB Instance Identifier” Name

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.

 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](#)

.....

Write the Master Username and Password

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

Choose the “DB instance class”

**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

Storage Type & Unselect the “Storage autoscaling”

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.

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.

Connect to the EC2 Instance

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](#)

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.

2. Remain as it is

Choose the “Automatic Setup” Here

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

1.

Create new

Create new VPC security group

Additional VPC security group

Choose one or more options

AWS_EC2_Project1 X

2.

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

Choose the VPC Security Group

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"

Leave the Password Authentication By Default

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](#)

default.mysql8.0

Option group [Info](#)

default:mysql-8-0

Backup

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

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](#)

2. Disable by Deselecting these options

Put the Initial Database Name

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

Create the MySQL RDS Database

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

RDS > Databases



Consider creating a Blue/Green Deployment to minimize downtime during upgrades

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. [RDS User Guide](#) [Aurora User Guide](#)

Databases (1)

Group resources



Modify

Actions ▾

Restore from S3

Create database

Filter by databases

< 1 > ⌂

DB identifier

Status

Role

Engine

Region & AZ

Size

Actions

CPU

Current a



[application-database](#)

Creating

Instance

MySQL Community

ap-south-1a

db.t3.micro

-

-

Database Creating

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

Databases (1)		<input checked="" type="checkbox"/> Group resources		Modify	Actions ▾	
<input type="text"/> Filter by databases						
DB identifier	Status	Role	Engine	Region & AZ	Size	
application-database	Available	Instance	MySQL Community	ap-south-1a	db.t3.micro	

Database Available

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

Successfully set up a connection between application-database and EC2 Instance i-0824471f793e5106d					
Successfully created database application-database					
You can use settings from application-database to simplify configuration of suggested database add-ons while we finish creating your DB for you.					
RDS > Databases > application-database					
application-database					
Summary					
DB identifier application-database	CPU 22.38%	Status Available	Class db.t3.micro		
Role Instance	Current activity 0 Connections	Engine MySQL Community	Region & AZ ap-south-1a		
Connectivity & security	Monitoring	Logs & events	Configuration	Maintenance & backups	Tags
application-database					

Connectivity & security	Monitoring	Logs & events	Configuration	Maintenance & backups	Tags
Connectivity & security Copy this Endpoint					
Endpoint & port Endpoint application-database.calupg2c7sxe.ap-south-1.rds.amazonaws.com Port 3306	Networking Availability Zone ap-south-1a VPC vpc-04c3e6ef2b9c7b60f Subnet group rds-ec2-db-subnet-group-1 Subnets subnet-0cc15942d40e40f23 subnet-06f63da6a038b0fd3 subnet-0c6de8b75819dbb74 Network type IPv4	Security VPC security groups AWS_EC2_Project1 (sg-04d792595bc10f95d) Active rds-ec2-2 (sg-065ff9e0f5f70e541) Active Publicly accessible No Certificate authority Info rds-ca-2019 Certificate authority date August 22, 2024, 22:38 (UTC+05:30)			

Database Details & Copy the Endpoint from here

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

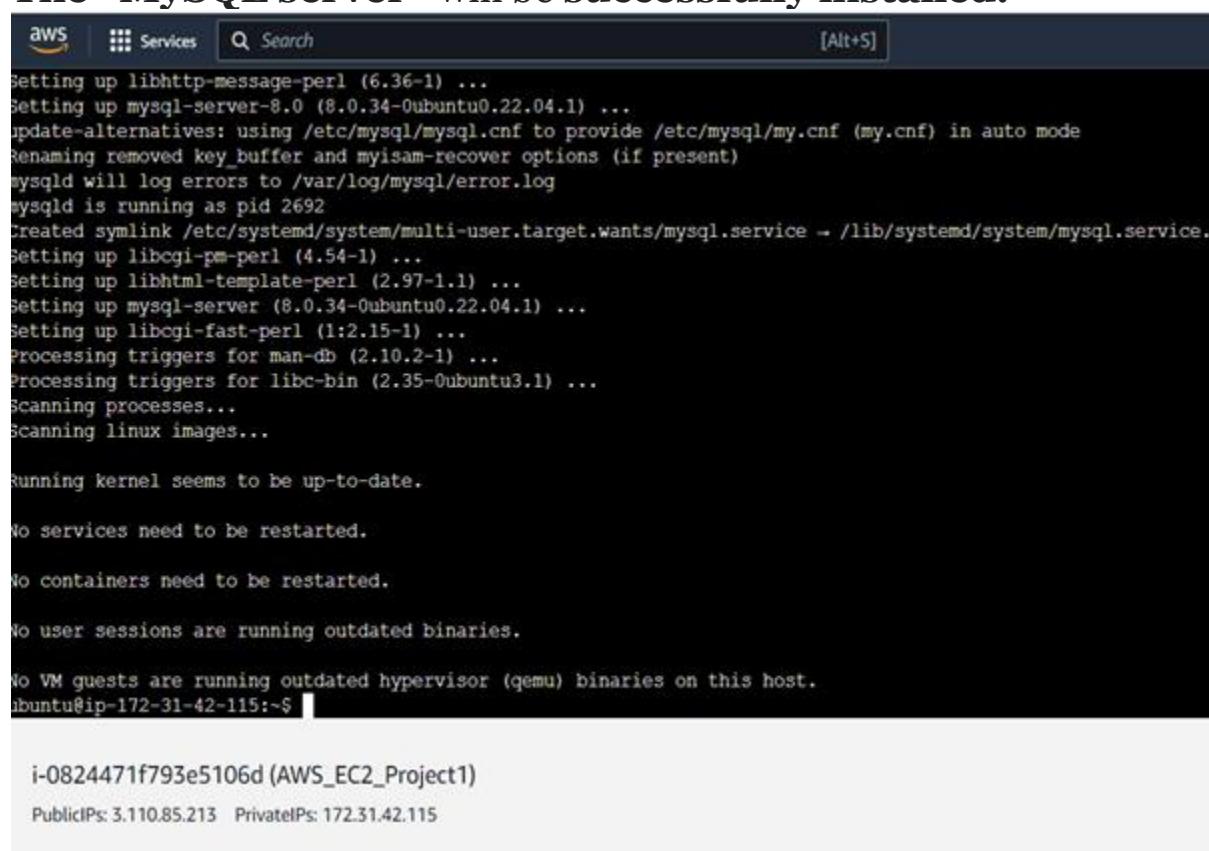
Step 1: First, we have to install the “MySQL” over “EC2” server using the below-given command:

```
sudo apt install mysql-server -y
```

```
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 libtimage-perl liburi-perl mecab-ipadic mecab-ipadic-utf8
er-8.0 mysql-server-core-8.0

libsharedcache-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 libtimage-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.
[...]
Install the MySQL Server
```

The “MySQL server” will be successfully installed.



A screenshot of a terminal window titled "Services" in the AWS CloudWatch interface. The window shows the output of a package installation command. The output includes several lines of text indicating the installation of MySQL Server and its dependencies, such as libhttp-message-perl, mysql-server-8.0, and libxml-perl. It also shows the creation of symlinks and the configuration of MySQL services. The terminal concludes with a message stating there are no VM guests running outdated hypervisor binaries on the host. At the bottom, the AWS CloudWatch footer displays the instance ID (i-0824471f793e5106d) and the project name (AWS_EC2_Project1), along with public and private IP addresses.

```
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.

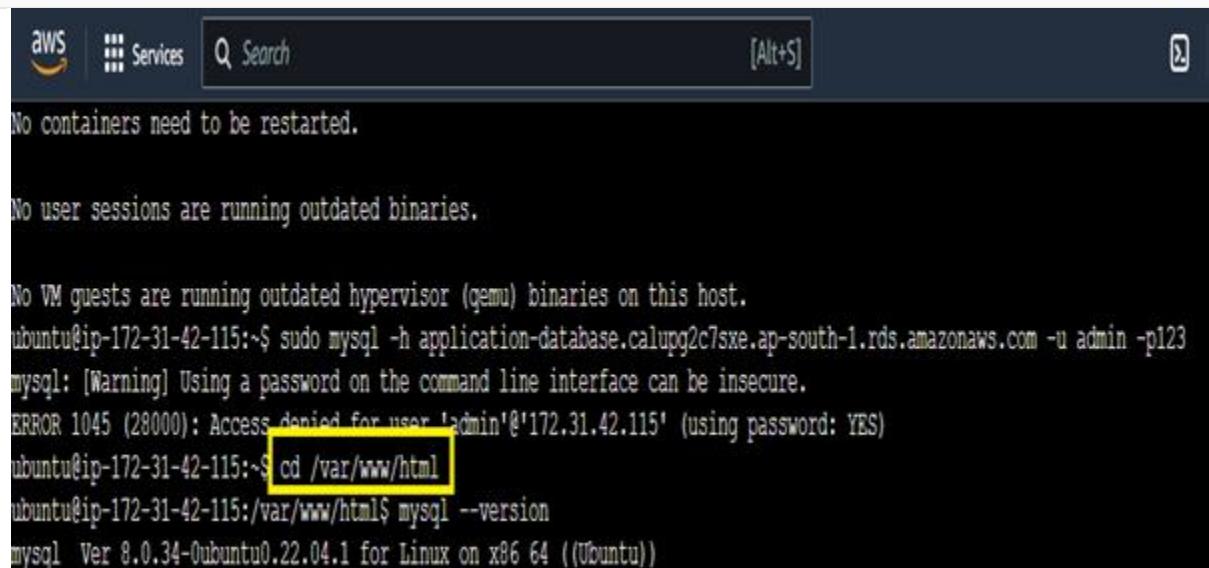
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-42-115:~$
```

i-0824471f793e5106d (AWS_EC2_Project1)
Public IPs: 3.110.85.213 Private IPs: 172.31.42.115

MySQL Server Installed Successfully

Step 2: First, you have to change the directory. Go to the html directory using the below-given command:

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



The screenshot shows the AWS CloudWatch Metrics interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, a search bar, and a help icon. Below the bar, a message says 'No containers need to be restarted.' and 'No user sessions are running outdated binaries.' Further down, it states 'No VM guests are running outdated hypervisor (qemu) binaries on this host.' A terminal window is displayed with the following MySQL command history:

```
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))
```

Go to the “html” directory

Step 3: Now, we will set up the MySQL database connection with the EC2 server using the below-given 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

Connected to the RDS MySQL Server

Step 4: Now, type the below-given 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 |  
| mysql |  
| performance_schema |  
| sys |  
+-----+  
5 rows in set (0.00 sec)
```

2. Database shown

