

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

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 EC2 instance

Ans:

1. Open up the AWS Management Console
2. Check for the region [us-west-2(Oregon)]
3. Search for EC2 in the search box
4. Click on instances to go to the EC2 console
5. Click on Launch Instance and setup the instance for Ubuntu OS:
 - a. Name: Project-EC2

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.

Name and tags Info

Name Add additional tags

Summary

Number of instances Info
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.2.2... read more
ami-09ac7e749b0a8d2a1

Virtual server type (instance type)
t2.micro

Firewall (security group)

b. AMI: QuickStart >> Ubuntu OS [Any version which is free tier eligible]

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.

Quick Start

Recent AMIs

Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
ami-0fcce6bed30fd98 (64-bit (x86)) / ami-03fd0aa14bd102718 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Summary

Number of instances Info
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.2.2... read more
ami-09ac7e749b0a8d2a1

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

Cancel **Launch instance** Review commands

c. Instance type: t2. micro [free tier eligible]

d. Key-pair: Create a key pair [rsa and .pem] with a name Project-key

Instance type Info

Instance type
t2.micro
Family: t2 1 vCPU 1 GiB Memory Current generation: true
Free tier eligible
On-Demand Linux base pricing: 0.0116 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand RHEL base pricing: 0.0716 USD per Hour

All generations Compare instance types

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.

Key pair name - required

Summary

Number of instances Info
1

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ... read more
ami-0fcce6bed30fd98

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

Cancel **Launch instance**

- e. Create security group with a name Project-sg and allow all traffic from anywhere click on launch instance.

The screenshot shows the AWS EC2 Launch Instance wizard. In the left panel under 'Network settings', a VPC is selected (vpc-0a6103f2c13973ee4), and a security group named 'Project-sg' is chosen. In the right panel under 'Summary', the instance type is set to 't2.micro', and the 'Launch instance' button is highlighted.

- f. Check it Project-EC2 instance is in running state.

The screenshot shows the AWS EC2 Dashboard. The 'Instances' section lists one instance named 'Project-EC2' with the status 'Running'. The 'Actions' dropdown menu for this instance includes a 'Launch instances' option.

6. Now go to search bar type RDS and click on Create database.

The screenshot shows the AWS Amazon RDS Dashboard. A modal dialog is open, prompting to 'Try the new Amazon RDS Multi-AZ deployment option for MySQL and PostgreSQL'. It also mentions 'Introducing Aurora I/O-Optimized' and provides links to learn more and create a database. The 'Create database' button is visible at the bottom of the dialog.

a. Choose a Standard create

The screenshot shows the 'Create database' page in the AWS RDS console. In the top left, there's a breadcrumb navigation: RDS > Create database. Below it, a section titled 'Choose a database creation method' contains two options: 'Standard create' (selected) and 'Easy create'. The 'Standard create' box includes a note: 'You set all of the configuration options, including ones for availability, security, backups, and maintenance.'

b. Select engine type as MySQL database

The screenshot shows the 'Engine options' page in the AWS RDS console. On the left, there's a grid of engine icons. The 'MySQL' icon is selected and highlighted with a green border. To its right, a modal window titled 'MySQL' provides a brief description: '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.' Below this are several bullet points: '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.', and 'Supports up to 15 Read replicas per instance, within the same region.'

c. With a latest engine version as MySQL 8.0.33

The screenshot shows the 'Edition' selection page in the AWS RDS console. Under the 'Edition' heading, 'MySQL Community' is selected. Below it, a box titled 'Known issues/limitations' says: 'Review the [Known issues/limitations](#) to learn about potential compatibility issues with specific database versions.' Further down, there are filter options: 'Show versions that support the Multi-AZ DB cluster' (selected) and 'Show versions that support the Amazon RDS Optimized Writes'. A dropdown menu for 'Engine Version' has 'MySQL 8.0.33' selected. On the right, a modal window titled 'MySQL' reiterates the MySQL features and supports the same list of bullet points as the previous screenshot.

d. Template as Free tier.

The screenshot shows the AWS RDS 'Templates' page. On the left, there are three options: 'Production', 'Dev/Test', and 'Free tier'. The 'Free tier' option is selected and highlighted with a blue border. To its right, there is a detailed description of the Free tier: 'Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.' Below this, there is a section titled 'Availability and durability' which lists deployment options: 'Multi-AZ DB Cluster - new', 'Multi-AZ DB instance (not supported for Multi-AZ DB cluster snapshot)', and 'Single DB instance (not supported for Multi-AZ DB cluster snapshot)'. On the right side of the page, there is a sidebar titled 'MySQL' with information about MySQL and a bulleted list of features.

e. In setting DB instance identifier as Project-DB

The screenshot shows the AWS RDS 'Settings' page. Under the 'DB instance identifier' section, the identifier 'Project-DB' is entered into the input field. A note below the field states: '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.' On the right side, there is a sidebar titled 'MySQL' with information about MySQL and a bulleted list of features.

f. With master user as admin and master password as admin123 and confirm it.

