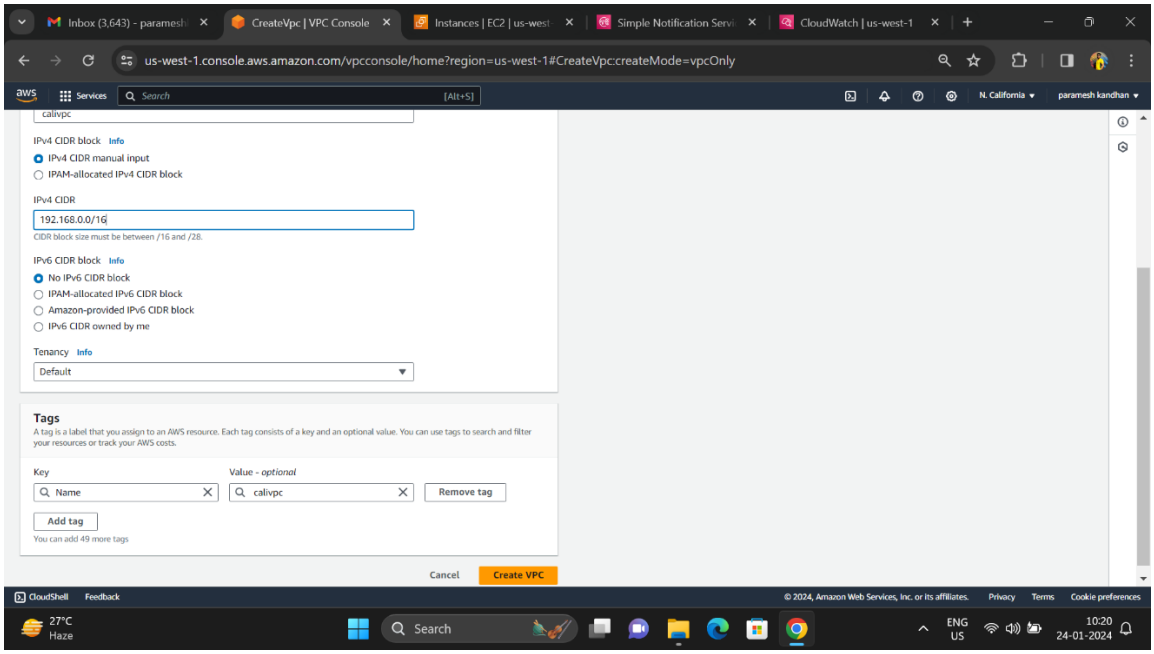
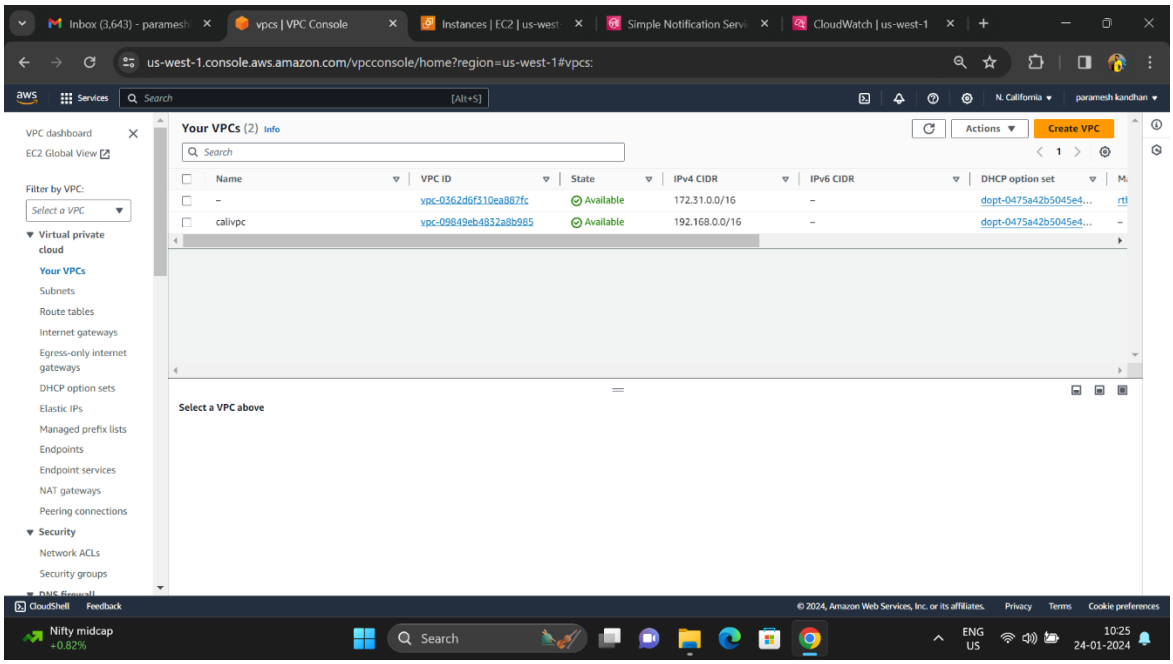
Implementing-Load-Balancing-with-Elastic-Load-Balancer

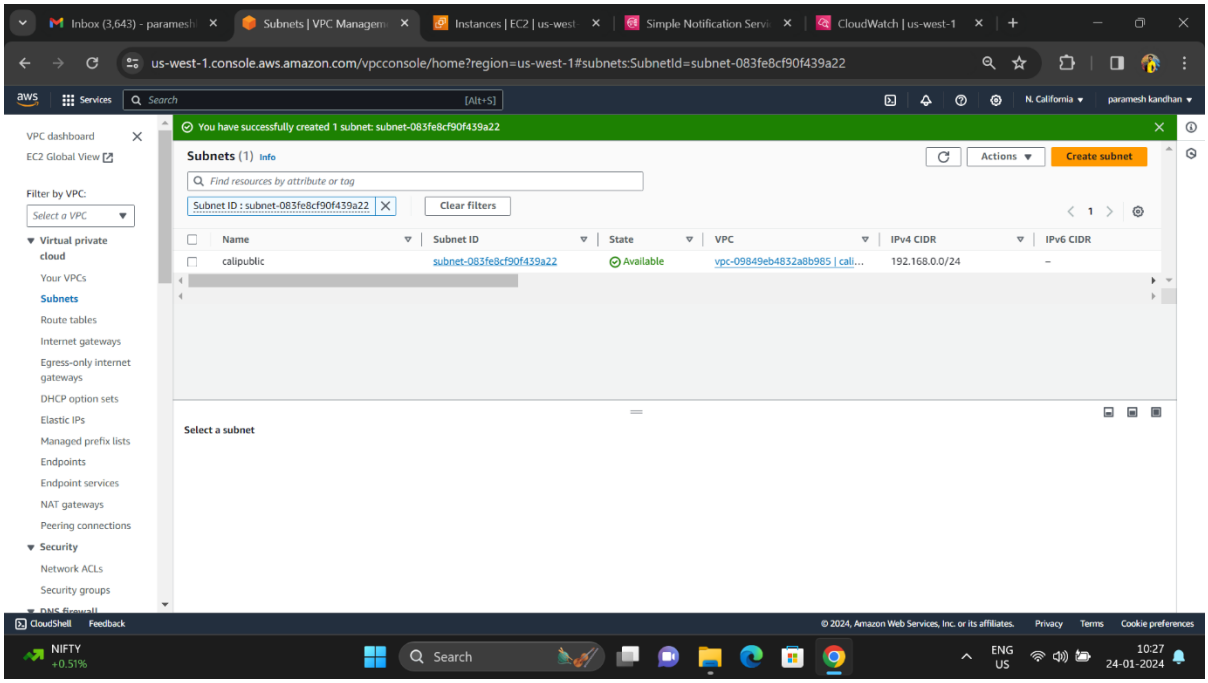
This project revolves around the implementation of Load Balancing with Elastic Load Balancer, employed to effectively regulate network traffic within a designated Virtual Private Cloud (VPC). In scenarios of heightened traffic, the system will dynamically spawn new instances through an auto-scaling mechanism, ensuring responsive and scalable infrastructure to accommodate varying workloads seamlessly. This strategic utilization of Elastic Load Balancer, coupled with auto-scaling capabilities, enhances the system's capacity to gracefully handle increased demand, thereby optimizing overall performance and resource allocation.

PROCESS:

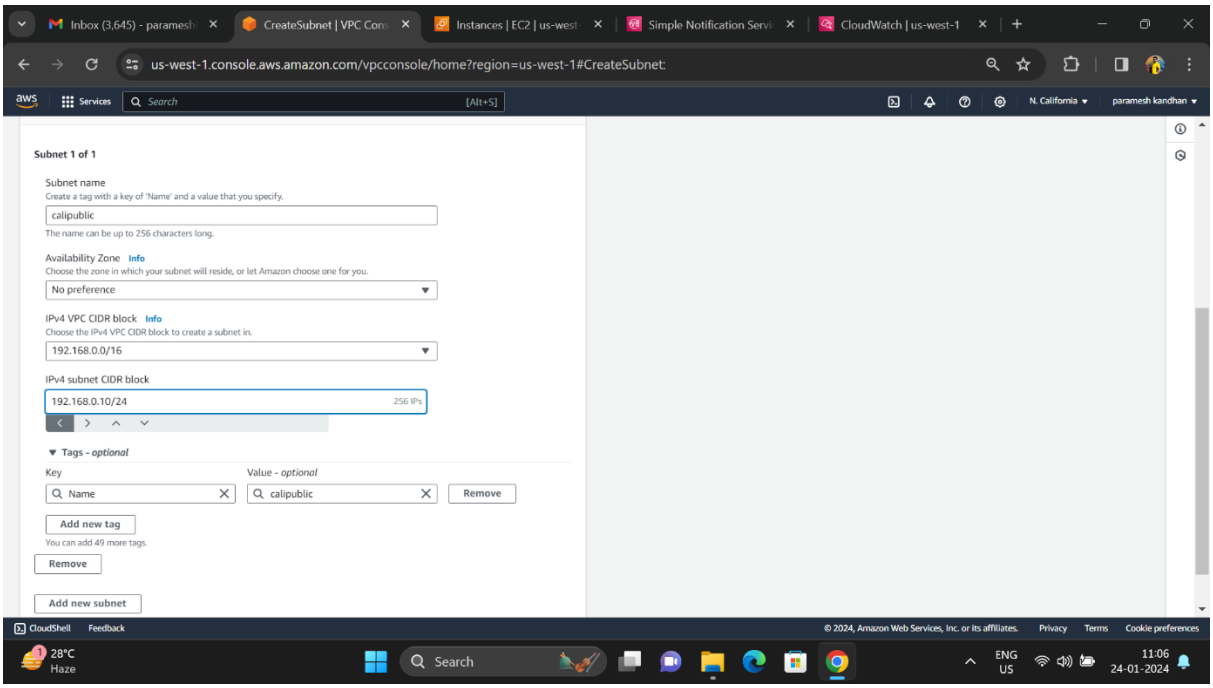
- To create a aws account and then need to use the service of EC2 .
- In the webserver need to create a VPC for the security and the safety purpose.



