**AWS Task-3**

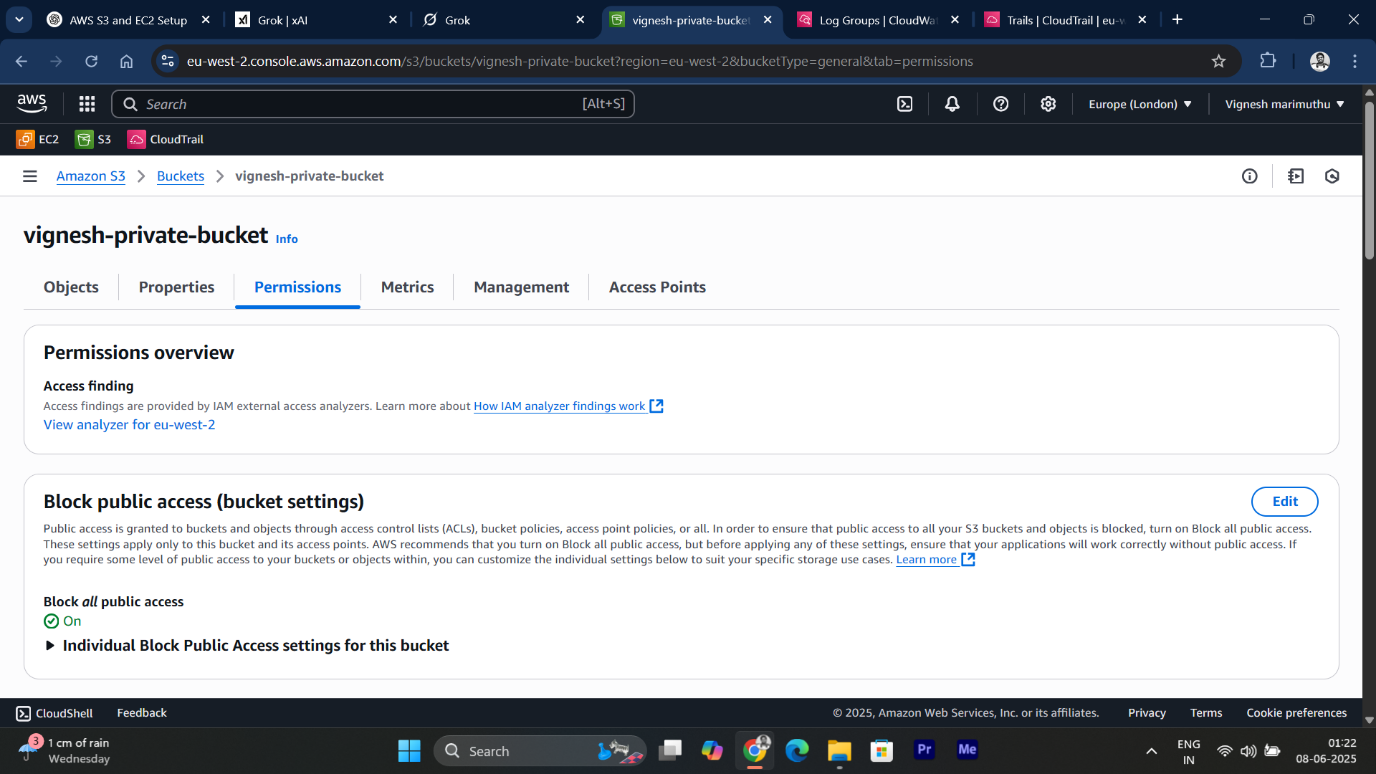
**Task Description:**

1.Create a S3 bucket, with no public access and upload files to the bucket & view the logs using cloudwatch for the uploaded files.

2.Launch two ec2-instances and connect it to a application load balancer, where the output traffic from the server must be an load balancer IP address.

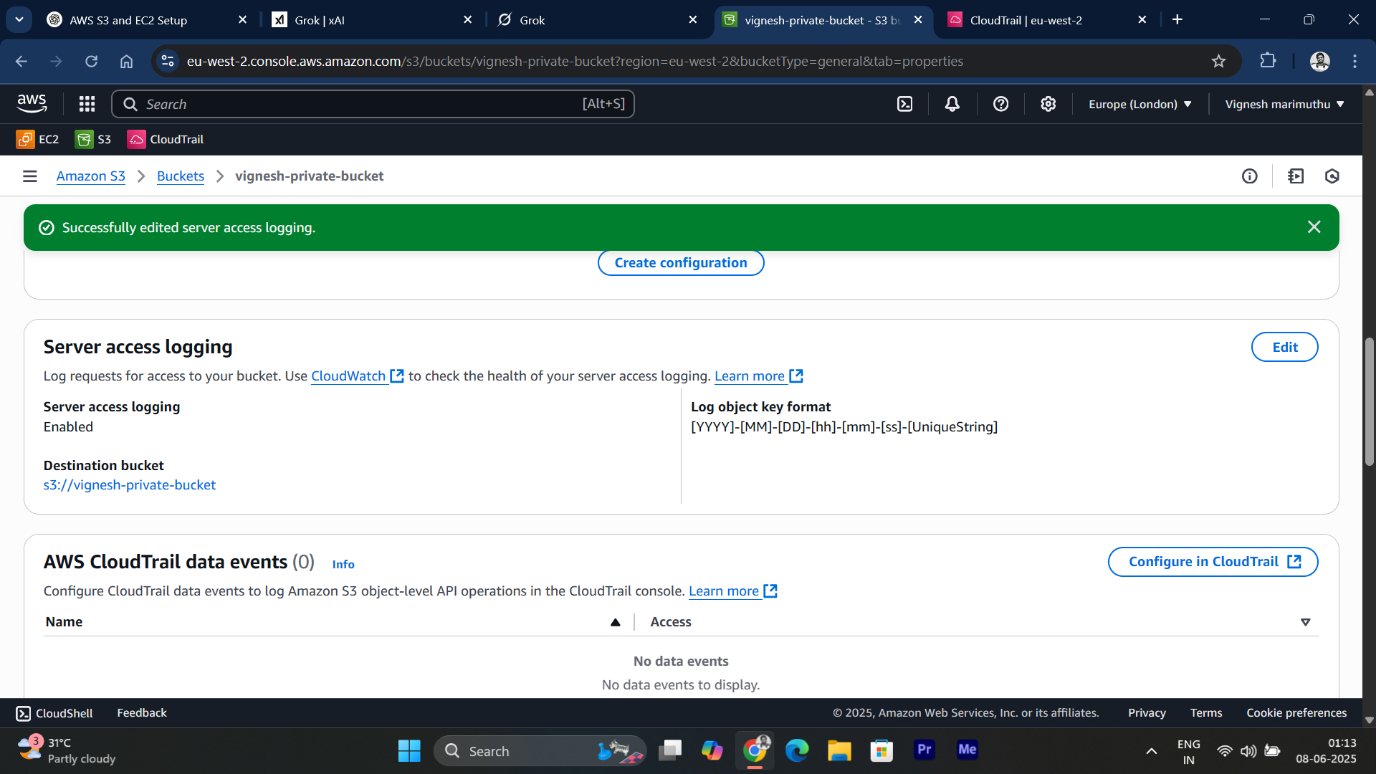
**Solution : 1**

**✅ 1.** **Create an S3 bucket (with no public access)**

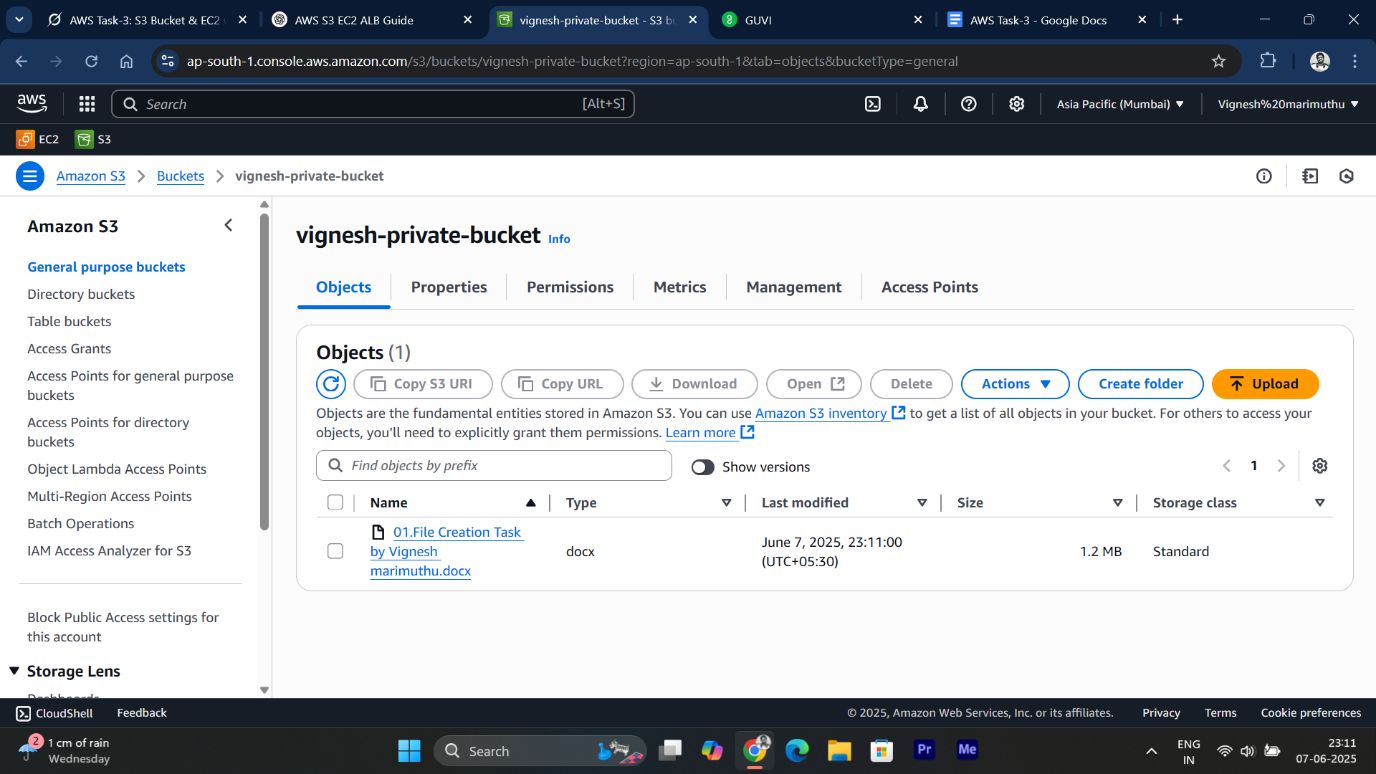
****

* Successfully created bucket with no public access for security

**✅ 2.** **Enabling Access logging for monitor logs in Cloudwatch**

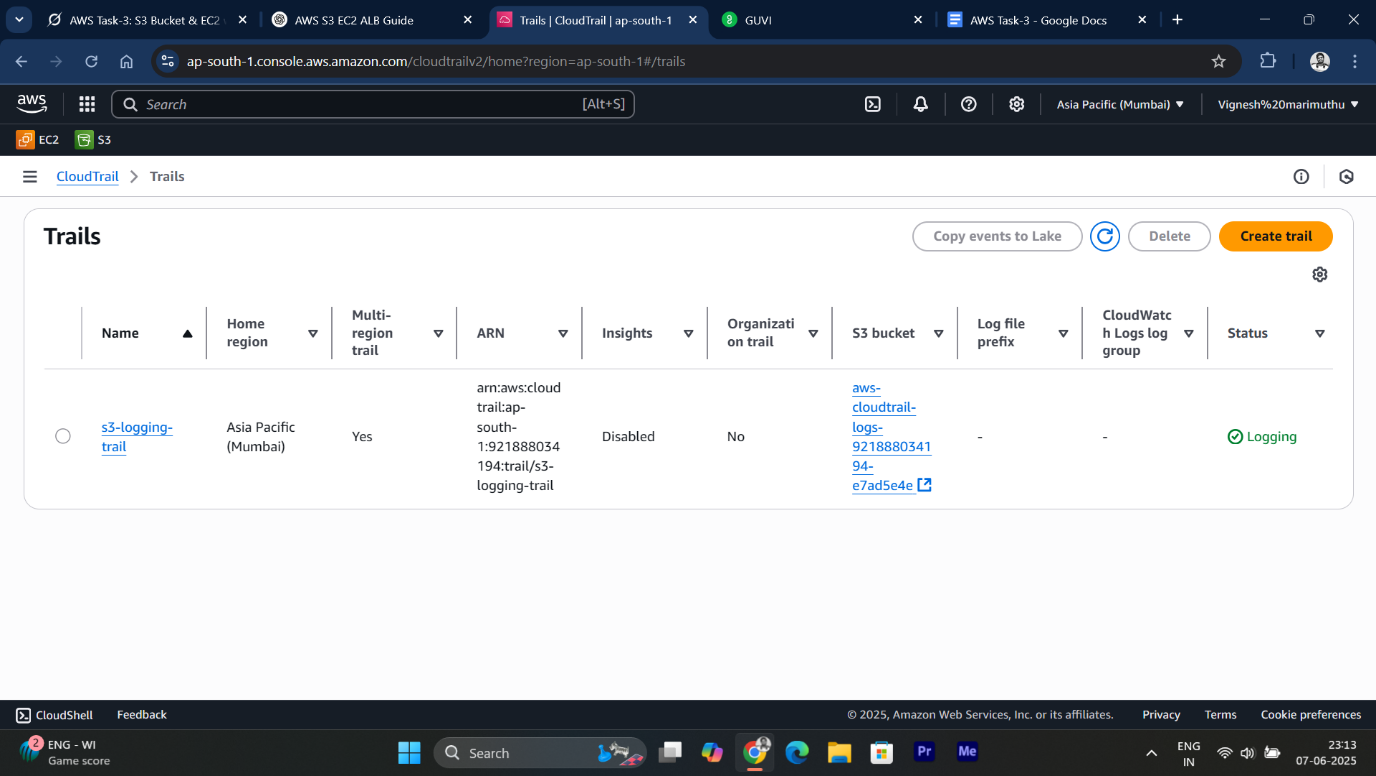
****

* This feature records details of requests made to an S3 bucket, such as uploads, downloads, or deletes.
* Each request generates a log entry with information like the requester, timestamp, action, and response status.

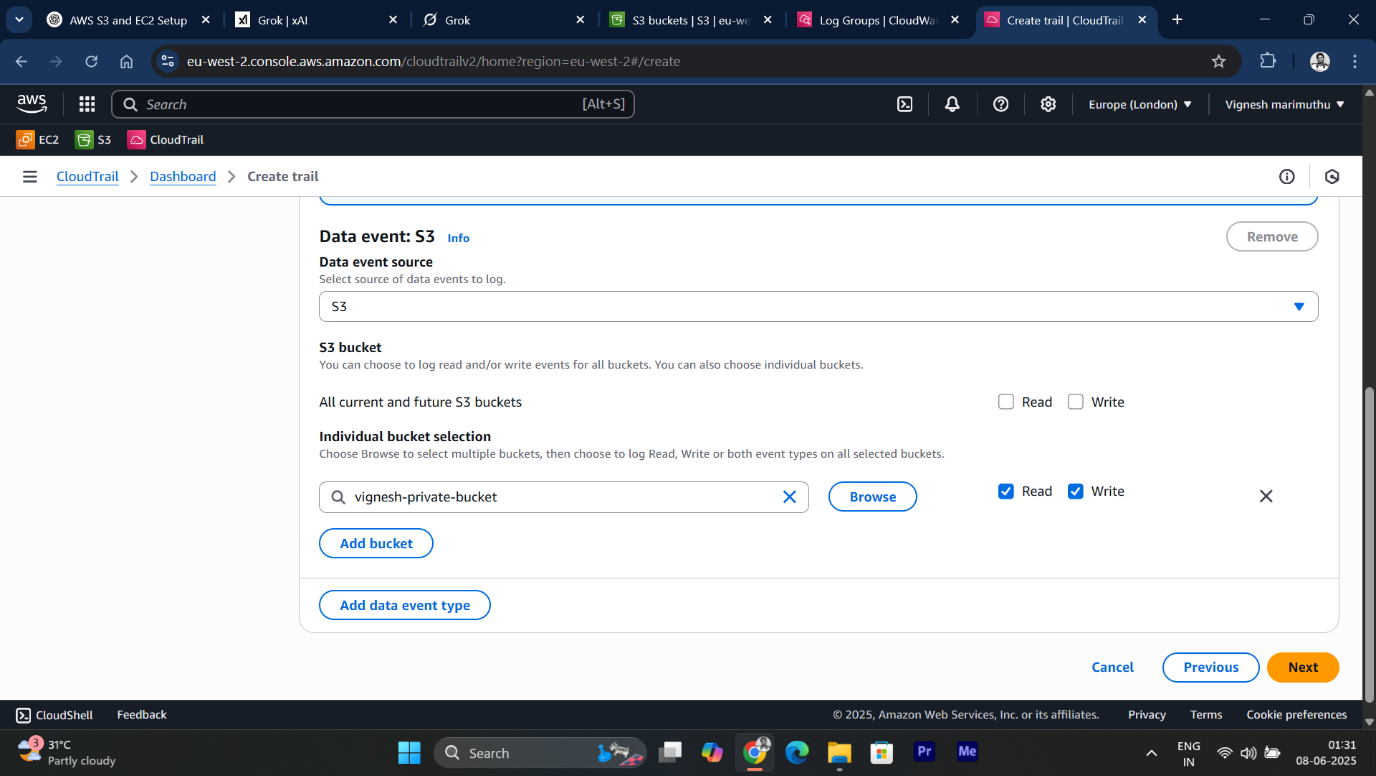
**✅ 3.** **Upload files to the bucket**

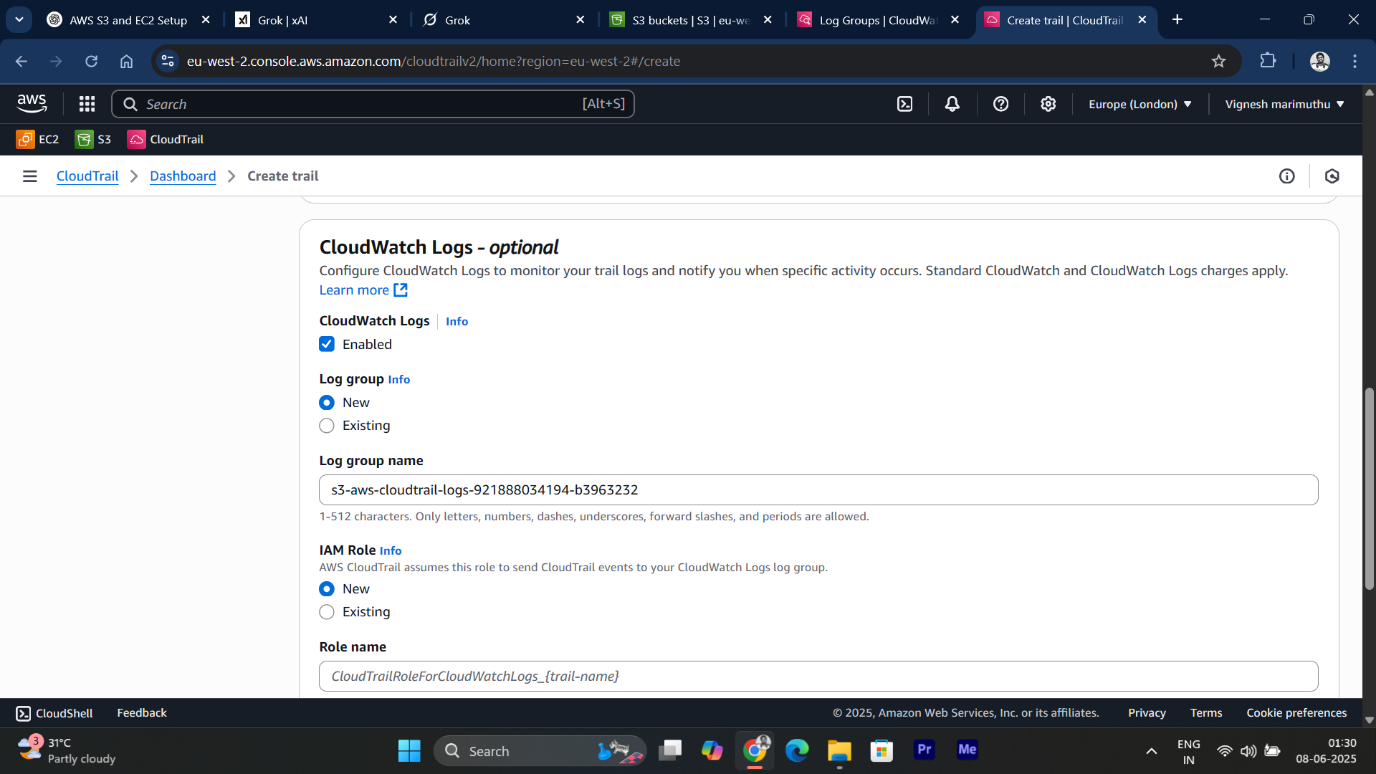
* files are stored in S3 Bucket

**✅ 4.** **View logs in CloudWatch**

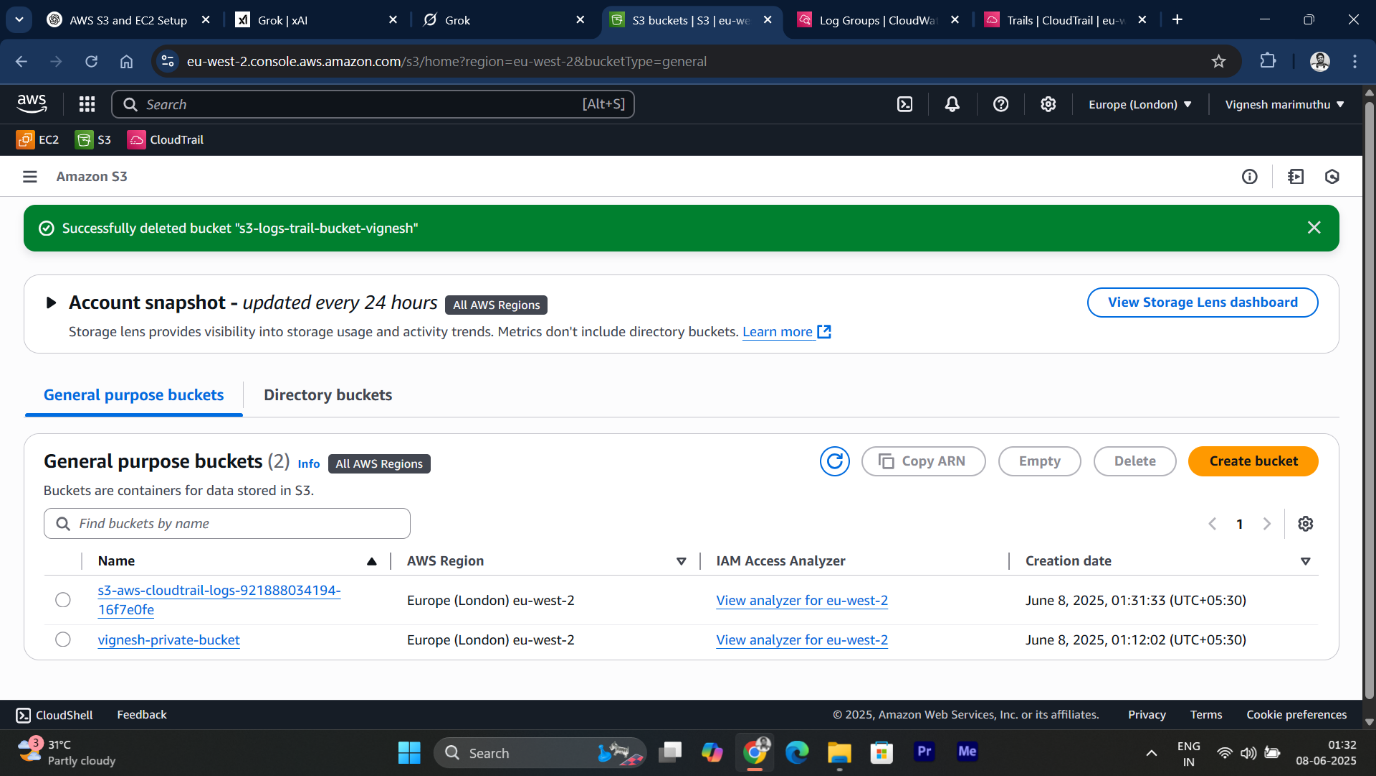
****

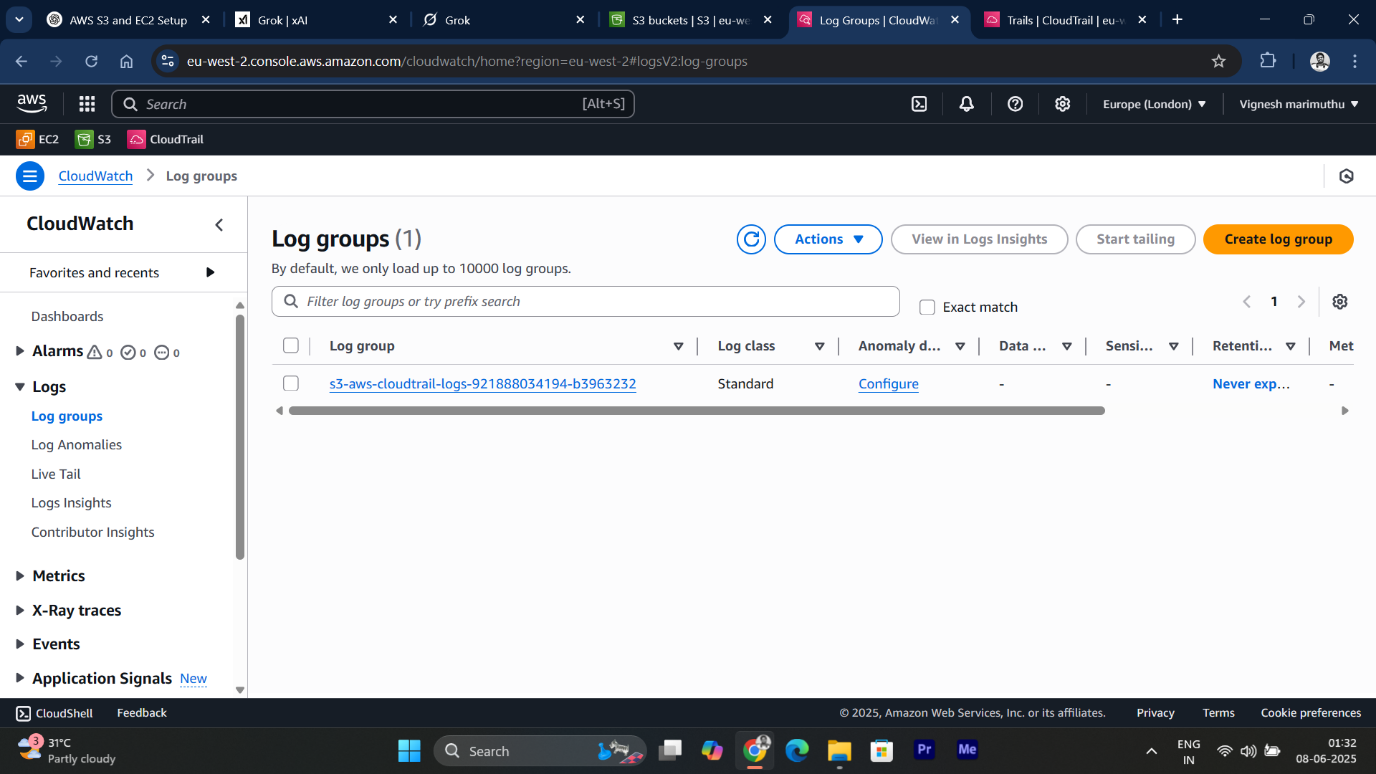
* I created a new CloudTrail.
* CloudTrail sends the S3 events to CloudWatch Logs → we can monitor "who uploaded what and when"



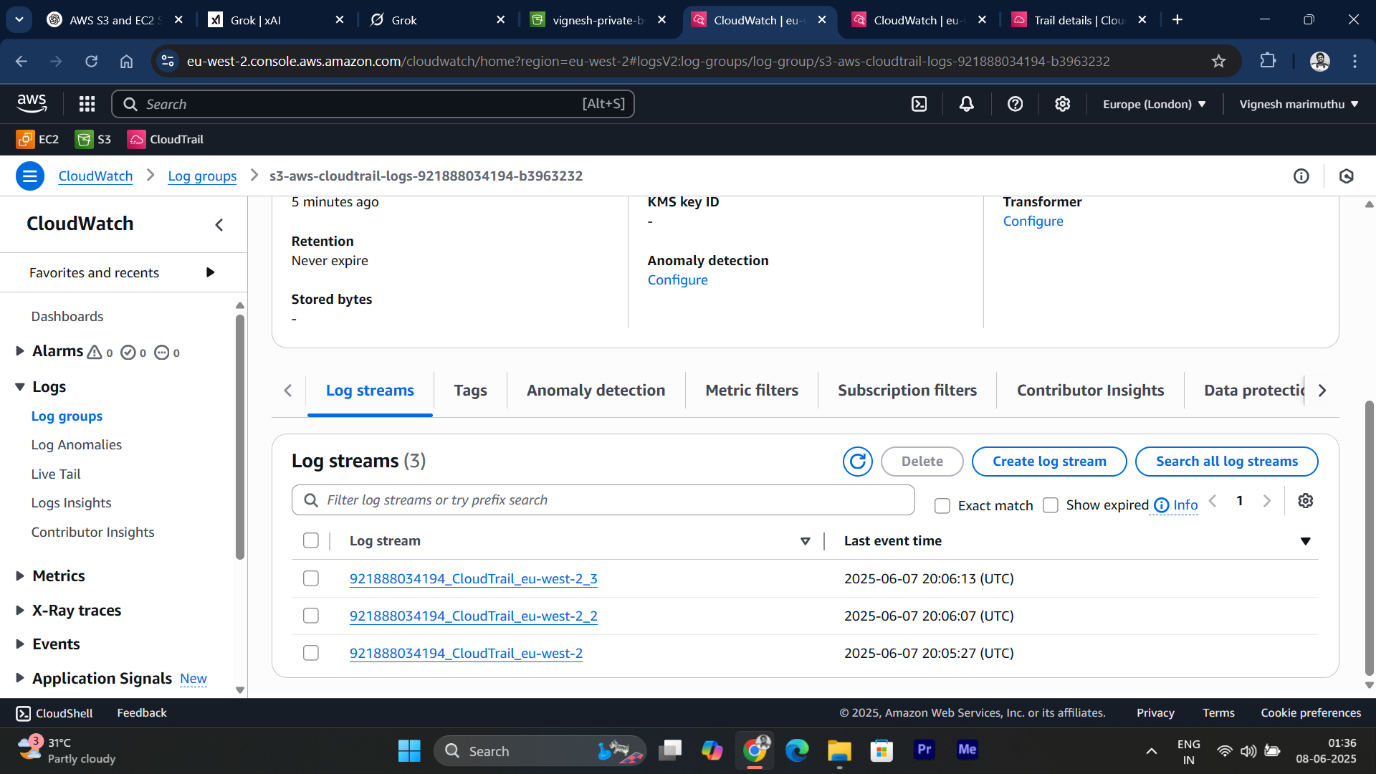


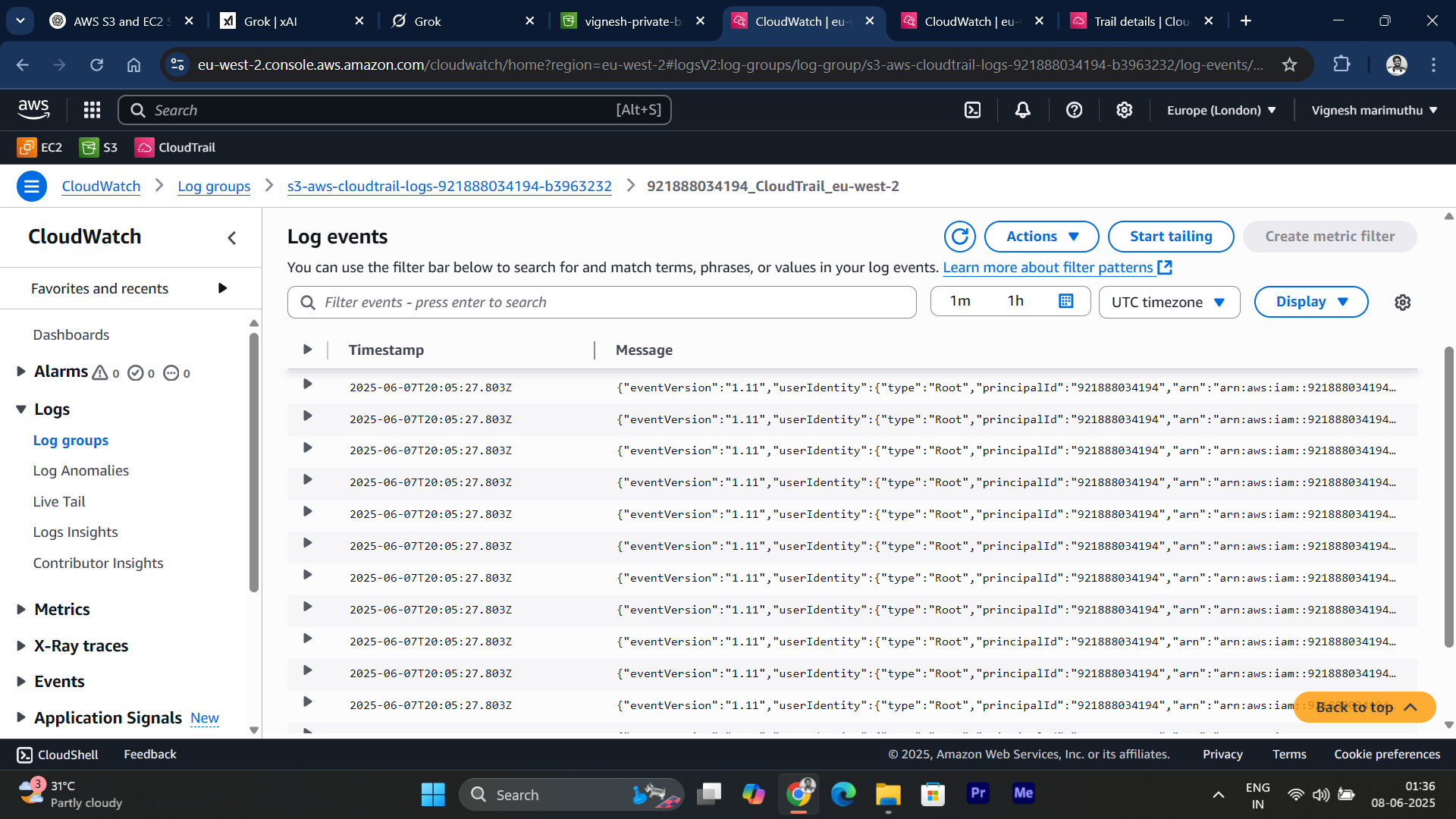
* While creating the Trail, I enabled it to capture **my S3 bucket actions** and configured it to send the logs to **CloudWatch Logs** for easy monitoring.





* After the Trail was set up, I uploaded a new file to my S3 bucket to generate activity.

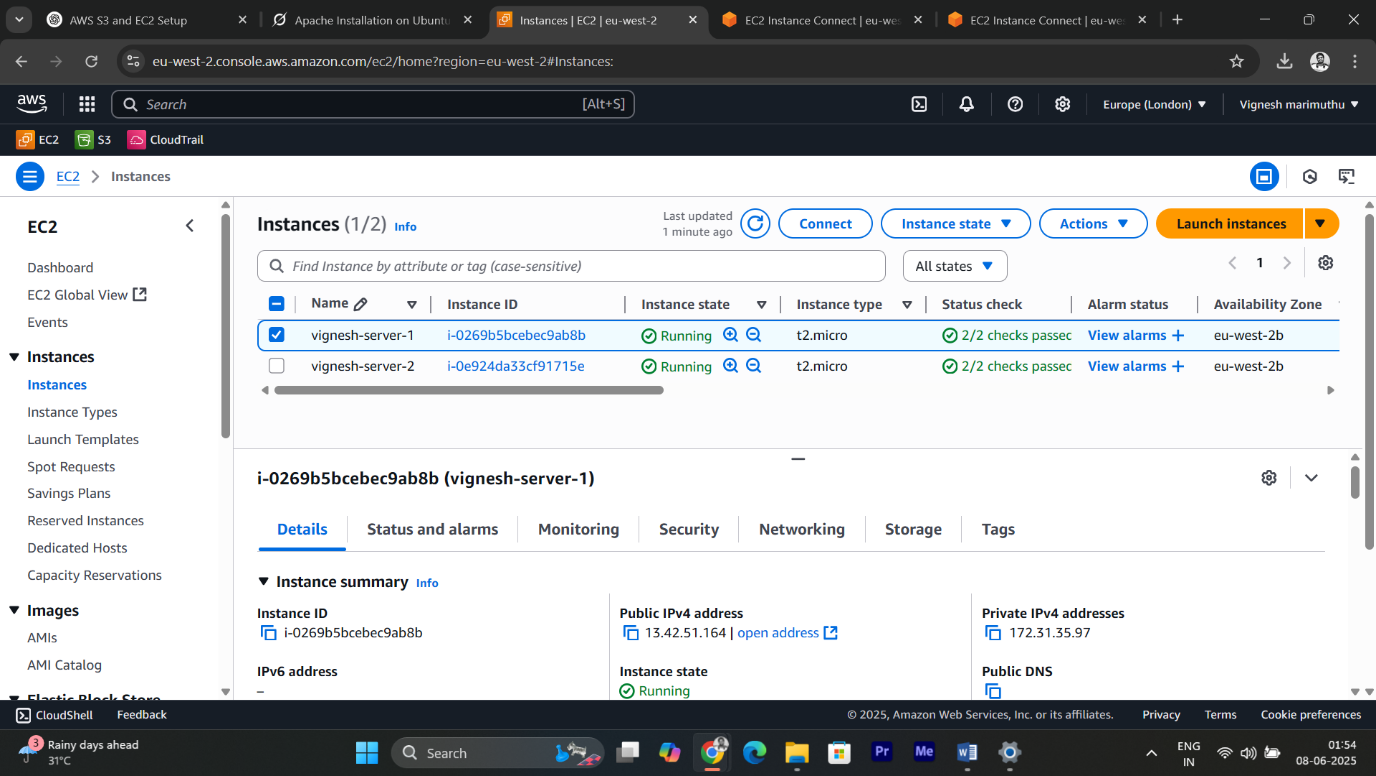


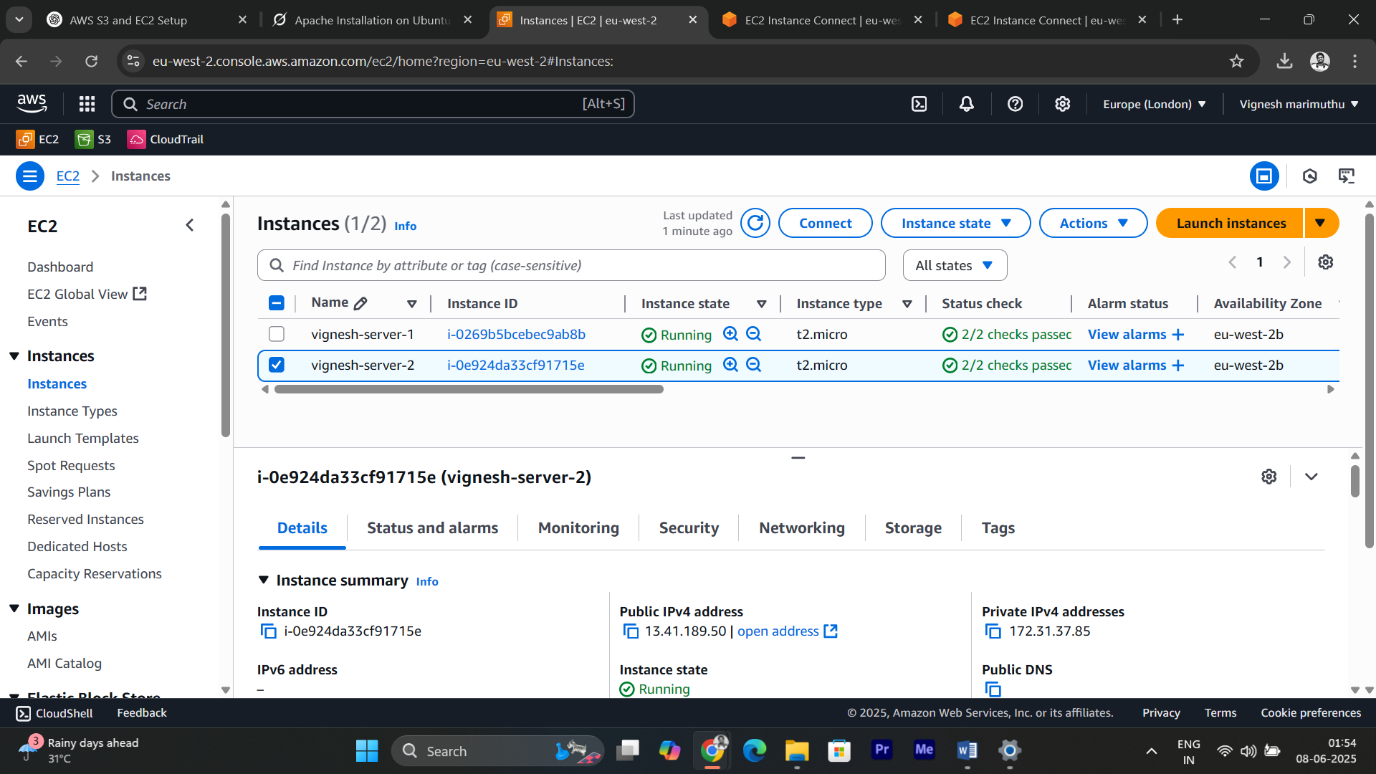


* Finally, I went to the **CloudWatch** service, opened **Log Groups**, selected the Log Group linked to my CloudTrail, and viewed the recorded events, which included the S3 PutObject action corresponding to my file upload.

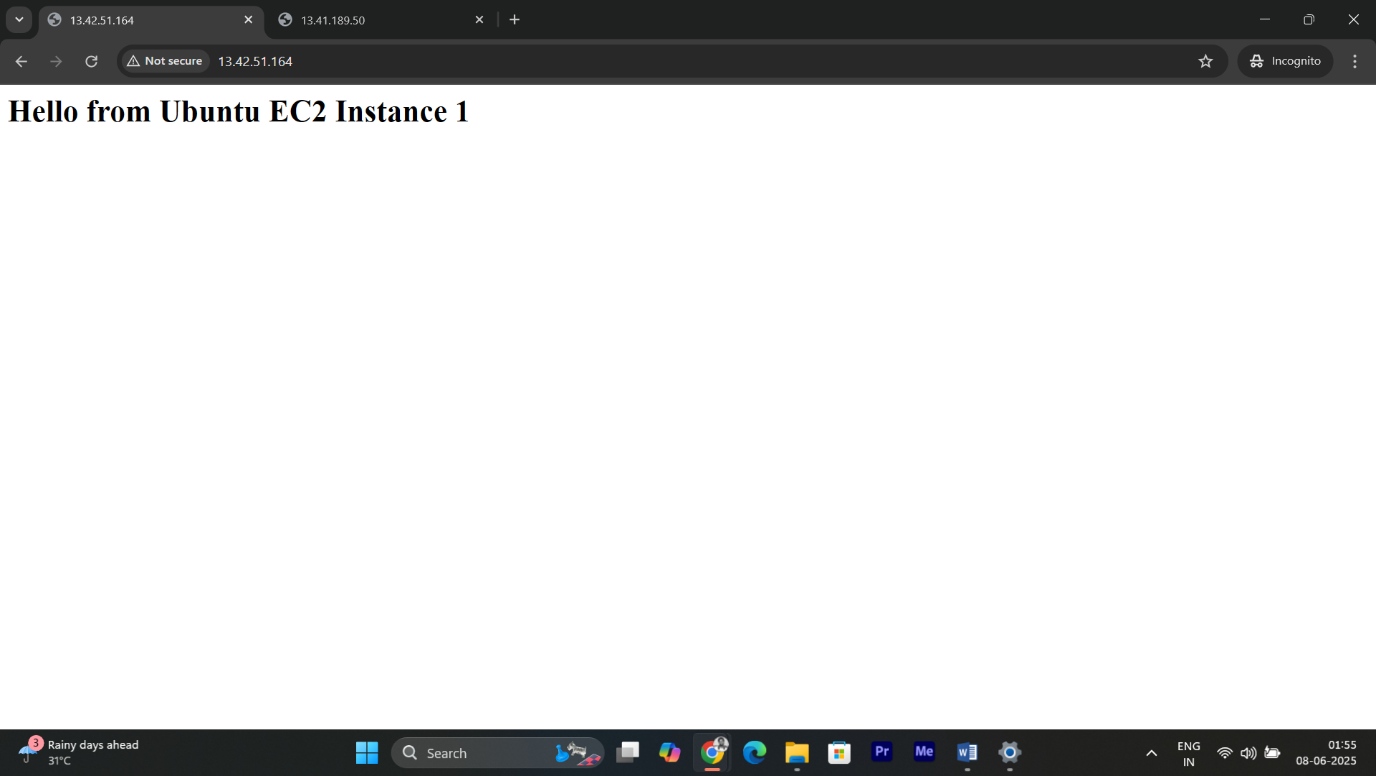
**Solution : 2**

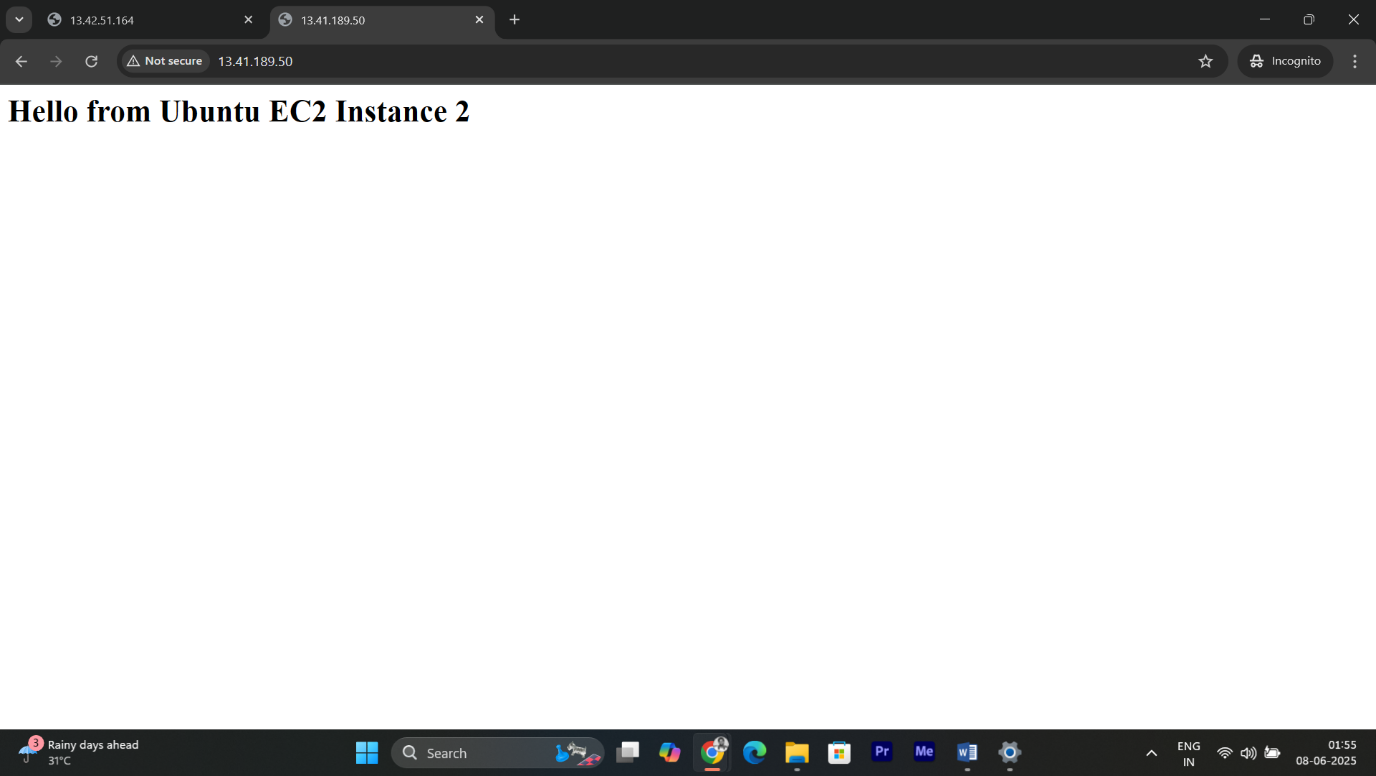
**✅ 1.** **Launch 2 EC2 instances with Ubuntu**

****

****

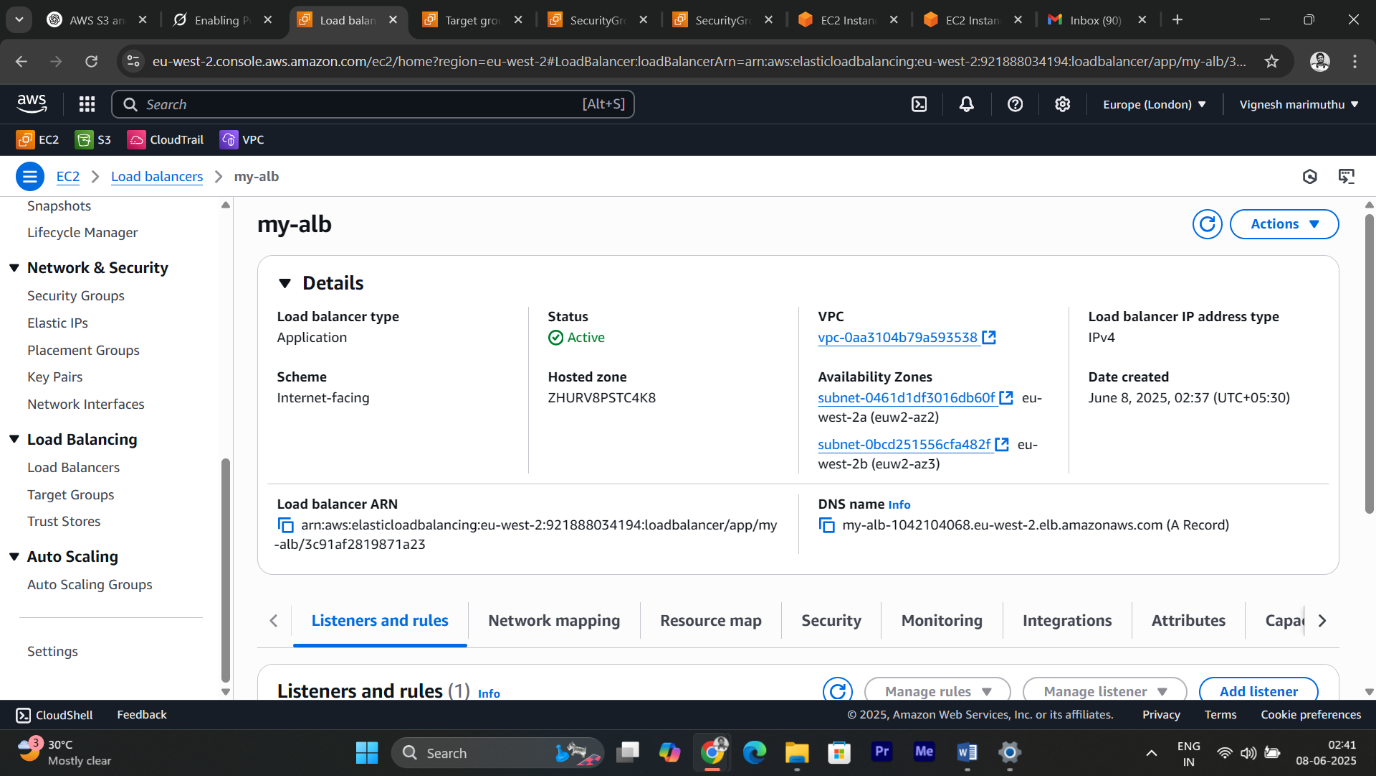
* I launched **two EC2 instances** using **Ubuntu**
* After launching the instances, I connected to each one.
* On both instances, I updated the package list and installed the **Apache web server.**



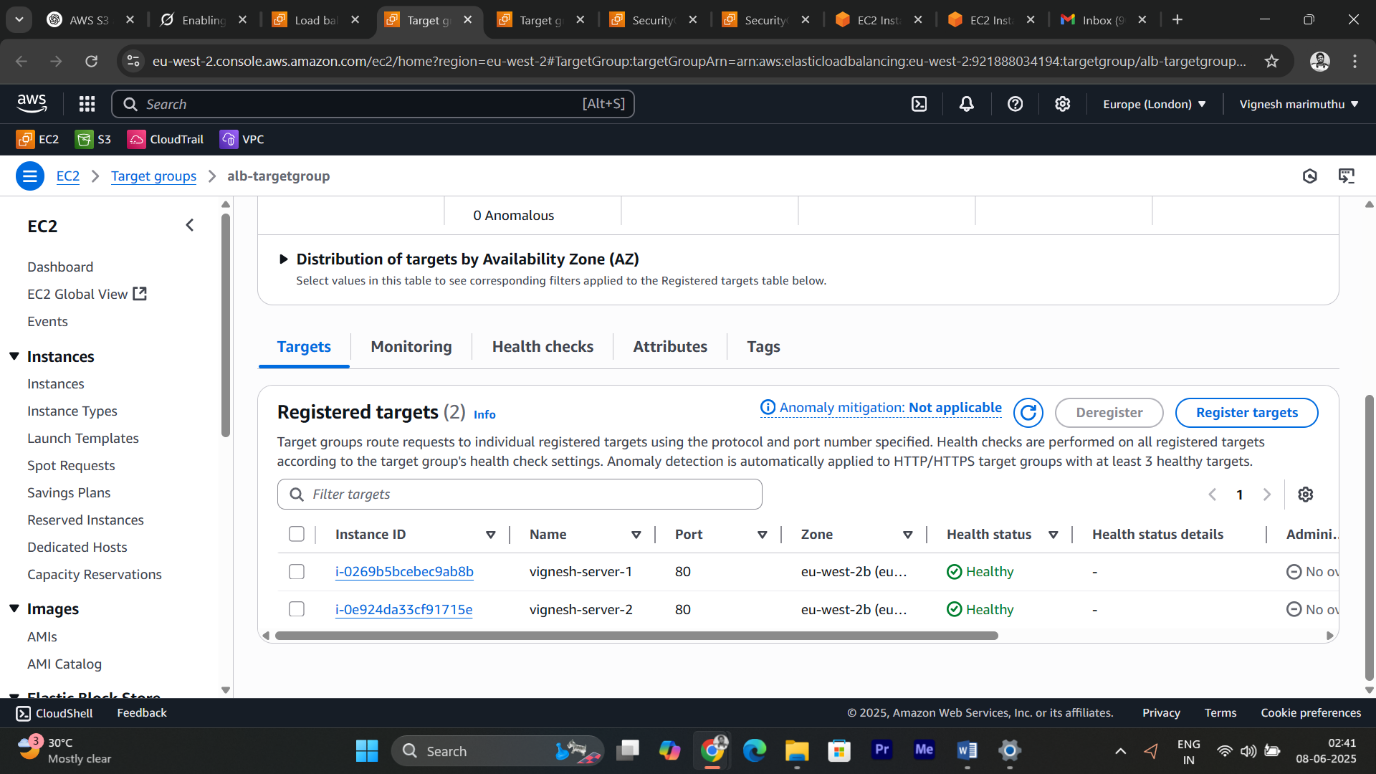


* I created a simple HTML page on each instance to display a unique message.

**✅ 2.** **Create Application Load Balancer (ALB)**

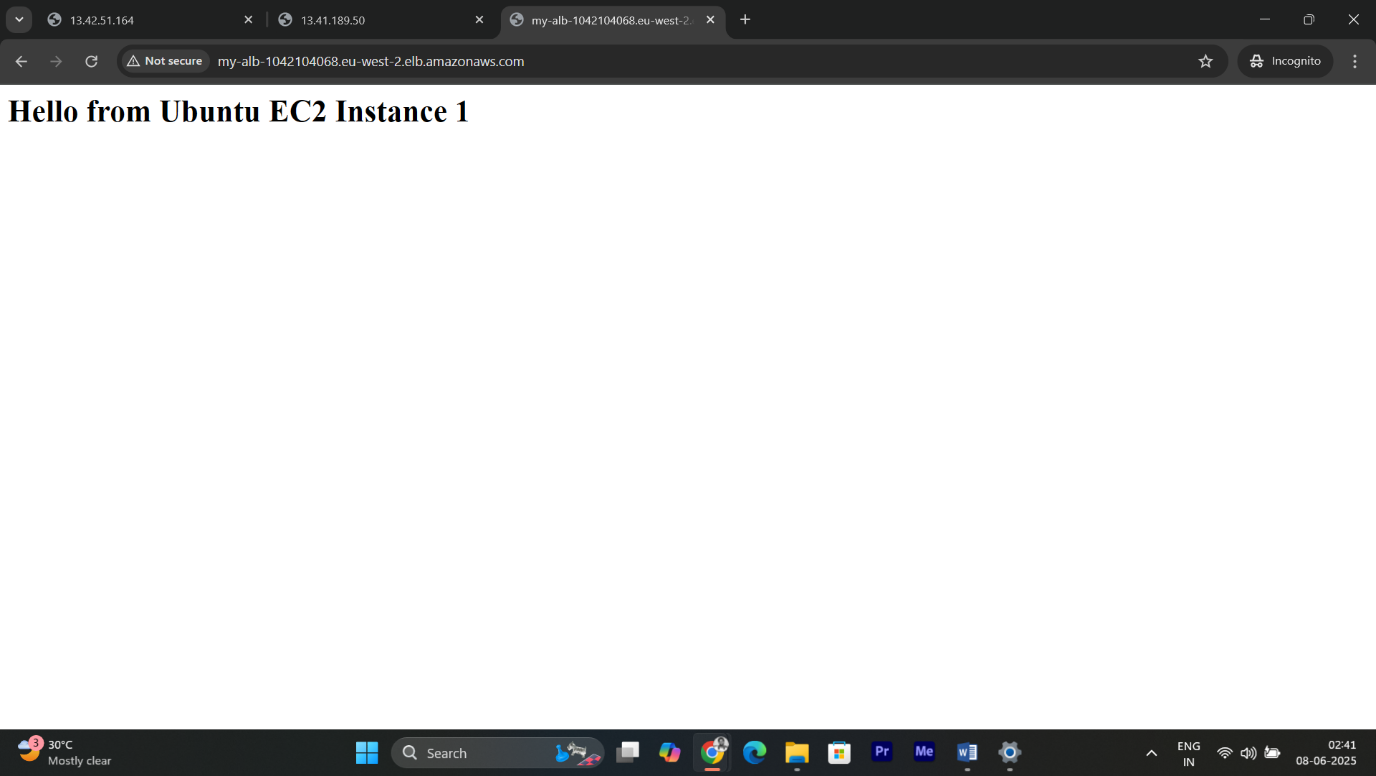
****

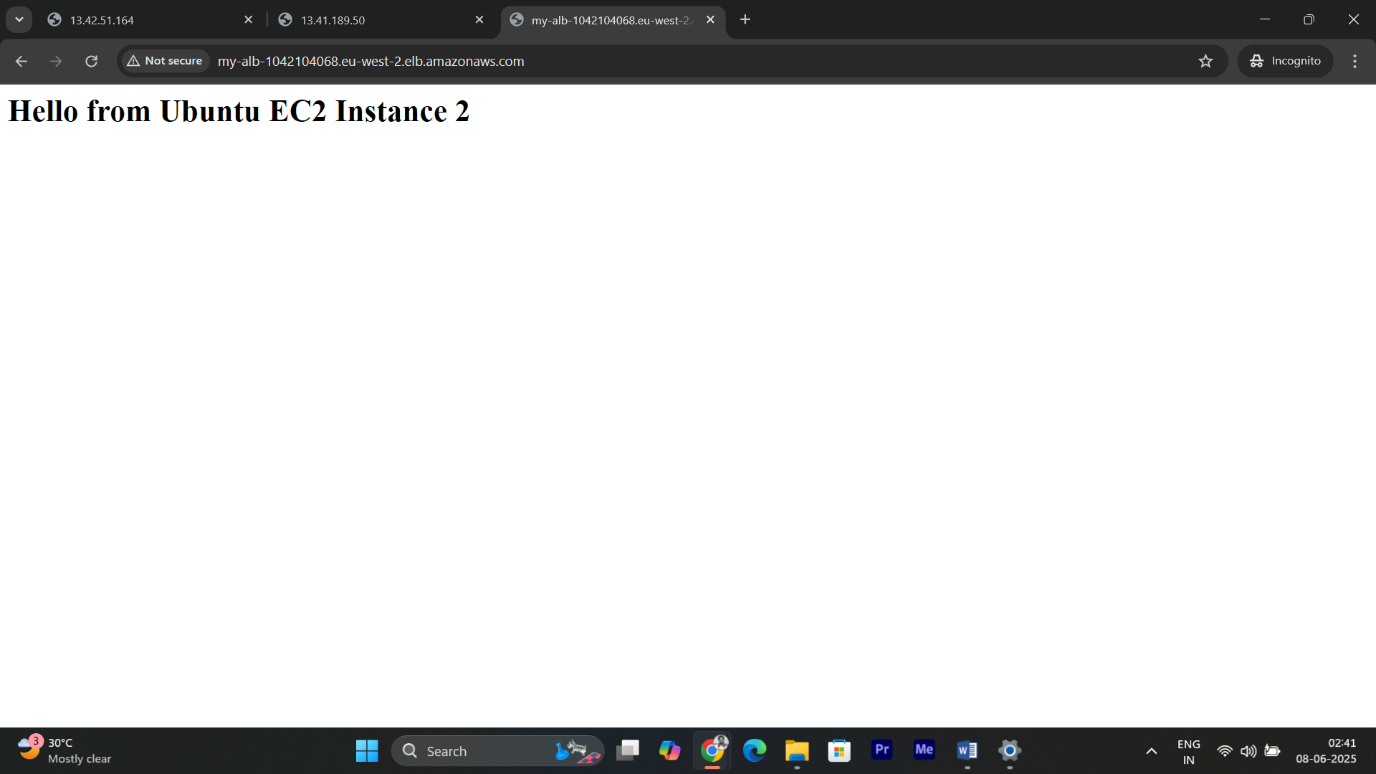
* After configuring the web servers, I proceeded to created an **Application Load Balancer (ALB)**.



* While creating **ALB** also created a new **Target Group** of type **Instance**, registered both of my EC2 instances with the Target Group.

**✅ 3.Test Application Load Balancer(ALB)**





* Once the ALB was created and became active, I copied its **DNS name** and accessed it via a web browser. Upon refreshing the page multiple times, I was able to observe that the ALB was successfully balancing the traffic between the two EC2 instances -as the message alternated between "Hello from Ubuntu EC2 Instance 1" and "Hello from Ubuntu EC2 Instance 2".
* **Application Load Balancer** was functioning correctly, and that users were accessing the EC2 instances through the ALB’s DNS, without seeing the individual EC2 IP addresses.