The screenshot shows the 'Create DB Instance' step of the AWS RDS setup wizard. Under the 'Master user' section, the login ID 'admin' is entered into the input field. A note below the field states: '1 to 16 alphanumeric characters. The first character must be a letter.' Under the 'Master password' section, the password 'admin123' is entered into both the 'Master password' and 'Confirm master password' input fields. A note below the 'Master password' field states: 'Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).' On the right side, there is a sidebar titled 'MySQL' with information about MySQL and a bulleted list of features.

g. In instance configuration selects burstable classes with dbt3.micro.

The screenshot shows the 'Instance configuration' section for a MySQL database. Under 'DB instance class', the 'Burstable classes (includes t classes)' option is selected. A dropdown menu shows 'db.t3.micro' is chosen, with details: 2 vCPUs, 1 GiB RAM, and Network: 2,085 Mbps. A note says 'Show instance classes that support Amazon RDS Optimized Writes'. On the right, a sidebar for MySQL lists its features, including support for up to 64 TiB, General Purpose, Memory Optimized, and Burstable Performance instance classes, automated backup, point-in-time recovery, and up to 15 Read Replicas per instance.

h. In Storage selects general purpose SSD (gp2) with allocated storage of 20GiB and autoscaling storage is disable.

The screenshot shows the 'Storage' configuration for a MySQL database. The 'Storage type' is set to 'General Purpose SSD (gp2)'. The 'Allocated storage' is set to '20 GiB'. A note states: '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.' Under 'Connectivity', 'Storage autoscaling' is disabled, with a note: 'Enabling this feature will allow the storage to increase after the specified threshold is exceeded.' On the right, a sidebar for MySQL lists its features, including support for up to 64 TiB, General Purpose, Memory Optimized, and Burstable Performance instance classes, automated backup, point-in-time recovery, and up to 15 Read Replicas per instance.

i. In connectivity chooses connect to an EC2 compute resource and choose Project-EC2 id for EC2 instance.

The screenshot shows the AWS RDS MySQL configuration page. Under the 'Compute resource' section, the 'Connect to an EC2 compute resource' option is selected. Below it, under 'EC2 instance info', an EC2 instance named 'Project-EC2' is chosen from a dropdown menu. On the right side, a sidebar provides information about MySQL, including its popularity and supported features like automated backup.

j. With the default VPC and for DB subnet group as automatic setup.

The screenshot shows the AWS RDS MySQL configuration page. A note states that some VPC settings can't be changed after a compute resource is added. Under 'Virtual private cloud (VPC) info', the 'Default VPC' is selected. Under 'DB subnet group info', the 'Automatic setup' option is selected. On the right side, a sidebar provides information about MySQL, including its popularity and supported features like automated backup.

k. No public access to the database.

The screenshot shows the AWS RDS MySQL configuration page. Under 'Public access info', the 'No' option is selected, indicating no public IP address will be assigned. Under 'VPC security group (firewall) info', the 'Choose existing' option is selected. On the right side, a sidebar provides information about MySQL, including its popularity and supported features like automated backup.

1. Choose existing group as Project-sg VPC security group.

The screenshot shows the AWS RDS MySQL configuration interface. In the top left, there's a sidebar with 'Services' and a search bar. The main area has a title 'MySQL'. On the left, under 'Additional VPC security group', there are two options: 'Choose existing' (selected) and 'Create new'. Below this is a dropdown menu set to 'Project-sg'. A note below the dropdown states: 'Amazon RDS will add a new VPC security group rds-ec2-3 to allow connectivity with your compute resource.' To the right, there's a detailed description of MySQL and a bulleted list of its features. At the bottom, there are links for CloudShell, Feedback, and various AWS services, along with system status information like temperature and battery level.

m. With default setting

This screenshot shows the same MySQL configuration page but with different settings. Under 'Certificate authority - optional', it shows 'rds-ca-2019 (default)' selected. Under 'Database authentication', 'Password authentication' is selected. The right panel contains the same MySQL information and feature list as the previous screenshot.

n. In additional configuration gives initial database name as intel with default mysql8.0 for DB parameter group ang option group.

This screenshot shows the 'Additional configuration' section expanded. It includes fields for 'Initial database name' (set to 'intel'), 'DB parameter group' (set to 'default.mysql8.0'), and 'Option group' (set to 'default:mysql-8-0'). The right panel continues to display MySQL information and features.

o. The backup is disable and also disable encryption. (Uncheck)

The screenshot shows the AWS RDS MySQL configuration page. Under the 'Backup' section, the 'Enable automated backups' checkbox is unchecked. Under the 'Encryption' section, the 'Enable encryption' checkbox is also unchecked. A note below the encryption checkbox states: '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.' To the right of the configuration area, there is a sidebar titled 'MySQL' with a brief description of the database and a bulleted list of features.

p. In the maintenance unchecks enable auto minor version upgrade.

The screenshot shows the AWS RDS MySQL configuration page. Under the 'Maintenance' section, the 'Enable auto minor version upgrade' checkbox is unchecked. Below it, under 'Maintenance window', the 'No preference' radio button is selected. A note states: 'Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.' To the right of the configuration area, there is a sidebar titled 'MySQL' with a brief description of the database and a bulleted list of features.

q. It is under free tier and click on create database.

The screenshot shows the AWS RDS MySQL configuration page. On the left, there is a section titled 'Estimated monthly costs' which details the availability of the free tier for 12 months and lists usage limits. A note at the bottom of this section states: 'Learn more about AWS Free Tier.' Below this, a callout box contains the text: '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.' At the bottom of the page, there are 'Cancel' and 'Create database' buttons. To the right of the configuration area, there is a sidebar titled 'MySQL' with a brief description of the database and a bulleted list of features.

r. Successfully project-db database is created.

The screenshot shows the Amazon RDS console. On the left, there's a sidebar with options like Dashboard, Databases, Query Editor, etc. The main area has a green header bar stating "Successfully created database project-db". Below it, a message says you can use settings from the database to simplify configuration of suggested database add-ons. A tooltip suggests creating a Blue/Green deployment. The main table lists one database entry:

DB identifier	Status	Role	Engine	Region & AZ	Size	Actions	CPU
project-db	Available	Instance	MySQL Community	us-west-2b	db.t3.micro	-	-

s. Now go to EC2 console and connect the Project-EC2 instance

The screenshot shows the EC2 Instance Connect interface. It displays connection details for instance i-0d0a41ae4c61bc2b8. The "Connect using EC2 Instance Connect" option is selected. Other fields include Public IP address (54.245.2.165) and User name (ubuntu). There's also a note about the user name defined in the AMI.

t. Update first \$ sudo apt-get update.

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

ubuntu@ip-172-31-30-232:~$ sudo apt-get update
Hit:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]

i-0d0a41ae4c61bc2b8 (Project-EC2)
Public IPs: 54.245.2.165 Private IPs: 172.31.30.232
```

u. Sudo apt-get install apache2 -y

```
Get:42 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Fetched 27.9 MB in 5s (5447 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-30-232:~$ sudo apt-get install apache2 -y
```

v. Copy and paste the public IP of Project-EC2 instance in the new browser to check it is installed.



w. To change the file \$ cd /var/www/html, sudo rm index.html and sudo nano index.php

```
buntu@ip-172-31-30-232:~$ cd /var/www/html
buntu@ip-172-31-30-232:/var/www/html$ ls
index.html
buntu@ip-172-31-30-232:/var/www/html$ sudo rm index.html
buntu@ip-172-31-30-232:/var/www/html$ sudo nano index.php
```

i-0d0a41ae4c61bc2b8 (Project-EC2)

Public IPs: 54.245.2.165 Private IPs: 172.31.30.232

x. Copy and paste the source code then ctrl + s for save and ctrl + x for exit.

```
GNU nano 6.2                                     index.php *
<html>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<body background="images/2.png" style="background-repeat:no-repeat;
background-size: 100% 100%>
<br><br><br><br>
<div class="container">
  <div class="jumbotron vertical-center">
    <table class="grid" cellspacing="0">
      <tr>
        <td colspan="4" style="text-align: center; padding: 10px; border: 1px solid black; border-radius: 5px;">
          <form method="post">
            <div class="form-group" style="width: 25%; margin-right: 20px;">
              <label for="firstname">Name:</label>
              <input type="text" class="form-control" name="firstname" style="width: 100%; height: 30px;">
            </div>
            <div class="form-group" style="width: 25%; margin-right: 20px;">
              <label for="email">Email:</label>
              <input type="text" class="form-control" name="email" style="width: 100%; height: 30px;">
            </div>
          </form>
        </td>
      </tr>
    </table>
  </div>
</div>
```

```
GNU nano 6.2                                     index.php *
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
if(isset($_POST['firstname']) && isset($_POST['email'])){
$sql = "INSERT INTO data (firstname,email)
VALUES ('".$_firstname."', '".$_email."')";

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

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

```
i-0d0a41ae4c61bc2b8 (Project-EC2)
Public IPs: 54.245.2.165 Private IPs: 172.31.30.232
```

y. Refresh the browser and see at the bottom some error is occurred.

The screenshot shows a web browser window with a simple form. The form has two input fields: 'Name:' and 'Email:', both with placeholder text. Below the fields is a green 'Submit' button. At the bottom of the page, there is a block of PHP code:

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

z. Now edit the source code and give RDS endpoint in source code.

