CASE 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.*

***Duration:*** *3 days*

***Tech Stack:***

1. *Terraform/AWS CloudFormation*
2. *Shell Scripting*

**Key Components of the Architecture:**

**Infrastructure:**

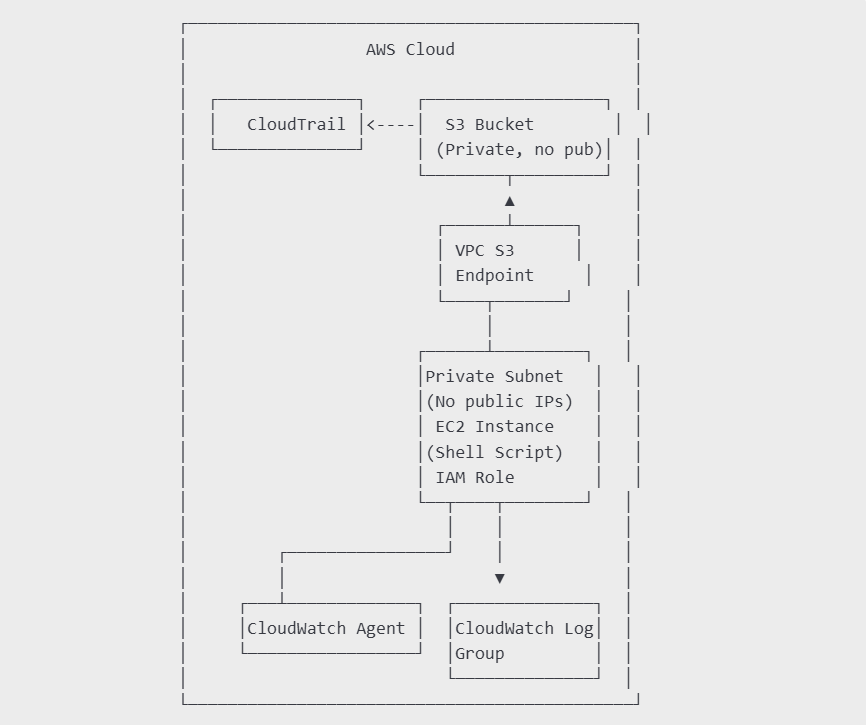
* **VPC with public and private subnets**
* **S3 Bucket** (Private)
* **Bastion host for accessing private EC2 instance**
* **VPC Endpoint for S3 (Gateway type)** → for private access
* **EC2 Instances (Linux, private subnet)**
* **IAM Role** attached to EC2 → with permissions to read from S3
* **CloudWatch Agent** → for collecting logs from /var/log/messages
* **CloudTrail** → to log activity on EC2 and S3

**==================================================================================**

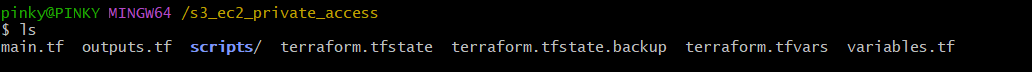
Private subnet EC2 = **secure**

S3 VPC Endpoint = **no public access**

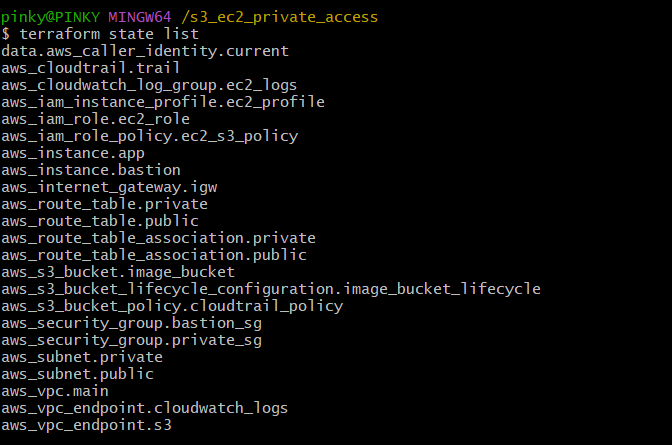
IAM Role = **fine-grained access**



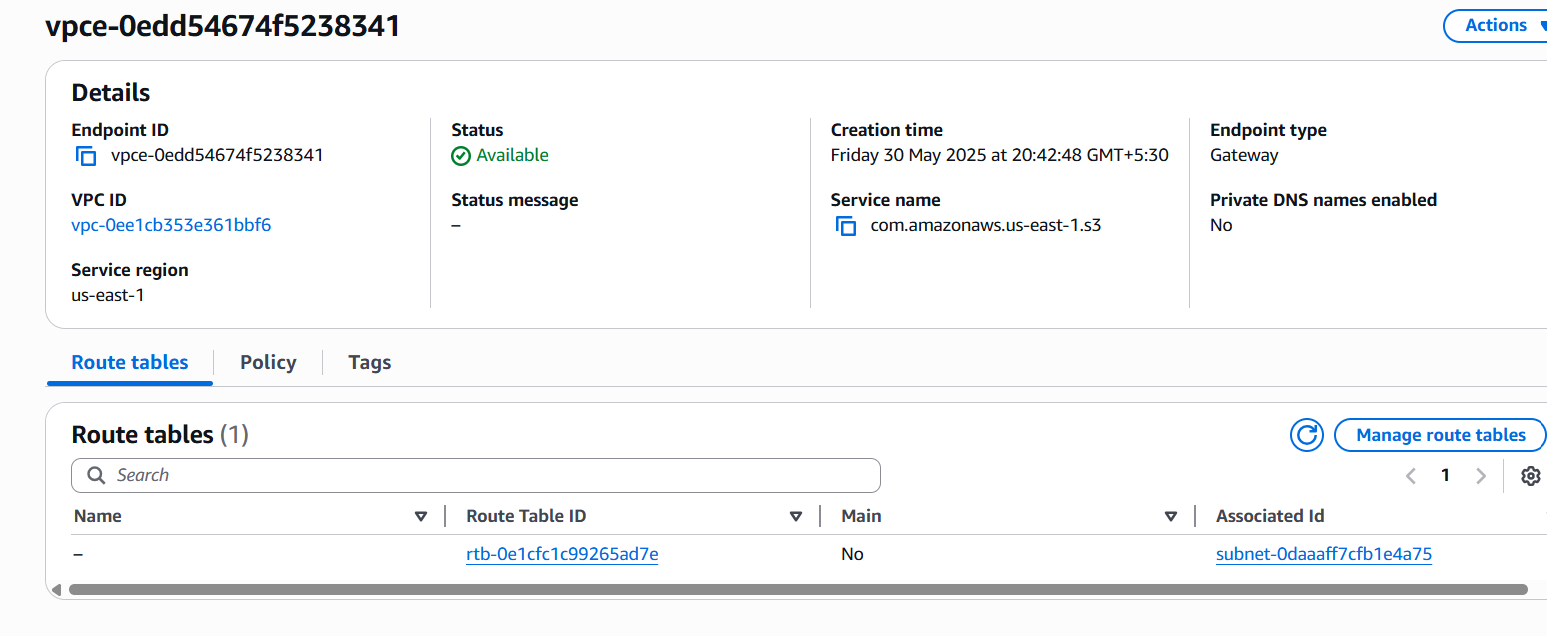
**Implementation: I have used Terraform for creating infrastructure for this case study**



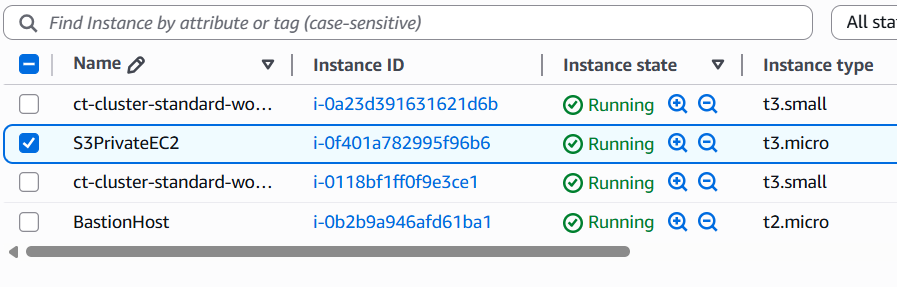
Below resources I have created



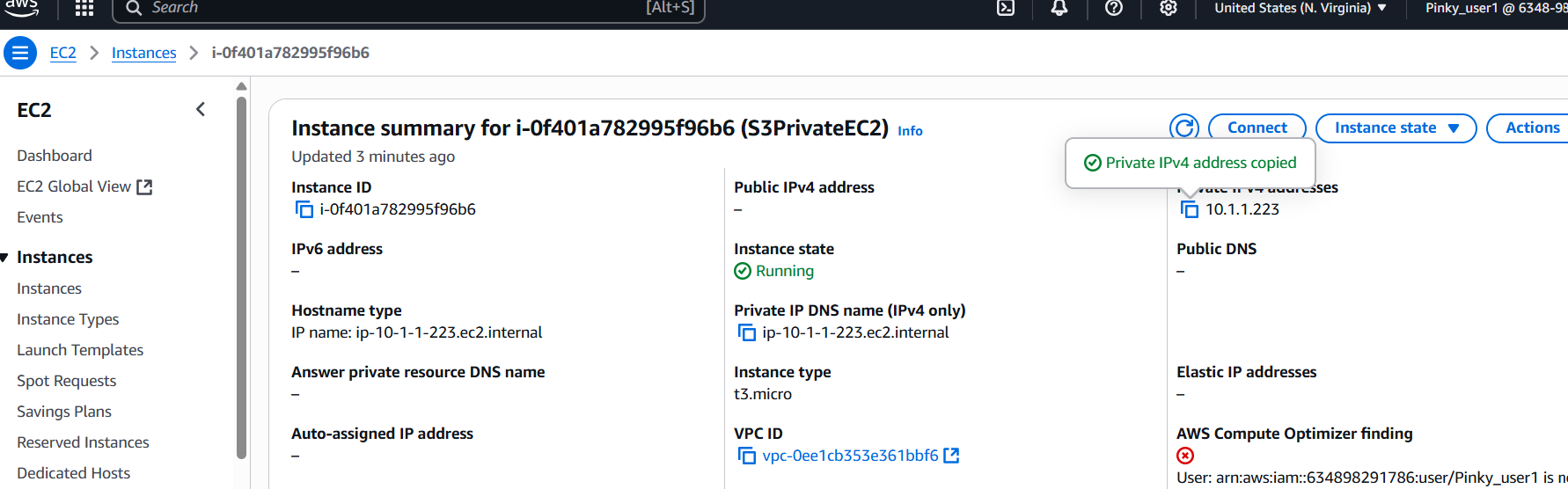
VPC endpoint for S3—Gateway endpoint



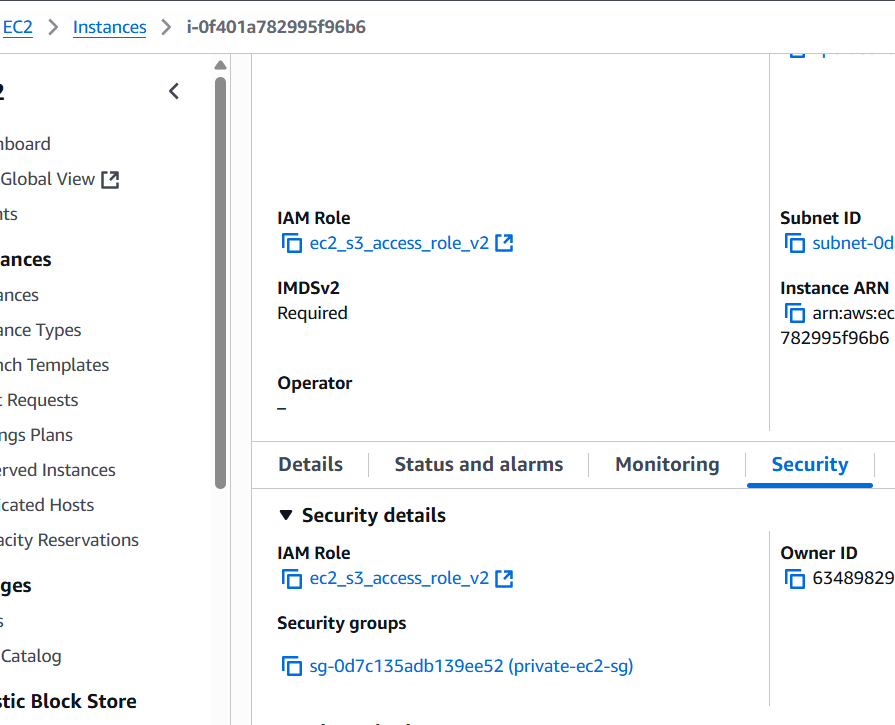
The private EC2 instance and Bastion Host are shown below.

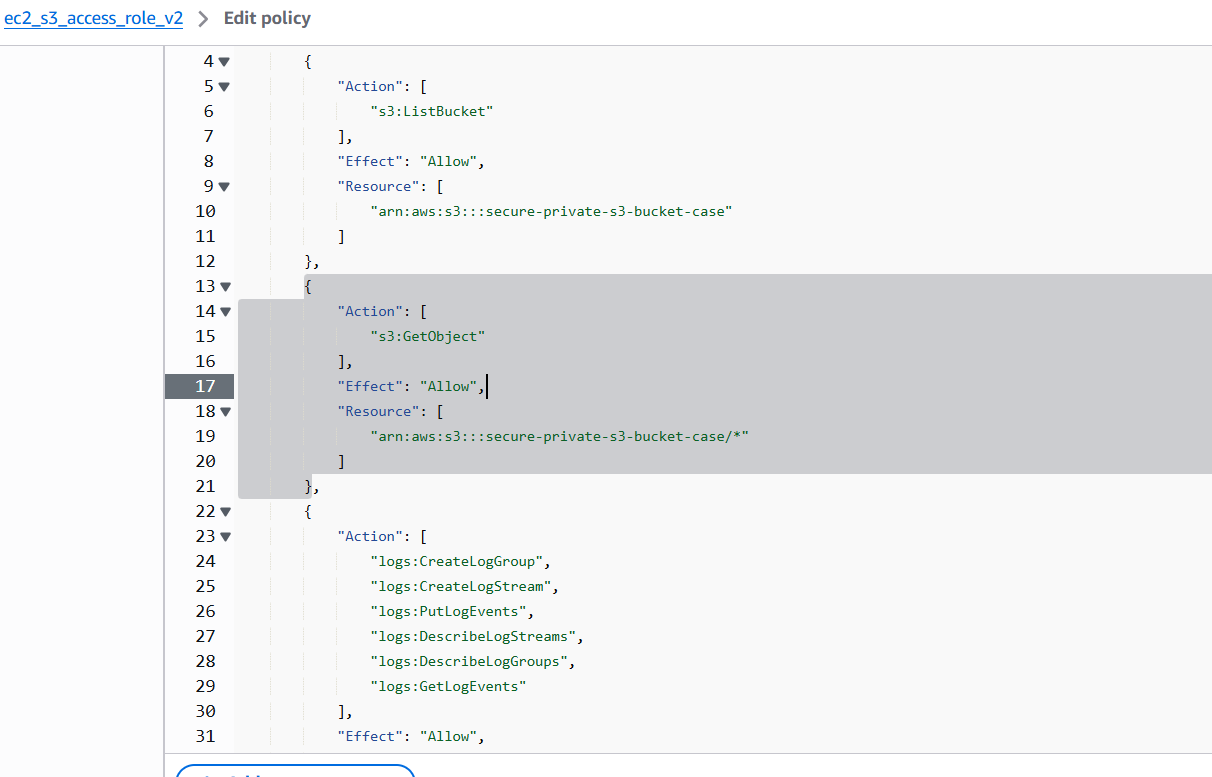


The private ec2 instance which has only the private Ip address is shown below:



This is the ec2 instance role and associated security group





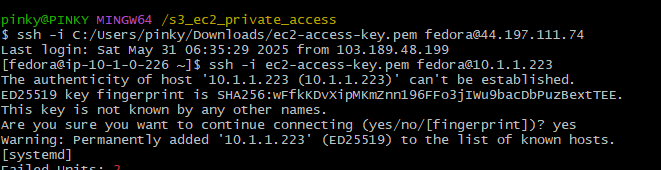
Private IP Address(S3PrivateEC2)🡪 is 10.1.1.223

**From local machine: ssh to Bastion host**

**Then SSH into the EC2 instance from Bastion host**

ssh -i C:/Users/pinky/Downloads/ec2-access-key.pem [fedora@44.197.111.74](mailto:fedora@44.197.111.74)

ssh -i ec2-access-key.pem [fedora@10.1.1.223](mailto:fedora@10.1.1.223)



**From the ec2 instance; curl google did not work, which proves it has no internet connection.**



**Verify S3 Access Through VPC Endpoint**

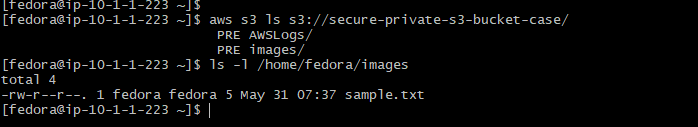
**From the EC2 instance, run:**

**Aws s3 ls bucket name**

**Able to list out the files inside the s3 bucket :**

**Ec2 instance is able to access s3 through VPC endpoint**

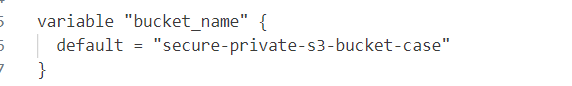
From ec2 instance we are able to see the images folder

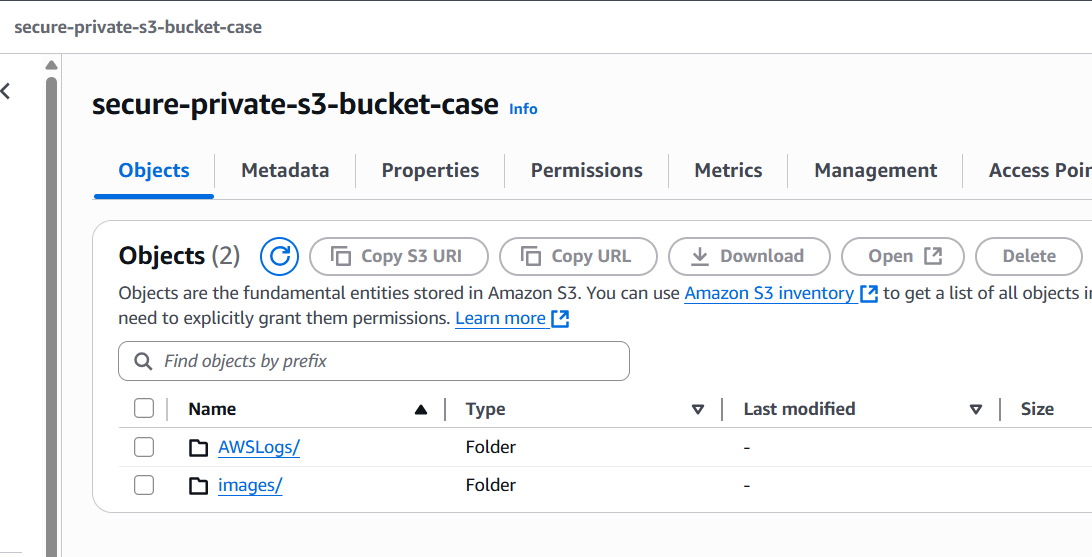


Using the script the images copied to the home folder of the ec2 so I am able to read the files from s3 and able to view the files in the s3

sample.txt file in the images folder in the s3 bucket is copied to the home/fedora/images in the ec2 instance .

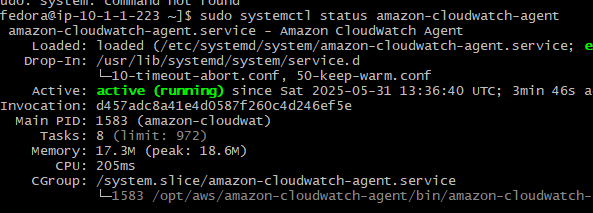


****

****

**----------------------------------------------------------------------------------------------------------------**

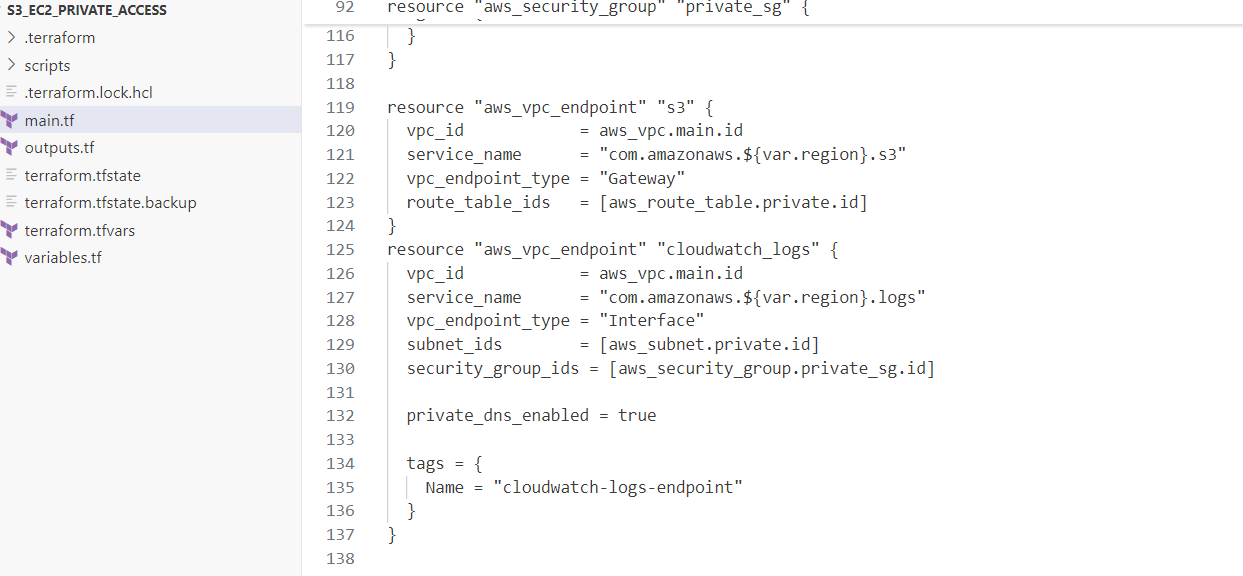
**Next we need to check whether the cloudwatch agent is installed on the ec2 instance**

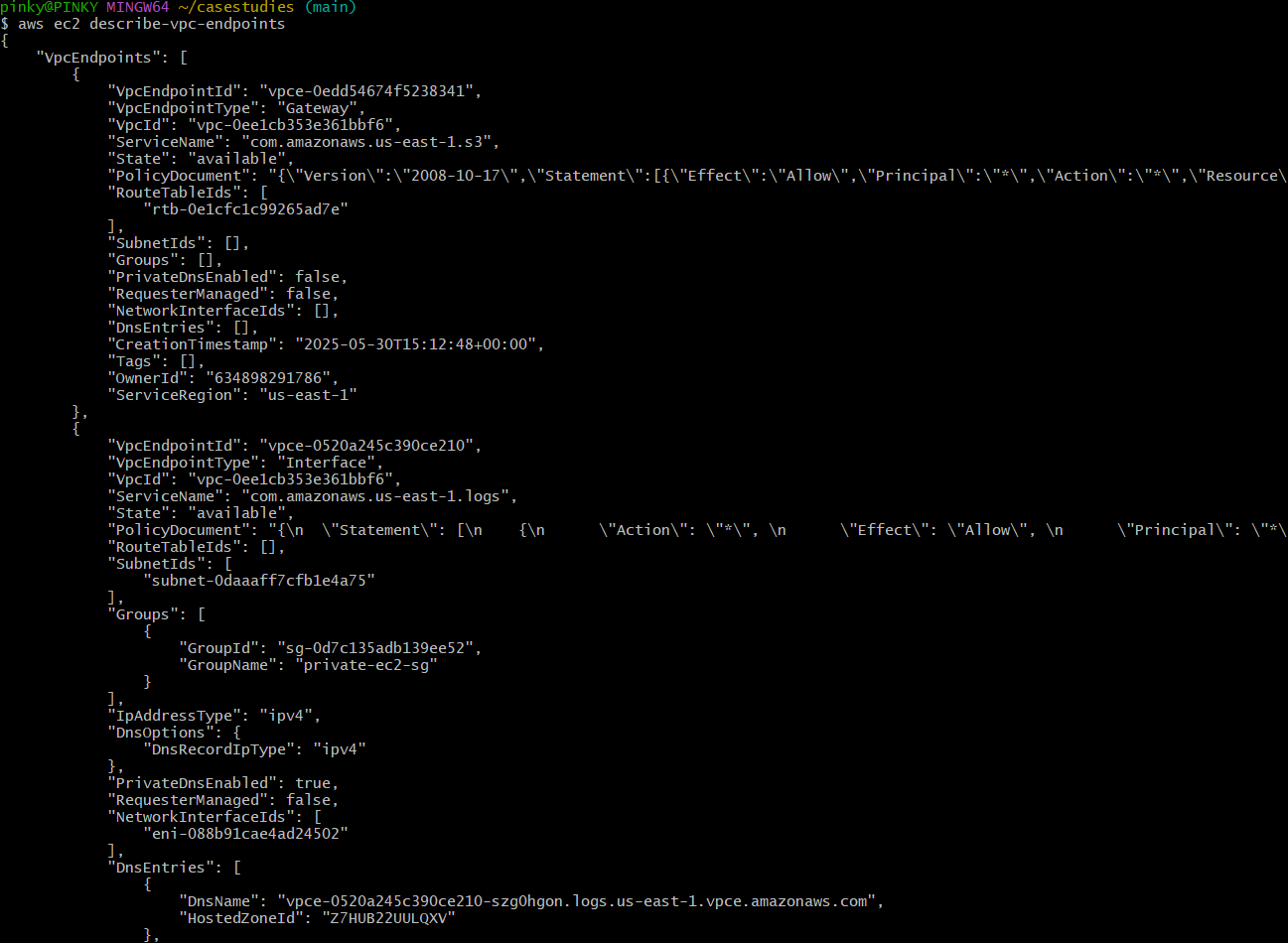




To allow your private EC2 to send logs to CloudWatch without internet access, you **must create a VPC interface endpoint** for the service logs.us-east-1.amazonaws.com.

Here you can see the vpc interface endpoint and gateway vpc endpoint





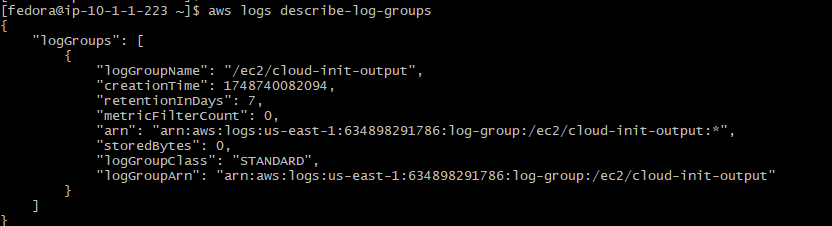
In **Fedora (and other cloud-init enabled AMIs)**, system boot logs and provisioning output are typically written to:

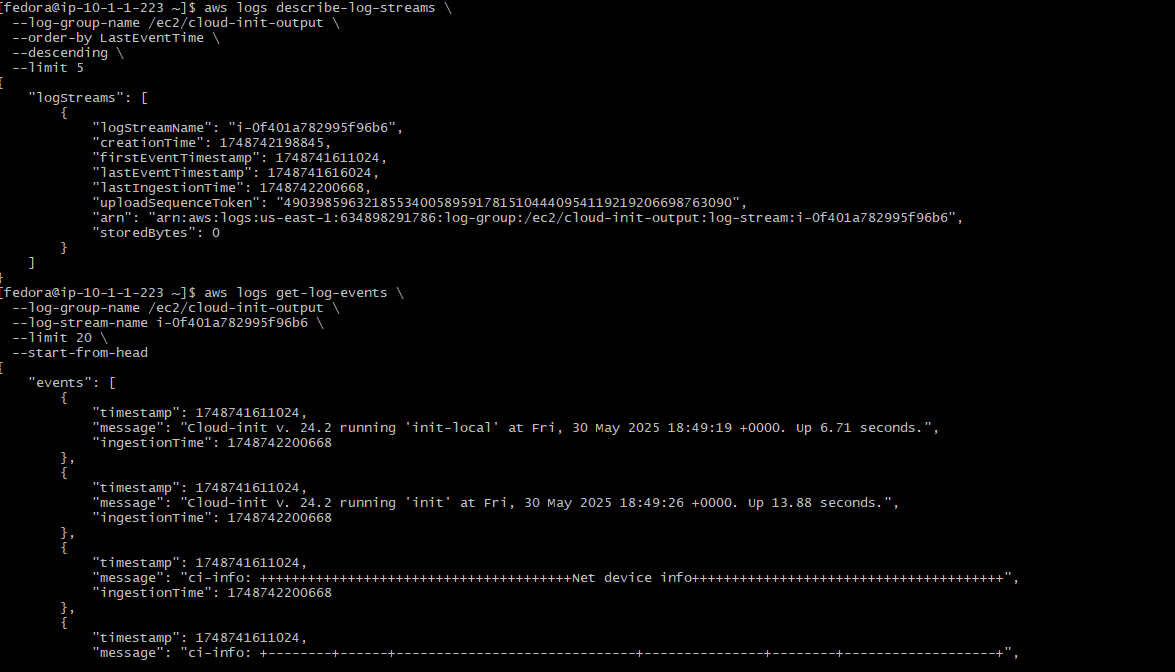
/var/log/cloud-init-output.log

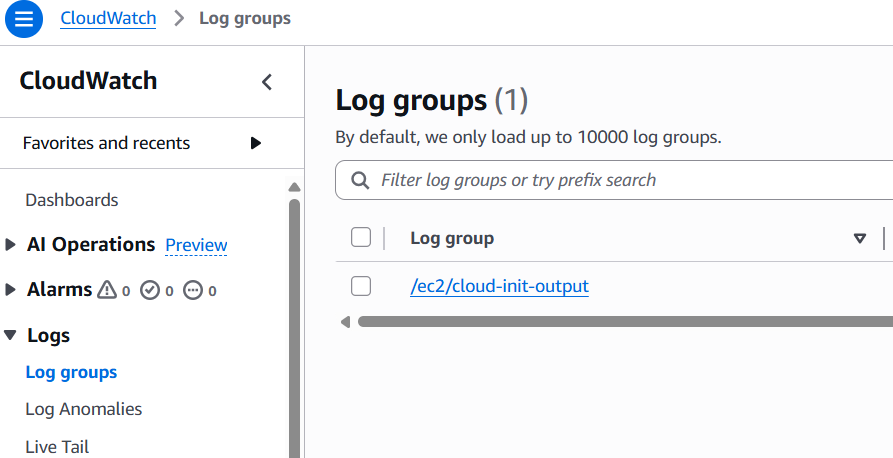
Instead of the more traditional:

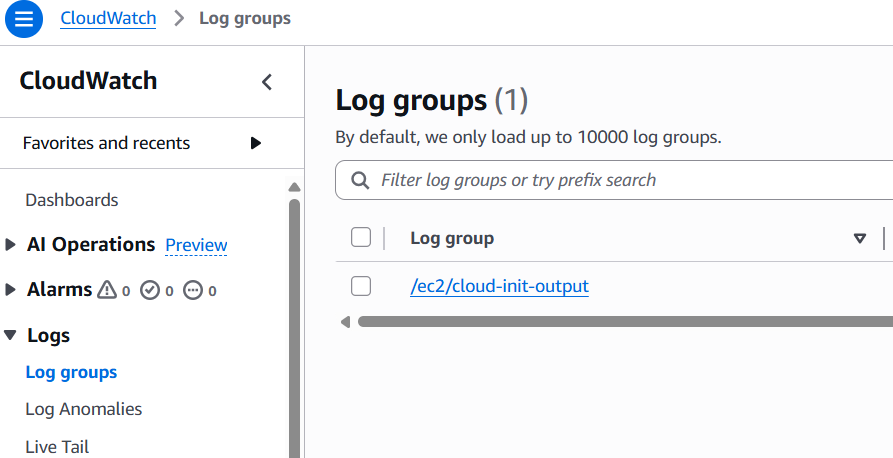
/var/log/messages

If you're configuring the **CloudWatch Agent** to send logs, you must ensure it's explicitly told to monitor:/var/log/cloud-init-output.log.

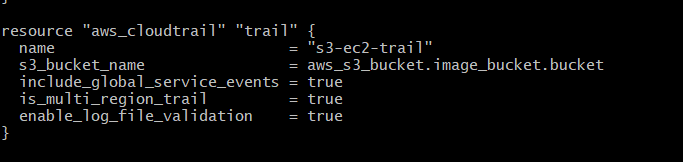


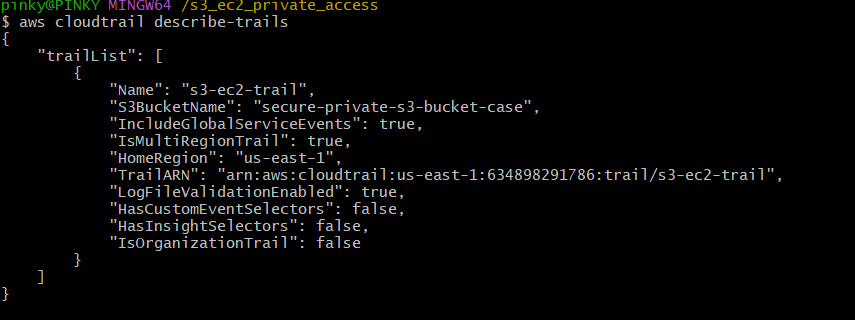




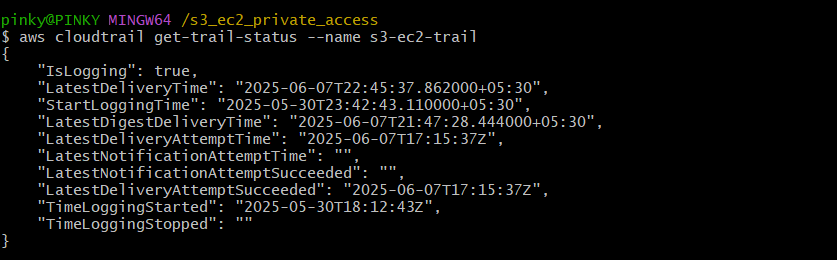


Next Check cloud trail creation:

The resource I created using terraform 



**S3-ec2-trail has been generated.**



This confirms that EC2 and S3 activity is being captured by CloudTrail.

Check the below screenshot:





Next **Check S3 Bucket for CloudTrail Logs**

If your CloudTrail is configured to log to an S3 bucket:

