**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**

|  |  |
| --- | --- |
| **Prakhar Shukla (Leader)** | **2215001261** |
| **Sakshi Tangar** | **2215001558** |
| **Ankur Sinsinwar** | **2215000273** |
| **Anuj Chaturvedi** | **2215000302** |
| **Tanisha Singh** | **2215001832** |

**Submitted By:**

**TABLE OF CONTENTS**

1. [**Introduction**](#Introduction)

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

1. [**Prerequisites**](#Prerequisites)

* AWS Account Setup
* Required Software and Tools

1. [**Setting Up AWS EC2 Instance**](#SeettingUpEC2Instance)

* Launching an EC2 Instance
* Connecting to Your EC2 Instance

1. **Configuring the Linux Server**

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

1. **Setting Up Virtual Hosts**

* Configuring Apache Virtual Hosts
* Configuring Nginx Server Blocks

1. **Domain Management with Route 53**

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

1. **Deploying Websites**

* Uploading Website Files
* Configuring Website Settings

1. **Security Best Practices**

* Setting Up Firewalls
* Implementing SSL Certificates

1. **Monitoring and Maintenance**

* Setting Up Monitoring Tools
* Regular Maintenance Tasks

1. **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: -**

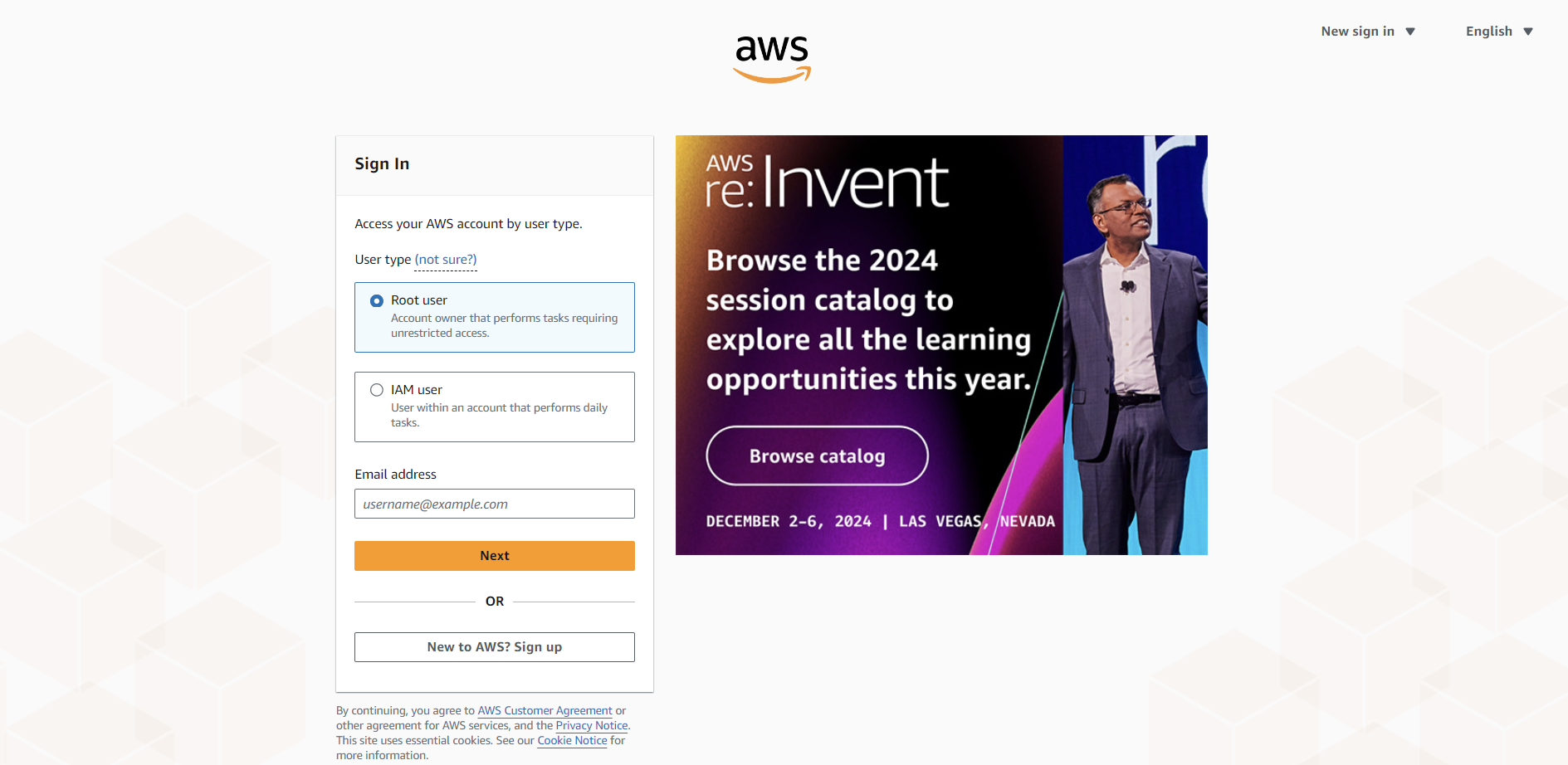
Before you can start hosting websites on AWS, you need to set up an AWS account. Visit the [**AWS website**](https://aws.amazon.com/) and follow the sign-up process. Ensure you have a valid payment method and complete the necessary identity verification steps.



**Setting Up AWS EC2 Instance**

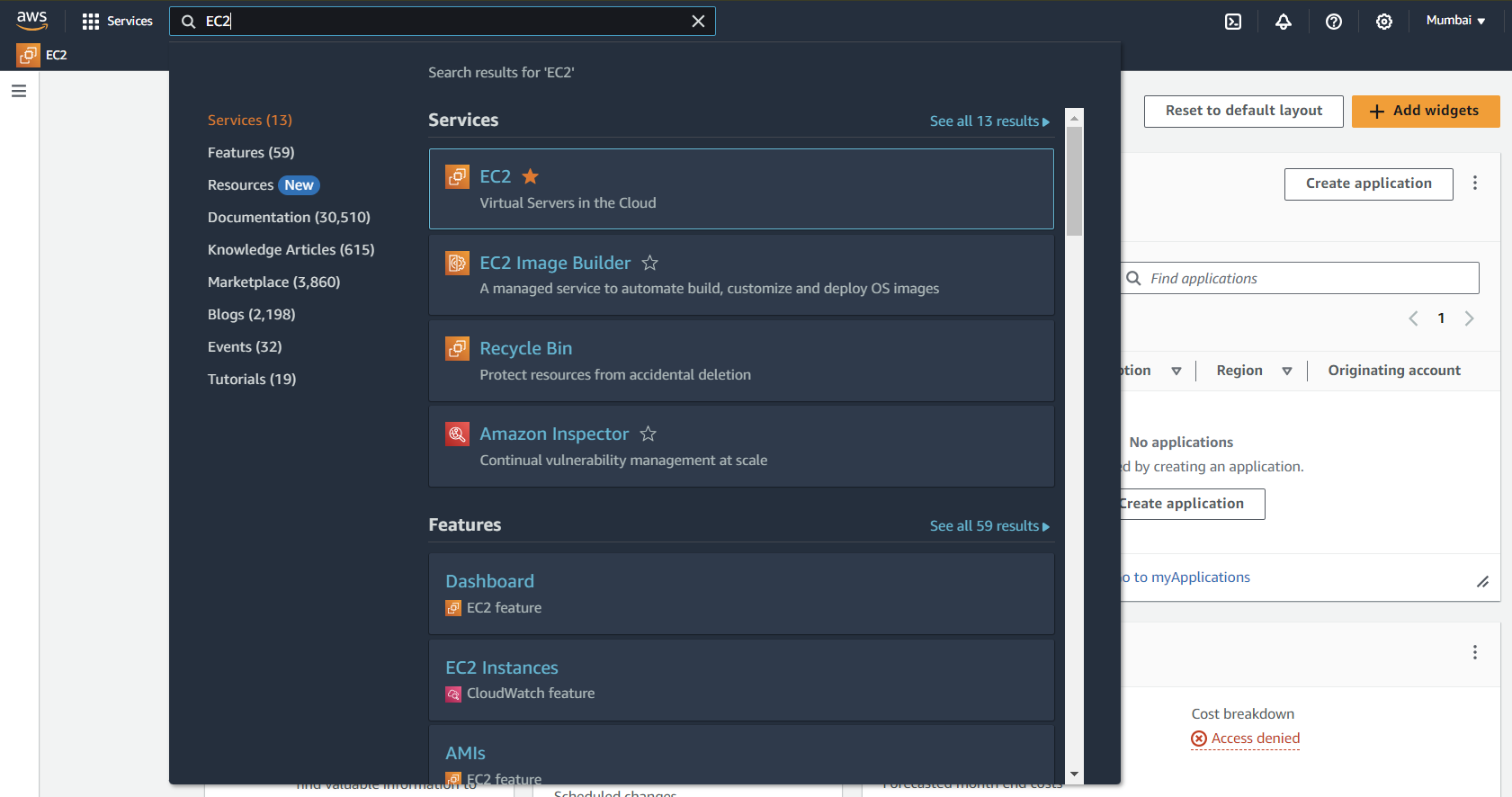
**Launching an EC2 Instance:**

1. Log in to the [AWS Management Console](https://aws.amazon.com/console/).



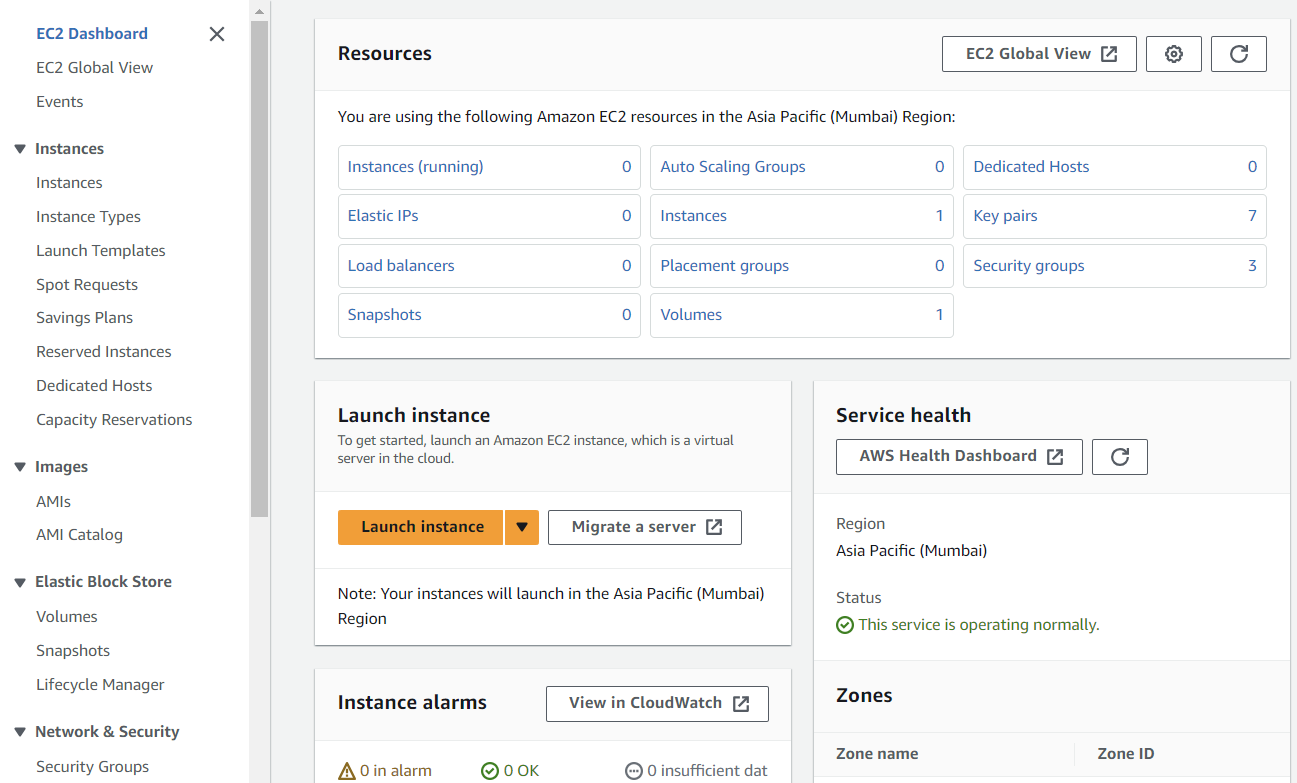
**Sign in as root user or with IAM to work in a Group**

1. **Navigate to the EC2 Dashboard.**

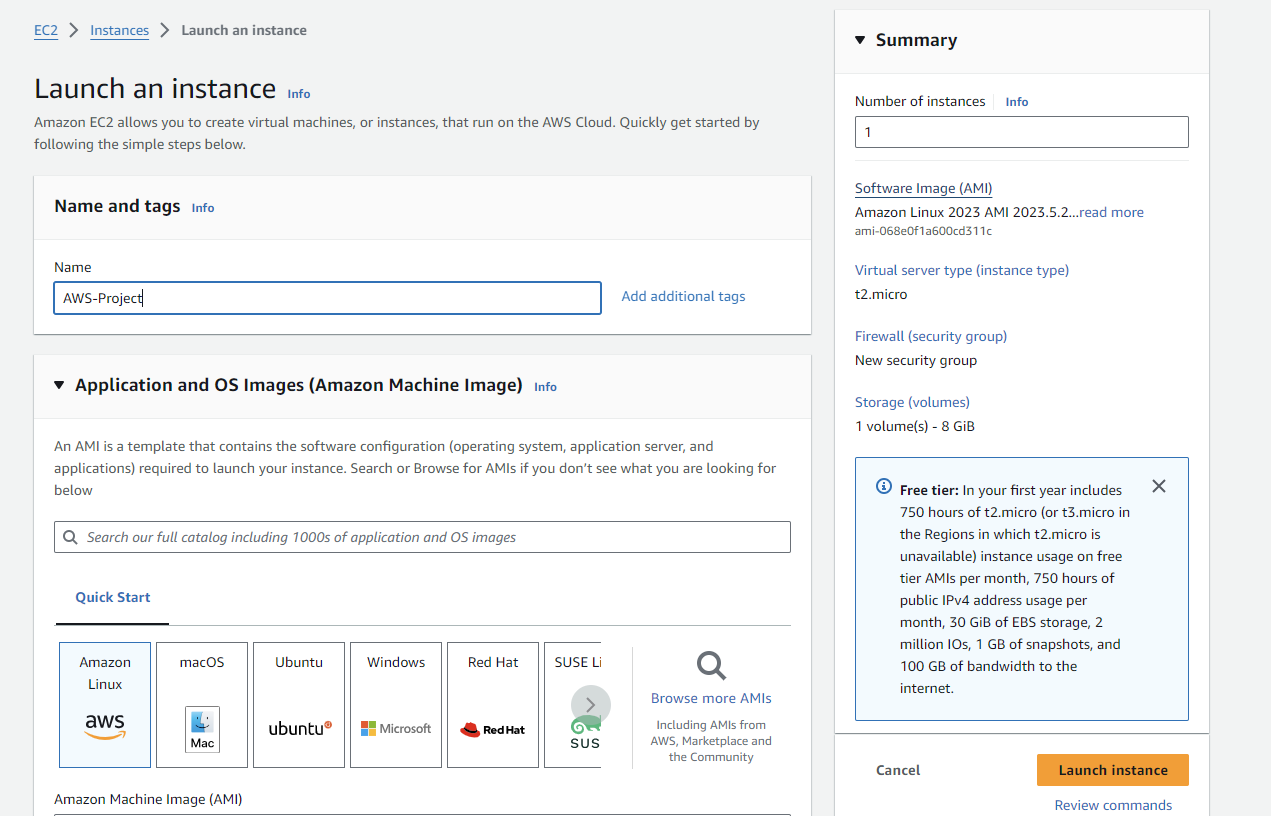


Go ahead and click the first link to open the EC2 dashboard

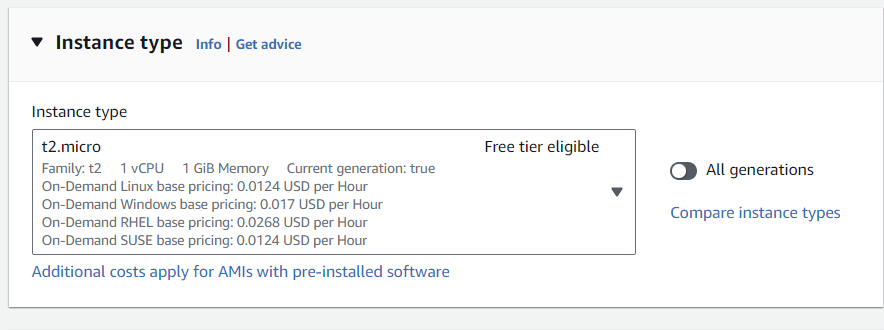
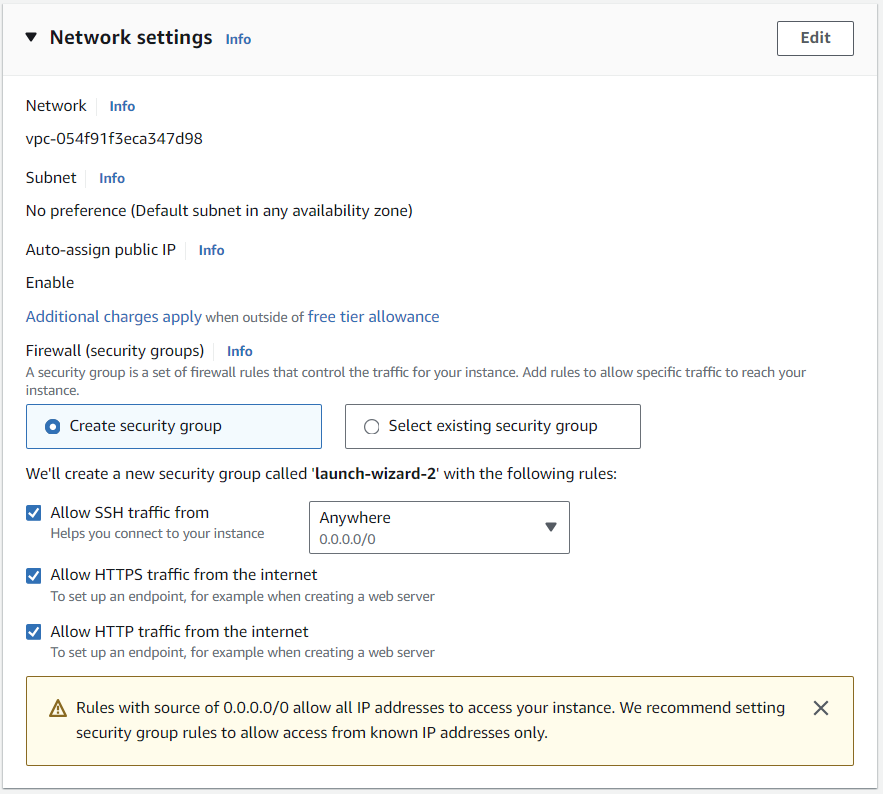
1. **Click on "Launch Instance."**

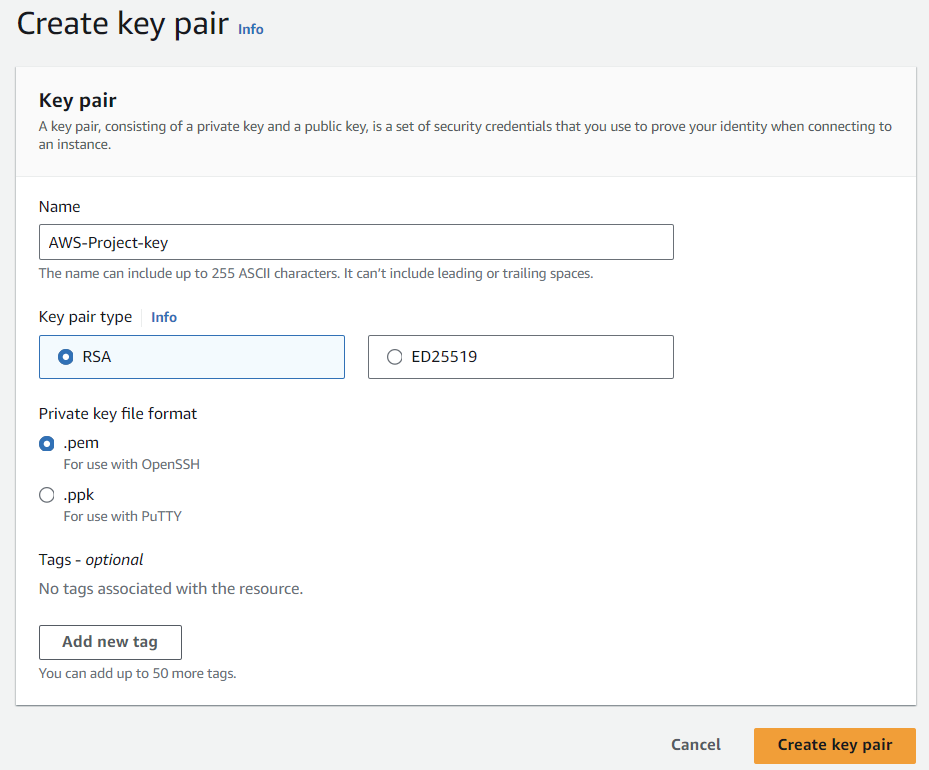


Click the Launch instance button to launch the instance

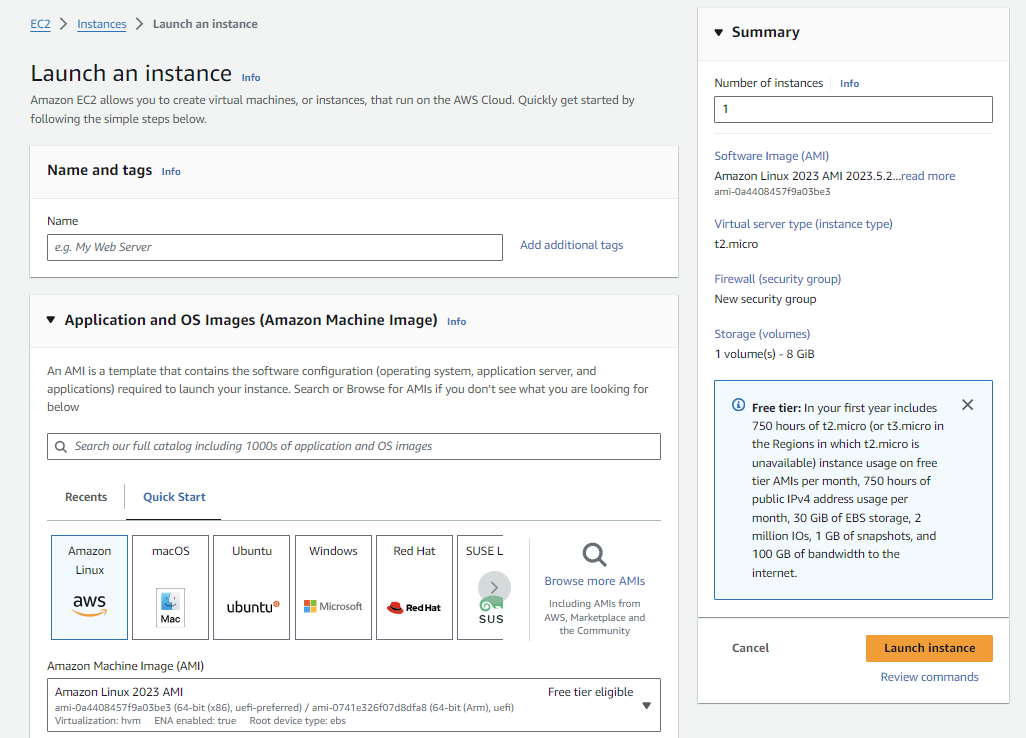
1. **Choose an Amazon Machine Image (AMI). For this guide, we will use Amazon Linux**.

Giving the ‘AWS-Project’ name to our instance and chousing Amazon Linux 2023 AMI

1. **Select an instance type (e.g., t2. micro for the free tier).**
2. **Configure instance details, including network settings and storage.**
3. **Add a key pair for SSH access.**

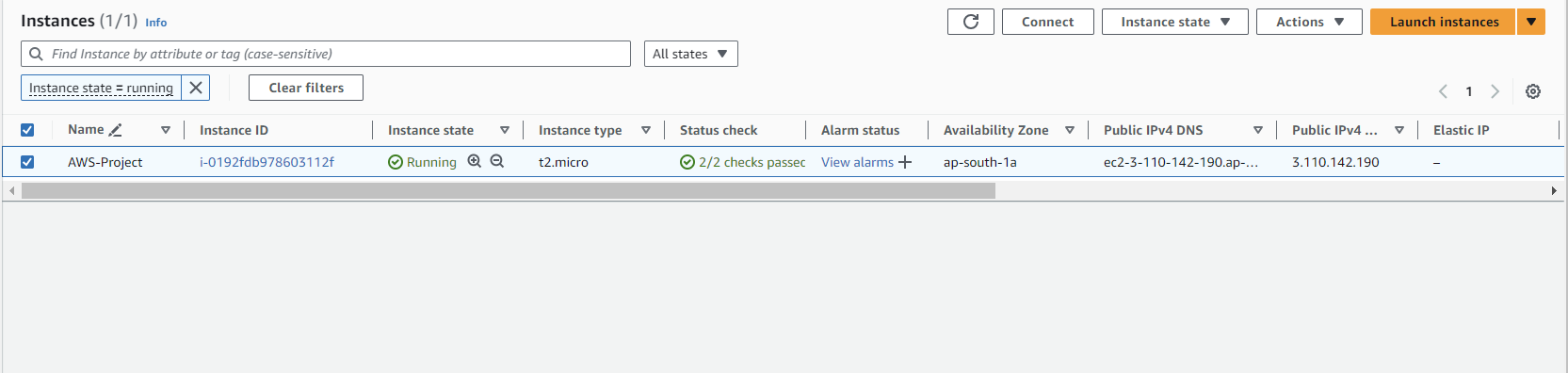
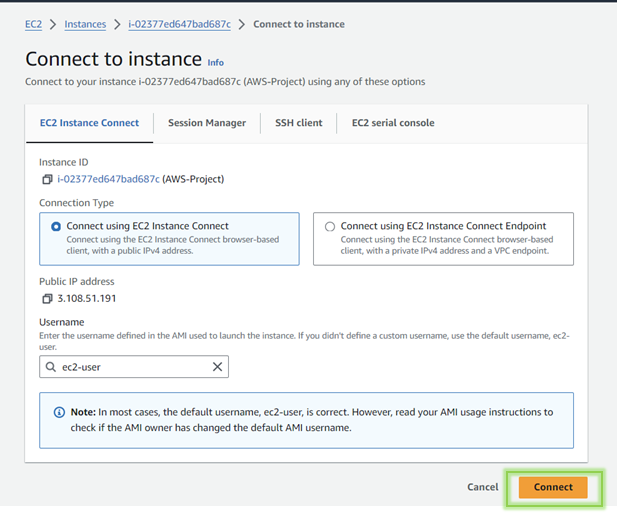


