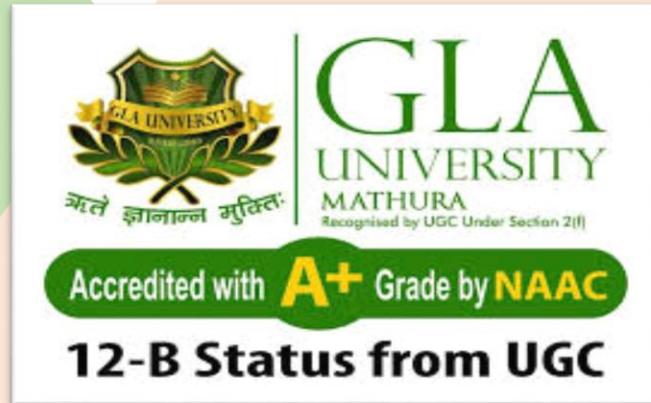


# **GLA UNIVERSITY**



## **Department of Computer Science and Engineering**

### **JOVAC – Project Report**

#### **Project Title**

**Multi-website hosting on Amazon  
Linux Server in Mumbai region**

**Mentor: Raushan Kumar Singh**

## **Submitted By:**

<b>Prakhar Shukla (Leader)</b>	<b>2215001261</b>
<b>Sakshi Tangar</b>	<b>2215001558</b>
<b>Ankur Sinsinwar</b>	<b>2215000273</b>
<b>Anuj Chaturvedi</b>	<b>2215000302</b>
<b>Tanisha Singh</b>	<b>2215001832</b>

## **TABLE OF CONTENTS**

### **1. Introduction**

- Overview of Multi-Website Hosting
- Benefits of Using AWS EC2 and Route 53

### **2. Prerequisites**

- AWS Account Setup
- Required Software and Tools

---

### 3. [Setting Up AWS EC2 Instance](#)

- Launching an EC2 Instance
- Connecting to Your EC2 Instance

### 4. [Configuring the Linux Server](#)

- Updating and Securing Your Server
- Installing Necessary Software (Apache, Nginx, etc.)

### 5. [Setting Up Virtual Hosts](#)

- Configuring Apache Virtual Hosts
- Configuring Nginx Server Blocks

### 6. [Domain Management with Route 53](#)

- Setting Up a Hosted Zone
- Configuring DNS Records for Multiple Domains

## 7. Deploying Websites

- Uploading Website Files
- Configuring Website Settings

## 8. Security Best Practices

- Setting Up Firewalls
- Implementing SSL Certificates

## 9. Monitoring and Maintenance

- Setting Up Monitoring Tools
- Regular Maintenance Tasks

## 10. Troubleshooting

- Common Issues and Solutions
- Useful Commands and Logs

# Introduction

## Overview of Multi-Website Hosting: -

Multi-website hosting refers to the practice of hosting more than one website on a single server. This approach can be cost-effective and efficient, especially for small to medium-sized websites. AWS EC2, combined with Route 53, provides a powerful solution for hosting multiple websites with high availability and scalability.

## Benefits of Using AWS EC2 and Route 53: -

- **Scalability:** Easily scale your resources up or down based on demand.
- **Cost-Effective:** Pay only for the resources you use.
- **Flexibility:** Full control over your server environment.
- **Reliability:** High availability and redundancy with AWS infrastructure.

## **Required Software and Tools:**

### **AWS Services:**

- **EC2 (Elastic Compute Cloud):**
  - Create and manage Amazon Linux instances.
  - Choose an appropriate instance type based on your workload (e.g., t2.micro for light loads, m5.large for more demanding applications).
  - Set up security groups to control inbound and outbound traffic.
- **Elastic Load Balancing (ELB):**
  - Distribute traffic across multiple EC2 instances to ensure high availability and fault tolerance.
- **Auto Scaling:**
  - Automatically scale your instances up or down based on demand to optimize costs and performance.
- **Amazon RDS (Relational Database Service):**
  - Managed database service that supports MySQL, PostgreSQL, MariaDB, and more.
  - Simplifies database management tasks like backups, patching, and scaling.

- **Amazon S3 (Simple Storage Service):**

- Store static assets, backups, or other data that need to be accessed by your applications.

- **Amazon CloudFront:**

- Content Delivery Network (CDN) to deliver your content with low latency.

- **Amazon Route 53:**

- Domain Name System (DNS) web service to route end users to your applications.

- **Amazon CloudWatch:**

- Monitor your AWS resources and applications in real-time.
- Set up alarms and logs to track performance and errors.