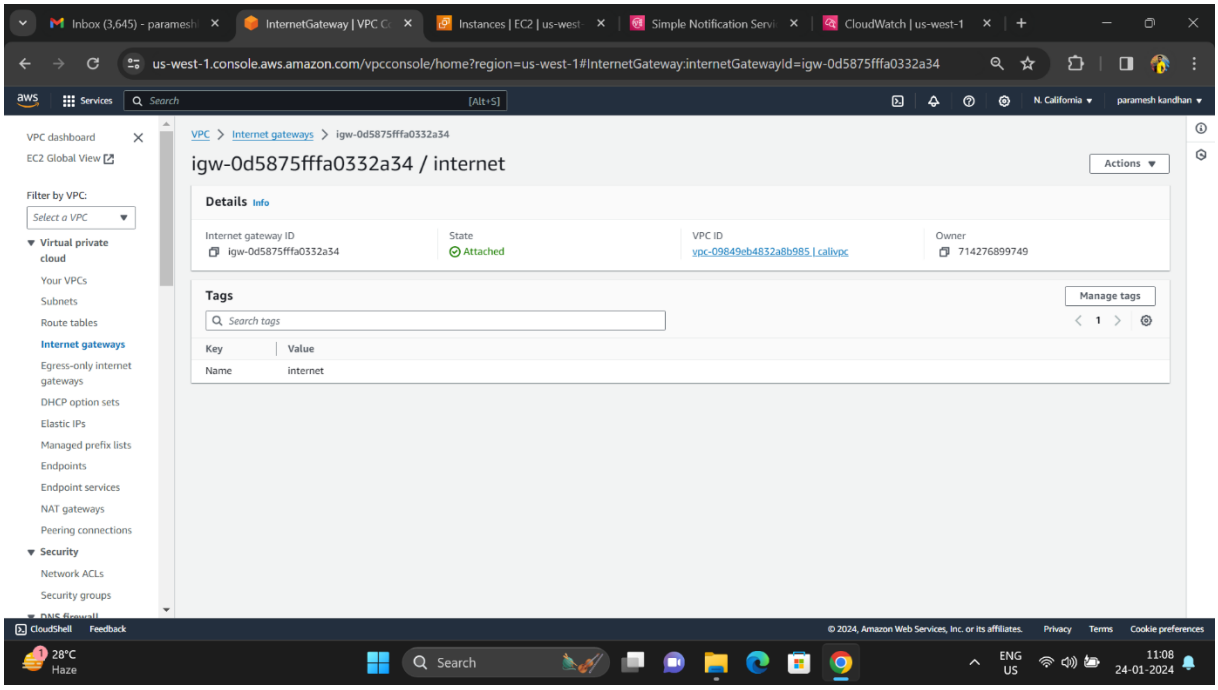




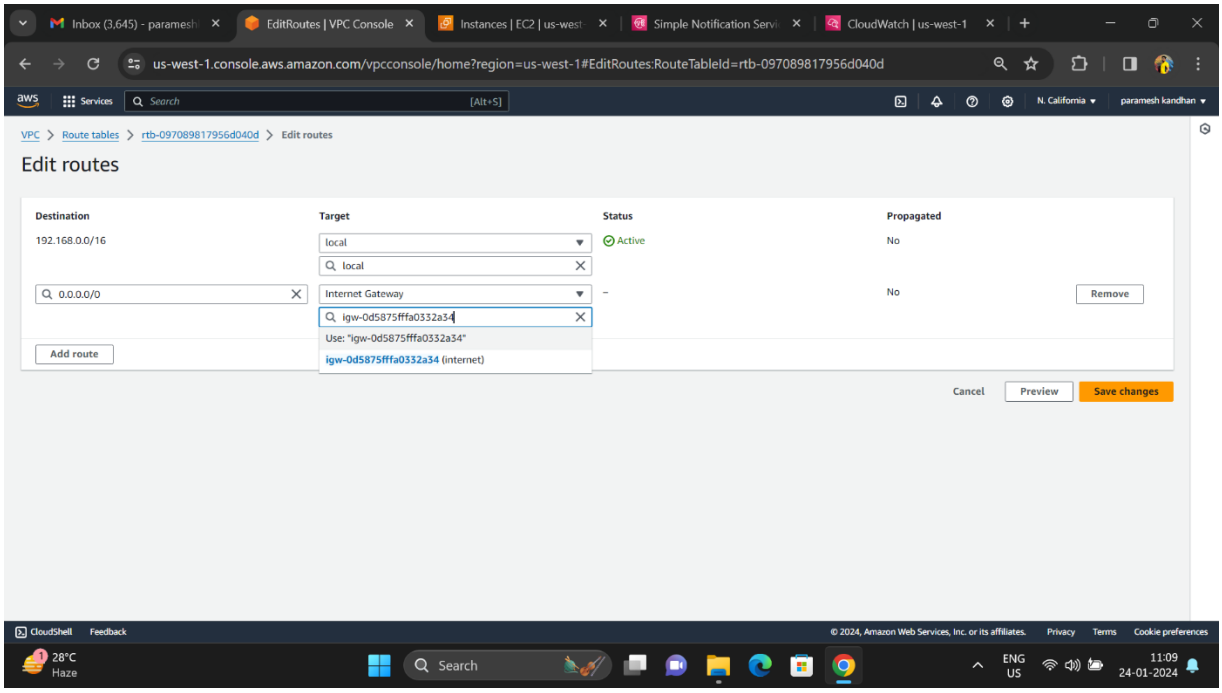
- Then create a subnet in the class c . it will be start the IP adds: 192.168.0.10/24.



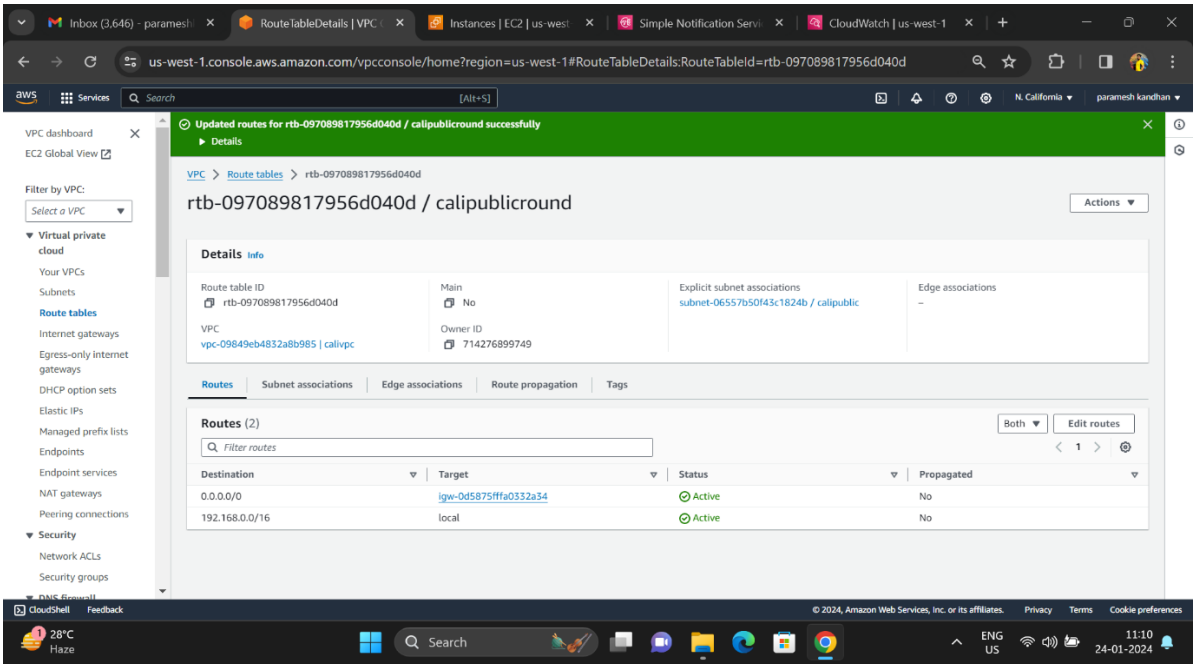
- Sub netting allows organizations to divide their IP address space into smaller, more manageable segments. This is especially important in situations where there might be a limited pool of IP addresses available.
- Sub netting enables organizations to allocate IP addresses based on their specific needs for different departments, locations, or network segments.
- Sub netting can be used to isolate different parts of a network. This isolation can enhance security by controlling the flow of traffic between subnets and making it more challenging for unauthorized access or attacks to spread across the network.
- Network administrators can implement different security policies and access controls for each subnet, providing granular control over network resources.

