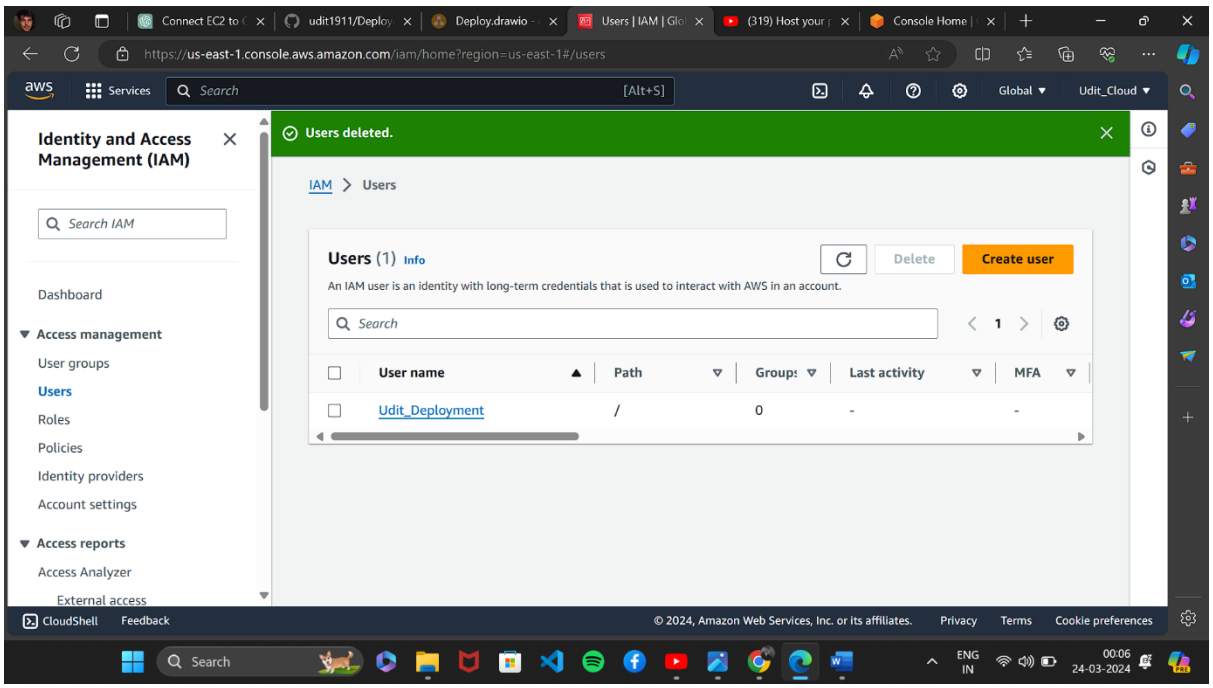
DEPLOYING APPLICATION ON AWS

The Feature and Functionality Our Application will have is that:

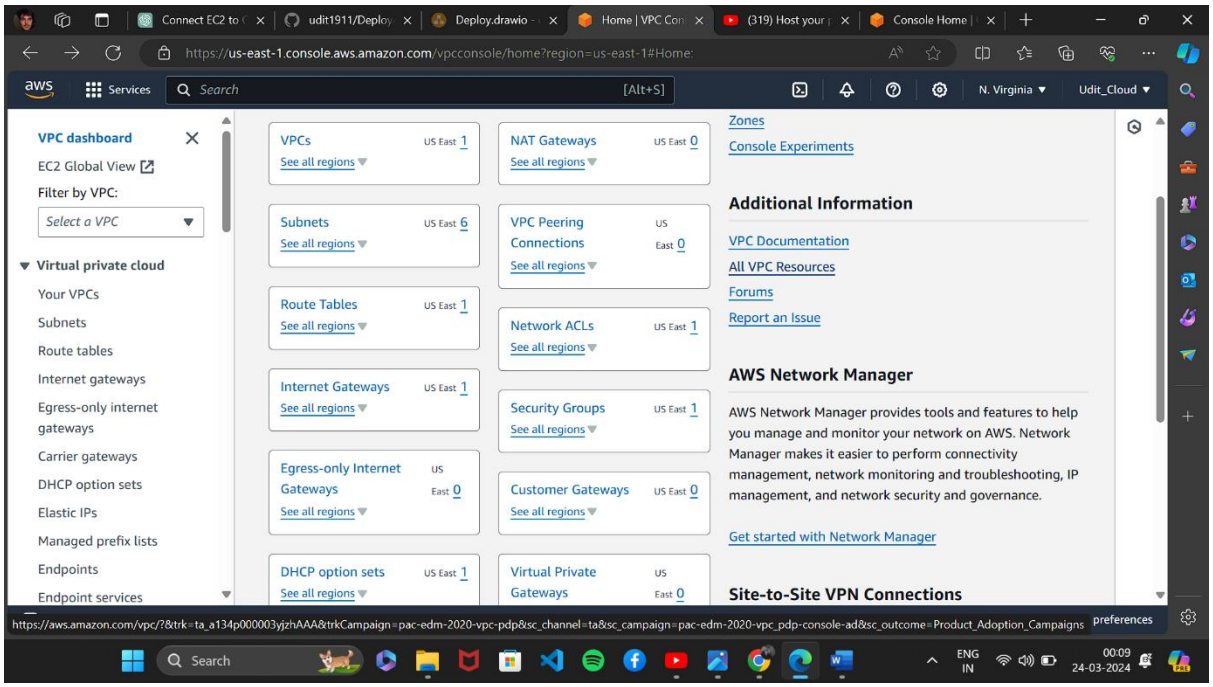
- Environment will be **isolated using VPC** which allows us to create a logically isolated section of the AWS cloud where we can launch AWS resources. This isolation ensures that our resources are secure and cannot be accessed by other users' resources.
- **Identity and Access Management (IAM)** that helps us to securely control access to AWS resources. IAM enables us to manage users, groups, roles, and permissions to securely interact with AWS services.
- **Amazon Elastic Compute Cloud (Amazon EC2)** which provides resizable compute capacity in the cloud, allowing us to quickly scale up or down to meet changing demands. EC2 offers a wide range of features that make it a versatile and powerful computing platform
- **AWS Auto Scaling** which automatically adjusts the number of instances or resources in our application based on demand, allowing us to maintain performance and optimize costs.
- **ELB** which automatically distributes incoming traffic across multiple targets in multiple Availability Zones to ensure that our application remains available and resilient to failures.
- **CloudFront** which has a large network of edge locations distributed globally, allowing it to deliver content to users with low latency and high transfer speeds regardless of their geographical location.
- **CloudWatch** which collects and stores metrics (such as CPU utilization, network traffic, and disk I/O) from AWS services, EC2 instances, and custom applications in real-time. We can view these metrics on dashboards, set alarms, and use them to monitor the health and performance of your resources.

STEP 1 – CREATE **IAM ROLE** FROM ROOT ACCOUNT WHICH HAVE APPROPRIATE POLICIES AND PERMISSION RELATED TO EC2 AND MORE



Identity and Access Management (IAM)

STEP 2 – CREATE **ISOLATED ENVIRONMENT USING VPC** WHICH WILL DEFINE SUBNETS, GATEWAYS, ROUTE TABLES.