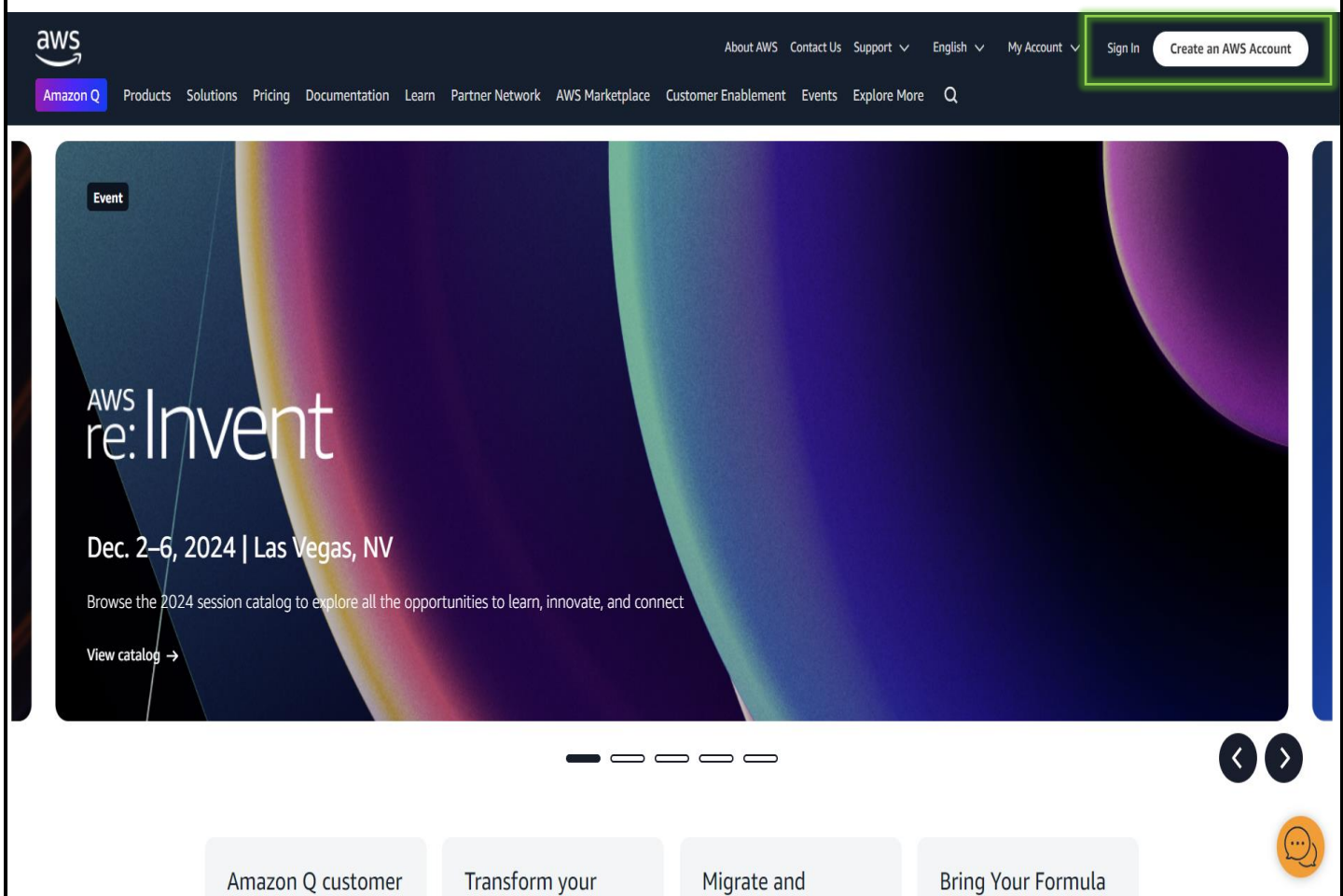
- **AWS IAM (Identity and Access Management):**

- Manage access to your AWS resources securely.
- Create users, groups, and roles with the necessary permissions.

# Prerequisites

## AWS Account Setup: -

Before you can start hosting websites on AWS, you need to set up an AWS account. Visit the [AWS website](#) and follow the sign-up process. Ensure you have a valid payment method and complete the necessary identity verification steps.