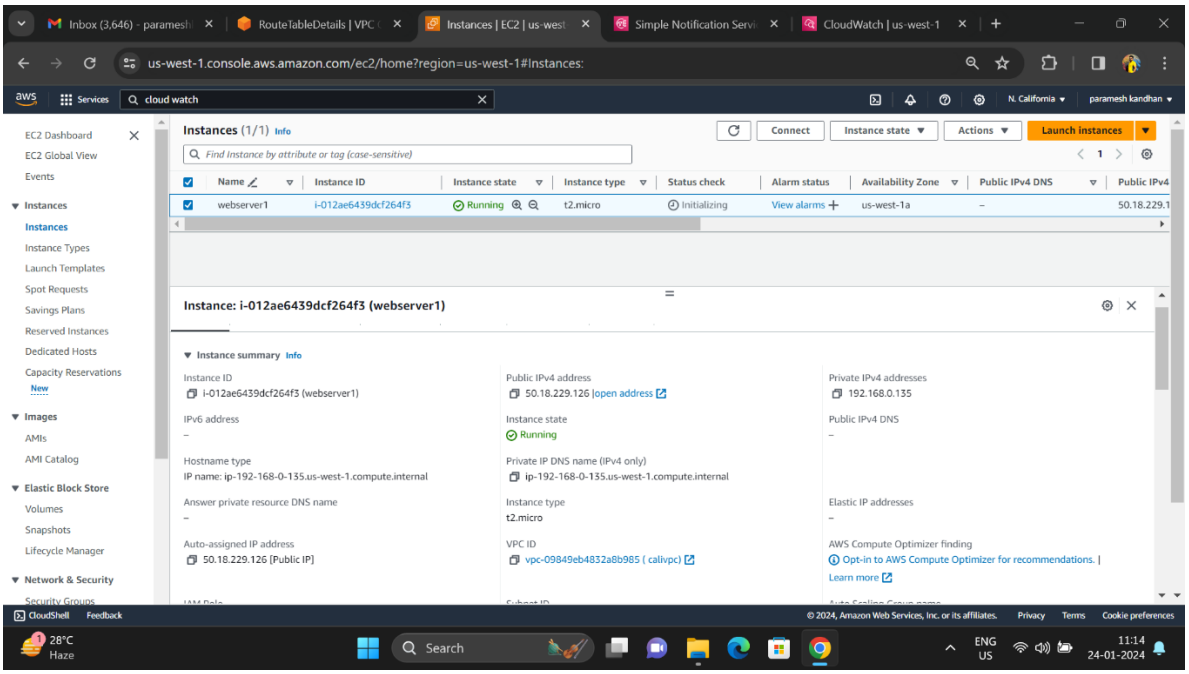


- Need to enable the internet gateways for the purpose of connect the internet connectivity in the Public IP.

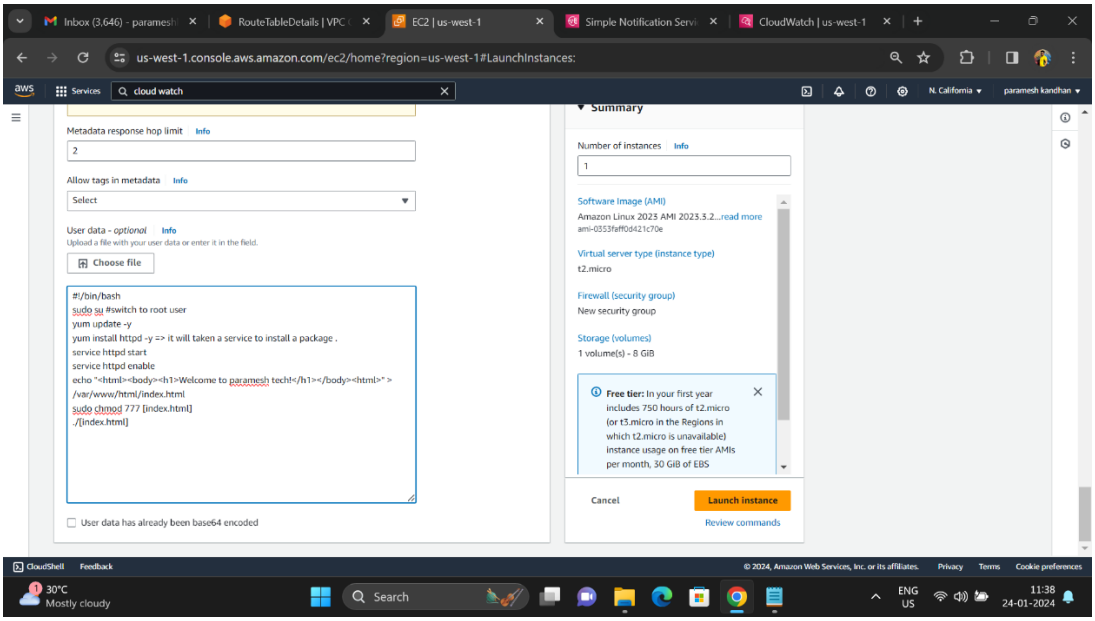


- Route tables can be populated through dynamic routing protocols (e.g., OSPF, EIGRP, BGP) or manually configured static routes. Dynamic routing protocols allow routers to exchange information about network reachability automatically, while static routes are manually configured by network administrators.
- Route tables are used in both IPv4 and IPv6 networks. They play a crucial role in directing traffic through the network based on the destination IP address.

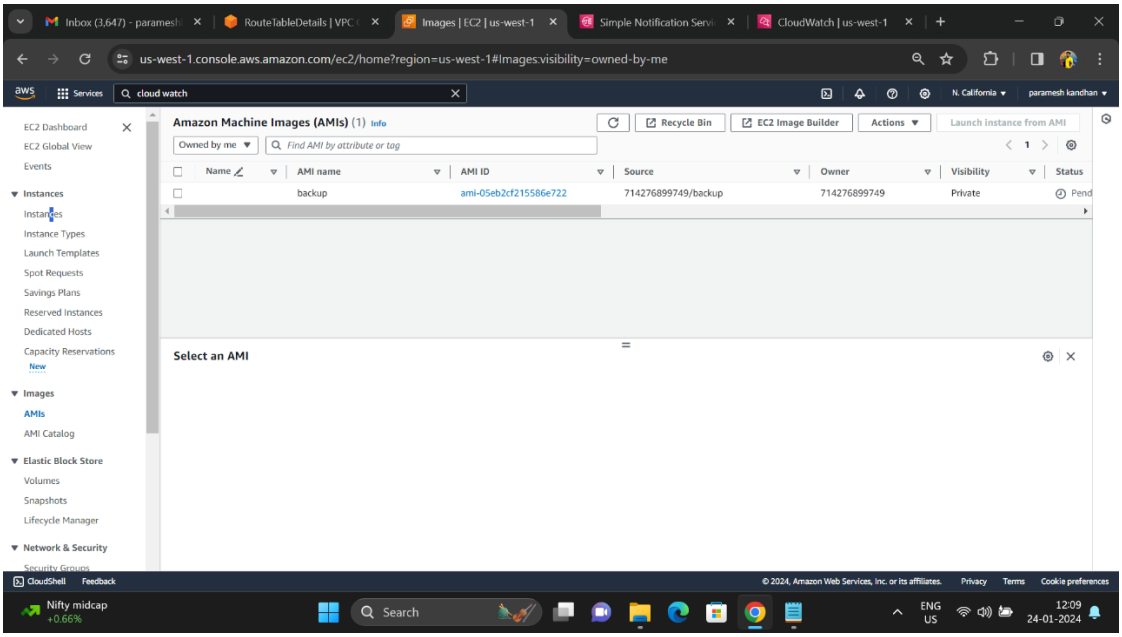


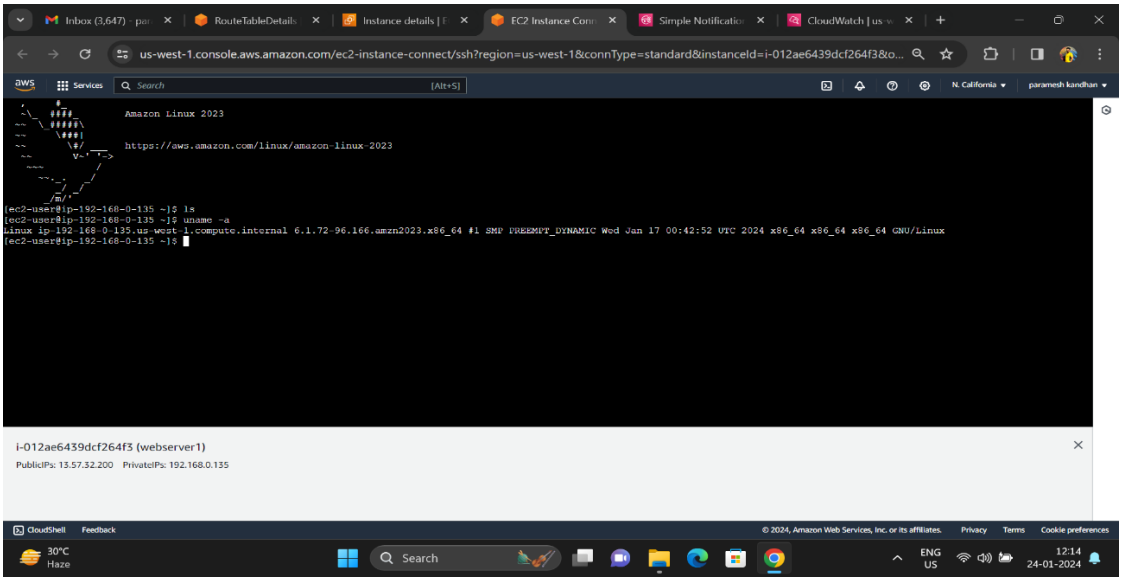


- Then need to create a ec2 instance under the VPC
- In the ec2 instance need to install the httpd package.
- # Yum install httpd* -y. Using a shell script enable the httpd on create a instance with the web sever.
- #!/bin/bash
- sudo su #switch to root user
- yum update -y
- yum install httpd -y => it will taken a service to install a package .
- service httpd start
- service httpd enable
- echo "<html><body><h1>Welcome to paramesh tech!</h1></body><html>" > /var/www/html/index.html
- sudo chmod 777 [index.html]
- ./[index.html]

