

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

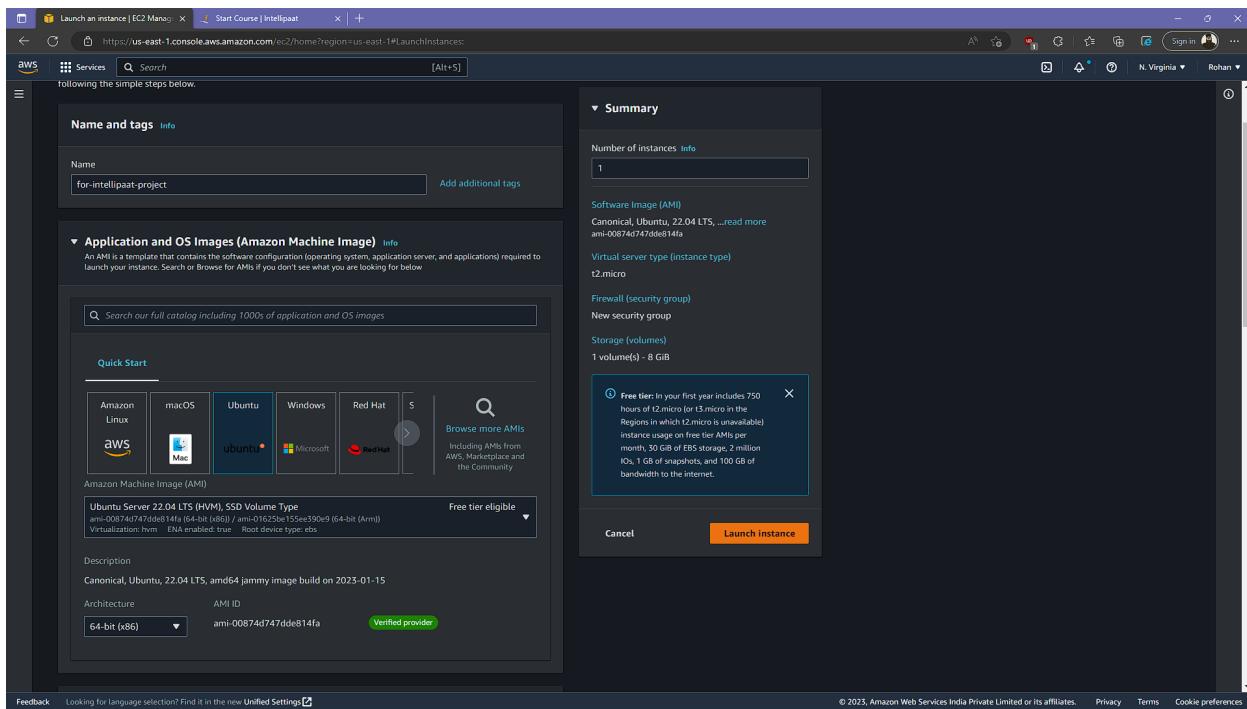
Topic: Deploy a Multi-tier website using EC2

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 setup right now:

1. MySQL DB
2. Website (PHP) The company wants high availability on this product, therefore wants autoscaling to be enabled on this website.

Lets's create a EC2 instance first with ubuntu AMI.



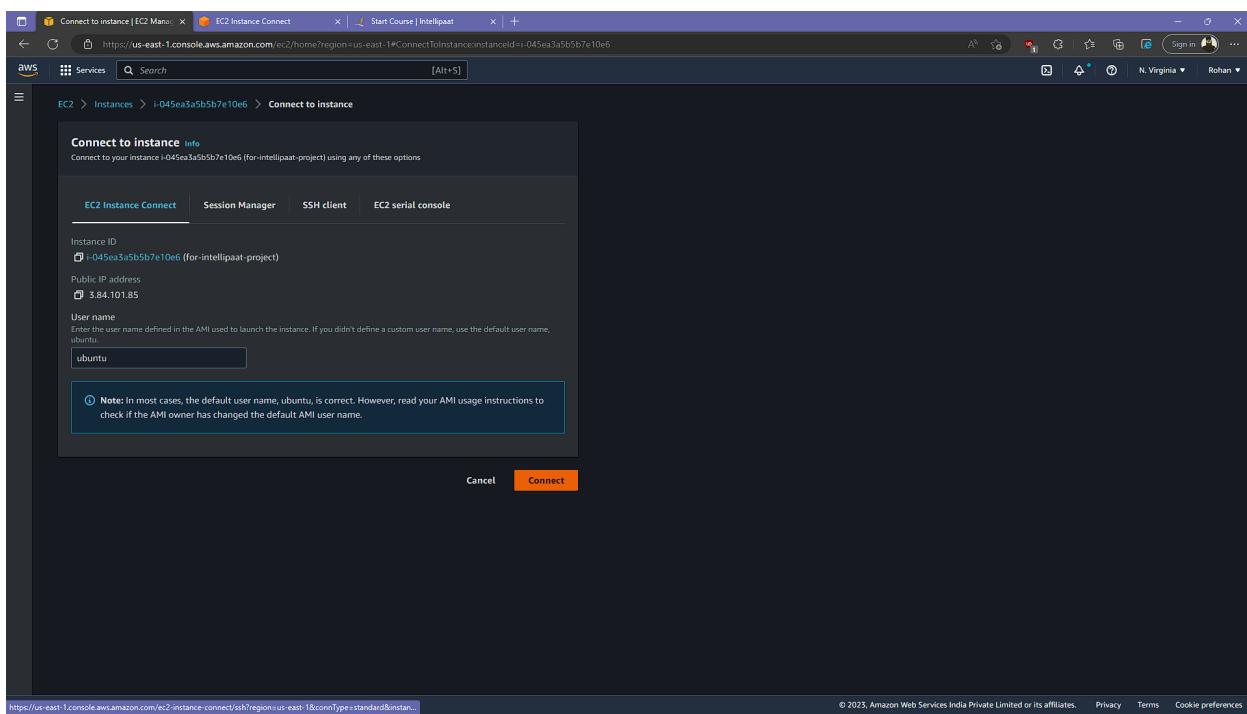
Create a keypair and assign to it. Allow only ssh for now from anywhere.