The screenshot shows the AWS Amazon RDS console. On the left, there is a sidebar with various options like Dashboard, Databases, Query Editor, etc. The main area shows a 'Summary' tab for a database named 'project-db'. The summary includes details such as DB identifier, CPU usage (2.46%), Status (Available), Class (db.t3.micro), Role, Instance, Current activity (0 Connections), Engine (MySQL Community), Region & AZ (us-west-2b), and a VPC section. Below the summary is a 'Connectivity & security' tab, which displays the Endpoint (project-db.c5ufjgm1epfh.us-west-2.rds.amazonaws.com), Networking (Availability Zone: us-west-2b, VPC), and Security (VPC security groups: rds-ec2-3 (sg-0e5b911362c9a5366), Active).

8. a copy and paste end point then ctrl + s for save and ctrl + x for exit.

The screenshot shows a terminal window with the following PHP code:

```
<?php
$firstname=$_POST['firstname'];
$email=$_POST['email'];
$servername = "project-db.c5ufjgm1epfh.us-west-2.rds.amazonaws.com";
$username = "admin";
$password = "admin123";
$db = "intel";
// create connection
$conn = new mysqli($servername, $username, $password, $db);

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

At the bottom of the terminal, there is a menu bar with various keyboard shortcuts for file operations like Help, Exit, Write Out, Read File, Cut, Paste, Execute, Location, Undo, Set Mark, To Bracket, etc. The status bar at the bottom shows the file path (i-0d0a41ae4c61bc2b8 (Project-EC2)) and IP addresses (PublicIPs: 54.245.2.165 PrivateIPs: 172.31.30.232).

b. Now install mysql by command line \$ sudo apt

```
ubuntu@ip-172-31-30-232:/var/www/html$ sudo nano index.php
ubuntu@ip-172-31-30-232:/var/www/html$ sudo nano index.php
ubuntu@ip-172-31-30-232:/var/www/html$ cd
ubuntu@ip-172-31-30-232:~$ sudo apt-get install mysql-server -y

i-0d0a41ae4c61bc2b8 (Project-EC2)
PublicIPs: 54.245.2.165 PrivateIPs: 172.31.30.232
```

c. sudo mysql -h endpoint-of-rds -u admin -padmin123

```
No VM guests are running. Detached hypervisor (qemu) bindings on this host.
ubuntu@ip-172-31-30-232:~$ sudo mysql -h project-db.c5ufjgmlepfh.us-west-2.rds.amazonaws.com -u admin -padmin123

i-0d0a41ae4c61bc2b8 (Project-EC2)
PublicIPs: 54.245.2.165 PrivateIPs: 172.31.30.232
```

d. Now use command show databases; see our database intel is present.

```
mysql> show databases;
+--------------------+
| Database          |
+----+-----+
| information_schema |
| intel              |
| mysql              |
| performance_schema |
| sys                |
+----+-----+
5 rows in set (0.00 sec)

mysql>
```

e. Type use intel;

```
mysql> use intel;
Database changed
mysql> create table data(firstname varchar(20),email varchar(20));
```

f. Type create table data (firstname varchar(20),email varchar(20)); and show tables;

```
mysql> use intel;
Database changed
mysql> create table data(firstname varchar(20),email varchar(20));
Query OK, 0 rows affected (0.03 sec)

mysql> select * from data;
Empty set (0.01 sec)

mysql>
```

g. Type insert into data values ('shiva','shiva@outlook.com'); and select * from data;

```
mysql> insert into data values ('shiva','shiva@outlook.com');

i-0d0a41ae4c61bc2b8 (Project-EC2)
PublicIPs: 54.245.2.165 PrivateIPs: 172.31.30.232

mysql> insert into data values ('shiva','shiva@outlook.com');
Query OK, 1 row affected (0.00 sec)

mysql> select * from data;
+-----+-----+
| firstname | email      |
+-----+-----+
| shiva    | shiva@outlook.com |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

h. Exit from the database. So type exit

```
1 row in set (0.00 sec)

mysql> exit
Bye
ubuntu@ip-172-31-30-232:~$
```

i. To remove errors type sudo add-apt-repository -y ppa:ondrej/php

```
mysql> exit
Bye
ubuntu@ip-172-31-30-232:~$ sudo add-apt-repository -y ppa:ondrej/php
i-0d0a41ae4c61bc2b8 (Project-EC2)
Public IPs: 54.245.2.165 Private IPs: 172.31.30.232
```

j. Also some dependencies sudo apt install php5.6 mysql-client php5.6-mysqli

```
Get:0 https://ppa.launchpadcontent.net/ondrej/php/ubuntu jammy/main amd64 Packages [113 kB]
Get:1 https://ppa.launchpadcontent.net/ondrej/php/ubuntu jammy/main Translation-en [35.6 kB]
Fetched 512 kB in 3s (168 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-30-232:~$ sudo apt install php5.6 mysql-client php5.6-mysqli
i-0d0a41ae4c61bc2b8 (Project-EC2)
Public IPs: 54.245.2.165 Private IPs: 172.31.30.232
```

k. Refresh again then this no errors occurs at the bottom.

A screenshot of a web browser window. At the top, there is a navigation bar with various links like Google, Gmail, YouTube, etc. Below the navigation bar is a form with two input fields: 'Name:' and 'Email:', each with a corresponding text input box. Below the input boxes is a green 'Submit' button. The background of the page is light gray.

l. Now add same info.

A screenshot of a web browser window, similar to the previous one. It shows a form with 'Name:' and 'Email:' fields, both containing the value 'sitamata'. Below the fields is a green 'Submit' button. The background is light gray.

New record created successfully

m. Once again connect to the database so type sudo mysql -h projectdb.c5ufjgm1epfh.us-west-2.rds.amazonaws.com -u admin -padmin123, use intel; and select * from data;

```

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

Database changed
mysql> select * from data;
+-----+-----+
| firstname | email      |
+-----+-----+
| shiva     | shiva@outlook.com |
| ram       | ramhari@yahoo.com |
| ramhari   | ramhari@gmail.com |
| sitamata  | sitamata@gmail.com |
| sitamata  | sitamata@gmail.com |
+-----+-----+
5 rows in set (0.00 sec)

mysql> 
```

9. For scalable solution create auto-scaling

a. Create image of the Project-EC2 instance. So give to action click

The screenshot shows the AWS EC2 Instances page. A single instance named "Project-EC2" is listed, showing it's running and of type t2.micro. The "Actions" menu is open, and the "Create image" option is highlighted with a blue border. Other options in the menu include Connect, View details, Manage instance state, Instance settings, Networking, Security, Image and templates, and Monitor and troubleshoot.

b. Create with a name Project-image and click on create image

The screenshot shows the "Create image" wizard. It has several input fields: "Instance ID" (i-06f4716732c844c94 (Project-EC2)), "Image name" (Project-image), "Image description - optional" (Project-image), and a "No reboot" checkbox. At the bottom, there are tabs for "Tags" and "Create image" buttons. The "Create image" button is highlighted with a blue border.

c. On left side under Image click on AMIs and Project-image is created.

The screenshot shows two overlapping AWS EC2 management console windows. The top window is titled 'Instances (1) Info' and displays a single instance named 'Project-EC2' with the ID 'i-06f4716732c844c94'. A green status bar at the top indicates: 'Currently creating AMI ami-Ocea915eca09d477b from instance i-06f4716732c844c94. Check that the AMI status is 'Available' before deleting the instance or carrying out other actions related to this AMI.' The bottom window is titled 'Amazon Machine Images (AMIs) (1/1) Info' and shows a single AMI named 'Project-image' with the ID 'ami-Ocea915eca09d477b'. The details pane below shows the AMI ID, Source (297081084983/Project-image), and Owner (2970810849).

d. On left side under Instance click on Launch Templates

The screenshot shows the 'Compute' section of the AWS EC2 management console. The left sidebar highlights 'Launch Templates'. The main area features a large title 'EC2 launch templates' with the subtitle 'Streamline, simplify and standardize instance launches'. Below this is a descriptive text about using launch templates for automation and best practices. A callout box labeled 'New launch template' contains a 'Create launch template' button. At the bottom, there's a 'Benefits and features' section.

e. Create with a name Project-lt

The screenshot shows the 'Create launch template' wizard. The left panel is titled 'Launch template name and description' and includes fields for 'Launch template name - required' (containing 'Project-lt') and 'Template version description' (containing 'Project-lt'). It also has sections for 'Auto Scaling guidance' and checkboxes for 'Provide guidance to help me set up a template that I can use with EC2 Auto Scaling'. The right panel is titled 'Summary' and lists configuration items: 'Software Image (AMI)', 'Virtual server type (instance type)', 'Firewall (security group)', 'Storage (volumes)', and a note about the 'Free tier'. At the bottom are 'Cancel' and 'Create launch template' buttons.

f. In AMI choose My AMIs owned by me that is Project-image.

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 | **My AMIs** | Quick Start

Don't include in launch template Owned by me Shared with me

Amazon Machine Image (AMI)

Project-image
ami-0cea915eca09d477b
2023-10-23T08:59:51.000Z Virtualization: hvm ENA enabled: true Root device type: ebs

Summary

Software Image (AMI)
Project-image
ami-0cea915eca09d477b

Virtual server type (instance type)
t2.micro

Firewall (security group)
Project-sg

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

Free tier: In your first year includes 750 hours of t2.micro (or

Create launch template

g. Choose t2.micro under instance type and select Project-key as keypair.

Instance type

t2.micro Family: t2 1 vCPU 1 GiB Memory Current generation: true Free tier eligible

On-Demand Linux base pricing: 0.0116 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand RHEL base pricing: 0.0716 USD per Hour

All generations Compare instance types

Additional costs apply for AMIs with pre-installed software

Key pair (login)

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

Key pair name Create new key pair

Summary

Software Image (AMI)
Project-image
ami-0cea915eca09d477b

Virtual server type (instance type)
t2.micro

Firewall (security group)
Project-sg

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

Free tier: In your first year includes 750 hours of t2.micro (or

Create launch template

h. In the network settings select Project-sg as security group and go down click on create launch template

Network settings

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)

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.

Select existing security group Create security group

Security groups Info

Select security groups

Project-sg sg-02ebd86827da4d61b X
VPC: vpc-0a6103f2c13973ee4

Compare security group rules

Advanced network configuration

Summary

Software Image (AMI)
Project-image
ami-0cea915eca09d477b

Virtual server type (instance type)
t2.micro

Firewall (security group)
Project-sg

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

Free tier: In your first year includes 750 hours of t2.micro (or

Create launch template

i. Project-lt is created.

The screenshot shows the AWS RDS console with the 'Launch Templates' section selected. A single launch template is listed:

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By
lt-06ea327603dea3777	Project-lt	1	1	2023-10-23T09:18:11.000Z	arn:aws:iam::2970...

A modal window titled 'Project-lt (lt-06ea327603dea3777)' is open, displaying the details of the launch template.

j. On left side at the end click on Auto Scaling Groups.

The screenshot shows the AWS EC2 Auto Scaling Groups page. The sidebar includes options like Volumes, Snapshots, Lifecycle Manager, Network & Security, Load Balancing, and Auto Scaling. The main content area features a large heading 'Amazon EC2 Auto Scaling helps maintain the availability of your applications' and a 'Create Auto Scaling group' button.

k. Create auto scaling group with a name Project-ASG

The screenshot shows the 'Choose launch template' step of the 'Create Auto Scaling group' wizard. It lists steps 1 through 7. Step 1 is 'Choose launch template'. The main panel shows a 'Name' input field where 'Project-ASG' has been typed. A note states: 'Auto Scaling group name. Enter a name to identify the group.' Below it, a note says: 'Must be unique to this account in the current Region and no more than 255 characters.'

l. Choose Project-lt as launch template and click on next.

The screenshot shows the 'Launch template' step of the wizard. It lists steps 5 through 7. Step 5 is 'Add notifications', Step 6 is 'Add tags', and Step 7 is 'Review'. The main panel shows a search bar with 'Project-lt' selected. A note at the top of the panel states: 'For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.' At the bottom are 'Cancel' and 'Next' buttons.

m. For Maximum availability selects two to three subnets in the network and click on next

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
Project-It Edit	Default	Project-It

Launch template
Project-It [Edit](#)
lt-06ea327603dea3777

Version
Default

Description
Project-It

Instance type
t2.micro

Add tags

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.

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-0a6103f2c13973ee4
172.31.0.0/16 Default [Edit](#)

Create a VPC [Edit](#)

Availability Zones and subnets

Select Availability Zones and subnets [Edit](#)

us-west-2a | subnet-0219a97adec4d42ae X
172.31.32.0/20 Default

us-west-2b | subnet-012f23ee1f5a18177 X
172.31.16.0/20 Default

us-west-2c | subnet-0c964a4d6f4c2d82a X
172.31.0.0/20 Default

Step 7
Review

n. With a default setting (optional) goes down click on next

Configure advanced options - *optional* [Info](#)

Integrate your Auto Scaling group with other services to distribute network traffic across multiple servers using a load balancer or to establish service-to-service communications using VPC Lattice. You can also set options that give you more control over health check replacements and monitoring.

Load balancing [Info](#)

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.

VPC Lattice integration options [Info](#)

Step 1
Choose launch template

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

Step 6 - *optional*
Add tags

o. In group size chooses Maximum: 2, Minimum: 1 and Desired: 1 capacity. Go down click on next

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: 1

Minimum capacity: 1

Maximum capacity: 2

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

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

Review (1)

Cancel Skip to review Previous Next

p. Just click on next

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

Cancel Skip to review Previous Next

q. Under tags give key as Name and Value as ASG click on next

The screenshot shows the 'Tags (1)' configuration screen. On the left, a sidebar lists steps from 'Step 2' to 'Step 7'. Step 2 is 'Choose instance launch options', Step 3 is 'Configure advanced options', Step 4 is 'Configure group size and scaling policies', Step 5 is 'Add notifications', Step 6 is 'Add tags', and Step 7 is 'Review'. Step 6 is currently active. The main area shows a table with one tag: 'Key' is 'Name' and 'Value - optional' is 'ASG'. A checkbox 'Tag new instances' is checked. Below the table is a button 'Add tag' and a note '49 remaining'. At the bottom are 'Cancel', 'Previous', and 'Next' buttons, with 'Next' being orange.

r. Check once all in the review section go down and click on create auto scaling group

The screenshot shows the 'Review' step of the wizard. The sidebar shows steps 1 through 7. Step 1 is 'Choose launch template', Step 2 is 'Choose instance launch options', Step 3 is 'Configure advanced options', Step 4 is 'Configure group size and scaling policies', Step 5 is 'Add notifications', and Step 6 is 'Add tags'. Step 1 is active. The main area shows 'Step 1: Choose launch template' with a 'Group details' section containing an 'Auto Scaling group name' of 'Project-ASG' and a 'Launch template' of 'Project-It'. It also shows 'Step 2: Choose instance launch options'. At the bottom are 'Cancel', 'Previous', and 'Create Auto Scaling group' buttons, with 'Create Auto Scaling group' being orange.

The screenshot shows the final confirmation step. The sidebar shows steps 1 through 7. Step 1 is 'Choose launch template', Step 2 is 'Choose instance launch options', Step 3 is 'Configure advanced options', Step 4 is 'Configure group size and scaling policies', Step 5 is 'Add notifications', and Step 6 is 'Add tags'. Step 6 is active. The main area shows 'Step 5: Add notifications' with a 'Notifications' section showing 'No notifications'. It also shows 'Step 6: Add tags' with a 'Tags (1)' section showing a single tag 'Name' with value 'ASG' and 'Tag new instances' checked. At the bottom are 'Cancel', 'Previous', and 'Create Auto Scaling group' buttons, with 'Create Auto Scaling group' being orange.

s. Project-ASG is created.

The screenshot shows the AWS Auto Scaling Groups console. At the top, there is a search bar and a table with columns: Name, Launch..., Instances, Status, Desired capacity, Min, Max, and Available... The table contains one row for 'Project-ASG'. Below this, a modal window titled 'Auto Scaling group: Project-ASG' is open, showing tabs for Details, Activity, Automatic scaling, Instance management, Monitoring, and Instance refresh. The 'Details' tab is selected, and a 'Group details' section is visible with an 'Edit' button.

t. Go to EC2 console and check auto scaling group launch an instance or not

The screenshot shows the AWS EC2 Instances console. On the left sidebar, under 'Instances', there are sections for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, and Capacity Reservations. The main area displays a table of instances with columns: Name, Instance ID, Instance state, Instance type, Status check, and Alarm status. Two instances are listed: 'ASG' (i-02920bde65adca10c) and 'Project-EC2' (i-06f4716732c844c94), both in a 'Running' state. Below the table, a modal window titled 'Instance: i-02920bde65adca10c (ASG)' is open, showing tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. The 'Details' tab is selected, displaying information such as Public IPv4 address (34.219.169.42), Private IPv4 addresses (172.31.16.96), and Public IPv4 DNS (ec2-34-219-169-42.us-west-2).

u. Copy and paste public ip of ASG instance and check for the application running or not

The screenshot shows a web browser window with the URL '34.219.169.42'. The page content is a simple form with fields for 'Name:' and 'Email:', and a 'Submit' button. The browser's address bar also shows 'Instances | EC2 | us-west-2' and 'RDS | us-west-2'.

v. Add some info

New record created successfully

w. Connect the ASG instance

EC2 > Instances > i-02920bde65adca10c > Connect to instance

Connect to instance [Info](#)

Connect to your instance i-02920bde65adca10c (ASG) using any of these options

[EC2 Instance Connect](#) [Session Manager](#) [SSH client](#) [EC2 serial console](#)

Instance ID
i-02920bde65adca10c (ASG)

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
34.219.169.42

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, root.

x. Update it by sudo apt-get update

```
root@ip-172-31-16-96:~# sudo apt-get update
```

i-02920bde65adca10c (ASG)

Public IPs: 34.219.169.42 Private IPs: 172.31.16.96

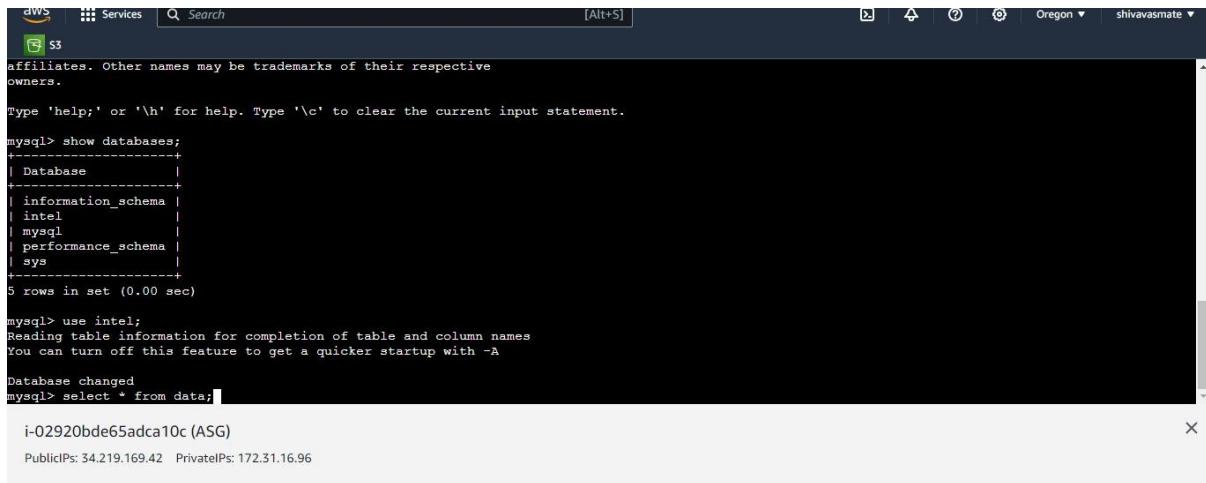
y. Install mysql by sudo apt-get install mysql-server -y

```
Fetched 338 kB in 1s (242 kB/s)
Reading package lists... Done
root@ip-172-31-16-96:~# sudo apt-get install mysql-server -y
```

i-02920bde65adca10c (ASG)

Public IPs: 34.219.169.42 Private IPs: 172.31.16.96

z. Connect to RDS instance by sudo mysql -h projectdb.c5ufjgm1epfh.us-west-2.rds.amazonaws.com -u admin -padmin123, show databases; and use intel;



A screenshot of a terminal window titled "aws" with the tab "Services". The title bar includes "Search [Alt+S]" and "Oregon". The user "shivavasmate" is logged in. The terminal shows a MySQL session:

```
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> show databases;
+-----+
| Database |
+-----+
| information_schema |
| intel |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

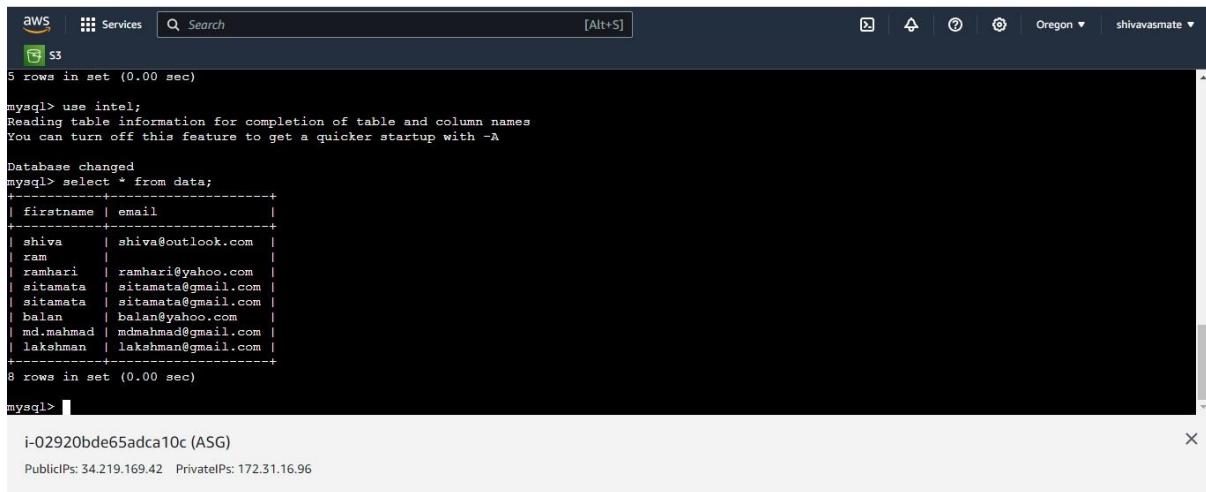
mysql> use intel;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from data;
+-----+
| firstname | email           |
+-----+
| shiva    | shiva@outlook.com |
| ram      | ramhari@yahoo.com  |
| ramhari  | ramhari@yahoo.com  |
| sitamata | sitamata@gmail.com |
| sitamata  | sitamata@gmail.com |
| balan    | balan@yahoo.com   |
| md.mahmad | mdmahmad@gmail.com |
| lakshman | lakshman@gmail.com |
+-----+
8 rows in set (0.00 sec)

mysql>
```

The terminal also displays the instance ID "i-02920bde65adca10c (ASG)" and IP addresses "PublicIPs: 34.219.169.42 PrivateIPs: 172.31.16.96".

10. To check the add info by select * from data;



A screenshot of a terminal window titled "aws" with the tab "Services". The title bar includes "Search [Alt+S]" and "Oregon". The user "shivavasmate" is logged in. The terminal shows a MySQL session:

```
5 rows in set (0.00 sec)

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

Database changed
mysql> select * from data;
+-----+
| firstname | email           |
+-----+
| shiva    | shiva@outlook.com |
| ram      | ramhari@yahoo.com  |
| ramhari  | ramhari@yahoo.com  |
| sitamata | sitamata@gmail.com |
| sitamata  | sitamata@gmail.com |
| balan    | balan@yahoo.com   |
| md.mahmad | mdmahmad@gmail.com |
| lakshman | lakshman@gmail.com |
+-----+
8 rows in set (0.00 sec)

mysql>
```

The terminal also displays the instance ID "i-02920bde65adca10c (ASG)" and IP addresses "PublicIPs: 34.219.169.42 PrivateIPs: 172.31.16.96".