**Name your Key pair and chouse key format (.ppm for Linux)**

1. **Review and launch the instance.**

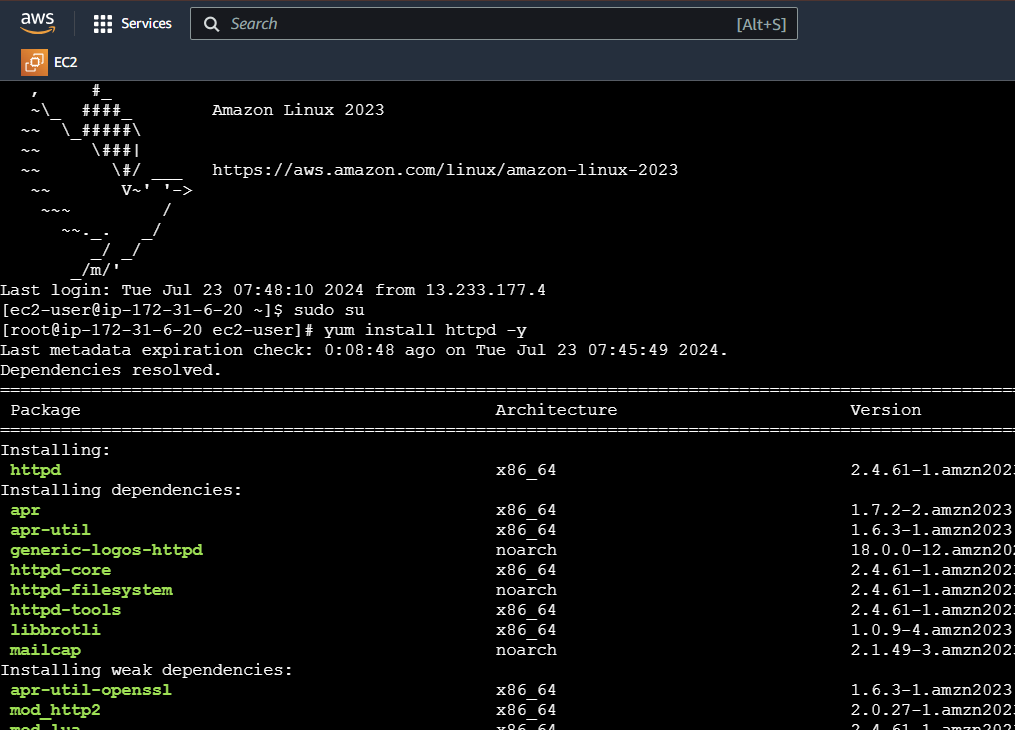
Click the Launch instance and wait until the instance is ready

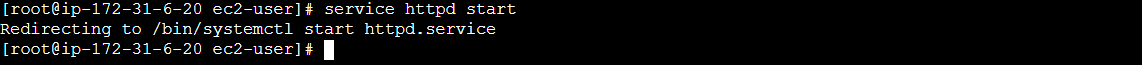
**Connecting To EC-2 Instance:**

**Select your instance and connect.**

**Configuring a Linux Server**

**Step 1: Install Apache**





Starting Service

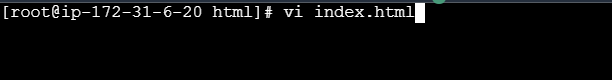
**Step 4 : Deploy Website Content**

* Making directories [inside ‘var/www/html’ directory]



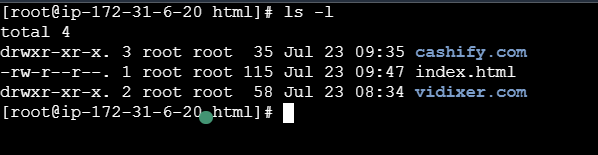


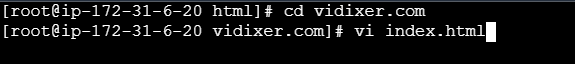
* Create Index.html as base webpage [for testing].