```
mysql> |
```

i-0824471f793e5106d (AWS_EC2_Project1)

Public IPs: 3.110.85.213 Private IPs: 172.31.42.115

“intel” database

Step 5: Now, go to the “intel” database using the below-given command:

```
use intel;
```

Now, we will be 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

Database Changed

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 below-given 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 'data' at line 1
mysql> show tables;
+-----+
| Tables_in_intel |
+-----+
| data           |
+-----+
1 row in set (0.00 sec)

mysql>
```

The “data” table created successfully

Step 7: Now, we will insert the value into the data table using the below-given 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)

Public IPs: 3.110.85.213 Private IPs: 172.31.42.115

The “Insert into” Query Successfully Executed

Step 8: If you want to check that data is successfully inserted or not, type the below-given 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

Inserted Data Shown

Step 9: Now, we are exiting from the database using the below-given command:

```
exit
```

```
mysql> exit
Bye
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

Exit from the Database from the EC2 Server

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

Follow these steps:

1. Configure the Right Database Settings in the PHP File

Step 1: Open the “index.php” file using the below-given 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

Open the index.php file

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">
<tr>
<td colspan="4" style="text-align: center;">
Change servername & username here

<td colspan="4" style="text-align: center;">































































































































































































































































































































































































>
```

Change the Server Name & Username

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/www/html
-bash: cd: /var/www/html: 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$ █
```

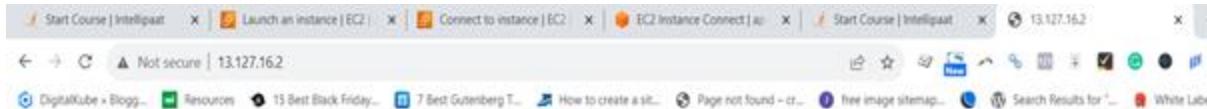
i-0824471f793e5106d (AWS_EC2_Project1)

PublicIPs: 3.110.85.213 PrivateIPs: 172.31.42.115

Save and exit from the index.php file

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.

A screenshot of a web page titled "Database Connection Error". The page contains a form with fields for "Name" and "Email", and a green "Submit" button. Below the form, there is a block of PHP code related to database connection errors.

```
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(); } ?>
```

Database Connection Error

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.

```
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 PHE
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
```

The PHP Repository Will be Added

Step 2: Now, install the mysql-client over the “EC2 Machine [AWS_EC2_Project1]” using the below-given 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

Install the MySQL Client

```
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) ...
```

i-0824471f793e5106d (AWS_EC2_Project1)

PublicIPs: 3.110.85.213 PrivateIPs: 172.31.42.115

Install the PHP & MySQL Dependencies

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.

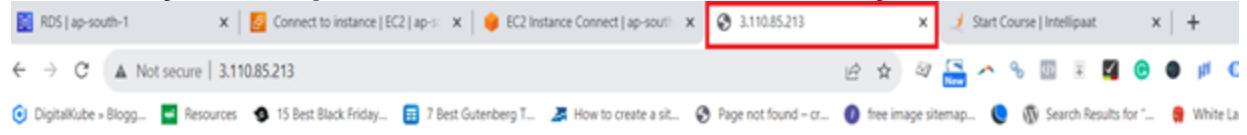
No user sessions are running outdated binaries.

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

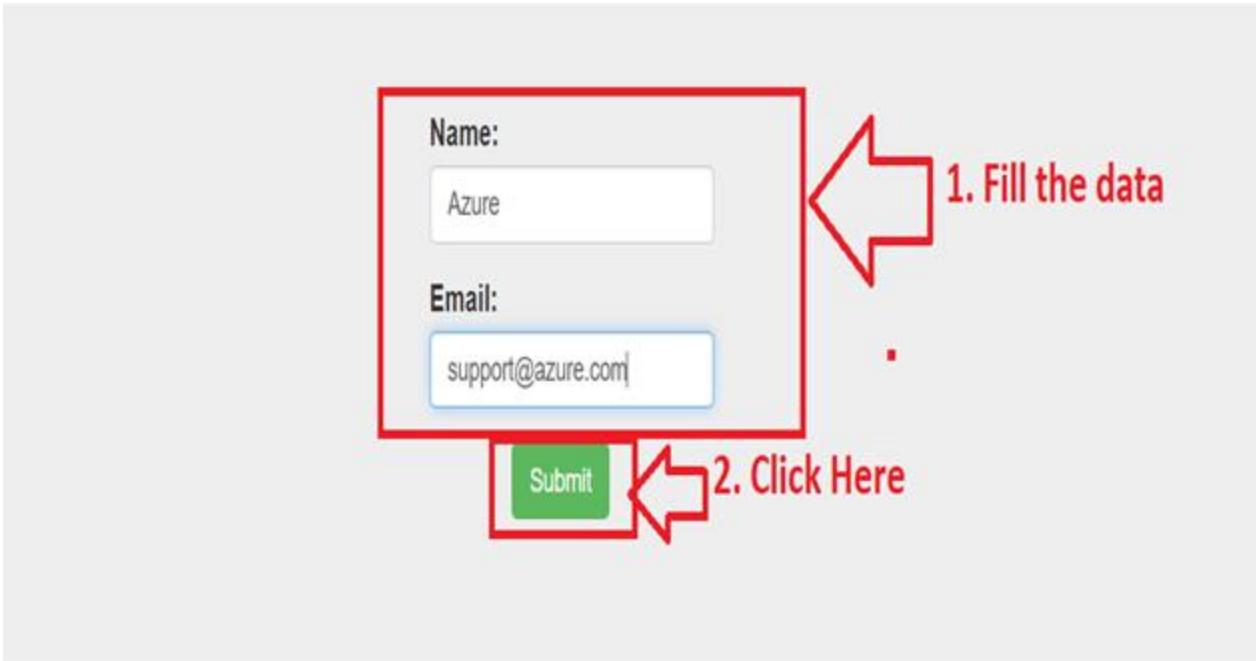
PHP & MySQL Dependencies Installed Successfully

A screenshot of a web form. The form contains two input fields: "Name:" and "Email:", both with empty text boxes. Below the email field is a green "Submit" button. The entire form area is highlighted with a thick red border.

No Database Dependencies Show Here & Error Resolved

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.



Insert the Data into a Webpage

Step 2: The record will be successfully created.

A screenshot of a web form. At the top right are two input fields labeled "Name:" and "Email:", each with a corresponding empty text input box below it. Below these fields is a green "Submit" button. At the bottom left of the page, there is a red-bordered box containing the text "New record created successfully".

Record Created Successfully

Step 3: Go to the “EC2” Machine & connect machine with database using the below-given command:

```
sudo mysql -h application-database.calupg2c7sxe.ap-south-1.rds.amazonaws.com  
-u admin -pintell123
```

The database will be successfully connected.

```
No VM guests are running outdated hypervisor (ovmf) 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> |
```

Connect to the “intel” database

Step 4: Now, go to the “intel” database using the below-given command:

```
use intel;
```

Now, we will 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

Go inside the “intel” database

Step 5: Type the below-given query:

```
select * from data;
```

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> █
```

i-0824471f793e5106d (AWS_EC2_Project1)

Public IPs: 3.110.85.213 Private IPs: 172.31.42.115

Records Inserted Successfully

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”.

The screenshot shows the AWS Management Console Services page. A yellow arrow labeled "1. Search EC2 Here" points to the search bar containing "EC2". A yellow box highlights the "EC2" service entry in the search results, with a yellow arrow labeled "2. Put Cursor Over Here" pointing to it. Another yellow arrow labeled "3. Click Here" points to the "Instances" link in the EC2 service card. The sidebar on the left lists various services like Billing, EC2, RDS, Route 53, and CloudWatch.

Go to the “Instances” section

Step 2: Select the Instance (“AWS_EC2_Project1”) & Go to the “Actions>Image and templates>Create image”.

The screenshot shows the AWS EC2 Instances page. A red box labeled "1." highlights the checkbox next to the instance "AWS_EC2_Proj...". A red box labeled "2." highlights the "Actions" button. A red box labeled "3." highlights the "Image and templates" option in the dropdown menu. A red box labeled "4." highlights the "Create image" button. The instance details show "Name: Assignment-1" and "Instance ID: i-0c06d1c9ddc101ad". The bottom navigation bar includes "Details", "Status checks", "Monitoring", and "Tags". A watermark at the bottom right says "Activate Windows Go to Settings to activate Win".

Create an Image [AMI for Instance]

Step 3: Write the “Image Name & Description” as the “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...

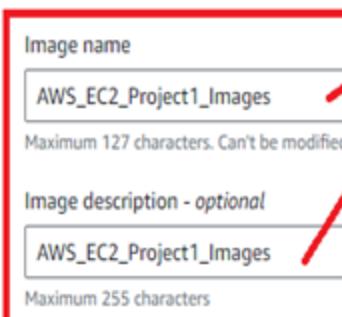
Instance ID
 i-0824471f793e5106d (AWS_EC2_Project1)

Image name Write Image Name & Description Here

AWS_EC2_Project1_Images
Maximum 127 characters. Can't be modified after creation.

Image description - optional

AWS_EC2_Project1_Images
Maximum 255 characters



Write the Image Name and 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”.

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

Add volume

1. 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.

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

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Choose the Tag image and snapshots together option

Click on the “Create image”.

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
Cancel
Go to Settings to activate Windows.
Create image

Create the AMI Image

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

Instances (2) [Info](#)

Find instance by attribute or tag (case-sensitive)

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check
<input type="checkbox"/>	Assignment-1	i-0c06d1c9ddc101ad1	Stopped	t2.micro	-
<input type="checkbox"/>	AWS_EC2_Proj...	i-0824471f793e5106d	Running	t2.micro	2/2 checks passed

Click on the AMI Hyperlink

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

AMI Status Pending

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

Amazon Machine Images (AMIs) (1) Info					C	Recycle Bin	EC2 Image Builder	Actions ▾	Launch	
					Available Status					
Owned by me ▾		<input type="text"/> Find AMI by attribute or tag			Clear filters					
AMI ID = ami-00307ae5e3b18418c	X	Source	Owner	Visibility	Status	Creation date				
es	037036564217/AWS_EC2_Project1_Im...	037036564217	Private		Available	2023/10/04 14:20 GMT+5:30				

AMI Available

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)**.

Amazon Machine Images (AMIs) (1/1) Info					C	Recycle Bin	EC2 Image Builder	Actions ▾		
					Click Here					
Owned by me ▾		<input type="text"/> Find AMI by attribute or tag			Clear filters					
AMI ID = ami-00307ae5e3b18418c	X	Name	AMI ID	AMI name	Source					
		ami-00307ae5e3b18418c		AWS_EC2_Project1_Images	037036564217/AWS_EC2_Project1_Im...					

