

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

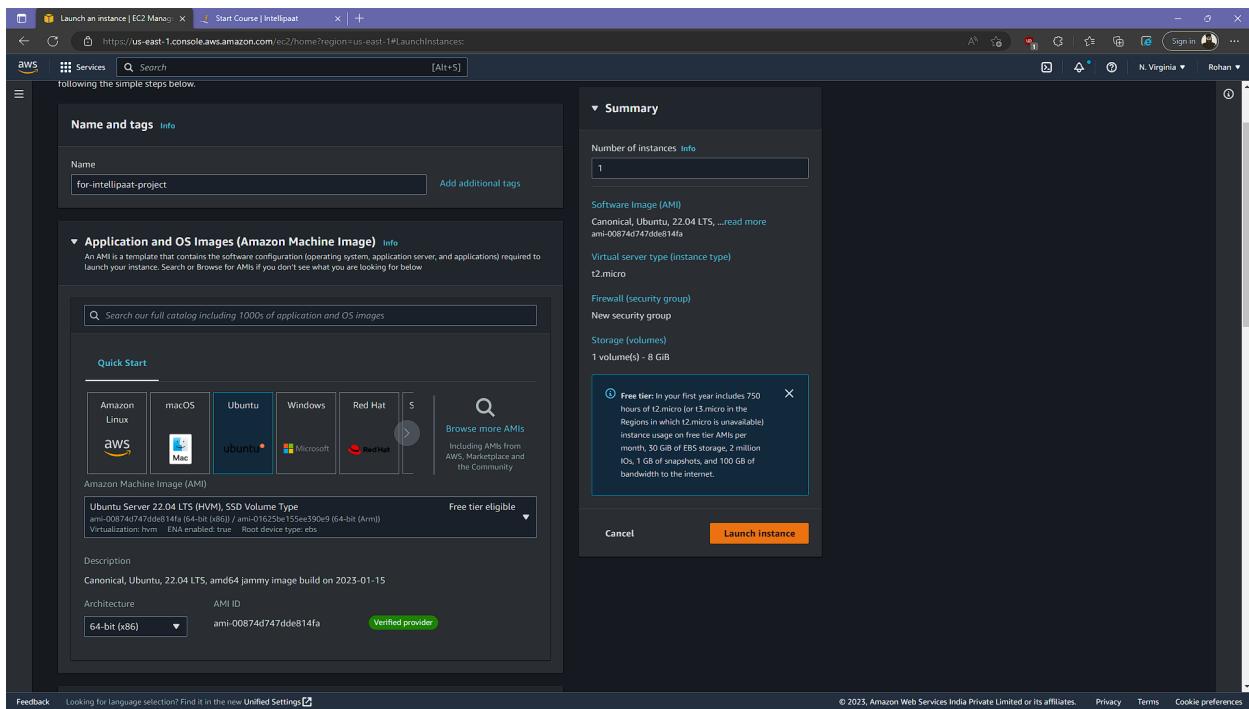
Topic: Deploy a Multi-tier website using EC2

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.



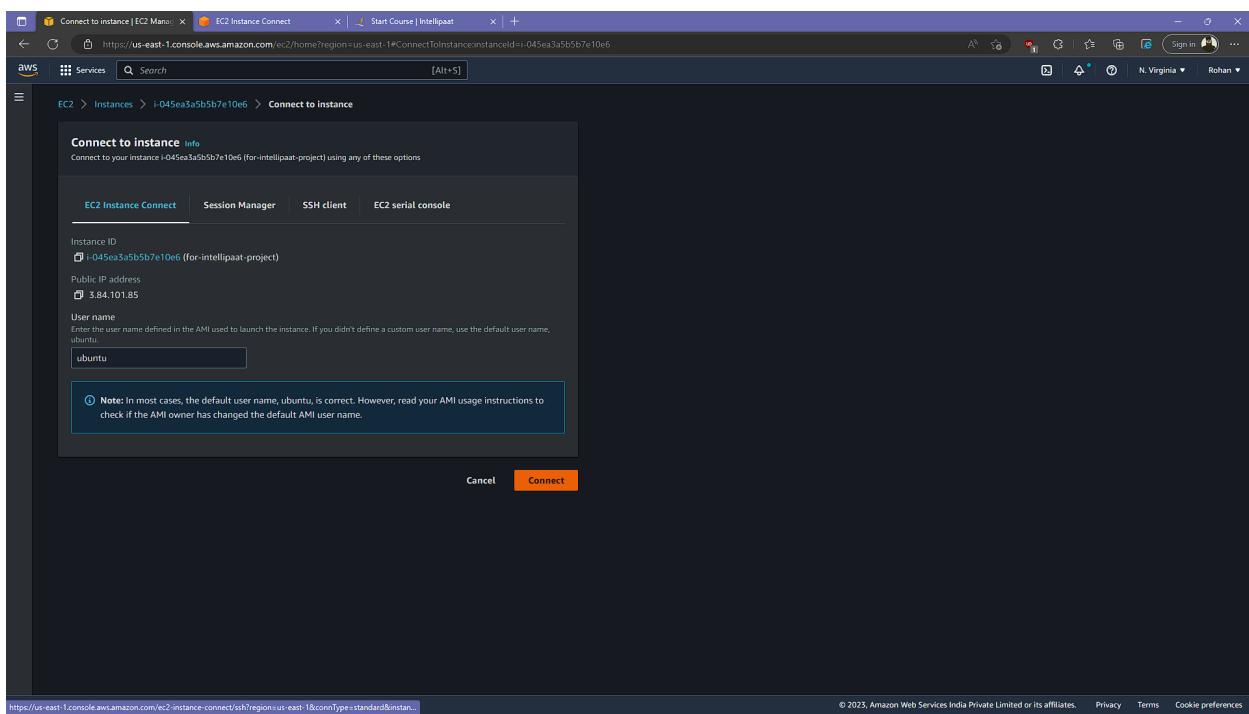
Create a keypair and assign to it. Allow only ssh for now from anywhere.

The screenshot shows the AWS EC2 Launch Instance wizard. In the 'Network settings' section, a security group named 'launch-wizard-10' is being created with rules allowing SSH traffic from anywhere. The 'Configure storage' section indicates 1 volume(s) - 8 GiB. A summary box shows 1 instance launching with the Canonical, Ubuntu, 22.04 LTS AMI. A modal window provides details about the free tier, including 750 hours of t2.micro usage per month. The 'Launch instance' button is highlighted.

Instance has been created.

The screenshot shows the AWS EC2 Instances page. A single instance named 'for-intellipaat-project' is listed, showing it is running (Public IP: 3.84.101.85). The instance details page is open, showing the instance ID (i-045ea3a5b5b7e10e6), state (Running), type (t2.micro), and subnet (us-east-1d). It also lists the public and private IP addresses, along with the instance's host name and security group.

Let's SSH to it.



Use following commands once connecting to instance:

Sudo su

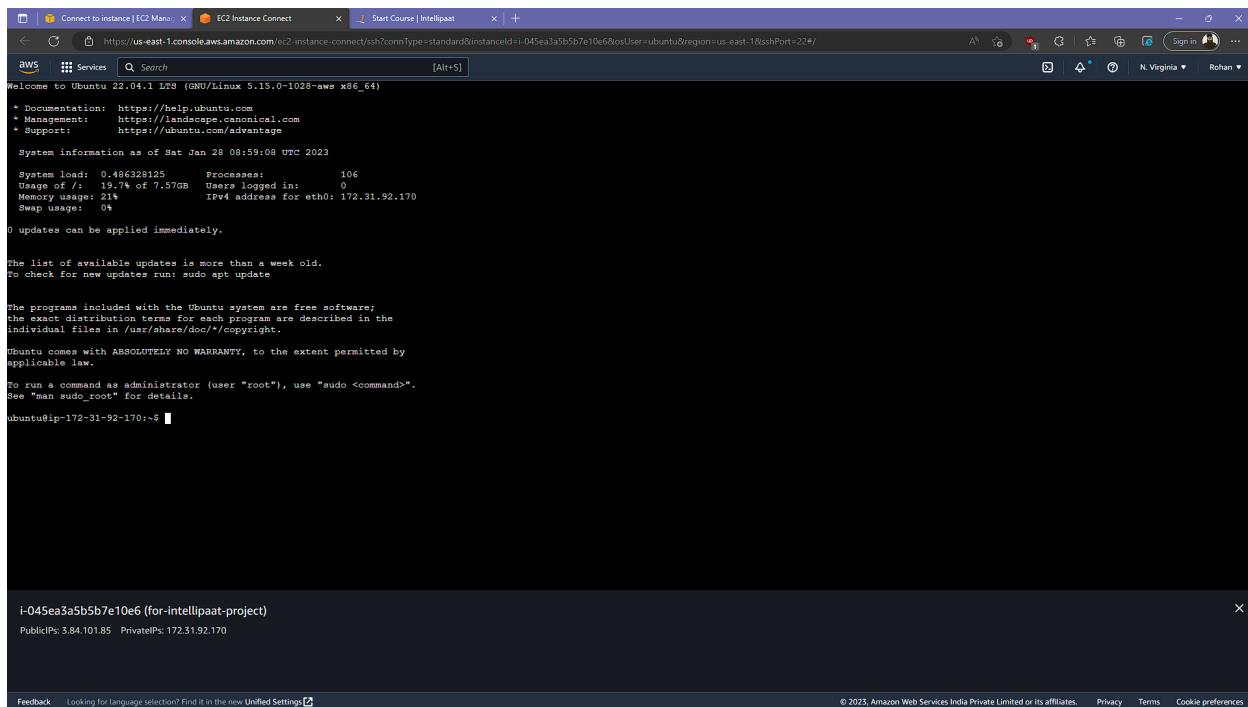
Sudo apt-get update

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 within the AWS EC2 Instance Connect interface. The terminal displays the following system information:

```
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:~$
```

At the bottom of the terminal window, it shows the instance ID: i-045ea3a5b5b7e10e6, PublicIP: 58.410.185, and PrivateIP: 172.31.92.170.

Below the terminal window, the AWS navigation bar includes 'AWS Services' and 'Search' fields, along with account information for 'N. Virginia' and 'Rohan'.

Done

```
Creating config file /etc/php/5.6/mods-available/mysqlind.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/mysql.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-cl (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/phar5.6 to provide /usr/bin/phar.phar (phar.phar) in auto mode
Creating config file /etc/php/5.6/cl/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 module php5.6
Setting up libapache2-mod-php5.6 (5.6.40-64+ubuntu22.04.1+deb.sury.org+1) ...
Processing triggers for mysql-db (2.10.2-1) ...
Processing triggers for php5.6-cl (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 (forintellipaat-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, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Events, Event subscriptions, Recommendations, and Certificate update. The main content area displays RDS resources: DB Instances (0/40), Allocated storage (0 TB/100 TB), DB Clusters (0/40), Reserved instances (0/40), Snapshots (2), and Events (7). A large 'Create database' button is visible. To the right, there are 'Recommended for you' sections for PostgreSQL, RDS operational tasks, backup and restore, and cross-region disaster recovery. The bottom right contains additional information links and footer links for feedback, copyright, and cookie preferences.

Choose MySQL

The screenshot shows the 'Create Database' wizard in the AWS RDS Management Console. The 'MySQL' engine is selected. Key configuration choices include:

- DB instance identifier:** database-1
- DB instance size:** Free tier (db.t2.micro, 2 vCPUs, 1 GiB RAM, 20 GiB, 0.020 USD/hour)
- Network type:** IPv4
- Public access:** Yes (public IP assigned)
- Existing VPC security groups:** default

A sidebar on the right provides a summary of MySQL features.

Let the public accessibility be YES.

The screenshot shows the 'Create Database' wizard with public accessibility set to 'Yes'. Other configurations remain the same as the previous screenshot.

And create Database.

The screenshot shows the AWS RDS Management Console interface. A modal window titled 'Create database' is open, prompting the user to enter a database identifier. The identifier 'database-1' is already filled in. To the right of the modal, a detailed description of MySQL is displayed, highlighting its features such as support for up to 64 TiB of database size, General Purpose, Memory Optimized, and Burstable Performance instance classes, automated backup and point-in-time recovery, and up to 15 Read Replicas per instance. Below the MySQL info, a note states: '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.'

Endpoint and port has been generated.

The screenshot shows the 'Summary' and 'Connectivity & security' tabs for the 'database-1' database. In the 'Summary' tab, key details are listed: DB identifier (database-1), CPU usage (7.37%), Status (Backing-up), Engine (MySQL Community), Class (db.t3.micro), and Region & AZ (us-east-1f). In the 'Connectivity & security' tab, the endpoint is listed as 'database-1.cfifgfwreco.us-east-1.rds.amazonaws.com' and the port is listed as '3306'. The networking section shows the VPC (vpc-0c0b71ad96deea470) and subnet group (rds-ec2-db-subnet-group-1). The security section lists the VPC security groups (default sg-0ab535c87b560d71d, active) and the publicly accessible status (Yes). Certificate authority and expiration date information is also provided.

Let's change security group rules of RDS.

The screenshot shows the AWS EC2 Instances page with a single instance listed:

- Instance Details:** Name: for-intellipaat-project, Instance ID: i-045ea3a5b5b7e10e6, Instance state: Running, Instance type: t2.micro, Status check: 2/2 checks passed, Alarm status: No alarms, Availability Zone: us-east-1d, Public IPv4 DNS: ec2-3-84-101-85.com., Public IPv4: 3.84.101.85, Elastic IP: -.
- Inbound Rules:** One rule exists: Name: sgr-09e75c4b58bf7ad7, Security group rule ID: sgr-09e75c4b58bf7ad7, Port range: 22, Protocol: TCP, Source: 0.0.0.0/0, Security groups: launch-wizard-10.
- Outbound Rules:** No rules exist.

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

Amazon RDS

Databases

Database identifier: database-1

Role: Instance

CPU: 7.37% Status: Backing-up Class: db.t3.micro

Current activity: 0 Connections Engine: MySQL Community Region & AZ: us-east-1f

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

Endpoint & port Networking Security

Events Event subscriptions

Recommendations 0 Certificate update

DB identifier: database-1 CPU: 7.37% Status: Backing-up Class: db.t3.micro

Role: Instance Current activity: 0 Connections Engine: MySQL Community Region & AZ: us-east-1f

VPC: vpc-0c0b71ad96deea470 Publicly accessible: Yes

Subnet group: rds-e2-db-subnet-group-1 Certificate authority: rds-ca-2019

Subnets: subnet-06fa51d020719f3a, 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)

© 2025, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

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

New EC2 Experience

EC2 Dashboard

Events

Tags

Limits

Instances

Images

Elastic Block Store

Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Security group ID: sg-0ab535c87b560d71d

Name: default VPC ID: vpc-0c0b71ad96deea470 Description: default VPC security gr... Owner: 905048332148 Inbound rules count: 1 Outbound rules count: 1

Inbound rules (1/1)

sg-0e3ff6ee39a4ab0e9 All traffic All All sg-0ab535c87b560d7...

Run Reachability Analyzer

© 2025, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

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 breadcrumb navigation shows "EC2 > Security Groups > sg-0ab535c87b560d71d - default > Edit inbound rules".

The main content area displays the "Inbound rules" table. A single rule is listed:

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.

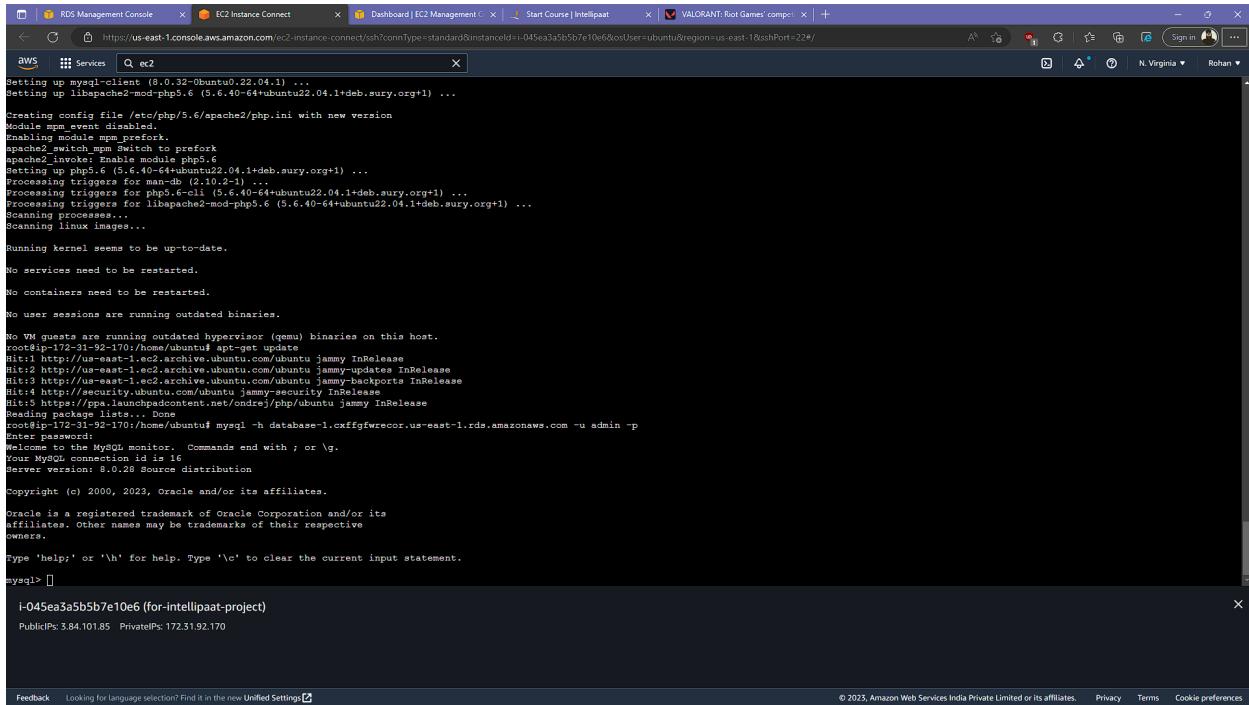
Use following commands:

```
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>;
```



The screenshot shows an AWS EC2 Instance Connect session. The terminal window displays the following MySQL command history:

```
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 Q 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.cxffgfwreco.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.cxffgfwreco.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 Q ec2
root@ip-172-31-92-170:/var/www/html$ mysql -h database-1.cxffgfwreco.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)

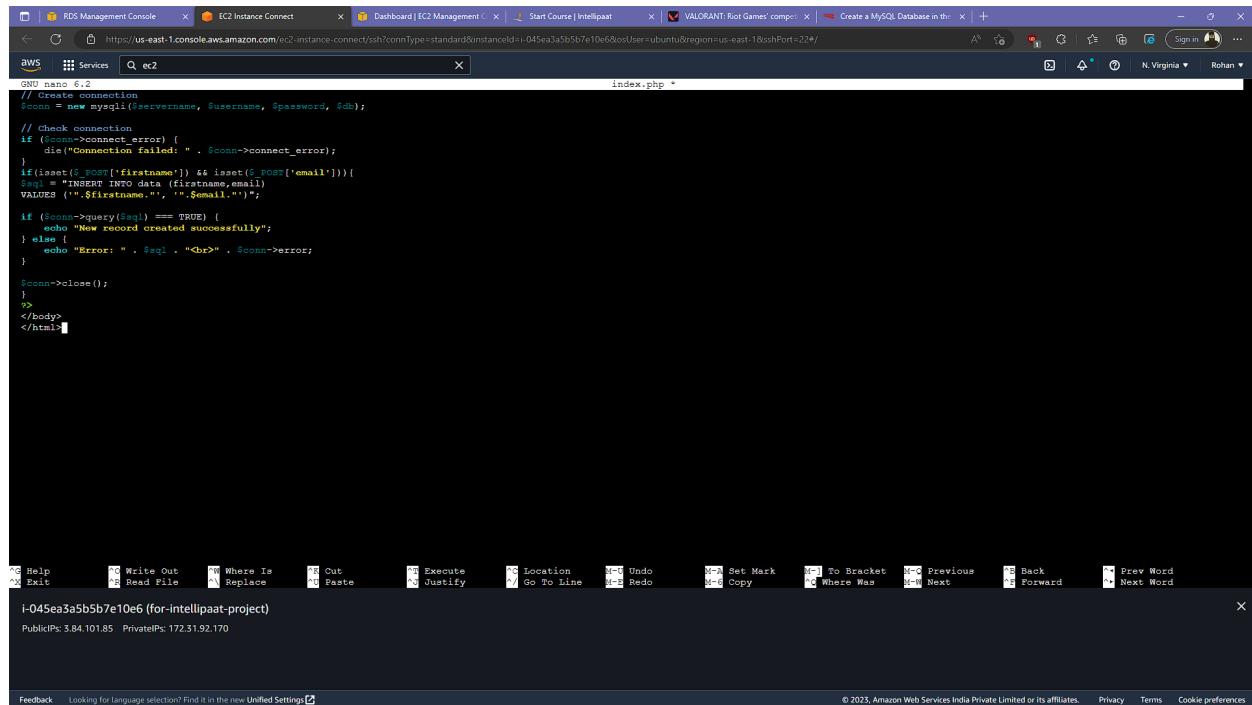
mysql> [1]

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);

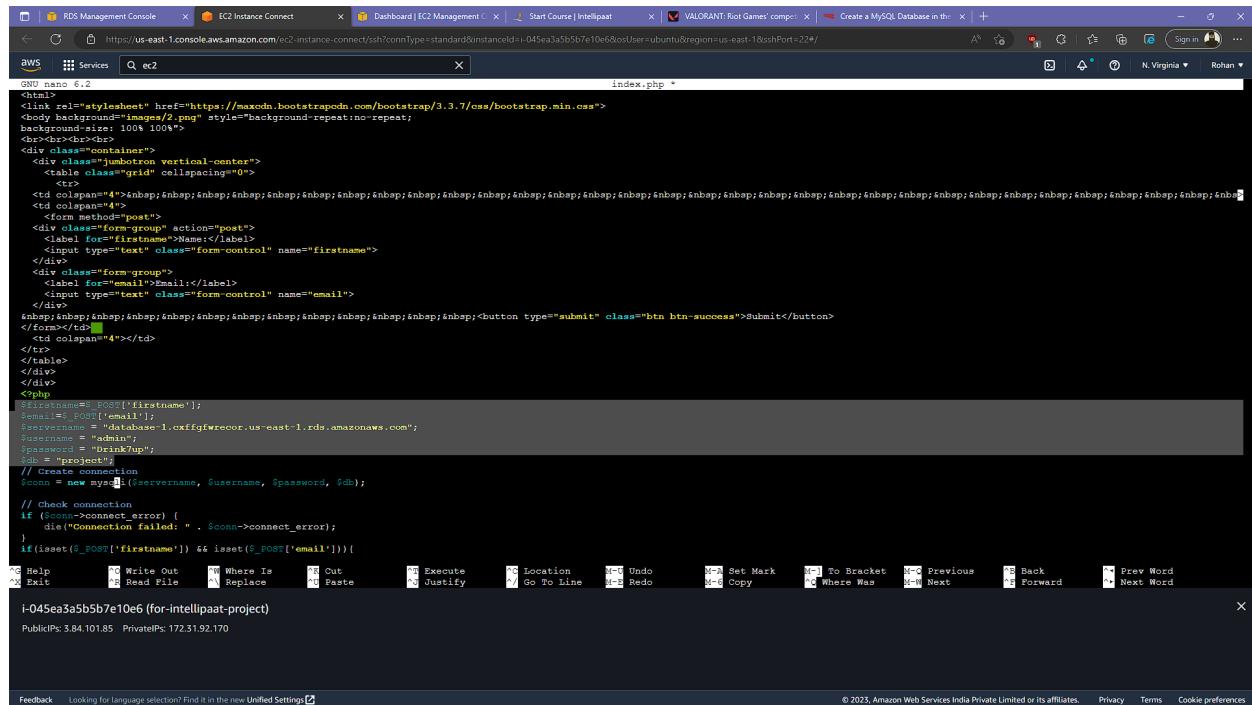
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

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 class="container">
<div class="jumbotron vertical-center">
<div class="grid" cellspacing="0">
<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">
</td>
</tr>
</table>
</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
Exit      Read File  Replace  Paste   Justify   Go To Line  Redo   Copy       Where Was   Next    Forward  Next Word
Feedback  Looking for language selection? Find it in the new Unified Settings. © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences
```

Inbound rules [Info](#)

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

[Add rule](#)

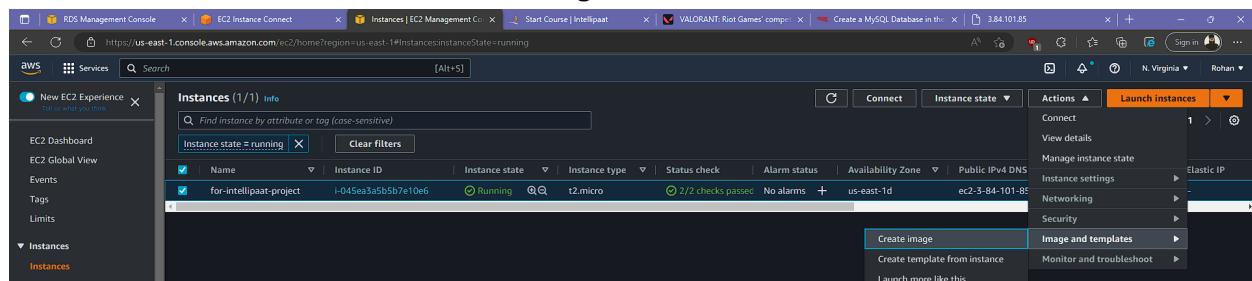
Cancel [Preview changes](#) [Save rules](#)

Using public ip of ec2. WEbsite is accessible.

Name:

Email:

Create image of the instance.



The screenshot shows the AWS EC2 Instances page. A single instance, `i-045ea3a5b5b7e10e6` (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	i-045ea3a5b5b7e10e6 (for-intellipaat-project)
Public IPv4 address	3.84.101.85 [open address]
Instance state	Running
Private IP DNS name (IPv4 only)	ip-172-31-92-170.ec2.internal
Instance type	t2.micro
VPC ID	vpc-00b71ad96deea470
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 service links like Instances, Instance Types, Launch Templates, etc. The main content area features a large heading "Amazon EC2 Auto Scaling helps maintain the availability of your applications". Below this, there is a section titled "How it works" with a diagram illustrating an Auto Scaling group as a collection of EC2 instances. A callout box states: "An Auto Scaling group is a collection of Amazon EC2 instances that are treated as a logical unit. You configure settings for a group and its instances as well as define the group's minimum, maximum, and desired capacity. Setting different minimum and maximum capacity values forms the bounds of the group, which allows the group to scale as the load on your application spikes higher or lower, based on demand. To scale the Auto Scaling group, you can either make manual adjustments to the desired capacity or let Amazon EC2 Auto Scaling automatically add and remove capacity to meet change in demand." Another callout box says: "When launching fleets of instances, you can specify what percentage of your capacity should be fulfilled by On-Demand instances, and what percentage with Spot Instances, to save up to 90% on costs." On the right side, there are sections for "Create Auto Scaling group", "Pricing", and "Getting started".

The screenshot shows the "Create launch template with default config." wizard. The left panel contains fields for "Key pair name" (set to "project"), "Network settings" (subnet info, security group selection), and "Storage (volumes)" (EBS volumes). The right panel is a "Summary" step showing the selected configuration: Software Image (AMI) is Canonical, Ubuntu, 22.04 LTS; Virtual server type (instance type) is t2.micro; Firewall (security group) is launch-wizard-10; Storage (volumes) is 1 volume(s) - 8 GiB. A callout box provides details about the free tier: "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 on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet." At the bottom, there are "Cancel" and "Create launch template" buttons.

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 Key pair name (project). The 'Additional details' section shows the storage volume was created on Jan 28, 2023, at 15:45:46 GMT+0530. At the bottom right 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 'Subnets' section lists several subnets across different availability zones (AZs): us-east-1b, us-east-1c, us-east-1d, us-east-1e, us-east-1f, and us-east-1f. The 'Instance type requirements' section shows the launch template 'for-project' (version 1) with an instance type of 't2.micro'. At the bottom right are 'Cancel', 'Previous', 'Skip to review', and 'Next' buttons.

Attach a load balancer now self.

Configure advanced options Info

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

Load balancing - optional 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.

Attach to a new load balancer

Define a new load balancer to create for attachment to this Auto Scaling group.

Load balancer type

Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, visit the [Load Balancing console](#).

Application Load Balancer HTTP, HTTPS

Network Load Balancer TCP, UDP, TLS

Load balancer name

Name cannot be changed after the load balancer is created.

for-project-1

Load balancer scheme

Scheme cannot be changed after the load balancer is created.

Internal

Internet-facing

Network mapping

Your new load balancer will be created using the same VPC and Availability Zone selections as your Auto Scaling group. You can select different subnets and add subnets from additional Availability Zones.

vpc-0-0b71a9d9deea470

We will create a application load balancer.

Configure advanced options Info

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.

Attach to a new load balancer

Define a new load balancer to create for attachment to this Auto Scaling group.

Load balancer type

Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, visit the [Load Balancing console](#).

Application Load Balancer HTTP, HTTPS

Network Load Balancer TCP, UDP, TLS

Load balancer name

Name cannot be changed after the load balancer is created.

for-project-1

Load balancer scheme

Scheme cannot be changed after the load balancer is created.

Internal

Internet-facing

Network mapping

Your new load balancer will be created using the same VPC and Availability Zone selections as your Auto Scaling group. You can select different subnets and add subnets from additional Availability Zones.

VPC

vpc-0-0b71a9d9deea470

Availability Zones and subnets

You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.

us-east-1a Select a subnet

us-east-1b subnet-0fb310d8014c6ae2e

us-east-1d subnet-08e7dedf2dfa7148e

Screenshot of the AWS CloudFormation Create Auto Scaling Group wizard, Step 3: Configure Auto Scaling Group.

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

Screenshot of the AWS Auto Scaling Group creation wizard Step 4: Configure group size and scaling policies.

Group size - optional

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.

Target tracking scaling policy
Choose a desired outcome and issue 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

Cancel Previous Skip to review Next

Screenshot of the AWS Auto Scaling Group creation wizard Step 5: Add notifications.

Add notifications

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

Add notification

Cancel Previous Skip to review 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 single metric named 'Lambda function'. The chart shows two data series: 'Latency' and 'Throughput'. The 'Latency' series has two data points: one at approximately 10 seconds and another at approximately 15 seconds. The 'Throughput' series has two data points: one at approximately 100 MB/s and another at approximately 150 MB/s. The x-axis represents time, and the y-axis represents metric values.

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

The screenshot shows a web browser window with a single page. The URL in the address bar is 'for-project-1-845613599.us-east-1.elb.amazonaws.com'. The page contains a simple form with two input fields: 'Name:' and 'Email:', both with placeholder text. Below the fields is a green 'Submit' button. The background of the page is white, and the overall layout is clean and minimalist.




```
$email=$_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>
```