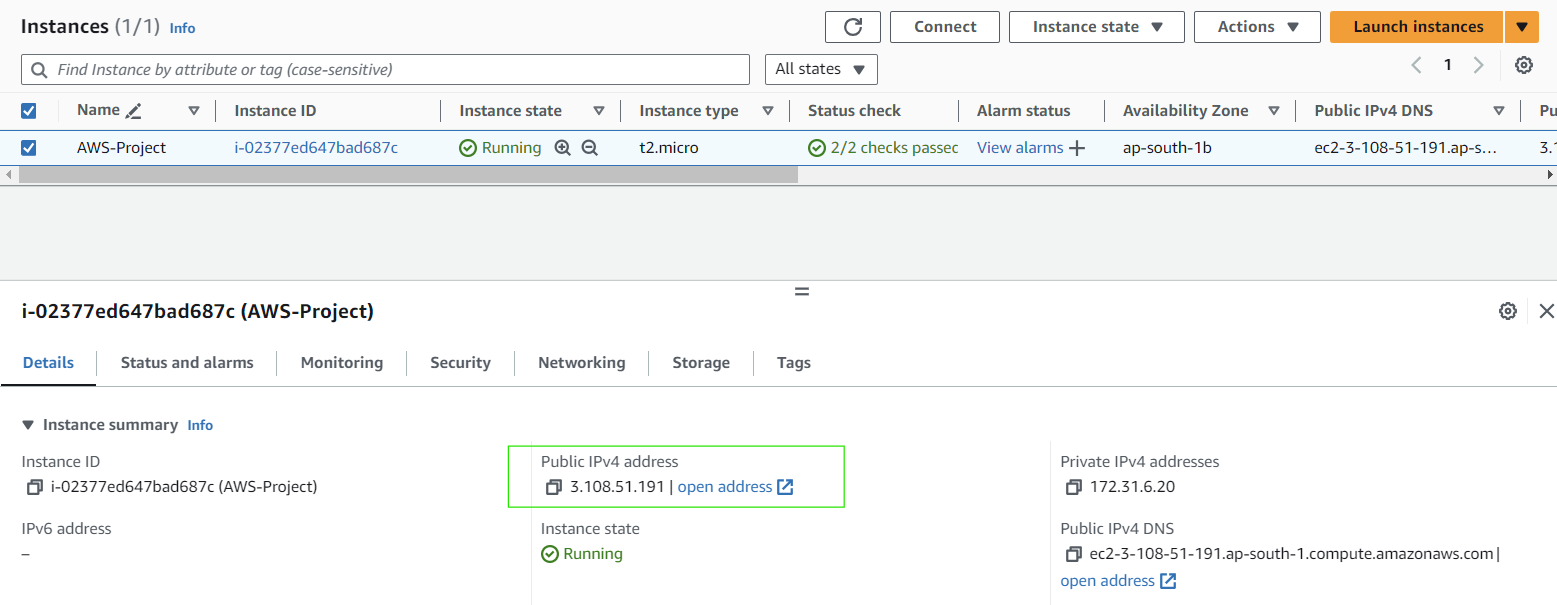
* Creating index.html for both cashify.com and vidixer.com

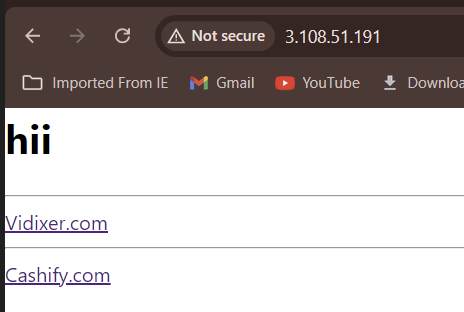






**Step 5. Test your website.**

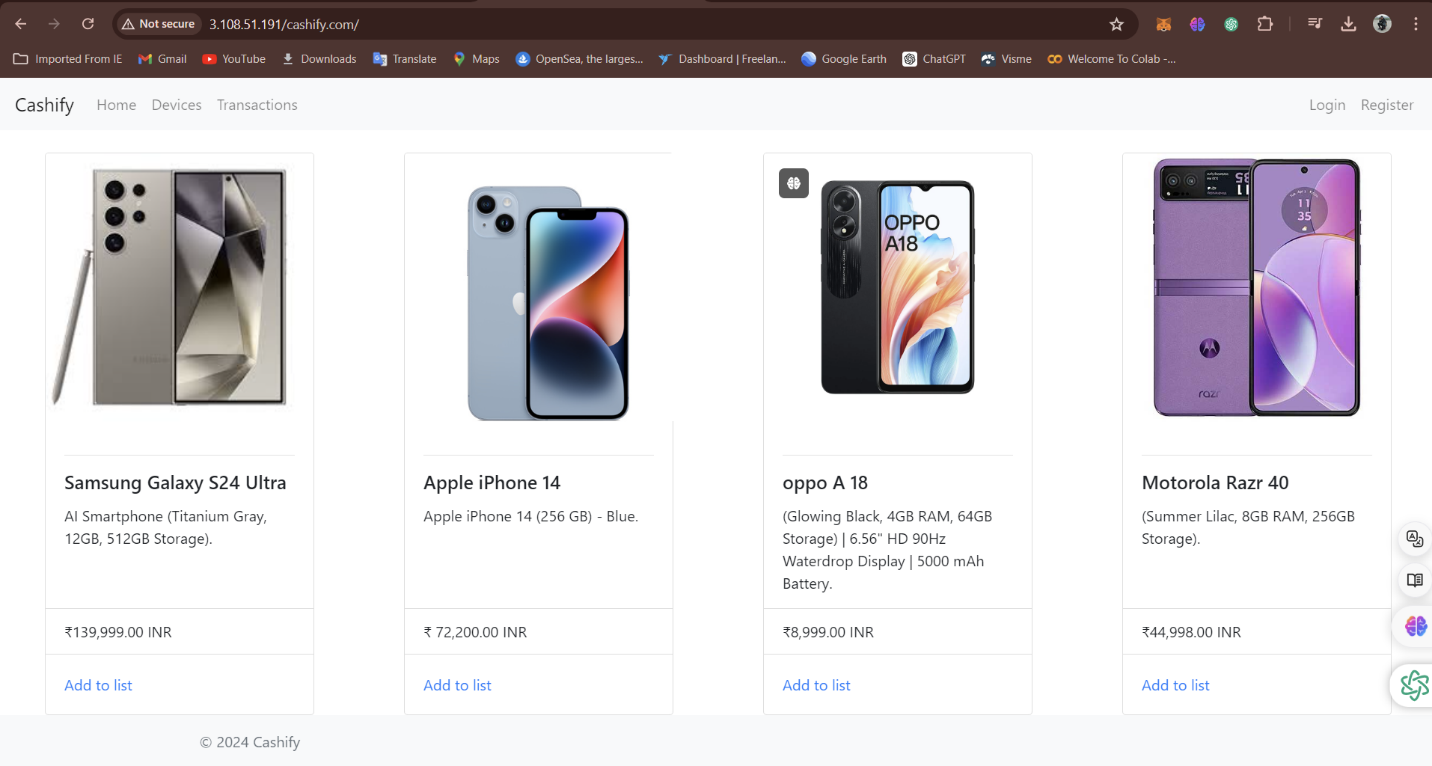
* Copy instance’s public IPv4 address and past in your browser.
* Now you will See the base index.html [for testing]



Go ahead and visit each website

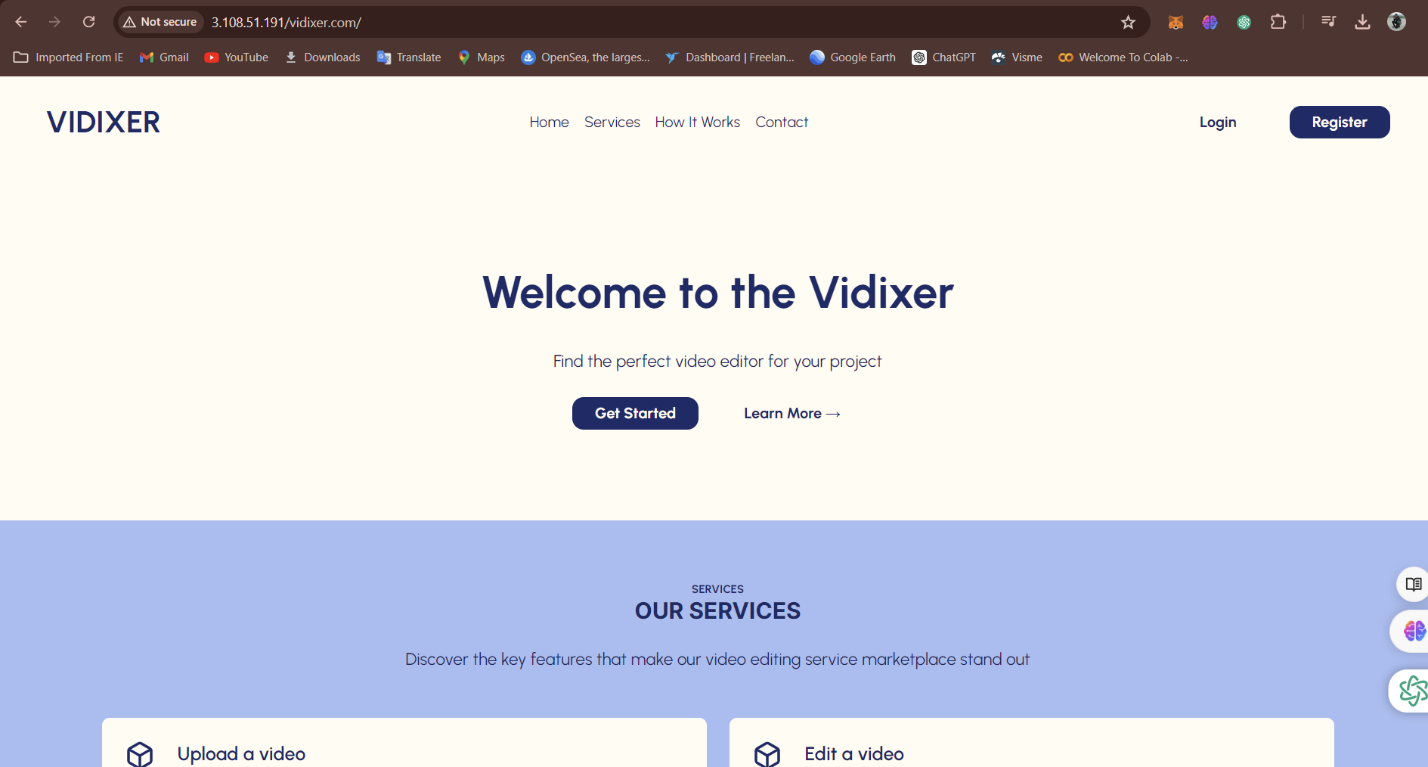
**Our First Website :**

**cashify.com**

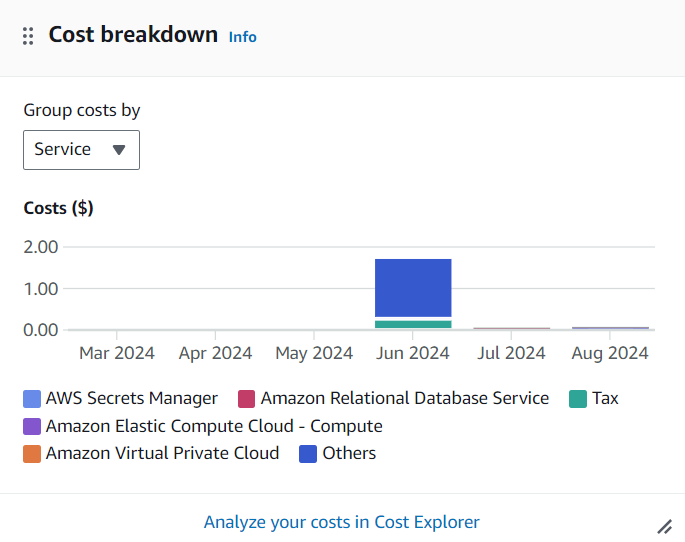


**Second Website:**

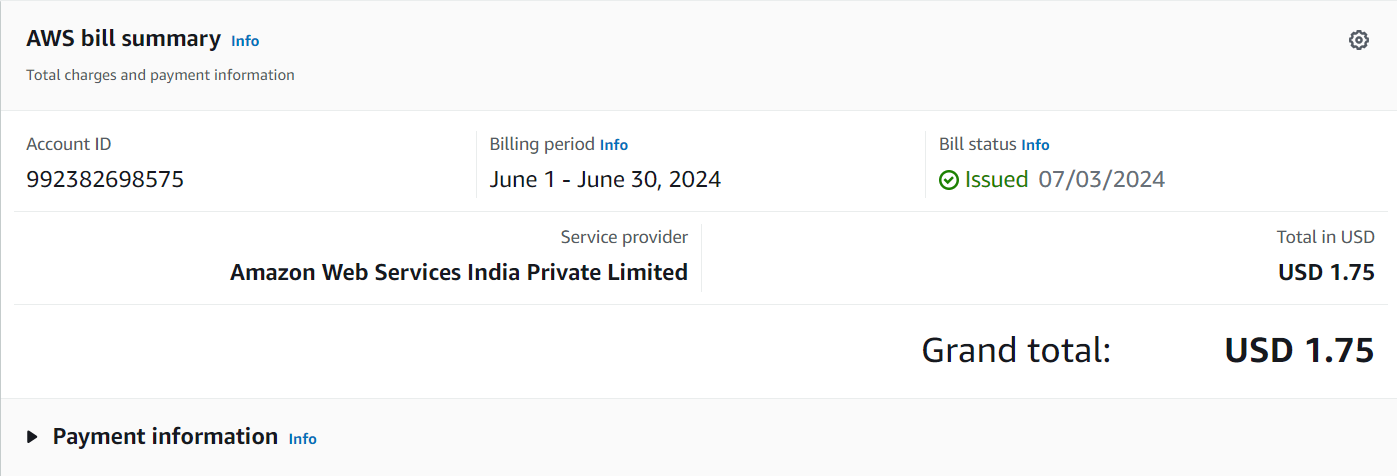
vidixer.com



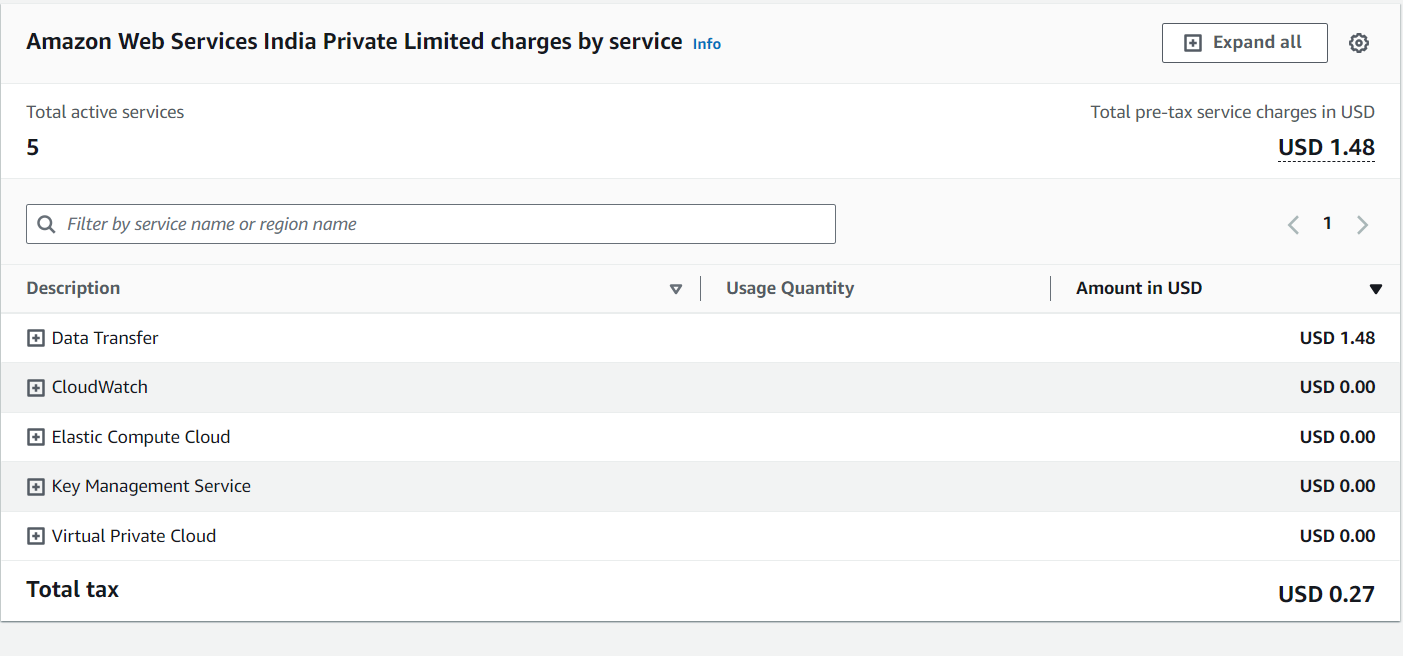
**Cost Analyses**

****

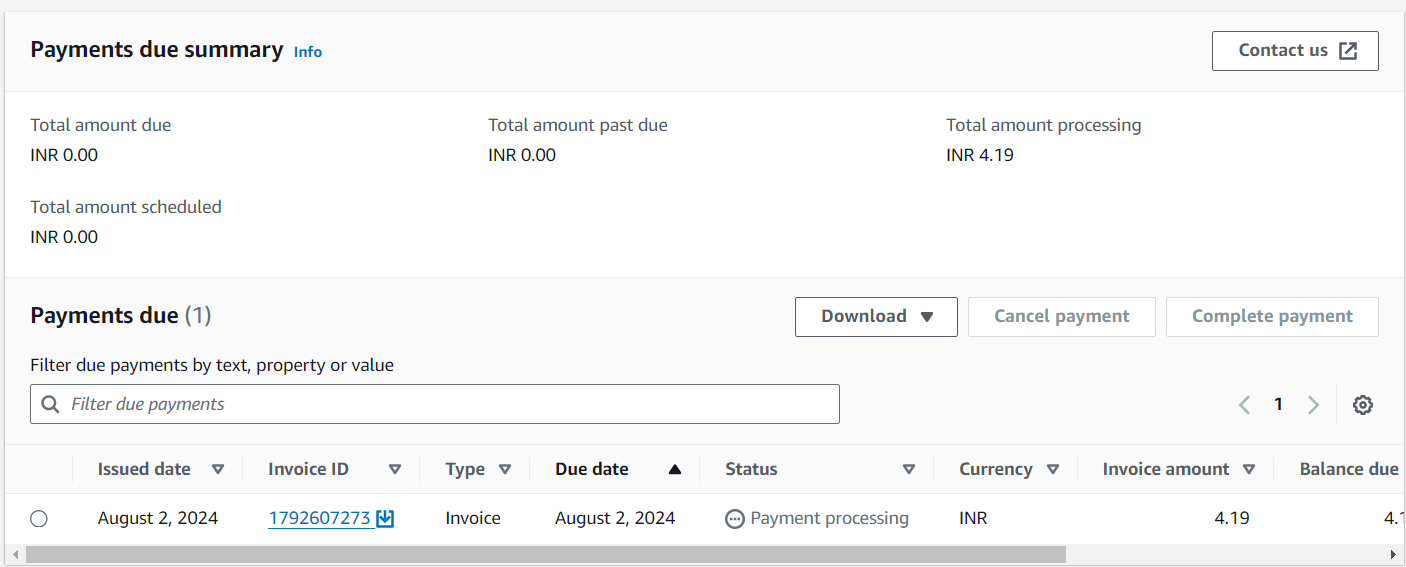
**AWS Provide service as billing and cost management to see how much amount of bill is generated in the services we have used.**

****

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

****

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

****

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

**System Design And Architecture :**

­

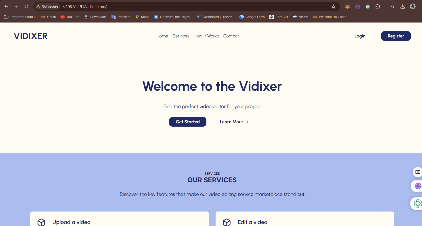
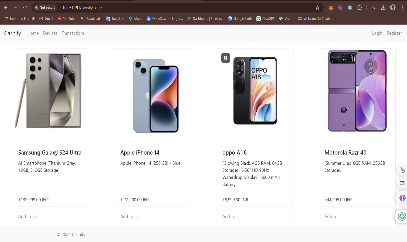
AWS EC2-Linux

REQUESTS

{Cashify.com},

{Vidixer.com}

USERS



Requests {Vidixer.com}

Requests {Cashify.com}

**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.