Fostering Inclusion

Project Done by :

G. Girish kumar

Gmail:521136@student.nitandhra.ac.in

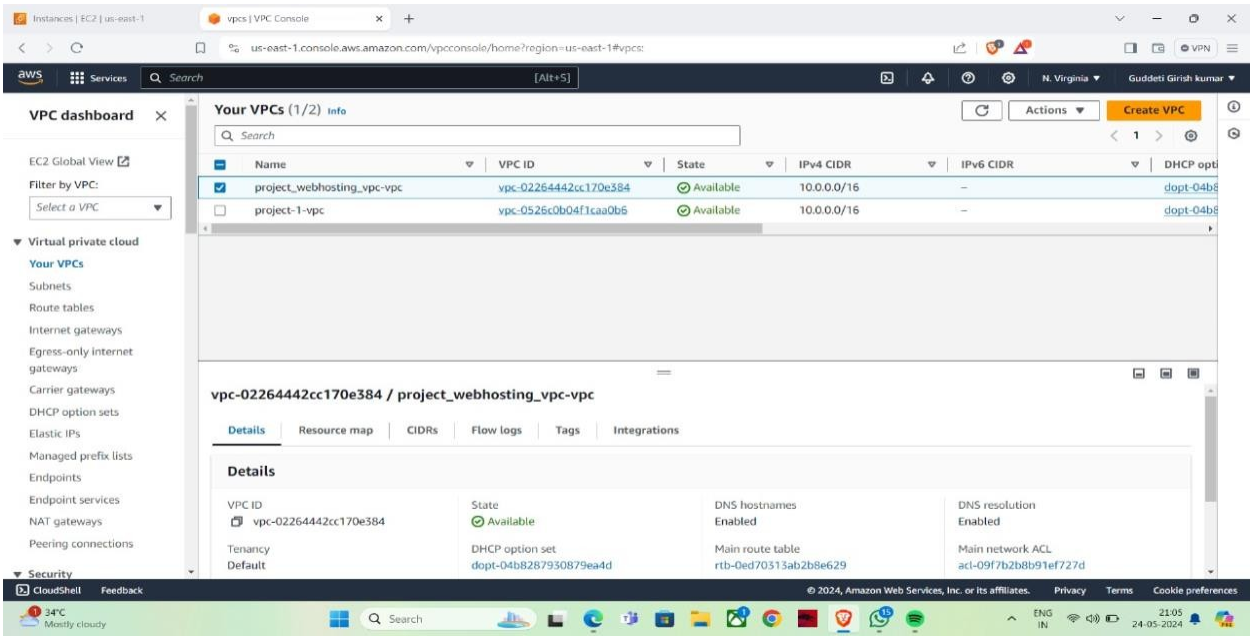
Hosting web application based on traffic High Availability and Auto Scaling for a Web Application using EC2, ELB, ASG and EBS

Step 1: Create a New VPC and Subnet:

- Navigate to VPC Dashboard*:
- Open the AWS Management Console.
- Go to the VPC service.

1. *Create a VPC*:

- Click on "Create VPC".
- Give it a name (e.g., project_webhosting_vpc-vpc).
- Choose an IPv4 CIDR block (e.g., 10.0.0.0/16).
- Leave other settings as default and create the VPC.



2. *Create Subnets:

- Within your VPC, create at least two subnets in different availability zones for high availability.
- Give them names (e.g., Subnet-A and Subnet-B).
- Choose IPv4 CIDR blocks for each (e.g., 10.0.1.0/24 and 10.0.2.0/24).

Step 2: Launch Two EC2 Instances:

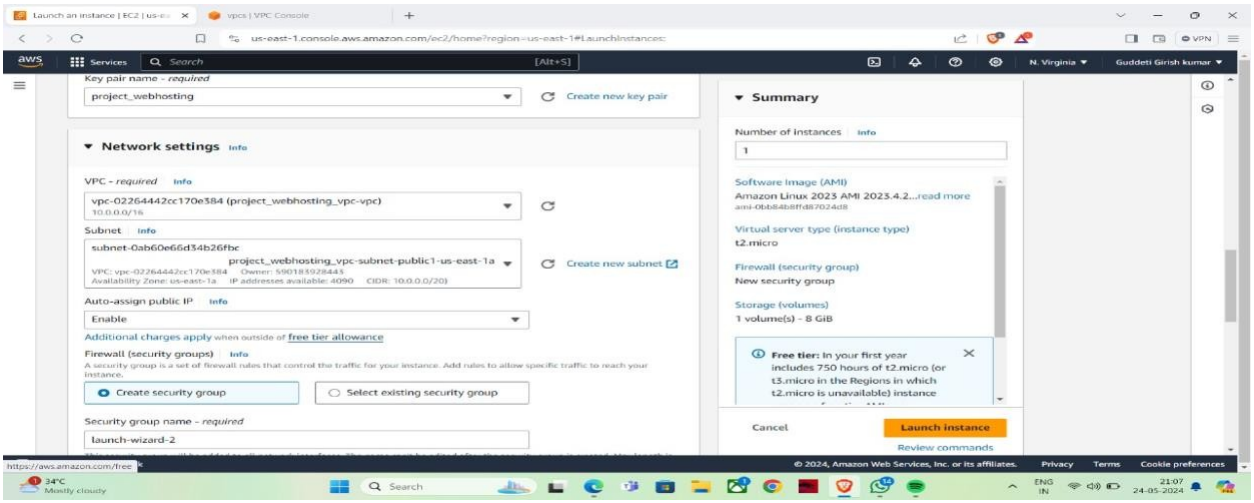
1. Navigate to EC2 Dashboard:

- Open the AWS Management Console.
- Go to the EC2 service.

2. *Launch Instances*:

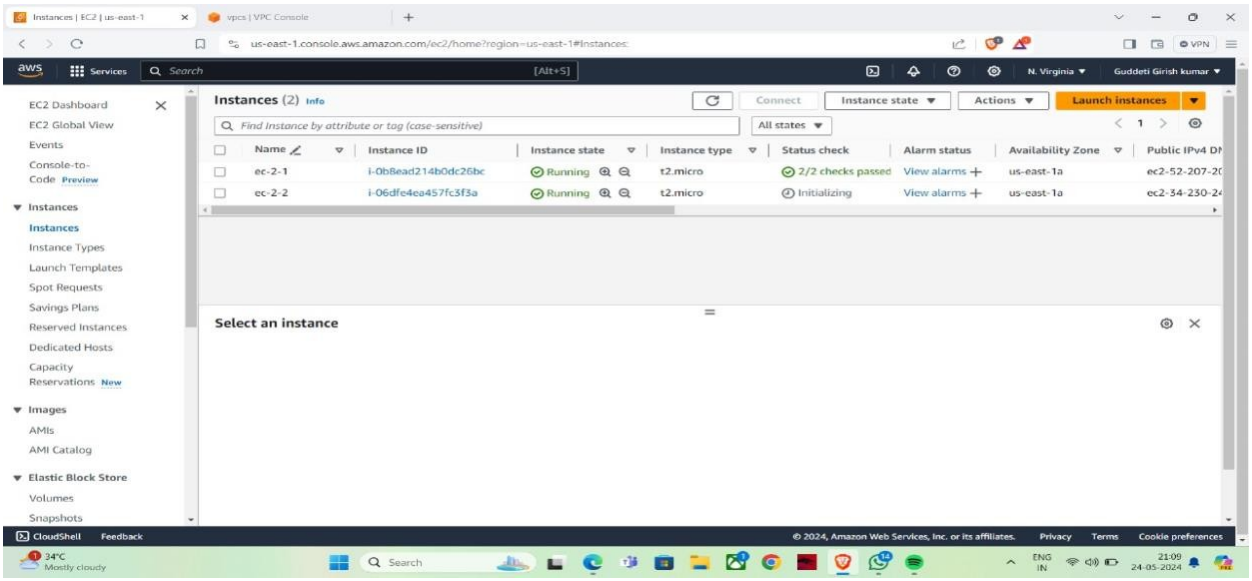
- Click on "Launch Instance".
- Choose an Amazon Machine Image (AMI), such as Amazon Linux 2.
- Choose an instance type (e.g., t2.micro for free tier).
- Configure instance details:
- Network: Select your newly created VPC.
- Subnet: Select Subnet-a for the first instance and Subnet-B for the second instance.
- Add Storage: Configure your root volume as needed.
- Configure Security Group:
- Allow HTTP (port 80) and SSH (port 22) traffic.