Edit the Name

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

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

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

<input checked="" type="checkbox"/>	Name	AMI ID	AMI name	Source
<input checked="" type="checkbox"/>	AWS_EC2_Proj...	ami-00307ae5e3b18418c	AWS_EC2_Project1_Images	037036564217,

Name after AMI Id

AMI ID **ami-00307ae5e3b18418c (AWS_EC2_Project1_AMI)**

Details Permissions Storage Tags

AMI ID Put the AMI Name Here Image type Platform details

5. Create the Launch Template for Auto Scaling Groups

Step 1: Go to the “Launch Template”.

New EC2 Experience [Tell us what you think](#)

EC2 Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types **Click Here**
Launch Templates [Select an AMI](#)
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations

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

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

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

<input type="checkbox"/>	Name	AMI ID	AMI name
<input type="checkbox"/>	AWS_EC2_Proj...	ami-00307ae5e3b18418c	AWS_EC2_Project1_Images

Go to the Launch Templates Section

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

EC2 launch templates

Streamline, simplify and standardize instance launches

Use launch templates to automate instance launches, simplify permission policies, and enforce best practices across your organization. Save launch parameters in a template that can be used for on-demand launches and with managed services, including EC2 Auto Scaling and EC2 Fleet. Easily update your launch parameters by creating a new launch template version.

Click here

New launch template

Create launch template

Benefits and features

Create a launch template

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

Auto Scaling guidance [Info](#)
Select this if you intend to use this template with EC2 Auto Scaling

Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

► [Template tags](#)

3. Enable this option by clicking

Write the Template Name With Autoscaling Enabled

1. Write Template Name Here

2. Write Template Description Here

3. Enable this option by clicking

Write the Template Name With Autoscaling Enabled

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

The screenshot shows the AWS AMI search interface. At the top, there is a search bar with the placeholder text "Search our full catalog including 1000s of application and OS images". Below the search bar, there are three tabs: "Recents", "My AMIs" (which is highlighted with a red box), and "Quick Start". Under the "My AMIs" tab, there are two buttons: "Owned by me" (which is selected and highlighted with a red box) and "Shared with me". To the right of these buttons is a search icon and the text "Browse more AMIs" followed by a link "Including AMIs from AWS, Marketplace and the Community". At the bottom of the interface, there is a section titled "Amazon Machine Image (AMI)" containing a list of AMIs. The first item in the list is "AWS_EC2_Project1_Images" (highlighted with a red box), with details: AMI ID: ami-00307ae5e3b18418c, Creation Date: 2023-10-04T08:50:41.000Z, Virtualization: hvm, ENA enabled: true, Root device type: ebs.

Choose the Created AMI

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

AMI ID with Description

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	All generations
On-Demand Linux base pricing: 0.0124 USD per Hour	Compare instance types
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	

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

The screenshot shows the AWS EC2 instance creation interface. A red box highlights the 't2.micro' option under 'Instance type', with a red arrow pointing to it labeled '1. Choose t2.micro here'. Another red box highlights the 'Key pair name' field containing 'MY-WEB-SERVER', with a red arrow pointing to it labeled '2. Choose your key pair name'. The interface also includes sections for 'Subnet', 'Security group', and 'Volume' configuration.

Choose the Instance type & the key pair name

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](#)

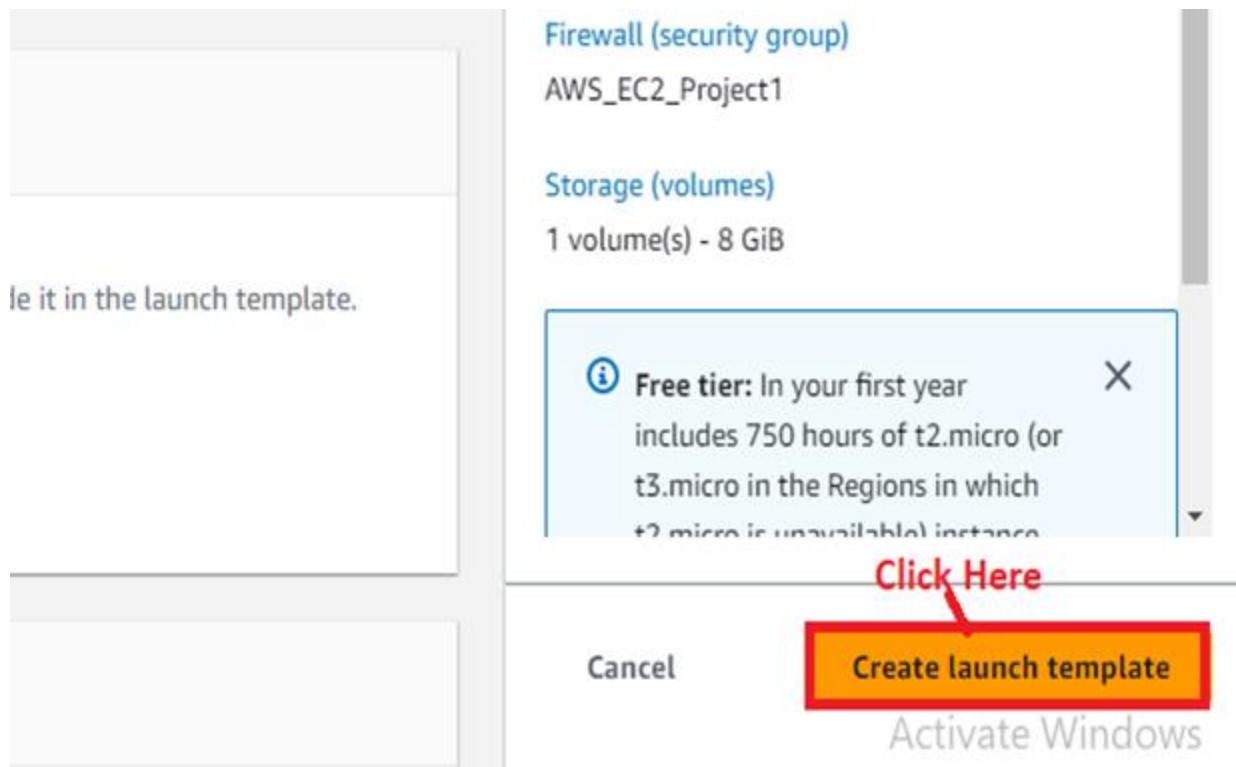
Select security groups [▼](#) [Compare security group rules](#)

AWS_EC2_Project1 sg-04d792595bc10f95d VPC: vpc-04c3e6ef2b9c7b60f

▼ Advanced network configuration **2. Choose your security group here**

Select the Existing Security Group

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



Create a Launch Template

Step 8: Your launch template

(AWS_EC2_Project_Template(lt-0798f90e53f8c8c27) will be successfully created. Click on the hyperlink.

The screenshot shows a screenshot of the AWS Management Console. At the top, there's a dark header bar with the 'Services' icon, a search bar containing 'Search' with a 'Q' icon, and a keyboard shortcut '[Alt+S]'. Below the header, the navigation path is 'EC2 > Launch templates > Create launch template'. A green success message box is displayed, containing a 'Success' icon, the word 'Success', and a message 'Successfully created [AWS EC2 Project Template\(lt-0798f90e53f8c8c27\)](#)'. A yellow arrow points from the text 'Click on the Hyperlink' to the word 'Hyperlink' in the message. Below the message box, there's a link 'Actions log' and a small 'Next steps' section.

Success

Successfully created [AWS EC2 Project Template\(lt-0798f90e53f8c8c27\)](#)

Actions log

Next steps

Click on the Hyperlink

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

EC2 > Launch templates > AWS_EC2_Project_Template

AWS_EC2_Project_Template (lt-0798f90e53f8c8c27)

Actions ▾ Delete template

Launch template details

Launch template ID lt-0798f90e53f8c8c27	Launch template name AWS_EC2_Project_Template	Default version 1	Owner arn:aws:iam::037036564217:user/Admin@AWS
--	--	--------------------------------------	---

Details Versions Template tags

Launch template version details

Version 1 (Default) ▾	Description A launch template for EC2 Project	Date created 2023-10-04T10:05:50.000Z	Created by arn:aws:iam::037036564217:user/Admin@AWS
--	--	--	--

Launch Template for Auto Scaling

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”.

Elastic IPs
Placement Groups
Key Pairs
Network Interfaces

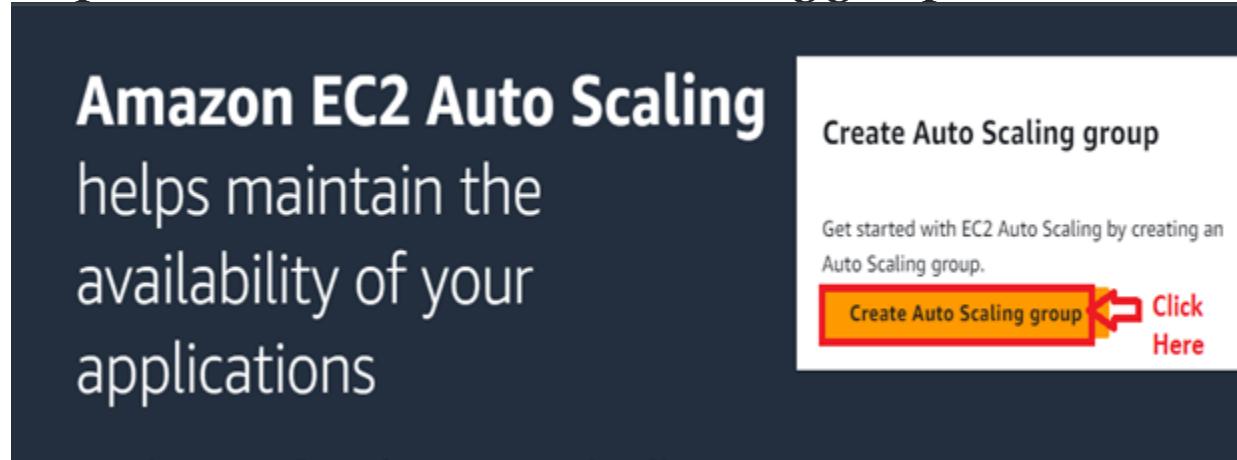
▼ Load Balancing
Load Balancers
Target Groups

▼ Auto Scaling
[Auto Scaling Groups](#) Click Here

Select a launch template

Go to the Auto Scaling Groups

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



Create the 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.	

Write the Auto Scaling Group Name Here

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 [▼](#) [C](#)

[Create a launch template](#)

Version

Default (1) [▼](#) [C](#)

[Create a launch template version](#)

Description
A launch template for EC2 Project

Launch template AWS_EC2_Project_Template [Edit](#) 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 sg-04d792595bc10f95d [Edit](#)

Choose launch template here

Choose the Launch Template

Step 5: Click on the “Next”.

Date created

Wed Oct 04 2023 15:35:50

GMT+0530 (India Standard Time)

Click Here

Cancel

Next Step

Go to Settings to

Go to the Next Steps

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 Info

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

Instance type requirements Info

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

Override launch template

Launch template	Version	Description
AWS_EC2_Project_Template <small>Info</small>	Default	A launch template for EC2 Project

It-0798f90e53f8c8c27

Instance type
t2.micro

Instance type requirements

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).

Click on the “Next”.

Network Info

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

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 X 172.31.32.0/20 Default
ap-south-1b | subnet-00cf17266e1b37cef (Zone_B) X 172.31.16.0/20

Cancel

Choose the Network Setting Options

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.

Choose the “No load balancer” option

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



Cancel

Skip to review

Previous

Next

Go to the Next Settings After Load Balancer

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	2
Minimum capacity	1
Maximum capacity	3

Choose "Desired" & "Maximum" Capacity

Group Size Selection with Capacity Options

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

Choose the No Scaling Option

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

Add notifications - optional [Info](#)

Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.

Add notification

Click Here



Cancel

Skip to review

Previous

Next

Click on the Next

Step 12: Click on the “Add tag”.

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.

Tags (0)

[Add tag](#)

 **Click Here**

50 remaining

[Add a Tag](#)

Step 13: Choose the following options here:

Key — Name

Value-optional: — asg-autoscaling-grp

Tag new instances — Leave enabled

Click on the “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.

i 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

Create a Tag

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

Create the 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 Auto Scaling groups page. At the top left, there is a breadcrumb navigation: EC2 > Auto Scaling groups. Below the navigation, the heading "Auto Scaling groups (1) Info" is displayed. A red arrow points from the text "Click Here" to the "Info" link. A search bar labeled "Search your Auto Scaling groups" is present. Under the search bar, there are filter options: "Name" (with a dropdown arrow), "Launch template/configuration" (with a dropdown arrow), and "Instances" (with a dropdown arrow). A table lists one Auto Scaling group: "AWS_EC2_Project1_ASG". This row is highlighted with a red border. To the right of the group name, it says "AWS_EC2_Project_Template | Version De 0".

Auto Scaling Group Created

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

The screenshot shows the details page for the "AWS_EC2_Project1_ASG" Auto Scaling group. At the top left, there is a breadcrumb navigation: EC2 > Auto Scaling groups > AWS_EC2_Project1_ASG. Below the navigation, the group name "AWS_EC2_Project1_ASG" is displayed, followed by a red arrow pointing to the "Click Here" link. Below the group name, there is a horizontal navigation bar with tabs: Details (selected), Activity (highlighted with a red box), Automatic scaling, Instance management, Monitoring, and Instance refresh. The "Activity" tab is currently active. The main content area is titled "Group details". It contains two columns of information:

Auto Scaling group name	Desired capacity	Status
AWS_EC2_Project1_ASG	2	Updating capacity

Below this, there is another set of information:

Date created	Minimum capacity
Fri Oct 06 2023 06:48:28 GMT+0530 (India Standard Time)	1

At the bottom of the page, there is a red box around the text "Go to the Activity Section".

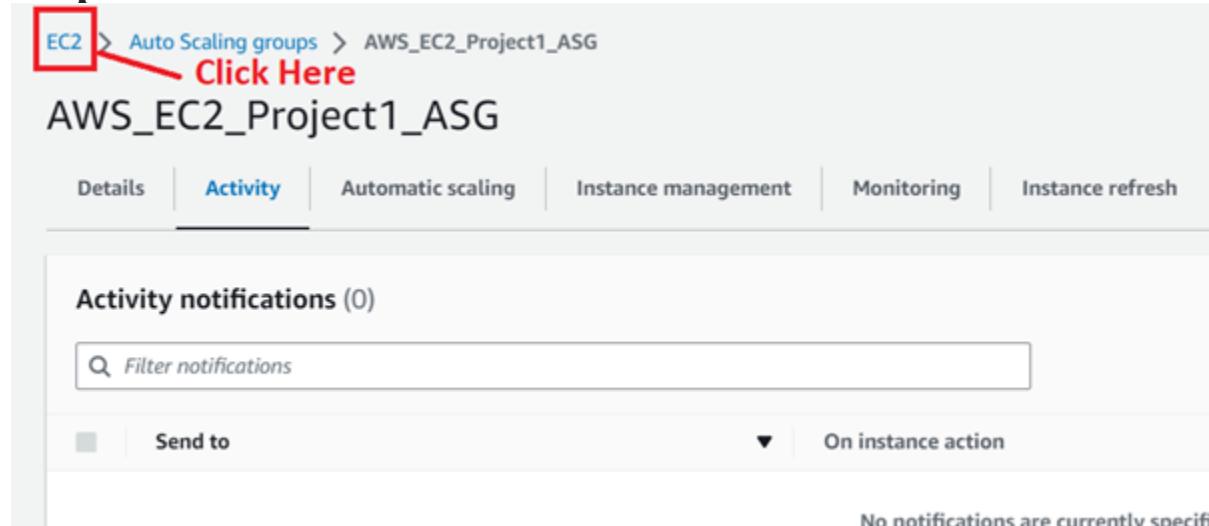
Go to the Activity Section

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

Activity history (2)				
Instances				
Status	Description	Cause	Start time	
Successful	Launching a new EC2 instance: i-05aadcd03887b7ab07	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	

EC2 Instance Created

Step 18: Click on the “EC2”.



EC2 > Auto Scaling groups > AWS_EC2_Project1_ASG
Click Here

AWS_EC2_Project1_ASG

Details Activity Automatic scaling Instance management Monitoring Instance refresh

Activity notifications (0)

Filter notifications

Send to On instance action

No notifications are currently specified

Go to the EC2

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

Click Here

You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region:

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

Go to the Running Instances Section

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

Auto Scaling Instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
asg-autoscalin...	i-06954abe8b725bb84	Running	t2.micro	2/2 checks passed	No alarms +
AWS_EC2_Proj...	i-0824471f793e5106d	Running	t2.micro	2/2 checks passed	No alarms +
asg-autoscalin...	i-05aadc03887b7ab07	Running	t2.micro	2/2 checks passed	No alarms +

Running Instances

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”.

Elastic IPs
Placement Groups
Key Pairs
Network Interfaces

▼ Load Balancing
Load Balancers
Target Groups  **Click Here**

▼ Auto Scaling
Auto Scaling Groups

Select an instance

Go to the Target Groups

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

 **Click Here**

Create target group

No target groups
You don't have any target groups in ap-south-1

Create target group

Create a Target Group

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

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Settings in this section can't be changed after the target group is created.

Choose "instances"

Choose a target type

Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.

Choose the “instances” as 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

1. Choose Target Group Name Here

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

Protocol

Port

HTTP

:

80

1-65535

2. Leave as it is, No change here

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 instance's 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

Write the Target Group Name with the Desired Options

Step 5: Click on the “Next”.

Attributes

Note: 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**

Go to the Next Steps in Target Groups

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. Choose this Instance

1 selected

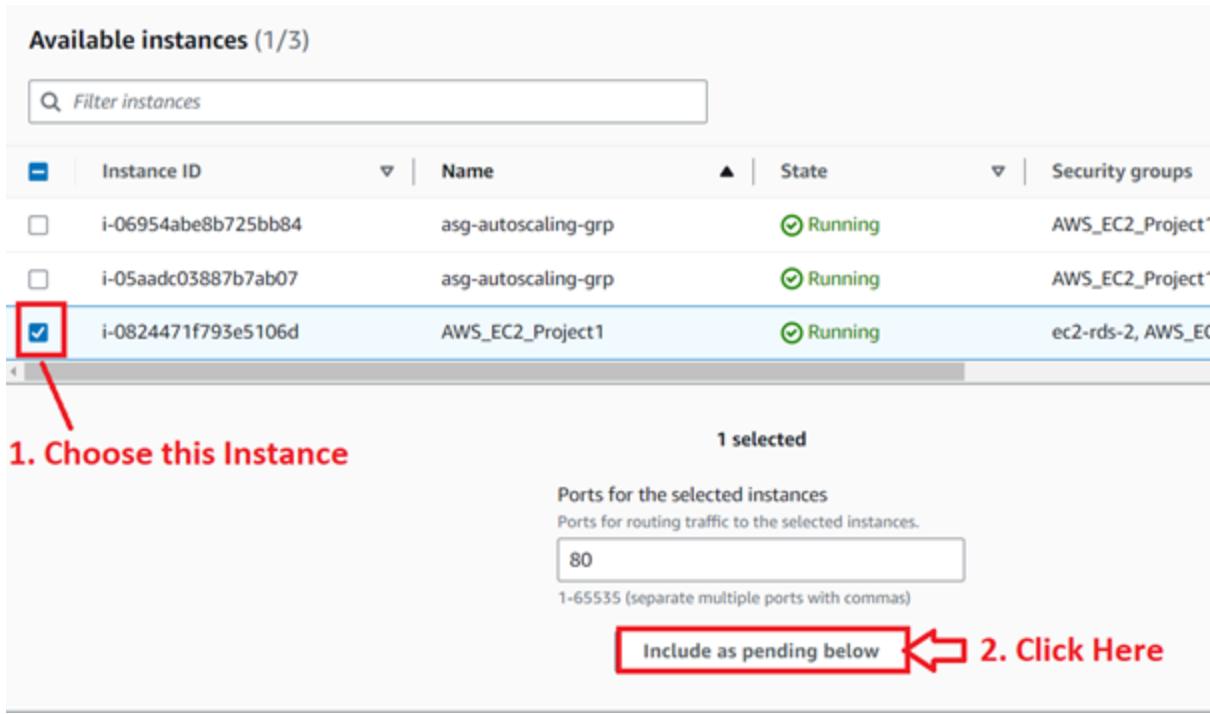
Ports for the selected instances
Ports for routing traffic to the selected instances.