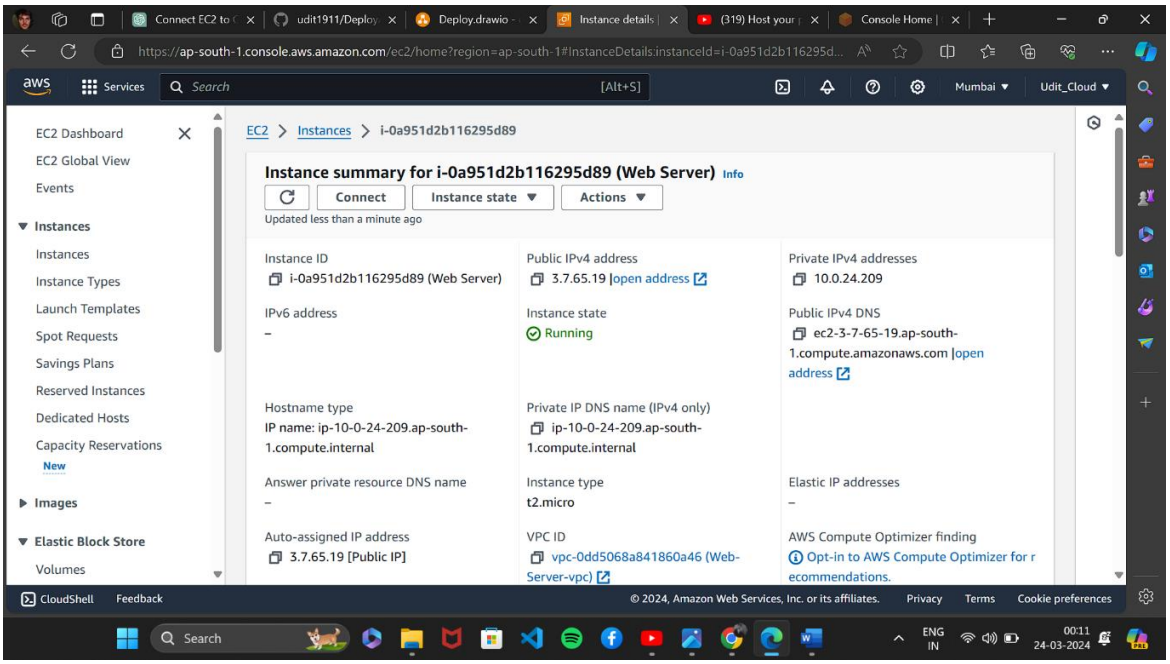


Virtual Private Cloud (VPC)

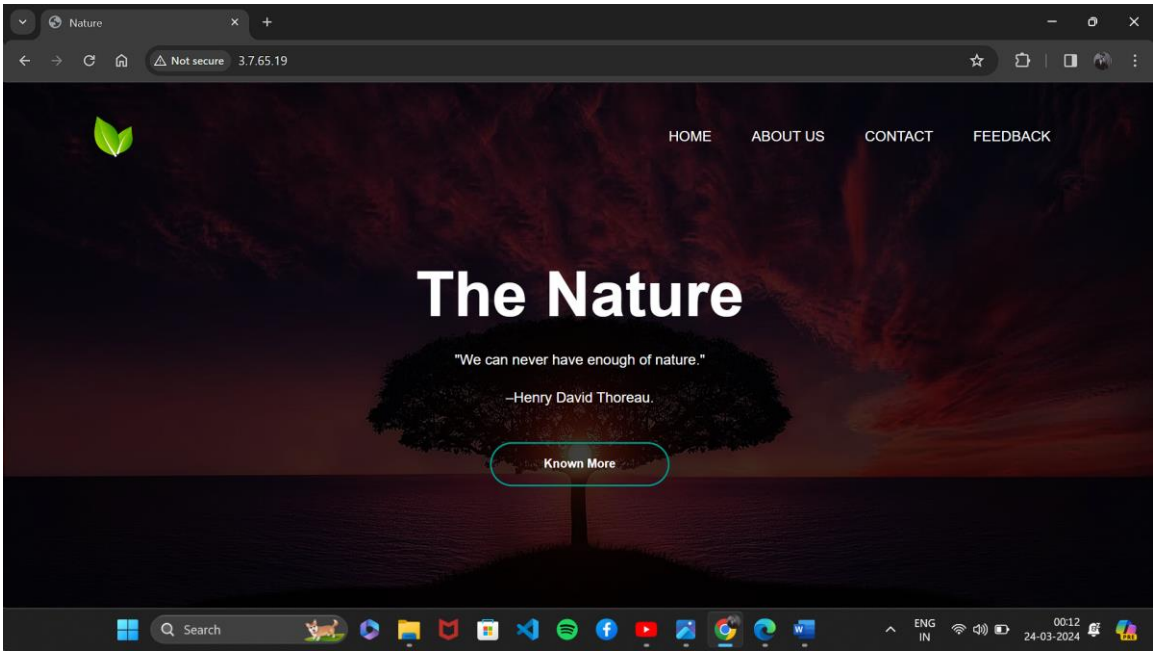
STEP 3 – LAUNCH AN **EC2 INSTANCES** WITH PRECISE CONFIGURATION.