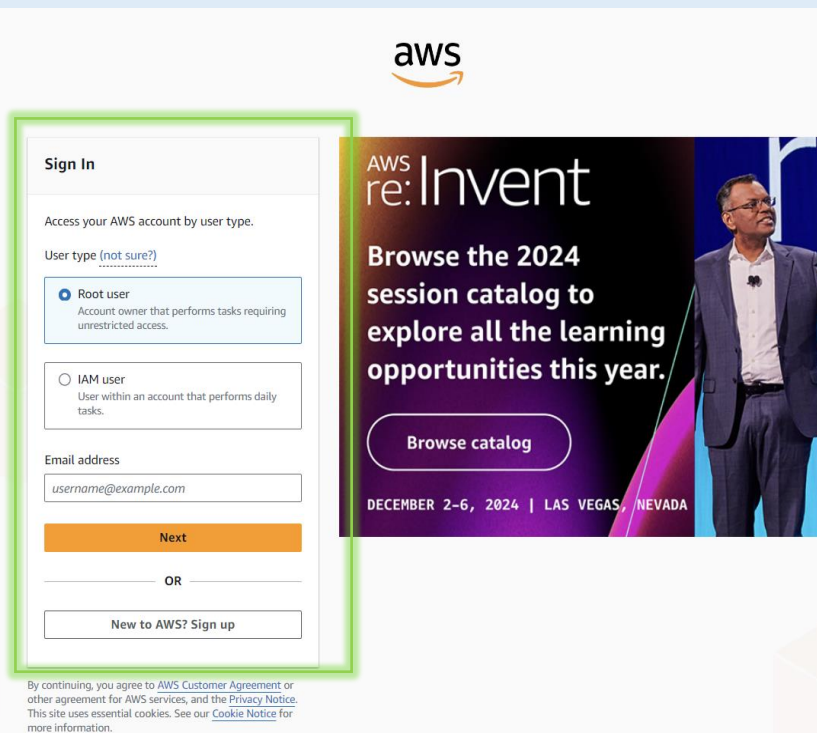




# Setting Up AWS EC2 Instance

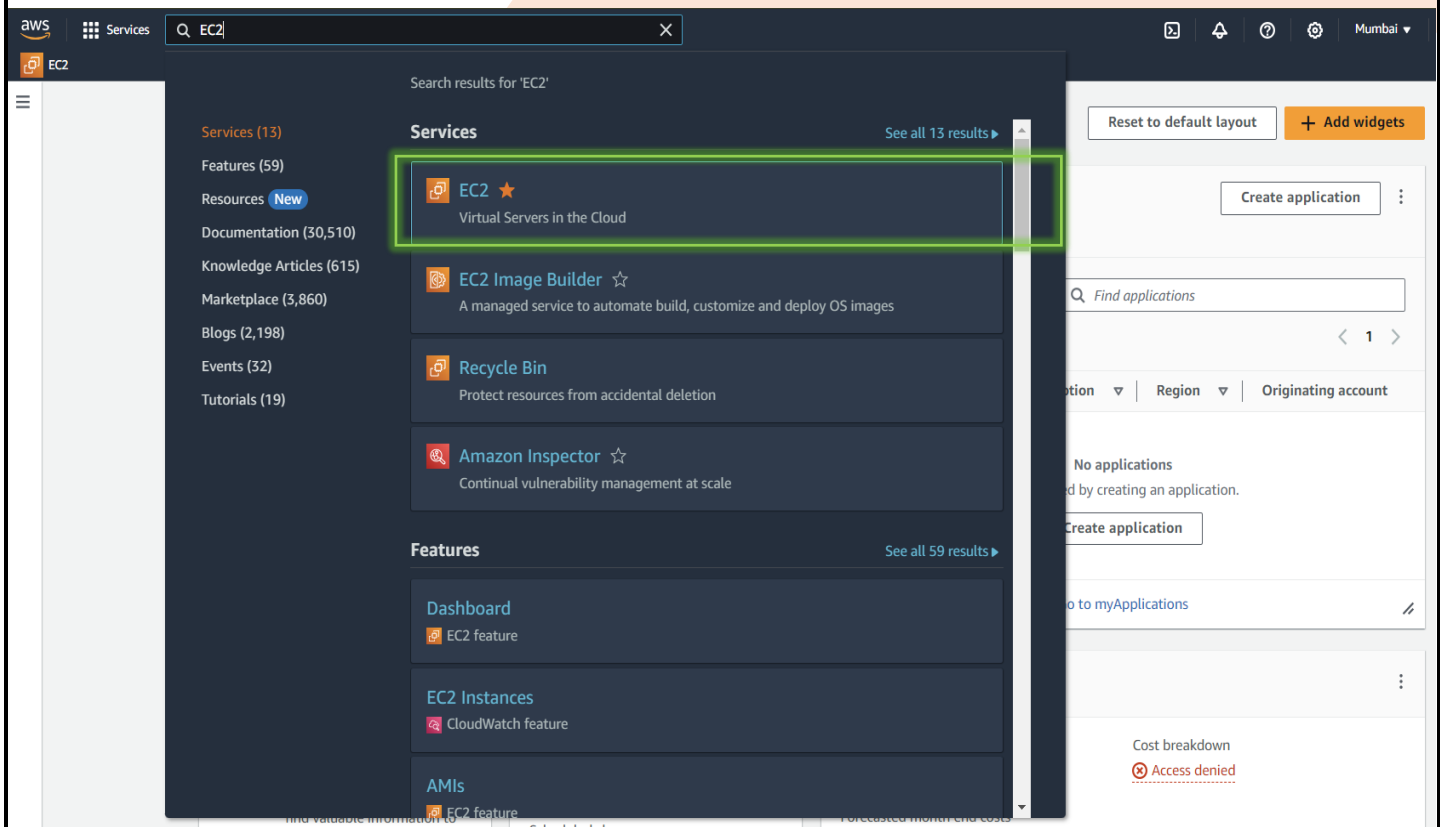
## Launching an EC2 Instance:

1. Log in to the [AWS Management Console](#).



[Sign in as root user or with IAM to work in a Group](#)

## 2. Navigate to the EC2 Dashboard.



Go ahead and click the first link to open the EC2 dashboard

### 3. Click on "Launch Instance."

The screenshot displays the AWS Management Console's EC2 Dashboard for the Asia Pacific (Mumbai) Region. The left-hand navigation pane includes sections for 'Instances', 'Images', 'Elastic Block Store', and 'Network & Security'. The main content area is divided into several panels. The 'Resources' panel at the top shows a summary of EC2 resources. Below it, the 'Launch instance' panel features a prominent orange 'Launch instance' button, which is highlighted with a green rectangular box. To the right of this button is a 'Migrate a server' link. The 'Service health' panel on the right indicates that the AWS service is operating normally. At the bottom, the 'Instance alarms' panel shows that there are no active alarms.