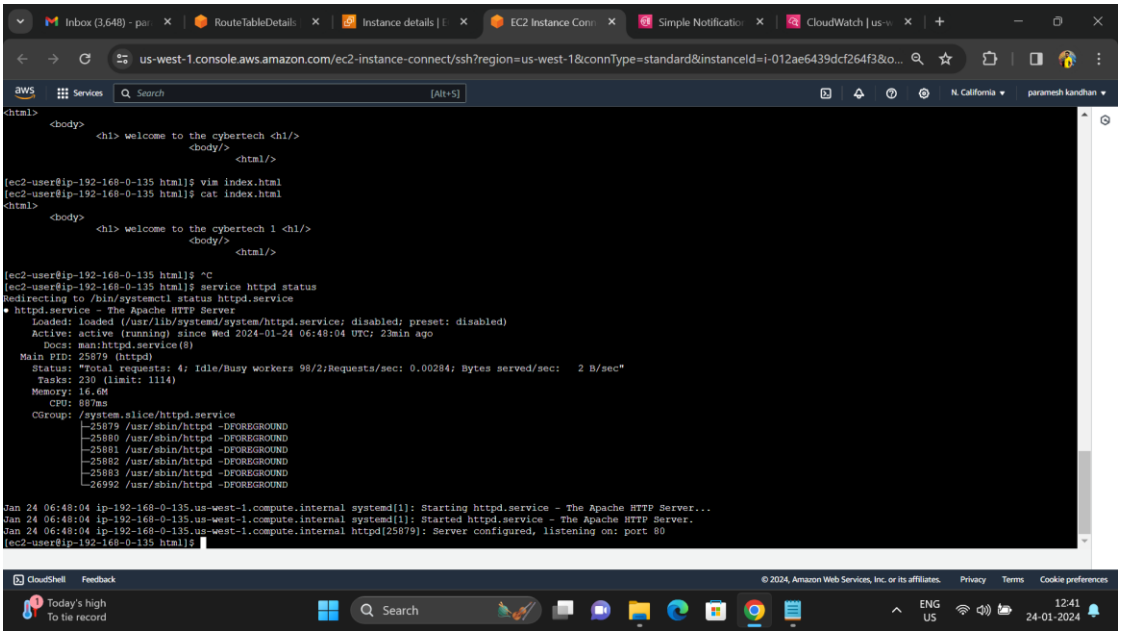


- We using a auto scaling concept for the purpose of taken a AMI as a backup.

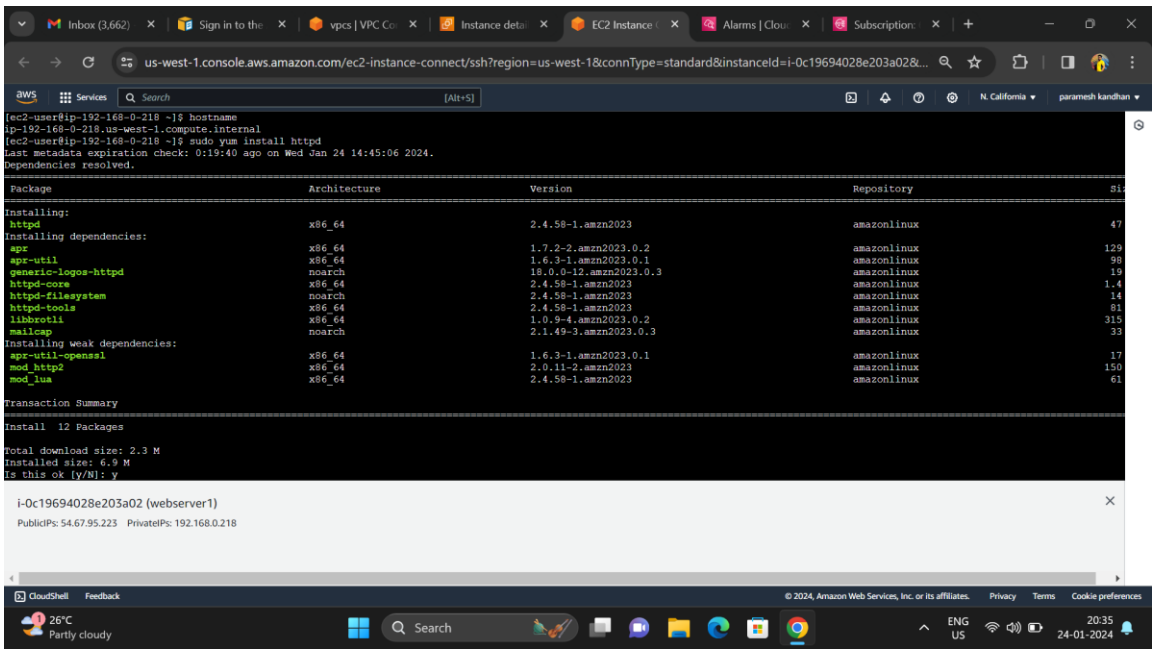


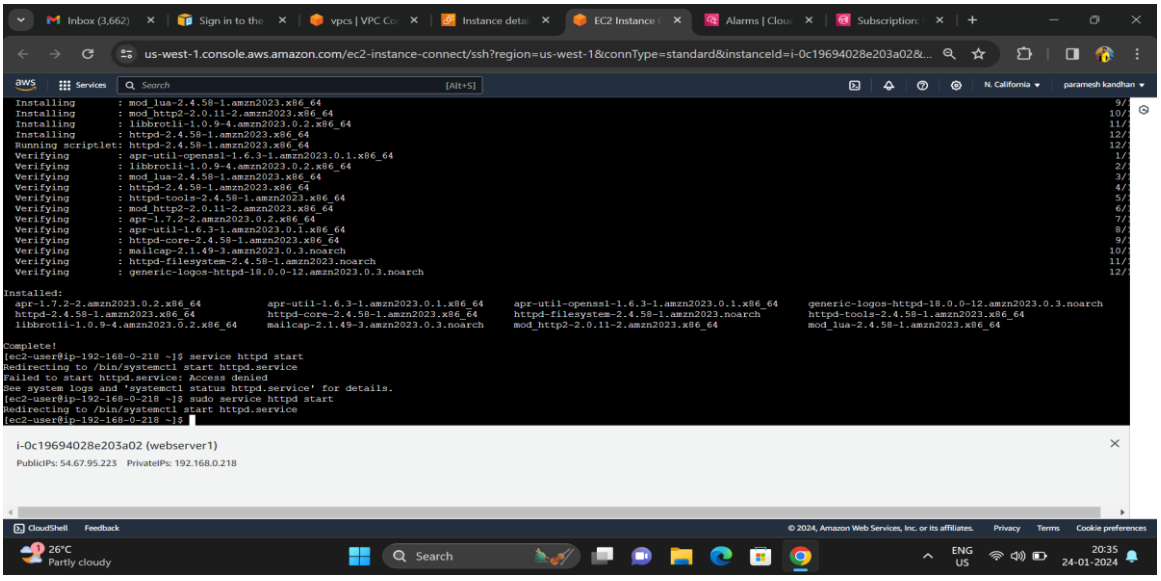


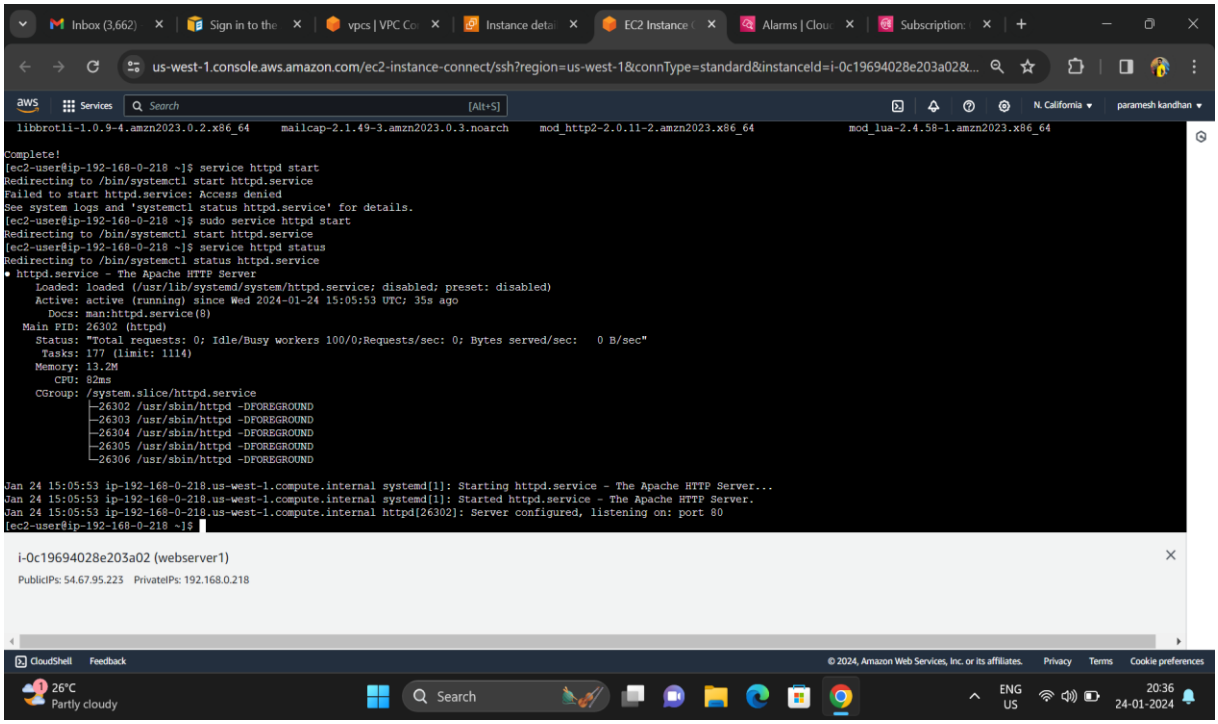
- Now create a linux server for the web deployment.