The screenshot shows the AWS EC2 Launch Instance wizard. In the 'Network settings' section, a new security group named 'launch-wizard-10' is being created. Under 'Allow SSH traffic from', the 'Anywhere' checkbox is selected. A tooltip indicates that this allows connections from all IP addresses. Other options like 'Allow HTTPS traffic from the internet' and 'Allow HTTP traffic from the internet' are also present but not selected. In the 'Summary' section, it shows 1 instance, the Canonical Ubuntu 22.04 LTS AMI, t2.micro instance type, and 1 volume (8 GiB). A modal window displays the free tier information: 750 hours of t2.micro or t3.micro usage in the region, 50 GB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth. At the bottom right, there are 'Cancel' and 'Launch instance' buttons.

Instance has been created.

The screenshot shows the AWS EC2 Instances page. It lists a single instance named 'for-intellipaat-project' with the ID 'i-045ea3a5b5b7e10e6'. The instance is currently running. The details panel for this instance shows its configuration: Instance ID (i-045ea3a5b5b7e10e6), Public IPv4 address (3.84.101.85), Private IP (172.31.92.170), Instance type (t2.micro), VPC ID (vpc-0c0b71ad96deeaa470), and Subnet ID (-). The status is 'Running' and the alarm status is 'No alarms'. The public IPv4 DNS is ec2-3-84-101-85.compute-1.amazonaws.com. The page also includes tabs for Security, Networking, Storage, Status checks, Monitoring, and Tags.

Let's SSH to it.



Use following commands once connecting to instance:

Sudo su

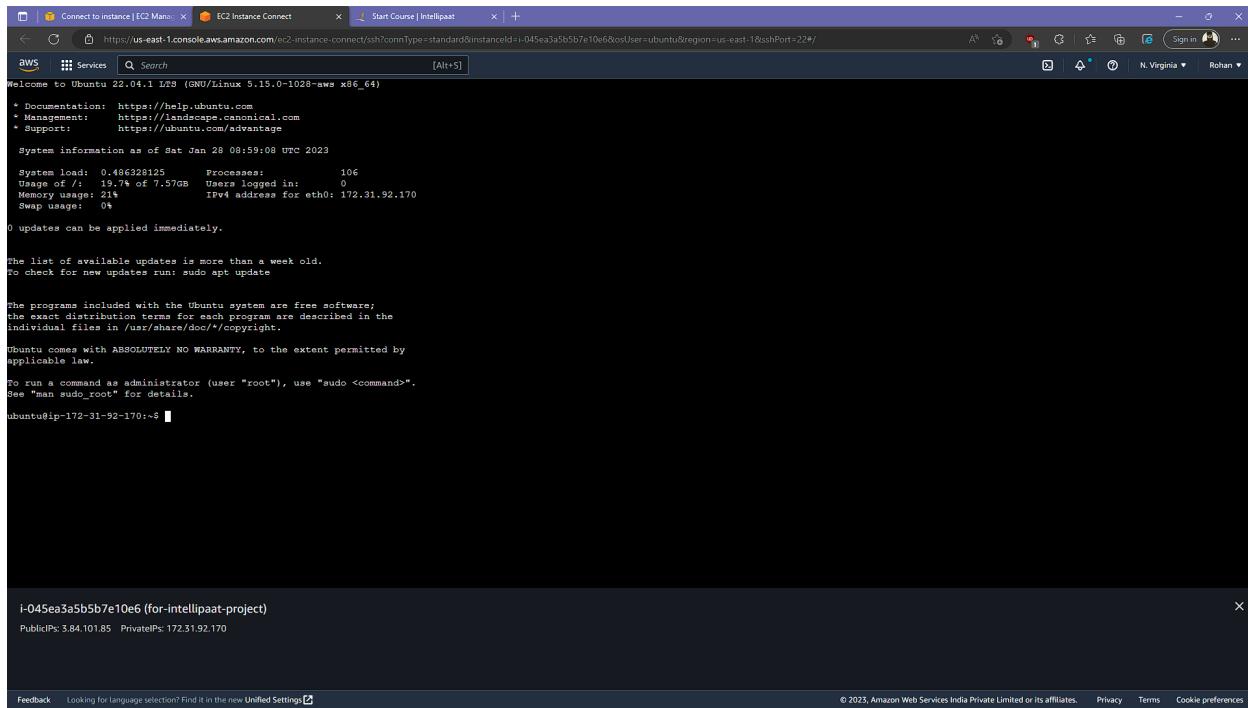
Sudo apt-get update

Apt-get install apache2

Then we install mysql client:

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

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



The screenshot shows a terminal window titled "EC2 Instance Connect" with the URL "https://us-east-1.console.aws.amazon.com/ec2-instance-connect/sh?connType=standard&instanceId=i-045ea3a5b5b7e10e6&osUser=ubuntu®ion=us-east-1&sshPort=22#/".

The terminal displays the following text:

```
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1028-sws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

system information as of Sat Jan 28 08:59:08 UTC 2023

System load: 0.46328125  Processes: 106
Usage of /: 19.7% of 7.57GB  Users logged in: 0
Memory usage: 21%
Swap usage: 0%

0 updates can be applied immediately.

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-172-31-92-170:~$
```

The terminal window has a dark background and light-colored text. At the bottom, it shows the instance ID "i-045ea3a5b5b7e10e6 (for-intellipaat-project)", public IP "5.84.101.85", and private IP "172.31.92.170".

At the very bottom, there is footer text: "Feedback" and "Looking for language selection? Find it in the new Unified Settings.", followed by "© 2023, Amazon Web Services India Private Limited or its affiliates.", "Privacy", "Terms", and "Cookie preferences".

Done

```
Creating config file /etc/php/5.6/mods-available/mysqlnd.ini with new version
Creating config file /etc/php/5.6/mods-available/mysqli.ini with new version
Creating config file /etc/php/5.6/mods-available/pdo_mysql.ini with new version
Creating config file /etc/php/5.6/mods-available/mysqli.ini with new version
Setting up php5.6-opcache (5.6.40-64+ubuntu22.04.1+deb.sury.org+1) ...
Creating config file /etc/php/5.6/mods-available/opcache.ini with new version
Setting up mysql-client-8.0 (8.0.32-0ubuntu0.22.04.1) ...
Setting up php5.6-readline (5.6.40-64+ubuntu22.04.1+deb.sury.org+1) ...
Creating config file /etc/php/5.6/mods-available/readline.ini with new version
Setting up php5.6-cl1 (5.6.40-64+ubuntu22.04.1+deb.sury.org+1)
update-alternatives: using /usr/bin/php5.6 to provide /usr/bin/php (php) in auto mode
update-alternatives: using /usr/bin/phar5.6 to provide /usr/bin/phar (phar) in auto mode
update-alternatives: using /usr/bin/phar.phar5.6 to provide /usr/bin/phar.phar (phar.phar) in auto mode
Creating config file /etc/php/5.6/cl1/php.ini with new version
Setting up mysql-client (8.0.32-0ubuntu0.22.04.1) ...
Setting up libapache2-mod-php5.6 (5.6.40-64+ubuntu22.04.1+deb.sury.org+1) ...
Creating config file /etc/php/5.6/apache2/php.ini with new version
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch_mpm Switch to prefork
apache2_invoke() for mpm module php5.6
Setting up libapache2-mod-php5.6 (5.6.40-64+ubuntu22.04.1+deb.sury.org+1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for php5.6-cl1 (5.6.40-64+ubuntu22.04.1+deb.sury.org+1) ...
Processing triggers for libapache2-mod-php5.6 (5.6.40-64+ubuntu22.04.1+deb.sury.org+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.
root@ip-172-31-92-170:/home/ubuntu# i-045ea3a5b5b7e10e6 (for intellipaat-project)
PublicIPs: 3.84.101.85 PrivateIPs: 172.31.92.170
```

Now let's create a DB.

The screenshot shows the AWS RDS Management Console interface. The left sidebar includes options like Dashboard, Databases, Query Editor, Performance insights, Snapshots, and more. The main content area displays RDS resources such as DB Instances, Allocated storage, DB Clusters, Reserved instances, Snapshots, and Events. A large 'Create database' button is visible. To the right, there are recommended sections for PostgreSQL, operational tasks, backup and restore, and cross-region disaster recovery. The footer contains standard AWS links for feedback, copyright, and cookie preferences.

Choose MySQL

The screenshot shows the 'Create Database' wizard for MySQL on the AWS RDS Management Console. The 'Easy create' option is selected. The configuration section shows MySQL chosen as the engine type. The DB instance size section shows the 'Free tier' selected. The DB instance identifier field contains 'database-1'. A right-hand sidebar provides details about MySQL support on RDS.

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.

Let the public accessibility be YES.

The screenshot shows the 'Create Database' wizard for MySQL on the AWS RDS Management Console. The 'IPv4' network type is selected. Under 'Public access', the 'Yes' option is selected. Under 'VPC security group (firewall)', the 'Choose existing' option is selected with 'default' chosen. A right-hand sidebar provides details about MySQL support on RDS.

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.

And create Database.

The screenshot shows the AWS RDS Management Console interface. A modal window titled 'Create database' is open, prompting the user to confirm they have the necessary rights for third-party products or services. Below the modal, there are sections for roles, monitoring, additional configuration, and estimated monthly costs. On the right side, a detailed description of the MySQL service is provided, listing its features such as 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. At the bottom right of the modal is a 'Create database' button.

Endpoint and port has been generated.

The screenshot shows the AWS RDS Management Console interface, specifically the 'Databases' section for a database named 'database-1'. The 'Summary' tab is selected, displaying basic information like DB identifier, CPU usage, status (Backing-up), engine (MySQL Community), and class (db.t3.micro). The 'Connectivity & security' tab is also visible, showing the endpoint (database-1.cxxfgfwrecor.us-east-1.rds.amazonaws.com) and port (3306). The 'Networking' section lists the VPC (vpc-0c0b71ad96deea470), subnet group (rds-ec2-db-subnet-group-1), and subnets (subnet-067c13933f05be019, subnet-0e13a73d98e7f6c). The 'Security' section shows VPC security groups (default sg-0ab535c87b560d71d) and a certificate authority (rds-ca-2019).

Let's change security group rules of RDS.

The screenshot shows the AWS EC2 Instances page. A single instance, "for-intellipaat-project" (ID: i-045ea3a5b5b7e10e6), is listed as "Running". The instance details pane shows the following information:

- IAM Role: -
- Owner ID: 905048332148
- Launch time: Sat Jan 28 2023 14:28:10 GMT+0530 (India Standard Time)
- Security groups: sg-01e704fb1f1d94e4b (launch-wizard-10)

The "Inbound rules" section displays the following table:

Name	Security group rule ID	Port range	Protocol	Source	Security groups	Description
-	sgr-09e75c4b58bf7ad7	22	TCP	0.0.0.0/0	launch-wizard-10	-

The "Outbound rules" section shows a table with no visible data.

RDS Management Console | EC2 Instance Connect | Dashboard | EC2 Management | Start Course | Intellipaat | VALORANT: Riot Games' competitive | +

https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#database-id=database-1&is-cluster=false

aws Services Search [Alt+S]

Amazon RDS Actions

Databases Modify

database-1

Summary

DB identifier database-1	CPU <div style="width: 7.37%;">7.37%</div>	Status Backing-up	Class db.t3.micro
Role Instance	Current activity <div style="width: 0%;">0 Connections</div>	Engine MySQL Community	Region & AZ us-east-1f

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance & backups | Tags

Connectivity & security

Endpoint & port	Networking	Security
Endpoint database-1.cxfqfwrecor.us-east-1.rds.amazonaws.com	Availability Zone us-east-1f	VPC security groups default (sg-0ab535c87b560d71d) Active
Port 3306	VPC vpc-0c0b71ad96deea470	Publicly accessible Yes
	Subnet group rds-e2-db-subnet-group-1	Certificate authority Info rds-ca-2019
	Subnets subnet-06fa51d320719f3a subnet-067c13933105be019 subnet-0e015a73d98e7f6c	Certificate authority date August 22, 2024, 22:38 (UTC+05:30)
	Network type IPv4	DB instance certificate expiration date August 22, 2024, 22:38 (UTC+05:30)

https://us-east-1.console.aws.amazon.com/rds/v2/home?region=us-east-1#SecurityGroups;search=sg-0ab535c87b560d71d.

EC2 Management Console | EC2 Instance Connect | Dashboard | EC2 Management | Start Course | Intellipaat | VALORANT: Riot Games' competitive | +

https://us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#SecurityGroups;search=sg-0ab535c87b560d71d

aws Services Search [Alt+S]

New EC2 Experience Create security group

Security Groups (1/1) Actions Export security groups to CSV

search: sg-0ab535c87b560d71d | Clear filters

Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count	Outbound rules count
-	sg-0ab535c87b560d71d	default	vpc-0c0b71ad96deea470	default VPC security gr...	905048332148	1 Permission entry	1 Permission entry

sg-0ab535c87b560d71d - default

Details Inbound rules Outbound rules Tags

You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer

Inbound rules (1/1) Actions Manage tags Edit inbound rules

Name	Security group rule ID	Type	Protocol	Port range	Source	Description
-	sg-0e3ff6ee39a4ab0e9	All traffic	All	All	sg-0ab535c87b560d7...	

https://us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#settings

Edit inbound rules of rds and choose source as Ec2 instance security group that we just created.