**EC2 Dashboard** X

EC2 Global View

Events

▼ **Instances**

- Instances
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Capacity Reservations

▼ **Images**

- AMIs
- AMI Catalog

▼ **Elastic Block Store**

- Volumes
- Snapshots
- Lifecycle Manager

▼ **Network & Security**

- Security Groups

**Resources** [EC2 Global View](#)

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

Instances (running)	0	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	1	Key pairs	7
Load balancers	0	Placement groups	0	Security groups	3
Snapshots	0	Volumes	1		

**Launch instance**  
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

**Launch instance** ▼ [Migrate a server](#)

Note: Your instances will launch in the Asia Pacific (Mumbai) Region

**Service health** [AWS Health Dashboard](#)

Region  
Asia Pacific (Mumbai)

Status  
✔ This service is operating normally.

**Zones**

Zone name	Zone ID
-----------	---------

**Instance alarms** [View in CloudWatch](#)

⚠ 0 in alarm    ✔ 0 OK    ⌚ 0 insufficient dat

Click the Launch instance button to launch the instance

## 4. Choose an Amazon Machine Image (AMI).

### For this guide, we will use Amazon Linux.

[EC2](#) > [Instances](#) > Launch an instance

### Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

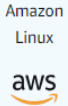
#### Name and tags Info

[Add additional tags](#)


#### ▼ 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




Amazon Linux




macOS




Ubuntu




Windows



Red Hat



SUSE Linux



Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

#### ▼ Summary

Number of instances Info

Software Image (AMI)

Amazon Linux 2023 AMI 2023.5.2...[read more](#)  
ami-068e0f1a600cd311c

Virtual server type (instance type)



t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

 **Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet. 

Cancel

Launch instance

[Review commands](#)

Giving the 'AWS-Project' name to our instance  
and choosing Amazon Linux 2023 AMI

## 5. Select an instance type (e.g., t2.micro for the free tier).

### ▼ Instance type [Info](#) | [Get advice](#)

#### Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Linux base pricing: 0.0124 USD per Hour

On-Demand Windows base pricing: 0.017 USD per Hour

On-Demand RHEL base pricing: 0.0268 USD per Hour

On-Demand SUSE base pricing: 0.0124 USD per Hour

☐ All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

## 6. Configure instance details, including network settings and storage.

### ▼ Network settings [Info](#)

Edit

Network [Info](#)

vpc-054f91f3eca347d98

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group

☐ Select existing security group



We'll create a new security group called 'launch-wizard-2' with the following rules:

☒ Allow SSH traffic from  
Helps you connect to your instance

Anywhere  
0.0.0.0/0

☒ Allow HTTPS traffic from the internet  
To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet  
To set up an endpoint, for example when creating a web server

 Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. 

## 7. Add a key pair for SSH access.

### Create key pair [Info](#)

#### Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type [Info](#)

☒ RSA

☐ ED25519

Private key file format

☒ .pem

For use with OpenSSH

☐ .ppk

For use with PuTTY

Tags - *optional*

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel

Create key pair

Name your Key pair and chouse key format  
(.ppm for Linux)

## 8. Review and launch the instance.

[EC2](#) > [Instances](#) > Launch an instance

### Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

#### Name and tags [Info](#)

Name

e.g. My Web Server

[Add additional tags](#)

#### ▼ 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

Recents

**Quick Start**

Amazon Linux  
aws


macOS  
Mac

Ubuntu  
ubuntu®

Windows  
Microsoft

Red Hat  
Red Hat

SUSE L  
SUS

  
Browse more AMIs  
Including AMIs from AWS, Marketplace and the Community

#### Amazon Machine Image (AMI)

Amazon Linux 2023 AMI  
ami-0a4408457f9a03be3 (64-bit (x86), uefi-preferred) / ami-0741e326f07d8dfa8 (64-bit (Arm), uefi)  
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 2023.5.2...[read more](#)  
ami-0a4408457f9a03be3

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

**Launch instance**

[Review commands](#)

Click the Launch instance and wait until the instance is ready

## Connecting To EC-2 Instance:

### Select your instance and connect.

Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive) All states

Instance state = running Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
AWS-Project	i-0192fdb978603112f	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1a	ec2-3-110-142-190.ap-...	3.110.142.190	-

EC2 > Instances > i-02377ed647bad687c > Connect to instance

### Connect to instance Info

Connect to your instance i-02377ed647bad687c (AWS-Project) using any of these options

EC2 Instance Connect

Session Manager

SSH client

EC2 serial console

Instance ID

i-02377ed647bad687c (AWS-Project)

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

3.108.51.191

Username

Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.

ec2-user

**Note:** In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel

Connect



## Starting Service

## Step 4 : Deploy Website Content

- Making directories [inside 'var/www/html' directory]

```
[root@ip-172-31-6-20 html]# mkdir vidixer.com
```

```
[root@ip-172-31-6-20 html]# mkdir cashify.com
```

- Create Index.html as base webpage [for testing].

```
[root@ip-172-31-6-20 html]# vi index.html
```

```
<h1> hii </h1>
<hr>

<a href="/vidixer.com"> Vidixer.com </a>
<hr>

<a href="/cashify.com"> Cashify.com </a>
~
~
~
~
~
~
~
```