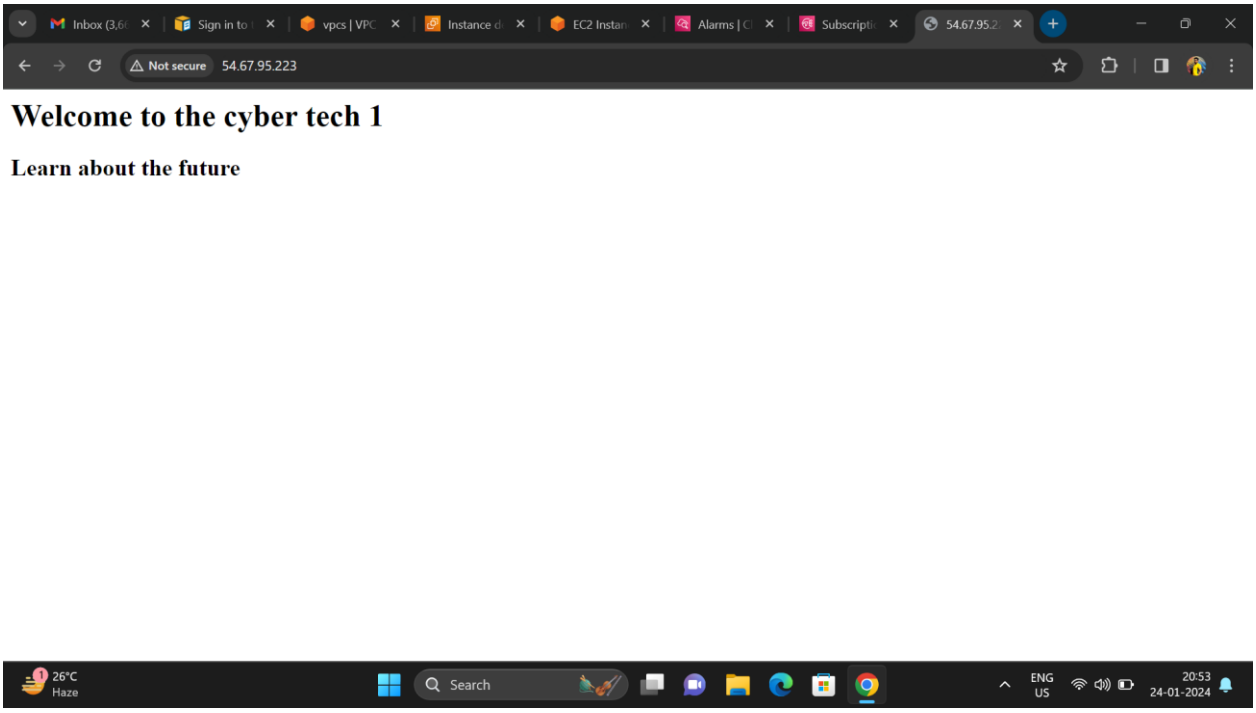
- Setup a html code for the web hosting .





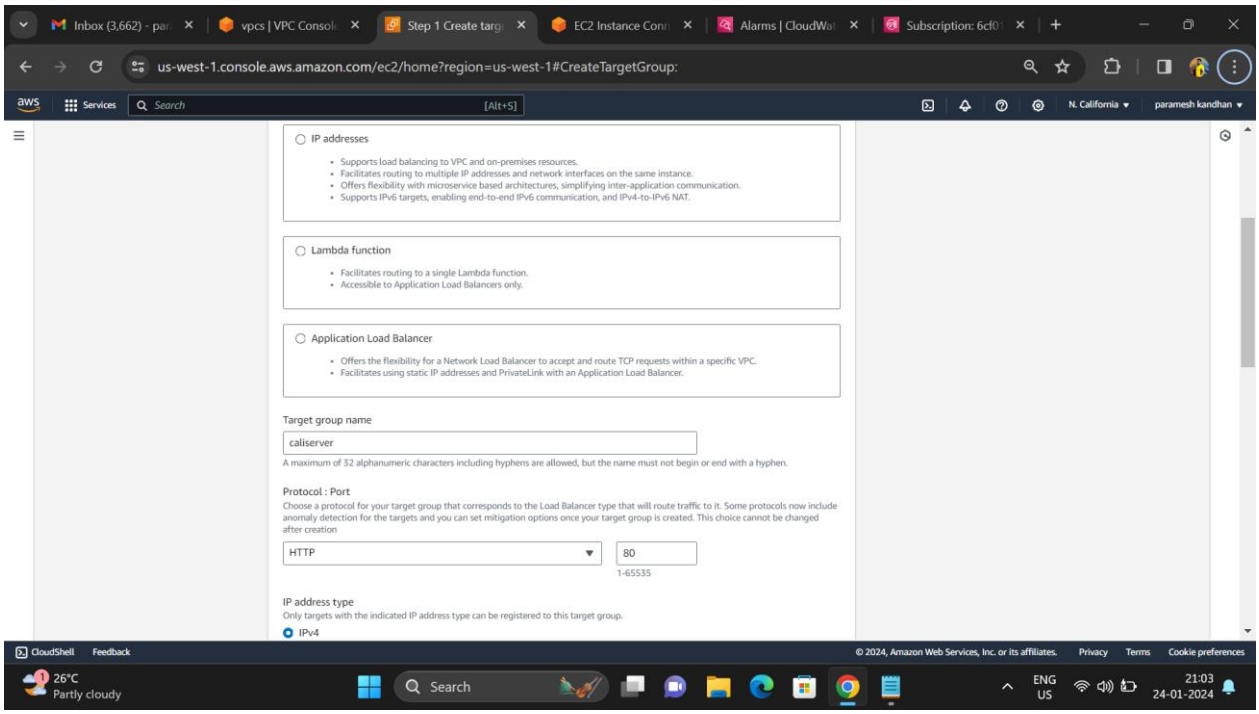


- And the server 1 it will be up in the server IP ad 54.69.95.233

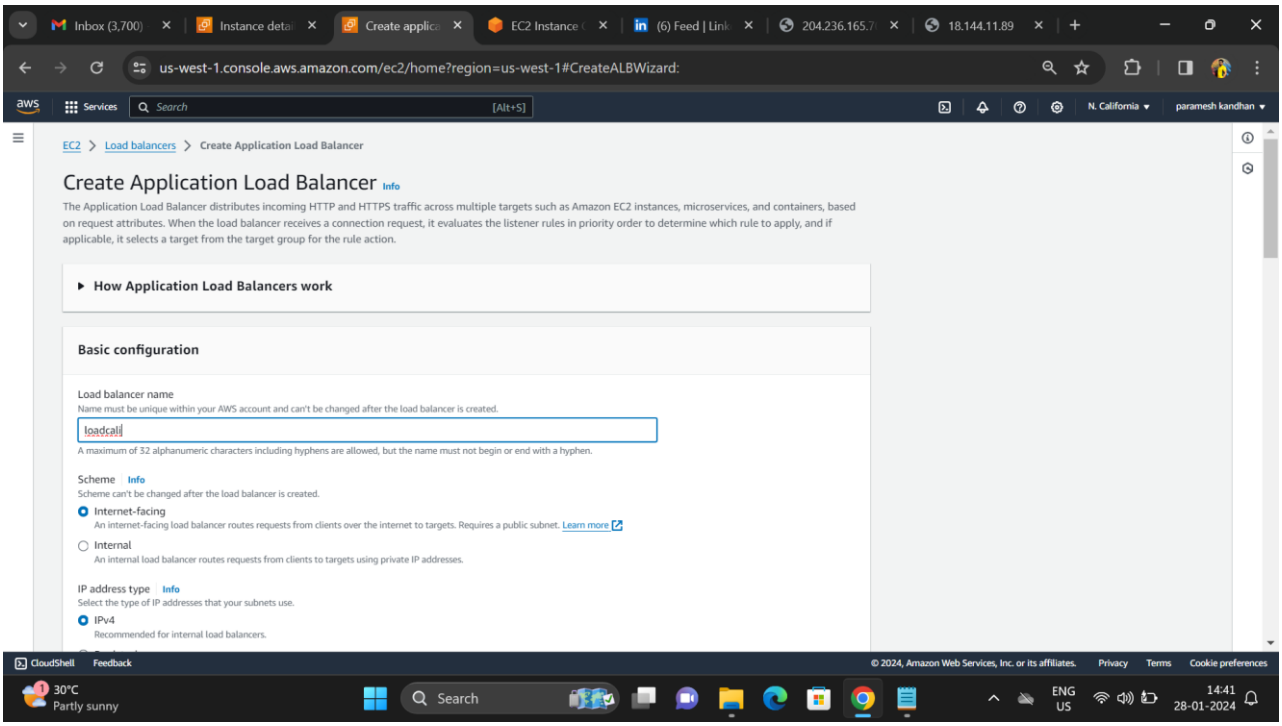


- And then again create a server for the purpose of the load balancer.
- Before create a load balancer need to be create a target groups.
- The target group is combine of two ec2 instances.
- A load balancer is a critical component in computer networking and web hosting that distributes incoming network traffic or application requests across multiple servers. The primary purpose of a load balancer is to ensure that no single server bears too much load, preventing potential performance bottlenecks and improving the overall reliability and availability of the application or website.
- Load balancers distribute incoming traffic across multiple servers based on various algorithms, such as Round Robin, Least Connections, or Weighted Round Robin. This helps evenly distribute the load and prevent overloading of any single server.