The screenshot shows the AWS EC2 Management Console with the URL <https://us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#ModifyInboundSecurityGroupRulessecurityGroupId=sg-0ab535c87b560d71d>. The page title is "Edit inbound rules". The navigation path is "EC2 > Security Groups > sg-0ab535c87b560d71d - default > Edit inbound rules".

The main section is titled "Inbound rules" with a "Info" link. It lists one rule:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0e3ff6ee39a4ab0e9	MySQL/Aurora	TCP	3306	Custom	sg-01e704fb1fd94e4b X

Below the table are buttons for "Add rule", "Cancel", "Preview changes", and "Save rules".

Back to ssh.Let's connect Our RDS to Ec2.

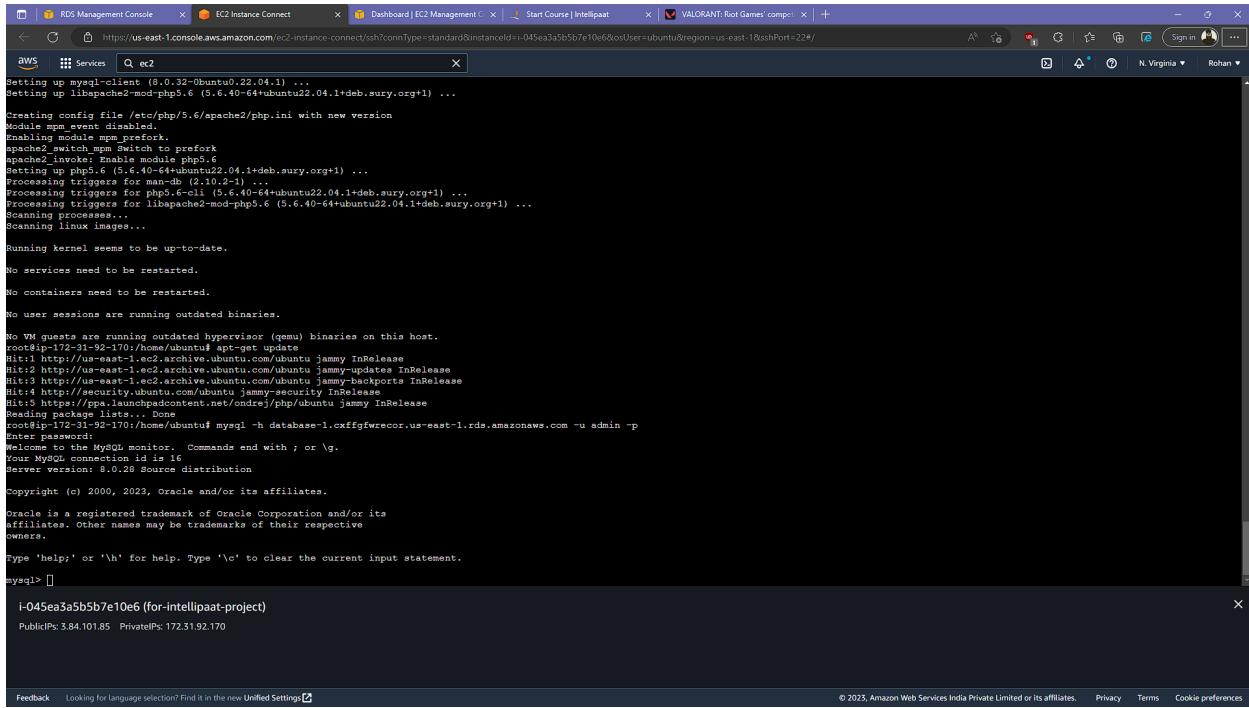
Use following commands:

```
mysql -h database-1.cxfffwrecom.us-east-1.rds.amazonaws.com -u admin -p
```

Then enter password for it.