- Creating index.html for both cashify.com and vidixer.com

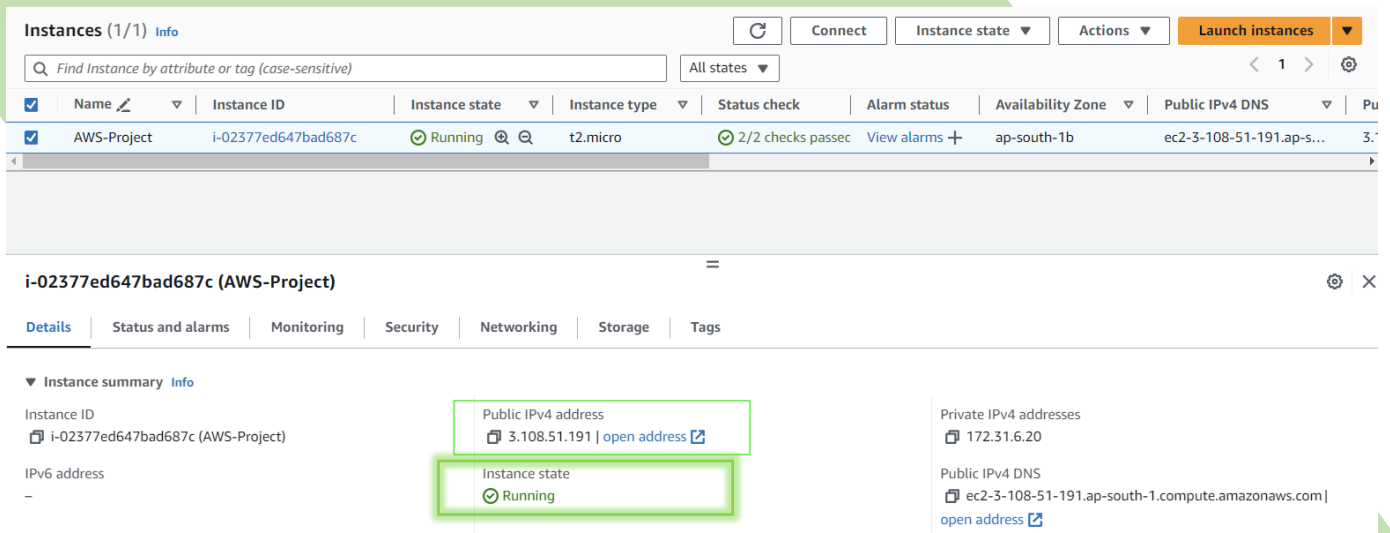
```
[root@ip-172-31-6-20 html]# ls -l
total 4
drwxr-xr-x. 3 root root 35 Jul 23 09:35 cashify.com
-rw-r--r--. 1 root root 115 Jul 23 09:47 index.html
drwxr-xr-x. 2 root root 58 Jul 23 08:34 vidixer.com
[root@ip-172-31-6-20 html]#
```

```
[root@ip-172-31-6-20 html]# cd vidixer.com
[root@ip-172-31-6-20 vidixer.com]# vi index.html
```

```
[root@ip-172-31-6-20 html]# cd cashify.com
[root@ip-172-31-6-20 cashify.com]# vi index.html
```

## Step 5. Test your website.

- Copy instance's public IPv4 address and past in your browser.



The screenshot displays the AWS Management Console interface for an EC2 instance. At the top, there's a search bar and a table of instances. The instance 'i-02377ed647bad687c' is selected, showing its details. The 'Instance summary' section is expanded, showing the instance ID, public IPv4 address (3.108.51.191), and the instance state (Running). The public IPv4 address and the 'Running' state are highlighted with green boxes.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
AWS-Project	i-02377ed647bad687c	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1b	ec2-3-108-51-191.ap-s...

**i-02377ed647bad687c (AWS-Project)**

**Instance summary**

Instance ID  
i-02377ed647bad687c (AWS-Project)

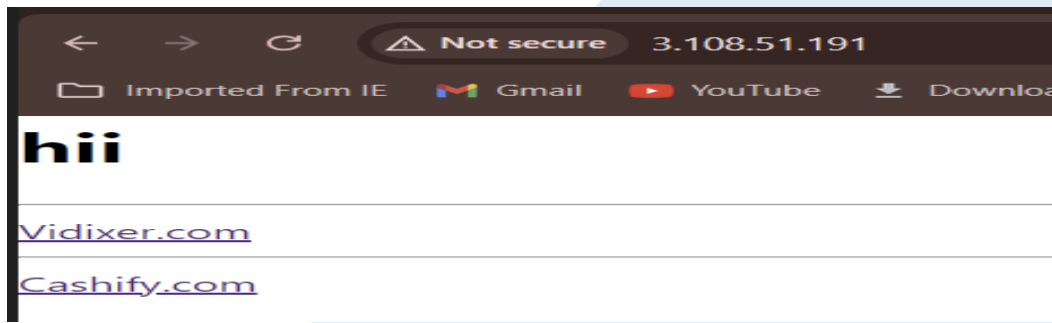
Public IPv4 address  
3.108.51.191 | [open address](#)