- EC2 SHOULD BE IN PUBLIC SUBNET OTHERWISE IT WILL BE NOT ACCESSIBLE FROM PUBLIC
- DEFINE SECURITY GROUP WHICH WILL ALLOW HTTP PORT 80 AND HTTPS PORT 443 FROM PUBLIC IPS.
- NOW CONNECT EC2 INSTANCE TO APPLICATION FOR THIS USE EC2 INSTANCE CONNECT
- NOW ON LINUX MACHINE FIRST GET TO ROOT ACCOUNT BY GIVING COMMAND "Sudo su -"
- THEN CHECK YOU DO HAVE UPDATED VERSION BY GIVING COMMAND "yum update -y"
- SO WE ARE USING YUM PACKAGE MANAGER SO WE WILL RUN COMMAND "yum install -y httpd" TO INSTALL THE APACHE HTTP SERVER.
- NOW IT'S TIME TO CHECK STATUS OF APACHE HTTP SERVER WE'LL DO THIS BY RUNNING COMMAND "systemctl status httpd"
- MAKE A NEW DIRECTORY IN LINUX WHERE APPLICATION WILL BE CLONED FROM SOURCE. EXECUTE COMMAND "mkdir <directory_name>
- NOW GET INTO DIRECTORY BY COMMAND "cd"
- WE'RE USING GITHUB IN THIS TO CLONE THE APPLICATION, FOR THIS YOU SHOULD HAVE GIT INSTALLED, DO THIS BY EXECUTING COMMAND "sudo apt install git"
- THEN CLONE THE APPLICATION FROM GITHUB REPO BY EXECUTING COMMAND "git clone <repo_add>"
- ONE YOU HAVE DONE ALL THIS TAKE ALL THE FILES FROM CREATED FOLDER TO var/www/html/, EXECUTE COMMAND "mv * /var/www/html/
- NOW WE'LL RUN COMMAND "systemctl enable httpd" SO THAT APACHE HTTP SERVER('HTTPD') SERVICE WILL START AUTOMATICALLY AT SYSTEM BOOT

- RUN COMMAND “systemctl start httpd” SO THAT APACHE SERVER WILL START ON LINUX SYSTEM.
- NOW GOTO EC2 AND SEARCH PUBLIC IP, IT’LL START WORKING



Running EC2 Instance

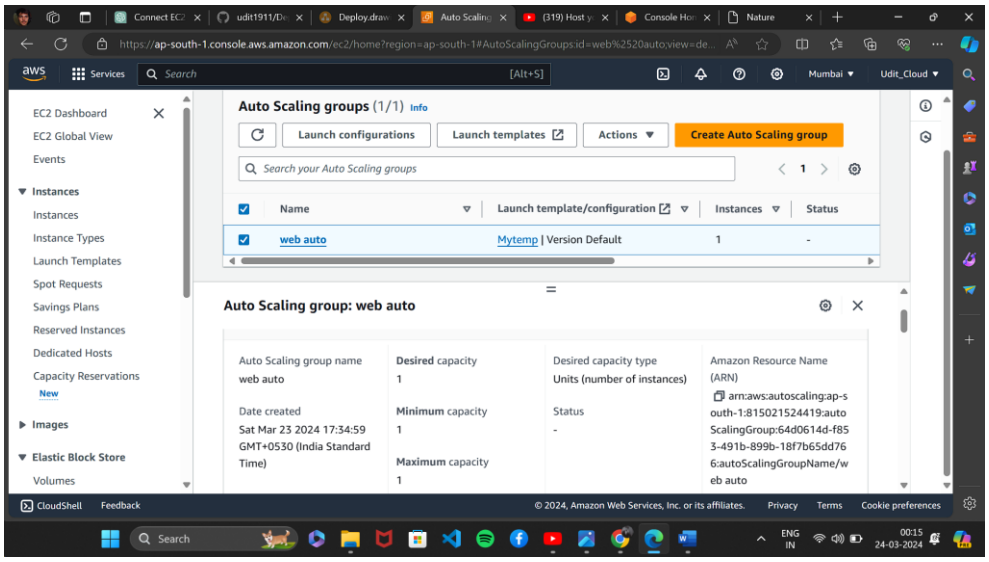


http://3.7.65.19/

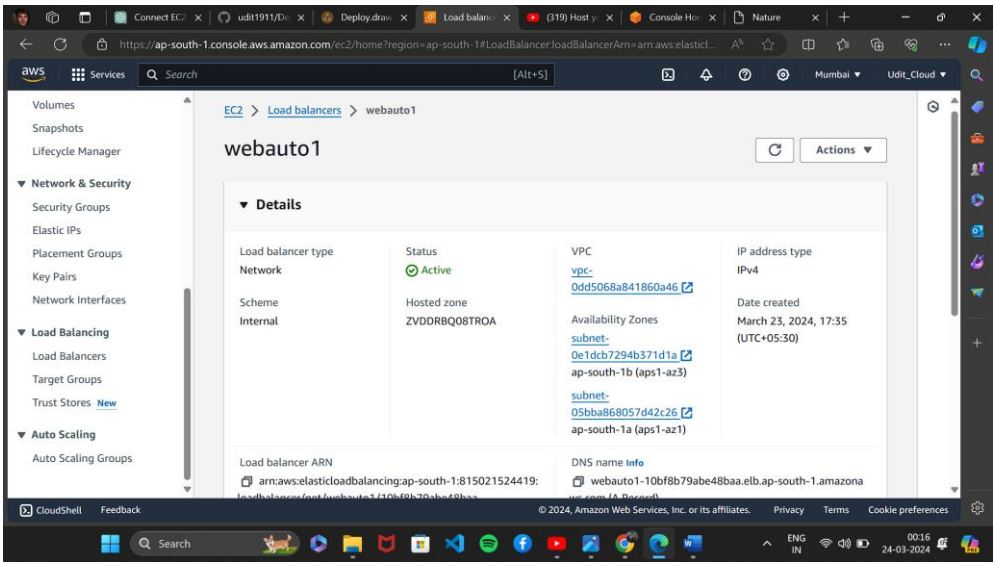
IP – “ http://3.7.65.19/ “

Step 4 – Enable **Auto Scaling**, For this Goto

- Auto Scaling group
- enter group name
- Select the Launch Template
- Choose instance launch options then Configure group size and scaling.
- You can add SNS feature as well which will gives notification about Auto scaling and Review and Launch.
- In between these process it'll also ask for load balancer and Monitoring (cloudwatch), Enable both of them.



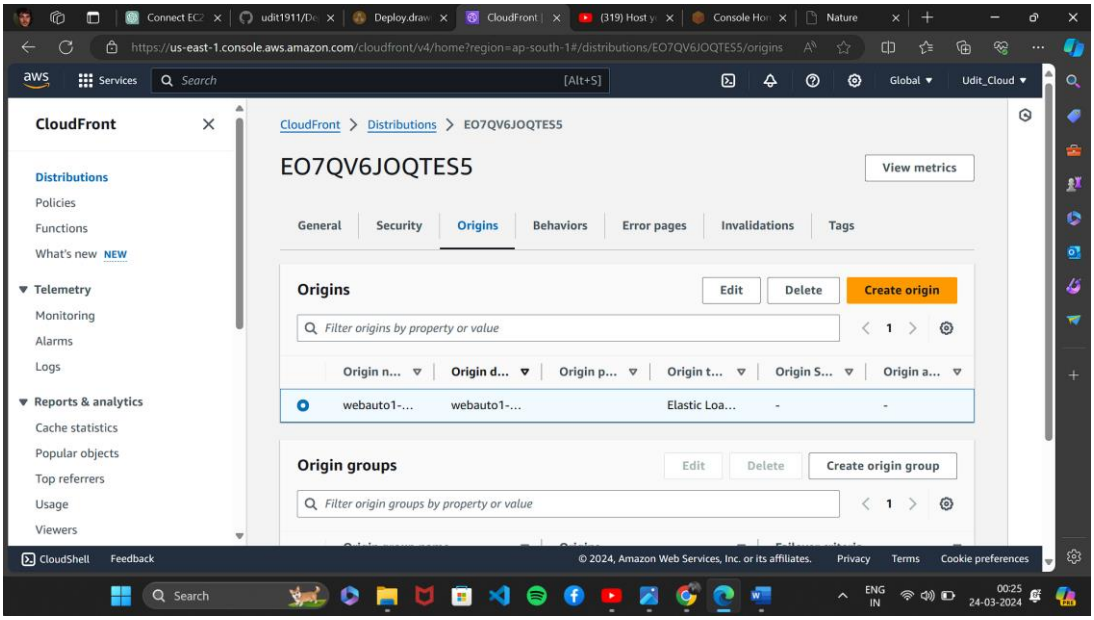
Auto Scaling



Load Balancer

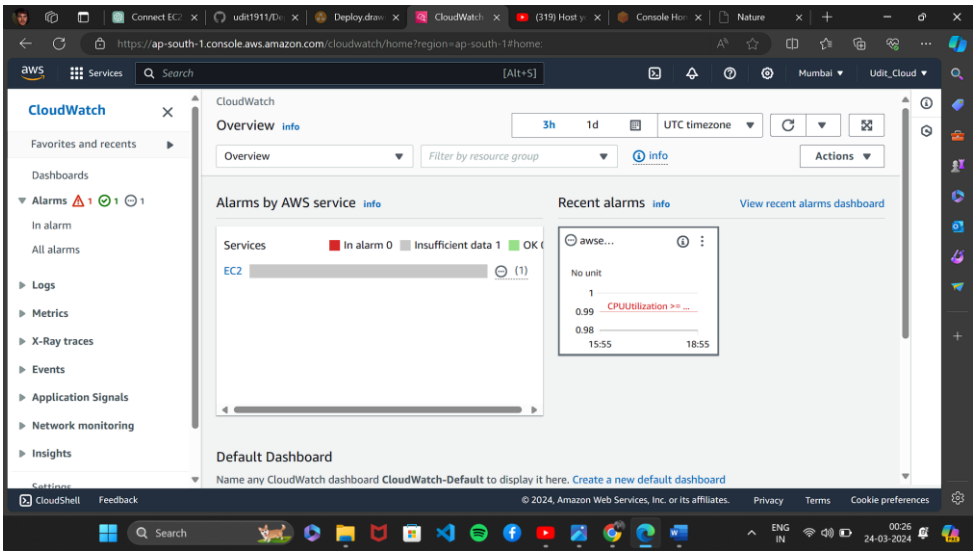
Step 4 – Enable **CloudFront** for Low Latency

- Get to the CloudFront
- Click on Create Distribution
- Choose origin domain (Elastic Load Balancer)
- And now give name to origin
- Choose between cache policy
- Enable Web Application Firewall(if it need to)
- Now Create Distribution is done



CloudFront Distribution

Cloud Watch Dashboard for Monitoring and Health



CloudWatch Dashboard