80

1-65535 (separate multiple ports with commas)

Include as pending below 

2. Click Here



Include the Main Instance

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

Targets (1)

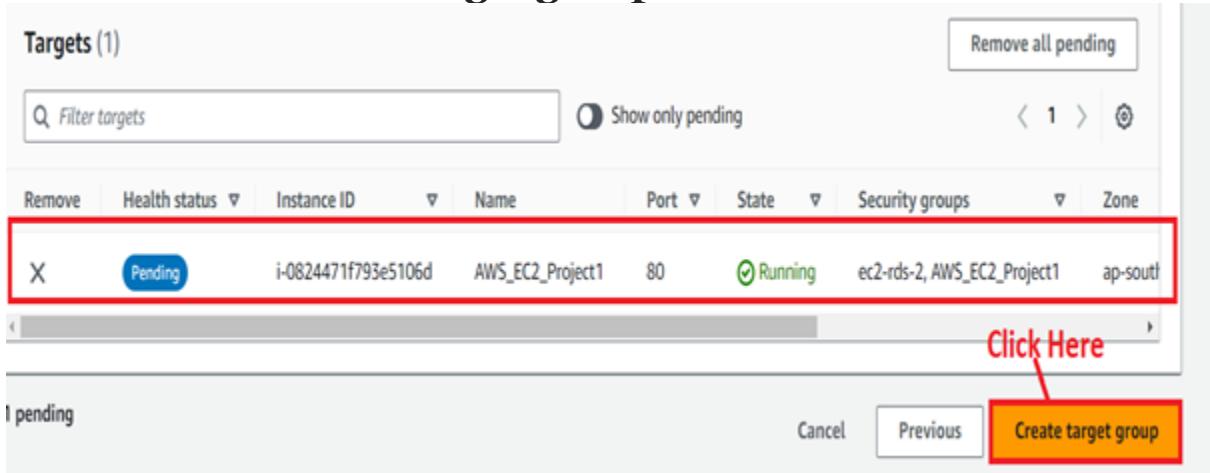
Remove all pending

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

Click Here 

1 pending

Cancel Previous **Create target group**



Create the 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](#)

C A

Filter target groups

<input type="checkbox"/>	Name	▼	ARN	▼	Port	▼	Protocol	▼	Target type
<input type="checkbox"/>	EC2-Project-Target-Group		<input type="text"/> arn:aws:elasticloadbalancing:ap-south-1:037036564217:targetgroup/EC2-Project-Group/70a42334c12f2a25		80		HTTP		Instance

Target Group Successfully Created

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-Group/70a42334c12f2a25

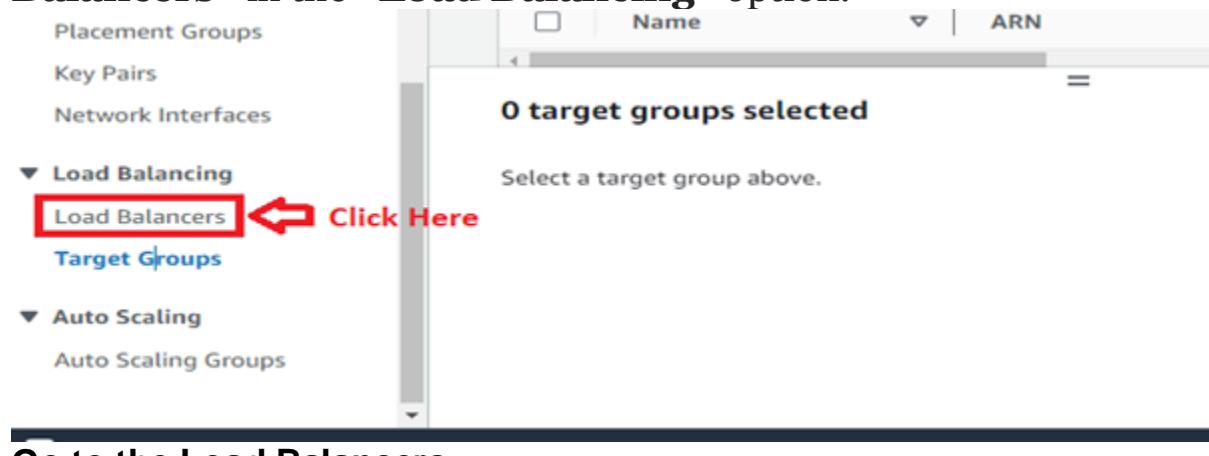
Target type	Protocol : Port	Protocol version
Instance	HTTP: 80	HTTP1
IP address type	Load balancer	
IPv4	None associated	
Total targets	Healthy	Unhealthy
1	⌚ 0	⌚ 0
		Unused
		⌚ 1

EC2 target group

In “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”.

EC2 > Load balancers

Click Here

Create load balancer

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

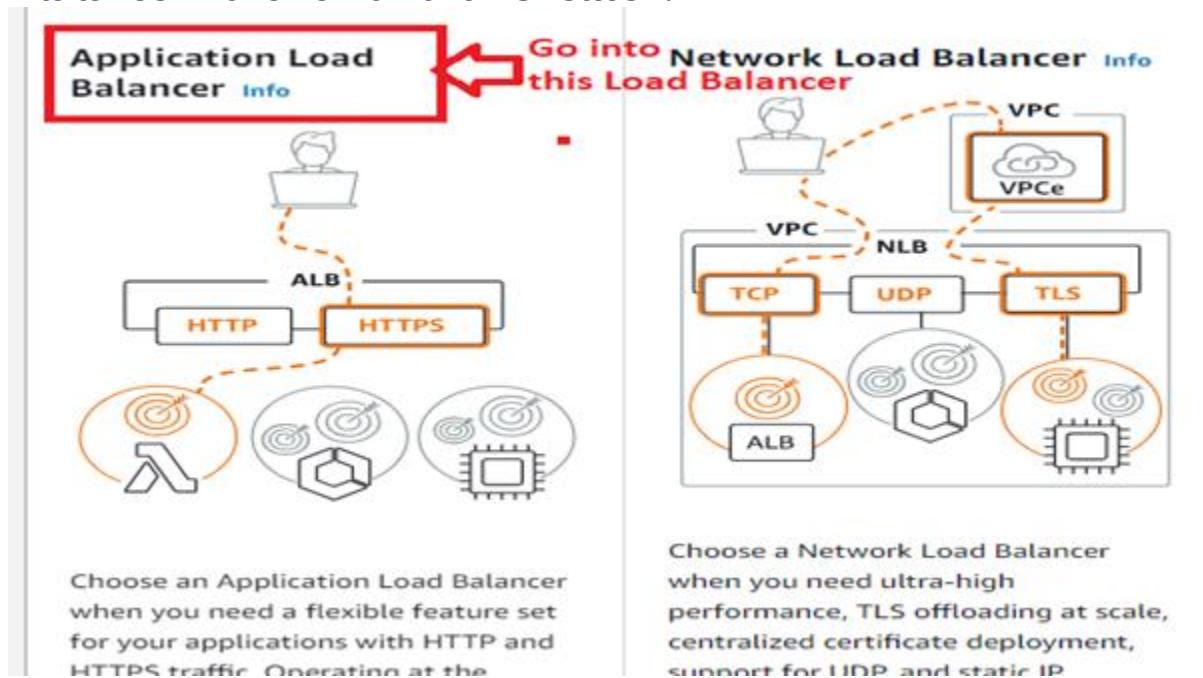
Filter Load balancers

Name | DNS name | State | VPC

0 load balancers selected

Create a 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

Click on the 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
e

Choose the Basic Load Balancer Configuration

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 zones 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

Choose the Subnets with Its Zones

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 one.

Choose your security group here

Security groups

Select up to 5 security groups

AWS_EC2_Project1 X
sg-04d792595bc10f95d VPC: vpc-04c3e6ef2b9c7b60f

Choose the Security Group in the Load Balancer

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

Listeners and routing [Info](#)

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

Choose "Target Group" Here

▼ Listener HTTP:80

Protocol: HTTP Port: 80
Default action: Forward to: EC2-Project-Target-Group
HTTP
Create target group

Choose the Listener with the target group

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

ur load balancer. You can view and edit them after creating the load

Click Here



Cancel

Create load balancer

Create the Load Balancer

Step 9: Your load balancer will be successfully created.

Click on the “Hyperlink” to view the load balancer details.

A screenshot of the AWS EC2 Load Balancers page. At the top, there's a navigation bar with the AWS logo, Services, Search, and other account information. Below it, a green success message box contains the text: "Successfully created load balancer: AWS-EC2-Project-Load-Balancer" followed by a yellow "Click Here" link. A yellow arrow points to this link. Below the message, 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 screenshot, the breadcrumb navigation shows: EC2 > Load balancers > AWS-EC2-Project-Load-Balancer > Create Application Load Balancer. There's also a "Suggested next steps" section with an info icon.

Click on the Hyperlink in the Load Balancer

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

Load balancers (1)					
	Name	DNS name	State	VPC	
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.					
<input type="checkbox"/>	AWS-EC2-Project-Load-Balancer	AWS-EC2-Project-Load-Bal...	<input checked="" type="checkbox"/> Active	vpc-0-	

Load Balancer Created

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”.

- Placement Groups
- Key Pairs
- Network Interfaces
- ▼ Load Balancing
 - [Load Balancers](#)
 - [Target Groups](#)
- ▼ Auto Scaling
 - [Auto Scaling Groups](#)

	Name	DNS name
<input type="checkbox"/>	AWS-EC2-Project-Load-Balancer	AWS-EC2-Project-Load-Ba

=

0 load balancers selected

Select a load balancer above.

Click Here

Go to the Auto Scaling Groups

Step 2: Click on the “AWS_EC2_Project1_ASG”.

The screenshot shows the AWS Auto Scaling Groups management interface. At the top left, there's a breadcrumb navigation: EC2 > Auto Scaling groups. Below it, a header bar includes a refresh icon, tabs for Launch configurations and Launch templates, and an Actions dropdown. A prominent orange button labeled "Create Auto Scaling group" is visible. A search bar with the placeholder "Search your Auto Scaling groups" is also present. The main content area displays a table of Auto Scaling groups. The first row, which contains the name "AWS_EC2_Project1_ASG", is highlighted with a red box and has a red arrow pointing to it from the text "Click here". The table columns include checkboxes, the Name column, Launch template/configuration dropdown, and instance counts. The "AWS_EC2_Project1_ASG" row also shows "AWS_EC2_Project_Template | Version De" and "2". Navigation controls like a previous/next button and a refresh icon are at the bottom of the table.

Click on the Auto Scaling Group Name

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

Go to the “Details” Section

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

Instance type requirements

Edit

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

Load balancing

Edit

Load balancer target groups

Classic Load Balancers

Click here

Click on the Edit

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 2. Select the target group here

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

Select target groups 3. Click Here

EC2-Project-Target-Group | HTTP 3. Click Here

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

Classic Load Balancers

Create and attach new load balancers

Add a new load balancer

Cancel Update

Enable the Load Balancer & Update the Settings

Step 6: Your load balancer will be successfully attached.

⌚ Auto Scaling group updated successfully

Load balancing	
Load balancer target groups EC2-Project-Target-Group	Classic Load Balancers -

Health checks

Health checks	
-	Edit

Load Balancer Attached to the Auto Scaling Groups Successfully

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 interface. On the left, there's a navigation sidebar with options like Placement Groups, Key Pairs, Network Interfaces, Load Balancing (expanded), Load Balancers (highlighted with a red box and a red arrow pointing to it labeled "Click Here"), Target Groups, Auto Scaling, and Auto Scaling Groups. The main content area has a header with columns for Name and ARN, followed by a message: "0 target groups selected" and "Select a target group above." A vertical scrollbar is visible on the right side of the content area.

Go to the Load Balancer Option

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

Load balancers (1)									
		Actions ▾		Create load balancer ▾					
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.									
<input type="text"/> Filter Load balancers									
	Name	DNS name	State	VPC					
<input type="checkbox"/>	AWS-EC2-Project-Load-Balancer	AWS-EC2-Project-Load-Bal...	Active	vpc-0...					

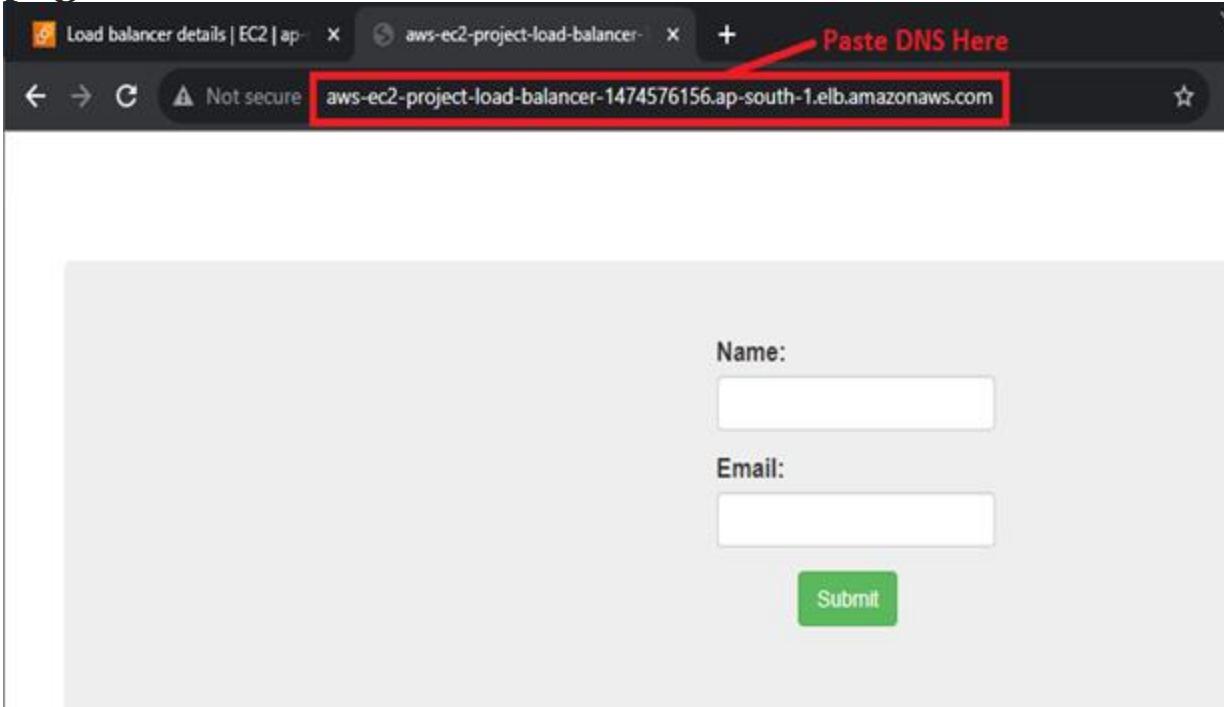
Click on the Load Balancer

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

Load balancer type	Status
Application	<input checked="" type="checkbox"/> Active
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	DNS name Info
<input type="checkbox"/> arn:aws:elasticloadbalancing:ap-south-1:037036564217:loadbalancer/app/AWS-EC2-Project-Load-Balancer/6bd68a365672496e	<input type="checkbox"/> AWS-EC2-Project-Load-Balancer-1474576156.ap-south-1.elb.amazonaws.com (A Record)

Copy the Load Balancer DNS Name

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

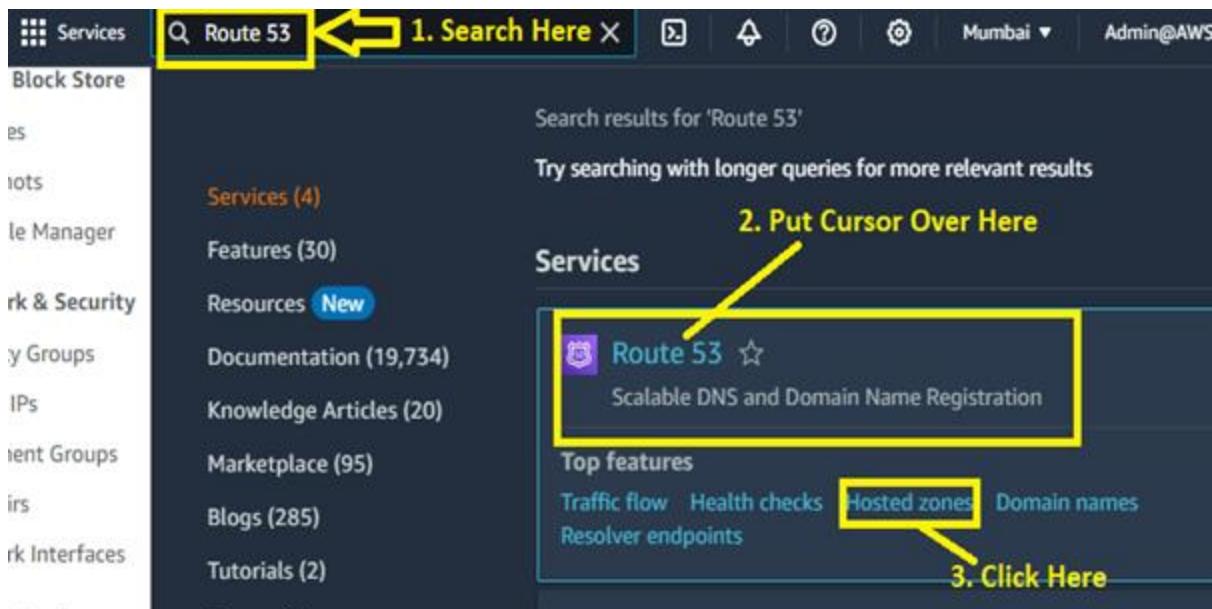


Paste the Load Balancer in the Browser

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”.



Go to the Route 53 Service

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 'Hosted zones (1)' list in the AWS Route 53 service. The list includes one item: 'visal.click'. A red arrow labeled 'Click here' points to the 'visal.click' entry. The list has columns for 'Hosted zone name', 'Type', 'Create...', and 'Records'. The 'visal.click' entry shows 'Public' in the Type column and 'Route 53' in the Create... column. The 'Records' column shows the number '2'.

Click on the Hosted Zone Name

Step 3: Click on the “Create record”.

The screenshot shows the AWS Route 53 'Records' page. At the top, there are buttons for 'Delete record' and 'Import zone file', followed by a large yellow 'Create record' button which is highlighted with a red box and a red arrow pointing to it. Below these buttons are search and filter fields for 'Filter records by property or v', 'Type', 'Routing policy', and 'Alias'. The main table lists two existing records: 'visal.click' (NS type) and 'visal.click' (SOA type). A 'Create a Record' button is located at the bottom left of the table area.

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

Create a Record

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

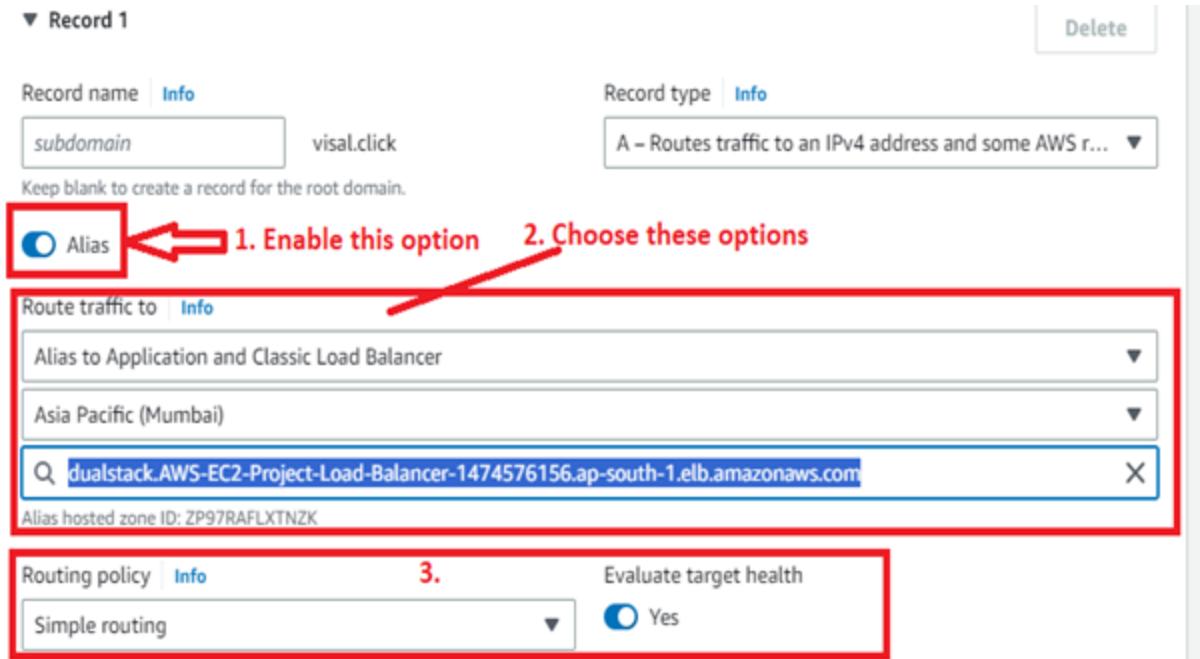
▼ Record 1 Delete

Record name Info visal.click
Keep blank to create a record for the root domain.

Record type Info A – Routes traffic to an IPv4 address and some AWS r...
 Alias 1. Enable this option 2. Choose these options

Route traffic to Info
Alias to Application and Classic Load Balancer
Asia Pacific (Mumbai)
Q X
Alias hosted zone ID: ZP97RAFLXTNZK

Routing policy Info 3. Evaluate target health
Simple routing Yes



Fill the Record Details

Routing policy – Simple routing

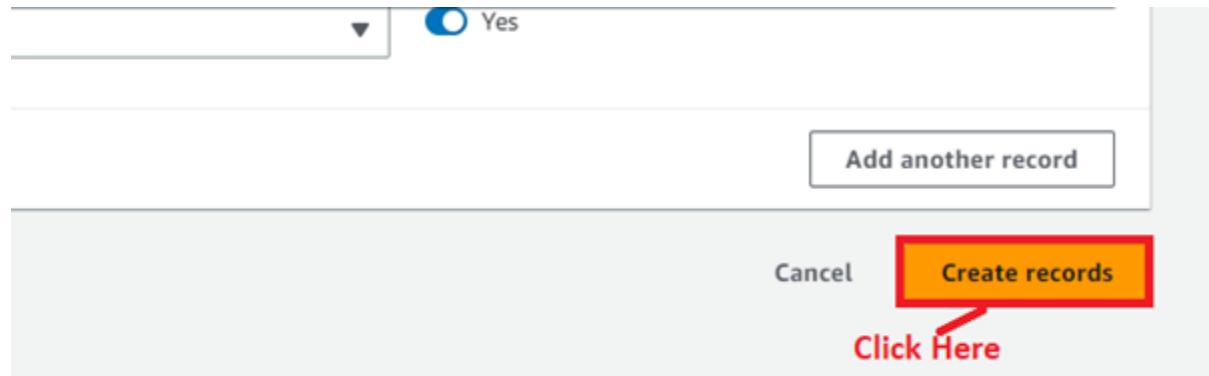
Click on the “Create records”.

Yes

Add another record

Cancel Create records

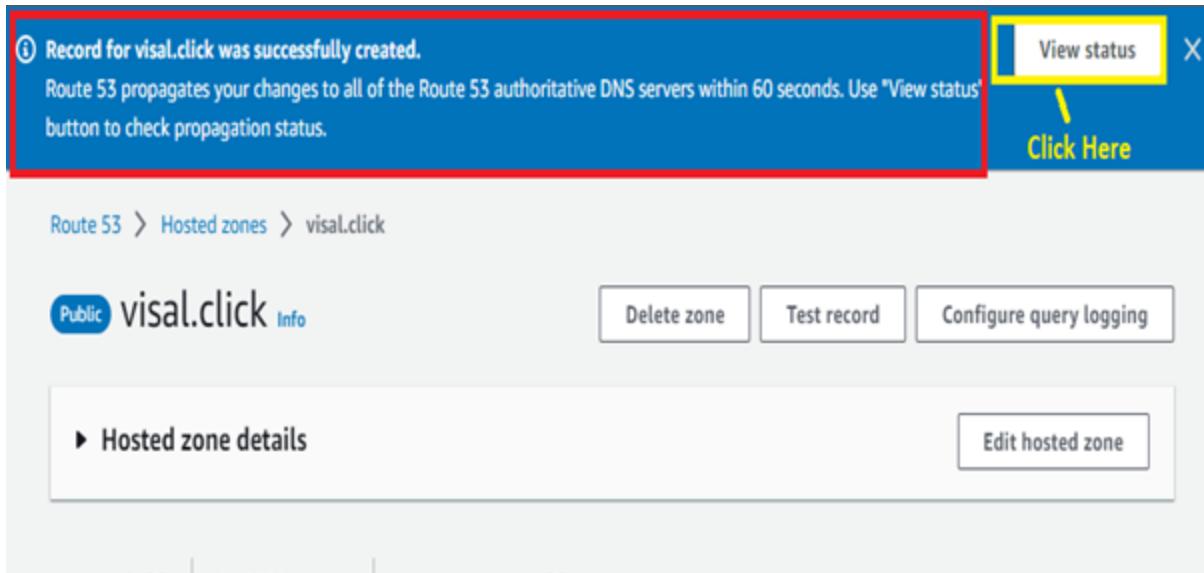
Click Here



Create a Record

Step 5: The record will be successfully created.

Click on the “View status”. It will take 60 seconds to be active.



Click on the View Status

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

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

C03999983LIT7D2GS63I7 [Info](#)

Change info details

ID

/change/C03999983LIT7D2GS63I7

Submitted at

October 06, 2023, 17:11 (UTC:+05:30)

Status

PENDING

Pending Status

Comment

-

Record Status Pending

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

/change/C03999983LIT7D2GS63I7

Submitted at

October 06, 2023, 17:11 (UTC:+05:30)

Status

 INSYNC

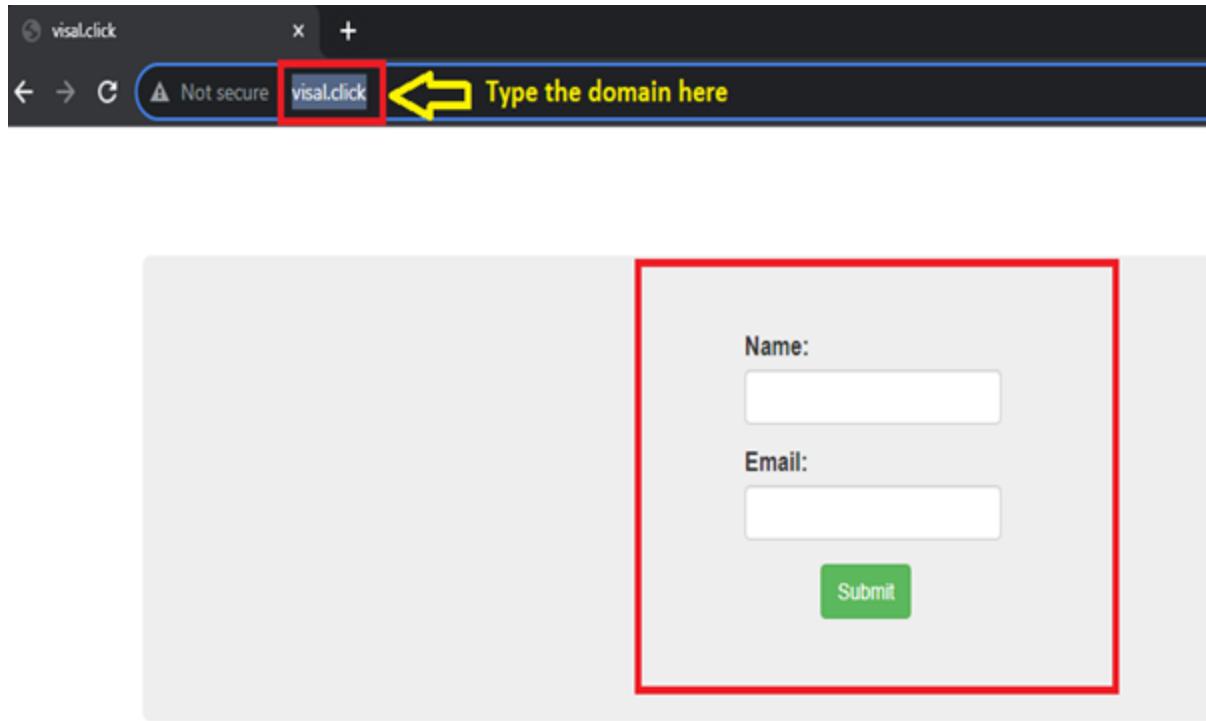
Active

Comment

-

Record is 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).

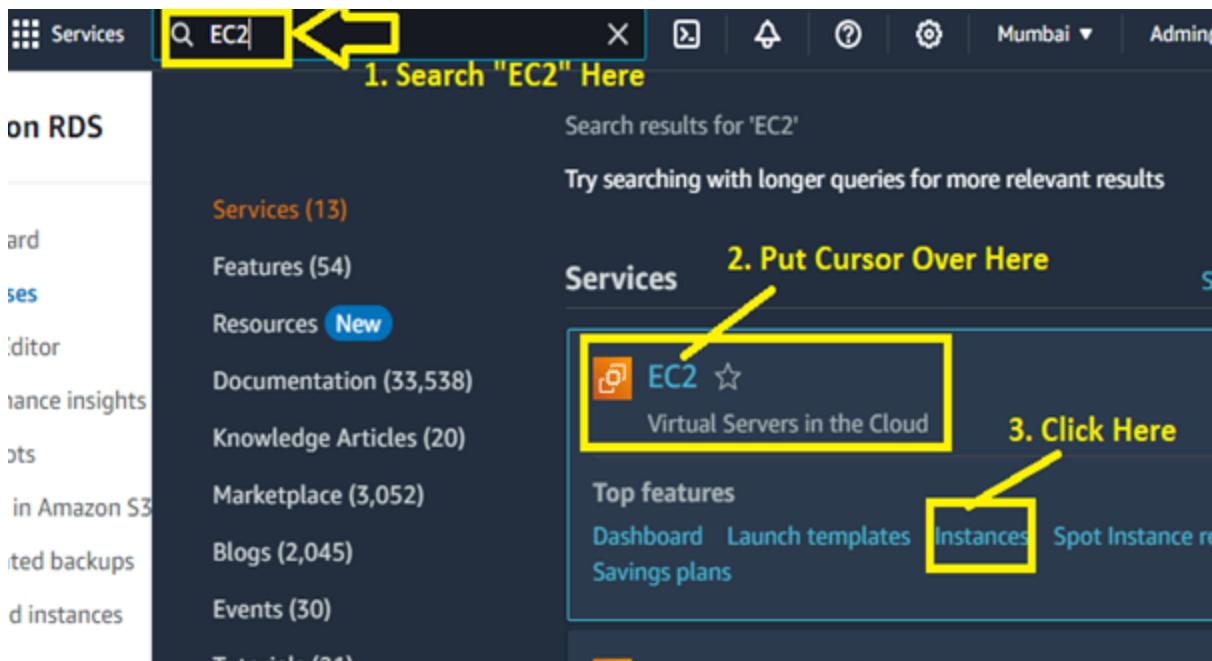


Type the Domain in the Browser & PHP Website Shown

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**”.



Go to the Instances

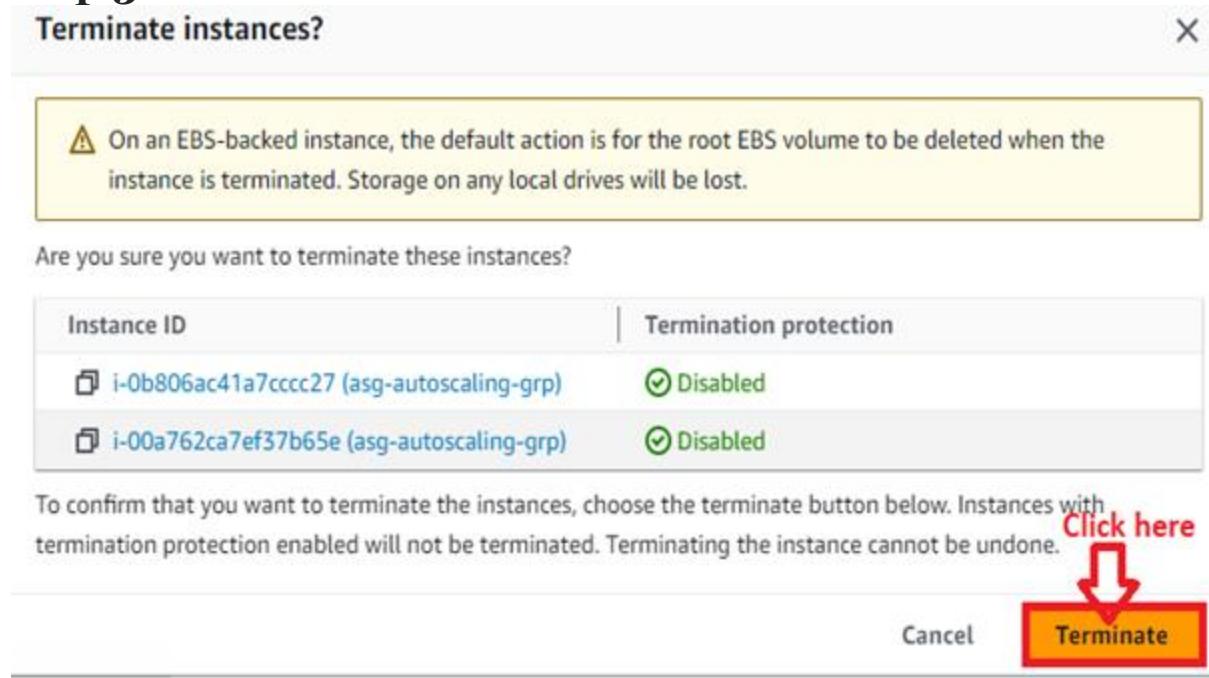
Step 2: We will select both the “asg-autoscaling-grp” Instances & click on the “Instance State”. Terminate the instance by clicking on the Terminate instance.

The screenshot shows the AWS EC2 Instances page with two instances selected for termination. A red box highlights the "Name" column, and another red box highlights the "Terminate instance" button. The "Instance state" dropdown is also highlighted with a red box and labeled "2.". The table below shows the details of the selected instances:

Name	Instance state	Instance type
asg-autoscalin...	Running	t2.micro
Assignment-1	Stopped	t2.micro
AWS_EC2_Proj...	Running	t2.micro
asg-autoscalin...	Running	t2.micro

Terminate Both the Running Instances

Step 3: Click on the “Terminate”.



Terminate the Instance Finally

Step 4: Your instance will be successfully terminated.

Instances (4) Info						
<input type="checkbox"/>	Name	1. ▾	Instance ID	Instance state	Actions ▾	Launch instances ▾
<input type="checkbox"/> Find instance by attribute or tag (case-sensitive)						
<input type="checkbox"/>	Name	1. ▾	Instance ID	Instance state	Actions ▾	Status
<input type="checkbox"/>	asg-autoscalin...	i-0b806ac41a7cccc27	Terminated	Terminated	Details Logs	t2.micro
<input type="checkbox"/>	Assignment-1	i-0c06d1c9ddc101ad1	Stopped	Stopped	Details Logs	t2.micro
<input type="checkbox"/>	AWS_EC2_Proj...	i-0824471f793e5106d	Running	Running	Details Logs	2/2
<input type="checkbox"/>	asg-autoscalin...	i-00a762ca7ef37b65e	Terminated	Terminated	Details Logs	t2.micro

=
Instance Will be Successfully Terminated

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) Info						
<input type="checkbox"/>	Name	Instance ID	Instance state	Actions	Launch instances	▼
<input type="text"/> Find instance by attribute or tag (case-sensitive)						
<input type="checkbox"/>	asg-autoscalin... 1.	i-0aae793c1ff842e1	Running	Stop Start	t2.micro	Init
<input type="checkbox"/>	Assignment-1	i-0c06d1c9ddc101ad1	Stopped	Start Stop	t2.micro	-
<input type="checkbox"/>	AWS_EC2_Proj...	i-0824471f793e5106d	Running	Stop Start	t2.micro	2/2
<input type="checkbox"/>	asg-autoscalin... 2.	i-00a762ca7ef37b65e	Terminated	Start Stop	t2.micro	-
<input type="checkbox"/>	asg-autoscalin...	i-05fb535b584884a89	Running	Stop Start	t2.micro	2/2

New Instances Launched

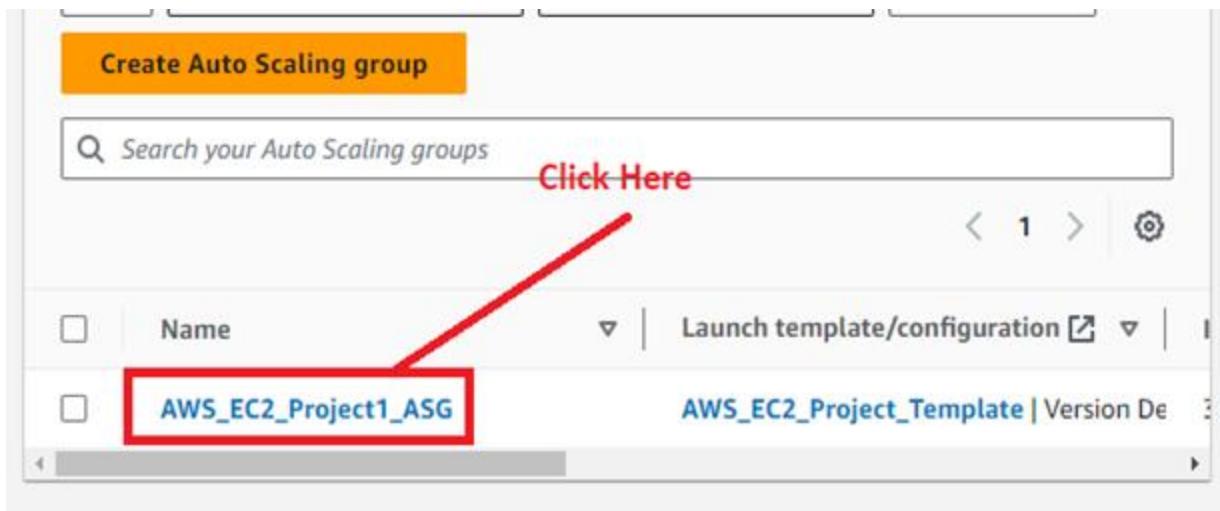
Step 6: Go to the left side & click on the “Auto Scaling Groups” in the “Auto Scaling”.

The screenshot shows the AWS CloudWatch Metrics console. On the left, there's a navigation sidebar with links like 'Placement Groups', 'Key Pairs', 'Network Interfaces', 'Load Balancing' (with 'Load Balancers' and 'Target Groups' sub-links), and 'Auto Scaling' (with 'Auto Scaling Groups' sub-link). A red arrow points to the 'Auto Scaling Groups' link. To the right, there's a summary panel with metrics: 'Instances' (6), 'Load balancers' (1), 'Security groups' (15), and 'Volumes' (4). At the bottom, there's a 'Launch instance' button.

Click Here

Go to the Auto Scaling Groups

Step 7: Click on the “AWS_EC2_Project1_ASG”.



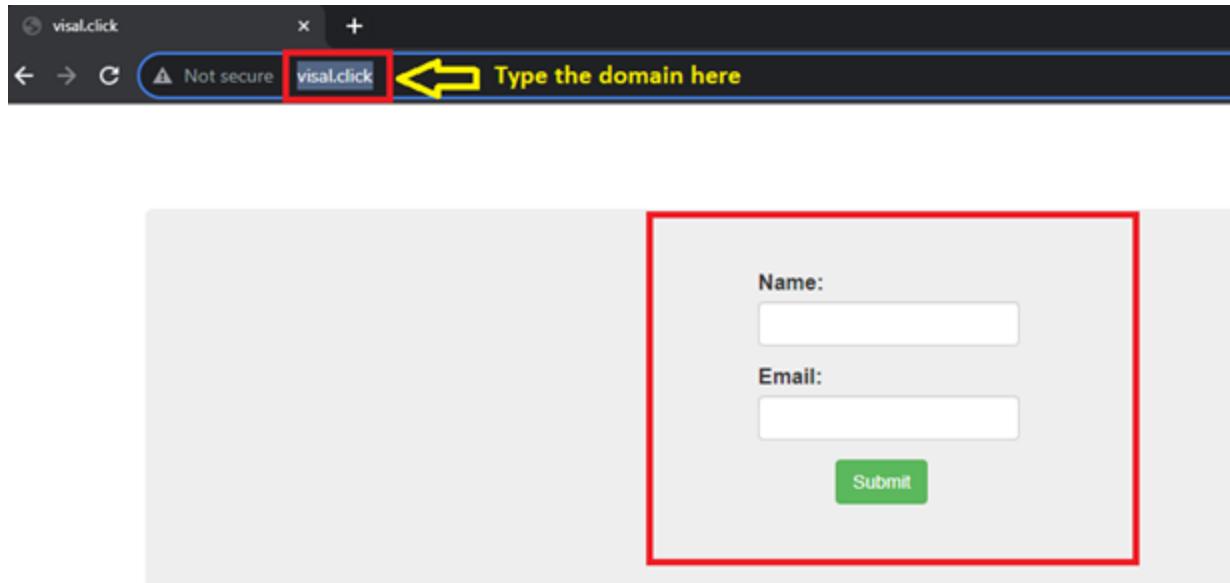
Click on the “Auto Scaling Group Name”

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-0aade793c1ff842e1	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.

Instance Terminated & New Instance Created

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



PHP Website Running

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

Click on the “Submit”.

Name:
Google

Email:
support@google.com

Submit

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

Submit the Details

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

Name:

Email:

Submit

Success Message

New record created successfully

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 below-given 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 (omniv) 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> |
```

Connect with the Database

Step 4: Now, go to the “intel” database. Use the below-given command to go inside the intel database:

```
use intel;
```

Now, we are inside the “intel” database.

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

i-0824471f793e5106d (AWS_EC2_Project1)

PublicIPs: 3.110.85.213 PrivateIPs: 172.31.42.115

Database Changed

Step 5: When we run the below-given command:

```
select * from data;
```

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

Show the Database Details

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

Check the Other EC2 Tasks:

[How to Replicate the AWS EC2 Instance to Other Region & Take the EBS Volume Backup – EC2 Case Study](#)