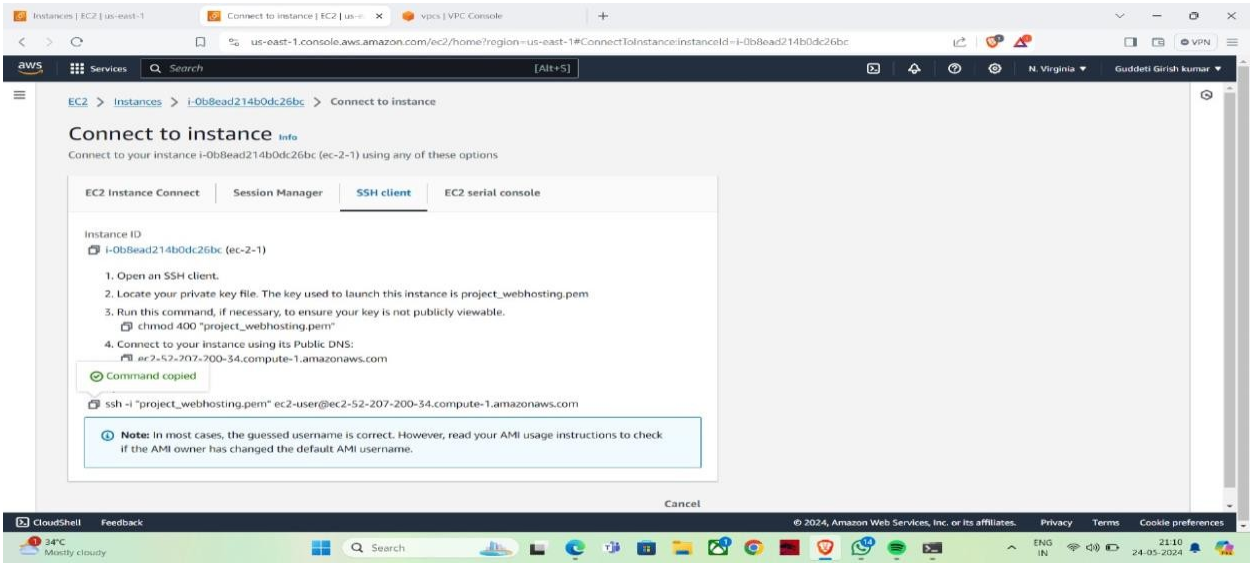
- Review and launch the instances.



- -in instance need to enable the auto assign ip adreess.

- -now connect the instances to Github.





- **-Command prompt for connecting to**

  github. :

cd Downloads # according to the keypair ,

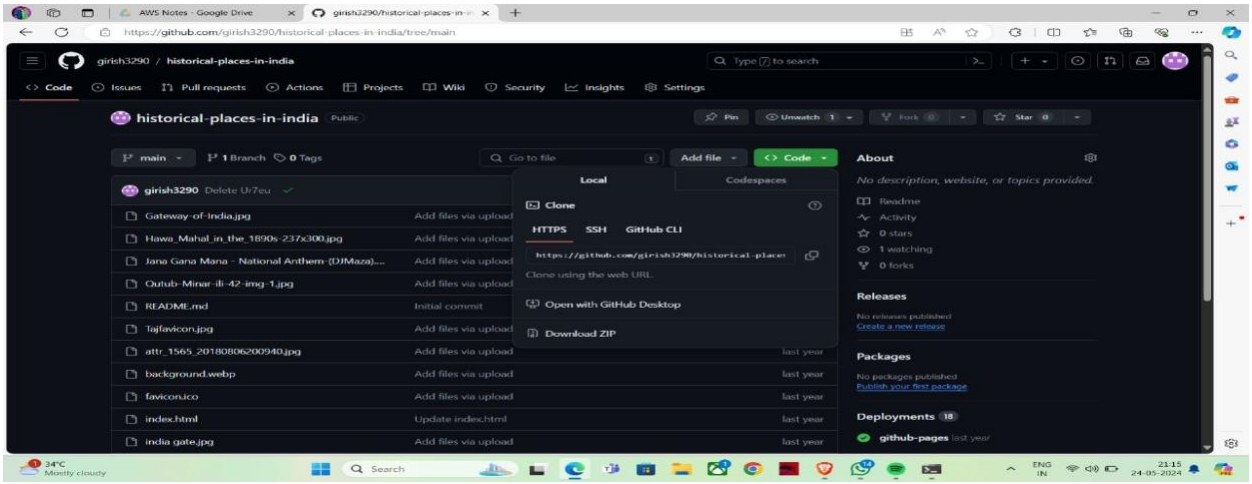
sudo su #switching to root user

yum install -y httpd # installing the apachi servers

yum update -y httpd # updsating the apachi serves

systemctl start httpd.#starting the apachi server in systems.

systemctl enable httpd.#enabling the appachi servers..



wget https://github.com/girish3290/historical-places-in-india.git #creating the directory.

wget https://github.com/girish3290/historical-places-in-india/archive/refs/heads/main.zip

unzip main.zip

mv * /var/www/html/

Cd /var/www/html/

Step 3: Configure an Application Load Balancer (ALB)

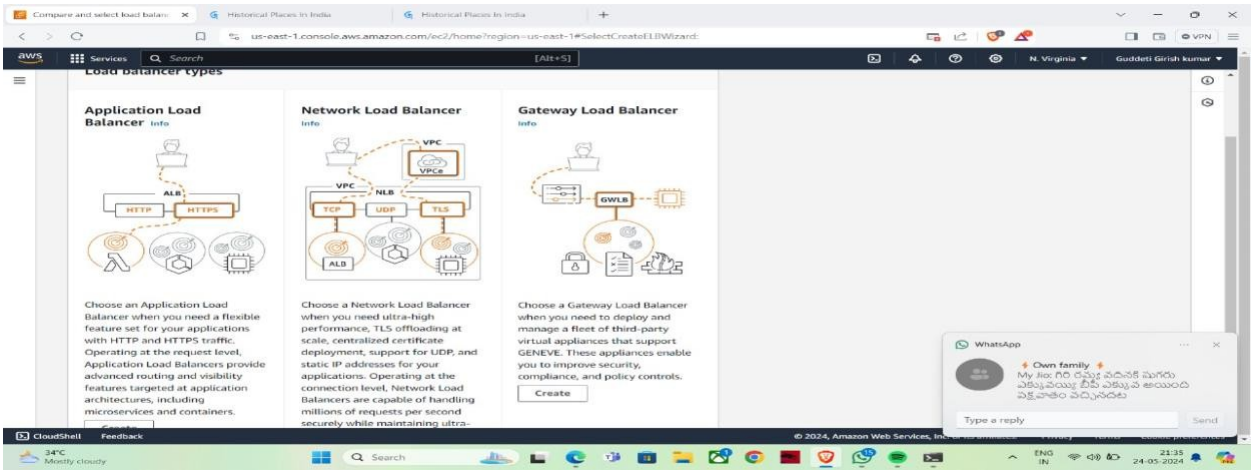
1. Navigate to the Load Balancers Section:

- Open the AWS Management Console.
- Go to the EC2 service and select "Load Balancers" from the left menu.

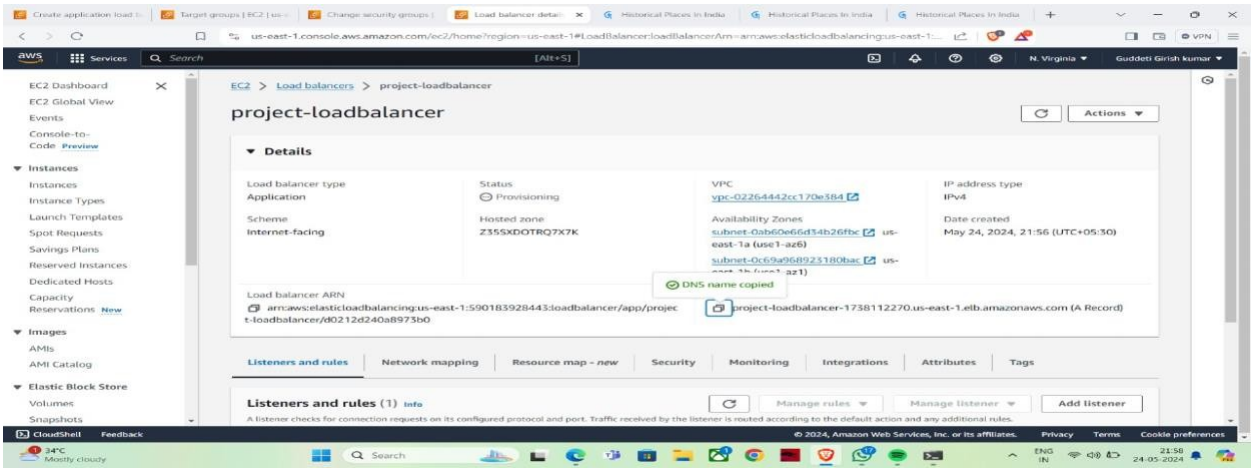
2. Create Load Balancer:

- Click "Create Load Balancer".
- Choose "Application Load Balancer".
- Name the load balancer (e.g., MyALB).
- Scheme: Internet-facing.

- IP address type: IPv4.



-



created load balancer and copied the DNS and hosted in the web browser....

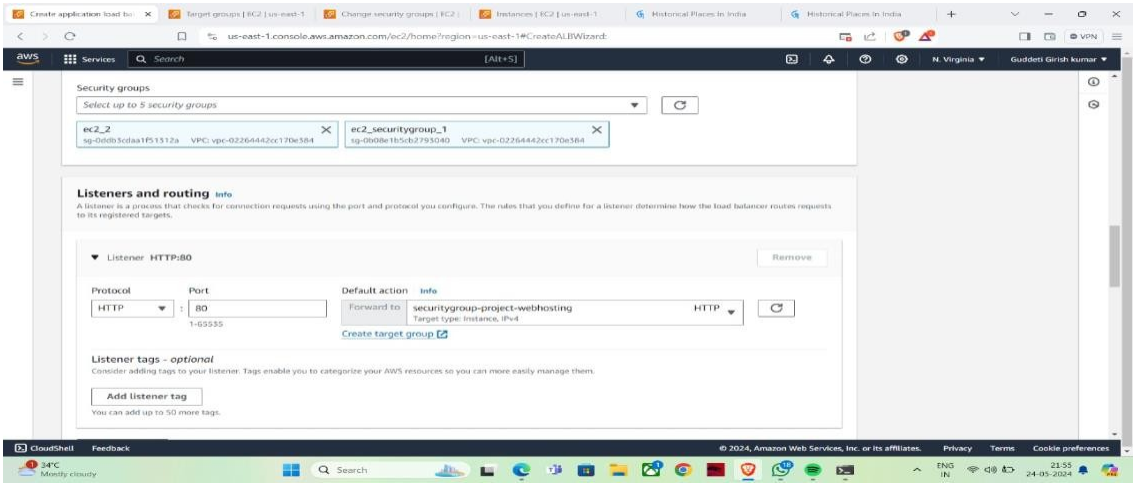
3. Configure Security Groups:

- Choose an existing security group or create a new one that allows HTTP traffic.

4. Configure Routing:

- Create a new target group (e.g., MyTargetGroup).
- Target type: Instance.
- Protocol: HTTP.
- Port: 80.
- Health checks: Use the default path (/).

5. Register Targets:



- Add your two EC2 instances to the target group.
- Review and create the load balancer.

Step 4: Create an Auto Scaling Group (ASG):

1. *Navigate to Auto Scaling Groups*:

- Open the AWS Management Console.
- Go to the EC2 service and select "Auto Scaling Groups" from the left menu.

2. *Create Auto Scaling Group*:

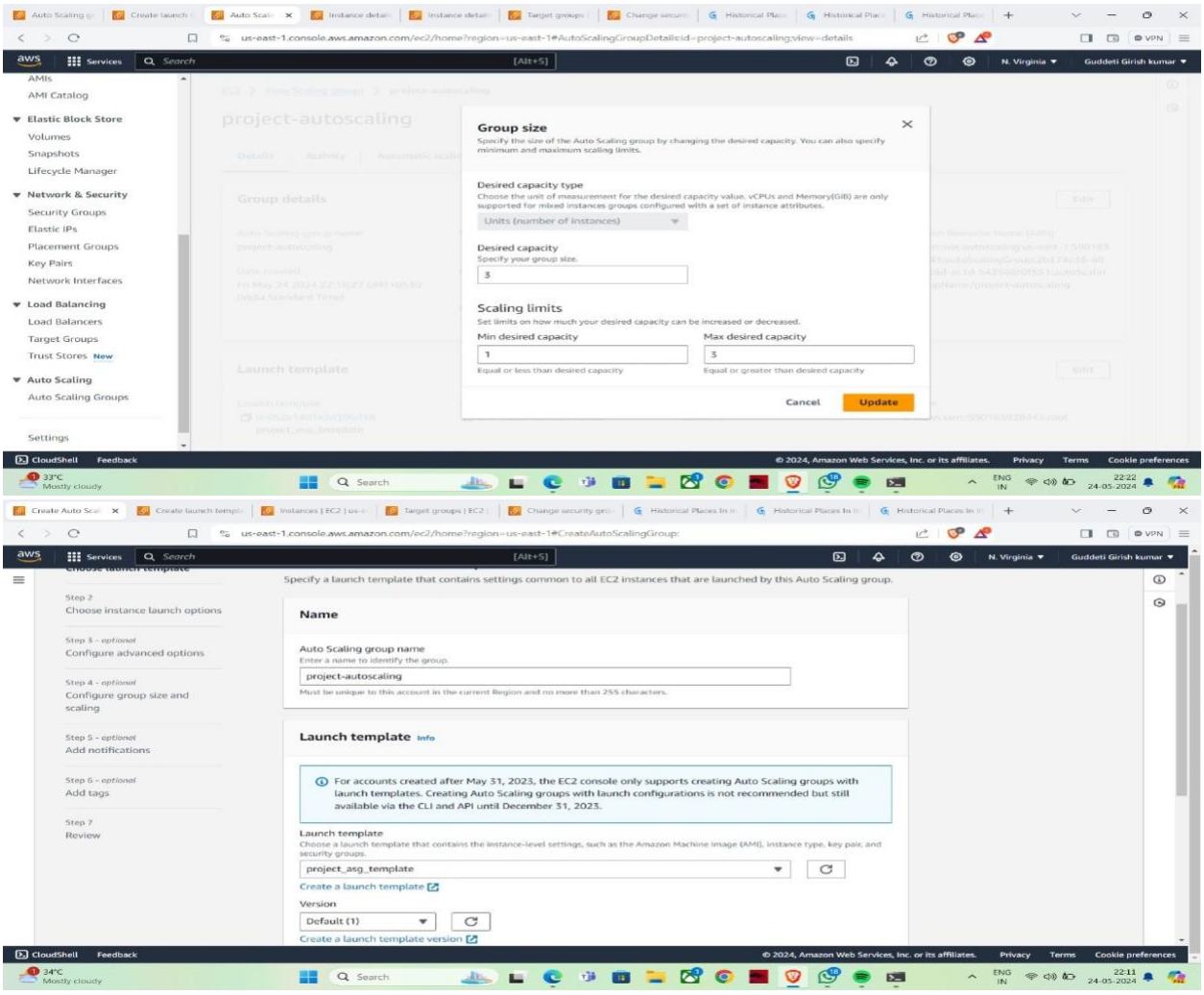
- Click "Create Auto Scaling Group".
- Name the group (e.g., MyASG).
- Launch template: Create a new launch template or use an existing one with your instance configuration.

3. *Configure Auto Scaling Group Details*:

- VPC: Select your newly created VPC.
- Subnets: Select the subnets created in Step 1.

4. *Configure Group Size and Scaling Policies*:

- Set the desired capacity, minimum, and maximum number of instances (e.g., desired: 3, min: 1, max: 4).
- Configure scaling policies to adjust the number of instances based on CPU utilization or other metrics.



Step 5: Attach EBS Volumes to EC2 Instances

1. Navigate to the Volumes Section:

- Open the AWS Management Console.
- Go to the EC2 service and select "Volumes" from the left menu.

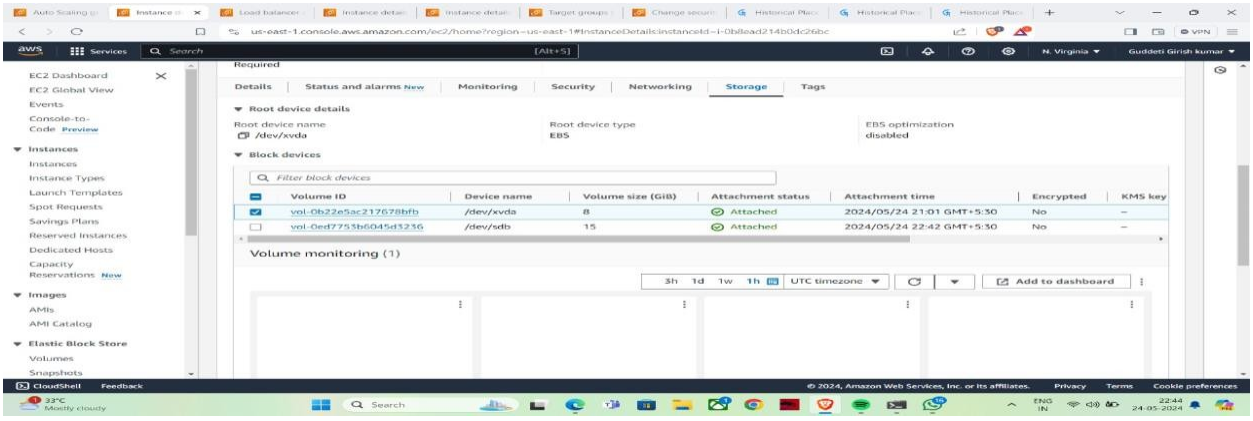
2. Create EBS Volumes:

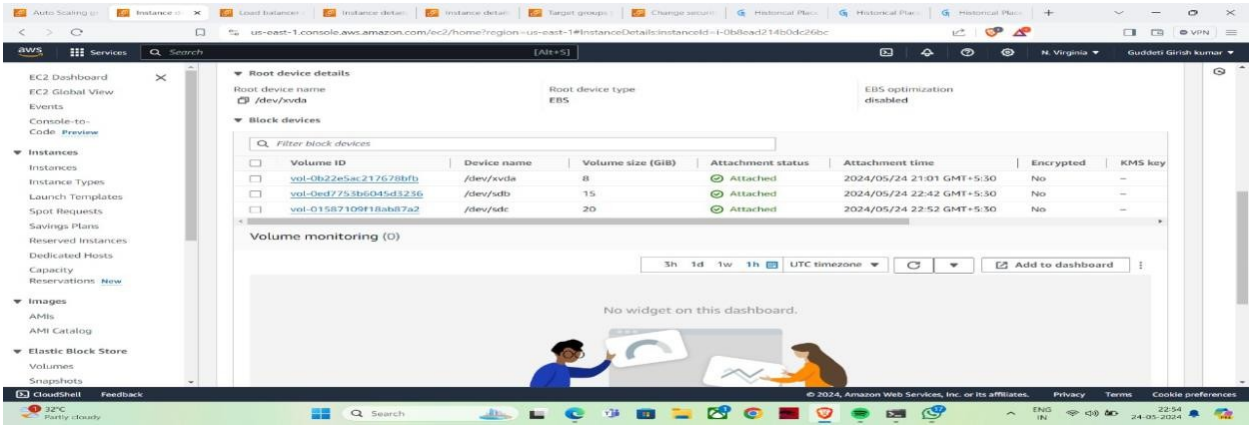
- Click "Create Volume".
- Choose volume type (e.g., General Purpose SSD).
- Specify size and availability zone matching your instances.
- Create the volumes.

3. Attach Volumes to Instances:

- Select each volume and click "Actions" -> "Attach Volume".

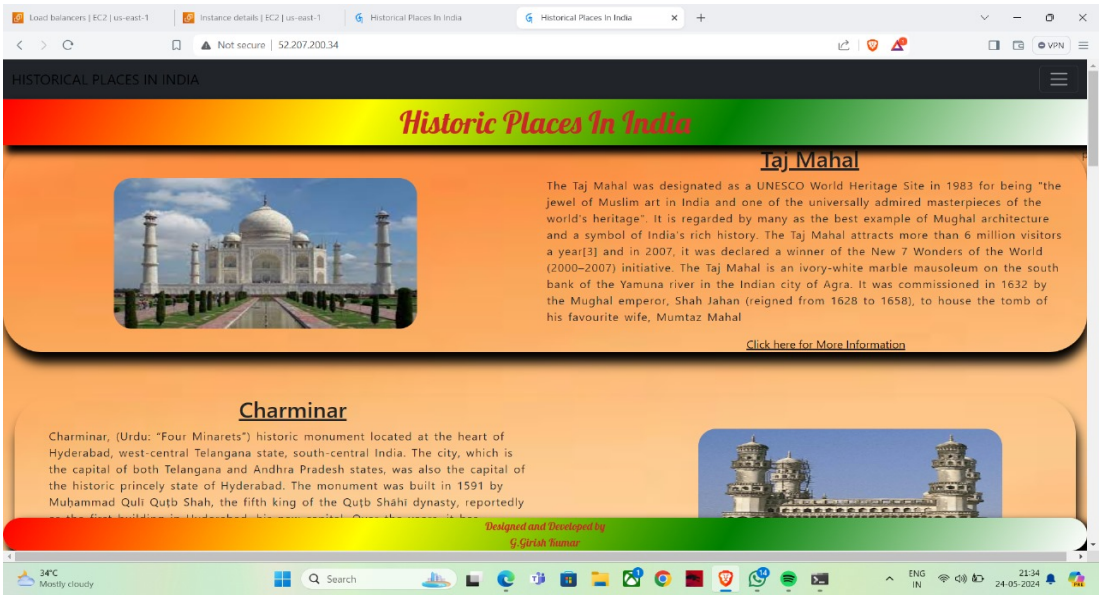
- Choose the corresponding instance and device name (e.g., /dev/sdf).
- -**attached to the volume to EC2 instance 1 ,2 successsfully.**





Output is changing according to the traffic :

- This is happened because of ELB (based on traffic , ELb is distributing the traffic..
- Based on the ASG (auto scalling group)instance is running automatically , where we have configured the instance as desired instance , minimum instance and maximum instance .
- Attached the EBS (Elastiic Block storage)volume is aattached to the instance.



- When we get the output , After refreshing the website , the output is changed based on instance ASG ,ELB (traffic and auto scalling groups) automattically.
- Here the output in the website footer level.