Create a database named project.

```
CREATE DATABASE <database_name>;
```



```
Setting up mysql-client (8.0.32-0ubuntu0.22.04.1) ...
Setting up libapache2-mod-php5.6 (5.6.40-64+ubuntu22.04.1+deb.sury.org+1) ...

Creating config file /etc/php/5.6/apache2/php.ini with new version
Module 'memcached' is available.
Enabling module mpm_prefork.
apache2: switch mpm Switch to prefork
apache2: invoke-rc.d: Enable module php5.6
Setting up php5.6 (5.6.40-64+ubuntu22.04.1+deb.sury.org+1) ...
Processing triggers for libapache2-mod-php5.6 (5.6.40-64+ubuntu22.04.1+deb.sury.org+1) ...
Processing triggers for libapache2-mod-php5.6 (5.6.40-64+ubuntu22.04.1+deb.sury.org+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.
root@ip-172-31-92-170:/home/ubuntu# apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 https://ppa.launchpadcontent.net/ondrej/php/ubuntu jammy InRelease
Reading package lists...
root@ip-172-31-92-170:/home/ubuntu# mysql -h database-1.cxfffwrecom.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 8.0.28 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> []
i-045ea3a5b5b7e10e6 (for-intellipaat-project)
PublicIPs: 3.84.101.85 PrivateIPs: 172.31.92.170
```

**Now cd /var/www/html
Remove index.html using rm index.html**

```

aws services ec2
Enabling module mpm_prefork.
apache2 switch mpm Switch to prefork
apache2 invoke: Enable module php5.6
Setting up php5.6 (5.6.40-64+ubuntu20.04.1+deb.sury.org+1) ...
Processing triggers for libapache2-mod-php5.6 (5.6.40-64+ubuntu20.04.1+deb.sury.org+1) ...
Processing triggers for libapache2-mod-php5.6 (5.6.40-64+ubuntu20.04.1+deb.sury.org+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.

root@ip-172-31-92-170:/home/ubuntu# apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 https://ppa.launchpadcontent.net/ondrej/php/ubuntu jammy InRelease
Reading package lists...
Done
root@ip-172-31-92-170:/home/ubuntu mysql -h database-1.cxfggfwreco.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 8.0.28 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> [1] Stopped                  mysql -h database-1.cxfggfwreco.us-east-1.rds.amazonaws.com -u admin -p
root@ip-172-31-92-170:/home/ubuntu$ cd /var/www/html
root@ip-172-31-92-170:/var/www/html$ ls
index.html
root@ip-172-31-92-170:/var/www/html$ [1]

i-045ea3a5b5b7e10e6 (for-intellipaat-project)

PublicIPs: 3.84.101.85 PrivateIPs: 172.31.92.170

```

```

aws services ec2
root@ip-172-31-92-170:/var/www/html$ mysql -h database-1.cxfggfwreco.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.0.28 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> show databases;
-> show databases;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'show databases' at line 2
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql> use project
ERROR 1049 (42000): Unknown database 'project'
mysql> CREATE DATABASE project;
Query OK, 1 row affected (0.01 sec)

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

mysql> [1]

i-045ea3a5b5b7e10e6 (for-intellipaat-project)

PublicIPs: 3.84.101.85 PrivateIPs: 172.31.92.170

```

now nano index.php and paste the code provided at the end.

```
GNU nano 6.2
index.php *
// Create connection
$conn = new mysqli($servername, $username, $password, $db);

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

if(isset($_POST['firstname']) && isset($_POST['email'])){
$sql = "INSERT INTO data (firstname,email)
VALUES ('".$_firstname."', '".$_email."')";

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

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

Make changes as per you database. Paste the username pwd and endpoint of your DB.

```
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="#f2f2f2">
<div style="background-color: #f2f2f2; height: 100% 100%">
<br><br><br>
<div class="container">
<div class="jumbotron vertical-center">
<div class="grid" style="text-align: center; margin-top: 10px">
<td colspan="4">
<form method="post">
<label for="firstname">Name:</label>
<input type="text" class="form-control" name="firstname">
</div>
<div class="form-group">
<label for="email">Email:</label>
<input type="text" class="form-control" name="email">
</div>
</form></td>
<td colspan="4">
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center; margin-bottom: 10px">
<tr>
<td style="padding: 5px;">First Name:</td>
<td style="padding: 5px;">Last Name:</td>
<td style="padding: 5px;">Address:</td>
<td style="padding: 5px;">City:</td>
</tr>
<tr>
<td style="padding: 5px;"><input type="text" name="first_name"></td>
<td style="padding: 5px;"><input type="text" name="last_name"></td>
<td style="padding: 5px;"><input type="text" name="address"></td>
<td style="padding: 5px;"><input type="text" name="city"></td>
</tr>
<tr>
<td colspan="4" style="text-align: right; padding-top: 10px; padding-bottom: 10px; font-weight: bold; font-size: 1.2em">
<button type="submit" class="btn btn-success">Submit</button>
</td>
</tr>
</table>
</div>
</div>
</div>
</body>
</html>

$firstname=$_POST['firstname'];
$email=$_POST['email'];
$servername = "database-1.cxxfgfwreco.us-east-1.rds.amazonaws.com";
$username = "admin";
$password = "DrInk7Up";
$db = "project";
// 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'])){

Help      Write Out   Where Is   Cut     Execute   Location   Undo   Set Mark   To Bracket   Previous   Back   Prev Word
X Exit    Read File  Replace  Paste   Justify   Go To Line  Redo   Copy       Where Was   Next    Forward  Next Word

```

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-09e75c4b588fb7ad7	SSH	TCP	22	Custom	0.0.0.0/0
-	HTTP	TCP	80	Anywhere-IPv4	0.0.0.0/0

Add rule

Cancel | Preview changes | Save rules

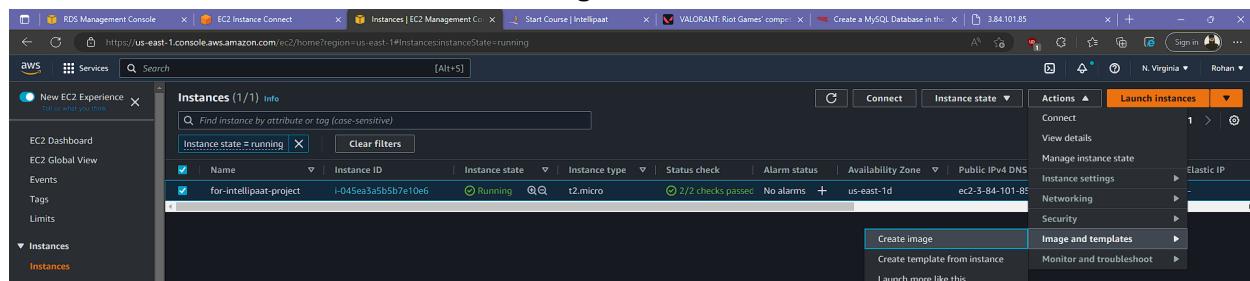
Using public ip of ec2. WEbsite is accessible.

Name:

Email:

Submit

Create image of the instance.



The screenshot shows the AWS EC2 Instances page. A single instance, `i-045ea3a5b5b7e10e6` (named `for-intellipaat-project`), is listed as `Running`. The context menu for this instance is open, with the `Create image` option highlighted.

Instance: i-045ea3a5b5b7e10e6 (for-intellipaat-project)

Details | **Security** | **Networking** | **Storage** | **Status checks** | **Monitoring** | **Tags**

Instance summary

Attribute	Value
Instance ID	<code>i-045ea3a5b5b7e10e6 (for-intellipaat-project)</code>
Public IPv4 address	<code>3.84.101.85</code> [open address]
Instance state	<code>Running</code>
Private IP DNS name (IPv4 only)	<code>ip-172-31-92-170.ec2.internal</code>
Instance type	<code>t2.micro</code>
VPC ID	<code>vpc-0c0b71ad96deea470</code>
Subnet ID	...

Image creation details

Instance ID: `i-045ea3a5b5b7e10e6 (for-intellipaat-project)`

Image name: `for-project`

Image description (optional): `for-project`

No reboot: Enable

Instance volumes:

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/sda1	Create new snapshot from volume	8	EBS General Purpose S...	100		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

Tags - optional:

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.

Add new tag

Cancel **Create image**

Let's create autoscaling group and load balancer.

The screenshot shows the AWS EC2 Auto Scaling homepage. On the left, there is a navigation sidebar with various services like Instances, Instance Types, Launch Templates, etc. The main content area features a large heading "Amazon EC2 Auto Scaling" with the subtext "helps maintain the availability of your applications". Below this is a section titled "How it works" with a diagram illustrating an Auto Scaling group as a collection of EC2 instances. A callout box provides a detailed explanation of how capacity values define the bounds of the group. To the right, there are sections for "Pricing" and "Getting started" with links to further documentation.

The screenshot shows the "Create launch template with default config." wizard. The left side contains configuration sections for "Key pair name" (set to "project"), "Network settings" (with "Subnet info" and "Firewall (security group)"), and "Storage (volumes)" (listing "Volume 1 (AMI Root) (8 GB, EBS, General purpose SSD (gp2))"). The right side displays a "Summary" panel with details like "Software Image (AMI)", "Virtual server type (instance type) t2.micro", and a note about the free tier. A modal window in the center provides additional information about the free tier, stating it includes 750 hours of t2.micro or t3.micro instance usage per month, 30 GiB of EBS storage, 2 million IOPS, 1 GB of snapshots, and 100 GB of bandwidth. At the bottom right is a prominent "Create launch template" button.

Attach that Launch Config to Autoscaling group.

The screenshot shows the AWS Create Auto Scaling Group wizard at Step 6: Launch template. The main panel displays the configuration for a launch template named "for-project". It includes fields for AMI ID (ami-00874d747dde814fa), Security groups (sg-01e704fb1f1d94e4b), and Instance type (t2.micro). Below this, the "Additional details" section shows the storage volume information. On the right, there are tabs for "Launch template" and "Switch to launch configuration". At the bottom, there are "Cancel" and "Next" buttons.

Allow all AZ's

The screenshot shows the AWS Create Auto Scaling Group wizard at Step 7: Instance type requirements. The "Additional details" section lists several subnets under "Available subnets": us-east-1b | subnet-0fb310d8014c6ae2e, us-east-1c | subnet-06e4a61da30719f5a (RDS-Pvt-subnet-2), us-east-1c | subnet-015b4d99c952086b4, us-east-1d | subnet-08e7edf2dfa714be, us-east-1e | subnet-02c27fc26c56faaa, us-east-1e | subnet-0e0e15a73d98e7f6c (RDS-Pvt-subnet-1), us-east-1f | subnet-0fd9b191871ed7d8f (private), and us-east-1f | subnet-067c13933f05be019 (RDS-Pvt-subnet-3). Below this, the "Instance type requirements" section shows the launch template "for-project" and instance type "t2.micro". At the bottom, there are "Cancel", "Previous", "Skip to review", and "Next" buttons.