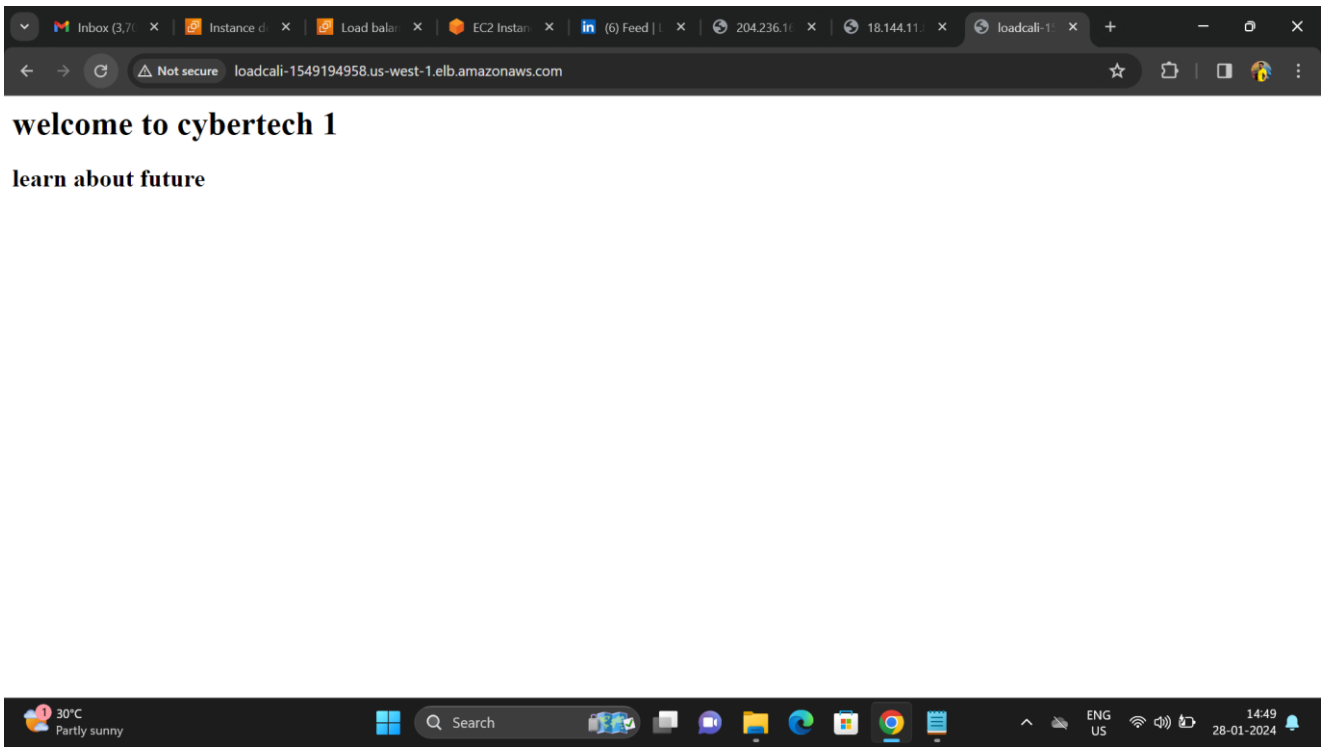
- To create a target group.

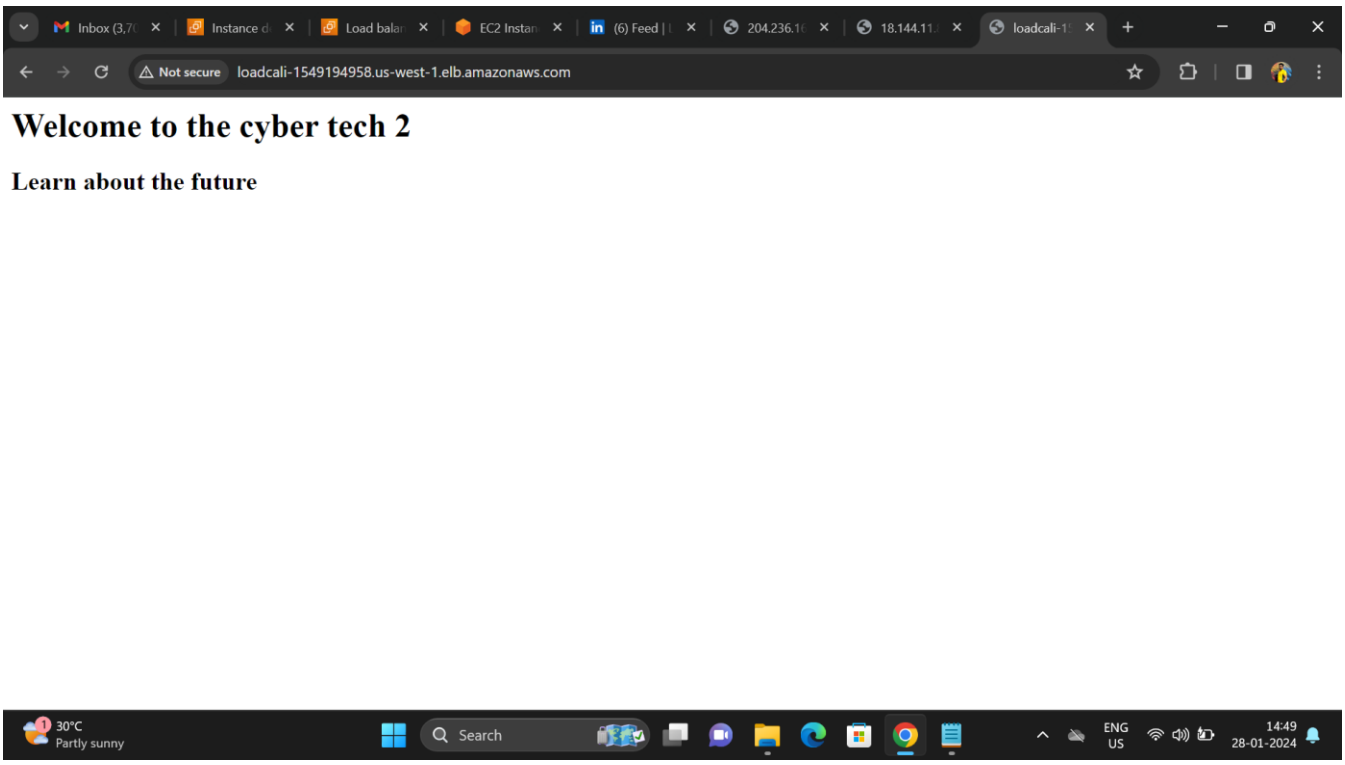


- And then create a load balancer.



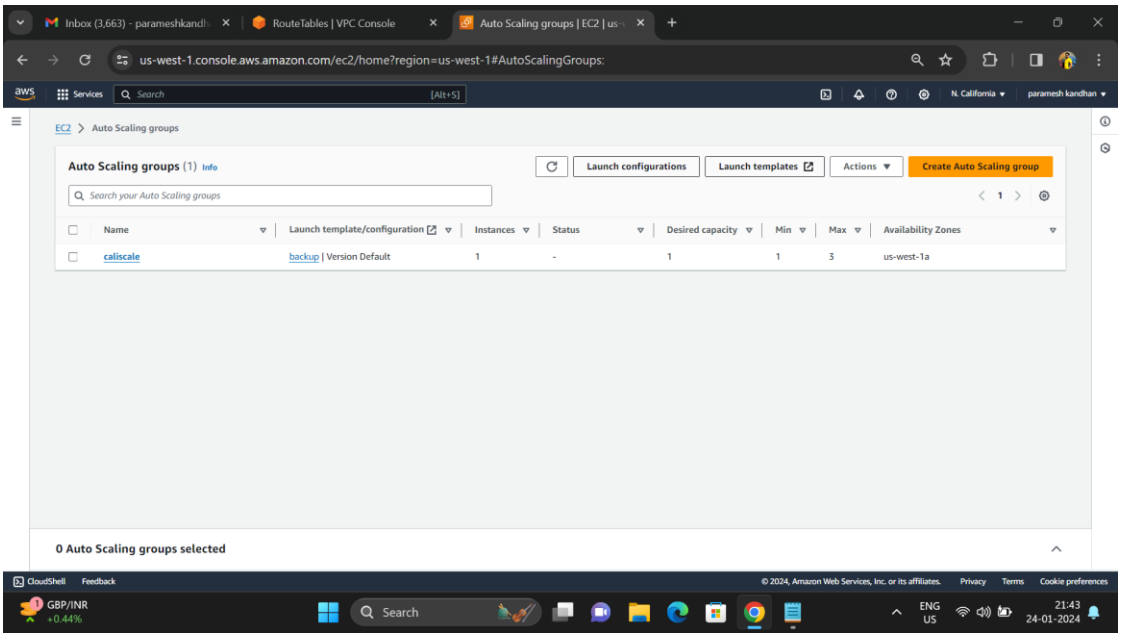
- It the output of load balancer,



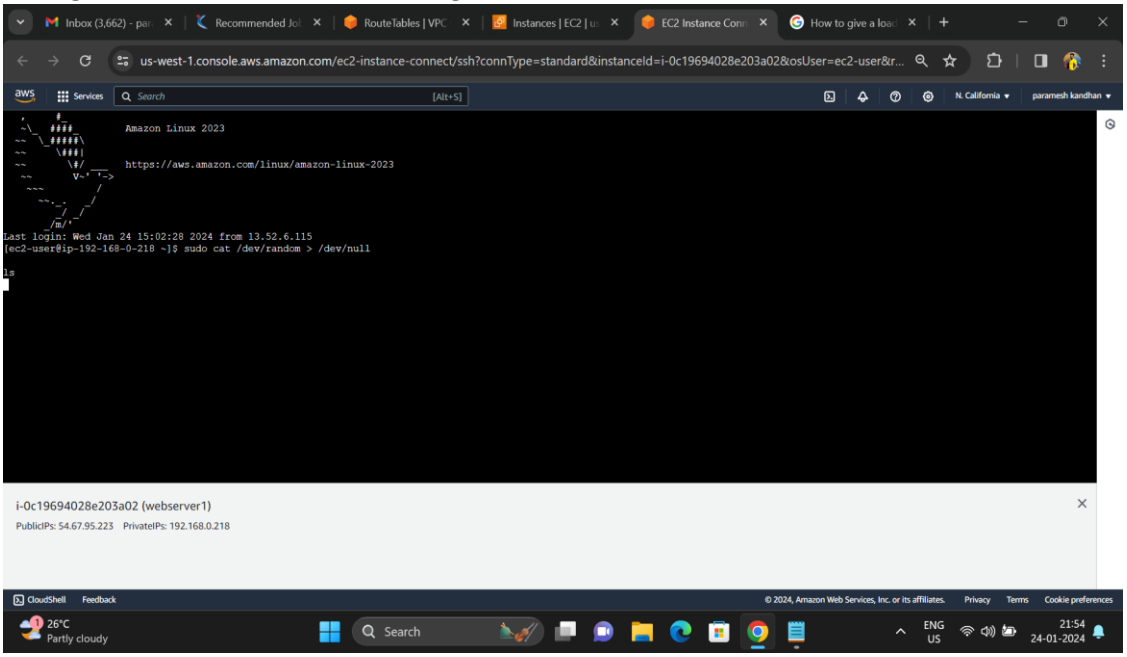


- In case the network traffic use a Auto Scaling helps you maintain application availability and allows you to scale your infrastructure dynamically based on predefined policies or schedules.
- The core component of Auto Scaling is the Auto Scaling Group. An Auto Scaling Group is a collection of Amazon EC2 instances that are created from a common Amazon Machine Image (AMI) and are designed to work together. The group automatically adjusts the number of instances in response to changes in demand.
- Scaling policies define when and how the Auto Scaling Group should adjust its capacity. There are two main types of scaling policies:

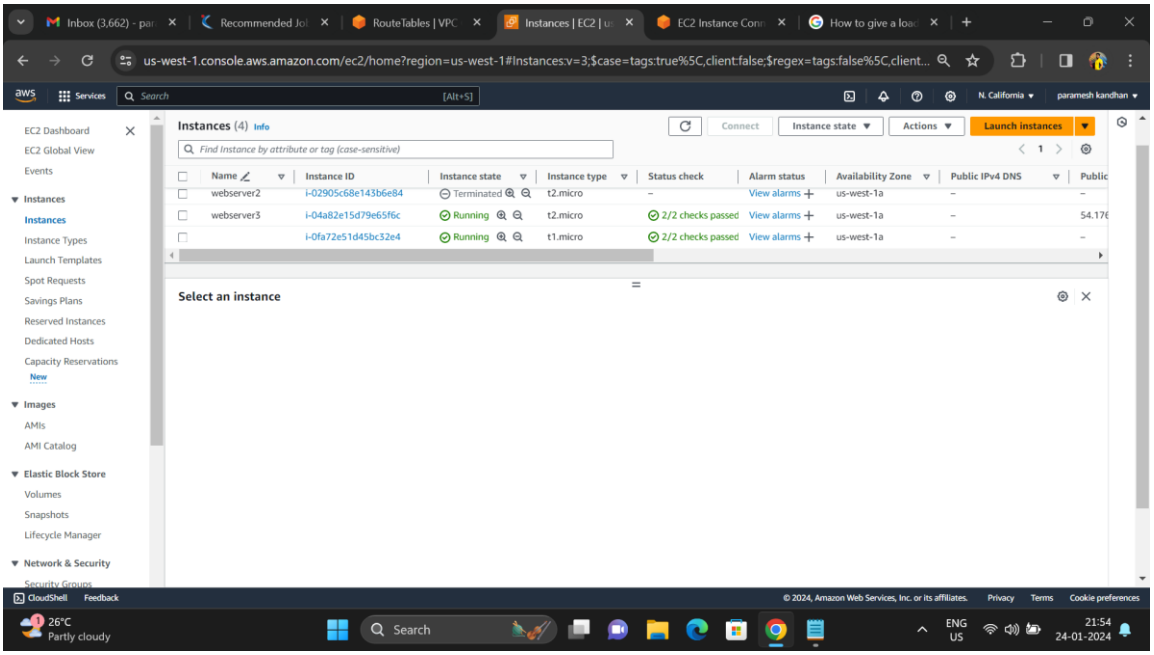
- Scale Out (Add Capacity): Triggered when a specified metric exceeds a certain threshold, leading to the addition of new instances.
- Scale In (Remove Capacity): Triggered when a specified metric falls below a certain threshold, resulting in the termination of instances.



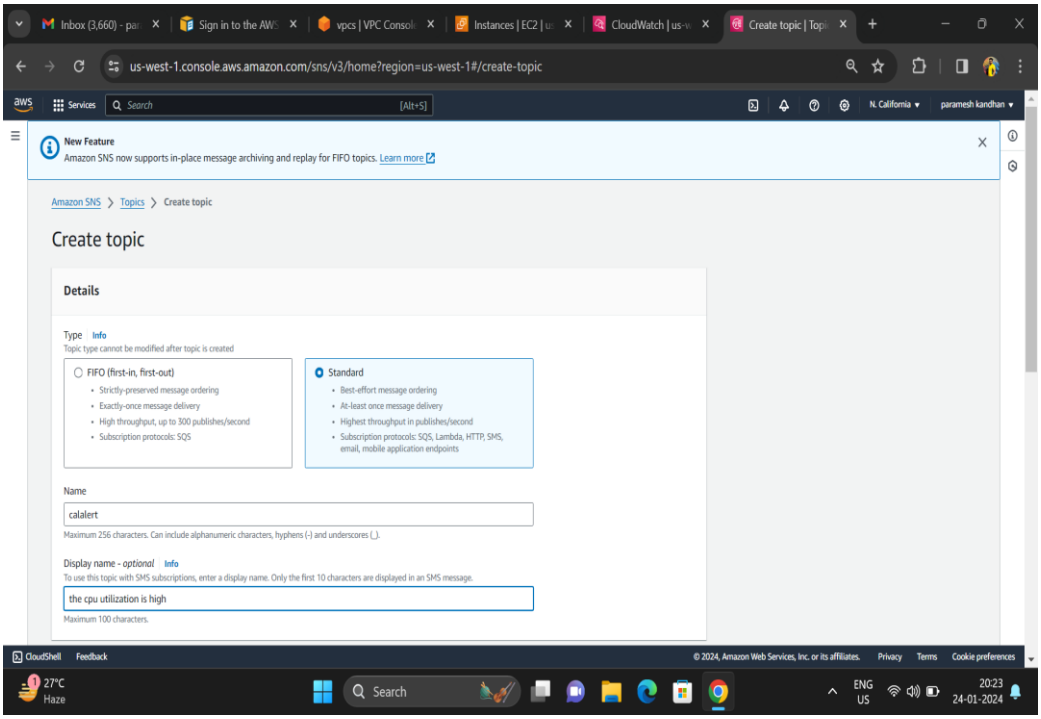
- Then given a artificial load to the using a command of “cat /dev/random > /dev/null/

