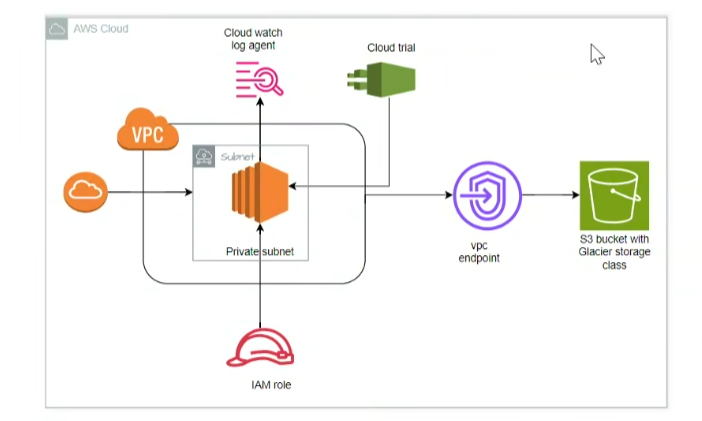
**Case Study-6:**

**Problem Statement:** You have been asked to create a **shell script** on AWS Linux EC2 instance to read all files from an AWS S3 bucket. However, accessing the images in the S3 bucket **must not be over the internet** and resist **public access**. Implement a **cost-effective, secure and highly available** solution for this scenario. Also, **centralize** the collection of operating system logs like var/log/messages for all your EC2 instances using CloudWatch. Setup AWS service to log **task/activity performs** on the EC2 instance and S3 bucket.

****

created an S3 bucket to store CloudTrail logs. To ensure security,

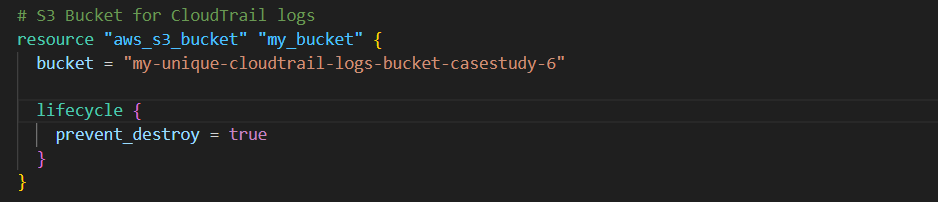
**Public Access**: used the aws\_s3\_bucket\_public\_access\_block resource to block any public access.

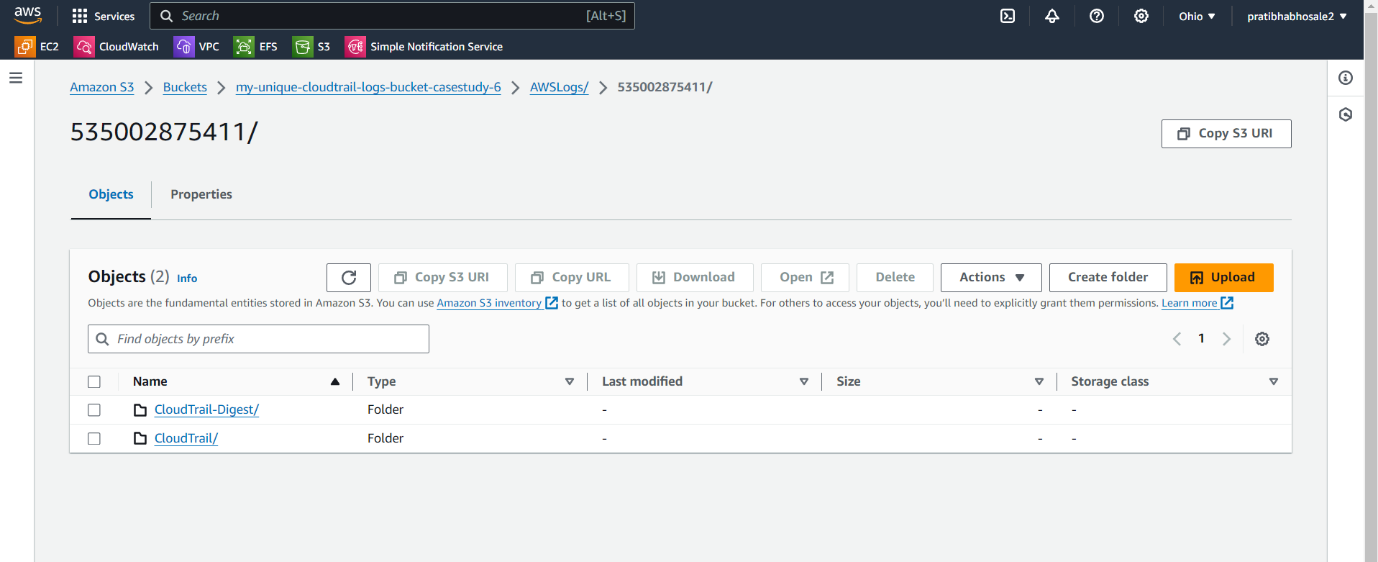
**Enforced Encryption**: applied a bucket policy to ensure that all objects stored in the bucket are encrypted.

**Added Lifecycle Rules**: To keep costs low, added a lifecycle policy that transitions older logs to **Infrequent Access** after 30 days and to **Glacier** after 90 days, with an expiration policy of 1 year.

This ensures that logs are securely stored, inaccessible publicly, and cost-efficient due to lifecycle management.

**1.Create S3 Bucket for CloudTrail Logs**





**2.Create S3 Bucket Policy for bucket**

****

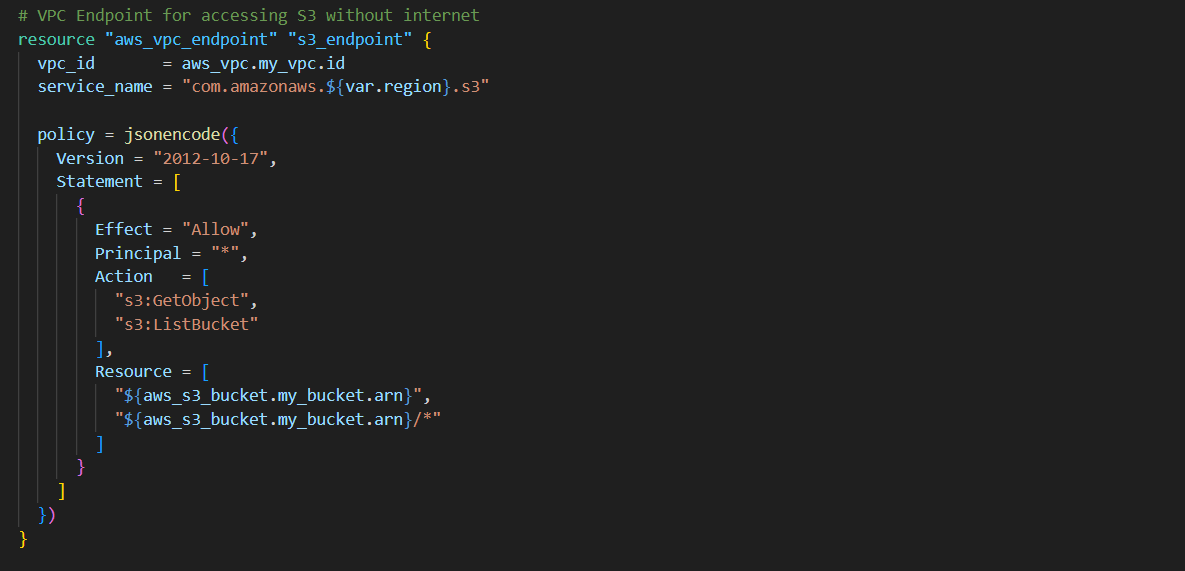
**3.VPC Endpoint for Secure Access to S3**:

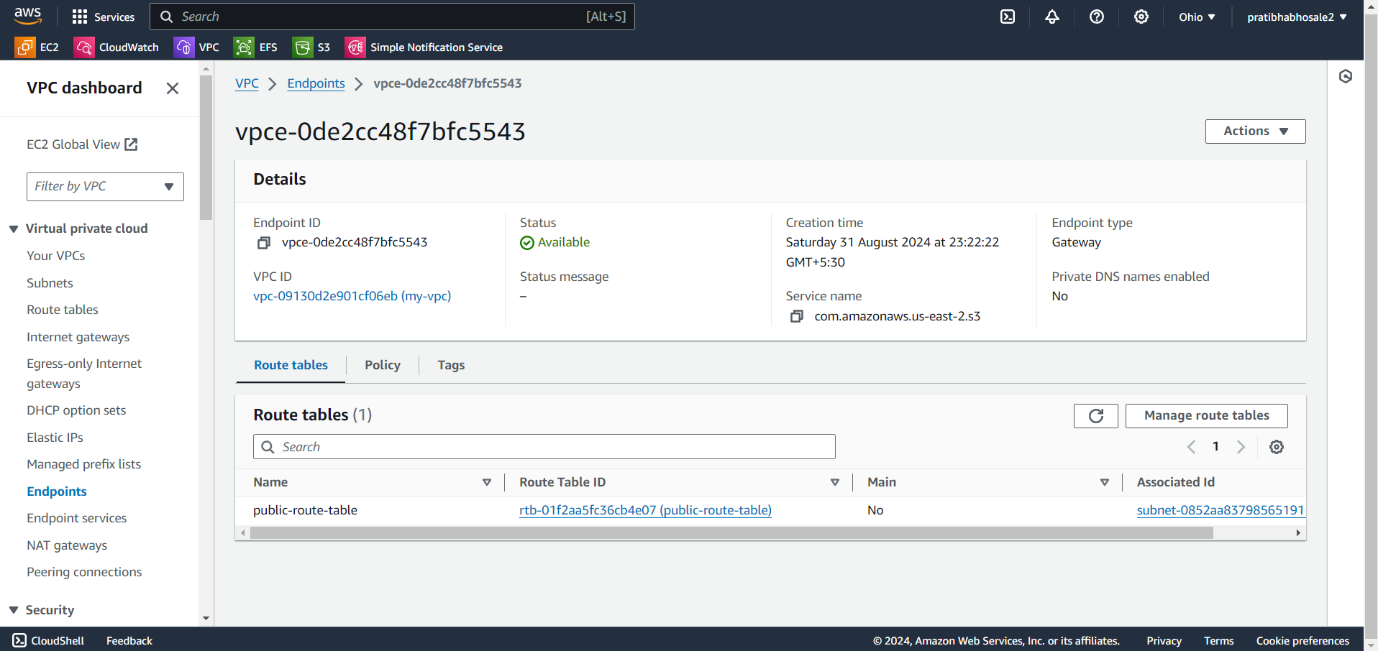
created a **VPC endpoint** (aws\_vpc\_endpoint) that allows EC2 instances to access the S3 bucket without going over the internet.

**VPC Setup**: configured a VPC and a subnet with an internet gateway for EC2 instances.

**Private Access to S3**: The VPC endpoint ensures that the EC2 instances can communicate with the S3 bucket directly within the private AWS network.

This avoids the need for public access or internet connectivity, ensuring **private and secure** communication between EC2 and S3.





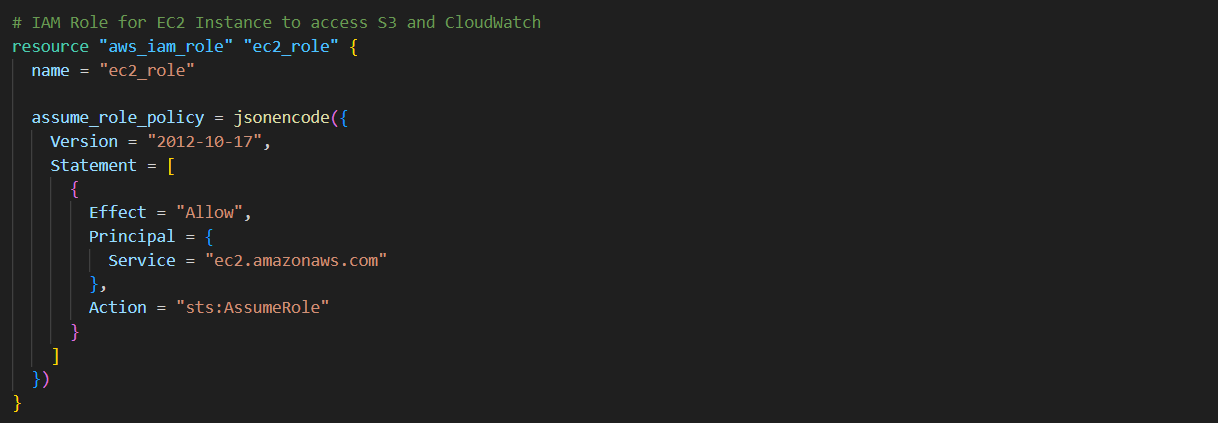
**Step-3 IAM Role for Secure Access:**

created an IAM role for the EC2 instance to access S3, CloudWatch, and SSM (for parameter store)

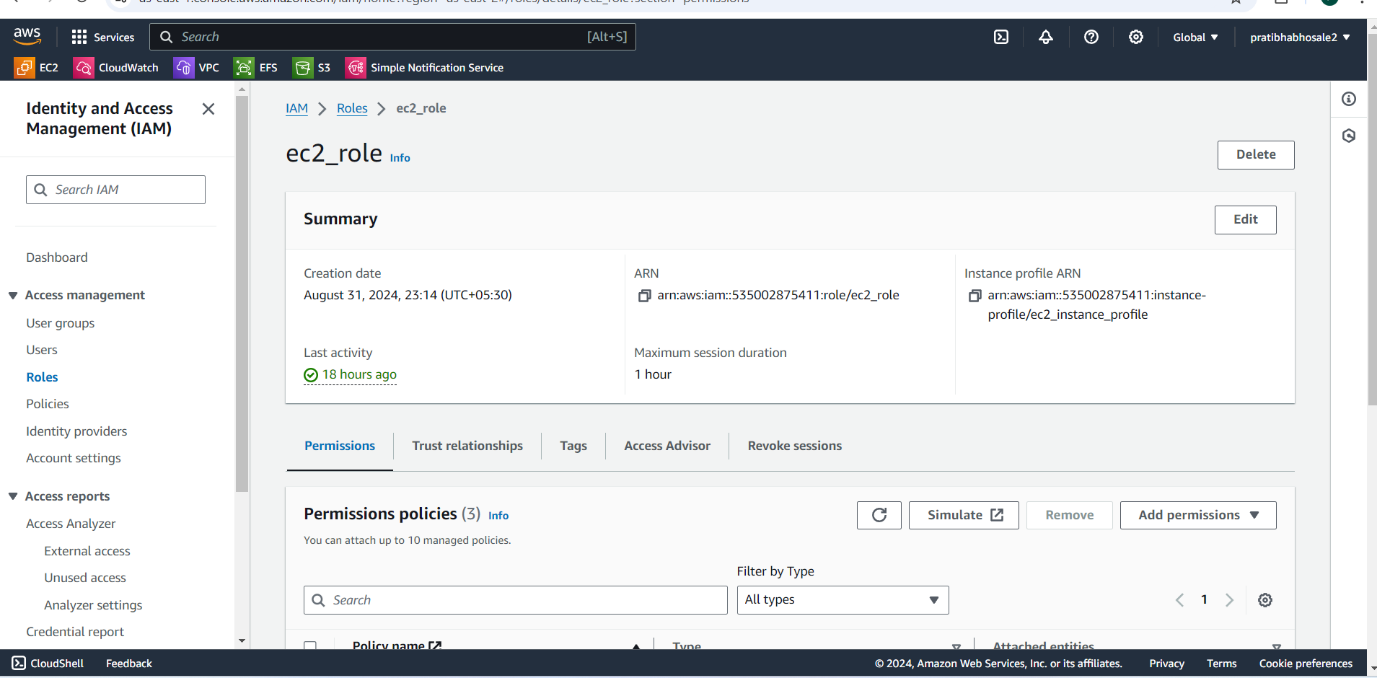
Assume Role Policy: This allows EC2 to securely assume the role.

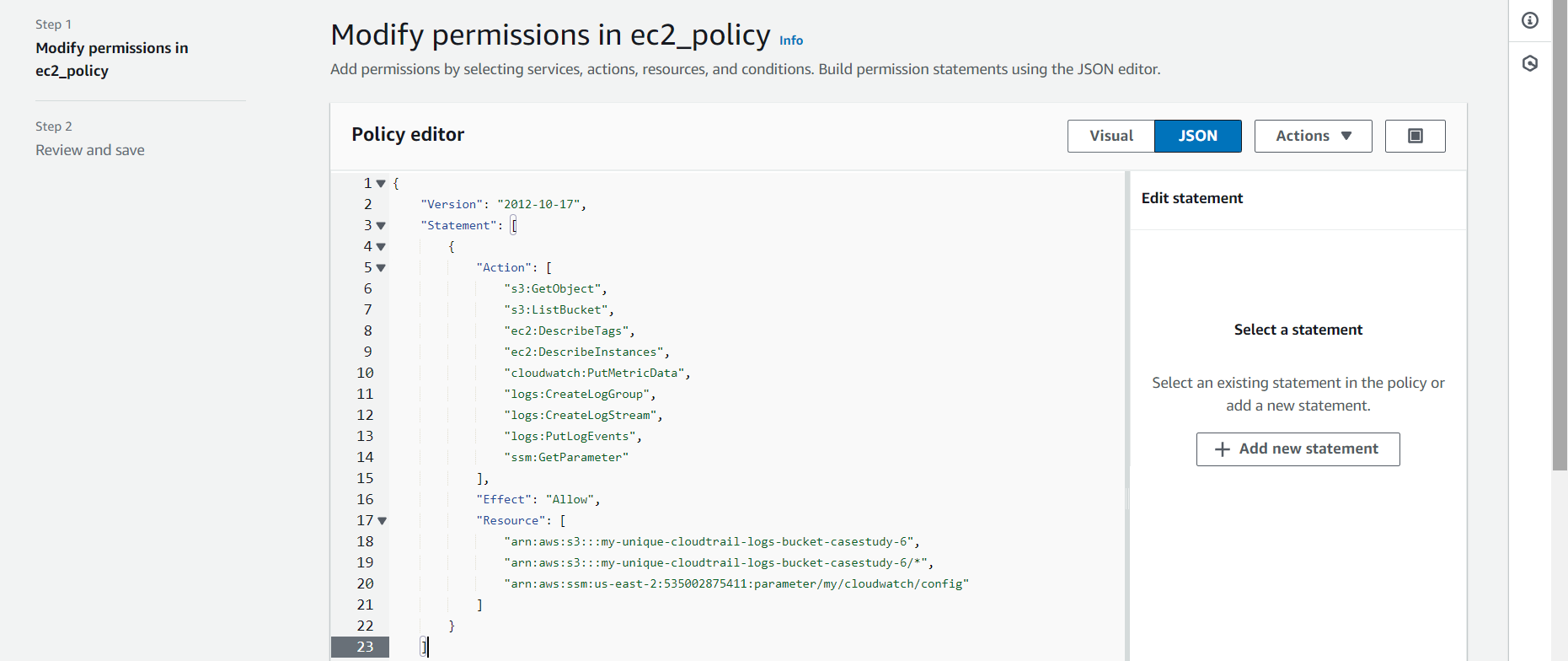
Policy for Access: I assigned permissions for S3 actions (s3:ListBucket, s3:GetObject),

CloudWatch actions (logs:CreateLogStream, logs:PutLogEvents), and SSM actions (ssm:GetParameter).

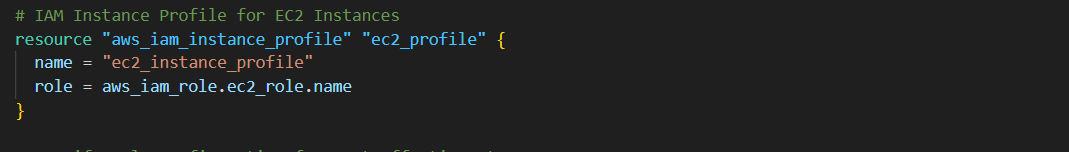


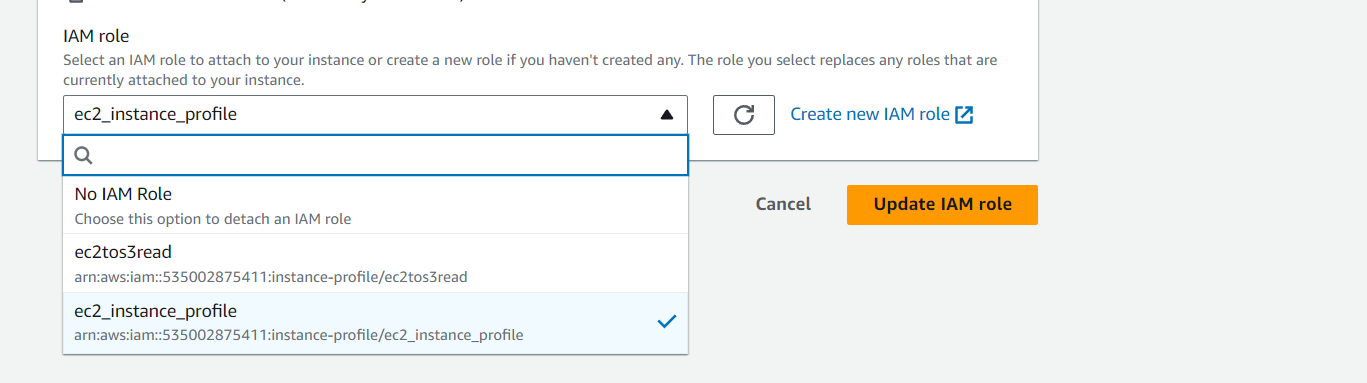




****

**Create IAM Instance Profile for EC2 Instances**

****

****

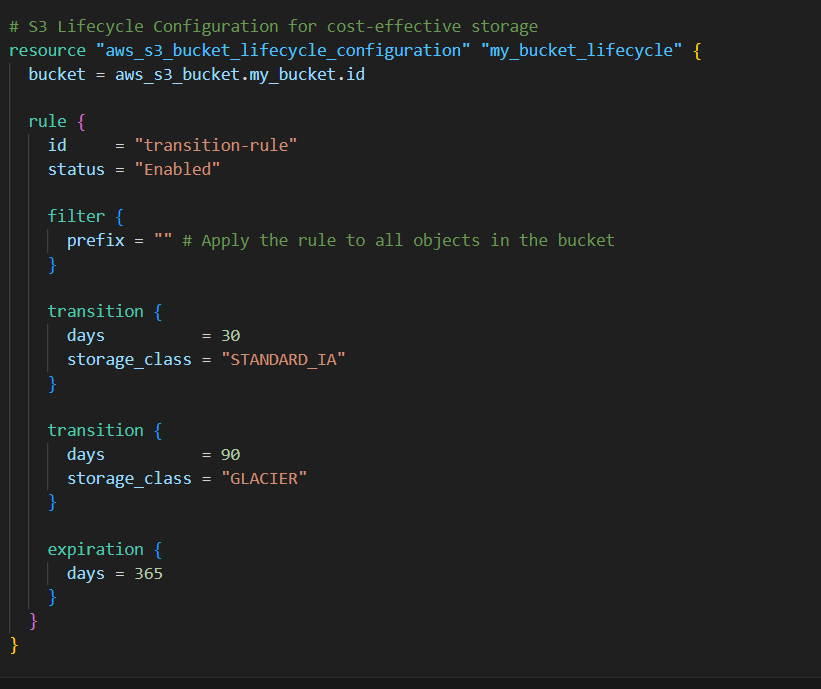
**Configure life cycle policy**

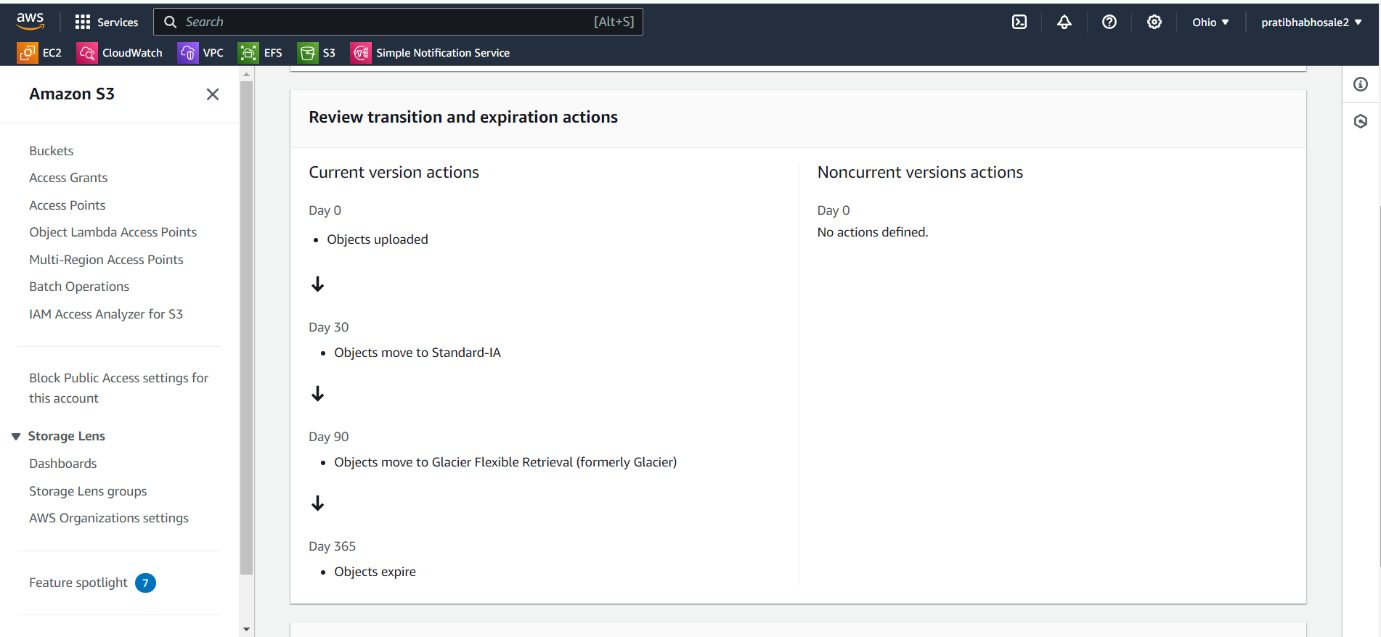
applied an **S3 Lifecycle configuration** to the CloudTrail logs bucket.

**Transition to Infrequent Access** after 30 days.

**Transition to Glacier** after 90 days.

**Expire objects** after 1 year.

****

****

**Provision EC2 Instances**

**Shell Script to Read Files from S3:**

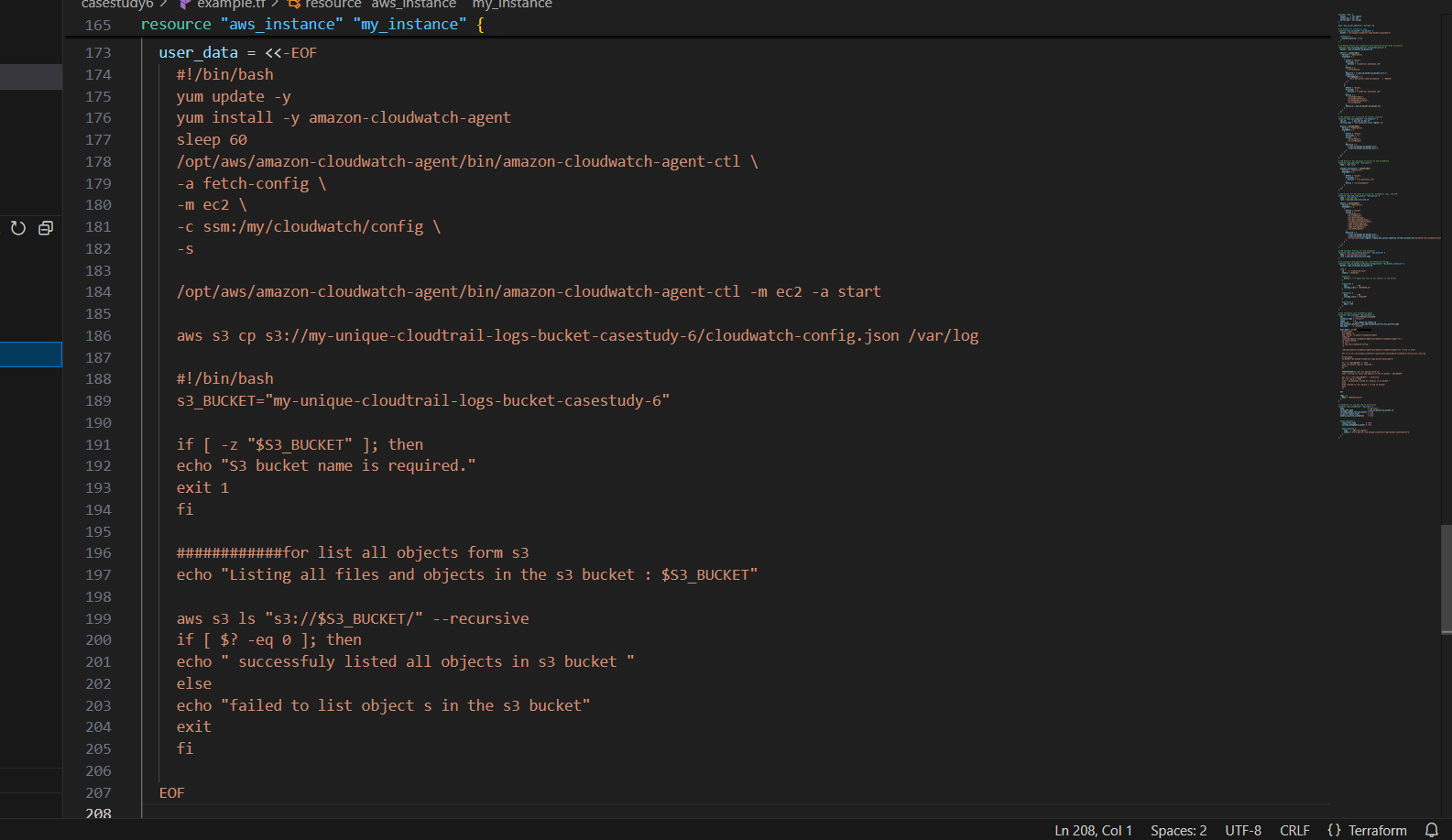
In the **user data** script attached to the EC2 instance:

Installed the **CloudWatch agent** to centralize logs from /var/log/messages.

Configured the instance to **list files** in the S3 bucket using aws s3 ls after verifying the bucket name.

Ensured that logs from CloudWatch were collected using an **SSM parameter** for configuration.

This allows the EC2 instance to access and list all objects in the S3 bucket securely without public or internet access.



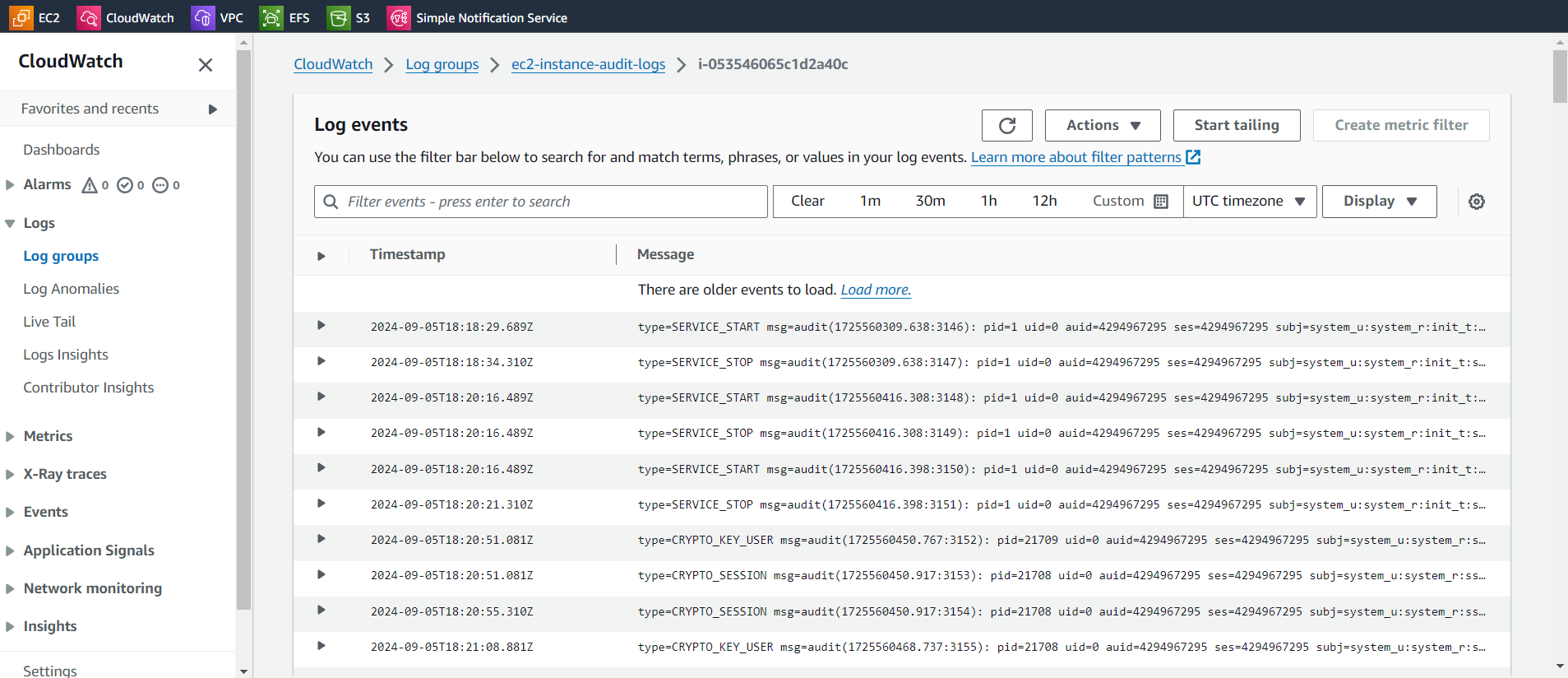
**5. Centralized Log Collection with CloudWatch:**

I installed and configured the **Amazon CloudWatch agent** to collect system logs (/var/log/messages) from the EC2 instance.

Used **SSM Parameter Store** to store the CloudWatch agent configuration.

Created an **SSM parameter** (/my/cloudwatch/config) to define the agent’s settings, allowing it to collect logs from all instances.

Centralizing logs is important for monitoring and auditing EC2 instances. This helps in **tracking system activities** and diagnosing issues.

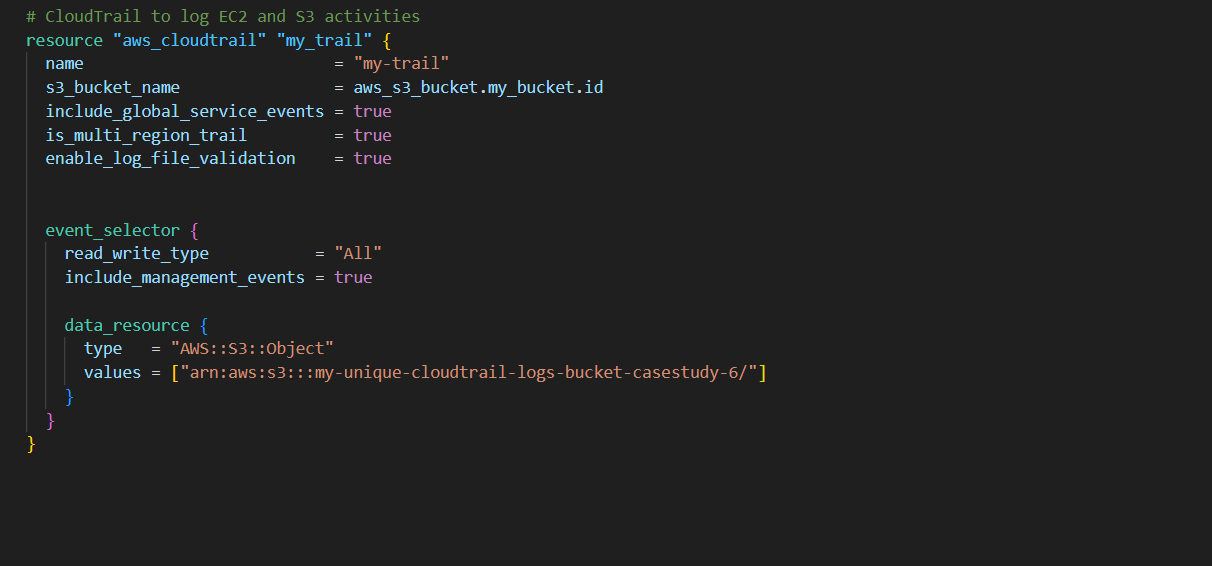


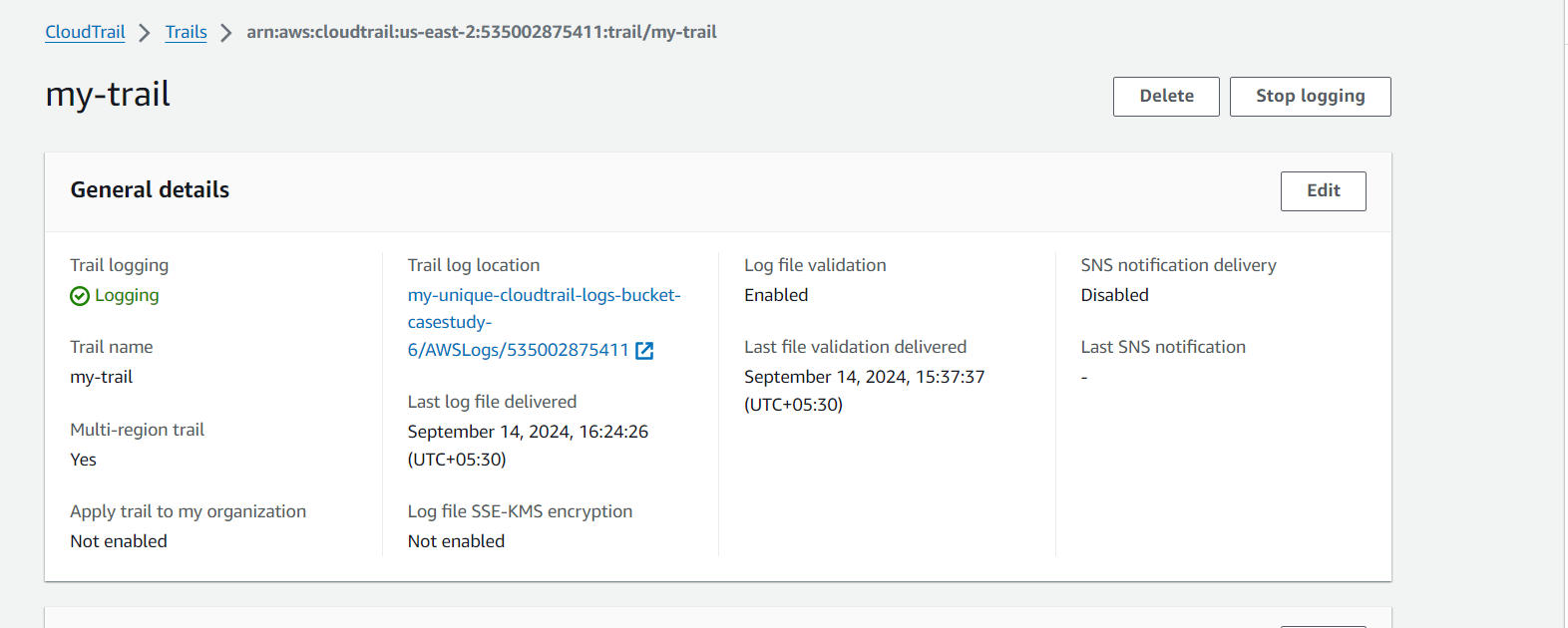
**6. CloudTrail for Logging EC2 and S3 Activity:**

I set up **AWS CloudTrail** to log **both management and data events** for the EC2 instance and the S3 bucket. All logs are stored in the **S3 bucket** I created earlier.

The **trail is multi-region**, and **log file validation** is enabled to ensure the integrity of logs.

CloudTrail helps track all API calls and actions performed on both the EC2 instance and the S3 bucket. This ensures I can audit any activity happening in the AWS environment, contributing to **security and transparency**.





GITHUB REPOSITORY FOR SCRIPT: <https://github.com/Pratibha251093/case-study-6.git>