



Securing and Monitoring Resources with AWS

Presented by/

Mennatallah Abdelkareem Habashi Saleh St-ID:21046685

Supervised by/

DEPI

Eng/Merihan Adel

Project overview and objectives:

In this project, I'm challenged to use familiar AWS services, to create resources in AWS and to implement security on them. I have used different AWS services and features to build a variety of solutions.

These specific sections of the project are meant to challenge me to practice skills that I have acquired throughout my learning experiences prior to this project.

By the end of this project, I should be able to do the following:

- Secure access to objects in an Amazon Simple Storage Service (Amazon S3) bucket.
- Secure network access to my virtual network.
- Encrypt data at rest by using AWS Key Management Service (AWS KMS) on an Amazon Elastic Block Store (Amazon EBS) volume.
- Manage encryption keys by using AWS KMS.
- Create a monitoring and incident response system by using Amazon CloudWatch and AWS Config.

Phase 1: Securing data in Amazon S3:

In this phase, I begin implementing security settings in the AWS account. I have been asked to secure customer PII data that is stored in Amazon S3. The leadership team of Any Company Financial has heard about recent data breaches at other companies and wants to protect customer data from unauthorized access.

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

The screenshot shows the AWS S3 console with the URL <https://us-east-1.console.aws.amazon.com/s3/buckets/data-bucket0e1570123fa91ff36?region=us-east-1&bucketType=general&tab=permissions>. The browser title bar also displays this URL. The top navigation bar includes links for IAM, VPC, EC2, CloudWatch, S3, Cloud9, Secrets Manager, and Systems Manager. A dropdown menu shows 'N. Virginia' and a user profile. The main content area shows a green success message: 'Successfully edited bucket policy.' Below this, the bucket name 'data-bucket0e1570123fa91ff36' is displayed with an 'Info' link. The 'Permissions' tab is selected in the navigation bar. Under 'Permissions overview', there is a section for 'Access finding' which says 'Access findings are provided by IAM external access analyzers. Learn more about How IAM analyzer findings work.' and a link to 'View analyzer for us-east-1'. The 'Block public access (bucket settings)' section is shown with an 'Edit' button. It contains a note about public access being granted through various methods like ACLs, bucket policies, and access point policies. It also mentions AWS recommends turning on 'Block all public access' before applying other settings. The 'Block all public access' setting is currently set to 'On'. The bottom of the page includes standard AWS footer links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

-*paulo* user's access to Amazon S3 but when Verify *Mary* user's access to Amazon S3, when she accesses any of the buckets, she sees an "Insufficient permissions to list objects" error , as the following:

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with various navigation options like Buckets, Storage Lens, and Feature spotlight. The main area shows a bucket named 's3-objects-access-log-0e1570123fa91ff36'. The 'Objects' tab is active. A prominent red error box is displayed, indicating insufficient permissions to list objects. It includes a link to learn more about identity and access management.

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

In this task, I'm challenged to enable versioning and object-level logging on the *data-bucket*. With versioning enabled, I can track all changes to objects that are stored in the bucket and revert any object to a previous version if needed. Object-level logging creates a detailed audit trail of the objects that are stored in a bucket, which helps me to detect security incidents quickly.

-Enable versioning on a bucket:

The screenshot shows the AWS S3 console with the 'Properties' tab selected for a bucket named 'data-bucket0e1570123fa91ff36'. A green success message box at the top indicates that 'Bucket Versioning' was successfully edited. Below, the 'Bucket overview' section displays basic information: AWS Region (US East (N. Virginia) us-east-1), Amazon Resource Name (ARN) (arn:aws:s3:::data-bucket0e1570123fa91ff36), and Creation date (September 24, 2024, 19:13:54 (UTC+03:00)). The 'Bucket Versioning' section shows that versioning is currently 'Enabled'.

-Enable object-level logging on a bucket:

The screenshot shows the AWS S3 Bucket Properties page for a bucket named "data-bucket0e1570123fa91ff36". The "Properties" tab is selected. A green success message at the top states "Successfully edited server access logging.". The "Bucket overview" section displays the AWS Region as "US East (N. Virginia) us-east-1", the Amazon Resource Name (ARN) as "arn:aws:s3:::data-bucket0e1570123fa91ff36", and the Creation date as "September 24, 2024, 19:13:54 (UTC+03:00)". The "Bucket Versioning" section shows "Bucket Versioning Enabled". The status bar at the bottom indicates "CloudShell Feedback".

Task 1.3: Implement the S3 Inventory feature on a bucket

In this task, I'm challenged to enable the S3 Inventory feature to monitor changes to objects that are stored in an S3 bucket. S3 Inventory provides a scheduled report of the metadata and object-level changes to my S3 objects and buckets. By using the feature, I can track changes to the stored objects and detect potential security incidents.

The screenshot shows the AWS S3 Management page for the same bucket. It displays two success messages: "Inventory Inventory successfully created" and "Bucket policy successfully created". The "Management > Inventory configurations" section shows one inventory configuration named "Inventory" with the following details:

Name	Status	Scope	Destination	Frequency	Last export	Format
Inventory	Enabled	Entire bucket	s3://s3-inventory-0e1...	Daily	-	Apache Parquet

The status bar at the bottom indicates "CloudShell Feedback".

Task 1.4: Confirm that versioning works as intended

In this task, I'm challenged to access the AWS account as the *paulo* user and upload an object to the *data-bucket*. Then, you are challenged to confirm that versioning is enabled on the object. You are also challenged to test access as the *mary* user.

The screenshot shows the Amazon S3 console interface. The left sidebar has sections for Buckets, Storage Lens, and Feature spotlight. The main area displays a list of objects in the 'data-bucket0e1570123fa91ff36' bucket. The objects listed are:

Name	Type	Version ID	Last modified	Size	Storage class
customers.csv.xlsx	xlsx	IHkguD_AgM A89pkdlShql 9N7yCMwttN 2	September 28, 2024, 12:51:04 (UTC+03:00)	8.4 KB	Standard
customers.csv.xlsx	xlsx	yFSW5_5ZnLJ W9dc1tYeDS V.iRBgm4dPP	September 27, 2024, 15:37:27 (UTC+03:00)	8.4 KB	Standard
myfile1.txt	txt	null	September 26, 2024, 19:40:36 (UTC+03:00)	11.0 B	Standard

- Log in as the *mary* user and take action in the Amazon S3 console to generate log events After test versioning , notice that the *mary* user cannot access this bucket or its contents.

The screenshot shows the Amazon S3 console interface for the same bucket as the previous screenshot, but from the perspective of the *mary* user. The error message 'Insufficient permissions to list objects' is prominently displayed in a red box. The message states: 'After you or your AWS administrator has updated your permissions to allow the s3>ListBucket action, refresh the page.' Below the message is a link 'Learn more about Identity and access management in Amazon S3'.

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

In this task, I'm challenged to confirm the S3 object-level logging I enabled earlier is successfully writing log data to S3. I will also use Athena to query these logs.

- Create s3 bucket to observe the objects that are stored in the *s3-objects-access-log* bucket:

The screenshot shows the AWS S3 console. On the left, the navigation pane includes options like Buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings for this account, Storage Lens (Dashboards, Storage Lens groups, AWS Organizations settings), and Feature spotlight. The main content area shows a green success message: "Successfully edited bucket policy." Below it, the path is Amazon S3 > Buckets > s3-objects-access-log-0e1570123fa91ff36. The "Permissions" tab is selected. Under "Permissions overview", there's a section for "Access finding" which links to "How IAM analyzer findings work" and "View analyzer for us-east-1". In the "Block public access (bucket settings)" section, the "Edit" button is visible. The "Block all public access" switch is turned "On".

- Create an Athena table from the access logs:

The screenshot shows the AWS S3 console with a green success message: "Successfully created bucket 'athena-results-9876nhgg'. To upload files and folders, or to configure additional bucket settings, choose View details." Below this, the "Account snapshot - updated every 24 hours" section is shown. The "General purpose buckets" tab is selected, displaying a list of six buckets: athena-results-9876nhgg, aws-config-0e1570123fa91ff36, cloudtrail-logs-0e1570123fa91ff36, data-bucket0e1570123fa91ff36, and s3-inventory-0e1570123fa91ff36. Each bucket entry includes a "View details" link, "Copy ARN" button, "Empty" button, and a "Create bucket" button. The "Find buckets by name" search bar is present. At the bottom, there are CloudShell, Feedback, and footer links for 2024, Privacy, Terms, and Cookie preferences.

-Query the table to discover access log details:

The screenshot shows the AWS S3 console interface. The left sidebar is titled "Amazon S3" and includes sections for Buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings, Storage Lens (Dashboards, Storage Lens groups, AWS Organizations settings), and a Feature spotlight. The main content area shows a list of objects in the bucket "s3-objects-access-log-0e1570123fa91ff36". There are 90 objects listed, all of which are type "Folder". The first few objects are named "data-bucket2024-09-26-19-14-30-A6628913A97BF1C5", "data-bucket2024-09-26-19-22-15-E2DDDB0C479BC90D", and "data-bucket2024-09-26-19-25-41-0C72C4E2E0028A89". The table includes columns for Name, Type, Last modified, Size, and Storage class.

-Cost assessment to secure Amazon S3:

The screenshot shows the AWS Pricing Calculator interface. The top navigation bar includes links for Feedback, Language: English, Contact Sales, and Create an AWS Account. The main section is titled "My Estimate" with a "Edit" button. It displays an "Estimate summary" table with the following data: Upfront cost 0.00 USD, Monthly cost 76.10 USD, and Total 12 months cost 913.20 USD (Includes upfront cost). To the right, there is a "Getting Started with AWS" section with "Get started for free" and "Contact Sales" buttons. Below this is a "My Estimate" table with columns for Service Name, Status, Upfront cost, Monthly cost, Description, Region, and Config Summary. The table lists two services: "Amazon Simple Stora..." and "Amazon Athena".

The screenshot shows the AWS Pricing Calculator interface. At the top, there are two informational banners: one about possible data loss if saving as CSV and another about genuine Office software. The main area is titled "Estimate summary". It displays a breakdown of costs:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W		
1	Estimate summary																								
2	Upfront	cc Monthly	0	Total 12 m		Currency																			
3	0	76.1	913.2	USD																					
4	* Includes upfront cost																								
5																									
6																									
7	Detailed Estimate																								
8	Group hier	Region	Description	Service	Upfront	Monthly	First 12 m	Currency		Status	Configuration summary														
9	My Estima	US East (N. Virginia)	S3 Standar		0	0.4	4.8	USD		S3 Standard storage (15 GB per month), PUT, COPY, POST, LIST requests to S3 Standard (555), GET, SELECT, and all other requests from S3 Standard (555), Dat															
10	My Estima	US East (N. Virginia)	S3 Manage		0	64.37	772.44	USD		S3 Storage Lens Objects (6 million per month), S3 Batch Operations Jobs (5 per month), S3 Batch Operations Objects (5 million per month), Size of encrypted c															
11	My Estima	US East (N. Virginia)	Amazon AI		0	11.33	135.96	USD		Total number of queries (120 per month), Amount of data scanned per query (5 GB), Number of DPUs (25), Length of time (hours) capacity is active (1 hours p															
12																									
13																									
14																									
15	Acknowledgement																								
16	* 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.																								
17																									
18																									
19																									
20																									
21																									
22																									
23																									
24																									
25																									

At the bottom, there's a navigation bar with tabs for "My Estimate (5)" and "+", and a set of standard browser control buttons (back, forward, search, etc.). The status bar at the bottom left shows "Ready" and "Accessibility: Unavailable".

Phase 2: Securing VPCs:

After securing the data in Amazon S3, the leadership team for Any Company Financial wants to focus on securing the *network* in the AWS Cloud that hosts the company's critical applications. They are aware of recent network security incidents and want to ensure that their network is protected from unauthorized access and attacks. So this task is to secure the virtual private clouds (VPCs) for the company's web servers.

Task 2.1: Review LabVPC and its associated resources:

In this task, I will familiarize mySelf with resources that already exist in the lab environment.

The screenshot shows the AWS VPC dashboard for the 'LabVPC' virtual private cloud. The main table lists three VPCs:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options
LabVPC	vpc-08317e3e87eb8e83e	Available	10.1.0.0/16	-	dopt-0e
-	vpc-0ab851ea5b719b0a	Available	172.31.0.0/16	-	dopt-0e
NetworkFirewallVPC	vpc-03710259a31c41b4a	Available	10.1.0.0/16	-	dopt-0e

Below the table, there are four cards: 'PC Show details' (Your AWS virtual network), 'Subnets (1)' (Subnets within this VPC, listing 'us-east-1a' and 'WebServerSubnet'), 'Route tables (1)' (Route network traffic to resources, listing 'rtb-09c48ab1653ffae9'), and 'Network conn' (Connections to other).

In the IAM console, locate the *VPCFlowLogsRole* IAM role. Review the permissions that are granted to this role.

The screenshot shows the AWS IAM console with the 'Identity and Access Management (IAM)' menu open. Under 'Access management', the 'Roles' section is selected, showing a list of roles. One role, 'VPCFlowLogPolicy', is highlighted. The details page for this role is displayed, showing its policy document:

```
1 ~ [ {  
2 ~ "Statement": [  
3 ~ {  
4 ~ "Action": [  
5 ~ "logs:CreateLogGroup",  
6 ~ "logs:CreateLogStream",  
7 ~ "logs:Describe*",  
8 ~ "logs:PutLogEvents"  
9 ~ ],  
10 ~ "Resource": "*",  
11 ~ "Effect": "Allow"  
12 ~ }  
13 ~ ]  
14 ~ ]
```

Below the policy document, there is a 'Permissions boundary (not set)' section.

In the Amazon EC2 console, observe the details for the *WebServer* instance.

The screenshot shows the AWS EC2 Instances page. The left sidebar shows various navigation options like 'EC2 Dashboard', 'Instances', 'Images', and 'Elastic Block Store'. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
WebServer	i-078e8be1899d7691f	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-107...
WebServer2	i-09d1f42acf88fd700	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-3-2...
aws-cloud9-Cl...	i-0863e0c671d41f91d	Running	t2.micro	2/2 checks passed	View alarms	us-east-1d	ec2-54...

Below the table, a specific instance, 'i-078e8be1899d7691f (WebServer)', is selected and its details are shown in a modal window. The 'Details' tab is active, displaying information such as:

- Instance ID: i-078e8be1899d7691f (WebServer)
- Public IPv4 address: 107.20.93.49
- Private IP4 addresses: 10.1.3.4
- Public IPv4 DNS: ec2-107-20-93-49.compute-1.amazonaws.com

The screenshot shows the AWS EC2 Instances page. The left sidebar includes links for EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances (with sub-links for Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots). The main content area displays a table of instances. The first instance, 'WebServer' (ID: i-078e8be1899d7691f), is selected and shown in a detailed view below. The detailed view shows the instance ID, state (Running), type (t2.micro), status (2/2 checks passed), alarm status (View alarms), availability zone (us-east-1a), and public IP (ec2-107-254-11-123). It also shows security details, including an IAM Role ('WebServerRole') and a Security group ('sg-04559738d19c085a5 (WebServerSecurityGroup)').

Task 2.2: Create a VPC flow log

In this task, I'm challenged to create a VPC flow log for *LabVPC*. The VPC Flow Logs feature can help me understand how inbound and outbound traffic flows through the VPC.

The screenshot shows the AWS VPC dashboard. The left sidebar includes links for EC2 Global View, Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, Peering connections), and Security. The main content area shows a table of VPCs. A success message at the top states 'Successfully created flow log for vpc-08317e3e87eb8e83e'. The table lists three VPCs: 'LabVPC' (selected, ID: vpc-08317e3e87eb8e83e, State: Available, IPv4 CIDR: 10.1.0.0/16, IPv6 CIDR: -), '-' (ID: vpc-0ab851eae5b719b0a, State: Available, IPv4 CIDR: 172.31.0.0/16, IPv6 CIDR: -), and 'NetworkFirewallVPC' (ID: vpc-03710259a31c41b4a, State: Available, IPv4 CIDR: 10.1.0.0/16, IPv6 CIDR: -). Below the table, a detailed view for 'vpc-08317e3e87eb8e83e / LabVPC' is shown, with tabs for Details, Resource map, CIDs, Flow logs, Tags, and Integrations. The 'Details' tab shows information such as VPC ID, State, Tenancy, DNS hostnames, and DNS resolution.

The screenshot shows the AWS VPC dashboard. At the top, there's a success message: "Successfully created flow log for vpc-08317e3e87eb8e83e." Below it, the "Your VPCs (1/3) Info" section lists three VPCs:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options
LabVPC	vpc-08317e3e87eb8e83e	Available	10.1.0.0/16	-	dopt-0e
-	vpc-0ab851ea5b719b0a	Available	172.31.0.0/16	-	dopt-0e
NetworkFirewallVPC	vpc-03710259a31c41b4a	Available	10.1.0.0/16	-	dopt-0e

Below the VPC list, the "Flow logs" tab is selected. It shows one flow log entry:

Name	Flow log ID	Filter	Destination type	Destination name
LabVPCFlowLogs	fl-0ac52a739c005cc0b	ALL	cloud-watch-logs	LabVPCFlowLogs

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

The page doesn't load. This is expected. Recall that an inexperienced employee of Any Company Financial made some mistakes and the result is that the network is not properly configured.

The browser window shows a connection error:

This site can't be reached
107.20.93.49 took too long to respond.

Try:

- Checking the connection
- Checking the proxy and the firewall
- Running Windows Network Diagnostics

ERR_CONNECTION_TIMED_OUT

[Reload](#) [Details](#)

-Use the AWS Cloud9 Cloud9Instance IDE to test access to the *WebServer* instance over the standard HTTP port (80) After some time passes, the connection fails or times out, Next, try to access the same instance on standard SSH port (22) This connection also fails or times out.

```

AWS Cloud9
Welcome to your development environment

Running transaction
  Installing : openssl111-pkcs11-0.4.10-6.amzn2.0.1.x86_64 1/6
  Installing : libopenssl111-libs-1.1.1g-12.amzn2.0.23.x86_64 2/6
  Installing : libretlts-3.8.1-1.el7.x86_64 3/6
  Installing : liblbd-1.1.0-1.el7.x86_64 4/6
  Installing : liblbsd-0.12.2-1.el7.x86_64 5/6
  Installing : netcat-1.22c-1.el7.x86_64 6/6
  Verifying : liblbd-0.12.2-1.el7.x86_64
  Verifying : libopenssl111-libs-1.1.1g-12.amzn2.0.23.x86_64
  Verifying : openssl111-pkcs11-0.4.10-6.amzn2.0.1.x86_64
  Verifying : liblbd-1.1.0-1.el7.x86_64
  Verifying : libretlts-3.8.1-1.el7.x86_64

Installed:
  netcat.x86_64 0:1.22c-1.el7

Dependency Installed:
  liblbsd.x86_64 0:0.12.2-1.el7  liblbd.x86_64 0:1.1.0-1.el7  libretlts.x86_64 0:3.8.1-1.el7  openssl111-libs.x86_64 1:1.1.1g-12.amzn2.0.23  openssl111-pkcs11.x86_64 0:0.4.10-6.amzn2.0.1

Complete!
voclabs:~/environment $ nc -vz 107.20.93.49 80
^C
voclabs:~/environment $ nc -vz 107.20.93.49 22

```

look at the log stream for the *LabVPCFlowLogs* log group to verify logs were generated by my actions in the previous step.

The screenshot shows the AWS CloudWatch Log Groups interface. On the left, there's a sidebar with navigation links like Dashboards, Alarms, Logs (Log groups, Log Anomalies, Live Tail, Logs Insights, Contributor Insights), Metrics, X-Ray traces, Events (Rules, Event Buses), Application Signals, and Network monitoring. The main area shows the 'CloudWatch > Log groups > LabVPCFlowLogs > All events' path. A table lists log events with columns for Timestamp, Message, and Log stream name. Each row contains a timestamp (e.g., 2024-09-29T17:57:03.000Z), a long message ID, and a log stream name (e.g., eni-0b3be1b6192aec322-all). The log entries are timestamped between 2024-09-29T17:57:03.000Z and 2024-09-29T17:58:33.000Z.

Timestamp	Message	Log stream name
2024-09-29T17:57:03.000Z	2 234683256885 eni-0b3be1b6192aec322 143.42.164.34 10.1.3...	eni-0b3be1b6192aec322-all
2024-09-29T17:57:03.000Z	2 234683256885 eni-0b3be1b6192aec322 165.232.183.249 10.1...	eni-0b3be1b6192aec322-all
2024-09-29T17:57:03.000Z	2 234683256885 eni-0b3be1b6192aec322 162.216.150.130 10.1...	eni-0b3be1b6192aec322-all
2024-09-29T17:57:03.000Z	2 234683256885 eni-0b3be1b6192aec322 35.203.211.76 10.1.3...	eni-0b3be1b6192aec322-all
2024-09-29T17:57:03.000Z	2 234683256885 eni-0b3be1b6192aec322 8.222.175.173 10.1.3...	eni-0b3be1b6192aec322-all
2024-09-29T17:58:03.000Z	2 234683256885 eni-0b3be1b6192aec322 124.92.87.40 10.1.3...	eni-0b3be1b6192aec322-all
2024-09-29T17:58:03.000Z	2 234683256885 eni-0b3be1b6192aec322 181.67.137.153 10.1.3...	eni-0b3be1b6192aec322-all
2024-09-29T17:58:03.000Z	2 234683256885 eni-0b3be1b6192aec322 65.49.1.116 10.1.3.4 ...	eni-0b3be1b6192aec322-all
2024-09-29T17:58:33.000Z	2 234683256885 eni-0b3be1b6192aec322 45.56.83.149 10.1.3...	eni-0b3be1b6192aec322-all
2024-09-29T17:58:33.000Z	2 234683256885 eni-0b3be1b6192aec322 98.98.170.134 10.1.3...	eni-0b3be1b6192aec322-all
2024-09-29T17:58:33.000Z	2 234683256885 eni-0b3be1b6192aec322 147.185.133.88 10.1.3...	eni-0b3be1b6192aec322-all

Filter the log entries to display only the events that originated from the public IP address of the AWS Cloud9 instance:

The screenshot shows the AWS CloudWatch Log groups interface. The left sidebar is collapsed, and the main area displays a table of log events. The table has columns for 'Timestamp' and 'Message'. A filter bar at the top shows '54.144.31.150'. The log entries are as follows:

Timestamp	Message	Log stream name
2024-10-02T18:04:16.000Z	2 234683256885 eni-0b3be1b6192aec322 54.144.31.150 10.1.3...	eni-0b3be1b6192aec322-all
	2 234683256885 eni-0b3be1b6192aec322 54.144.31.150 10.1.3.4 44792 80 6 6 360 1727892256 1727892314 REJECT OK	
2024-10-02T18:05:16.000Z	2 234683256885 eni-0b3be1b6192aec322 54.144.31.150 10.1.3...	eni-0b3be1b6192aec322-all
	2 234683256885 eni-0b3be1b6192aec322 54.144.31.150 10.1.3.4 44792 80 6 1 60 1727892316 1727892366 REJECT OK	
2024-10-02T18:05:50.000Z	2 234683256885 eni-0b3be1b6192aec322 54.144.31.150 10.1.3...	eni-0b3be1b6192aec322-all
2024-10-02T18:06:16.000Z	2 234683256885 eni-0b3be1b6192aec322 54.144.31.150 10.1.3...	eni-0b3be1b6192aec322-all
2024-10-02T18:07:16.000Z	2 234683256885 eni-0b3be1b6192aec322 54.144.31.150 10.1.3...	eni-0b3be1b6192aec322-all

Task 2.4: Configure route table and security group settings:

In this task, I'm challenged to create a route for traffic from the internet to access the *WebServerSubnet* through an internet gateway. This will allow inbound HTTP traffic to be directed to the *WebServer* instance. I will also be challenged to modify the security group that is associated with the *WebServer* instance to allow inbound traffic on ports 22 (SSH) and 80 (HTTP).

-Modify the route table that is associated with the *WebServerSubnet*.

The screenshot shows the AWS VPC dashboard. The left sidebar is collapsed, and the main area shows the details of a route table named 'rtb-09c48ab1653ffea9'. The 'Details' tab is selected, showing the following information:

- Route table ID: rtb-09c48ab1653ffea9
- Main: Yes
- Explicit subnet associations: -
- Edge associations: -
- VPC: vpc-08317e5e87eb8e83e | LabVPC
- Owner ID: 234683256885

The 'Routes' tab shows two routes:

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0e1570123fa91ff36	Active	No
10.1.0.0/16	local	Active	No

-Test access from port 80 to the *WebServer* instance again

```

nc -l -p 116 < /dev/null
Verifying : netcat-1.7.0-1.el7.x86_64
Verifying : libbsd-0.12.2-1.el7.x86_64
Verifying : openssl111-libs-1.1.1g-12.amzn2.0.23.x86_64
Verifying : openssl111-pkcs11-0.4.10-6.amzn2.0.1.x86_64
Verifying : libstdc++-11.1.0-1.el7.x86_64
Verifying : libretl-3.8.1.1.el7.x86_64

Installed:
  netcat.x86_64 0:1.22-1.el7

Dependency Installed:
  libbsd.x86_64 0:0.12.2-1.el7  libstdc++.x86_64 0:1.1.0-1.el7  libretl.x86_64 0:3.8.1-1.el7  openssl111-libs.x86_64 1:1.1.1g-12.amzn2.0.23  openssl111-pkcs11.x86_64 0:0.4.10-6.amzn2.0.1

Complete!
voclabs:/environment $ nc -vz 107.20.93.49 80
nc: connect to 107.20.93.49 port 80 (tcp) failed: Connection timed out
voclabs:/environment $ curl http://169.254.169.254/latest/meta-data/public-ipv4
54.144.31.150voclabs:/environment $
voclabs:/environment $ nc -vz 107.20.93.49 22
nc: connect to 107.20.93.49 port 22 (tcp) failed: Connection timed out
voclabs:/environment $ nc -vz 107.20.93.49 80
Connection to 107.20.93.49 80 port [tcp/http] succeeded!
voclabs:/environment $ 

```

-Modify the inbound rules for the security group

Inbound security group rules successfully modified on security group sg-04559738d19c085a5 | WebServerSecurityGroup

sg-04559738d19c085a5 - WebServerSecurityGroup

Details			
Security group name	Security group ID	Description	VPC ID
WebServerSecurityGroup	sg-04559738d19c085a5	WebServerSecurityGroup	vpc-08317e3e87eb8e83e
Owner	234683256885	Inbound rules count	Outbound rules count
		3 Permission entries	1 Permission entry

Inbound rules (3)

IP range	Protocol	Port	Action
0.0.0.0/0	TCP	80	Allow
0.0.0.0/0	TCP	22	Allow
0.0.0.0/0	HTTP	443	Allow

Details

Security group name	Security group ID	Description	VPC ID
WebServerSecurityGroup	sg-04559738d19c085a5	WebServerSecurityGroup	vpc-08317e3e87eb8e83e
Owner	234683256885	Inbound rules count 3 Permission entries	Outbound rules count 1 Permission entry

Inbound rules (3)

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-039cf925095737c5f	IPv4	SSH	TCP	22
-	sgr-09a094e6874c2ab...	IPv4	HTTP	TCP	80
-	sgr-0cac9cead89a2073a	IPv4	Custom TCP	TCP	8080

-test access from port 80 to the webpage on the *WebServer* instance. The test should be successful. The message "Hello world from WebServer!" displays:

```
vocabs:~/environment $ nc -vz 107.20.93.49 80
Connection to 107.20.93.49 80 port [tcp/http] succeeded!
vocabs:~/environment $
```

CodeWhisperer AWS profile.default

← → C ⚠ Not secure 107.20.93.49

Hello world from WebServer!

Task 2.5: Secure the WebServerSubnet with a network ACL

In this task, I'm challenged to configure a network access control list (ACL) to secure the subnet where the web server is running. The network ACL will provide an additional layer of security beyond the security group that i already configured.

-To test whether you can access port 22 on the *WebServer* instance, modify the *100* rule from *Allow* to *Deny*, **The connection fails or times out:**

```
vocabs:~/environment $ nc -vz 107.20.93.49 22
Connection to 107.20.93.49 22 port [tcp] failed: Connection timed out
vocabs:~/environment $
```

CodeWhisperer AWS profile.default

-Modify rule number 100 to *allow* connections on port 22 only and Edit the inbound rules for the network ACL again. Add a *new* rule that allows *HTTP* traffic from *anywhere*. Use *90* as the rule number:

Name	Network ACL ID	Associated with	Default	VPC ID
acl-05a222458c46059db	6 Subnets	Yes	vpc-0ab851eae5b719b0a	
acl-095a53f831228de31	2 Subnets	Yes	vpc-03710259a31c41b4a / NetworkFir...	
acl-0044c73e9f9e8a753	subnet-0e41184e2789af69b / WebServerSubnet	Yes	vpc-08317e3e87eb8e83e / LabVPC	

Inbound rules (3)						
Rule number	Type	Protocol	Port range	Source	Allow/Deny	
90	HTTP (80)	TCP (6)	80	0.0.0.0/0	Allow	
100	SSH (22)	TCP (6)	22	0.0.0.0/0	Allow	
*	All traffic	All	All	0.0.0.0/0	Deny	

Task 2.6: Review NetworkFirewallVPC and its associated resources:

In this phase so far, I worked to secure *LabVPC* by updating a route table, network ACL, and security group.

-Observe the existing NetworkFirewallVPC:

The screenshot shows the AWS VPC dashboard. On the left, there's a sidebar with options like EC2 Global View, Filter by VPC, Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, Peering connections), and Security (CloudShell, Feedback). The main area displays 'Your VPCs (1/3) Info' with three entries:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP o...
LabVPC	vpc-08317e3e87eb8e83e	Available	10.1.0.0/16	-	dopt-0...
-	vpc-0ab851ea5b719b0a	Available	172.31.0.0/16	-	dopt-0...
NetworkFirewallVPC	vpc-03710259a31c41b4a	Available	10.1.0.0/16	-	dopt-0...

A network diagram below the table shows 'Subnets (2)' and 'Route tables (1)'. The subnets are 'us-east-1a' (WebServer2Subnet, FirewallSubnet) and 'us-east-1b' (rtb-0a255b759afa75b91). The route table 'rtb-0a255b759afa75b91' connects to the subnets.

-observe the network ACL settings for *NetworkFirewallVPC*:

The screenshot shows the AWS Network ACL settings for the 'NetworkFirewallVPC'. The left sidebar is identical to the previous VPC dashboard screenshot. The main area shows the 'Network ACL' tab for 'acl-095a53f831228de31'.

IPv6-only: No
DNS64: Disabled
IP name: Owner
Owner: 234683256885

Network ACL: acl-095a53f831228de31

Inbound rules (2)

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	<input checked="" type="radio"/> Allow
*	All traffic	All	All	0.0.0.0/0	<input type="radio"/> Deny

Outbound rules (2)

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	<input checked="" type="radio"/> Allow
*	All traffic	All	All	0.0.0.0/0	<input type="radio"/> Deny

- observe the settings for the *WebServer2* instance and Confirm access to the *WebServer2* instance on ports 80 and 22, Verify HTTP access using a browser, should see the "Hello world from WebServer2!" message:

The screenshot shows a browser window with the URL '3.218.7.222'. The page content is 'Hello world from WebServer2!'

-Start an additional website that runs on *WebServer2* port 8080 and test access, The attempt should be successful, and should see the "**Hello world from WebServer2 port 8080!**" message:



Task 2.7: Create a network firewall

A screenshot of the AWS VPC Network Firewall console. The left sidebar shows navigation options like VPC dashboard, EC2 Global View, Virtual private cloud, Security, and CloudShell. The main area shows a success message: "You've successfully created firewall NetworkFirewall" and "You've successfully created firewall policy FirewallPolicy". Below this, the "NetworkFirewall" details are shown in an "Overview" tab, indicating a provisioning status, associated policy, and VPC. The "Firewall details" tab shows the name "NetworkFirewall".

Task 2.8: Create route tables

In this task, I'm challenged to create and configure three new route tables, including one for each subnet in the *NetworkFirewallVPC* and one to handle inbound (ingress) traffic for the internet gateway in *NetworkFirewallVPC*.

-Create a new route table in the *NetworkFirewallVPC*:

The screenshot shows the AWS VPC dashboard. In the top navigation bar, the 'Route tables' option is selected. Below it, a success message states: "Route table rtb-054d8d46cf24cb4ec | IGW-Ingress-Route-Table was created successfully." The main content area displays the details of the newly created route table, "rtb-054d8d46cf24cb4ec". The table has one route entry:

Destination	Target	Status	Propagated
10.1.0.0/16	local	Active	No

-Edit the route table to add a new route:

The screenshot shows the AWS VPC dashboard. The route table "rtb-054d8d46cf24cb4ec" now contains two routes:

Destination	Target	Status	Propagated
10.1.0.0/16	local	Active	No
10.1.3.0/28	vpc-e04a3f4da00cc8739d	Active	No

- Add an edge association to the *IGW-Ingress-Route-Table* so that the *NetworkFirewallIG* internet gateway is associated with the route table:

VPC dashboard

rtb-054d8d46cf24cb4ec / IGW-Ingress-Route-Table

Details

Route table ID rtb-054d8d46cf24cb4ec	Main No	Explicit subnet associations -	Edge associations igw-08e7d3175bcf5126d / NetworkFirewallVPC
VPC vpc-03710259a31c41b4a NetworkFirewallVPC	Owner ID 234683256885		

Routes

Destination	Target	Status	Propagated
10.1.0.0/16	local	Active	No
10.1.3.0/28	vpc-04a3f4da00cc8739d	Active	No

- Create another route table in *NetworkFirewallVPC* for the *Webserver2Subnet*:

VPC dashboard

rtb-067e0ddb28ae28913 / WebServer2-Route-Table

Details

Route table ID rtb-067e0ddb28ae28913	Main No	Explicit subnet associations -	Edge associations -
VPC vpc-03710259a31c41b4a NetworkFirewallVPC	Owner ID 234683256885		

Routes

Destination	Target	Status	Propagated
10.1.0.0/16	local	Active	No

Task 2.9: Configure logging for the network firewall

In this task, I'm challenged to configure logging for the network firewall so that I can analyze details of network traffic requests.

- Create a CloudWatch log group named *NetworkFirewallVPCLogs* with a retention setting of *6 months*:

The screenshot shows the AWS CloudWatch Log groups interface. A green success message at the top states: "Log group 'NetworkFirewallVPCLogs' has been created." The main table lists four log groups:

Log group	Log class	Anomaly d...	Data pr...	Sensiti...	Retenti...	Me
/aws/lambda/c133601a3382972l7710701t1-AdjustA...	Standard	Configure	-	-	Never expire	-
/aws/lambda/c133601a3382972l7710701t1-AdjustB...	Standard	Configure	-	-	Never expire	-
LabVPCFlowLogs	Standard	Configure	-	-	Never expire	-
NetworkFirewallVPCLogs	Standard	Configure	-	-	6 months	-

-in Firewall details area, and configure both Alert and Flow type logging:

The screenshot shows the AWS VPC Firewall details configuration page. A green success message at the top states: "You've successfully updated the firewall NetworkFirewall". The "Logging" section is expanded, showing the following configuration:

Log type	Alert log destination	Flow log destination	TLS log destination
Flow, Alert	CloudWatch log group - NetworkFirewallVPCLogs	CloudWatch log group - NetworkFirewallVPCLogs	Not configured

-Test the logging configuration by attempting to load *WebServer2* instance's website on port 80 :

The screenshot shows a web browser displaying the response from the WebServer2 instance. The address bar shows the IP address 3.218.7.222. The page content is "Hello world from WebServer2!".

in the *NetworkFirewallVPCLogs* log group. Filter the log events by public IP address:

The screenshot shows the AWS CloudWatch Log Events interface. The left sidebar is collapsed, showing options like 'Logs' (selected), 'Metrics', 'X-Ray traces', 'Events', and 'Application Signals'. The main area displays a table of log events. The first column is 'Timestamp', the second is 'Message', and the third is 'Log stream name'. A search bar at the top contains the query '54.144.31.150'. The log entries show multiple instances of a Network Firewall rule being applied to traffic from 54.144.31.150 to 10.1.3.0/24, resulting in a 'REJECT OK' response.

Task 2.10: Configure the firewall policy and test access:

In this task, I'm challenged to define and add a stateful rule group to the network firewall's policy.

The screenshot shows the AWS VPC Firewall Policies interface. The left sidebar lists 'Virtual private cloud' and 'Security' sections. The main area shows a success message: 'You've successfully created rule group NetworkFirewallVPCRuleGroup'. Below this, the 'NetworkFirewall' page is displayed. It has tabs for 'Overview' (selected) and 'Info'. Under 'Overview', it shows 'Firewall status' as 'Ready', 'Associated firewall policy' as 'FirewallPolicy', and 'Associated VPC' as 'vpc-03710259a31c41b4a'. There are tabs for 'Firewall details', 'Firewall policy settings' (selected), and 'Monitoring'. The 'Firewall policy settings' tab shows a section for 'Stateless default actions' with two columns: 'Actions for full packets' (Forward to stateful rule groups) and 'Actions for fragmented packets' (Pass). An 'Edit' button is located in the top right corner of this section.

- Cost estimate to secure a VPC with a network firewall:

Estimate summary															
Upfront & Monthly costs															
1	Estimate summary														
2	Upfront & Monthly costs	Total 12 months	Currency USD												
3	0	426.39	5116.68 USD												
4	* Includes upfront cost														
5															
6															
7	Detailed Estimate														
8	Group hierarchy	Region	Description	Service	Upfront	Monthly	First 12 months	Currency	Status	Configuration summary					
9	My Estimate	US East (N. Virginia)	Amazon ElastiCache	Amazon ElastiCache	0	8.468	101.62	USD	Active	Tenancy (Shared Instances), Operating system (Linux), Workload (Consistent, Number of instances: 1), Advance EC2 instance (t2.micro), Pricing strategy (On-Demand)					
10	My Estimate	US East (N. Virginia)	IPAM	IPAM	0	0.2	2.4	USD	Active	Number of active IP addresses (1)					
11	My Estimate	US East (N. Virginia)	Data Transfer	Data Transfer	0	0	0	USD	Active	DT Inbound: Internet (100 GB per month), DT Outbound: US East (Verizon) - Nashville (1 TB per month), DT Intra-Region: (0 TB per month)					
12	My Estimate	US East (N. Virginia)	IPAM	IPAM	0	0.2	2.4	USD	Active	Number of active IP addresses (1)					
13	My Estimate	US East (N. Virginia)	AWS Network Firewall	AWS Network Firewall	0	417.52	5010.24	USD	Active	Number of AWS Network Firewall endpoints (1), Usage per endpoint (30 days), Data processed per month (2 TB)					
14															
15															
16															
17	Acknowledgement														
18	* 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.														
19															
20															
21															
22															
23															
24															
25															

Phase 3: Securing AWS resources by using AWS KMS:

The legal department at Any Company Financial received notice from the U.S. Federal Deposit Insurance Corporation (FDIC) that the company needs to encrypt sensitive information, such as PII, credit card numbers, and social security numbers. The legal department contacted manager, the director of IT, and they tasked to implement encryption and tokenization to meet regulatory compliance standards.

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

In this task, I will create an AWS KMS customer managed key. I will then configure automatic key rotation on the key.

- Create an AWS KMS customer managed key:

The screenshot shows the AWS KMS console. In the top navigation bar, the 'Customer managed keys' option is selected. A green success message at the top right states: 'Your AWS KMS key was created with alias MyKMSKey and key ID e11a4eae-d16b-4c31-8a63-fd75df995af.'. Below this, the 'Customer managed keys (1)' section displays a single key entry:

Aliases	Key ID	Status	Key type	Key spec	Key usage
MyKMSKey	e11a4eae-d16b-4c31-8a63-fd75df995af	Enabled	Symmetric	SYMMETRIC_DEFAULT	Encrypt and decrypt

- Configure AWS KMS key rotation on the new key so that it is automatically rotated every year.

The screenshot shows the detailed view of the 'MyKMSKey' key. The 'Key rotation' tab is selected. Key rotation settings include:

Status	Rotation period	Date of last automatic rotation	Next rotation date
Enabled	365	-	Oct 03, 2025

On-demand key rotation is also mentioned, with a 'Rotate Now' button.

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

In this task, I'm challenged to modify the policy of the AWS KMS key that I created so that the *sofia* user will be authorized to use the key. I'm also challenged to analyze the IAM policy that controls what the *sofia* user can do in the AWS account.

- Modify the key policy:

The screenshot shows the AWS KMS console. On the left, there's a sidebar with 'Key Management Service (KMS)' selected. Under 'Customer managed keys', there's a list of custom key stores: 'AWS CloudHSM key stores' and 'External key stores'. The main panel shows a key named 'voclabs'. Under 'Key deletion', there's a checked checkbox for 'Allow key administrators to delete this key'. Under 'Key users (2)', there are two entries: 'voclabs' (Role) and 'sofia' (User). Both users are listed under the 'Role' column. At the bottom, there's a section for 'Other AWS accounts' with a button to 'Add other AWS accounts'.

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

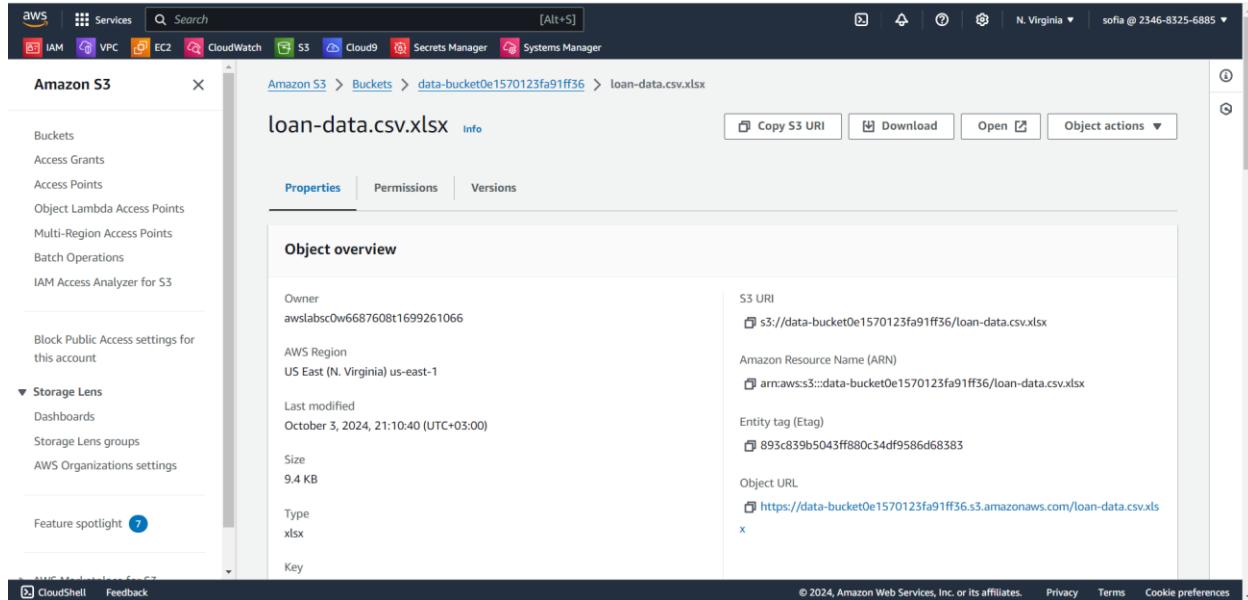
In this task, I'm challenged to use the AWS KMS key that I created to encrypt an object in the *data-bucket* S3 bucket. I will then test access to the object.

- Modify the encryption settings on the *data-bucket* S3 bucket:

The screenshot shows the AWS S3 console. On the left, there's a sidebar with 'Amazon S3' selected. Under 'Buckets', there are several options: 'Access Grants', 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', 'IAM Access Analyzer for S3', 'Block Public Access settings for this account', 'Storage Lens' (with 'Dashboards', 'Storage Lens groups', and 'AWS Organizations settings'), and 'Feature spotlight'. The main panel shows a bucket named 'data-bucket'. A green success message at the top says 'Successfully edited default encryption. Objects uploaded, modified, or copied into this bucket will inherit this encryption configuration unless otherwise specified.' Below this, there's a 'Bucket overview' section with details: 'AWS Region: US East (N. Virginia) us-east-1', 'Amazon Resource Name (ARN): arn:aws:s3:::data-bucket0e1570123fa91ff36', and 'Creation date: September 24, 2024, 19:13:54 (UTC+03:00)'. There are sections for 'Bucket Versioning' (Enabled), 'Multi-factor authentication (MFA) delete' (Disabled), and 'Tags (0)'. At the bottom, there are buttons for 'Edit' and 'CloudShell Feedback'.

- log in to the AWS Management Console as the *sofia* IAM user:

- Note that: The upload, open or download should be successful



The screenshot shows the AWS Management Console interface for the Amazon S3 service. On the left, there's a sidebar with various navigation options. The main area shows a file named "loan-data.csv.xlsx" in a bucket. The "Properties" tab is selected, providing detailed information about the object. At the top right, there are buttons for "Copy S3 URI", "Download", "Open", and "Object actions". The overall layout is clean and professional, typical of AWS's design.

- Sign the *sofia* user out of the console, and then sign in as the *paulo* user. Then, try to open or download, The attempt should fail:



```
This XML file does not appear to have any style information associated with it. The document tree is shown below.

<Error>
  <Code>AccessDenied</Code>
  <Message>User: arn:aws:iam::234683256885:user/paulo is not authorized to perform: kms:Decrypt on resource: arn:aws:kms:us-east-1:234683256885:key/e11a4eae-d16b-4c31-8a63-fd75d9f995af because no identity-based policy allows the kms:Decrypt action</Message>
  <RequestId>JH2ZPQKXDKWV521</RequestId>
  <HostId>DR1MDtZak63Xdyt89Qs31/KLYBCCum30gPQvHG511Cu0zDYC9qPHYb01JAkx9ioTe9HNZPU31/BdEz1X7t44aPWFy6s+KxaZDsCGo+Iqu=</HostId>
</Error>
```

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

In this task, I'm challenged to use the AWS KMS key again, but now I will use it to encrypt the root volume of a new EC2 instance.

- create a new EC2 instance:

The screenshot shows the AWS EC2 'Launch an instance' success page. At the top, there's a green success banner stating 'Successfully initiated launch of instance (i-01d363d20fd0c744b)'. Below it, a 'Next Steps' section contains several links: 'Create billing and free tier usage alerts', 'Connect to your instance', 'Connect an RDS database', and 'Create EBS snapshot policy'. Each link has a brief description and a 'Learn more' link.

- verify that the instance root volume is encrypted:

The screenshot shows the AWS EC2 Instances page. The left sidebar lists various EC2-related options like EC2 Dashboard, Instances, Images, and Elastic Block Store. The main pane displays a table of instances, with one row selected: 'EncryptedInst...' (Instance ID: i-01d363d20fd0c744b). The detailed view for this instance shows 'Root device details' (Root device name: /dev/xvda, Root device type: EBS, EBS optimization: disabled) and 'Block devices' (Volume ID: vol-01c9208f97cf958bb, Device name: /dev/xvda, Volume size (GiB): 8, Attachment status: Attached, Attachment time: 2024/10/03 21:49 GMT+3, Encrypted: Yes, KMS key ID: e11a4eae-d16b-4c3). The table header includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP.

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

In this task, I'm challenged to use the AWS Command Line Interface (AWS CLI) to encrypt data in place by using the AWS KMS key. I'm also challenged to see how to decrypt the encrypted data.

- create a file that you will later try to encrypt

AWS CloudShell terminal window showing the creation of a file named 'data_unencrypted.txt'. The terminal output shows:

```
Amazon Linux 2
AI2 End of Life is 2025-06-30.
A newer version of Amazon Linux is available!
Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-10-1-3-7 ~]$ touch data_unencrypted.txt
[ec2-user@ip-10-1-3-7 ~]$ vi data_unencrypted.txt
[ec2-user@ip-10-1-3-7 ~]$ ll
total 4
-rw-rw-r-- 1 ec2-user ec2-user 122 Oct  3 19:11 data_unencrypted.txt
[ec2-user@ip-10-1-3-7 ~]$
```

The terminal is titled 'i-01d363d20fd0c744b (EncryptedInstance)'. The status bar at the bottom right shows '© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

- generate a data key from the AWS KMS key:

AWS CloudShell terminal window showing the generation of a data key from an AWS KMS key. The terminal output shows:

```
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Wed Oct  2 20:30:22 2024 from 18.206.107.29
[ec2-user@webserver2 ~]$ touch data_unencrypted.txt
[ec2-user@webserver2 ~]$ vi data_unencrypted.txt
[ec2-user@webserver2 ~]$ aws kms list-keys
{
  "Keys": [
    {
      "KeyId": "8f19dd94-64d2-4ae9-ad93-cb60ff882ca5",
      "KeyArn": "arn:aws:kms:us-east-1:234683256885:key/8f19dd94-64d2-4ae9-ad93-cb60ff882ca5"
    },
    {
      "KeyId": "90b449e1-fcf3-409d-b9ab-8b15c556201a",
      "KeyArn": "arn:aws:kms:us-east-1:234683256885:key/90b449e1-fcf3-409d-b9ab-8b15c556201a"
    },
    {
      "KeyId": "ella4eae-d16b-4c31-8a63-fd75d9f995af",
      "KeyArn": "arn:aws:kms:us-east-1:234683256885:key/ella4eae-d16b-4c31-8a63-fd75d9f995af"
    }
  ]
}
[ec2-user@webserver2 ~]$
```

The terminal is titled 'i-09d1f42acf88fd700 (WebServer2)'. The status bar at the bottom right shows '© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

- generate a *data key* for the MyKMSKey, save it to a bash variable, and then echo the data key details:

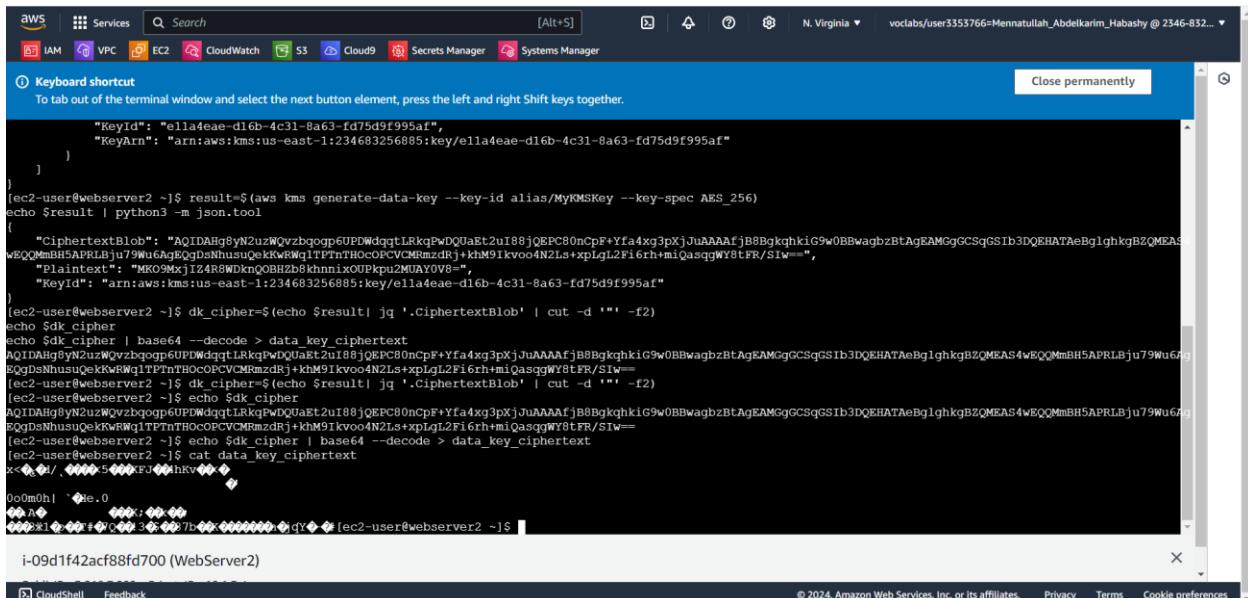
```

        KeyArn : arn:aws:kms:us-east-1:23468325685:key/e11a4eae-d16b-4c31-8a63-fd75d9f995af
    }
}

[ec2-user@webserver2 ~]$ result=$(aws kms generate-data-key --key-id alias/MyKMSKey --key-spec AES_256)
echo $result | python3 -m json.tool
{
    "CiphertextBlob": "AQIDAHg8yN2uz#Wvzbqogp6UPDwdqgtLRkgPwDQJaEt2uI88jQEPC80ncpF+Yfa4xg3pxjJuAAAAfjb8BgkqhkiG9w0BBwagbzBtAgEAMggGCSqGSIB3DQEhATAeBglghkgBZQMEAS4wQQMmBH5APRLBju79Wu6AgEqQpNsHusuQeKvRWq1TPThOcoPCVCMRmzdrj+kH9Ikvo04N2Ls+xpLgL2F16rh+miQasggW8tFR/SIV==",
    "Plaintext": "MK09MxjIZ4R8DknQOBHbz8khnnixOUPku2MUAY0V8=",
    "KeyId": "arn:aws:kms:us-east-1:23468325685:key/e11a4eae-d16b-4c31-8a63-fd75d9f995af"
}
[ec2-user@webserver2 ~]$ i-09d1f42acf88fd700 (WebServer2)

```

- Save the data key to disk to a text file in base64-encoded format:



```

aws Services Search [Alt+S] N. Virginia v vocabs/user3553766=Mennatullah_Abdelkarim_Habashy @ 2346-832...
  IAM VPC EC2 CloudWatch S3 Cloud9 Secrets Manager Systems Manager

① Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together.

[ec2-user@webserver2 ~]$KeyId": "e11a4eae-d16b-4c31-8a63-fd75d9f995af",
"KeyArn": "arn:aws:kms:us-east-1:23468325685:key/e11a4eae-d16b-4c31-8a63-fd75d9f995af"
}

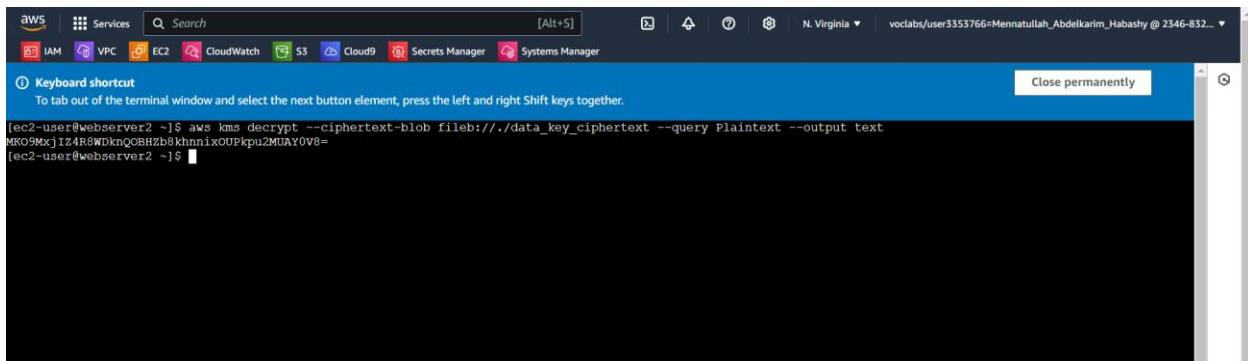
[ec2-user@webserver2 ~]$ result=$(aws kms generate-data-key --key-id alias/MyKMSKey --key-spec AES_256)
echo $result | python3 -m json.tool
{
    "CiphertextBlob": "AQIDAHg8yN2uz#Wvzbqogp6UPDwdqgtLRkgPwDQJaEt2uI88jQEPC80ncpF+Yfa4xg3pxjJuAAAAfjb8BgkqhkiG9w0BBwagbzBtAgEAMggGCSqGSIB3DQEhATAeBglghkgBZQMEAS4wQQMmBH5APRLBju79Wu6AgEqQpNsHusuQeKvRWq1TPThOcoPCVCMRmzdrj+kH9Ikvo04N2Ls+xpLgL2F16rh+miQasggW8tFR/SIV==",
    "Plaintext": "MK09MxjIZ4R8DknQOBHbz8khnnixOUPku2MUAY0V8=",
    "KeyId": "arn:aws:kms:us-east-1:23468325685:key/e11a4eae-d16b-4c31-8a63-fd75d9f995af"
}
[ec2-user@webserver2 ~]$ dk_cipher=$(/echo $result| jq '.CiphertextBlob' | cut -d '"' -f2)
echo $dk_cipher
echo $dk_cipher | base64 --decode > data_key_ciphertext
AQIDAHg8yN2uz#Wvzbqogp6UPDwdqgtLRkgPwDQJaEt2uI88jQEPC80ncpF+Yfa4xg3pxjJuAAAAfjb8BgkqhkiG9w0BBwagbzBtAgEAMggGCSqGSIB3DQEhATAeBglghkgBZQMEAS4wQQMmBH5APRLBju79Wu6AgEqQpNsHusuQeKvRWq1TPThOcoPCVCMRmzdrj+kH9Ikvo04N2Ls+xpLgL2F16rh+miQasggW8tFR/SIV==

[ec2-user@webserver2 ~]$ echo $dk_cipher=$(/echo $result| jq '.CiphertextBlob' | cut -d '"' -f2)
[ec2-user@webserver2 ~]$ echo $dk_cipher
AQIDAHg8yN2uz#Wvzbqogp6UPDwdqgtLRkgPwDQJaEt2uI88jQEPC80ncpF+Yfa4xg3pxjJuAAAAfjb8BgkqhkiG9w0BBwagbzBtAgEAMggGCSqGSIB3DQEhATAeBglghkgBZQMEAS4wQQMmBH5APRLBju79Wu6AgEqQpNsHusuQeKvRWq1TPThOcoPCVCMRmzdrj+kH9Ikvo04N2Ls+xpLgL2F16rh+miQasggW8tFR/SIV==

[ec2-user@webserver2 ~]$ echo $dk_cipher | base64 --decode > data_key_ciphertext
[ec2-user@webserver2 ~]$ cat data_key_ciphertext
x-09d1f42acf88fd700 (WebServer2)

```

- regenerate the plaintext version of the data key from the base64-encoded ciphertext at anytime:



```

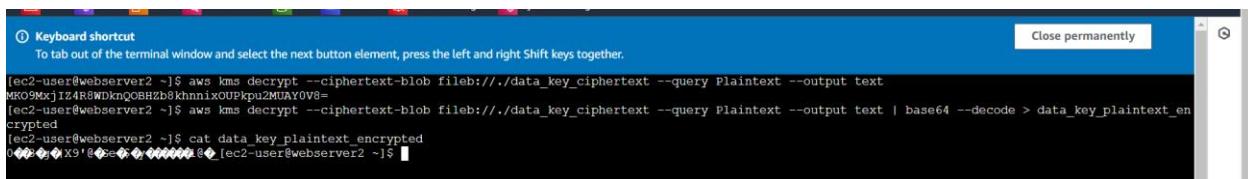
aws Services Search [Alt+S] N. Virginia v vocabs/user3553766=Mennatullah_Abdelkarim_Habashy @ 2346-832...
  IAM VPC EC2 CloudWatch S3 Cloud9 Secrets Manager Systems Manager

① Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together.

[ec2-user@webserver2 ~]$ aws kms decrypt --ciphertext-blob fileb://./data_key_ciphertext --query Plaintext --output text
MK09MxjIZ4R8DknQOBHbz8khnnixOUPku2MUAY0V8=
[ec2-user@webserver2 ~]$ 

```

- also save the previous result to a file in base64-encoded format:



```

① Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together.

[ec2-user@webserver2 ~]$ aws kms decrypt --ciphertext-blob fileb://./data_key_ciphertext --query Plaintext --output text
MK09MxjIZ4R8DknQOBHbz8khnnixOUPku2MUAY0V8=
[ec2-user@webserver2 ~]$ aws kms decrypt --ciphertext-blob fileb://./data_key_ciphertext --query Plaintext --output text | base64 --decode > data_key_plaintext_encrypted
[ec2-user@webserver2 ~]$ cat data_key_plaintext_encrypted

```

- To encrypt the *data_unencrypted.txt* file and try to read the file:

It's not human-readable. Then, Try To delete the unencrypted version of the file

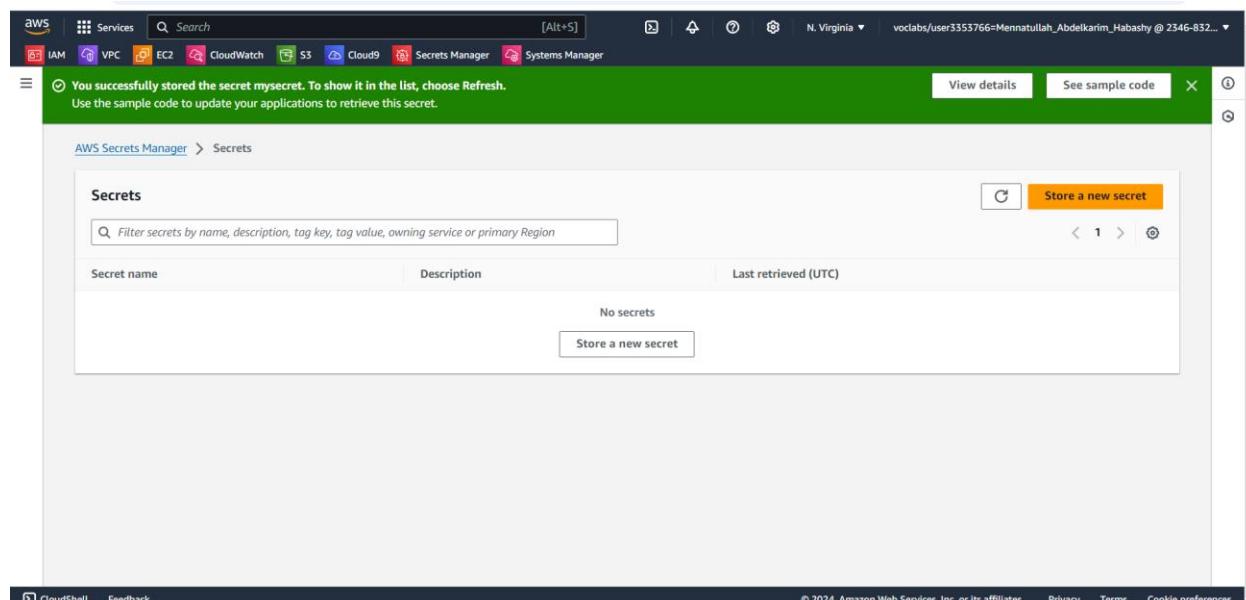
- Decrypt the file to prove that the data is retrievable and print the results to ensure the data is now readable:

```
[ec2-user@webserver2 ~]$ openssl enc -d -aes-256-cbc -pbkdf2 -in data_encrypted -out data_decrypted.txt -pass file:./data_key_plaintext_encrypted  
[ec2-user@webserver2 ~]$ cat data_decrypted.txt  
xxxxxxxxxx  
echo "Let's encrypt these file contents. Sensitive data here." > data_unencrypted.txt  
cat data_unencrypted.txt  
[ec2-user@webserver2 ~]$ █
```

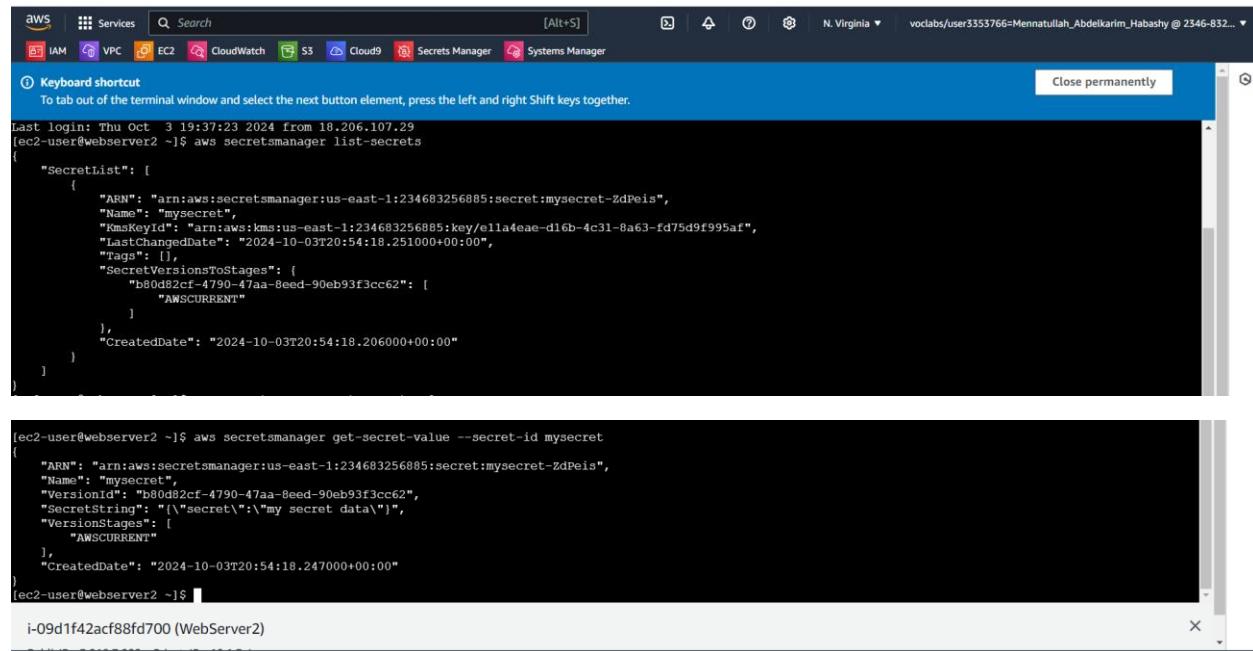
Task 3.6: Use AWS KMS to encrypt a Secrets Manager secret

In this task, I'm challenged to create a key-value pair (a *secret*), which I will encrypt with my AWS KMS key and store in Secrets Manager. I'm then challenged to verify that I can retrieve the secret by using the AWS CLI.

- create a new secret of type *Other type of secret*:



-Invoking and retrieve the secret:

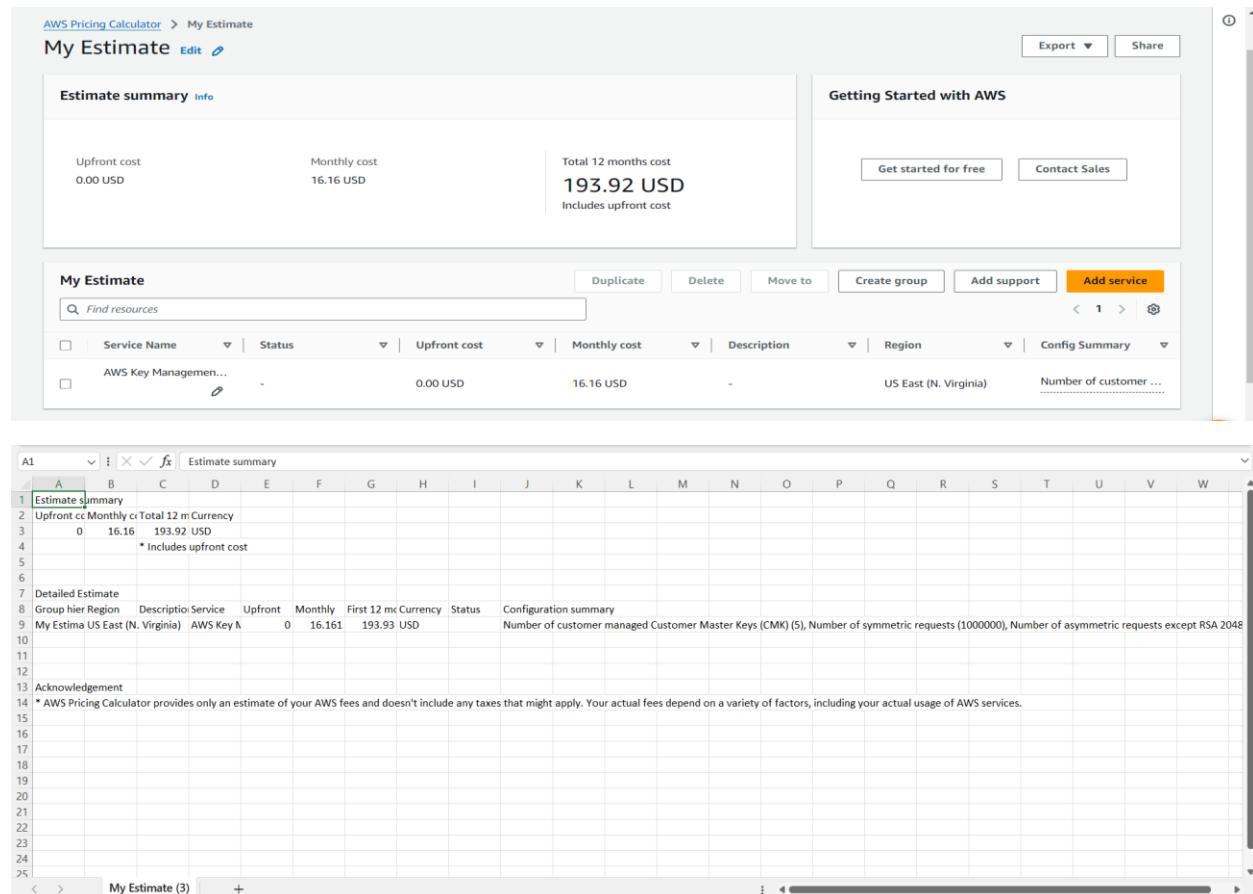


```
aws Services Search [Alt+S] N. Virginia vocabs/user3353766=Mennatullah_Abdelkarim_Habashy @ 2346-832... ▾
IAM VPC EC2 CloudWatch S3 Cloud9 Secrets Manager Systems Manager
Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together.

Last login: Thu Oct 3 19:37:23 2024 from 18.206.107.29
[ec2-user@webserver2 ~]$ aws secretsmanager list-secrets
{
    "SecretList": [
        {
            "ARN": "arn:aws:secretsmanager:us-east-1:234683256885:secret:mysecret-ZdPeis",
            "Name": "mysecret",
            "KmsKeyId": "arn:aws:kms:us-east-1:234683256885:key/e11a4eae-d16b-4c31-8a63-fd75d9f995af",
            "LastChangedDate": "2024-10-03T20:54:18.251000+00:00",
            "Tags": [],
            "SecretVersionsToStages": [
                {
                    "b80d82cf-4790-47aa-8eed-90eb93f3cc62": [
                        "AWSCURRENT"
                    ],
                    "CreatedDate": "2024-10-03T20:54:18.206000+00:00"
                }
            ]
        }
    ]
}

[ec2-user@webserver2 ~]$ aws secretsmanager get-secret-value --secret-id mysecret
{
    "ARN": "arn:aws:secretsmanager:us-east-1:234683256885:secret:mysecret-ZdPeis",
    "Name": "mysecret",
    "VersionId": "b80d82cf-4790-47aa-8eed-90eb93f3cc62",
    "SecretString": "{\"secret\":\"my secret data\"}",
    "VersionStages": [
        "AWSCURRENT"
    ],
    "CreatedDate": "2024-10-03T20:54:18.247000+00:00"
}
[ec2-user@webserver2 ~]$ i-09d1f42acf88fd700 (WebServer)
```

Cost assessment for using AWS KMS:



AWS Pricing Calculator > My Estimate

My Estimate Edit

Estimate summary Info

Upfront cost: 0.00 USD Monthly cost: 16.16 USD Total 12 months cost: **193.92 USD** Includes upfront cost

Getting Started with AWS

Get started for free Contact Sales

My Estimate

Service Name	Status	Upfront cost	Monthly cost	Description	Region	Config Summary
AWS Key Management...	-	0.00 USD	16.16 USD	-	US East (N. Virginia)	Number of customer ...

Estimate summary

Upfront	Monthly	Total 12 m	Currency
0	16.16	193.92	USD

* Includes upfront cost

Detailed Estimate

Group	Hier Region	Description	Service	Upfront	Monthly	First 12 m	Currency	Status	Configuration summary
My Estima	US East (N. Virginia)	AWS Key M	N	0	16.161	193.93	USD		Number of customer managed Customer Master Keys (CMK) (5), Number of symmetric requests (1000000), Number of asymmetric requests except RSA 2048

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:

in this phase The solution needs to do the following:

- Track all API calls to S3 buckets.
- Monitor application logs.
- Notify team members in case of security incidents.
- Monitor AWS resource configurations and automatically modify configurations that are out of compliance.

Task 4.1: Use CloudTrail to record Amazon S3 API calls:

-Create a CloudTrail trail:

The screenshot shows the AWS CloudTrail console interface. At the top, there are several notifications: "We're continuing to improve the CloudTrail Lake console experience to make it easier to use. Let us know what you think.", "New CloudTrail event data store pricing tier", and "Trail successfully created". Below these, the "Trails" section is displayed with a table. The table has columns: Name, Home region, Multi-region trail, Insights, Organization trail, S3 bucket, Log file prefix, CloudWatch Logs log group, and Status. A single row is present, representing a trail named "data-bucket-reads-writes". The "data-bucket-reads-writes" link is underlined. The "Home region" is listed as "US East (N. Virginia)". The "Multi-region trail" and "Organization trail" status is "Yes". The "Insights" and "Status" are "Disabled". The "S3 bucket" is "cloudtrail-logs-0e1570123fa91ff56". The "Log file prefix" and "CloudWatch Logs log group" are both "-". The "Status" is "Logging". There are buttons for "Copy events to Lake", "Delete", and "Create trail". At the bottom of the page, there are links for "CloudShell", "Feedback", "© 2024, Amazon Web Services, Inc. or its affiliates.", "Privacy", "Terms", and "Cookie preferences".

-Upload the *customer-data.csv* file to *data-bucket*:

The screenshot shows the AWS S3 console with a green success banner at the top stating "Upload succeeded". Below it, a summary table shows the destination as "s3://data-bucket0e1570123fa91ff36" with 1 file uploaded (9.6 KB, 100.00% success rate) and 0 files failed. A table below lists the uploaded file "customer-data.csv" with details like name, type (application/csv), size (9.6 KB), status (Succeeded), and error (none). The interface includes tabs for "Files and folders" and "Configuration".

-Use the CloudTrail console to create an Athena table:

The screenshot shows the AWS CloudTrail console under the "Data" tab. In the main area, a SQL query is being run to create an external table named "default.bucket". The query is as follows:

```
1 CREATE EXTERNAL TABLE `default.bucket`(`bucketowner` STRING, `bucket_name` STRING, `requestdatetime` STRING, `remoteip` STRING, `requester` STRING, `requestid` STRING, `operation` STRING, `key` STRING, `request_url` STRING, `httpstatus` STRING, `errorcode` STRING, `bytessent` BIGINT, `objectsize` BIGINT, `totaltime` STRING)
```

The results of the query show it was completed successfully. The interface includes tabs for "Tables and views" and "Views (0)".

-Run an Athena query to retrieve the CloudTrail event log data in Result area:

The image contains two screenshots of the Amazon Athena Query Editor interface.

Screenshot 1 (Top): Settings Configuration

This screenshot shows the "Settings" tab of the Query editor. It displays "Query result location" as `s3://athena5234/`, "Encrypt query results" as "Turned off", "Expected bucket owner" as "None", and "Assign bucket owner full control over query results" as "Turned off". A green banner at the top indicates "Settings successfully updated."

Screenshot 2 (Bottom): Query Execution

This screenshot shows the "Editor" tab of the Query editor. It displays a message about typeahead code suggestions and a sample SQL query:

```
1 SELECT eventtime, useridentity.principalid, requestparameters, eventname
2 FROM default.bucket
3 WHERE
4   eventname in ('PutObject') AND
5   requestparameters LIKE '%customer-data.csv'
6 Limit 10;
```

The "Data" sidebar on the left shows a database named "default" and tables "bucket" and "bucket_logs". The bottom of the screen includes standard AWS navigation links like CloudShell, Feedback, and Copyright information.

Task 4.2: Use CloudWatch Logs to monitor secure logs

In this task I have been create a solution to monitor access to *EncryptedInstance*.

-Create a CloudWatch log group with all default settings:

The screenshot shows the AWS CloudWatch service interface. On the left, there's a navigation sidebar with sections like Favorites and recent, Alarms, Logs (with Log groups selected), Metrics, X-Ray traces, Events, Application Signals, and Network monitoring. The main content area is titled "Log groups (5)" and displays a table of log groups. One row is highlighted in blue, representing the newly created "EncryptedInstanceSecureLogs" log group. The table includes columns for Log group name, Log class, Anomaly detection status, Data processing status, Sensitivity, Retention period, and Metrics. A green banner at the top of the main content area says "Log group 'EncryptedInstanceSecureLogs' has been created." At the bottom right of the main content area, there are links for "CloudShell", "Feedback", and copyright information.

-install the CloudWatch agent and a Linux daemon named collectd, which the CloudWatch agent will use:

The screenshot shows an AWS CloudShell terminal window. The terminal output shows the installation of the "collectd" package using the "yum" command. The output includes the package name, version, and source. The terminal window also shows a keyboard shortcut message and a "Close permanently" button in the top right corner. The bottom of the window shows standard CloudShell navigation links: "CloudShell", "Feedback", and "Cookie preferences".

```
Installed:
  collectd.x86_64 0:5.8.1-1.amzn2.0.2

Complete!
 2 httpd_modules      available  [ =1.0  =stable ]
 3 memcached1.5       available  \
    [ =1.5.1  =1.5.16  =1.5.17 ]
 9 R3.4                 available  [ =3.4.3  =stable ]
10 rustl               available  \
    [ =1.22.1  =1.26.0  =1.26.1  =1.27.2  =1.31.0  =1.38.0
     =stable ]
18 libreoffice          available  \
    [ =5.0.6.2_15  =5.3.6.1  =stable ]
19 gimp                available  [ =2.8.22 ]
20 fdocker=latest      enabled   \
    [ =17.12.1  =18.03.1  =18.06.1  =18.09.9  =stable ]
21 mate-desktop1.x      available  \
    [ =1.19.0  =1.20.0  =stable ]
22 GraphicsMagick1.3   available  \
    [ =1.3.29  =1.3.32  =1.3.34  =stable ]
24 epel                 available  [ =7.11  =stable ]
25 testing              available  [ =1.0  =stable ]
26 ecs                 available  [ =stable ]
27 icorrecto8           available  \
    [ =1.8.0_192  =1.8.0_202  =1.8.0_212  =1.8.0_222  =1.8.0_232
     =1.8.0_242  =stable ]
32 lustre2.10            available  \
    [ =2.10.5  =2.10.8  =stable ]
34 lynis                available  [ =stable ]
```

-Download and configure a JSON file that provides configuration details for the CloudWatch agent.

- To download the template file:

```

aws | Services Search [Alt+S] N. Virginia vocabs/user3553766=Mennatullah_Abdelkarim_Habashy @ 2346-832...
IAM VPC EC2 CloudWatch S3 Cloud9 Secrets Manager Systems Manager

Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together.

! Note on end-of-support. Use 'info' subcommand.
[ec2-user@ip-10-1-3-7 ~]$ 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/
--2024-10-03 22:49:41-- 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)... 52.92.145.82, 3.5.79.177, 52.92.133.178, ...
Connecting to aws-tc-largeobjects.s3.us-west-2.amazonaws.com (aws-tc-largeobjects.s3.us-west-2.amazonaws.com)|52.92.145.82|: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

2024-10-03 22:49:41 (181 MB/s) - '/opt/aws/amazon-cloudwatch-agent/bin/config.json' saved [2278/2278]

```

- To print out the file template:

```

aws | Services Search [Alt+S] N. Virginia vocabs/user3553766=Mennatullah_Abdelkarim_Habashy @ 2346-832...
IAM VPC EC2 CloudWatch S3 Cloud9 Secrets Manager Systems Manager

Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together.

[ec2-user@ip-10-1-3-7 ~]$ 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}"
    }
  }
}

i-01d363d20fd0c744b (EncryptedInstance)

```

- Start the CloudWatch agent and confirm that it is running:

```

aws | Services Search [Alt+S] N. Virginia vocabs/user3553766=Mennatullah_Abdelkarim_Habashy @ 2346-832...
IAM VPC EC2 CloudWatch S3 Cloud9 Secrets Manager Systems Manager

Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together.

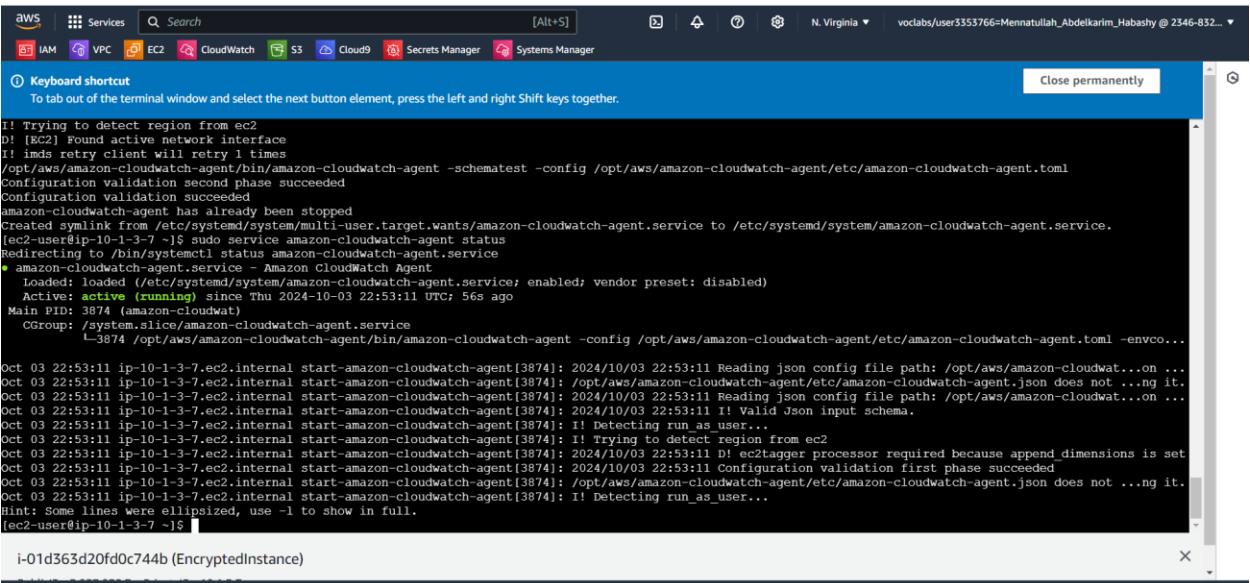
"metrics_collection_interval": 60
  },
  "statsd": {
    "metrics_aggregation_interval": 60,
    "metrics_collection_interval": 10,
    "service_address": ":8125"
  }
}

[ec2-user@ip-10-1-3-7 ~]$ 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 *****
! 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-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/file_config.json.tmp
Start configuration validation...
2024/10/03 22:53:10 Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/file_config.json.tmp ...
2024/10/03 22:53:10 ! Valid Json input schema.
2024/10/03 22:53:10 D! ec2tagger processor required because append_dimensions is set
2024/10/03 22:53:10 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.

i-01d363d20fd0c744b (EncryptedInstance)

```

- To verify the CloudWatch agent status:



The screenshot shows a terminal window within the AWS Management Console. The title bar indicates the user is in the N. Virginia region. The terminal displays the output of a command to check the CloudWatch agent status. The output shows the agent has been loaded and is active (running). It also lists several log entries from the CloudWatch agent's log file, including messages about configuration validation and various log levels (INFO, DEBUG) related to the agent's internal operations and network detection.

```

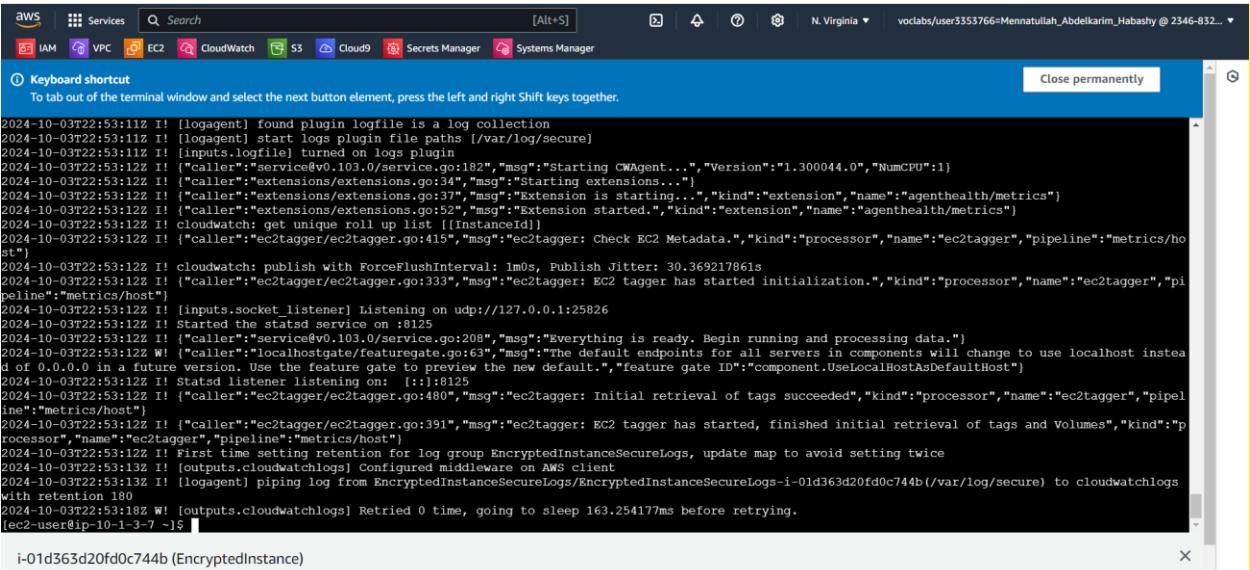
aws Services Search [Alt+S] N. Virginia voclabs/user3353766=Mennatullah_Abdelkarim_Habashy @ 2346-832...
[ IAM VPC EC2 CloudWatch S3 Cloud9 Secrets Manager Systems Manager ] Close permanently

Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together.

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-10-1-3-7 ~]$ 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 Thu 2024-10-03 22:53:11 UTC; 56s ago
      Main PID: 3874 (amazon-cloudwatch)
     CGroup: /system.slice/amazon-cloudwatch-agent.service
             └─3874 /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent -config /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.toml -envco...
Oct 03 22:53:11 ip-10-1-3-7.ec2.internal start-amazon-cloudwatch-agent[3874]: 2024/10/03 22:53:11 Reading json config file path: /opt/aws/amazon-cloudwat...
Oct 03 22:53:11 ip-10-1-3-7.ec2.internal start-amazon-cloudwatch-agent[3874]: /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json does not ...ng it...
Oct 03 22:53:11 ip-10-1-3-7.ec2.internal start-amazon-cloudwatch-agent[3874]: 2024/10/03 22:53:11 Reading json config file path: /opt/aws/amazon-cloudwat...
Oct 03 22:53:11 ip-10-1-3-7.ec2.internal start-amazon-cloudwatch-agent[3874]: 2024/10/03 22:53:11 ! Valid Json input schema.
Oct 03 22:53:11 ip-10-1-3-7.ec2.internal start-amazon-cloudwatch-agent[3874]: ! Detecting run_as_user...
Oct 03 22:53:11 ip-10-1-3-7.ec2.internal start-amazon-cloudwatch-agent[3874]: ! Trying to detect region from ec2
Oct 03 22:53:11 ip-10-1-3-7.ec2.internal start-amazon-cloudwatch-agent[3874]: 2024/10/03 22:53:11 D! ec2tagger processor required because append_dimensions is set
Oct 03 22:53:11 ip-10-1-3-7.ec2.internal start-amazon-cloudwatch-agent[3874]: 2024/10/03 22:53:11 configuration validation first phase succeeded
Oct 03 22:53:11 ip-10-1-3-7.ec2.internal start-amazon-cloudwatch-agent[3874]: /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json does not ...ng it...
Oct 03 22:53:11 ip-10-1-3-7.ec2.internal start-amazon-cloudwatch-agent[3874]: ! Detecting run_as_user...
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-10-1-3-7 ~]$ 
```

i-01d363d20fd0c744b (EncryptedInstance)

- To confirm that the CloudWatch agent is able to reach the CloudWatch service in my AWS account:



The screenshot shows a terminal window within the AWS Management Console. The title bar indicates the user is in the N. Virginia region. The terminal displays the output of a command to publish logs to CloudWatch Logs. The output shows the agent successfully publishing logs to the specified log group and stream. It includes messages about the log collection plugin, the ForceFlushInterval, and the initial retrieval of tags and volumes.

```

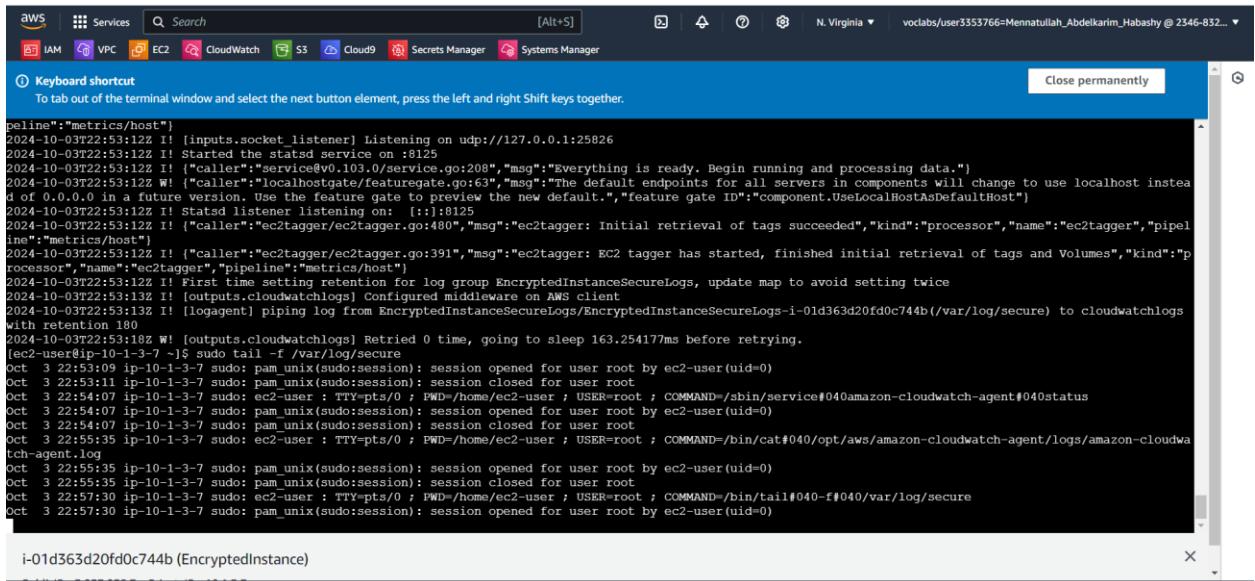
aws Services Search [Alt+S] N. Virginia voclabs/user3353766=Mennatullah_Abdelkarim_Habashy @ 2346-832...
[ IAM VPC EC2 CloudWatch S3 Cloud9 Secrets Manager Systems Manager ] Close permanently

Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together.

2024-10-03T22:53:11Z I! [logagent] found plugin logfile is a log collection
2024-10-03T22:53:11Z I! [logagent] start logs plugin file paths [/var/log/secure]
2024-10-03T22:53:11Z I! [inputs.logfile] turned on logs plugin
2024-10-03T22:53:11Z I! ["caller":"service@v0.103.0/service.go:182","msg":"Starting CWAgent...","Version":"1.300044.0.0","NumCPU":1}
2024-10-03T22:53:11Z I! ["caller":"extensions/extensions.go:34","msg":"Starting extensions..."]
2024-10-03T22:53:11Z I! ["caller":"extensions/extensions.go:37","msg":"Extension is starting...","kind":"extension","name":"agenthealth/metrics"}
2024-10-03T22:53:11Z I! ["caller":"extensions/extensions.go:52","msg":"Extension started.","kind":"extension","name":"agenthealth/metrics"]
2024-10-03T22:53:11Z I! cloudwatch: get unique roll up list {[InstanceId]}
2024-10-03T22:53:11Z I! ["caller":"ec2tagger/ec2tagger.go:15","msg":"ec2tagger: Check EC2 Metadata.","kind":"processor","name":"ec2tagger","pipeline":"metrics/host"}
2024-10-03T22:53:11Z I! cloudwatch: publish with ForceFlushInterval: 1m0s, Publish Jitter: 30.369217861s
2024-10-03T22:53:11Z I! ["caller":"ec2tagger/ec2tagger.go:333","msg":"ec2tagger: EC2 tagger has started initialization.","kind":"processor","name":"ec2tagger","pipeline":"metrics/host"}
2024-10-03T22:53:11Z I! [inputs.socket_listener] Listening on udp://127.0.0.1:25826
2024-10-03T22:53:11Z I! Started the statsd service on :8125
2024-10-03T22:53:11Z I! ["caller":"service@v0.103.0/service.go:208","msg":"Everything is ready. Begin running and processing data."]
2024-10-03T22:53:11Z W! ["caller":"localhostgate/featuregate.go:63","msg":"The default endpoints for all servers in components will change to use localhost instead of 0.0.0.0 in a future version. Use the feature gate to preview the new default.", "feature gate ID": "component.UseLocalHostAsDefaultHost"}
2024-10-03T22:53:11Z I! Statsd listener listening on: [::]:8125
2024-10-03T22:53:11Z I! ["caller":"ec2tagger/ec2tagger.go:480","msg":"ec2tagger: Initial retrieval of tags succeeded","kind":"processor","name":"ec2tagger","pipeline":"metrics/host"}
2024-10-03T22:53:11Z I! ["caller":"ec2tagger/ec2tagger.go:391","msg":"ec2tagger: EC2 tagger has started, finished initial retrieval of tags and volumes","kind":"processor","name":"ec2tagger","pipeline":"metrics/host"]
2024-10-03T22:53:11Z I! First time setting retention for log group EncryptedInstanceSecureLogs, update map to avoid setting twice
2024-10-03T22:53:11Z I! [outputs.cloudwatchlogs] Configured middleware on AWS client
2024-10-03T22:53:11Z I! [logagent] piping log from EncryptedInstanceSecureLogs/EncryptedInstanceSecureLogs-i-01d363d20fd0c744b(/var/log/secure) to cloudwatchlogs with retention 180
2024-10-03T22:53:11Z W! [outputs.cloudwatchlogs] Retried 0 time, going to sleep 163.254177ms before retrying.
[ec2-user@ip-10-1-3-7 ~]$ 
```

i-01d363d20fd0c744b (EncryptedInstance)

- To actively tail the `/var/log/secure` file:



The screenshot shows a terminal window within the AWS CloudWatch interface. The title bar indicates the session is for user `voclabs/user3353766=Mennatullah_Abdelkarim_Habashy @ 2346-832...`. The terminal displays the contents of the `/var/log/secure` log file. The logs show various system events, including the start of the statsd service, configuration of the EC2 tagger, and user sessions. A message at the bottom of the log indicates that the log group `EncryptedInstanceSecureLogs` has been updated to avoid setting twice.

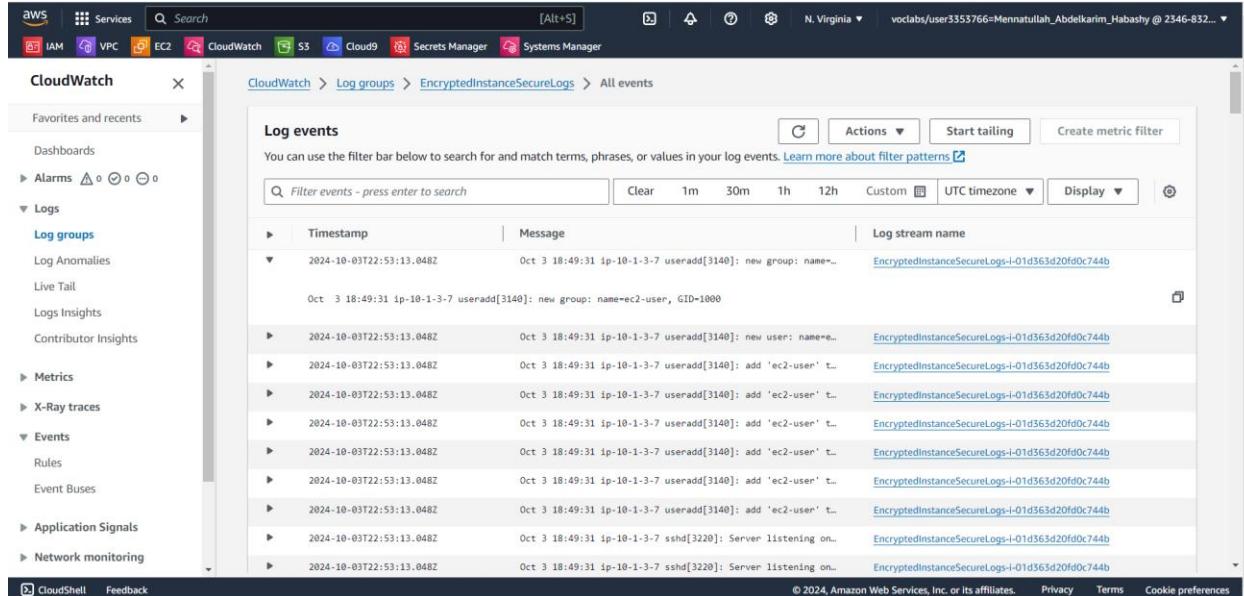
```

reline": "metrics/host"
2024-10-03T22:53:12Z I! [inputs.socket.listener] Listening on udp://127.0.0.1:25826
2024-10-03T22:53:12Z I! Started the statsd service on :8125
2024-10-03T22:53:12Z I! {"caller":"service@v0.103.0/service-go:208","msg":"Everything is ready. Begin running and processing data."}
2024-10-03T22:53:12Z W! {"caller":"localhostfeature/featuregate.go:63","msg":"The default endpoints for all servers in components will change to use localhost instead of 0.0.0.0 in a future version. Use the feature gate to preview the new default.","feature gate ID":"component.UseLocalHostAsDefaultHost"}
2024-10-03T22:53:12Z I! Statsd listener listening on: [:]:8125
2024-10-03T22:53:12Z I! {"caller":"ec2tagger/ec2tagger.go:80","msg":"ec2tagger: Initial retrieval of tags succeeded","kind":"processor","name":"ec2tagger","pipeline": "metrics/host"}
2024-10-03T22:53:12Z I! {"caller":"ec2tagger/ec2tagger.go:391","msg":"ec2tagger: EC2 tagger has started, finished initial retrieval of tags and Volumes","kind":"processor","name":"ec2tagger","pipeline": "metrics/host"}
2024-10-03T22:53:12Z I! first time setting retention for log group EncryptedInstanceSecureLogs, update map to avoid setting twice
2024-10-03T22:53:12Z I! [outputs.cloudwatchlogs] Configured middleware on AWS client
2024-10-03T22:53:12Z I! [logagent] piping log from EncryptedInstanceSecureLogs/EncryptedInstanceSecureLogs-i-01d363d20fd0c744b(/var/log/secure) to cloudwatchlogs with retention 180
2024-10-03T22:53:12Z W! [outputs.cloudwatchlogs] Retried 0 time, going to sleep 163.254177ms before retrying.
[ec2-user@ip-10-1-3-7 ~]$ sudo tail -f /var/log/secure
Oct 3 22:53:09 ip-10-1-3-7 sudo: pam_unix(sudo:session): session opened for user root by ec2-user(uid=0)
Oct 3 22:53:11 ip-10-1-3-7 sudo: pam_unix(sudo:session): session closed for user root
Oct 3 22:54:07 ip-10-1-3-7 sudo: ec2-user : TTY-pts/0 ; PWD=/home/ec2-user ; USER=root ; COMMAND=/sbin/service#040amazon-cloudwatch-agent#040status
Oct 3 22:54:07 ip-10-1-3-7 sudo: pam_unix(sudo:session): session opened for user root by ec2-user(uid=0)
Oct 3 22:55:38 ip-10-1-3-7 sudo: pam_unix(sudo:session): session closed for user root
Oct 3 22:55:38 ip-10-1-3-7 sudo: pam_unix(sudo:session): session opened for user root by ec2-user(uid=0)
Oct 3 22:57:30 ip-10-1-3-7 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 3 22:57:30 ip-10-1-3-7 sudo: pam_unix(sudo:session): session opened for user root by ec2-user(uid=0)
Oct 3 22:57:30 ip-10-1-3-7 sudo: pam_unix(sudo:session): session closed for user root
Oct 3 22:57:30 ip-10-1-3-7 sudo: pam_unix(sudo:session): session opened for user root by ec2-user(uid=0)

i-01d363d20fd0c744b (EncryptedInstance)

```

-Verify that the secure logs are being written to the CloudWatch log group:



The screenshot shows the AWS CloudWatch Log Events page. The left sidebar is collapsed, and the main area shows the `Log events` section for the `EncryptedInstanceSecureLogs` log group. The log stream name is `EncryptedInstanceSecureLogs-i-01d363d20fd0c744b`. The log events table lists several entries, all timestamped at `Oct 3 18:49:31` and originating from `ip-10-1-3-7`. The messages describe user additions to a group named `ec2-user` with `GID=1000`.

Timestamp	Message	Log stream name
Oct 3 18:49:31 ip-10-1-3-7 useradd[3140]:	new group: name=ec2-user, GID=1000	EncryptedInstanceSecureLogs-i-01d363d20fd0c744b
Oct 3 18:49:31 ip-10-1-3-7 useradd[3140]:	new user: name=ec2-user, uid=1000	EncryptedInstanceSecureLogs-i-01d363d20fd0c744b
Oct 3 18:49:31 ip-10-1-3-7 useradd[3140]:	add 'ec2-user' to group 'ec2-user'	EncryptedInstanceSecureLogs-i-01d363d20fd0c744b
Oct 3 18:49:31 ip-10-1-3-7 useradd[3140]:	add 'ec2-user' to group 'root'	EncryptedInstanceSecureLogs-i-01d363d20fd0c744b
Oct 3 18:49:31 ip-10-1-3-7 useradd[3140]:	add 'ec2-user' to group 'sys'	EncryptedInstanceSecureLogs-i-01d363d20fd0c744b
Oct 3 18:49:31 ip-10-1-3-7 useradd[3140]:	add 'ec2-user' to group 'kmem'	EncryptedInstanceSecureLogs-i-01d363d20fd0c744b
Oct 3 18:49:31 ip-10-1-3-7 useradd[3140]:	add 'ec2-user' to group 'dialout'	EncryptedInstanceSecureLogs-i-01d363d20fd0c744b
Oct 3 18:49:31 ip-10-1-3-7 useradd[3140]:	add 'ec2-user' to group 'audio'	EncryptedInstanceSecureLogs-i-01d363d20fd0c744b
Oct 3 18:49:31 ip-10-1-3-7 sshd[3220]:	Server listening on 0.0.0.0 port 22 (0.0.0.0).	EncryptedInstanceSecureLogs-i-01d363d20fd0c744b

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

In this task, I will create a CloudWatch alarm to notify these team members when such an incident occurs.

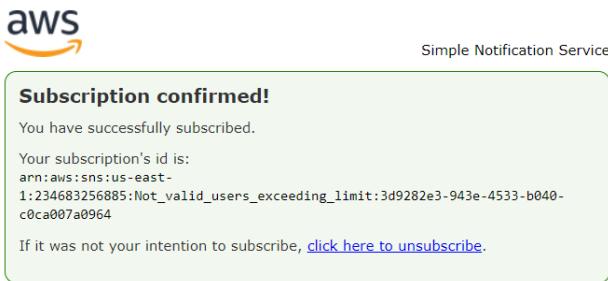
-create a metric filter:

The screenshot shows the AWS CloudWatch Metrics Filter creation interface. At the top, a green banner indicates "Metric filter 'Not valid users' has been created." Below this, the navigation path is "CloudWatch > Log groups > EncryptedInstanceSecureLogs". The main section is titled "EncryptedInstanceSecureLogs" and displays "Log group details". Key metrics include ARN (arn:aws:logs:us-east-1:234683256885:log-group:EncryptedInstanceSecureLogs:*) and Metric filters (0). The bottom navigation bar includes tabs for Log streams, Tags, Anomaly detection, Metric filters (which is selected), Subscription filters, Contributor Insights, and Data protection.

-Create a CloudWatch alarm from the metric filter:

The screenshot shows the AWS CloudWatch Alarms page. A green banner at the top says "Successfully created alarm Not valid users exceeding limit on EncryptedInstance." The main area lists "Alarms (1)" with a single entry: "Not valid users exceeding limit on EncryptedInstance". The alarm details show it was created on "2024-10-03 23:54:07" and has the condition "NotValidUsers >= 5 for 1 datapoints within 1 day". The "Actions enabled" status is shown as "Actions enabled". The left sidebar shows the navigation menu for CloudWatch, including Logs, Metrics, and Alarms.

-Confirm subscription link.



-in the latest log stream for the *EncryptedInstanceSecureLogs* log group, filter the log events for Invalid user:

The screenshot shows the AWS CloudWatch Logs interface. The left sidebar is collapsed. The main area displays the 'Log events' for the 'EncryptedInstanceSecureLogs' log group. A search bar at the top is set to 'Invalid user'. Below the search bar, there are time range controls (Clear, 1m, 30m, 1h, 12h, Custom, UTC timezone). The log entries listed are:

Timestamp	Message
Oct 3 19:13:46 ip-10-1-3-7 sshd[3974]	: Invalid user user from 35.192.211.207 port 55214
Oct 3 19:13:46 ip-10-1-3-7 sshd[3974]	: Invalid user user from 35.192.211.207 port 55264
Oct 3 19:13:46 ip-10-1-3-7 sshd[3978]	: Invalid user user from 35.192.211.207 port 55310
Oct 3 19:13:46 ip-10-1-3-7 sshd[3980]	: Invalid user user from 35.192.211.207 port 55390
Oct 3 19:13:47 ip-10-1-3-7 sshd[3982]	: Invalid user user from 35.192.211.207 port 55434
Oct 3 19:13:47 ip-10-1-3-7 sshd[3984]	: Invalid user user from 35.192.211.207 port 55458
Oct 3 19:13:47 ip-10-1-3-7 sshd[3986]	: Invalid user user from 35.192.211.207 port 55500
Oct 3 19:13:47 ip-10-1-3-7 sshd[3988]	: Invalid user user from 35.192.211.207 port 55556
Oct 3 19:13:47 ip-10-1-3-7 sshd[3990]	: Invalid user user from 35.192.211.207 port 55620
Oct 3 19:13:48 ip-10-1-3-7 sshd[3992]	: Invalid user user from 35.192.211.207 port 55664

-confirm that the *Invalid users* alarm has a state of *In alarm*:

The screenshot shows the AWS CloudWatch Alarms page. The left sidebar has sections for Dashboards, Alarms (with 1 in alarm), All alarms, Billing, Logs, Metrics, X-Ray traces, Events, and Rules. The main area shows a table for 'Alarms (1)'. The table has columns for Name, State, Last state update (Local), Conditions, and Actions. One row is listed: 'Not valid users exceeding limit on EncryptedInstance' with a green 'OK' icon, updated on 2024-10-04 02:58:04, and the condition 'NotValidUsers >= 5 for 1 datapoints within 1 day'. The 'Actions enabled' button is also visible.

Checking email to confirm that I received a notification that the alarm threshold was crossed:

The screenshot shows an email from AWS Notifications. The subject is "OK: 'Not valid users exceeding limit on EncryptedInstance' in US East (N. Virginia)". The email body contains the following text:
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 OK state, because "Threshold Crossed: 1 out of the last 1 datapoints [0.0 (02/10/24 23:58:00)] was not greater than or equal to the threshold (5.0) (minimum 1 datapoint for ALARM -> OK transition)." at "Thursday 03 October, 2024 23:58:04 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 -> OK
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [0.0 (02/10/24 23:58:00)] was not greater than or equal to the threshold (5.0) (minimum 1 datapoint for ALARM -> OK transition).
- Timestamp: Thursday 03 October, 2024 23:58:04 UTC
- AWS Account: 234683256885
- Alarm Arn: arn:aws:cloudwatch:us-east-1:234683256885: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

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

In this task I'm challenged to use AWS Config to report on whether object logging is configured on the S3 buckets and configure an automation script to remediate noncompliance.

-Create a new S3 bucket named compliance-bucket-unique-ID:

The screenshot shows the AWS S3 console with a green success message at the top: "Successfully created bucket 'compliance-bucket-0e1570123fa91ff36'". Below this, the "General purpose buckets" section lists eight buckets. The table includes columns for Name, AWS Region, IAM Access Analyzer, and Creation date. The newly created bucket is listed as the eighth item.

Name	AWS Region	IAM Access Analyzer	Creation date
athena-results-9876nhgg	US East (N. Virginia) us-east-1	View analyzer for us-east-1	September 29, 2024, 19:58:44 (UTC+03:00)
athena3234	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 4, 2024, 01:12:04 (UTC+03:00)
aws-config-0e1570123fa91ff36	US East (N. Virginia) us-east-1	View analyzer for us-east-1	September 24, 2024, 19:04:52 (UTC+03:00)
cloudtrail-logs-0e1570123fa91ff36	US East (N. Virginia) us-east-1	View analyzer for us-east-1	September 24, 2024, 19:04:52 (UTC+03:00)
compliance-bucket-0e1570123fa91ff36	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 5, 2024, 13:30:45 (UTC+03:00)
data-bucket0e1570123fa91ff36	US East (N. Virginia) us-east-1	View analyzer for us-east-1	September 24, 2024, 19:13:54 (UTC+03:00)
s3-inventory-0e1570123fa91ff36	US East (N. Virginia) us-east-1	View analyzer for us-east-1	September 24, 2024, 19:04:52 (UTC+03:00)
s3-objects-access-log-0e1570123fa91ff36	US East (N. Virginia) us-east-1	View analyzer for us-east-1	September 24, 2024, 19:04:53 (UTC+03:00)

-Enable object ownership on the *s3-objects-access-log* bucket:

The screenshot shows the "Permissions overview" section of the S3 bucket settings. It includes an "Access findings" panel with a link to "How IAM analyzer findings work". Below this is the "Block public access (bucket settings)" section, which has "Block all public access" turned "On". The "Bucket policy" section is also visible, with a note about JSON policies and a "Learn more" link.

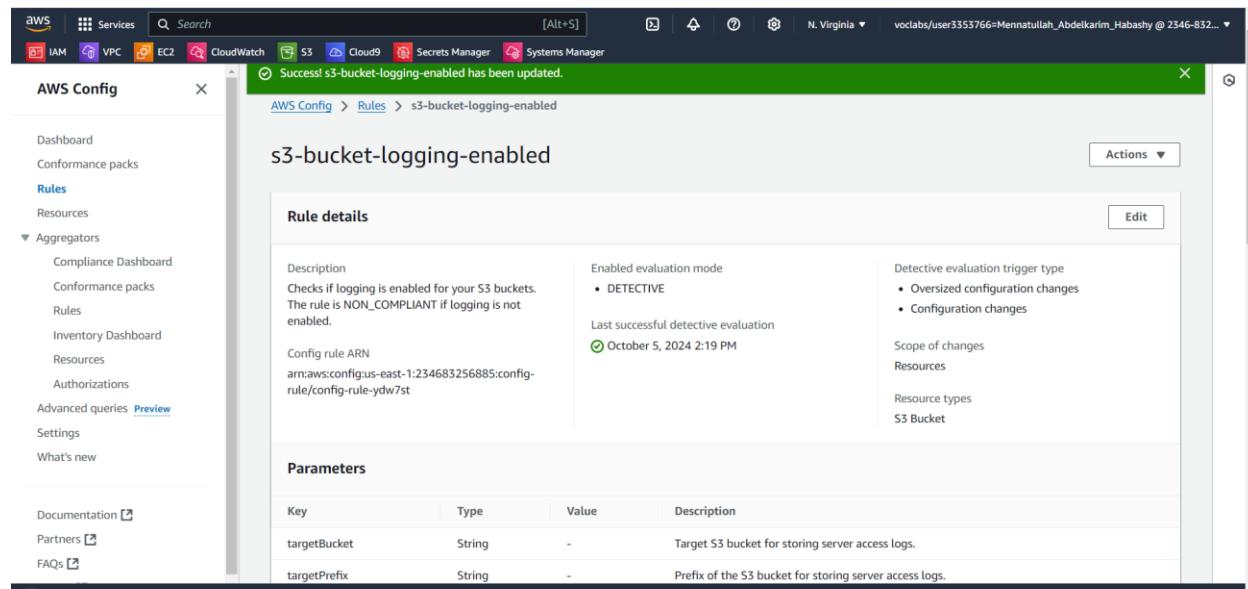
-Set up AWS Config:

The screenshot shows the AWS Config Dashboard. On the left, there's a sidebar with links like Dashboard, Conformance packs, Rules, Resources, Aggregators, Documentation, Partners, and FAQs. The main area has a "Conformance Packs by Compliance Score" section which says "No conformance packs deployed. Try deploying a new conformance pack." Below it is a "Compliance status" section showing 0 Noncompliant rule(s) and 0 Compliant rule(s) for both Rules and Resources. To the right is a "AWS Config usage metrics" section with two charts: "Configuration Items Recorded" and "Configuration Recorder Insuff...". Both charts show no data available.

-Confirm that the *s3-bucket-logging-enabled* rule is finding resources that are in scope. Also, confirm that the *compliance-bucket* is flagged as noncompliant:

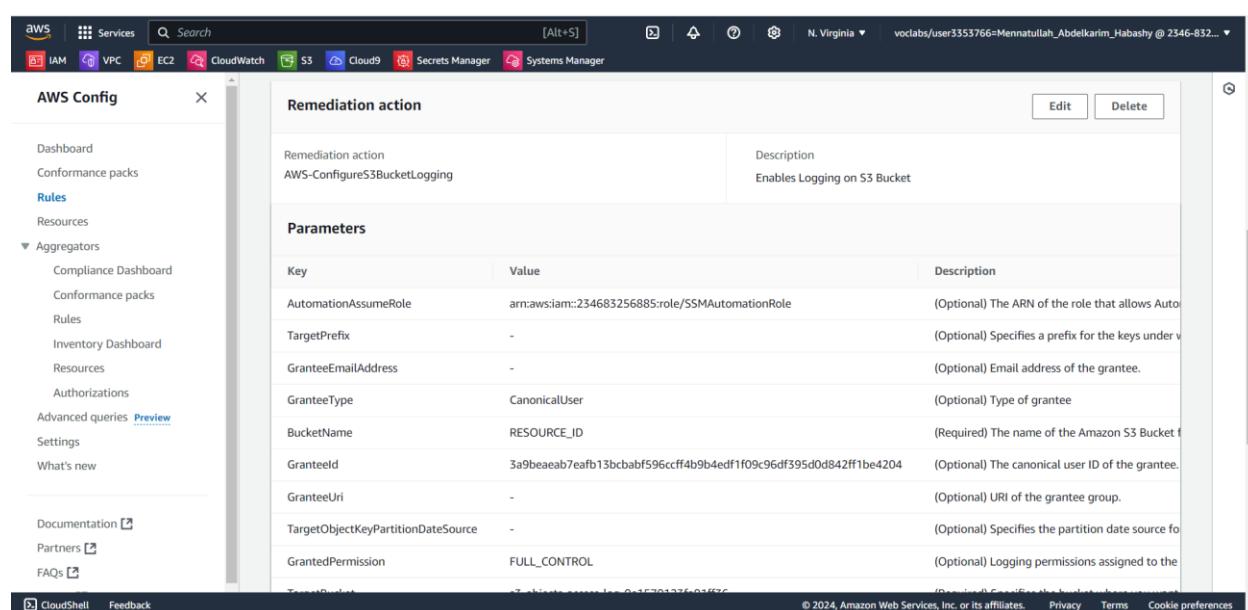
The screenshot shows the AWS Config Rules page. The sidebar includes links for Dashboard, Conformance packs, Rules, Resources, Aggregators, Documentation, Partners, and FAQs. Under the "Rules" section, there's a table for the "s3-bucket-logging-enabled" rule with two parameters: "targetBucket" (String, Value: -, Description: Target S3 bucket for storing server access logs) and "targetPrefix" (String, Value: -, Description: Prefix of the S3 bucket for storing server access logs). Below this is a "Resources in scope" table with a dropdown set to "Noncompliant". It lists several S3 buckets: athena-results-9876nhgg, athena3234, aws-config-Qe1570123fa91ff36, cloudtrail-logs-0e1570123fa91ff36, compliance-bucket-0e1570123fa91ff36, and s3-inventory-0e1570123fa91ff36. All of these are marked as "Noncompliant".

-Configure manual remediation settings for the *s3-bucket-logging-enable* rule:



The screenshot shows the AWS Config Rule Details page for the rule "s3-bucket-logging-enabled". The rule description is: "Checks if logging is enabled for your S3 buckets. The rule is NON_COMPLIANT if logging is not enabled." The enabled evaluation mode is "DETECTIVE". The last successful detection was on October 5, 2024, at 2:19 PM. The config rule ARN is arn:aws:config:us-east-1:234683256885:config-rule/config-rule-ydw7st. The scope of changes is "Resources" under "S3 Bucket".

Key	Type	Value	Description
targetBucket	String	-	Target S3 bucket for storing server access logs.
targetPrefix	String	-	Prefix of the S3 bucket for storing server access logs.



The screenshot shows the AWS Config Remediation Action page for the action "AWS-ConfigureS3BucketLogging". The remediation action description is "Enables Logging on S3 Bucket". The parameters for the action are listed below:

Key	Value	Description
AutomationAssumeRole	arn:aws:iam::234683256885:role/SSMAutomationRole	(Optional) The ARN of the role that allows Automation to assume it.
TargetPrefix	-	(Optional) Specifies a prefix for the keys under which logs are stored.
GranteeEmailAddress	-	(Optional) Email address of the grantee.
GranteeType	CanonicalUser	(Optional) Type of grantee.
BucketName	RESOURCE_ID	(Required) The name of the Amazon S3 Bucket to log to.
GranteeId	3a9beaeab7eafb13bcbabf596ccff4b9b4edf1f09c96df395d0d842ff1be4204	(Optional) The canonical user ID of the grantee.
GranteeUri	-	(Optional) URI of the grantee group.
TargetObjectKeyPartitionDataSource	-	(Optional) Specifies the partition date source for the log file.
GrantedPermission	FULL_CONTROL	(Optional) Logging permissions assigned to the grantee.

-Invoke the AWS Config remediation action:

The screenshot shows the AWS Config console with a configuration rule applied to an S3 bucket. The rule specifies a target object key prefix of 's3-objects-access-log-0e1570123fa91ff36'. The 'Resources in scope' table lists several S3 buckets, all of which are marked as 'Noncompliant' except for one which has a green checkmark indicating the action was executed successfully.

ID	Type	Status	Annotation	Compliance
athena-results-9876nhgg	S3 Bucket	-	-	⚠ Noncompliant
athena3234	S3 Bucket	-	-	⚠ Noncompliant
aws-config-0e1570123fa91ff36	S3 Bucket	-	-	⚠ Noncompliant
cloudtrail-logs-0e1570123fa91ff36	S3 Bucket	-	-	⚠ Noncompliant
compliance-bucket-0e1570123fa91ff36	S3 Bucket	✓ Action executed successfully	-	⚠ Noncompliant
s3-inventory-0e1570123fa91ff36	S3 Bucket	-	-	⚠ Noncompliant

-Confirm that server access logging is now enabled on the *compliance-bucket*:

The screenshot shows the Amazon S3 console for a specific bucket. Under the 'Server access logging' section, it is configured to log requests to CloudWatch Logs with a destination bucket of 's3://s3-objects-access-log-0e1570123fa91ff36'. There are also sections for AWS CloudTrail data events and event notifications.

Cost assessment for monitoring and logging:

My Estimate [Edit](#) [Share](#)

Estimate summary [Info](#)

Upfront cost 0.00 USD	Monthly cost 219.31 USD	Total 12 months cost 2,631.72 USD Includes upfront cost	Get started for free	Contact Sales
--------------------------	----------------------------	----------------------------------------------------------------------	--------------------------------------	-------------------------------

My Estimate

Duplicate Delete Move to Create group Add support Add service

<input type="checkbox"/> Service Name	Status	Upfront cost	Monthly cost	Description	Region	Config Summary
AWS CloudTrail	-	0.00 USD	180.57 USD	-	US East (N. Virginia)	Management events ...
Amazon CloudWatch	-	0.00 USD	38.69 USD	-	US East (N. Virginia)	Number of Metrics (in...)
AWS Config	-	0.00 USD	0.05 USD	-	US East (N. Virginia)	Number of Continuou...

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format. [Don't show again](#) [Save As...](#)

GET GENUINE OFFICE Your license isn't genuine, and you may be a victim of software counterfeiting. Avoid interruption and keep your files safe with genuine Office today. [Get genuine Office](#) [Learn more](#)

A1 B C D E F G H I J K L M N O P Q R S T U V W

1	Estimate summary														
2	Upfront	cc Monthly	c Total 12 m	Currency											
3	0	219.31	2631.72	USD											
4	* Includes upfront cost														
5															
6															
7	Detailed Estimate														
8	Group hier	Region	Description	Service	Upfront	Monthly	First 12 m	Currency	Status	Configuration summary					
9	My Estimate	US East (N. Virginia)	AWS Cloud	0	180.57	2166.84	USD			Management events units (millions), Write management trails (1), Read management trails (1), Data events units (millions), 53 trails (1), Lambda trails (1), Insig					
10	My Estimate	US East (N. Virginia)	Amazon Cl	0	38.6853	464.22	USD			Number of Metrics (includes detailed and custom metrics) (50), GetMetricData: Number of metrics requested (10), GetMetricWidgetImage: Number of metri					
11	My Estimate	US East (N. Virginia)	AWS Confi	0	0.052	0.62	USD			Number of Continuous Configuration items recorded (3), Number of Periodic Configuration items recorded (3), Number of Config rule evaluations (3), Numbe					
12															
13															
14															
15	Acknowledgement														
16	* 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.														
17															
18															
19															
20															
21															
22															
23															
24															
25															

My Estimate (4)

And Finally this is My badge from AWS Academy ,

<https://www.credly.com/go/4ITNsuea>