Attach a load balancer now self.

The screenshot shows the AWS Create Auto Scaling Group wizard at Step 3: Configure advanced options. The 'Attach to a new load balancer' option is selected, highlighted with a blue border. The 'Load balancer type' section shows 'Application Load Balancer (HTTP, HTTPS)' selected. The 'Load balancer name' field contains 'for-project-1'. The 'Load balancer scheme' section shows 'Internet-facing' selected. The 'Network mapping' note indicates the new load balancer will inherit VPC and Availability Zone settings from the Auto Scaling group. The left sidebar lists steps 1 through 7.

We will create a application load balancer.

The screenshot shows the AWS Create Auto Scaling Group wizard at Step 3: Configure advanced options. The 'Attach to a new load balancer' option is selected. The 'Availability Zones and subnets' section is expanded, showing three subnets: us-east-1f, us-east-1b, and us-east-1d, each associated with a specific subnet ID. The left sidebar lists steps 1 through 7.

The screenshot shows the AWS CloudFormation console during the creation of an Auto Scaling group. The current step is "Configure Load Balancing".

Listeners and routing

If you require secure listeners, or multiple listeners, you can configure them from the Load Balancing console after your load balancer is created.

Protocol	Port	Default routing (forward to)
HTTP	80	Select new or existing target group

Tags - optional

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

Add tag

50 remaining

Health checks - optional

Health check type Info

EC2 Auto Scaling automatically replaces instances that fail health checks. If you enabled load balancing, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.

EC2 ELB

Health check grace period

The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.

10 seconds

Additional settings - optional

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

Cancel Previous Skip to review Next

The screenshot shows the AWS Auto Scaling Group creation wizard at Step 4. The left sidebar lists steps from 2 to 7. The main panel is titled "Configure group size and scaling policies". It includes fields for "Desired capacity" (set to 1), "Minimum capacity" (set to 1), and "Maximum capacity" (set to 2). Below this is a section for "Scaling policies - optional" with a note about using a scaling policy to resize the group. It shows a radio button for "Target tracking scaling policy" (disabled) and a selected radio button for "None". At the bottom are buttons for "Cancel", "Previous", "Skip to review", and "Next".

The screenshot shows the AWS Auto Scaling Group creation wizard at Step 5. The left sidebar lists steps from 1 to 7. The main panel is titled "Add notifications" with a note about sending SNS notifications for launches and terminations. It has a "Add notification" button. At the bottom are buttons for "Cancel", "Previous", "Skip to review", and "Next".

Load balancer is created.

The screenshot shows the AWS CloudWatch Metrics interface. On the left, there's a navigation pane with 'Metrics' selected. The main area displays a table of metrics for a single metric named 'HelloWorld'. The table has columns for 'Metric Name', 'Value', 'Unit', 'Period', 'Time Range', and 'Dimensions'. One dimension is visible: 'Region' with the value 'us-east-1'. The table shows two data points: one at 10:00 UTC on January 28, 2023, with a value of 1, and another at 10:05 UTC on January 28, 2023, with a value of 0.

Metric Name	Value	Unit	Period	Time Range	Dimensions
HelloWorld	1	1	1 minute	10:00 UTC on Jan 28, 2023	Region: us-east-1
HelloWorld	0	1	1 minute	10:05 UTC on Jan 28, 2023	Region: us-east-1

Using load balancers's DNS we can access the website.

The screenshot shows a web browser window with the URL 'for-project-1-845613599.us-east-1.elb.amazonaws.com'. The page displays a simple contact form with fields for 'Name' and 'Email', and a 'Submit' button. The background of the page is white, and the form is centered.




```
$email=$_POST['email'];
$servername = "intelli.coghw13fheqo.us-east-2.rds.amazonaws.com";
$username = "admin";
$password = "Drink7up";
$db = "project";
// Create connection
$conn = new mysqli($servername, $username, $password, $db);

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

if(isset($_POST['firstname']) && isset($_POST['email'])){
$sql = "INSERT INTO data (firstname,email)
VALUES ('".$firstname."', '".$email."')";

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

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