Project 2: Securing and Monitoring Resources with AWS

Name: Yousef Hassan Yousef Eltobgy

Student ID: 21031713

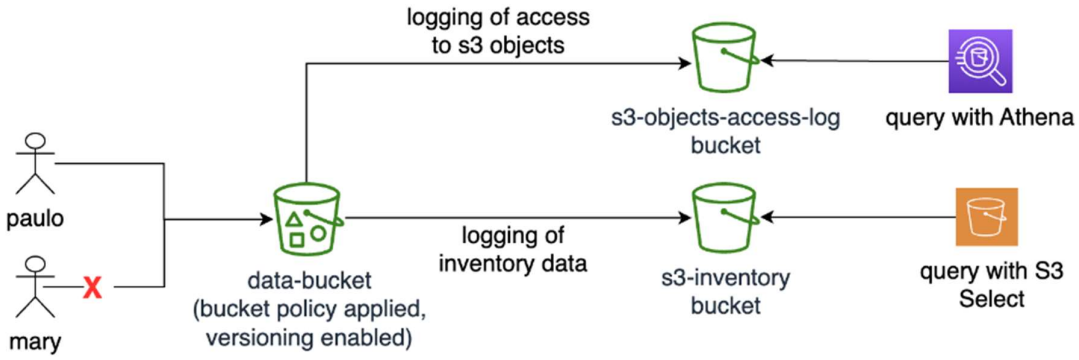
Email: eltobgyy@gmail.com

Track: AWS Cloud Solution Admin & Architect

Group Code: ALX1_ISS4_M1e

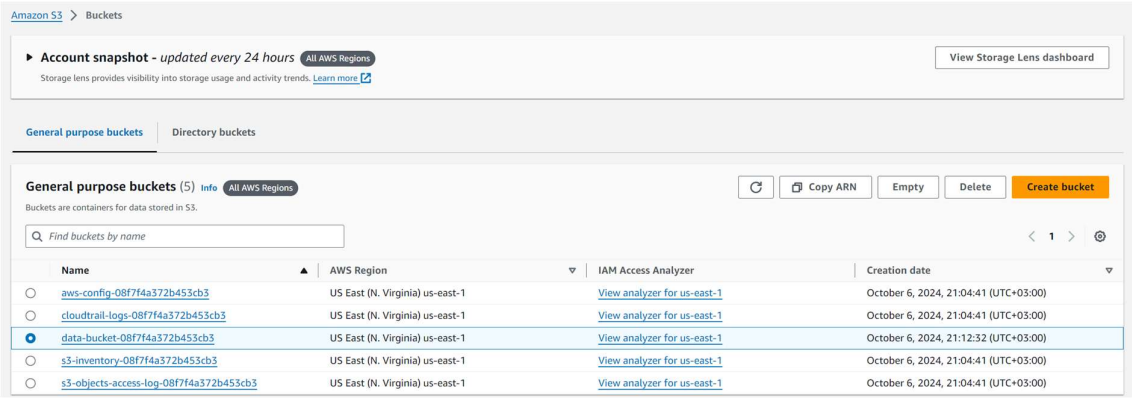
Project 2: Securing and Monitoring Resources with AWS

Phase 1: Securing data in Amazon S3

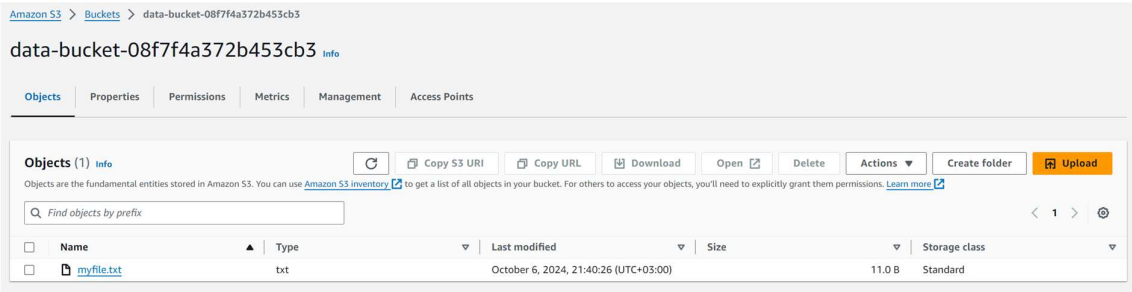


Task 1.1: Create a bucket, apply a bucket policy, and test access

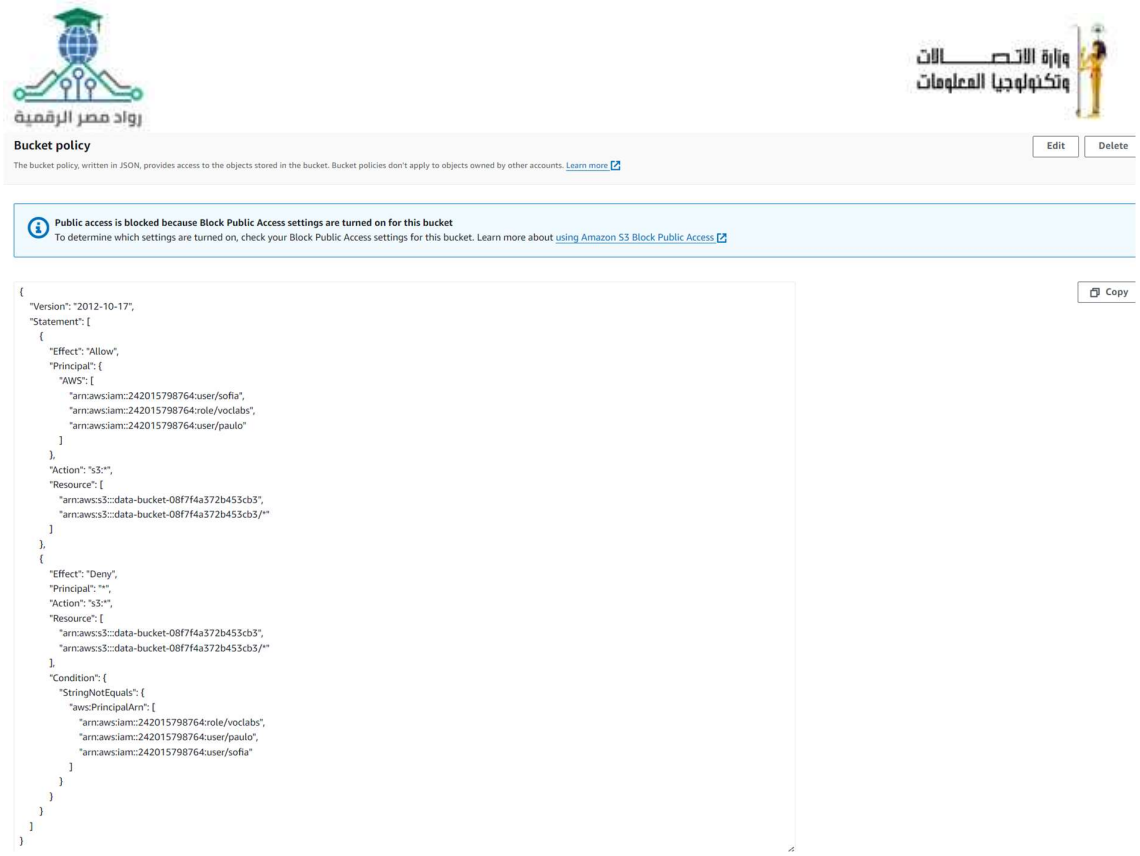
Created the bucket below



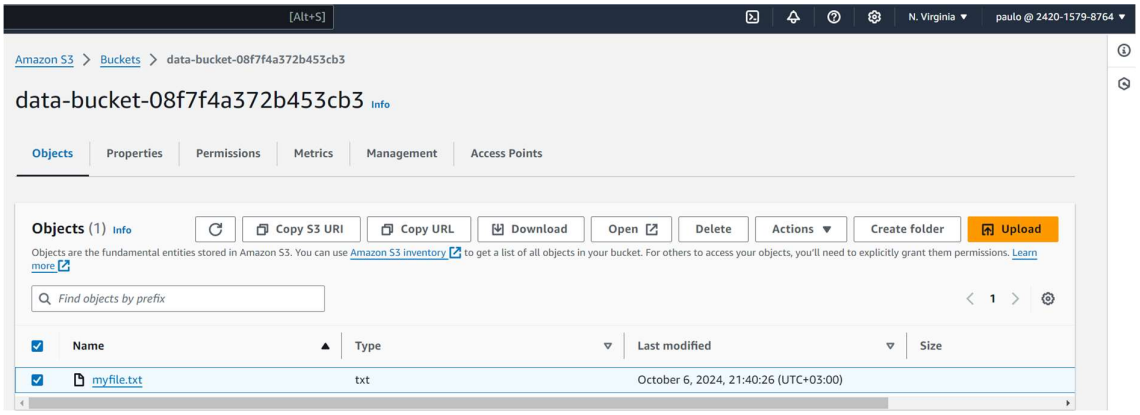
Uploaded a file



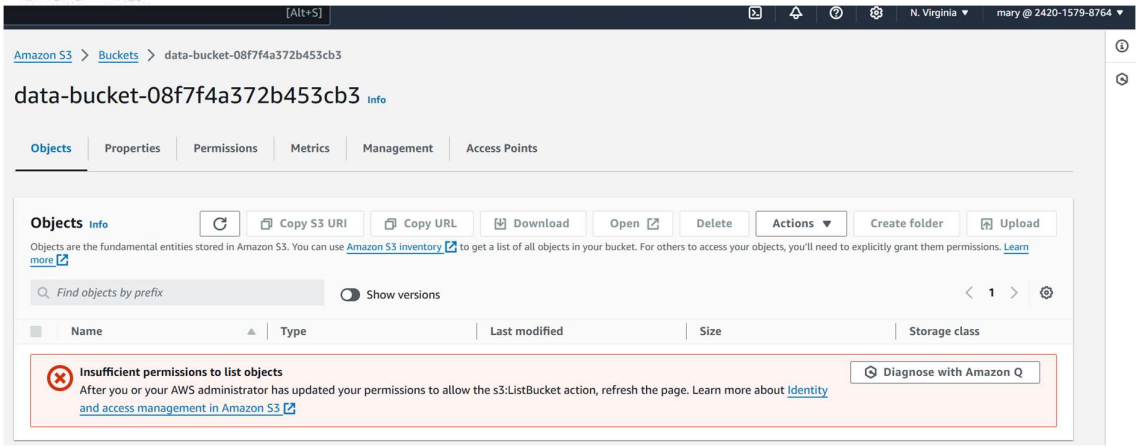
I added the below policy to allow all Amazon S3 service actions for all principals for data-bucket and all objects in it, only for the ARN equals the ARN for the voclabs IAM role, the paulo IAM user, or the sofia IAM user.



As per below I can verify that the user named Paulo has permission to access the S3 bucket and can download the file and has other actions that he can do.

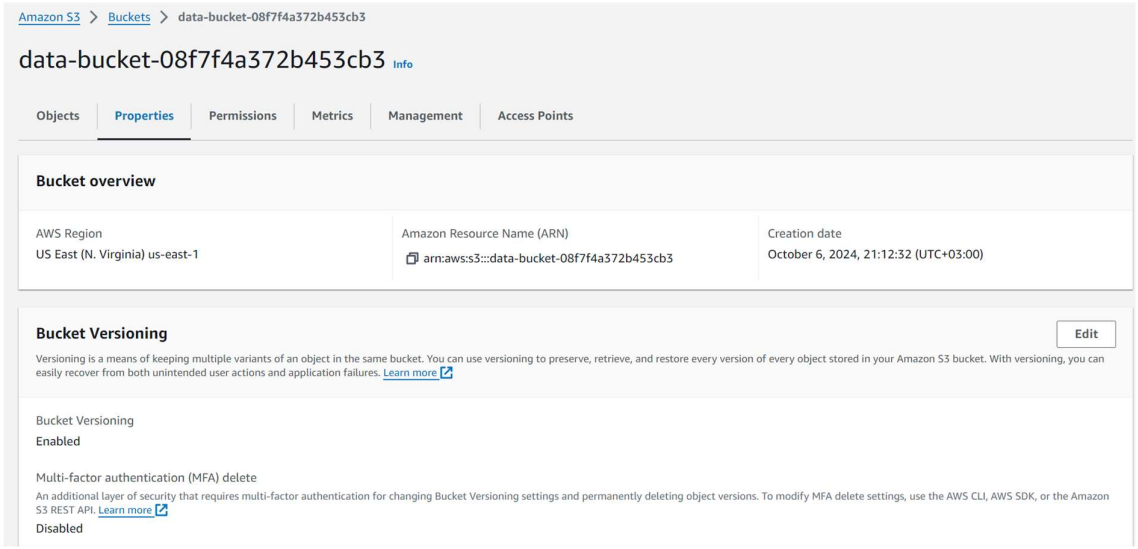


But the user named Mary whom I didn't put her ARN in the policy to allow her to make actions in the bucket, can't view its content.

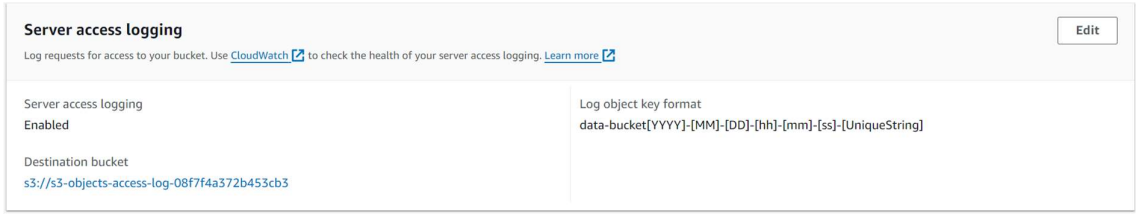


Task 1.2: Enable versioning and object-level logging on a bucket

Enabled the versioning on the bucket.



Enabled the Server access logging on the bucket.

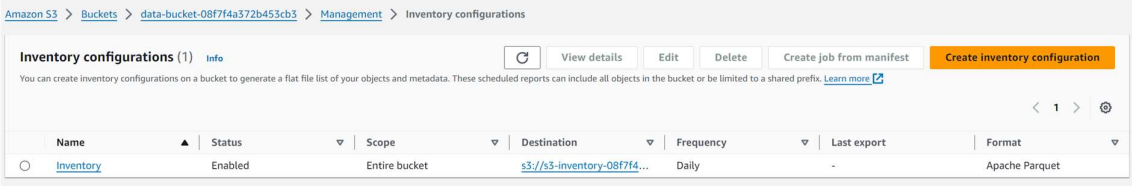


I verified that the policy in the bucket allows the whole account to add files in the bucket so there would be no problems when the data bucket put the logs in it. I added the ARN of the data bucket to be allowed to put logs in it.

```
{
  "Version": "2012-10-17",
  "Id": "S3-Console-Auto-Gen-Policy-1728241602971",
  "Statement": [
    {
      "Sid": "S3PolicyStmt-DO-NOT-MODIFY-1728241602736",
      "Effect": "Allow",
      "Principal": {
        "Service": "logging.s3.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::s3-objects-access-log-08f7f4a372b453cb3/*",
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": "arn:aws:s3:::data-bucket-08f7f4a372b453cb3",
          "aws:SourceAccount": "242015798764"
        }
      }
    }
  ]
```


Task 1.3: Implement the S3 Inventory feature on a bucket

Created the inventory for our data-bucket, and the destination is the “s3-inventory” bucket.

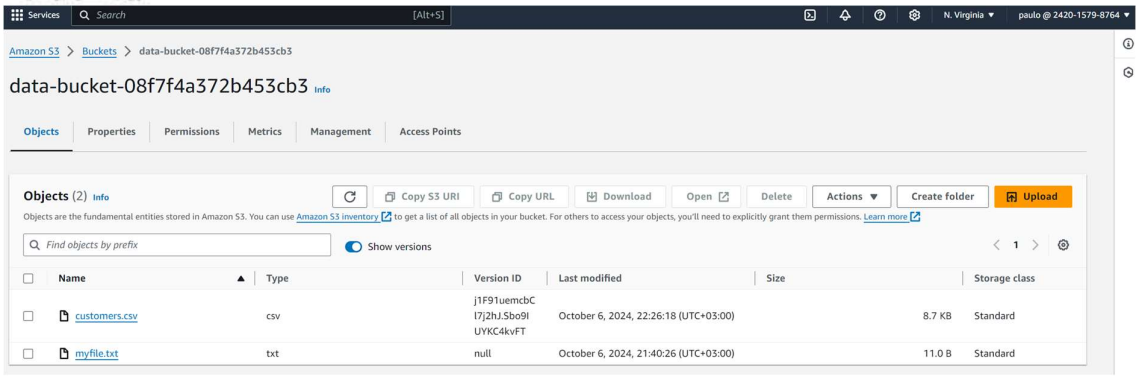


Task 1.4: Confirm that versioning works as intended

Created a file named “customers.csv” with the below data.

A	B	C	D	E	F	G
CustomerID	First Name	Last Name	Join Date	Street Address	City,State	Phone
1	Alejandro	Rosalez	12/12/2013	123 Main St.	Any Town,MD	301-555-0158
2	Jane	Doe	10/5/2014	456 State St.	Anywhere,WA	360-555-0163

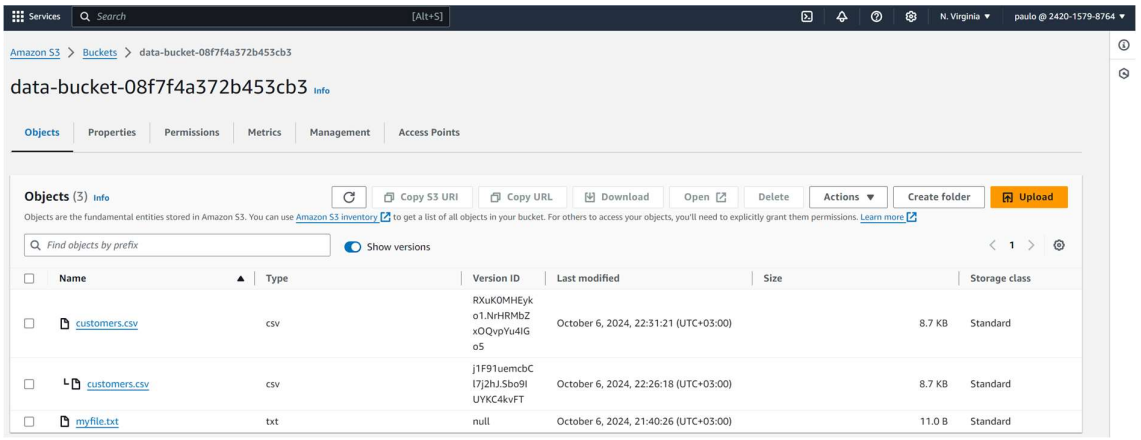
I uploaded the file with the user named Paulo



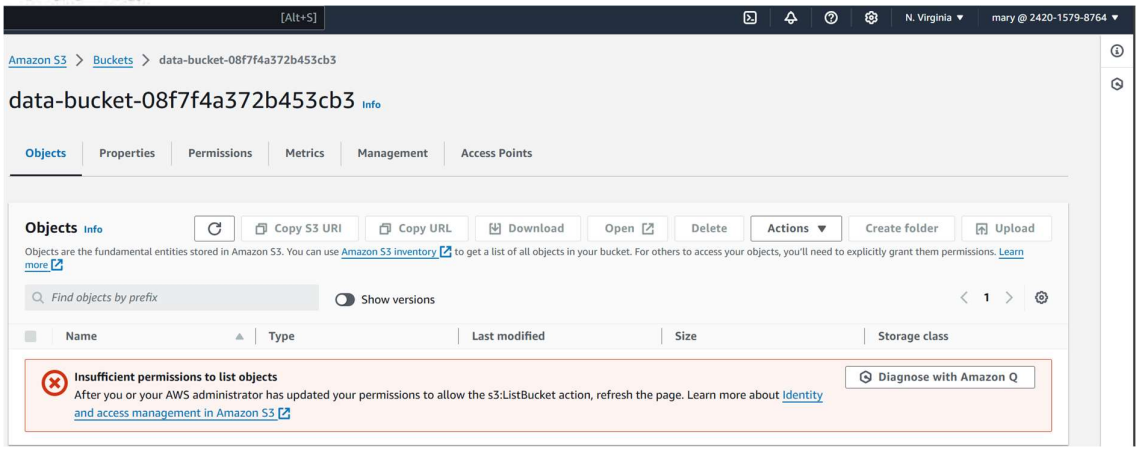
Then I made the below change in the file.

A	B	C	D	E	F	G
CustomerID	First Name	Last Name	Join Date	Street Address	City,State	Phone
1	Alejandro	Rosalez	12/12/2013	123 Main St.	Any Town,MD	301-555-0158
2	Jane	Doe	10/5/2014	456 State St.	Anywhere,WA	360-555-0163
3	x	x	x			

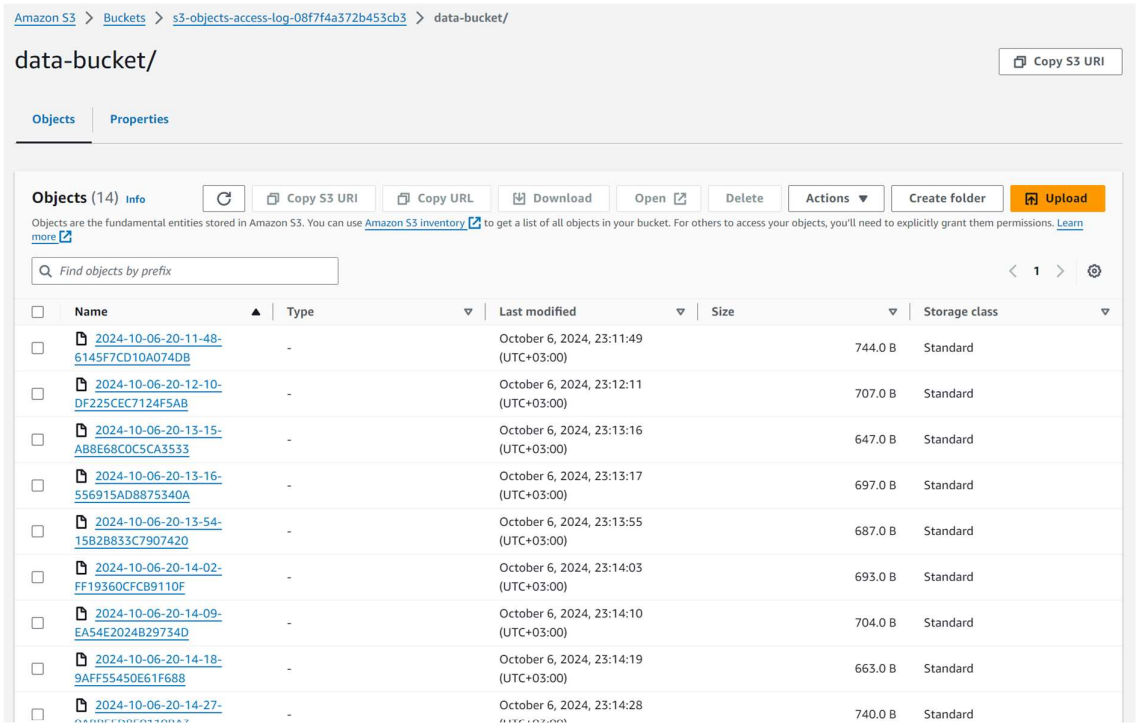
Then uploaded it again. As we can see the versioning is working, the same file has two versions.



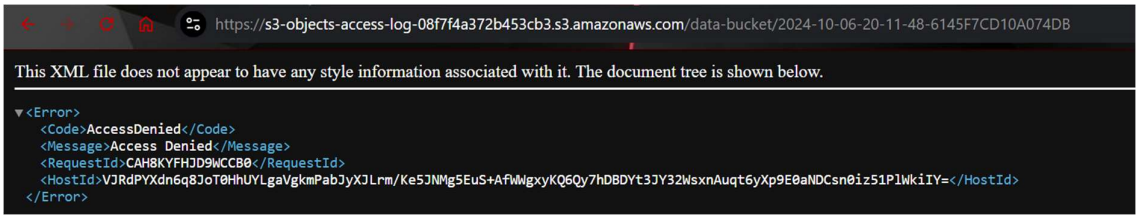
Tried to log in with the user named Mary to make logs of unauthorized access.



The logs folder was created successfully automatically and the logs were put inside it.



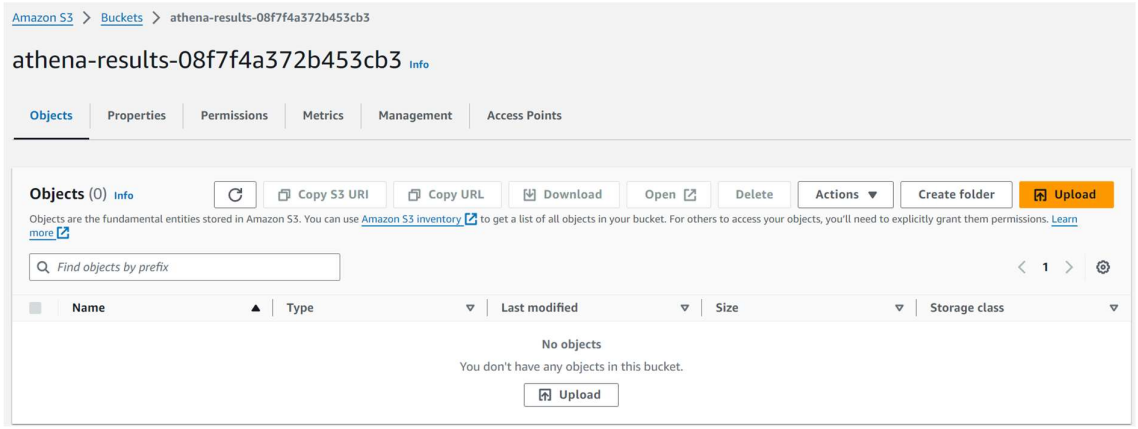
And the deny log appeared as per below.



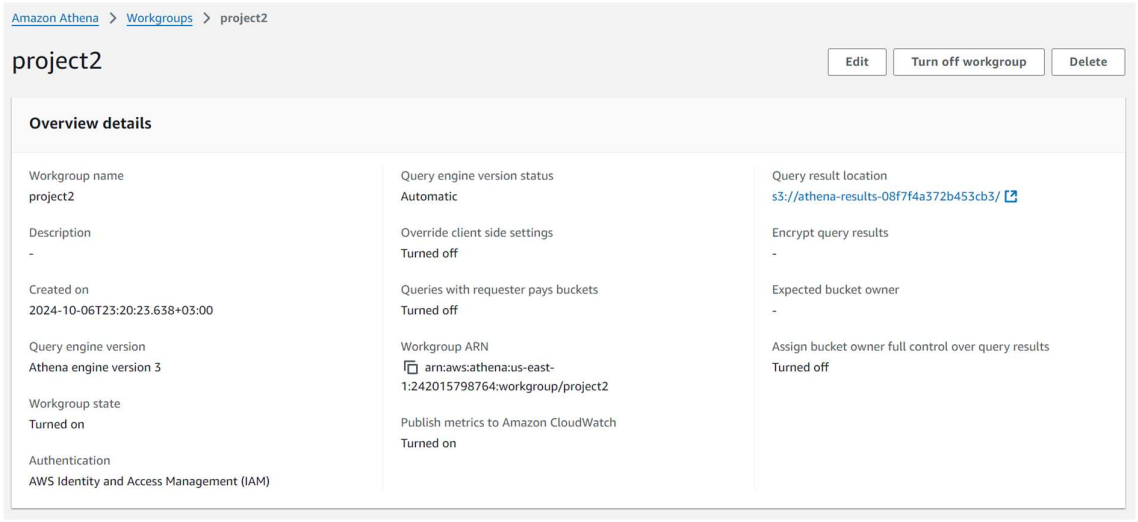
Task 1.5: Confirm object-level logging and query the access logs by using Athena

I downloaded one of the log files and this is what is inside it.

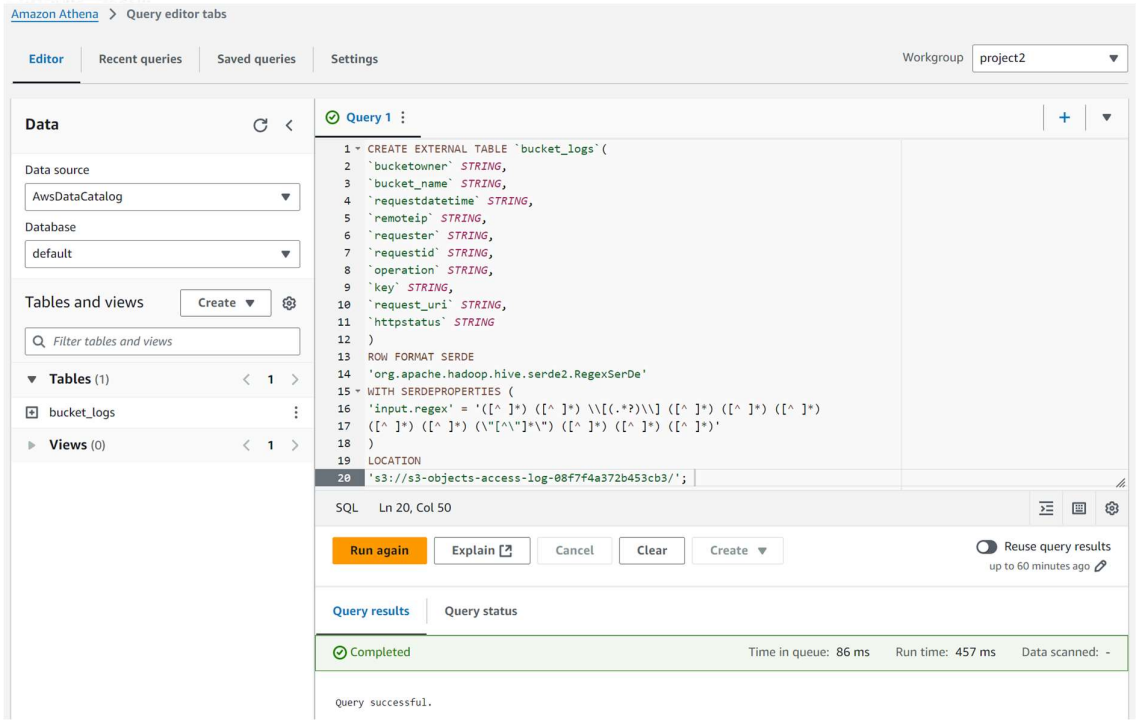
Then I created a bucket for athena results.



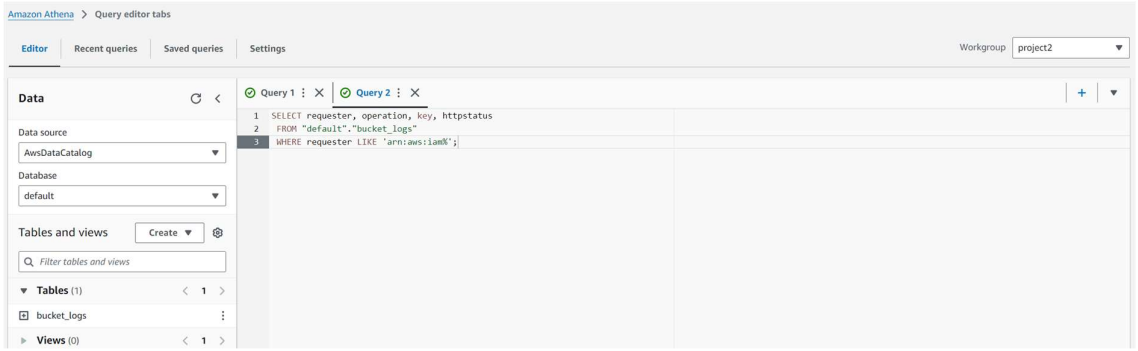
Then created a workgroup and configured it to store the results in the “athena results” bucket.

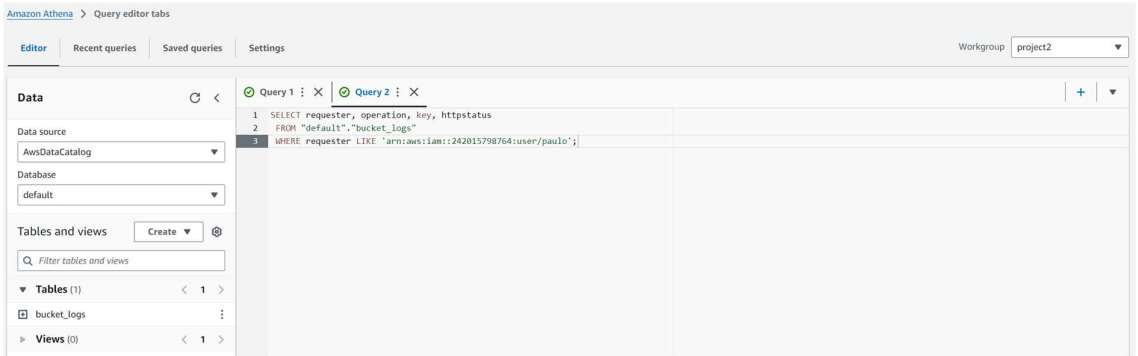


I ran the query in the workgroup that I just created. This created the table named “bucket_logs”.



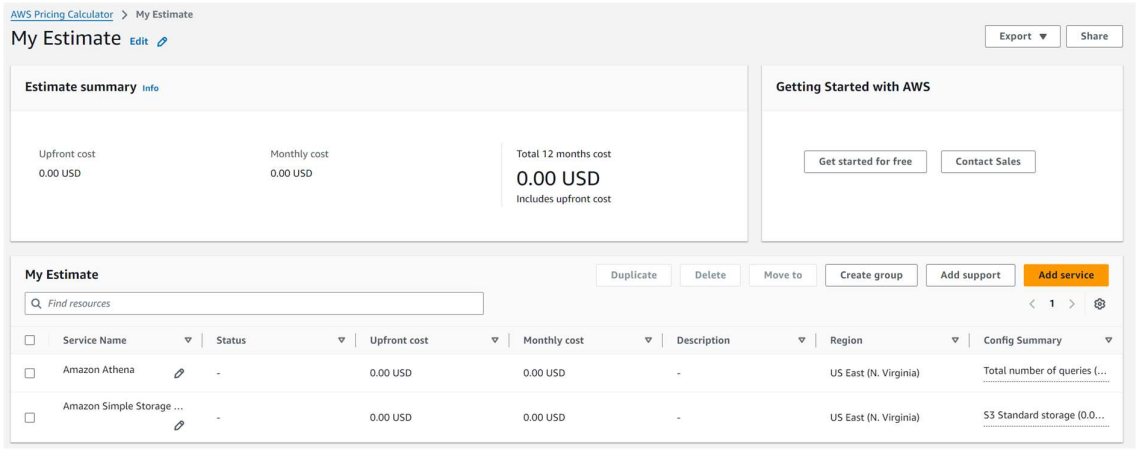
I used the below query to get records for actions taken as the paulo user, because that user can access the bucket and its objects.

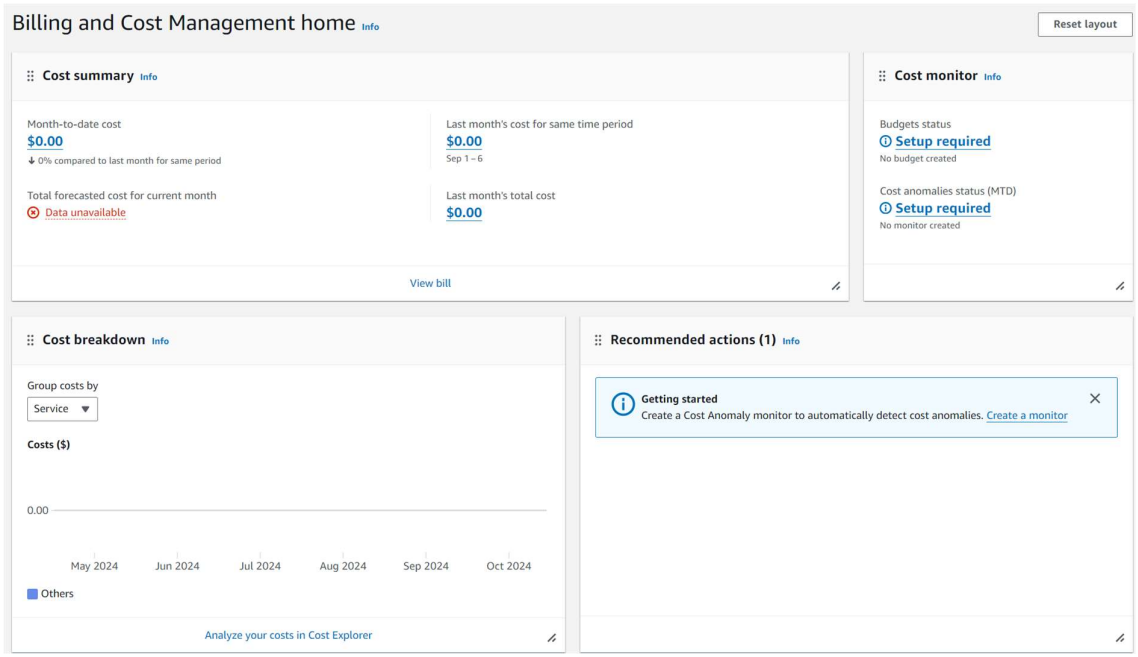




Cost assessment to secure Amazon S3

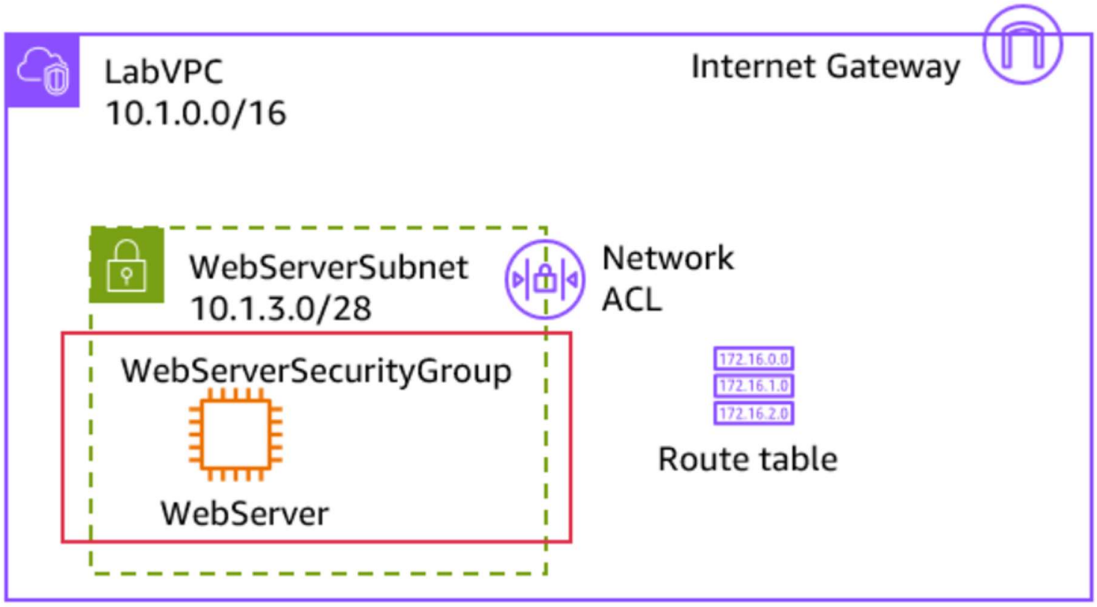
For the low size files that were used there was no additional costs for using versioning, service access logging, and S3 Inventory on a bucket. And Athena and S3 Select to analyze access logs.





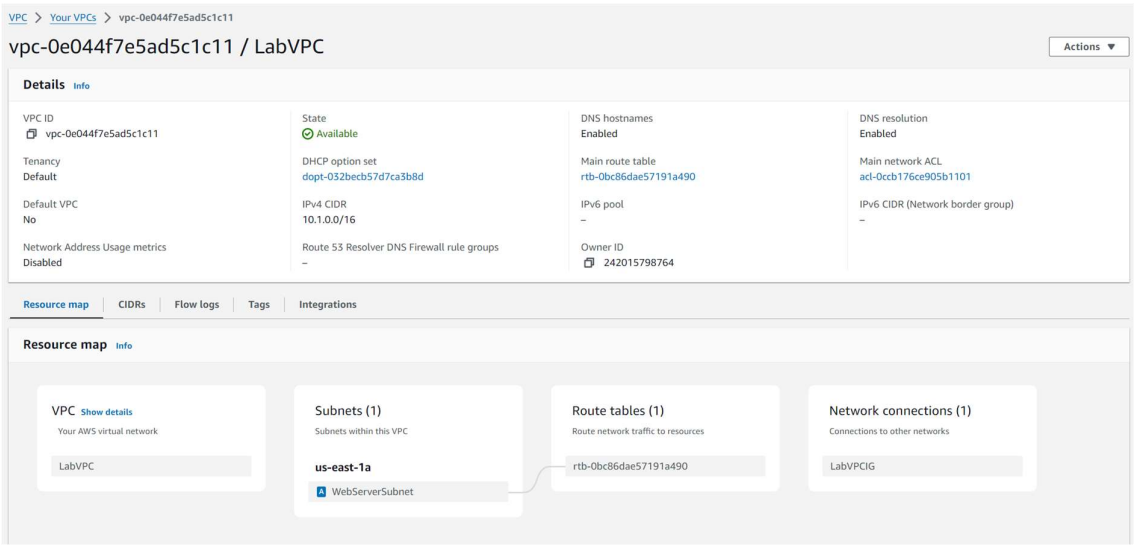
Phase 2: Securing VPCs

The following diagram shows the resources that already exist in the lab environment.

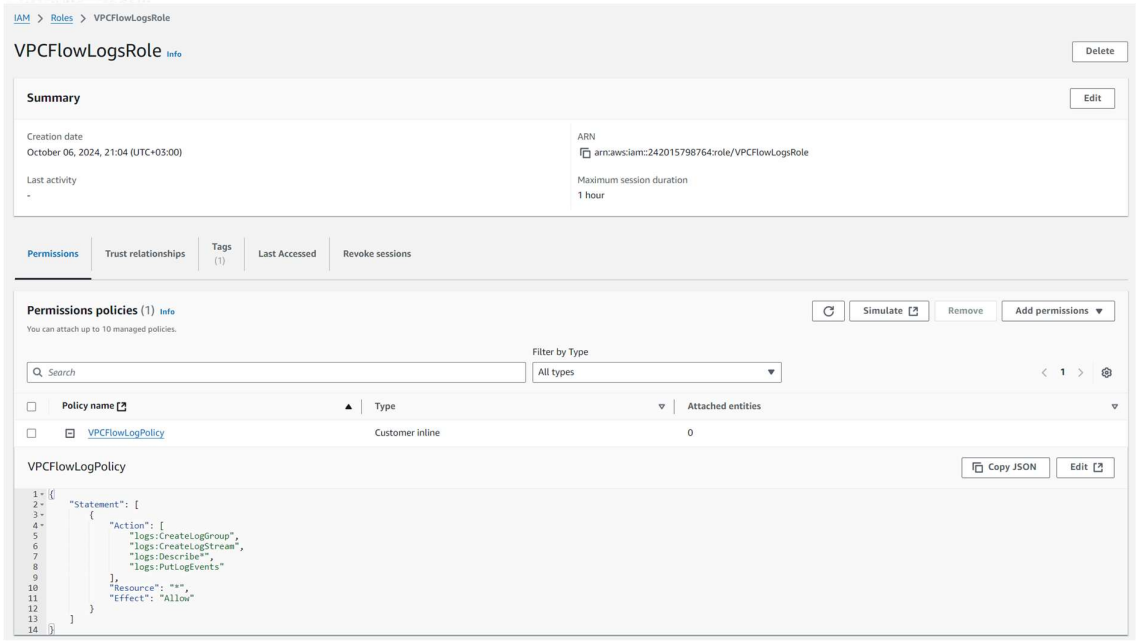


Task 2.1: Review LabVPC and its associated resources

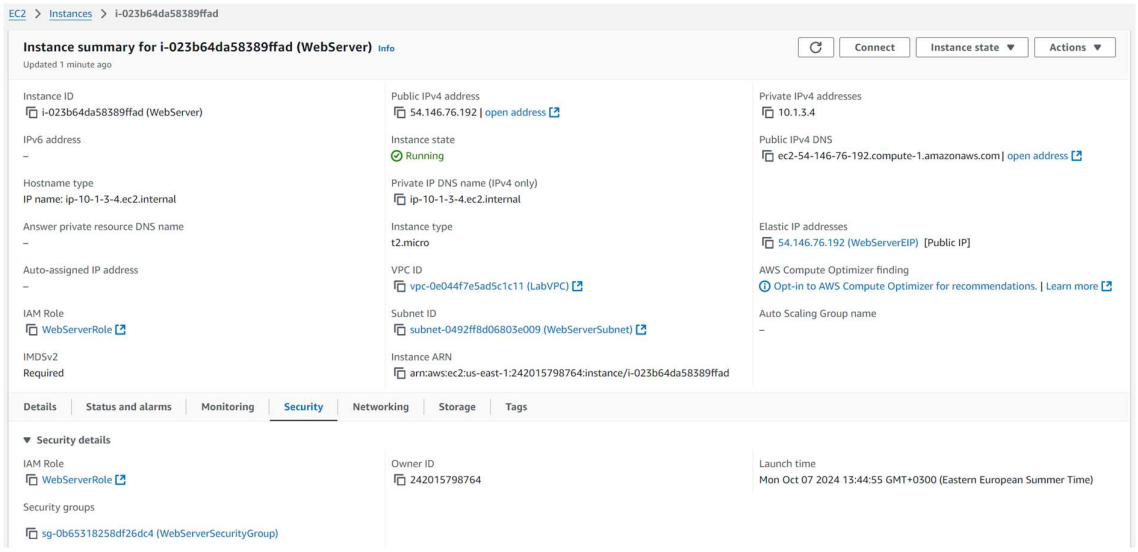
analyzed the resource map of the existing LabVPC. It contains the following resources: A subnet named WebServerSubnet, A route table which is the main route table that is created by default for any VPC, and an internet gateway.



Review the permissions that are granted to the “VPCFlowLogsRole” IAM role.

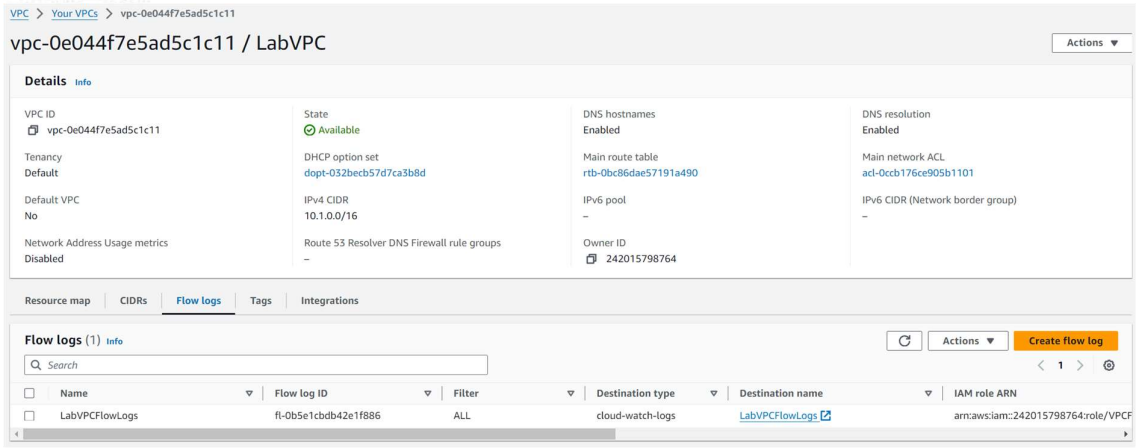


Then I observed the details for the WebServer instance.

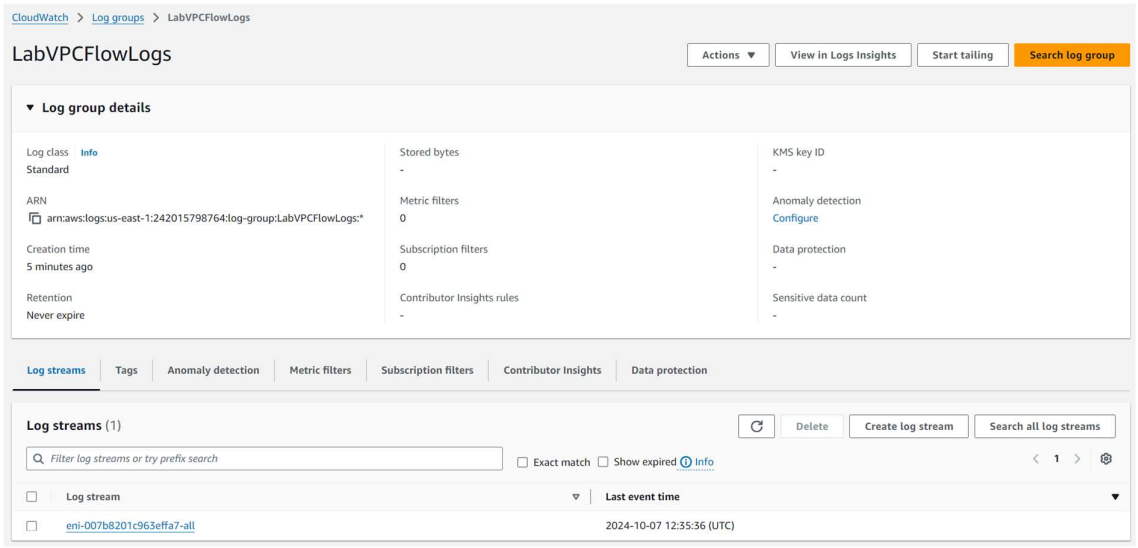


Task 2.2: Create a VPC flow log

Create VPC flow logs in our “LabVPC”.

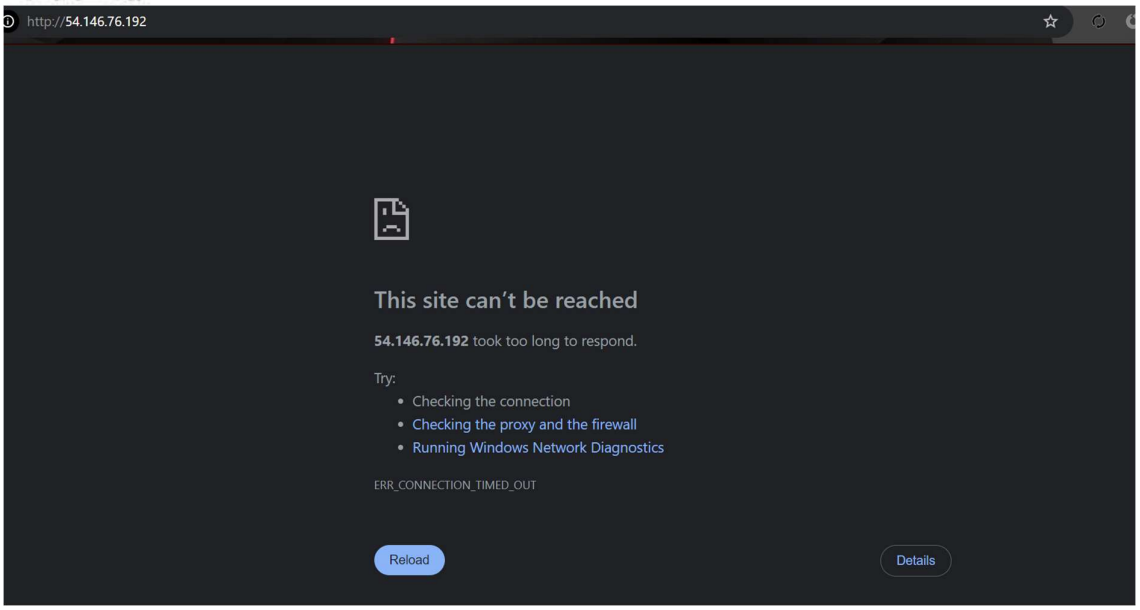


And I put the Destination log group to the group named “LabVPCFlowLogs”.



Task 2.3: Access the WebServer instance from the internet and review VPC flow logs in CloudWatch

Tried to access the web server instance from its public IP but it is not reachable

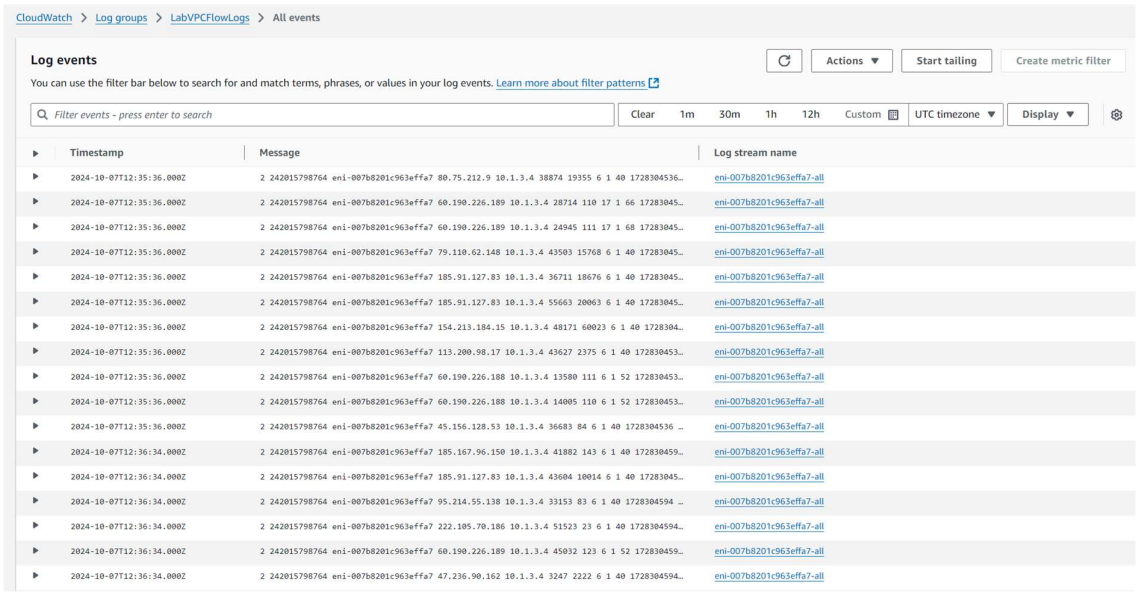


Tested the connectivity to it from port 80 but the command is loading and will lead to timeout. It means that it can't reach the instance from port 80.

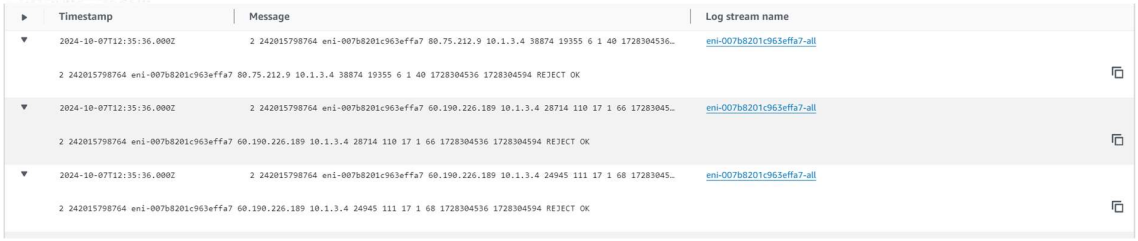
I also tried testing connection for port 22 but same problem

```
voclabs:~/environment $
voclabs:~/environment $ nc -vz 54.146.76.192 80
^C
voclabs:~/environment $ nc -vz 54.146.76.192 22
^C
voclabs:~/environment $
```

There are many logs in the log group



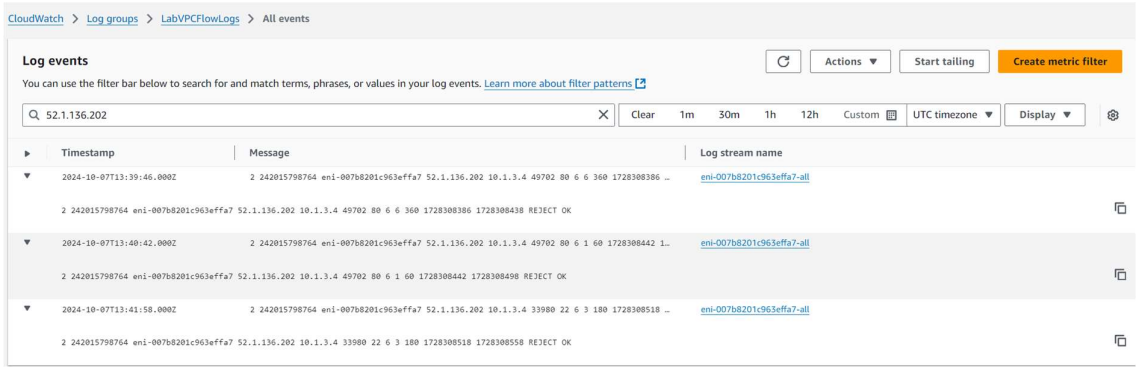
And Message field for all entries indicate "Reject".



I used the below command to get the IP of the Cloud9 instance.

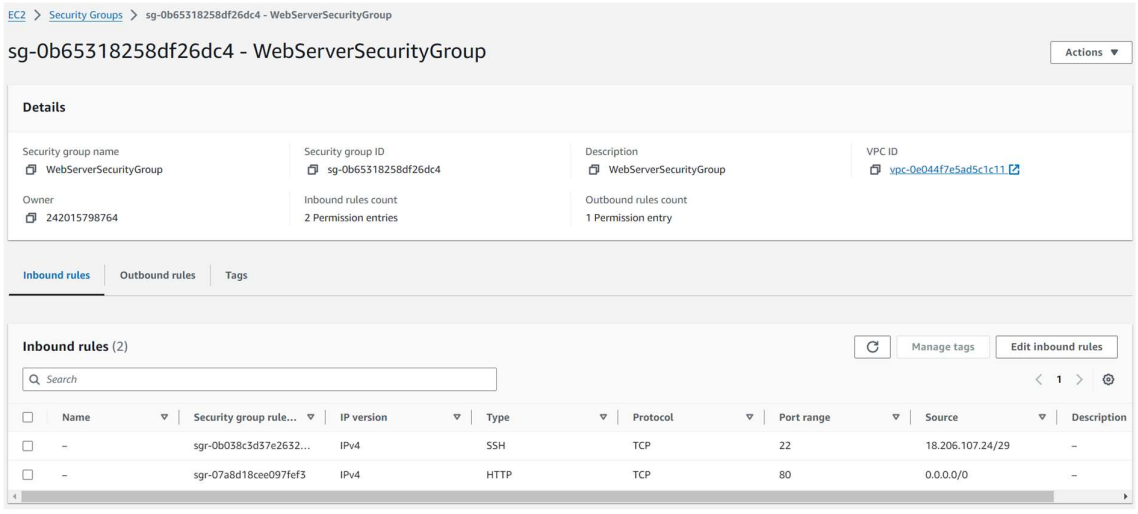
```
voclabs:~/environment $
voclabs:~/environment $ curl http://169.254.169.254/latest/meta-data/public-ipv4
52.1.136.202voclabs:~/environment $
voclabs:~/environment $
```

I filtered the logs with the cloud9 IP to get the logs related to it



Task 2.4: Configure route table and security group settings

I edited the security group for the web server instance to allow the HTTP from anywhere (0.0.0.0/0) and SSH from an IP range that EC2 Instance Connect uses in the us-east-1 Region. I also deleted unnecessary rules.



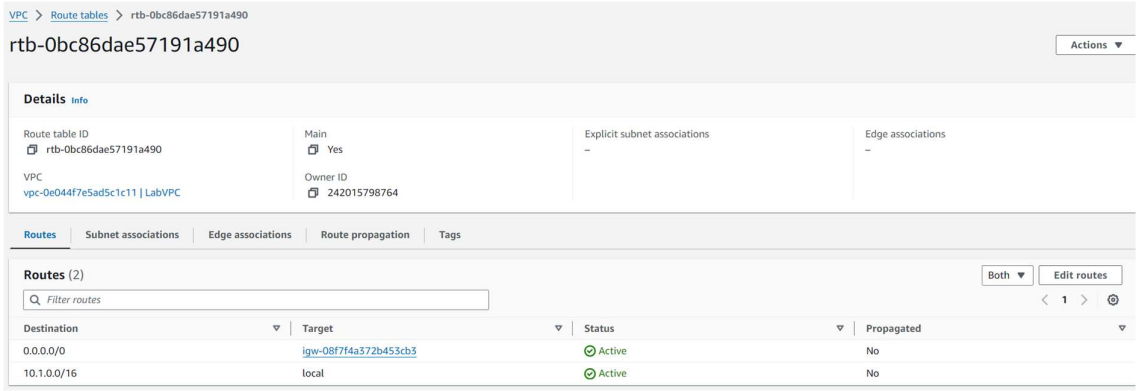
I got the IP range using the below commands

```
Installed:
  jq.x86_64 0:1.5-1.amzn2.0.2

Dependency Installed:
  oniguruma.x86_64 0:5.9.6-1.amzn2.0.7

Complete!
voclabs:~/environment $ curl https://ip-ranges.amazonaws.com/ip-ranges.json| jq -r '.prefixes[] | select(.region=="us-east-1") | select(.service=="EC2_INSTANCE_CONNECT") | .ip_prefix'
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 1673k  100 1673k    0     0  13.5M      0 --:--:-- --:--:-- --:--:-- 13.6M
18.206.107.24/29
voclabs:~/environment $
```

If we test from the cloud9 instance again, it will still not work.

```
voclabs:~/environment $
voclabs:~/environment $ nc -vz 54.146.76.192 80
^C
voclabs:~/environment $ nc -vz 54.146.76.192 22
^C
voclabs:~/environment $
```

So I added route in the route table that is associated with the web server instance to route traffic through the internet gateway

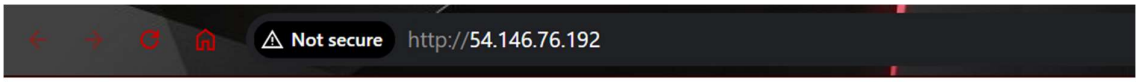


I

And now it succeeded

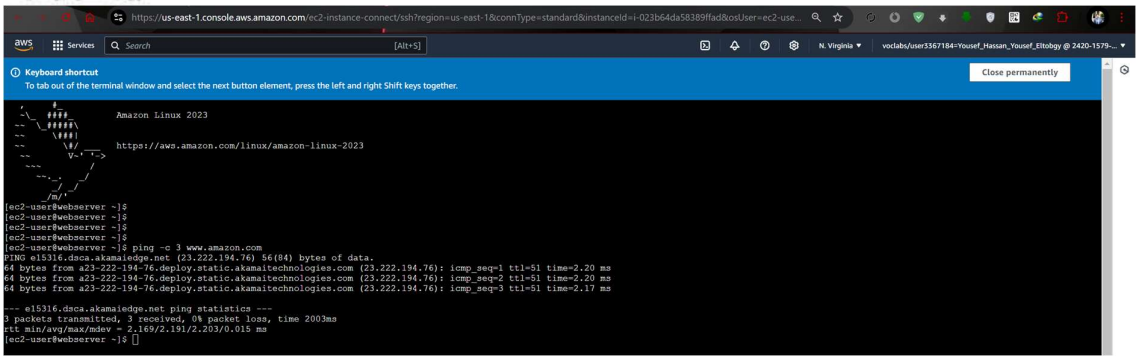
```
voclabs:~/environment $
voclabs:~/environment $ nc -vz 54.146.76.192 80
Connection to 54.146.76.192 80 port [tcp/http] succeeded!
voclabs:~/environment $
```

And we can access the webserver from the browser window



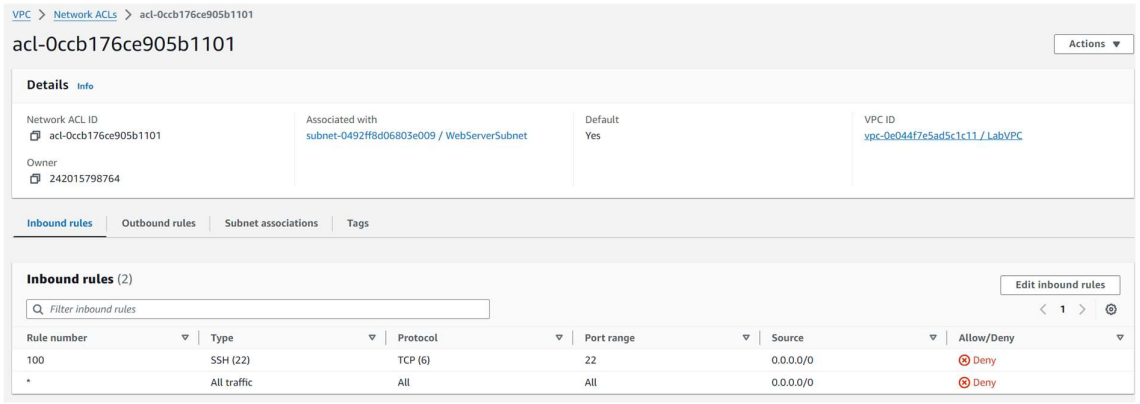
Hello world from WebServer!

And the SSH is working too through EC2 instance connect

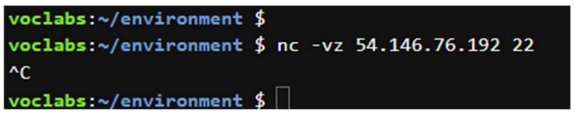


Task 2.5: Secure the WebServerSubnet with a network ACL

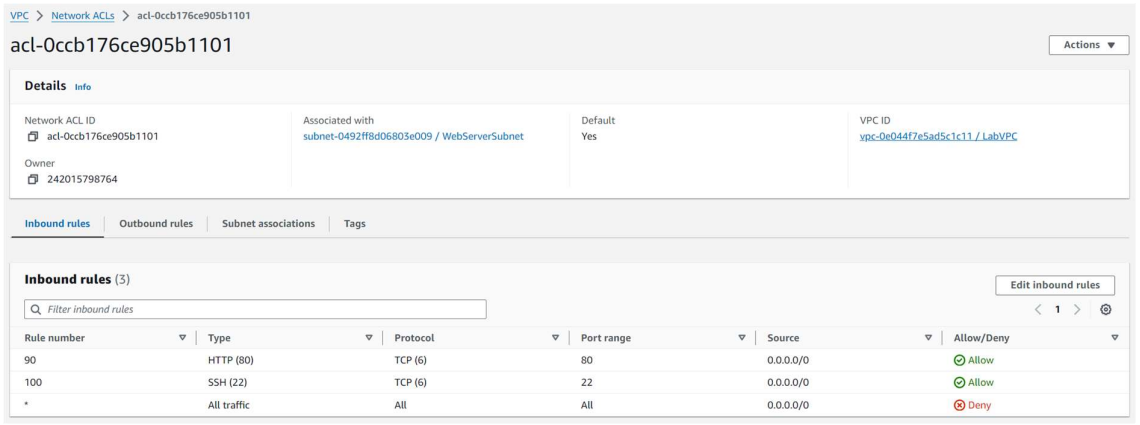
If I edited the rule number 100 from allow to deny the connection through port 22 will fail.



And it really failed from the cloud9 instance.



So now we allow the traffic on ports 80 and 22 in the NACL.

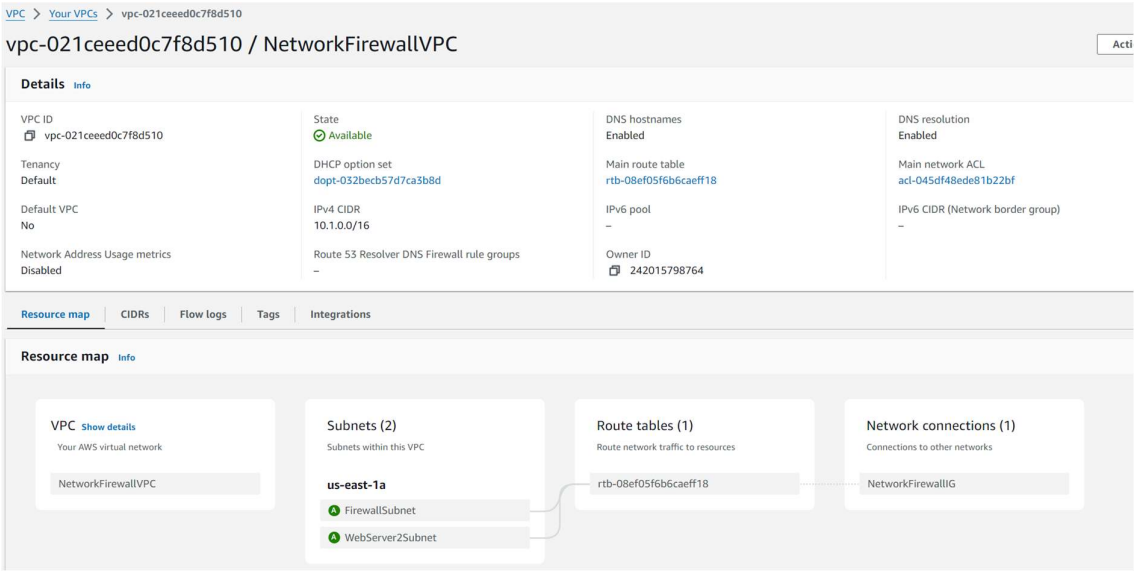


And connection worked again fine.

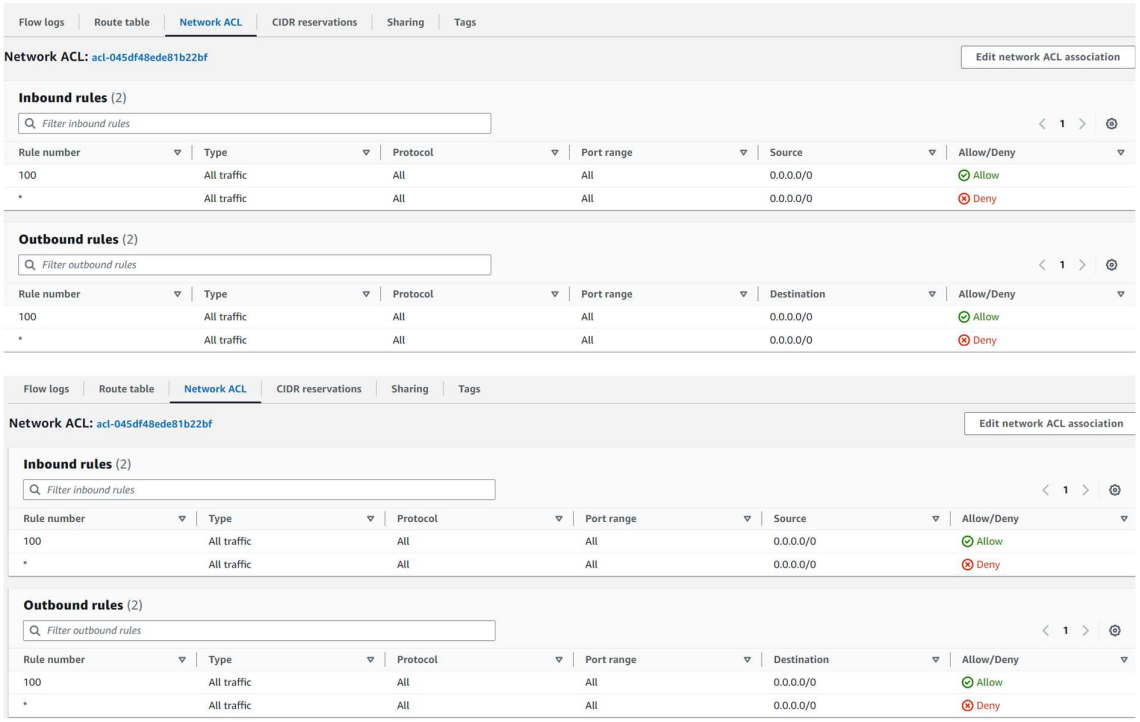
```
voclabs:~/environment $ nc -vz 54.146.76.192 80
Connection to 54.146.76.192 80 port [tcp/http] succeeded!
voclabs:~/environment $
```

Task 2.6: Review NetworkFirewallVPC and its associated resources

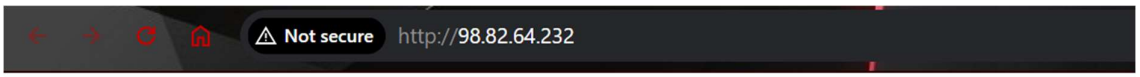
Here we have the below properties in the VPC named NetworkFirewallVPC.



The NACL of the two subnets allow all traffic.

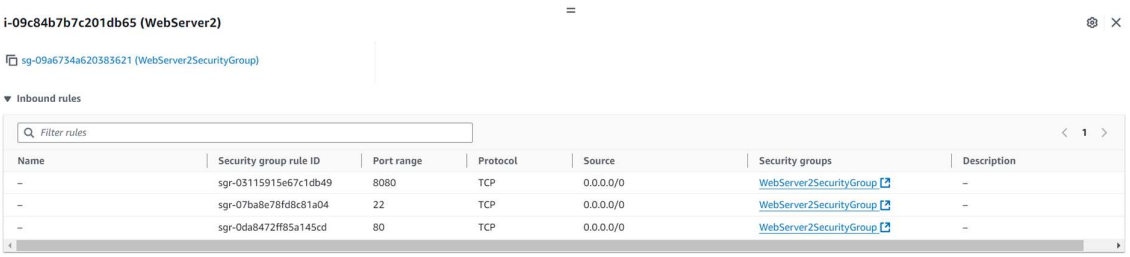


This WebServer2 instance is deployed in this VPC and it is reachable fine.

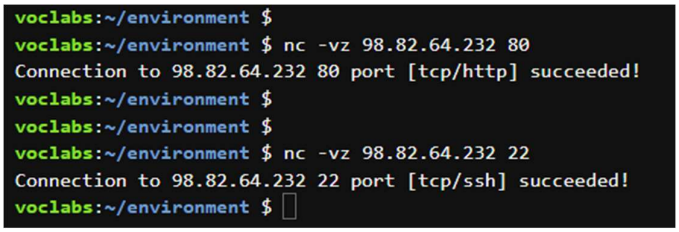


Hello world from WebServer2!

The below are the rules in its security group

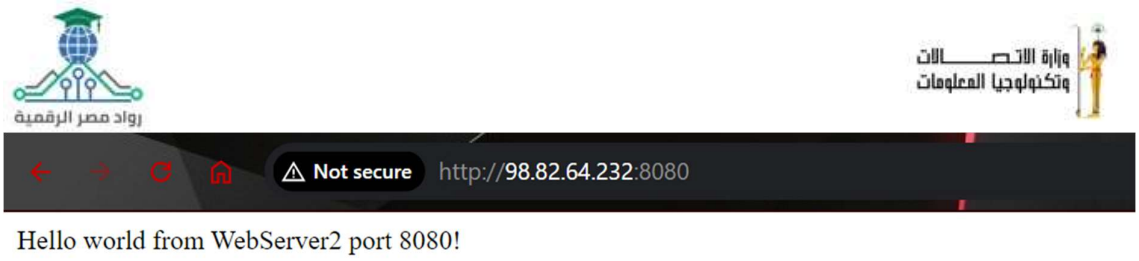


And connection is working fine from the cloud9



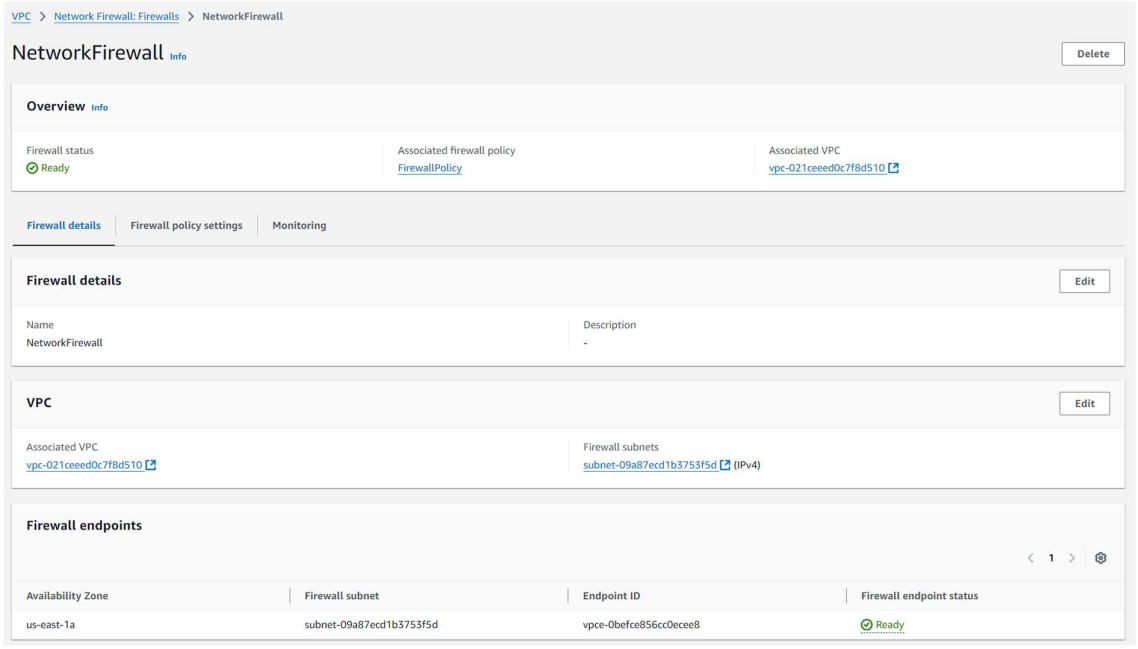
I will now test the connection on port 80 so I have to run the below command in the WebServer2 instance.

And then it will work fine.



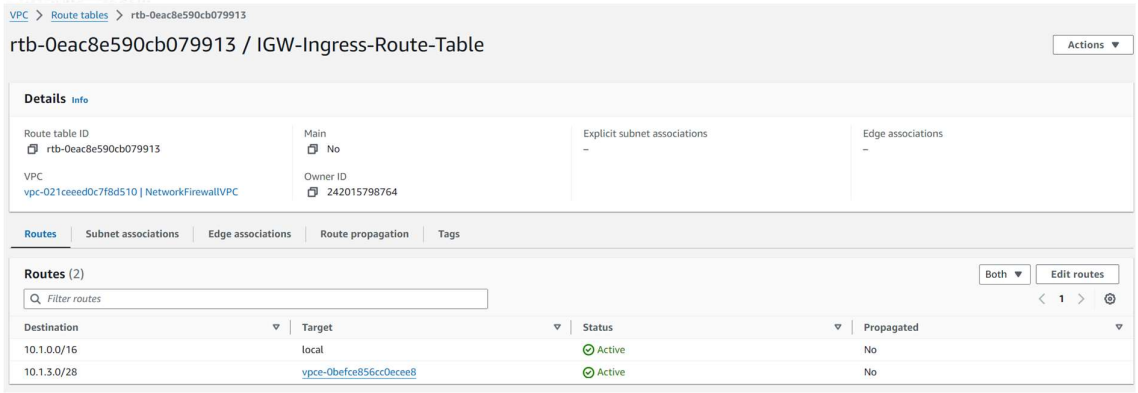
Task 2.7: Create a network firewall

I created the below firewall in the us-east-1a Availability Zone for FirewallSubnet and IP address type IPv4. And Named the firewall policy as FirewallPolicy.

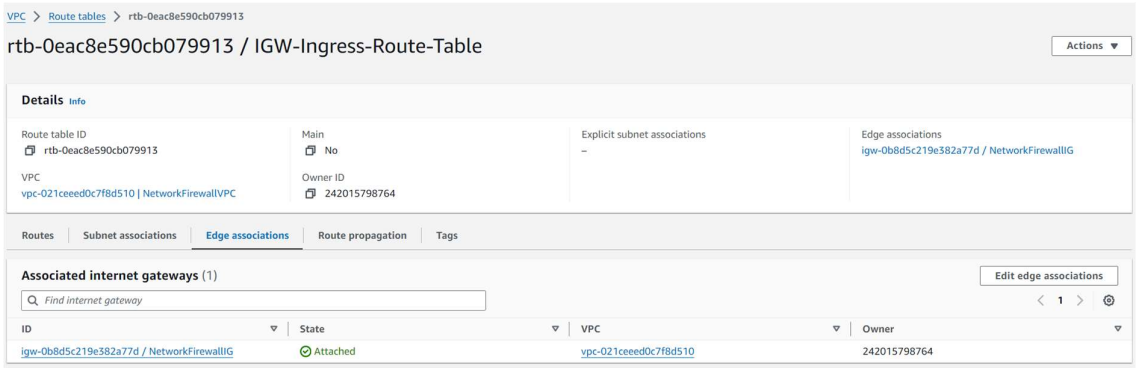


Task 2.8: Create route tables

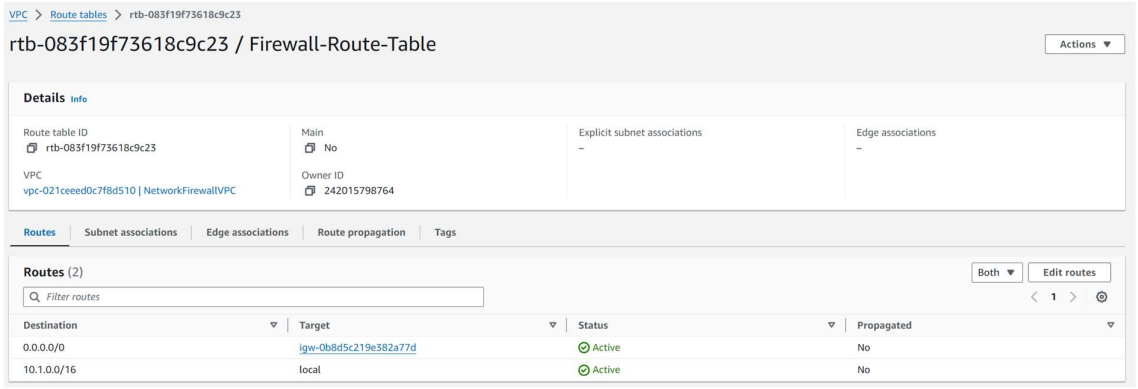
Created the below route tables and added route where I set the destination to be the CIDR block of the subnet that the WebServer2 instance runs in, and I set the target to the only Gateway Load Balancer Endpoint that is available.



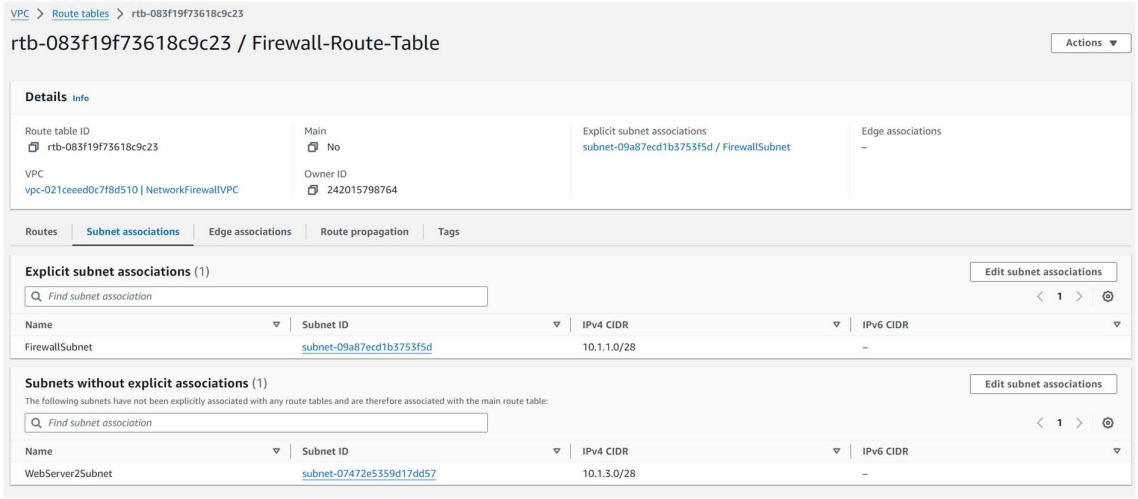
I then added an edge association to the IGW-Ingress-Route-Table so that the NetworkFirewallIG internet gateway is associated with the route table.



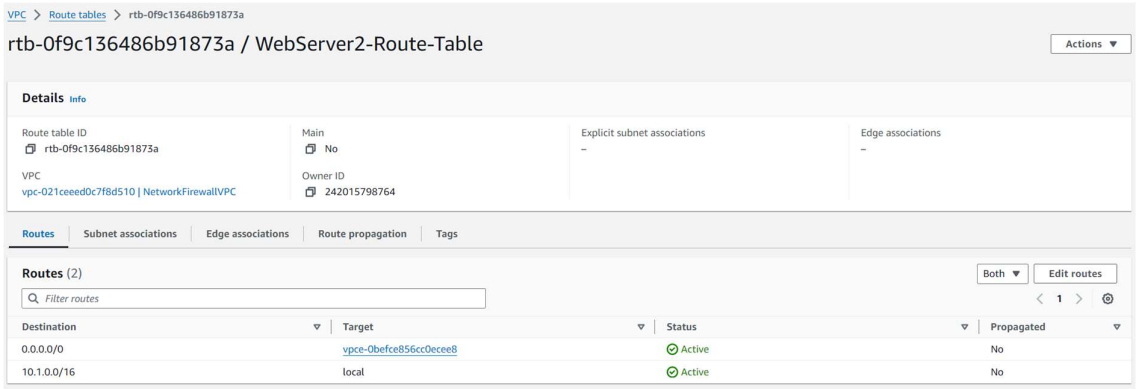
I then created a new route table.



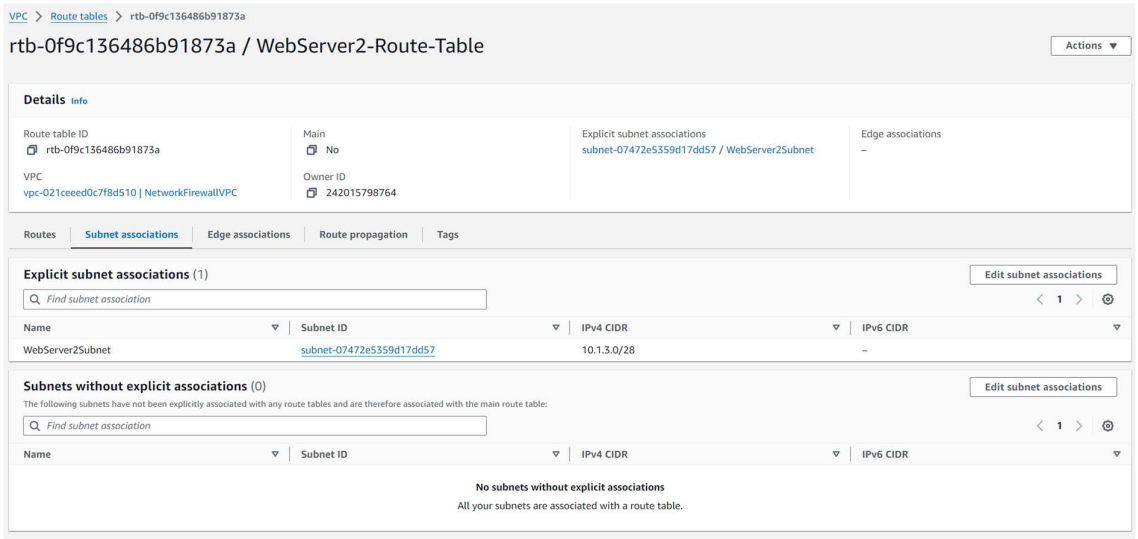
Then I associated it with the subnet named “FirewallSubnet”.



Then I created another route table named "WebServer2-Route-Table". And added a route so that 0.0.0.0/0 traffic is routed to the Gateway Load Balancer Endpoint.

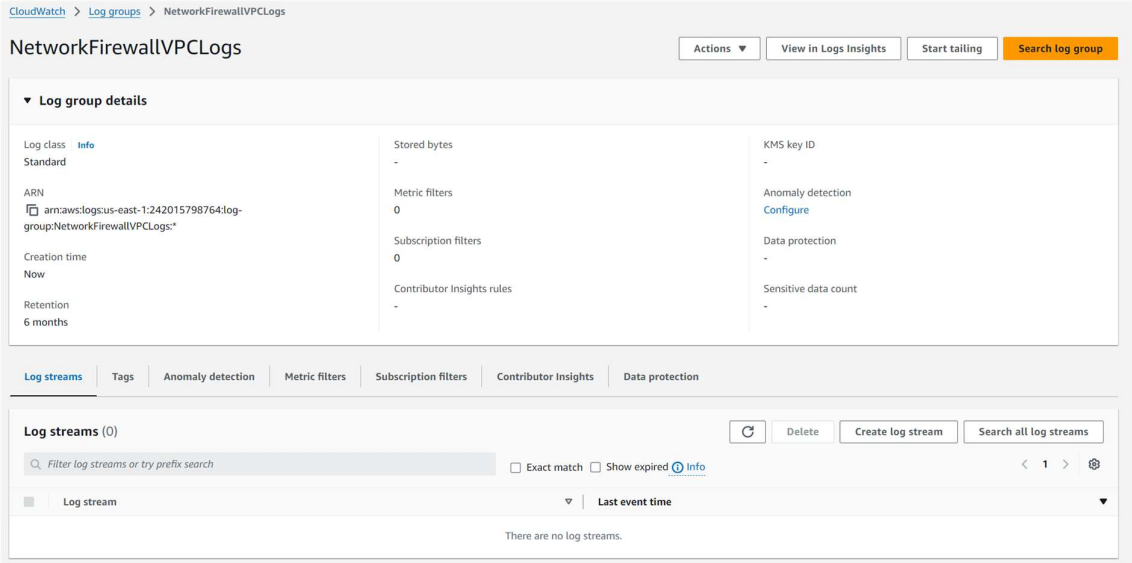


Then created association between the “Webserver2Subnet” subnet and the route table

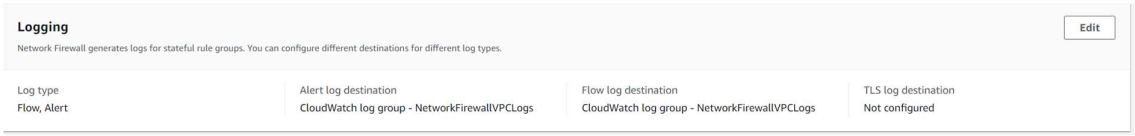


Task 2.9: Configure logging for the network firewall

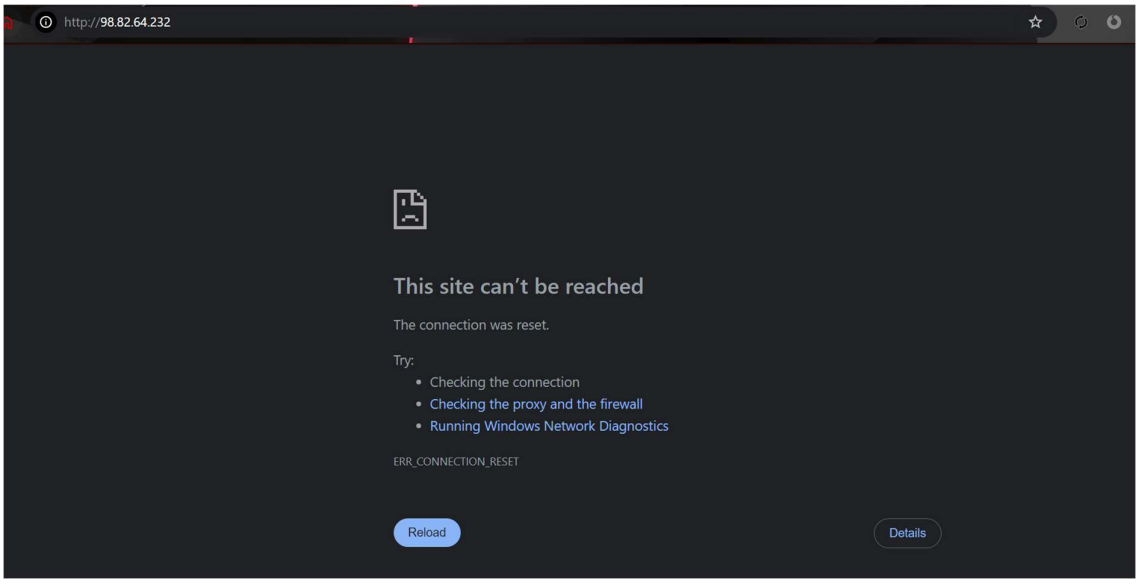
I created a log group



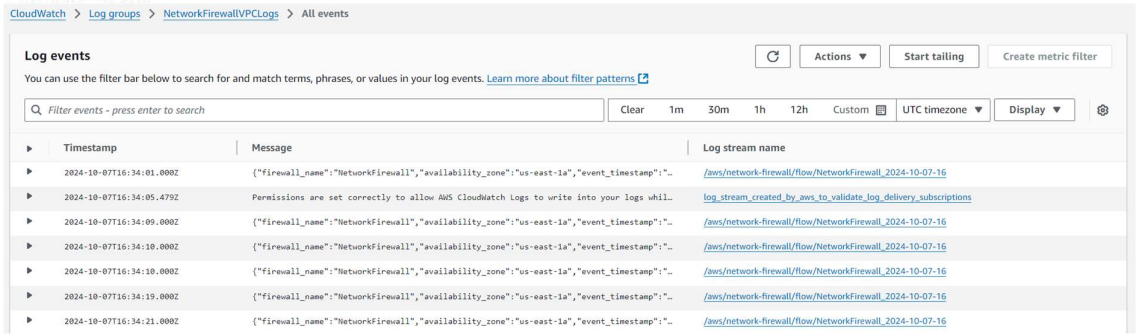
I then configured logging in the firewall that we created.



The webserver2 will not be accessible. This is expected because the network firewall policy isn't yet configured to allow HTTP traffic on port 80.

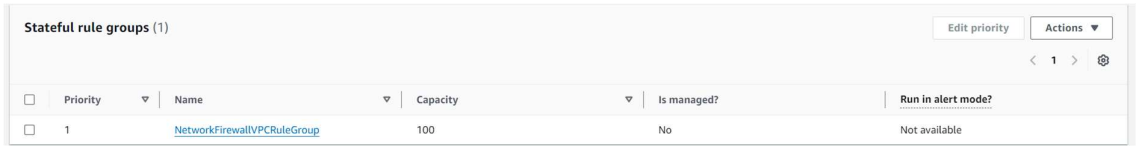


And we can check the logs in the cloudwatch log group "NetworkFirewallVPCLogs".

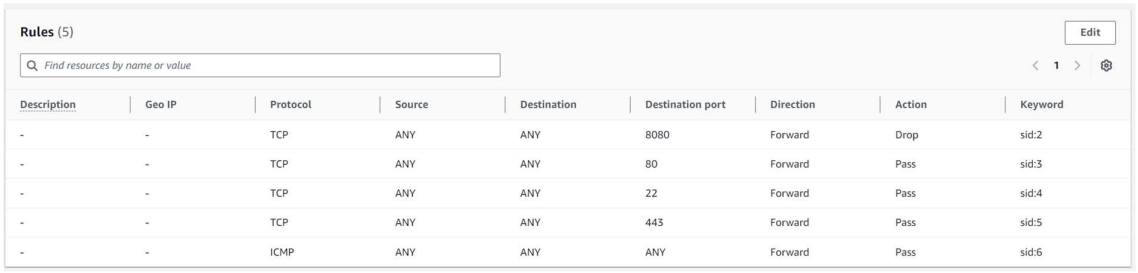


Task 2.10: Configure the firewall policy and test access

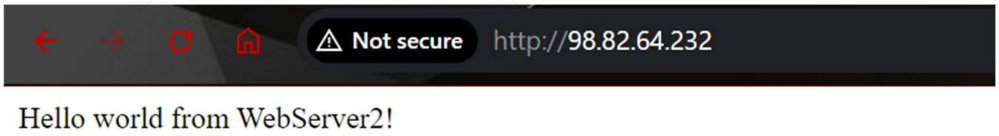
I then created a stateful route group in the firewall policy to allow connection on port 80, 22, 443 and ICMP protocol, but I set it to reject connection on port 8080.



Here are the rules.



And now the WebServer2 is accessible again. The below is on port 80.



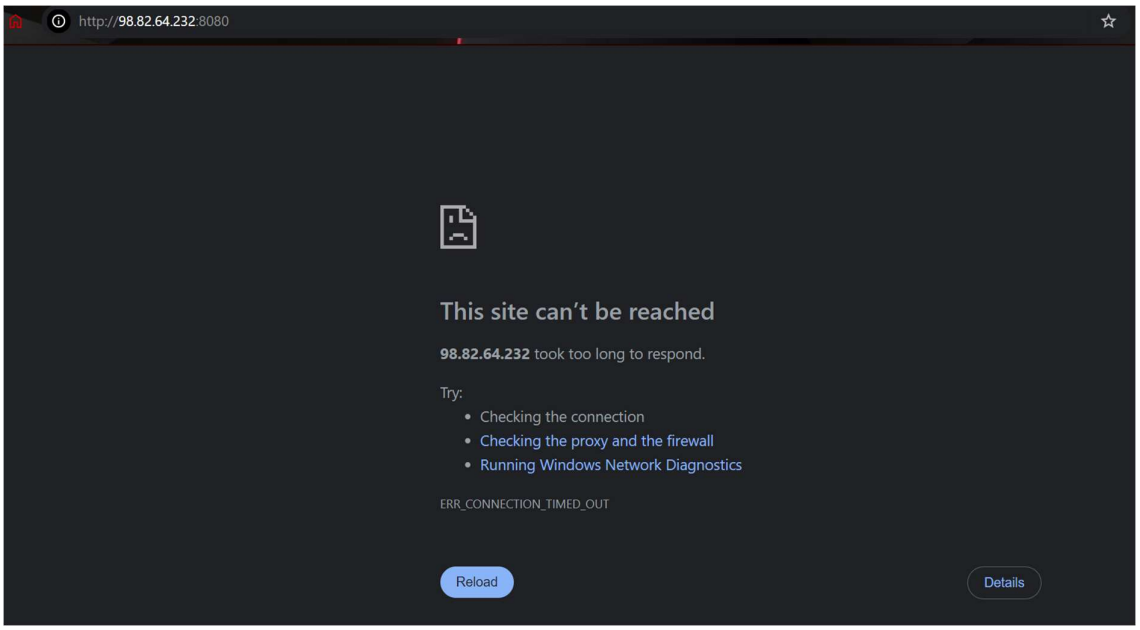
Hello world from WebServer2!

Here the below is on port 22 and it says successful from the cloud9 instance.

The ICMP is also working fine and we can see the active ports that the WebServer2 instance is listening to.

```
[ec2-user@webserver2 ~]$
[ec2-user@webserver2 ~]$ ping -c 3 www.amazon.com
PING www-amazon-com.customer.fastly.net (162.219.225.118) 56(84) bytes of data.
64 bytes from 162.219.225.118 (162.219.225.118): icmp_seq=1 ttl=56 time=3.29 ms
64 bytes from 162.219.225.118 (162.219.225.118): icmp_seq=2 ttl=56 time=3.28 ms
64 bytes from 162.219.225.118 (162.219.225.118): icmp_seq=3 ttl=56 time=2.67 ms

--- www-amazon-com.customer.fastly.net ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 2.665/3.077/3.285/0.291 ms
[ec2-user@webserver2 ~]$
[ec2-user@webserver2 ~]$ sudo netstat -tulpn | grep -i listen
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      2326/sshd: /usr/sbi
tcp        0      0 0.0.0.0:8080            0.0.0.0:*               LISTEN      4857/python3
tcp6       0      0 :::22                   :::*                    LISTEN      2326/sshd: /usr/sbi
tcp6       0      0 :::80                   :::*                    LISTEN      1993/httpd
[ec2-user@webserver2 ~]$
```

And if we tried to access on port 8080, the connection will fail because of the firewall rule.

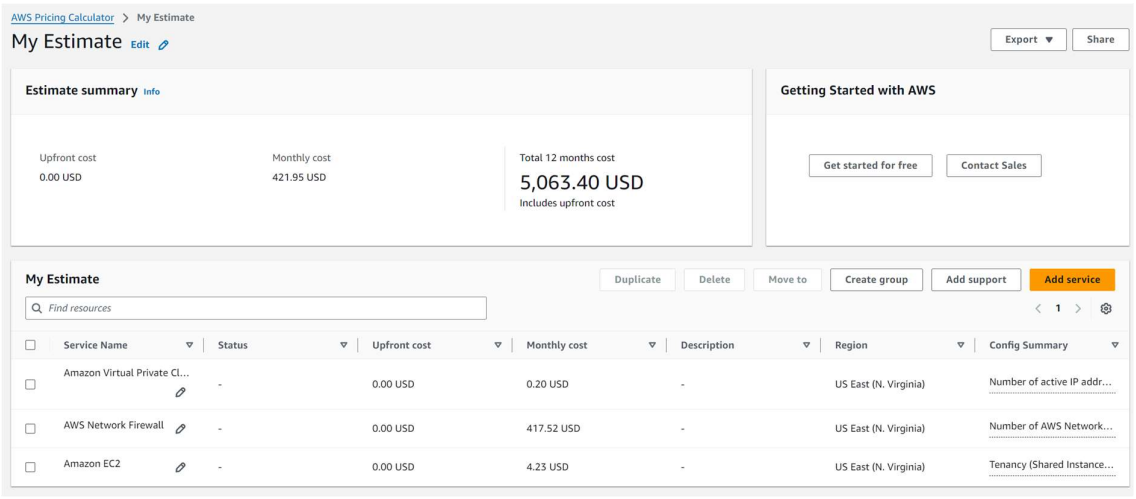


Cost estimate to secure a VPC with a network firewall

We need to get the estimated cost for the below data

CRITERIA	ESTIMATE PER MONTH
EC2 - WebServer2	1 On-demand (Shared Instance, 100% usage)
VPC - NetworkFirewallVPC	1 IPAM
Inbound data transfer from (internet free)	100 GB
Outbound data transfer to (US East, N. Virginia) - outbound not Intra-Region	1 TB
Network firewall - number of endpoints	1
Network firewall - usage per endpoint	30 days
Network firewall - data processed per month	2.0 TB

So, here is the cost estimate:



Here is the cost estimate in the pdf format.

Export date: **10/7/2024**

Language: **English**

Estimate URL: **https://calculator.aws/#/estimate?id=56d9639499ef1e1b45993e9079d27127a33df9ee**

Estimate summary

Upfront cost	Monthly cost	Total 12 months cost
0.00 USD	**421.95 USD**	**5,063.40 USD** Includes upfront cost

Detailed Estimate

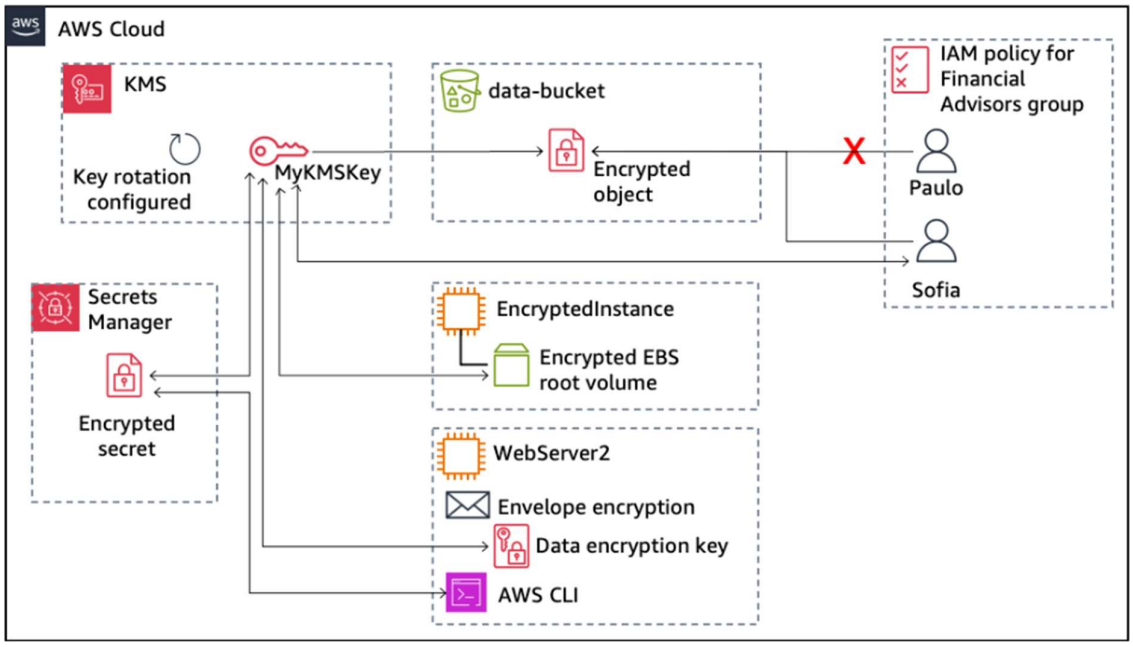
Name	Group	Region	Upfront cost	Monthly cost
Amazon Virtual Private Cloud (VPC)	No group applied	US East (N. Virginia)	0.00 USD	0.20 USD
Status: - **Description:** **Config summary:** Number of active IP addresses (1)				
AWS Network Firewall	No group applied	US East (N. Virginia)	0.00 USD	417.52 USD
Status: - **Description:** **Config summary:** Number of AWS Network Firewall endpoints (1), Usage per endpoint (30 days), Data processed per month (2 TB)				
Amazon EC2	No group applied	US East (N. Virginia)	0.00 USD	4.23 USD
Status: - **Description:** **Config summary:** Tenancy (Shared Instances), Operating system (Linux), Workload (Consistent, Number of instances: 1), Advance EC2 instance (t2.micro), Pricing strategy (Compute Savings Plans 3yr No Upfront), Enable monitoring (disabled), DT Inbound: Internet (100 GB per month),				

Here is it in the csv format.

Estimate summary								
Upfront co	Monthly cc	Total 12 m	Currency					
0	421.95	5063.4	USD					
		* Includes upfront cost						
Detailed Estimate								
Group hier	Region	Descriptio	Service	Upfront	Monthly	First 12 m	Currency	Status	Configuration summary
My Estima	US East (N. Virginia)	IPAM		0	0.2	2.4	USD		Number of active IP addresses (1)
My Estima	US East (N. Virginia)	AWS Netw		0	417.52	5010.24	USD		Number of AWS Network Firewall endpoints (1), Usage per endpoint (30 days), Data processed per month (2 TB)
My Estima	US East (N. Virginia)	Amazon EC		0	4.234	50.81	USD		Tenancy (Shared Instances), Operating system (Linux), Workload (Consistent, Number of instances: 1), Advance EC2 instance (t2.micro), Pricing strategy (Compute Savings Plans 3yr No
Acknowledgement									
* AWS Pricing Calculator provides only an estimate of your AWS fees and doesn't include any taxes that might apply. Your actual fees depend on a variety of factors, including your actual usage of AWS services.									

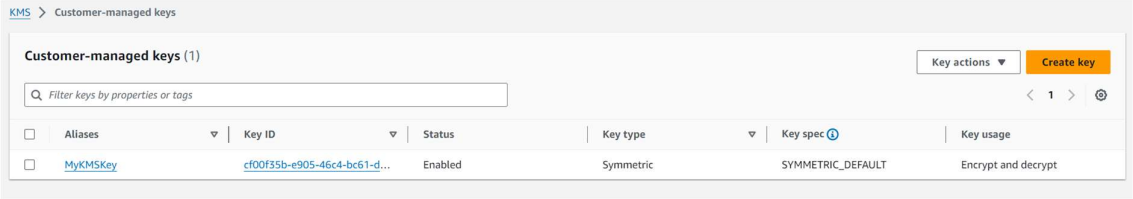
Phase 3: Securing AWS resources by using AWS KMS

By the end of this phase, you will have created the architecture that is shown in the following diagram.

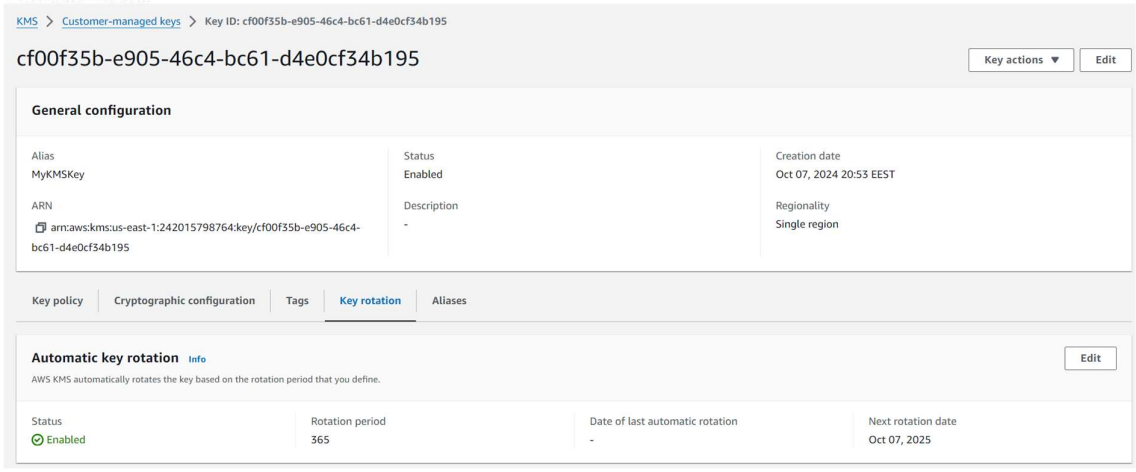


Task 3.1: Create a customer managed key and configure key rotation

Created a new key and granted the Key administrator and Key user permissions to the voclabs role

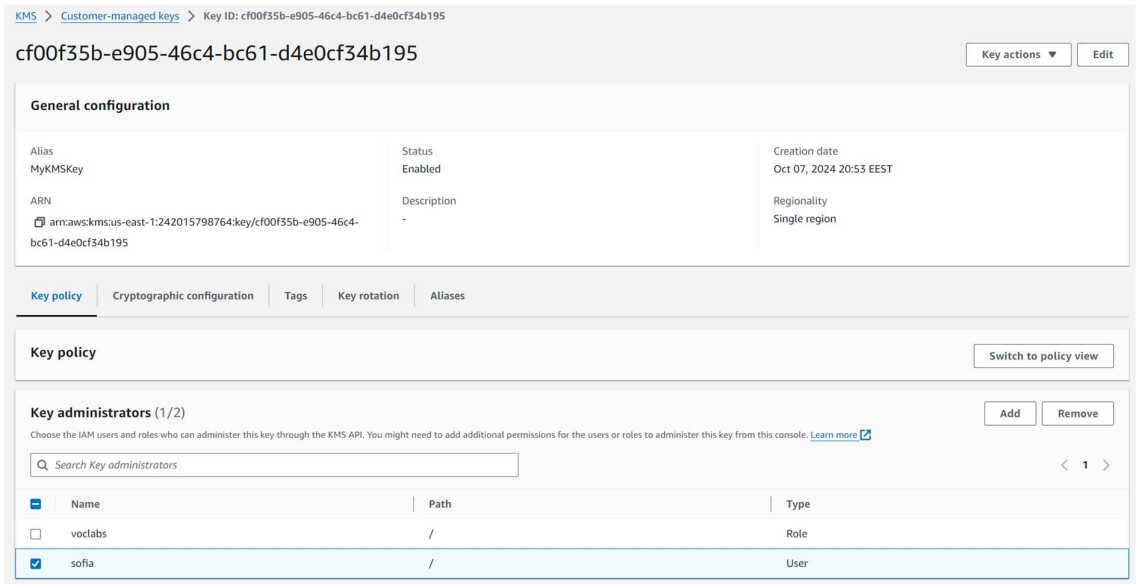


I then enabled the automatic key rotation. So that it is automatically rotated every year.



Task 3.2: Update the AWS KMS key policy and analyze an IAM policy

Added user named Sofia to the key administrators



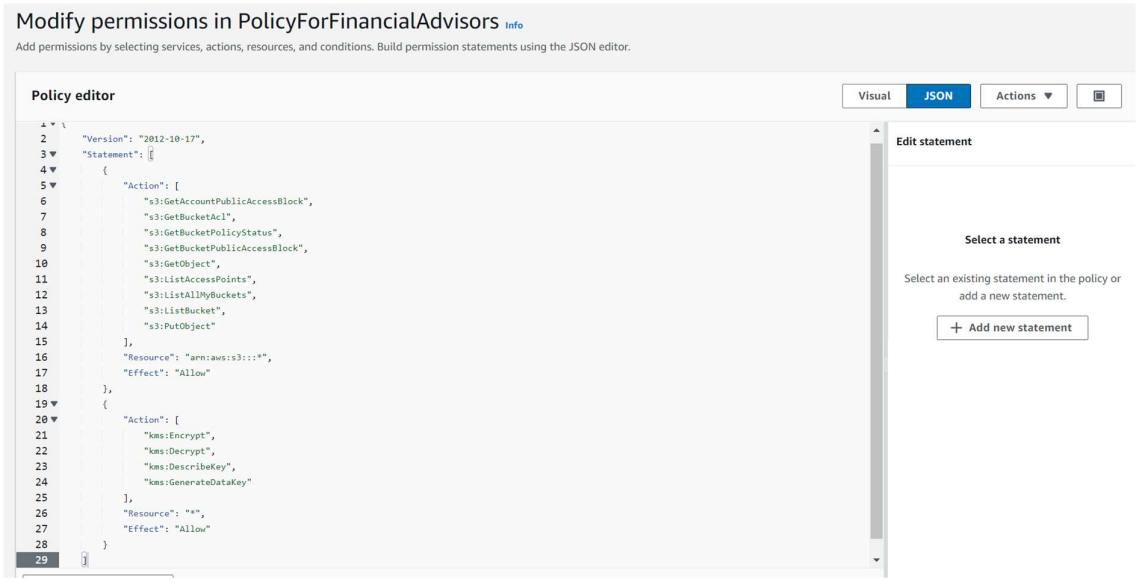
And here it is modified in the configuration

Key policy | Cryptographic configuration | Tags | Key rotation | Aliases

Key policy

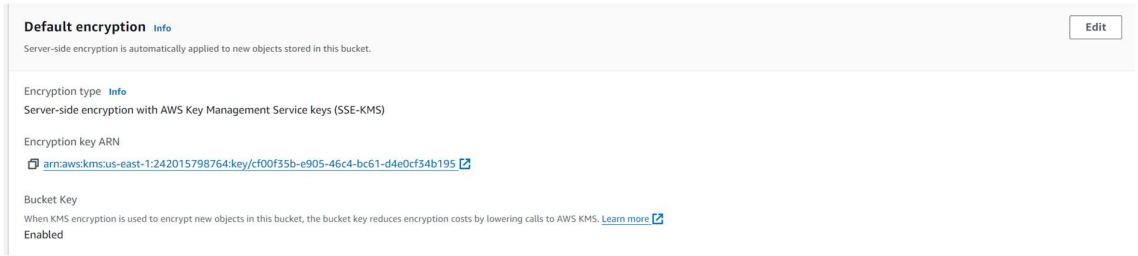
```
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Allow access for Key Administrators",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::242015798764:role/voclabs",
                    "arn:aws:iam::242015798764:user/sofia"
                ]
            },
```

I analyzed the policy named “PolicyForFinancialAdvisors”, It is configured to allow full actions on S3 (like PUT, GET, etc.) and also allow some actions, like: encrypt, decrypt, etc.



Task 3.3: Use AWS KMS to encrypt data in Amazon S3

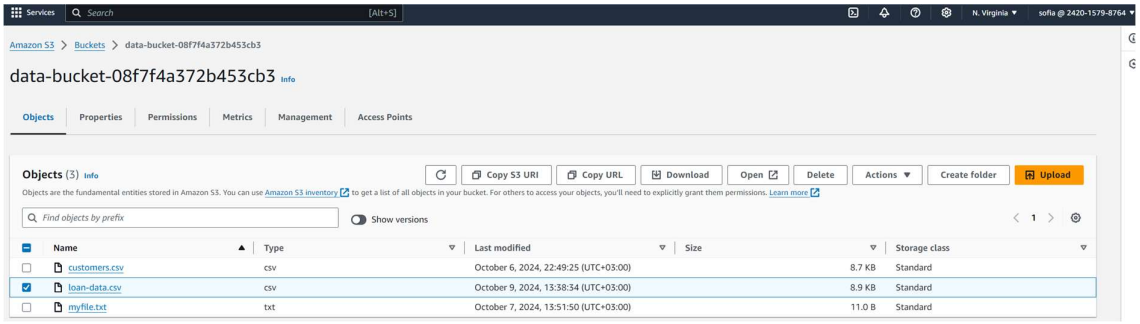
Modified the encryption settings on the data-bucket S3 bucket so that the bucket uses SSE-KMS encryption.



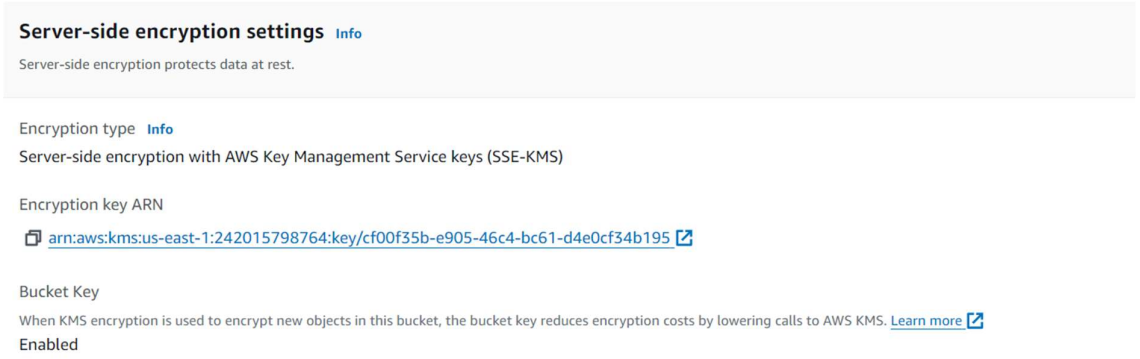
Created a file named “loan-data.csv” with the below data.

	A	B	C	D	E
1	date	descriptio	amount	principal	interest
2	1/14/2023	payment	1000	845.52	154.48
3	12/22/2022	payment	1021.52	742.8	278.72
4	11/15/2022	payment	1000	855.27	144.73

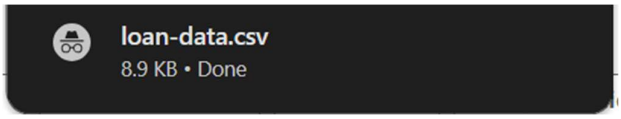
I uploaded it to the S3 bucket using the user named “Sofia”.



And The encryption type for the object is SSE-KMS.



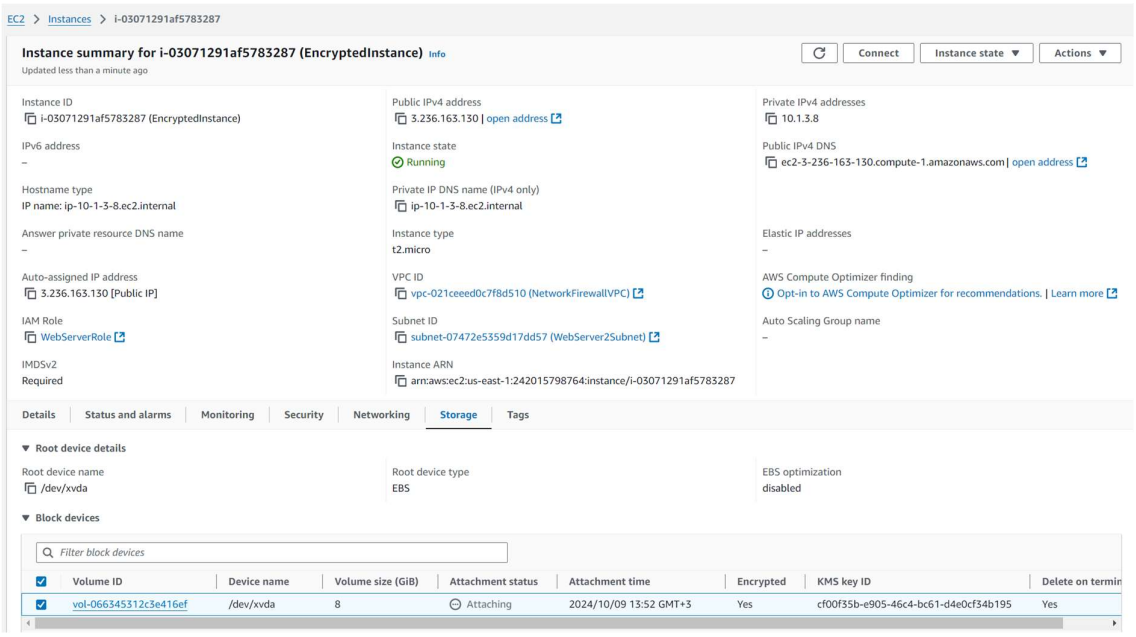
I successfully downloaded the file using the user Sofia. This means that the user can retrieve the KMS key and detect the file on S3 bucket and download the file in plain text.



If I try to open the file using the user Paulo, it will result in error and tell me the user is not authorized to perform: kms:Decrypt on resource.

Task 3.4: Use AWS KMS to encrypt the root volume of an EC2 instance

I created new EC2 instance and enabled encryption on its AMI root volume by using MyKMSKey.



Task 3.5: Use AWS KMS envelope encryption to encrypt data in place

I connected to the webserver2 instance and created a file in it.

```
[ec2-user@webserver2 ~]$
[ec2-user@webserver2 ~]$ echo "Let's encrypt these file contents. Sensitive data here." > data_unencrypted.txt
[ec2-user@webserver2 ~]$
[ec2-user@webserver2 ~]$ cat data_unencrypted.txt
Let's encrypt these file contents. Sensitive data here.
[ec2-user@webserver2 ~]$
```

I listed all the AWS KMS keys in the account using the below command.

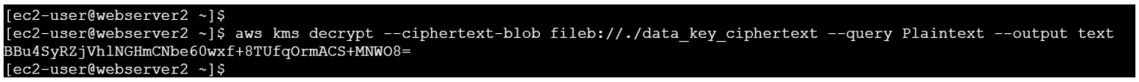
```
[ec2-user@webserver2 ~]$ aws kms list-keys
{
    "Keys": [
        {
            "KeyId": "71752706-a4ac-4d09-ae82-8797c1307811",
            "KeyArn": "arn:aws:kms:us-east-1:242015798764:key/71752706-a4ac-4d09-ae82-8797c1307811"
        },
        {
            "KeyId": "cf00f35b-e905-46c4-bc61-d4e0cf34b195",
            "KeyArn": "arn:aws:kms:us-east-1:242015798764:key/cf00f35b-e905-46c4-bc61-d4e0cf34b195"
        },
        {
            "KeyId": "d9010741-21da-44b2-bae1-3b0ab7031e05",
            "KeyArn": "arn:aws:kms:us-east-1:242015798764:key/d9010741-21da-44b2-bae1-3b0ab7031e05"
        }
    ]
}
[ec2-user@webserver2 ~]$
```

I ran the following commands to generate a data key for the MyKMSKey, saved it to a bash variable, and then echoed the data key details in JSON format:

Then to save the data key to disk. First I export the CiphertextBlob value from the result and save it to a text file in base64-encoded format,

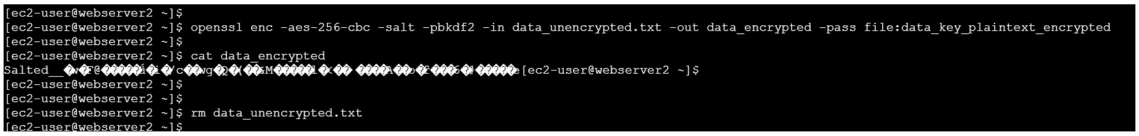
then to see that the data key is stored in encrypted (not human-readable) format:

To view it in clear text (human-readable) format:

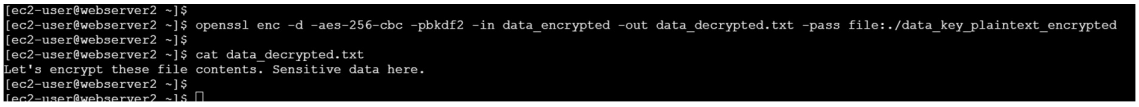


To save the result to a file in base64-encoded format

Then I will use the data key to encrypt the file that I created earlier, and delete the unencrypted version of the file.

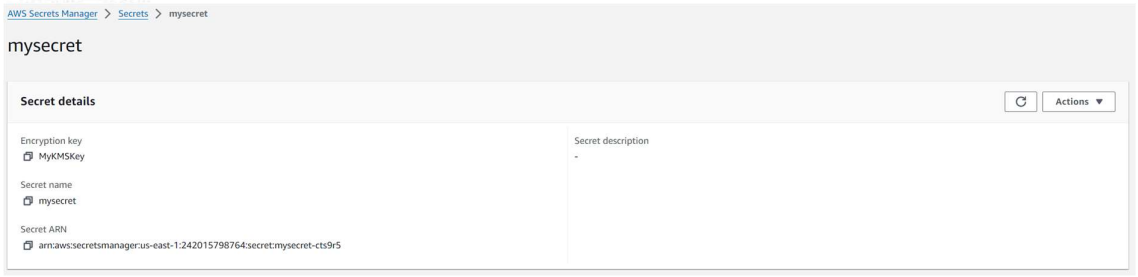


Then I will decrypt the file to prove that the data is retrievable.



Task 3.6: Use AWS KMS to encrypt a Secrets Manager secret

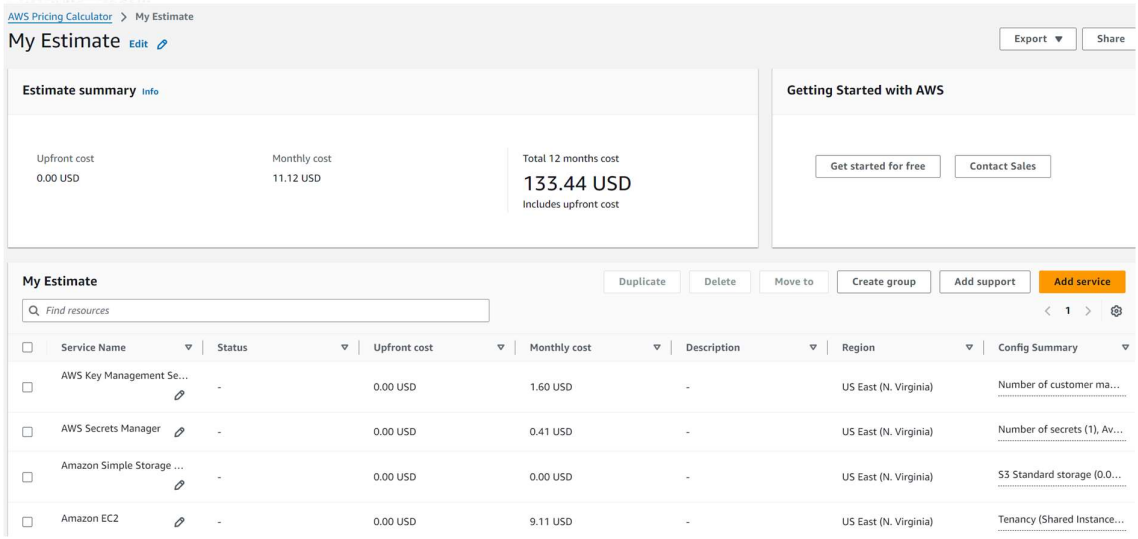
Created a new secret



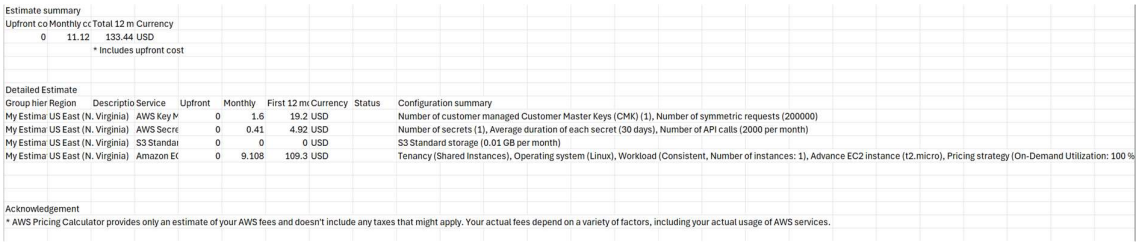
Then I used EC2 Instance Connect to connect to the WebServer2 instance, and then used the AWS CLI to retrieve the secret.

```
[ec2-user@webserver2 ~]$
[ec2-user@webserver2 ~]$ aws secretsmanager list-secrets
{
    "SecretList": [
        {
            "ARN": "arn:aws:secretsmanager:us-east-1:242015798764:secret:mysecret-cts9r5",
            "Name": "mysecret",
            "KmsKeyId": "arn:aws:kms:us-east-1:242015798764:key/cf00f35b-e905-46c4-bc61-d4e0cf34b195",
            "LastChangedDate": "2024-10-09T11:24:51.955000+00:00",
            "LastAccessedDate": "2024-10-09T00:00:00+00:00",
            "Tags": [],
            "SecretVersionsToStages": {
                "50e02c16-edd3-49cf-8e8f-246a3ed18a78": [
                    "AWSCURRENT"
                ]
            },
            "CreatedDate": "2024-10-09T11:24:51.879000+00:00"
        }
    ]
}
[ec2-user@webserver2 ~]$
[ec2-user@webserver2 ~]$ aws secretsmanager get-secret-value --secret-id mysecret
{
    "ARN": "arn:aws:secretsmanager:us-east-1:242015798764:secret:mysecret-cts9r5",
    "Name": "mysecret",
    "VersionId": "50e02c16-edd3-49cf-8e8f-246a3ed18a78",
    "SecretString": "{\"secret\":\"my secret data\"}",
    "VersionStages": [
        "AWSCURRENT"
    ],
    "CreatedDate": "2024-10-09T11:24:51.943000+00:00"
}
[ec2-user@webserver2 ~]$
```

Cost assessment for using AWS KMS



Here it is in csv format



Estimate summary

Upfront co	Monthly cc	Total 12 m	Currency
0	11.12	133.44	USD

* Includes upfront cost

Detailed Estimate

Group hier	Region	Descriptio	Service	Upfront	Monthly	First 12 mc	Currency	Status	Configuration summary
My Estima	US East (N. Virginia)		AWS Key M	0	1.6	19.2	USD		Number of customer managed Customer Master Keys (CMK) (1), Number of symmetric requests (200000)
My Estima	US East (N. Virginia)		AWS Secre	0	0.41	4.92	USD		Number of secrets (1), Average duration of each secret (30 days), Number of API calls (2000 per month)
My Estima	US East (N. Virginia)		S3 Standar	0	0	0	USD		S3 Standard storage (0.01 GB per month)
My Estima	US East (N. Virginia)		Amazon EC	0	9.108	109.3	USD		Tenancy (Shared Instances), Operating system (Linux), Workload (Consistent, Number of instances: 1), Advance EC2 instance (t2.micro), Pricing strategy (On-Demand Utilization: 100 %

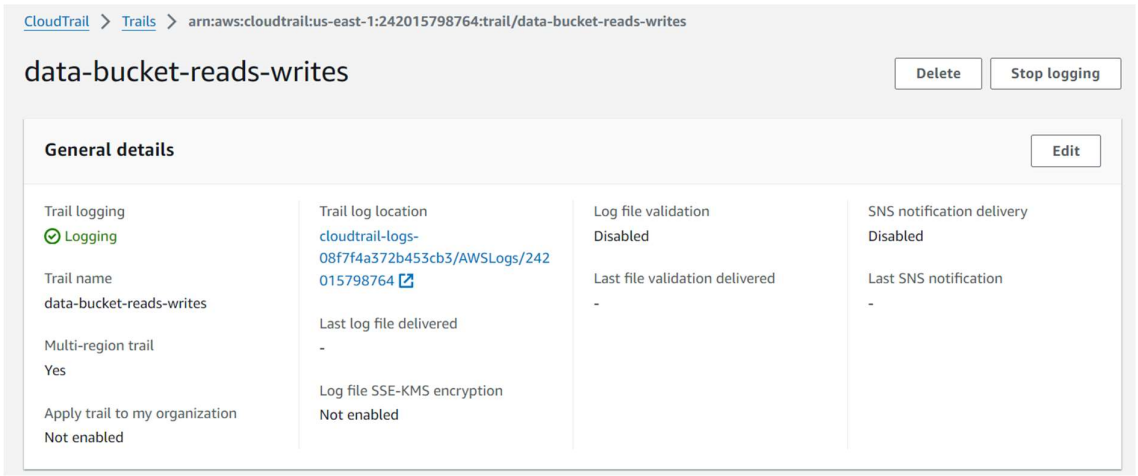
Acknowledgement

* AWS Pricing Calculator provides only an estimate of your AWS fees and doesn't include any taxes that might apply. Your actual fees depend on a variety of factors, including your actual usage of AWS services.

Phase 4: Monitoring and logging

Task 4.1: Use CloudTrail to record Amazon S3 API calls

Created a new CloudTrail trail. It will store the logs in the existing cloudtrail-logs S3 bucket



It will record both management events and data events in the trail. And for data events, it will log all S3 events.

Management events Edit

API activity
All

Exclude AWS KMS events
No
Exclude Amazon RDS Data API events
No

Data events Edit

Data events: S3
Log selector template
Log all events

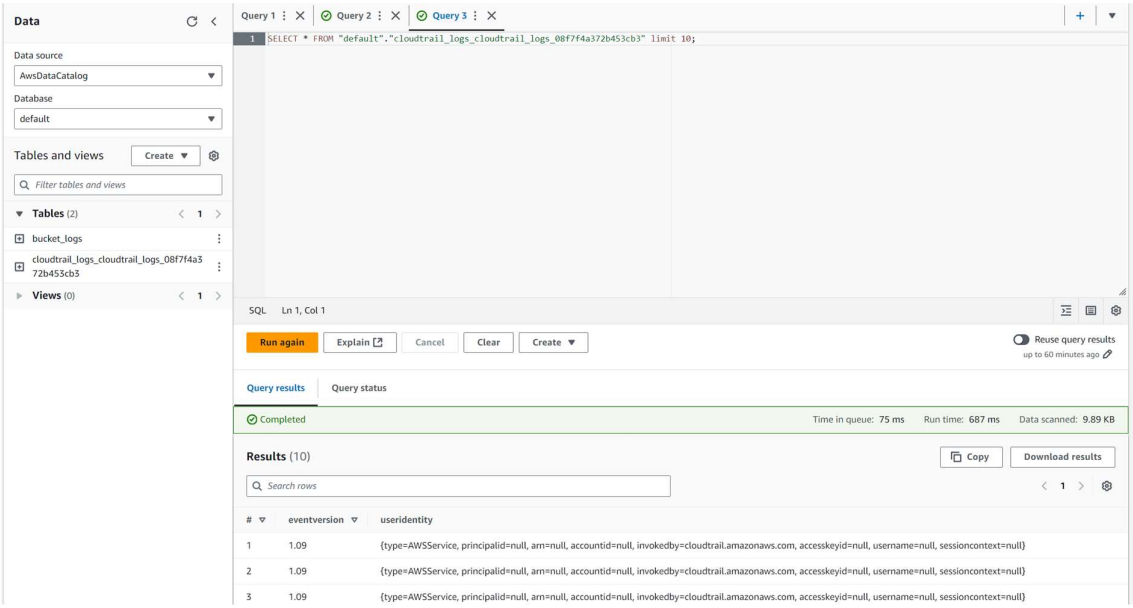
Selector name
--

All events

Created a new file locally named “customer-data.csv” and with the below contents:

Customer	First Name	Last Name	Join Date	Street Adc	City,State	Phone
1	Alejandro	Rosalez	12/12/2013	123 Main	Any Town	301-555-0158
2	Jane	Doe	10/5/2014	456 State	Anywhere	360-555-0163
3	John	Stiles	9/20/2016	1980 8th S	Nowhere,	914-555-0122
4	Li	Juan	6/29/2011	1323 22nd	Anytown,	914-555-0149

Created an Athena table that describes the format of the data in the *cloudtrail-logs* S3 bucket.



Results (10)

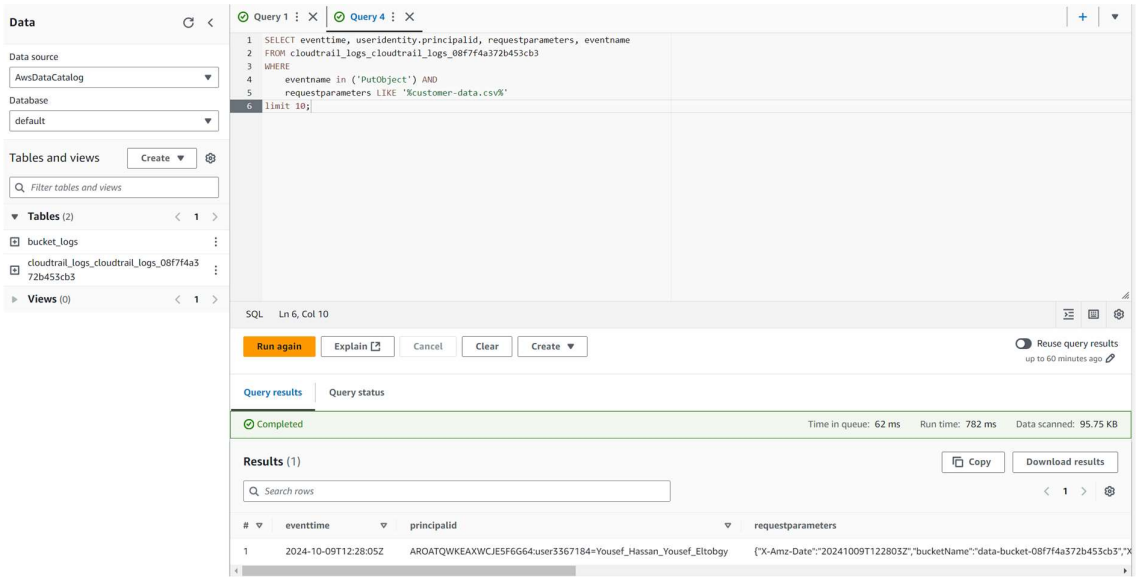
Copy | Download results

Search rows

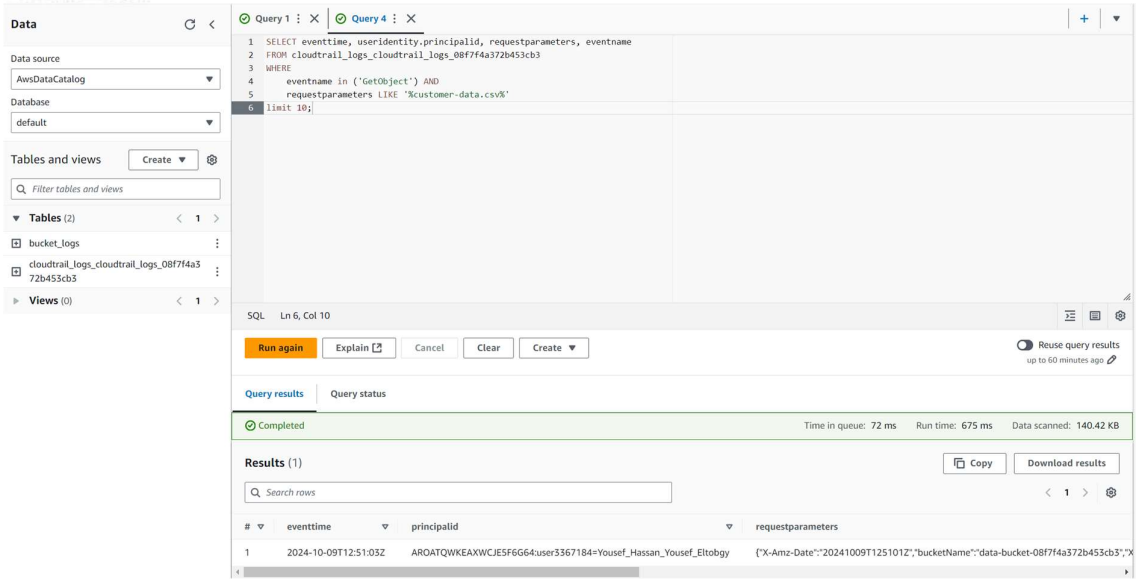
1

#	eventversion	useridentity
1	1.09	{type=AWSService, principalid=null, arn=null, accountid=null, invokedby=cloudtrail.amazonaws.com, accesskeyid=null, username=null, sessioncontext=null}
2	1.09	{type=AWSService, principalid=null, arn=null, accountid=null, invokedby=cloudtrail.amazonaws.com, accesskeyid=null, username=null, sessioncontext=null}
3	1.09	{type=AWSService, principalid=null, arn=null, accountid=null, invokedby=cloudtrail.amazonaws.com, accesskeyid=null, username=null, sessioncontext=null}
4	1.09	{type=AWSService, principalid=null, arn=null, accountid=null, invokedby=cloudtrail.amazonaws.com, accesskeyid=null, username=null, sessioncontext=null}
5	1.08	{type=AssumedRole, principalid=AROATQWKEAXWLMJPQKBR2:i-03071291af5783287, arn=arn:aws:sts::242015798764:assumed-role/WebServerRole/i-03071291af5783287, acco
6	1.08	{type=AssumedRole, principalid=AROATQWKEAXWLMJPQKBR2:i-09c84b7b7c201db65, arn=arn:aws:sts::242015798764:assumed-role/WebServerRole/i-09c84b7b7c201db65, acc
7	1.08	{type=AssumedRole, principalid=AROATQWKEAXWCJE5F6G64:user3367184=Yousef_Hassan_Yousef_Eltobgy, arn=arn:aws:sts::242015798764:assumed-role/voclabs/user3367184
8	1.08	{type=AssumedRole, principalid=AROATQWKEAXWCJE5F6G64:user3367184=Yousef_Hassan_Yousef_Eltobgy, arn=arn:aws:sts::242015798764:assumed-role/voclabs/user3367184
9	1.08	{type=AssumedRole, principalid=AROATQWKEAXWCJE5F6G64:user3367184=Yousef_Hassan_Yousef_Eltobgy, arn=arn:aws:sts::242015798764:assumed-role/voclabs/user3367184
10	1.08	{type=AssumedRole, principalid=AROATQWKEAXWCJE5F6G64:user3367184=Yousef_Hassan_Yousef_Eltobgy, arn=arn:aws:sts::242015798764:assumed-role/voclabs/user3367184

I ran an Athena query to retrieve the CloudTrail event log data for when I uploaded the customer-data.csv file to Amazon S3

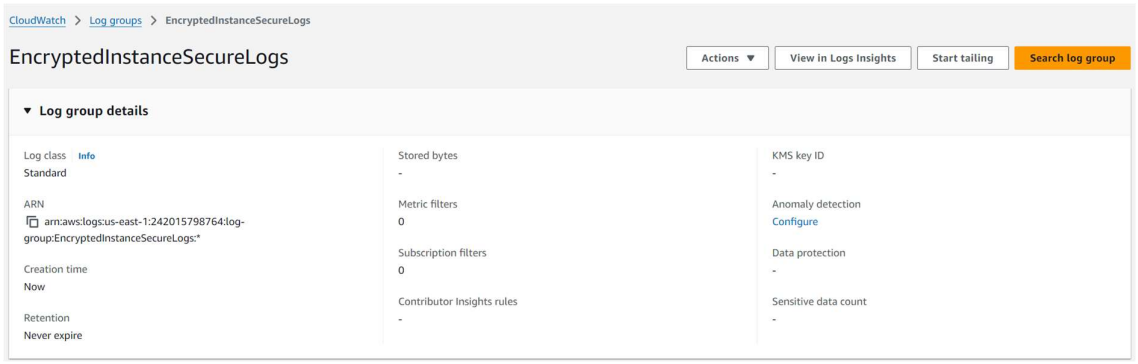


I also created a similar Athena query to retrieve the CloudTrail log information for when I opened (or downloaded) the customer-data.csv file.

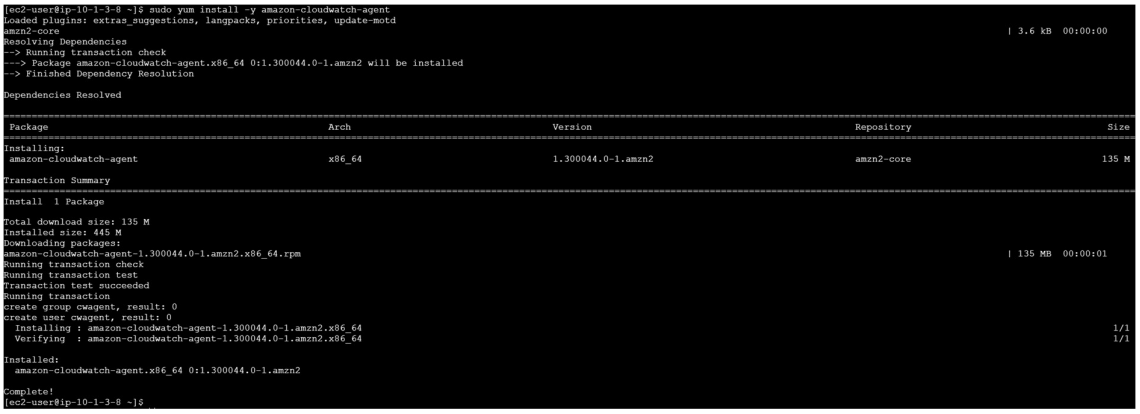


Task 4.2: Use CloudWatch Logs to monitor secure logs

I created a new CloudWatch log group.



Then I used EC2 Instance Connect to connect to EncryptedInstance, and installed the CloudWatch agent and a Linux daemon named collectd, which the CloudWatch agent will use,



```
[ec2-user@ip-10-1-3-8 ~]$ sudo amazon-linux-extras install -y collectd
Installing collectd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-collectd amzn2extra-docker amzn2extra-kernel-5.10
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                        | 3.6 kB  00:00:00
amzn2extra-collectd                                               | 3.0 kB  00:00:00
amzn2extra-docker                                                 | 2.9 kB  00:00:00
amzn2extra-kernel-5.10                                            | 3.0 kB  00:00:00
(1/9): amzn2-core/2/x86_64/updateinfo                             | 981 kB  00:00:00
(2/9): amzn2-core/2/x86_64/group_gz                               | 2.7 kB  00:00:00
(3/9): amzn2extra-docker/2/x86_64/primary_db                      | 113 kB  00:00:00
(4/9): amzn2extra-kernel-5.10/2/x86_64/updateinfo                 |  91 kB  00:00:00
(5/9): amzn2extra-collectd/2/x86_64/updateinfo                    |   76 B  00:00:00
(6/9): amzn2extra-collectd/2/x86_64/primary_db                    |  60 kB  00:00:00
(7/9): amzn2extra-docker/2/x86_64/updateinfo                      |  19 kB  00:00:00
(8/9): amzn2extra-kernel-5.10/2/x86_64/primary_db                 |  31 MB  00:00:00
(9/9): amzn2-core/2/x86_64/primary_db                             |  71 MB  00:00:01
Resolving Dependencies
--> Running transaction check
---> Package collectd.x86_64 0:5.8.1-1.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

==========================================================================================
 Package          Arch          Version                  Repository                 Size
==========================================================================================
Installing:
 collectd         x86_64        5.8.1-1.amzn2.0.2        amzn2extra-collectd       706 k
```

Then I downloaded and configured a JSON file that provides configuration details for the CloudWatch agent

```
[ec2-user@ip-10-1-3-8 ~]$ sudo wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACCAP6-91948/capstone-6-security/s3/config.json -P /opt/aws/amazon-cloudwatch-agent/bin/
--2024-10-09 13:01:32--  https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACCAP6-91948/capstone-6-security/s3/config.json
Resolving aws-tc-largeobjects.s3.us-west-2.amazonaws.com (aws-tc-largeobjects.s3.us-west-2.amazonaws.com)... 52.92.152.234, 52.218.232.177, 52.92.241.2, ...
Connecting to aws-tc-largeobjects.s3.us-west-2.amazonaws.com (aws-tc-largeobjects.s3.us-west-2.amazonaws.com)|52.92.152.234|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2278 (2.2K) [application/json]
Saving to: '/opt/aws/amazon-cloudwatch-agent/bin/config.json'

100%[==========================================================================>] 2,278      --.-K/s   in 0s

2024-10-09 13:01:32 (171 MB/s) - '/opt/aws/amazon-cloudwatch-agent/bin/config.json' saved [2278/2278]

[ec2-user@ip-10-1-3-8 ~]$
[ec2-user@ip-10-1-3-8 ~]$
```

To print out the file template so that I can see what it specifies:

```
[ec2-user@ip-10-1-3-8 ~]$ sudo cat /opt/aws/amazon-cloudwatch-agent/bin/config.json
{
   "agent": {
      "metrics_collection_interval": 60,
      "run_as_user": "root"
   },
   "logs": {
      "logs_collected": {
          "files": {
             "collect_list": [
             {
                "file_path": "/var/log/secure",
                "log_group_name": "EncryptedInstanceSecureLogs",
                "log_stream_name": "EncryptedInstanceSecureLogs-{instance_id}",
                "retention_in_days": 180
             }
             ]
          }
      }
   },
   "metrics": {
      "aggregation_dimensions": [
         [
                          "InstanceId"
                  ]
           ],
           "append_dimensions": {
                  "AutoScalingGroupName": "${aws:AutoScalingGroupName}",
                  "ImageId": "${aws:ImageId}",
                  "InstanceId": "${aws:InstanceId}",
                  "InstanceType": "${aws:InstanceType}"
           },
           "metrics_collected": {
                  "collectd": {
                          "metrics_aggregation_interval": 60
                  },
                  "disk": {
                          "measurement": [
                                  "used_percent"
                          ],
                          "metrics_collection_interval": 60,
                          "resources": [
                                  "*"
                          ],
                          "ignore_file_system_types": [
                            "sysfs", "devtmpfs"
                          ]
                  },
                  "mem": {
                          "measurement": [
                                  "mem_used_percent"
                          ],
                          "metrics_collection_interval": 60
                  },
                  "statsd": {
                          "metrics_aggregation_interval": 60,
                          "metrics_collection_interval": 10,
                          "service_address": ":8125"
                  }
           }
         }
 }[ec2-user@ip-10-1-3-8 ~]$
```

Then I started the CloudWatch agent,

```
[ec2-user@ip-10-1-3-8 ~]$ sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -s -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json
****** processing amazon-cloudwatch-agent ******
I! Trying to detect region from ec2 D! [EC2] Found active network interface I! imds retry client will retry 1 timesSuccessfully fetched the config and saved in /opt/aws/amazon-clo
watch-agent/etc/amazon-cloudwatch-agent.d/file_config.json.tmp
Start configuration validation...
2024/10/09 13:03:18 Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/file_config.json.tmp ...
2024/10/09 13:03:18 I! Valid Json input schema.
2024/10/09 13:03:18 D! ec2tagger processor required because append_dimensions is set
2024/10/09 13:03:18 Configuration validation first phase succeeded
I! Detecting run_as_user...
I! Trying to detect region from ec2
D! [EC2] Found active network interface
I! imds retry client will retry 1 times
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent -schematest -config /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.toml
Configuration validation second phase succeeded
Configuration validation succeeded
amazon-cloudwatch-agent has already been stopped
Created symlink from /etc/systemd/system/multi-user.target.wants/amazon-cloudwatch-agent.service to /etc/systemd/system/amazon-cloudwatch-agent.service.
[ec2-user@ip-10-1-3-8 ~]$
```

and confirmed that it is running:

```
[ec2-user@ip-10-1-3-8 ~]$ sudo service amazon-cloudwatch-agent status
Redirecting to /bin/systemctl status amazon-cloudwatch-agent.service
● amazon-cloudwatch-agent.service - Amazon CloudWatch Agent
   Loaded: loaded (/etc/systemd/system/amazon-cloudwatch-agent.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2024-10-09 13:03:19 UTC; 16s ago
 Main PID: 814 (amazon-cloudwat)
   CGroup: /system.slice/amazon-cloudwatch-agent.service
           └─814 /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent -config /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.toml -envconfig /opt/aws/amazon-
oudwatch-agent/etc...

Oct 09 13:03:20 ip-10-1-3-8.ec2.internal start-amazon-cloudwatch-agent[814]: 2024/10/09 13:03:20 Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwa
h-agent.json ...
Oct 09 13:03:20 ip-10-1-3-8.ec2.internal start-amazon-cloudwatch-agent[814]: /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json does not exist or cannot read. Skipp
g it.
Oct 09 13:03:20 ip-10-1-3-8.ec2.internal start-amazon-cloudwatch-agent[814]: 2024/10/09 13:03:20 Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwa
h-agent....g.json ...
Oct 09 13:03:20 ip-10-1-3-8.ec2.internal start-amazon-cloudwatch-agent[814]: 2024/10/09 13:03:20 I! Valid Json input schema.
Oct 09 13:03:20 ip-10-1-3-8.ec2.internal start-amazon-cloudwatch-agent[814]: I! Detecting run_as_user...
Oct 09 13:03:20 ip-10-1-3-8.ec2.internal start-amazon-cloudwatch-agent[814]: I! Trying to detect region from ec2
Oct 09 13:03:20 ip-10-1-3-8.ec2.internal start-amazon-cloudwatch-agent[814]: 2024/10/09 13:03:20 D! ec2tagger processor required because append_dimensions is set
Oct 09 13:03:20 ip-10-1-3-8.ec2.internal start-amazon-cloudwatch-agent[814]: 2024/10/09 13:03:20 Configuration validation first phase succeeded
Oct 09 13:03:20 ip-10-1-3-8.ec2.internal start-amazon-cloudwatch-agent[814]: /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json does not exist or cannot read. Skipp
g it.
Oct 09 13:03:20 ip-10-1-3-8.ec2.internal start-amazon-cloudwatch-agent[814]: I! Detecting run_as_user...
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-10-1-3-8 ~]$
```

To confirm that the CloudWatch agent is able to reach the CloudWatch service in your AWS account,

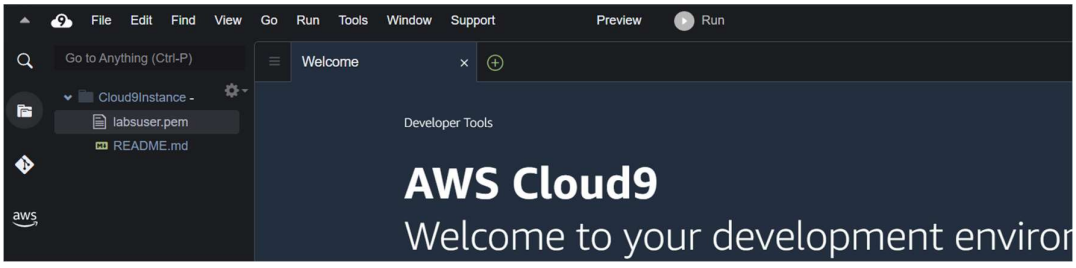
```
[ec2-user@ip-10-1-3-8 ~]$
[ec2-user@ip-10-1-3-8 ~]$ sudo cat /opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log
2024/10/09 13:03:20 I! Config has been translated into TOML /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.toml
2024/10/09 13:03:20 D! config [agent]
  collection_jitter = "0s"
  debug = false
  flush_interval = "1s"
  flush_jitter = "0s"
  hostname = ""
  interval = "60s"
  logfile = "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log"
  logtarget = "lumberjack"
  metric_batch_size = 1000
  metric_buffer_limit = 10000
```

```
2024-10-09T13:03:21Z I! First time setting retention for log group EncryptedInstanceSecureLogs, update map to avoid setting twice
2024-10-09T13:03:21Z I! [outputs.cloudwatchlogs] Configured middleware on AWS client
2024-10-09T13:03:21Z I! [logagent] piping log from EncryptedInstanceSecureLogs/EncryptedInstanceSecureLogs-i-03071291af5783287(/var/log/secure) to cloudwatchlogs with retention 180
2024-10-09T13:03:26Z W! [outputs.cloudwatchlogs] Retried 0 time, going to sleep 161.589796ms before retrying.
[ec2-user@ip-10-1-3-8 ~]$
```

To actively tail the /var/log/secure file so that you can monitor the logs that will be created in the next step, I used the following command:

```
[ec2-user@ip-10-1-3-8 ~]$
[ec2-user@ip-10-1-3-8 ~]$ sudo tail -f /var/log/secure
Oct  9 13:03:16 ip-10-1-3-8 sudo: pam_unix(sudo:session): session opened for user root by ec2-user(uid=0)
Oct  9 13:03:19 ip-10-1-3-8 sudo: pam_unix(sudo:session): session closed for user root
Oct  9 13:03:35 ip-10-1-3-8 sudo: ec2-user : TTY=pts/0 ; PWD=/home/ec2-user ; USER=root ; COMMAND=/sbin/service#040amazon-cloudwatch-agent#040status
Oct  9 13:03:35 ip-10-1-3-8 sudo: pam_unix(sudo:session): session opened for user root by ec2-user(uid=0)
Oct  9 13:03:35 ip-10-1-3-8 sudo: pam_unix(sudo:session): session closed for user root
Oct  9 13:07:20 ip-10-1-3-8 sudo: ec2-user : TTY=pts/0 ; PWD=/home/ec2-user ; USER=root ; COMMAND=/bin/cat#040/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log
Oct  9 13:07:20 ip-10-1-3-8 sudo: pam_unix(sudo:session): session opened for user root by ec2-user(uid=0)
Oct  9 13:07:20 ip-10-1-3-8 sudo: pam_unix(sudo:session): session closed for user root
Oct  9 13:08:52 ip-10-1-3-8 sudo: ec2-user : TTY=pts/0 ; PWD=/home/ec2-user ; USER=root ; COMMAND=/bin/tail#040-f#040/var/log/secure
Oct  9 13:08:52 ip-10-1-3-8 sudo: pam_unix(sudo:session): session opened for user root by ec2-user(uid=0)
^C
[ec2-user@ip-10-1-3-8 ~]$
```

Then I uploaded the key to the cloud9 instance



Then I became able to SSH to the EncryptedInstance instance from the cloud9 instance.

```
voclabs:~/environment $ chmod 400 labsuser.pem
voclabs:~/environment $
voclabs:~/environment $ ssh -i labsuser.pem ec2-user@3.236.163.130
The authenticity of host '3.236.163.130 (3.236.163.130)' can't be established.
ECDSA key fingerprint is SHA256:mkb8xuvMyb3LRtpmXs203EBDAYxpFRCWHr6IykX0q0Q.
ECDSA key fingerprint is MD5:c7:39:ac:8b:b0:49:8a:b7:c7:5e:4a:05:74:3e:57:57.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '3.236.163.130' (ECDSA) to the list of known hosts.
Last login: Wed Oct  9 12:59:08 2024 from ec2-18-206-107-27.compute-1.amazonaws.com
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-1-3-8 ~]$
```

But if I tried with user “ubuntu”, it will result in permission denied. Because it doesn't exist on this Amazon Linux instance.

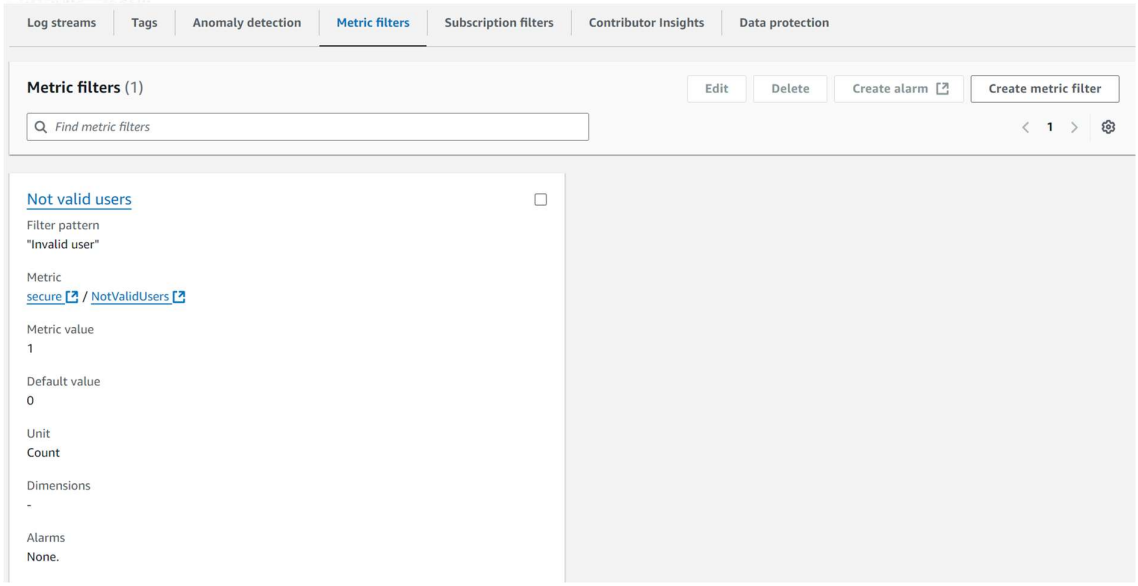
```
voclabs:~/environment $
voclabs:~/environment $ ssh -i labsuser.pem ubuntu@3.236.163.130
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
voclabs:~/environment $
```

And here are the logs that recorded the successful connection from ec2-user, and the failed request from user ubuntu

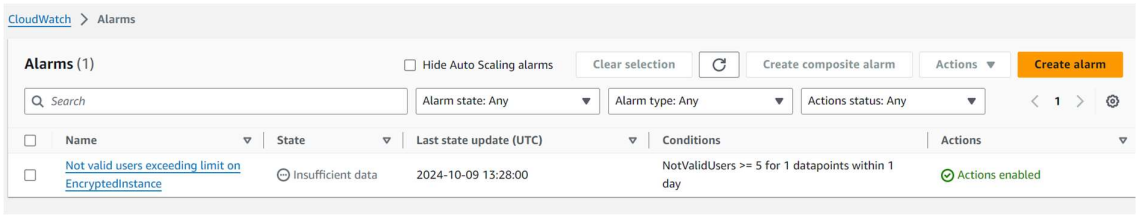
▼	2024-10-09T13:16:57.653Z	Oct 9 13:12:14 ip-10-1-3-8 sshd[919]: Accepted publickey for ec2-user f…	EncryptedInstanceSecureLogs-i-03071291af5783287
	Oct 9 13:12:14 ip-10-1-3-8 sshd[919]: Accepted publickey for ec2-user from 52.1.136.202 port 46392 ssh2: RSA SHA256:RJP8SO+vEvSAH6ZebbgkpienO2ZVFaRJPd2PtKSUfMw		
▶	2024-10-09T13:16:57.653Z	Oct 9 13:12:14 ip-10-1-3-8 sshd[919]: pam_unix(sshd:session): session o…	EncryptedInstanceSecureLogs-i-03071291af5783287
▶	2024-10-09T13:16:57.653Z	Oct 9 13:16:57 ip-10-1-3-8 sshd[1116]: Received disconnect from 52.1.13…	EncryptedInstanceSecureLogs-i-03071291af5783287
▶	2024-10-09T13:16:57.653Z	Oct 9 13:16:57 ip-10-1-3-8 sshd[1116]: Disconnected from 52.1.136.202 p…	EncryptedInstanceSecureLogs-i-03071291af5783287
▶	2024-10-09T13:17:02.224Z	Oct 9 13:16:57 ip-10-1-3-8 sshd[919]: pam_unix(sshd:session): session c…	EncryptedInstanceSecureLogs-i-03071291af5783287
▼	2024-10-09T13:17:19.959Z	Oct 9 13:17:19 ip-10-1-3-8 sshd[1147]: Invalid user ubuntu from 52.1.13…	EncryptedInstanceSecureLogs-i-03071291af5783287
	Oct 9 13:17:19 ip-10-1-3-8 sshd[1147]: Invalid user ubuntu from 52.1.136.202 port 51576		
▼	2024-10-09T13:17:19.959Z	Oct 9 13:17:19 ip-10-1-3-8 sshd[1147]: input_userauth_request: invalid …	EncryptedInstanceSecureLogs-i-03071291af5783287
	Oct 9 13:17:19 ip-10-1-3-8 sshd[1147]: input_userauth_request: invalid user ubuntu [preauth]		
▼	2024-10-09T13:17:24.224Z	Oct 9 13:17:19 ip-10-1-3-8 sshd[1147]: Connection closed by 52.1.136.20…	EncryptedInstanceSecureLogs-i-03071291af5783287
	Oct 9 13:17:19 ip-10-1-3-8 sshd[1147]: Connection closed by 52.1.136.202 port 51576 [preauth]		

Task 4.3: Create a CloudWatch alarm to send notifications for security incidents

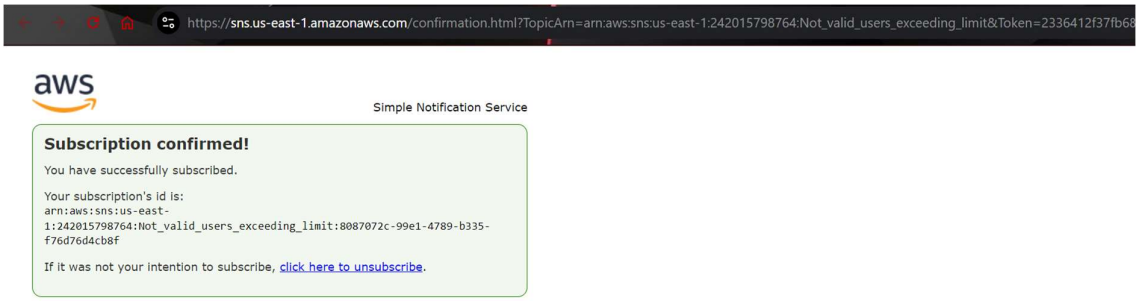
Created a new metric filter in the EncryptedInstanceSecureLogs CloudWatch log group.



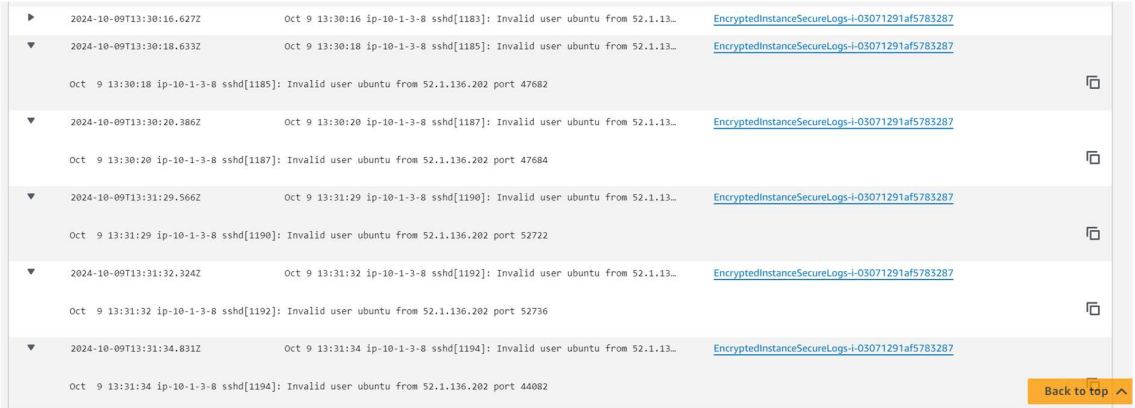
Created a CloudWatch alarm from the metric filter that I just created. It will send a notification for Not valid access attempts over SSH to the EncryptedInstance server are greater than or equal 5 in the last 24 hours.

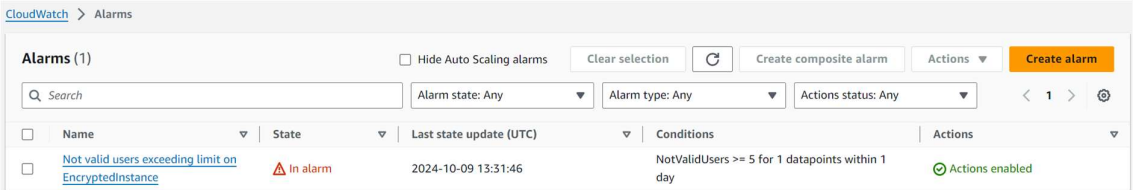


And confirmed the subscription.



I made five invalid SSH access attempts to connect to the EncryptedInstance public IP address over SSH





```
voclabs:~/environment $
voclabs:~/environment $ ssh -i labsuser.pem ubuntu@3.236.163.130
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
voclabs:~/environment $
voclabs:~/environment $
voclabs:~/environment $
voclabs:~/environment $ ssh -i labsuser.pem ubuntu@3.236.163.130
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
voclabs:~/environment $
voclabs:~/environment $
voclabs:~/environment $
voclabs:~/environment $ ssh -i labsuser.pem ubuntu@3.236.163.130
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
voclabs:~/environment $
voclabs:~/environment $
voclabs:~/environment $
voclabs:~/environment $ ssh -i labsuser.pem ubuntu@3.236.163.130
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
voclabs:~/environment $
voclabs:~/environment $
voclabs:~/environment $ ssh -i labsuser.pem ubuntu@3.236.163.130
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
voclabs:~/environment $
```

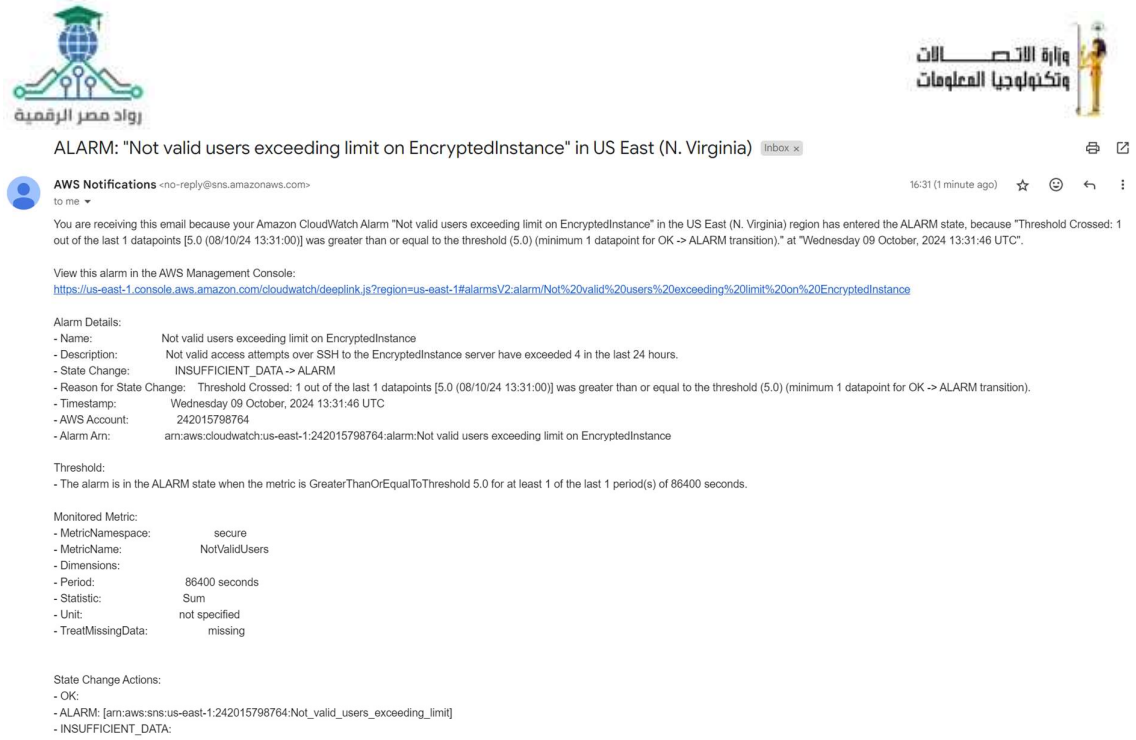
There are logs in the log group that detected the invalid user trying to log in

```
▶  2024-10-09T13:30:16.627Z    Oct 9 13:30:16 ip-10-1-3-8 sshd[1183]: Invalid user ubuntu from 52.1.13…    EncryptedInstanceSecureLogs-i-03071291af5783287
▼  2024-10-09T13:30:18.633Z    Oct 9 13:30:18 ip-10-1-3-8 sshd[1185]: Invalid user ubuntu from 52.1.13…    EncryptedInstanceSecureLogs-i-03071291af5783287

   Oct  9 13:30:18 ip-10-1-3-8 sshd[1185]: Invalid user ubuntu from 52.1.136.202 port 47682

▼  2024-10-09T13:30:20.386Z    Oct 9 13:30:20 ip-10-1-3-8 sshd[1187]: Invalid user ubuntu from 52.1.13…    EncryptedInstanceSecureLogs-i-03071291af5783287

   Oct  9 13:30:20 ip-10-1-3-8 sshd[1187]: Invalid user ubuntu from 52.1.136.202 port 47684

▼  2024-10-09T13:31:29.566Z    Oct 9 13:31:29 ip-10-1-3-8 sshd[1190]: Invalid user ubuntu from 52.1.13…    EncryptedInstanceSecureLogs-i-03071291af5783287

   Oct  9 13:31:29 ip-10-1-3-8 sshd[1190]: Invalid user ubuntu from 52.1.136.202 port 52722

▼  2024-10-09T13:31:32.324Z    Oct 9 13:31:32 ip-10-1-3-8 sshd[1192]: Invalid user ubuntu from 52.1.13…    EncryptedInstanceSecureLogs-i-03071291af5783287

   Oct  9 13:31:32 ip-10-1-3-8 sshd[1192]: Invalid user ubuntu from 52.1.136.202 port 52736

▼  2024-10-09T13:31:34.831Z    Oct 9 13:31:34 ip-10-1-3-8 sshd[1194]: Invalid user ubuntu from 52.1.13…    EncryptedInstanceSecureLogs-i-03071291af5783287

   Oct  9 13:31:34 ip-10-1-3-8 sshd[1194]: Invalid user ubuntu from 52.1.136.202 port 44082
```

Also the CloudWatch alarm became in “in alarm” state

☐	Name	State	Last state update (UTC)	Conditions	Actions
☐	Not valid users exceeding limit on EncryptedInstance	⚠ In alarm	2024-10-09 13:31:46	NotValidUsers >= 5 for 1 datapoints within 1 day	✓ Actions enabled

Plus I received a notification on the mail.



ALARM: "Not valid users exceeding limit on EncryptedInstance" in US East (N. Virginia) Inbox ×

AWS Notifications <no-reply@sns.amazonaws.com> 16:31 (1 minute ago)
to me

You are receiving this email because your Amazon CloudWatch Alarm "Not valid users exceeding limit on EncryptedInstance" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [5.0 (08/10/24 13:31:00)] was greater than or equal to the threshold (5.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Wednesday 09 October, 2024 13:31:46 UTC".

View this alarm in the AWS Management Console:
https://us-east-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-east-1#alarmsV2:alarm/Not%20valid%20users%20exceeding%20limit%20on%20EncryptedInstance

Alarm Details:
- Name: Not valid users exceeding limit on EncryptedInstance
- Description: Not valid access attempts over SSH to the EncryptedInstance server have exceeded 4 in the last 24 hours.
- State Change: INSUFFICIENT_DATA -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [5.0 (08/10/24 13:31:00)] was greater than or equal to the threshold (5.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Wednesday 09 October, 2024 13:31:46 UTC
- AWS Account: 242015798764
- Alarm Arn: arn:aws:cloudwatch:us-east-1:242015798764:alarm:Not valid users exceeding limit on EncryptedInstance

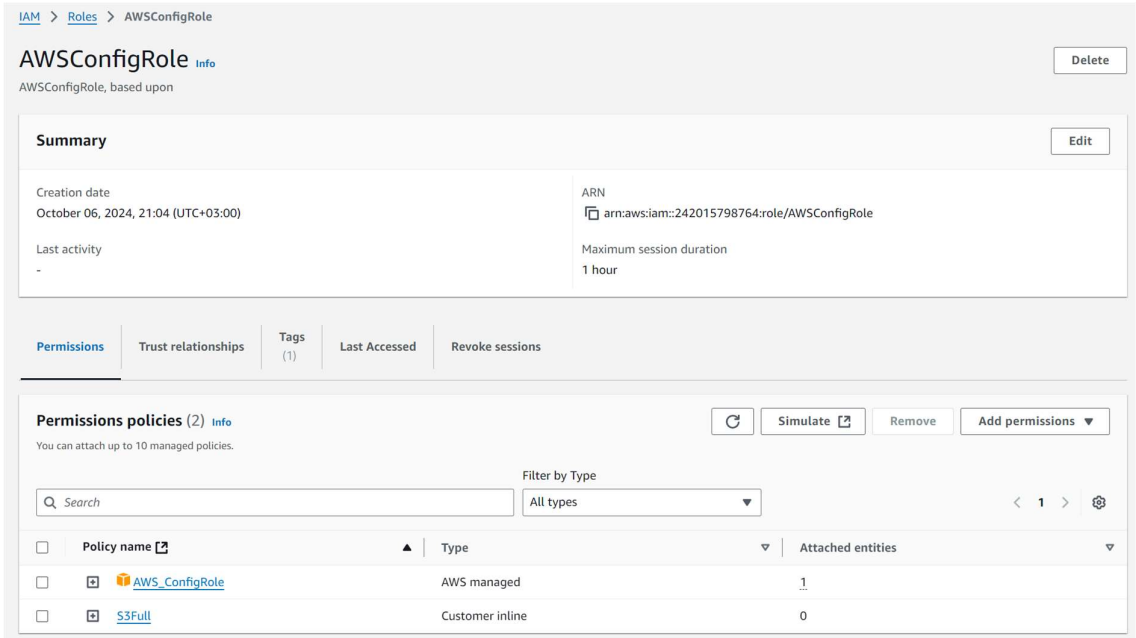
Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanOrEqualToThreshold 5.0 for at least 1 of the last 1 period(s) of 86400 seconds.

Monitored Metric:
- MetricNamespace: secure
- MetricName: NotValidUsers
- Dimensions:
- Period: 86400 seconds
- Statistic: Sum
- Unit: not specified
- TreatMissingData: missing

State Change Actions:
- OK:
- ALARM: [arn:aws:sns:us-east-1:242015798764:Not_valid_users_exceeding_limit]
- INSUFFICIENT_DATA:

Task 4.4: Configure AWS Config to assess security settings and remediate the configuration of AWS resources

I have two roles the first role grants permissions that you will need to set up AWS Config, and the second role is the role that AWS Systems Manager will use when it performs remediation actions with an AWS Config managed rule that you will use.



IAM > Roles > AWSConfigRole

AWSConfigRole Info
AWSConfigRole, based upon

Delete

Summary

Edit

Creation date
October 06, 2024, 21:04 (UTC+03:00)

ARN
arn:aws:iam::242015798764:role/AWSConfigRole

Last activity
-

Maximum session duration
1 hour

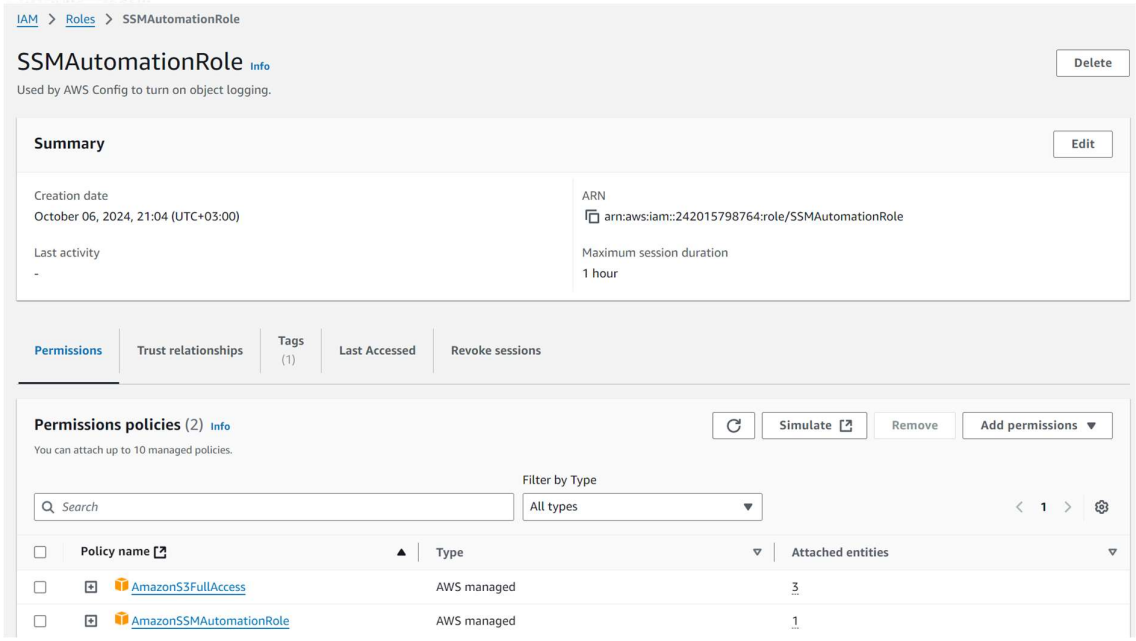
Permissions | Trust relationships | Tags (1) | Last Accessed | Revoke sessions

Permissions policies (2) Info
You can attach up to 10 managed policies.

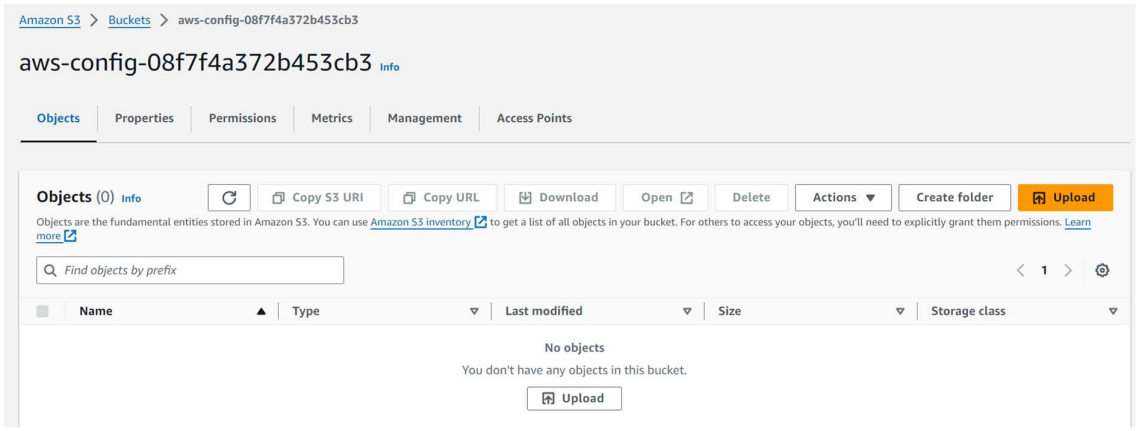
Simulate | Remove | Add permissions

Filter by Type
Search | All types

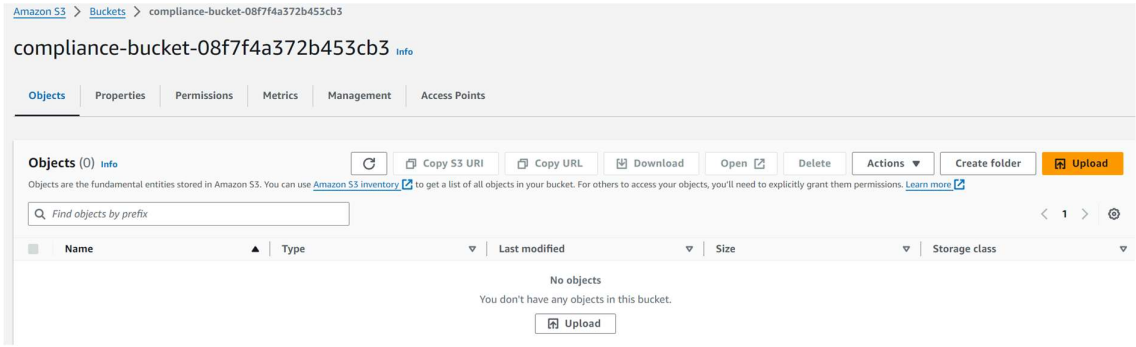
Policy name	Type	Attached entities
AWS_ConfigRole	AWS managed	1
S3Full	Customer inline	0



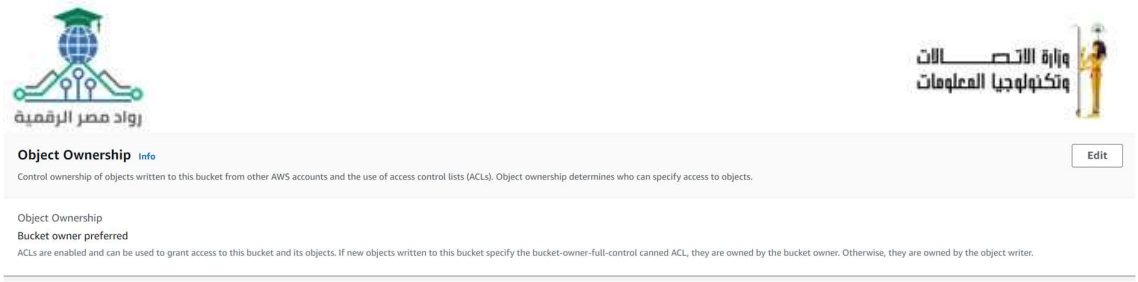
I also have the below bucket that AWS Config will use this bucket for logging purposes.



I created the below bucket

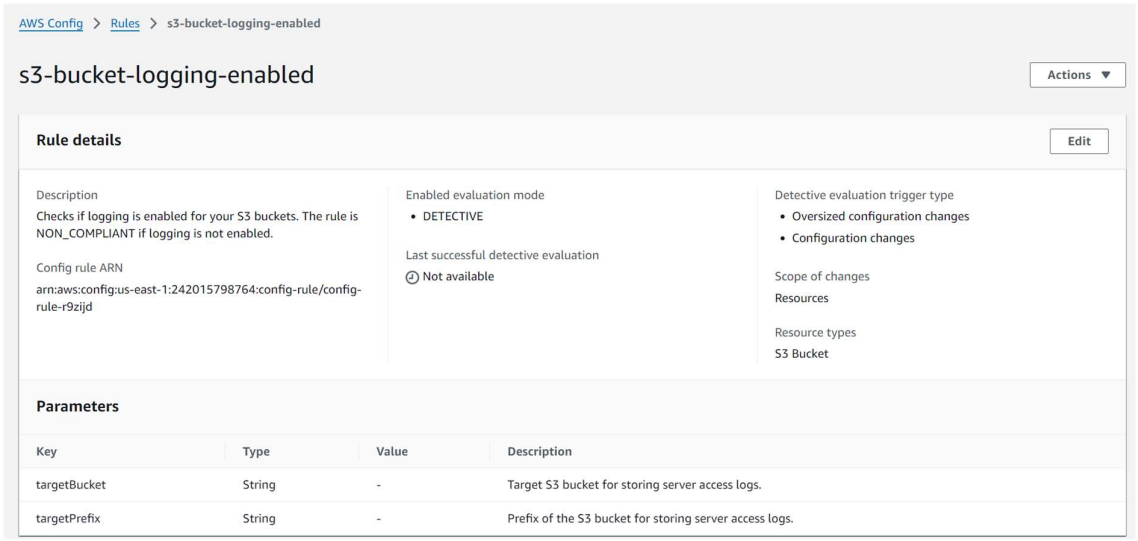


Then I edited the object ownership in the “s3-objects-access-log” bucket. I enabled ACLs.

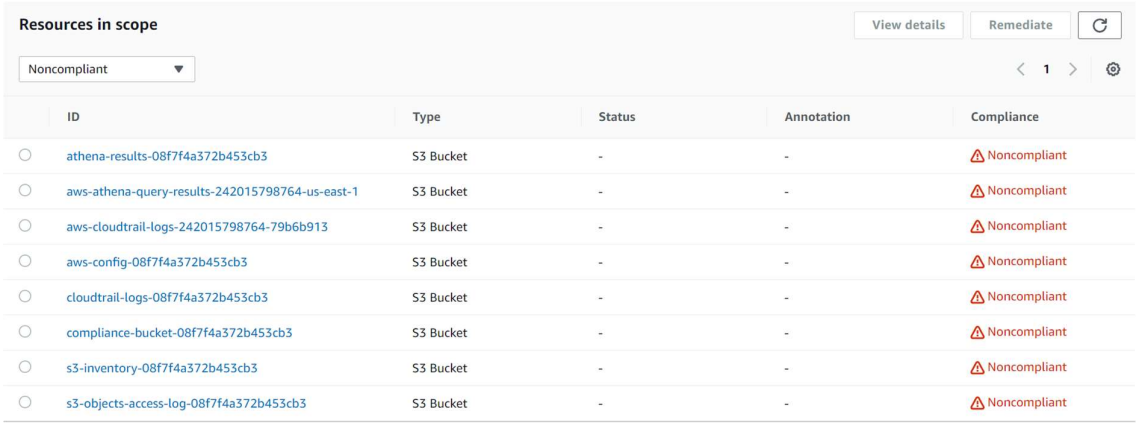


Then I set up AWS Config to record all current and future resources that are supported in the us-east-1 Region. For AWS Config role, I used the existing AWSConfigRole role. For delivery method, I chose the existing aws-config bucket. And I didn't select any managed rules.

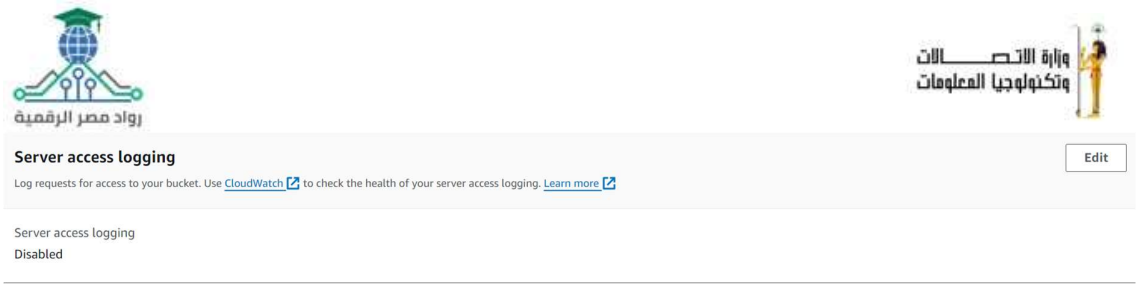
Then I created AWS Config Rule:



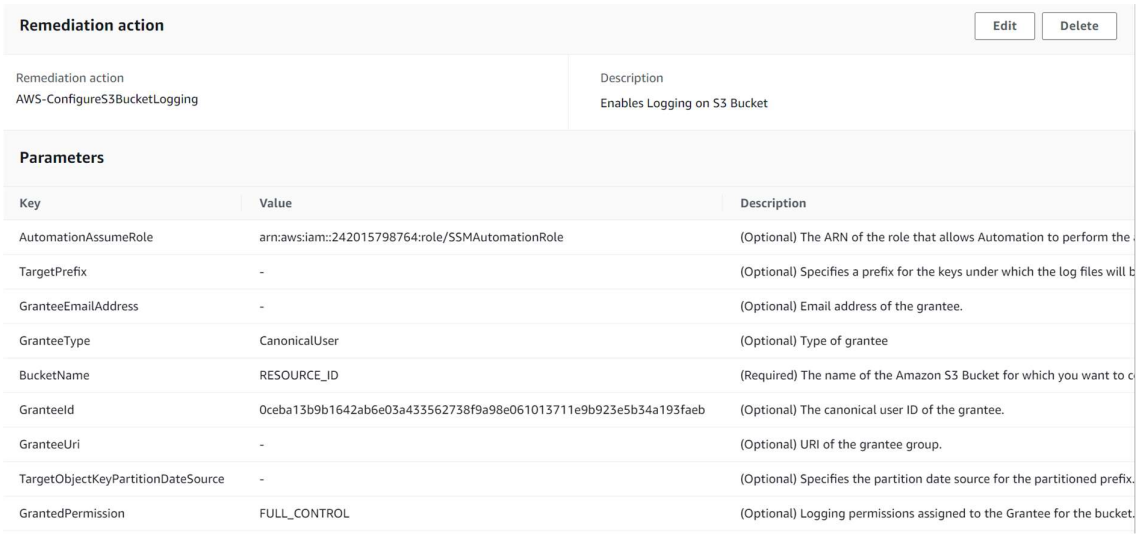
Confirmed that the s3-bucket-logging-enabled rule that I defined is finding resources that are in scope. I also confirmed that the compliance-bucket is flagged as noncompliant.



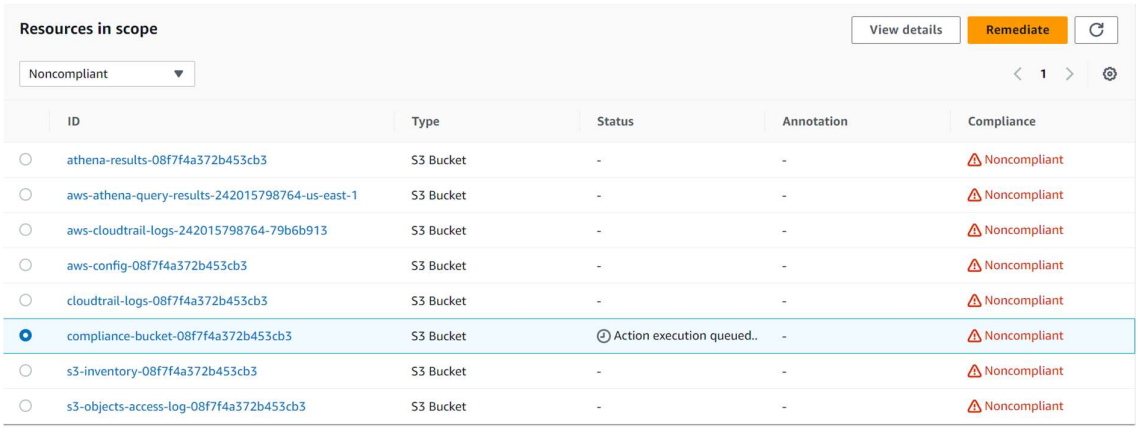
The compliance-bucket is flagged as Noncompliant, that is because the "server access logging" is disabled.



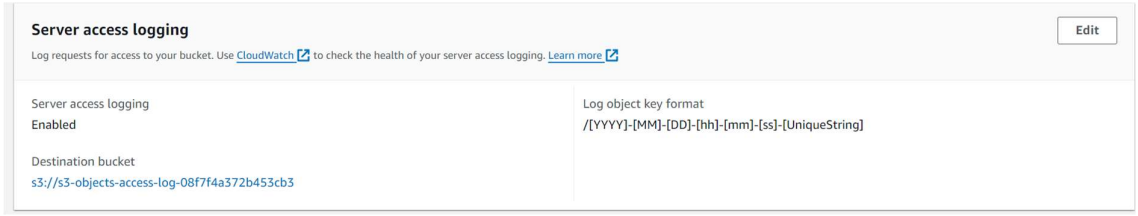
Then I configured manual remediation settings for the s3-bucket-logging-enable rule.



Then to test this, I select the “compliance bucket” bucket and then chose “remediate” button. That is to invoke the AWS Config remediation action so that object logging is enabled on the “compliance-bucket”.



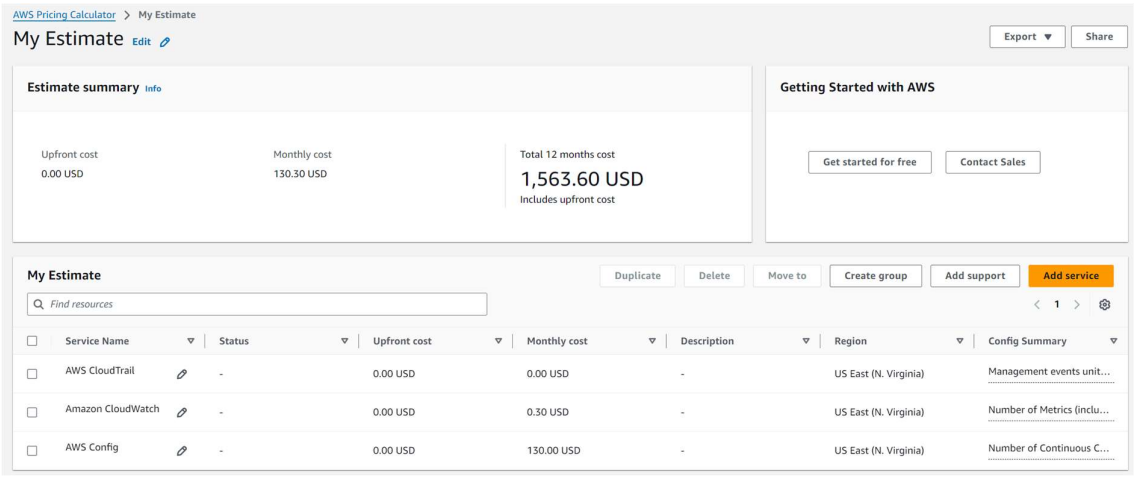
Then I confirmed that the server access logging is now enabled on the compliance-bucket. It was enabled by the remediation rule that I just ran.



Cost assessment for monitoring and logging

does the use of CloudTrail, CloudWatch, and AWS Config security features result in additional cost in the account?

Here is the answer.



AWS Pricing Calculator > My Estimate

My Estimate Edit

Export Share

Estimate summary Info

Upfront cost	Monthly cost	Total 12 months cost
0.00 USD	130.30 USD	1,563.60 USD
		Includes upfront cost

Getting Started with AWS

Get started for free | Contact Sales

My Estimate

Duplicate | Delete | Move to | Create group | Add support | Add service

Find resources

Service Name	Status	Upfront cost	Monthly cost	Description	Region	Config Summary
AWS CloudTrail	-	0.00 USD	0.00 USD	-	US East (N. Virginia)	Management events unit…
Amazon CloudWatch	-	0.00 USD	0.30 USD	-	US East (N. Virginia)	Number of Metrics (inclu…
AWS Config	-	0.00 USD	130.00 USD	-	US East (N. Virginia)	Number of Continuous C…

Here it is in csv format:

Estimate summary

Upfront cost	Monthly cc	Total 12 m	Currency
0	130.3	1563.6	USD

* Includes upfront cost

Detailed Estimate

Group hierarc	Region	Descriptio	Service	Upfront	Monthly	First 12 mc	Currency	Status	Configuration summary
My Estimate	US East (N. Virginia)		AWS CloudTrail	0	0	0	USD		Management events units (millions), Write management trails (1), Read management trails (1), Data events units (millions), S3 trails (1), Lambda trails (1), Insight events units (
My Estimate	US East (N. Virginia)		Amazon CloudW	0	0.3	3.6	USD		Number of Metrics (includes detailed and custom metrics) (1)
My Estimate	US East (N. Virginia)		AWS Config	0	130.003	1560.04	USD		Number of Continuous Configuration items recorded (1), Number of Periodic Configuration items recorded (10000), Number of Config rule evaluations (10000)

Acknowledgement

* AWS Pricing Calculator provides only an estimate of your AWS fees and doesn't include any taxes that might apply. Your actual fees depend on a variety of factors, including your actual usage of AWS services.

The Badge

This badge was issued to Yousef Hassan Yousef Eltobgy on October 09, 2024
View celebrations

Share

AWS Academy Graduate - AWS Academy Cloud Security Builder

Issued by Amazon Web Services Training and Certification

Earners of this badge have completed AWS Academy Lab Project - Cloud Security Builder.

Learn more

Skills

Amazon KMS | AWS Cloudwatch | AWS KMS | Data Security | Elastic Block Storage | Encryption | Network Security