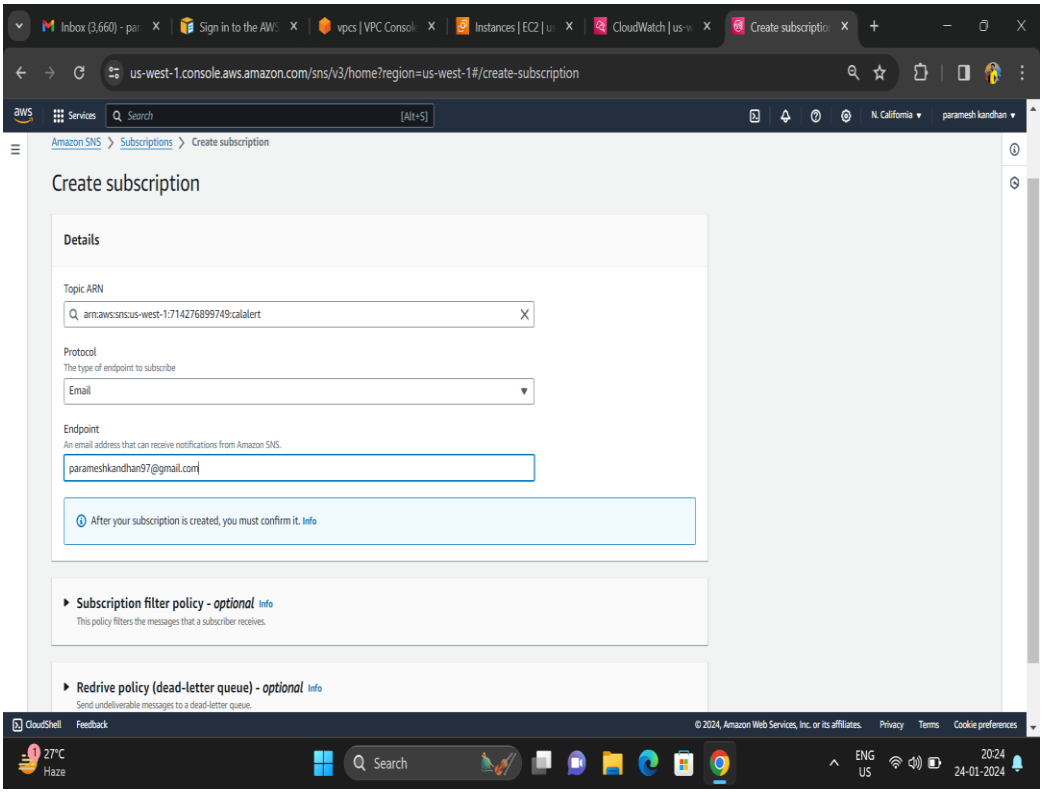


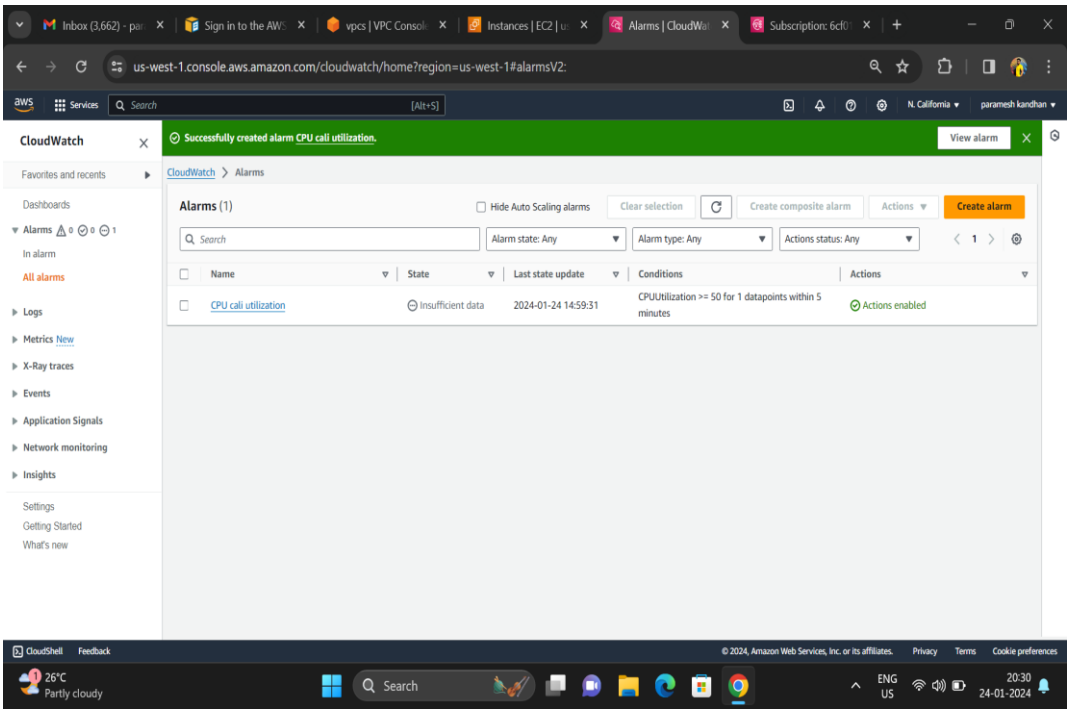
- Then it automatically create a new ec2 instance.



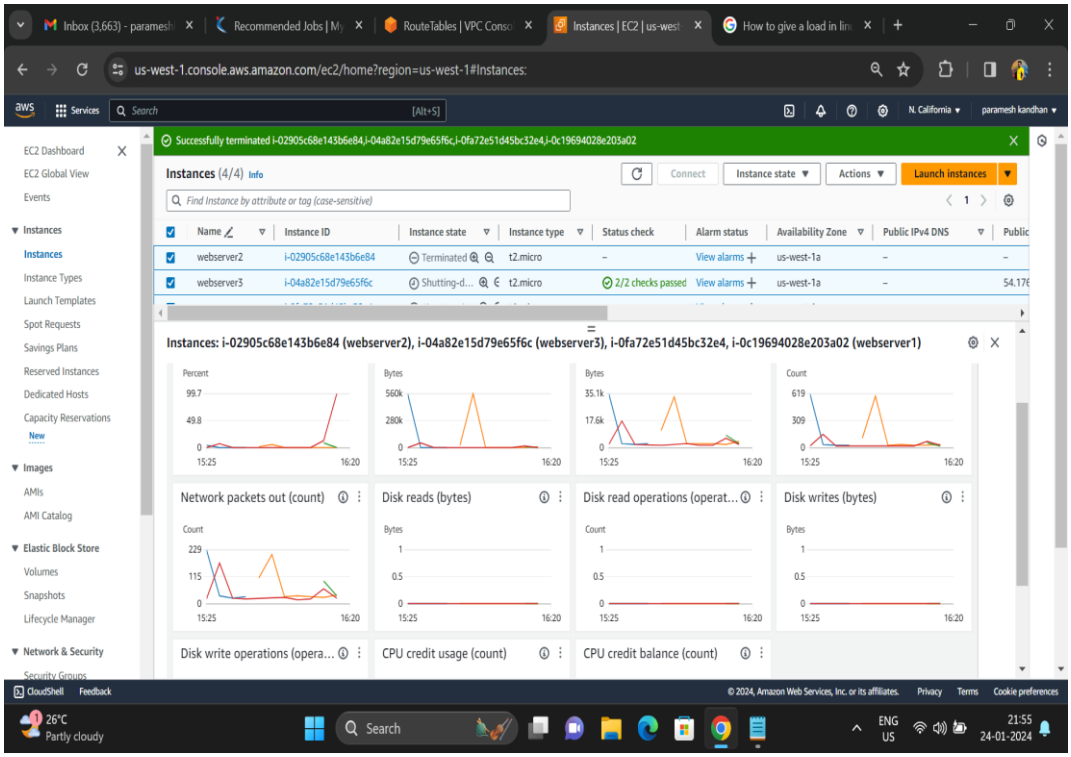
- For the monitoring purpose need to use the Amazon Cloud Watch and Amazon Simple Notification Service (SNS) are integral services in AWS that provide monitoring and notification capabilities, respectively. Combining these services can enable you to create professional and effective alerting and notification systems. Here's a guide on how to use CloudWatch and SNS in a professional way.
- Set up CloudWatch Alarms based on your custom metrics or predefined metrics for AWS resources. Alarms allow you to trigger actions when certain thresholds are breached.
- Integrate CloudWatch Logs to capture and analyze log data. Use CloudWatch Logs Insights for advanced querying and analysis of log data.
- To create a cloud watch need to be add the Organize subscribers using SNS Topics. Topics act as communication channels for sending messages to multiple endpoints.
- Subscribe your endpoints (email addresses, SMS numbers, HTTP endpoints, etc.) to relevant SNS Topics. This allows them to receive notifications.

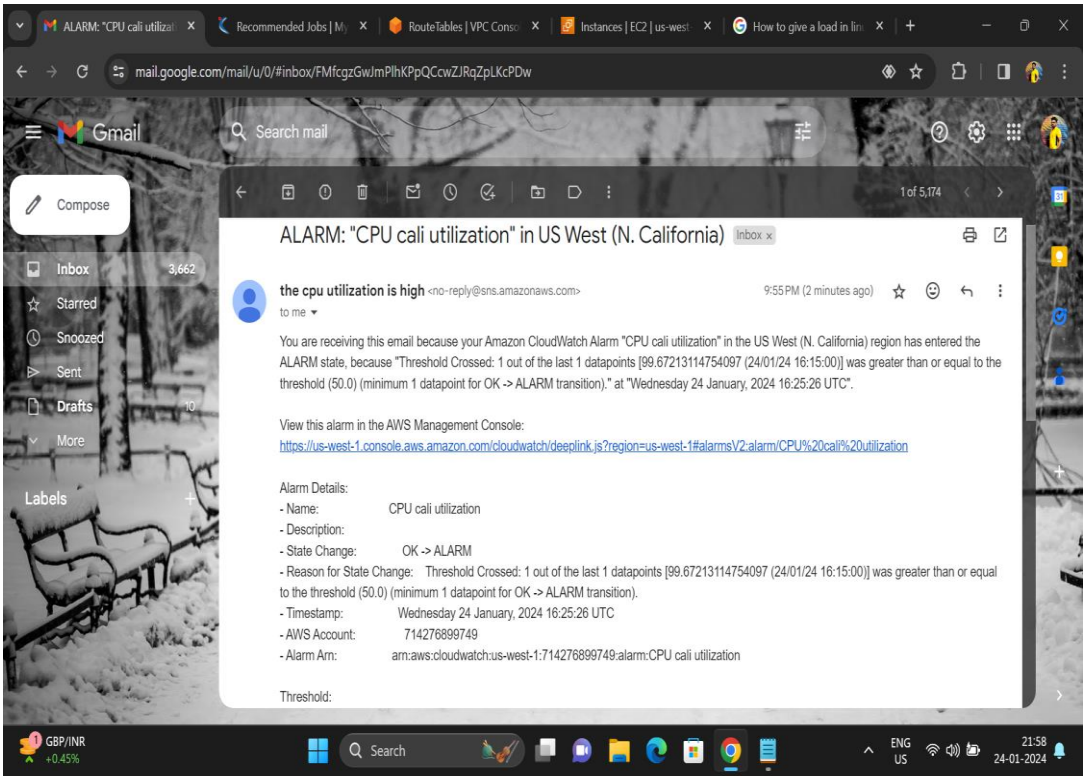






- Connect CloudWatch Alarms to SNS Topics for notification delivery. This ensures that when an alarm is triggered, relevant teams or individuals are notified via the chosen communication channels.
- If your application spans multiple AWS regions, use SNS for cross-region notifications. This ensures that teams in different regions are informed of any issues.
- The cpu utilization is a noticed and send through the cloud watch tab and send through the Mail.


us-west-1.console.aws.amazon.com/ec2/home?region=us-west-1#Instances:
Successfully terminated i-02905c68e143b6e84,i-04a82e15d79e65f6c,i-0fa72e51d45bc32e4,i-0c19694028e203a02
Instances (4/4) Info
webserver2 i-02905c68e143b6e84 Terminated t2.micro View alarms us-west-1a
webserver3 i-04a82e15d79e65f6c Shutting-d... t2.micro 2/2 checks passed View alarms us-west-1a
Instances: i-02905c68e143b6e84 (webserver2), i-04a82e15d79e65f6c (webserver3), i-0fa72e51d45bc32e4, i-0c19694028e203a02 (webserver1)
Network packets out (count)
Disk reads (bytes)
Disk read operations (operat...
Disk writes (bytes)
Disk write operations (opera...
CPU credit usage (count)
CPU credit balance (count)


mail.google.com/mail/u/0/#inbox/FMfcgzGwJmPlhKPpQCcwZJRqZpLKcPDw
ALARM: "CPU cali utilization" in US West (N. California)
the cpu utilization is high <no-reply@sns.amazonaws.com>
You are receiving this email because your Amazon CloudWatch Alarm "CPU cali utilization" in the US West (N. California) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [99.67213114754097 (24/01/24 16:15:00)] was greater than or equal to the threshold (50.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Wednesday 24 January, 2024 16:25:26 UTC".
View this alarm in the AWS Management Console:
https://us-west-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-west-1#alarmsV2:alarm/CPU%20cali%20utilization
Alarm Details:
- Name: CPU cali utilization
- Description:
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [99.67213114754097 (24/01/24 16:15:00)] was greater than or equal to the threshold (50.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Wednesday 24 January, 2024 16:25:26 UTC
- AWS Account: 714276899749
- Alarm Arn: arn:aws:cloudwatch:us-west-1:714276899749:alarm:CPU cali utilization
Threshold: