

## ***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

**Step 1: launch EC2 instance , create new security group, and install apache2 server**

Security group name - required

AWSproject1

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

Description - required info

Aws project1

Inbound Security Group Rules

▶ Security group rule 1 (TCP, 22, 0.0.0/0)

Remove

▶ Security group rule 2 (TCP, 443, 0.0.0/0)

Remove

▼ Security group rule 3 (TCP, 3306, 0.0.0/0)

Remove

Type info

MYSQL/Aurora

Protocol info

TCP

Port range info

3306

Source type info

Anywhere

Source info

Q Add CIDR, prefix list or security g.

0.0.0.0/0 ✕

Description - optional info

e.g. SSH for admin desktop

Summary

Number of instances info

1

Software Image (AMI)  
Canonical, Ubuntu, 22.04 LTS, ...read more  
ami-005fcdcf236362e99f

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
New security group

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

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

Cancel Launch instance Preview code

Now connect with the ec2 instance and install apache 2 server

Command -->

## Sudo apt update

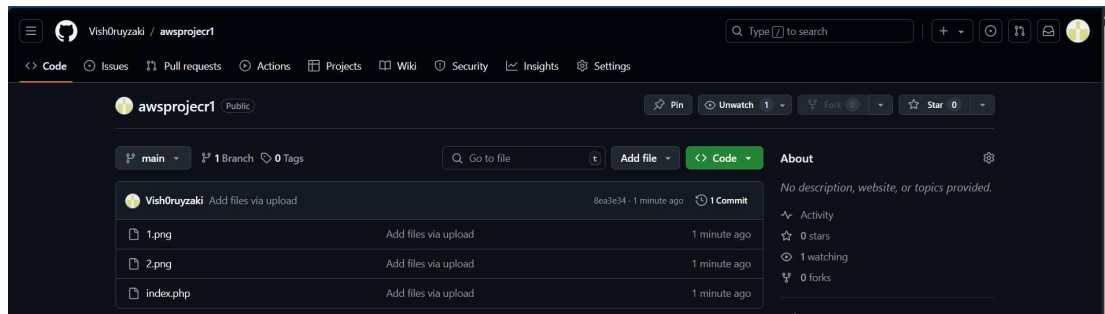
```
Sudo apt install apache2 -y
```

**i-0250c1a485858a544 (AWS project 1)**  
PublicIPs: 34.201.129.93 PrivateIPs: 172.31.88.92

i-0250c1a485858a544 (AWS project 1)

## Step2: Deploy PHP Website on Ec2 server

To deploy the website first upload the all the required files on git then clone it on your ec2 instance



We will remove the previous html file before cloning the repo

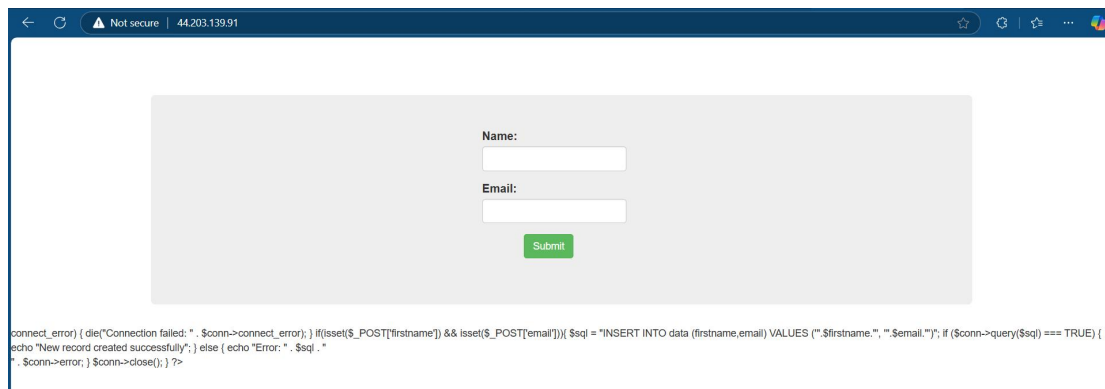
```
ubuntu@ip-172-31-88-92:~$ cd /var/www/html
ubuntu@ip-172-31-88-92:/var/www/html$ ls
index.html
ubuntu@ip-172-31-88-92:/var/www/html$ sudo rm index.html
ubuntu@ip-172-31-88-92:/var/www/html$ ls
ubuntu@ip-172-31-88-92:/var/www/html$
```

```
ubuntu@ip-172-31-88-92:/var/www/html$ sudo git clone https://github.com/Vish0ruyzaki/awsprojecr1.git
Cloning into 'awsprojecr1'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
```

```
ubuntu@ip-172-31-88-92:/var/www/html$ ls
awsprojecr1
ubuntu@ip-172-31-88-92:/var/www/html$ cd awsprojecr1
ubuntu@ip-172-31-88-92:/var/www/html/awsprojecr1$ ls
1.png 2.png index.php
```

```
ubuntu@ip-172-31-88-92:/var/www/html$ ls
1.png 2.png index.php
ubuntu@ip-172-31-88-92:/var/www/html$
```

Paste the public ip and u will see below php page on the website



## Step 3: - Create Mysql database and connect with the ec2 instance

Engine - MYSQL

Template - free tier

DB instance identifie - aws project 1

Credential setting - username - vishwa password - \*\*\*\*\*

Connectivity

### Connectivity Info

**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.

AWS project 1

**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.

### MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Remove auto backup and encryption

Rest keep default

### Databases (1)

☒ Group resources

[Modify](#) [Actions](#) [Restore from S3](#) [Create database](#)

☐ DB identifier

Status

Role

Engine

Region ...

Size

Recommendations

<a href="#">aws-project1</a>	Available	Instance	MySQL Co...	us-east-1b	db.t4g.mi...
------------------------------	-----------	----------	-------------	------------	--------------

Click on the hyperlink and we can see the endpoint which we can use to connect ec2 instance with mysql

### Amazon RDS

[Dashboard](#)[Databases](#)[Query Editor](#)[Performance insights](#)[Snapshots](#)[Exports in Amazon S3](#)[Automated backups](#)[Reserved instances](#)[Proxies](#)  
[Subnet groups](#)[Parameter groups](#)[Option groups](#)[Custom engine versions](#)[Zero-ETL integrations](#)  
[Events](#)[Event subscriptions](#)

### aws-project1

**Summary**

DB identifier  
aws-project1

CPU  
11.14%

Status  
Available

Class  
db.t4g.micro

Role  
Instance

Current activity  
0 Connections

Engine  
MySQL Community

Region & AZ  
us-east-1b

Recommendations

**Connectivity & security**

**Endpoint & port**

Endpoint  
[aws-project1.cvq8s2cq5ld.us-east-1.rds.amazonaws.com](#)

Port  
3306

**Networking**

Availability Zone  
us-east-1b

VPC  
[vpc-0240f8850455676cc](#)

Subnet group  
[rds-ec2-db-subnet-group-1](#)

Subnets  
[subnet-0886ed2b26247fec3](#)

**Security**

VPC security groups  
[rds-ec2-1 \(sg-06fe92a0bfb1b8fd0\)](#)  
[Active](#)  
[AWSProject1 \(sg-0c75884f49c80aedf\)](#)  
[Active](#)

Publicly accessible  
No

Certificate authority  
[Info](#)  
rds-ca-rsa2048-q1

## Step 4: install mysql server and then we will connect the database

```
ubuntu@ip-172-31-88-92:~$ sudo apt install mysql-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

i-0250c1a485858a544 (AWS project 1)

Now change the directory then connect with database

Use below command to connect with the datase

```
sudo mysql -h aws-project1.cvq8s2qcq5ld.us-east-1.rds.amazonaws.com -u vishwa -pSingh9047371241
```

```
ubuntu@ip-172-31-88-92:~$ cd /var/www/html
ubuntu@ip-172-31-88-92:/var/www/html$ ls
1.png 2.png index.php
ubuntu@ip-172-31-88-92:/var/www/html$ sudo mysql -h aws-project1.cvq8s2qcq5ld.us-east-1.rds.amazonaws.com -u vishwa -pSingh9047371241
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 30
Server version: 8.0.39 source distribution

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

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

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

mysql>
```

i-0250c1a485858a544 (AWS project 1)

Type command show databases;

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| intel |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)
```

mysql>

i-0250c1a485858a544 (AWS project 1)

We will switch to intel database then we will create table inside it

```
mysql> use intel;
Database changed
mysql> create table data (firstname varchar(20), email varchar(20));
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> show tables;
+-----+
| Tables_in_intel |
+-----+
| data |
+-----+
1 row in set (0.01 sec)
```

mysql>

i-0250c1a485858a544 (AWS project 1)

Insert data inside the table

```
mysql> insert into data values ('AWS', 'support@aws.com');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from data;
+-----+-----+
| firstname | email |
+-----+-----+
| AWS | support@aws.com |
+-----+-----+
1 row in set (0.01 sec)
```

mysql>

i-0250c1a485858a544 (AWS project 1)

Exit from mysql then nano index.php and edit the servername - username and password

```
<?php
$firstname=$_POST['firstname'];
$email=$_POST['email'];
$servername = "aws-project1.cvq8s2gqg5ld.us-east-1.rds.amazonaws.com";
$username = "vishwa";
$password = "vishwa";
$db = "intel";
// Create connection
$conn = new mysqli($servername, $username, $password, $db);

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

## Step 5: adding dependencies

Name:

Email:

Submit

We can see that there is some error on the webpage, hence we will install dependencies to remove the error

Use below command

`sudo add-apt-repository -y ppa:ondrej/php`

```
ubuntu@ip-172-31-88-92:/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 Supported Ubuntu Releases (https://wiki.ubuntu.com/Releases) are provided.
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
BUGS&FEATURES: This PPA now has a issue tracker:
https://deb.sury.org/#bug-reporting
```

Now install mysql client

`sudo apt install php5.6 mysql-client php5.6-mysql`

← → ↻ ⚠ Not secure 54.86.9.69 ☆ 📄 👤 ⋮

Google Chrome isn't your default browser [Set as default](#) ✕

Name:

Email:

Submit

We can see the error has been removed

### Step 6: add the data via website then check on database server

Google Chrome isn't your default browser [Set as default](#) ✕

Name:

Email:

Submit

New record created successfully

Login in mysql and check the db

```
mysql> select * from data;
+-----+-----+
| firstname | email |
+-----+-----+
| AWS       | support@aws.com |
| vishwa2   | vish@gmail.com |
| vishwa3   | vish2@gmail.com |
+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

i-0250c1a485858a544 (AWS project 1)  
PublicIPs: 54.86.9.69 PrivateIPs: 172.31.88.92

We can see the data has been stored in the server

### Step 7: create AMI from ec2 instance then use this template to launch ASG

sg-0c75884f49c80aedf - AWSproject1

Amazon Machine Images (AMIs) (1/1) Info [🔗](#) [Recycle Bin](#) [EC2 Image Builder](#) [Actions](#) [Launch instance from AMI](#)

Owned by me 🔽 🔍 Find AMI by attribute or tag

AMI name	AMI ID	Source	Owner	Visibility	Status
awsproject1images	ami-024b341698d4aea69	767398047488/awsproject1images	767398047488	Private	Available 🔍





Launch Templates (1/2) Info

Actions

Create launch template

Q

Search

<

1

>

<input checked="" type="checkbox"/>	Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By
<input type="checkbox"/>	lt-0fbd6cf138a9bfcd7	Temp1	1	1	2024-07-04T14:39:47.000Z	arn:aws:iam::76739
<input checked="" type="checkbox"/>	lt-06eae5196133c1145	awsproject1	1	1	2024-11-29T07:43:12.000Z	arn:aws:iam::76739

## Step8: setup auto scaling group

Auto Scaling groups (1) Info

Launch configurations

Launch templates

Actions

Create Auto Scaling group

Q

Search your Auto Scaling groups

<

1

>

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
<input type="checkbox"/>	<a href="#">AWS_project_asg</a>	<a href="#">awsproject1</a>   Version Default	2	-	2	1	3	us-east-1a, us-east-1b

EC2 > Auto Scaling groups > AWS\_project\_asg

Q

Filter notifications

<

1

>

Send to

On instance action

No notifications are currently specified

Create notification

Activity history (2)

Q

Filter activity history

<

1

>

Status	Description	Cause	Start time	End time
Successful	Launching a new EC2 instance: i-0b7f2f008e3a07205	At 2024-11-29T07:51:53Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2024-11-29T07:51:55Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.	2024 November 29, 01:21:57 PM +05:30	2024 November 29, 01:22 PM +05:30
Successful	Launching a new EC2 instance: i-0b09409653ad12c71	At 2024-11-29T07:51:53Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2024-11-29T07:51:55Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.	2024 November 29, 01:21:57 PM +05:30	2024 November 29, 01:22 PM +05:30

Instances (1/3) Info

Connect

Instance state

Actions

Launch instances

Q

Find Instance by attribute or tag (case-sensitive)

All states

<

1

>

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	AWS project 1	i-0250c1a485858a544	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1b
<input type="checkbox"/>	autoscalinggrp	i-0b7f2f008e3a07205	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1b
<input checked="" type="checkbox"/>	autoscalinggrp	i-0b09409653ad12c71	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1a

## Create target group for load balancer

EC2 > Target groups > Create target group

Step 1

Specify group details

Step 2

Register targets

Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (1/3)

Q

Filter instances

<

1

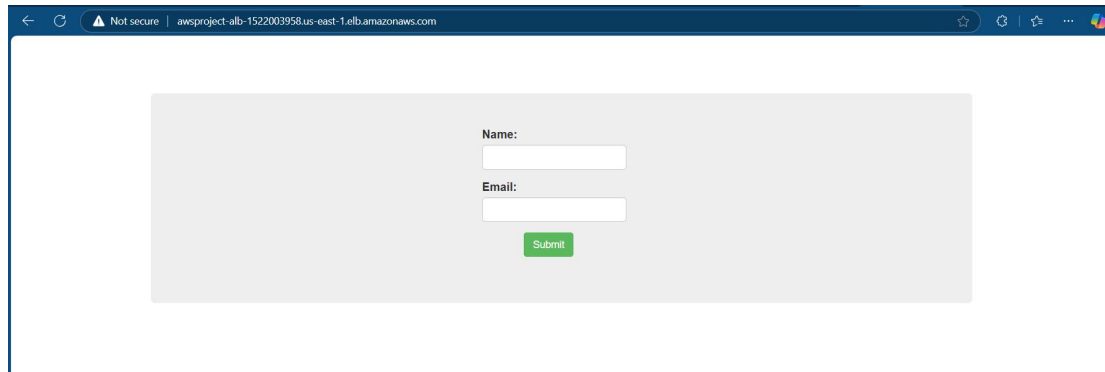
>

<input checked="" type="checkbox"/>	Instance ID	Name	State	Security groups	Zone
<input type="checkbox"/>	i-0b7f2f008e3a07205	autoscalinggrp	Running	AWSproject1	us-east-1b
<input type="checkbox"/>	i-0b09409653ad12c71	autoscalinggrp	Running	AWSproject1	us-east-1a
<input checked="" type="checkbox"/>	i-0250c1a485858a544	AWS project 1	Running	AWSproject1, ec2-rds-1	us-east-1b

**Target groups (1)** [Info](#)

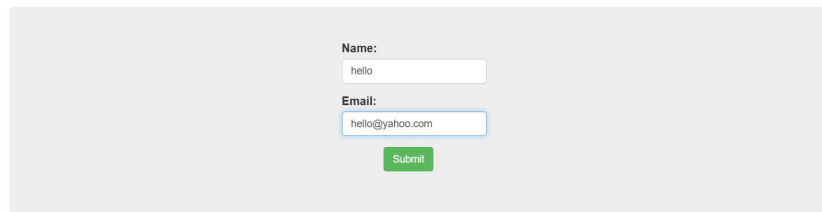
<input type="checkbox"/>	Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
<input type="checkbox"/>	<a href="#">Awsproject1</a>	arn:aws:elasticloadbalancing...	80	HTTP	Instance	None associated	vpc-0240f...

Paste the load balancer in url and check if the website is working or not



A screenshot of a web browser window. The address bar shows "awsproject-alb-1522003958-us-east-1.elb.amazonaws.com". The page content is a light gray rectangle containing a form with two input fields labeled "Name:" and "Email:", and a green "Submit" button below them.

Now we will test if we can add the data from here



A screenshot of the same web form as above, but now with data entered. The "Name:" field contains "hello" and the "Email:" field contains "hello@yahoo.com". The "Submit" button is still present.

```
mysql> select * from data;
+-----+-----+
| firstname | email |
+-----+-----+
| AWS       | support@aws.com |
| vishwa2   | vish@gmail.com |
| vishwa3   | vish2@gmail.com |
| hello     | hello@yahoo.com |
+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

i-0250c1a485858a544 (AWS project 1)  
PublicIPs: 54.86.9.69 PrivateIPs: 172.31.88.92

We can see data has been added to the database

We cant change the domain name as we don't have free domain .