Instance state  
Running

Private IPv4 addresses  
172.31.6.20

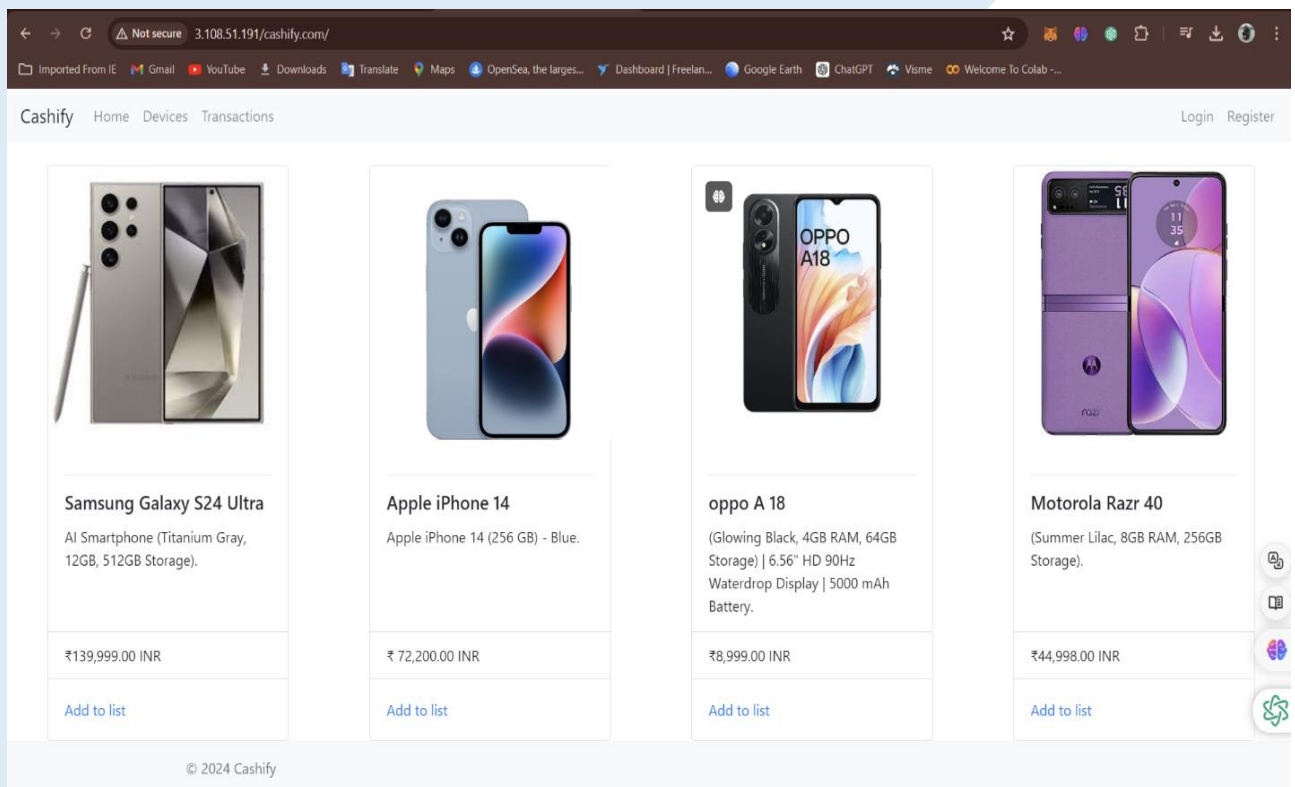
Public IPv4 DNS  
ec2-3-108-51-191.ap-south-1.compute.amazonaws.com | [open address](#)

- Now you will See the base index.html [for testing]



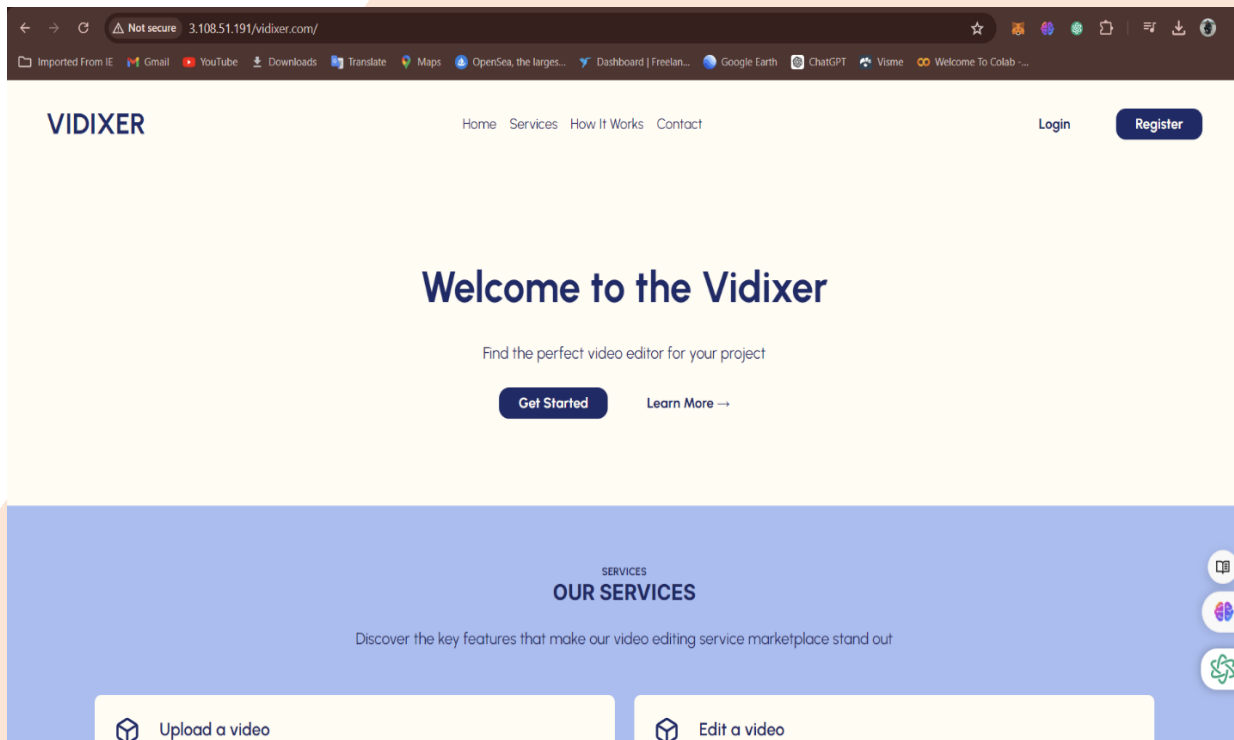
Go ahead and visit each website

Our First Website :  
**cashify.com**

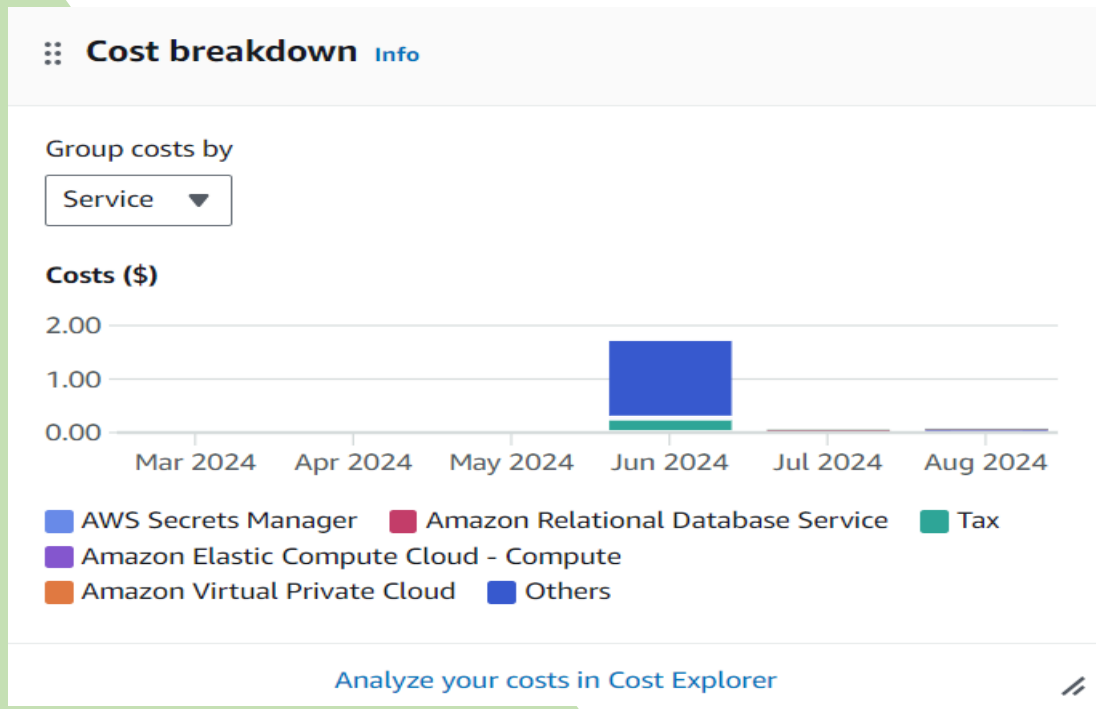


## Second Website:



### vidixer.com



## Cost Analyses



**AWS Provide service as billing and cost management to see how much amount of bill is generated in the services we have used.**

AWS bill summary <a href="#">Info</a>			
Total charges and payment information			
Account ID 992382698575	Billing period <a href="#">Info</a> June 1 - June 30, 2024	Bill status <a href="#">Info</a>  Issued 07/03/2024	
Service provider Amazon Web Services India Private Limited		Total in USD USD 1.75	
Grand total:			USD 1.75
<a href="#">▶ Payment information</a> <a href="#">Info</a>			

**This is all about the cost generated in the month of June as working with AWS Account**

Amazon Web Services India Private Limited charges by service <a href="#">Info</a>			<a href="#">Expand all</a>	<a href="#">Settings</a>
Total active services		Total pre-tax service charges in USD		
5		USD 1.48		
<input type="text" value="Filter by service name or region name"/>		< 1 >		
Description	Usage Quantity	Amount in USD		
Data Transfer		USD 1.48		
CloudWatch		USD 0.00		
Elastic Compute Cloud		USD 0.00		
Key Management Service		USD 0.00		
Virtual Private Cloud		USD 0.00		
<b>Total tax</b>		USD 0.27		

**It not only shows cost but also shows which service has used how much amount of cost each service has costed in there use.**

Payments due summary

Info

Contact us

Total amount due

INR 0.00

Total amount past due

INR 0.00

Total amount processing

INR 4.19

Total amount scheduled

INR 0.00

Payments due (1)

Download

Cancel payment

Complete payment

Filter due payments by text, property or value

Filter due payments

< 1 >

Issued date

Invoice ID

Type

Due date

Status

Currency

Invoice amount

Balance due

August 2, 2024

1792607273

Invoice

August 2, 2024

Payment processing

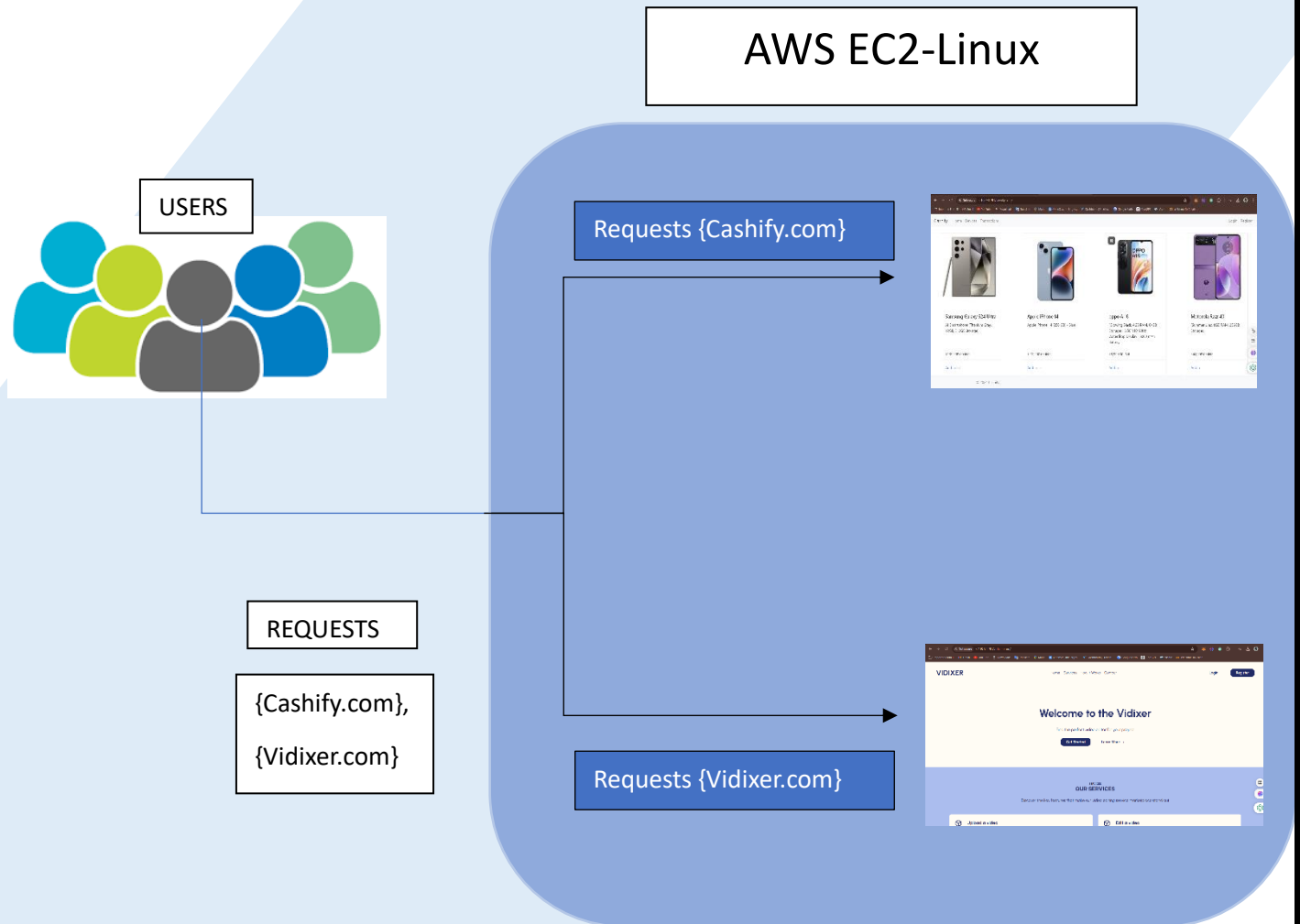
INR

4.19

4.19

**You get your payment information also in**  
**AWS**

## System Design And Architecture :





# Conclusion

**The successful implementation of a multi-website hosting environment on AWS EC2 has demonstrated the platform's flexibility, scalability, and cost-effectiveness.** By leveraging EC2 instances, we were able to efficiently host multiple websites with varying traffic patterns, ensuring optimal performance and resource utilization.

## Key achievements include:

- Effective configuration of EC2 instances for optimal resource allocation.
- Successful deployment and management of multiple websites.
- Implementation of robust security measures to protect sensitive data and prevent unauthorized access.
- Optimization of infrastructure costs through [Cost optimization strategies, e.g., reserved instances, spot instances].

## Lessons learned during the project highlight the importance of:

- Careful planning and resource estimation for optimal performance.
- Regular security audits and updates to mitigate potential threats.
- Continuous monitoring and optimization of infrastructure to ensure cost-efficiency.