

# **Project in AWS Practice Lab**

## **Work with AWS VPC Flow Logs for Network Monitoring**

**Andra-Diana Popescu**

**2025**

## **ABOUT THIS LAB**

Monitoring network traffic is a critical component of security best practices to meet compliance requirements, investigate security incidents, track key metrics, and configure automated notifications. AWS VPC Flow Logs captures information about the IP traffic going to and from network interfaces in your VPC. In this hands-on lab, we will set up and use VPC Flow Logs published to Amazon CloudWatch, create custom metrics and alerts based on the CloudWatch logs to understand trends and receive notifications for potential security issues, and use Amazon Athena to query and analyze VPC flow logs stored in S3.

## **LEARNING OBJECTIVES**

- Create a CloudWatch Log Group and a VPC Flow Log to CloudWatch
- Create CloudWatch Filters and Alerts
- Use CloudWatch Logs Insights
- Analyze VPC Flow Logs Data in Athena

### **AWS Documentation about CloudWatch, VPC and Athena:**

<https://docs.aws.amazon.com/cloudwatch/#amazon-cloudwatch>

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html>

<https://aws.amazon.com/athena/faqs/?nc=sn&loc=6#topic-0>

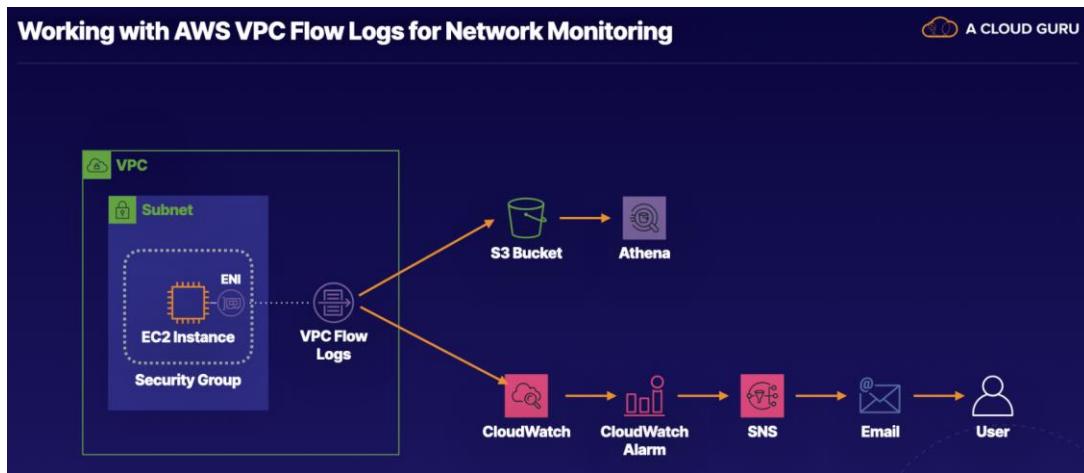
<https://aws.amazon.com/vpc/faqs/#topic-0>

**Source:** <https://learn.acloud.guru/course/certified-solutions-architect-associate/>

## Table of Contents

Lab Diagrams.....	4
Log in to your AWS account .....	5
1. Create a CloudWatch Log Group and VPC Flow Logs to CloudWatch.....	5
1.1. Create a VPC Flow Log to S3.....	5
1.2. Create the CloudWatch Log Group and VPC Flow Log .....	8
1.3. Generate Network Traffic .....	13
2. Create CloudWatch Filters and Alerts .....	17
2.1. Create a CloudWatch Log Metric Filter.....	17
2.2. Create an Alarm Based on the Metric Filter .....	21
2.3. Generate Traffic for Alerts.....	24
3. Use CloudWatch Logs Insights.....	28
4. Analyze VPC Flow Logs Data in Athena .....	29
4.1. Create the Athena Table.....	29
4.2. Create Partitions and Analyze the Data .....	31

## Lab Diagrams



We have the AWS account in **us-east-1** Region. Monitoring network traffic and alerting suspicious activity are critical components of a security strategy. In this lab, we'll be setting up AWS VPC Flow Logs to capture information about the IP traffic to network interfaces in your VPC. There are many uses for this kind of data, including alerting failed SSH attempts, determining the source IP addresses that are port probes, and performing ad hoc queries.

This lab is already pre-provisioned with the following resources: a VPC, one public subnet, an EC2 instance, and several security groups. In order to see the records in our VPC Flow Logs, we'll generate some network traffic to the EC2 instance later in the lab. To accomplish this, we'll be changing the EC2 instance's security groups to control whether SSH access is allowed or denied to the instance.

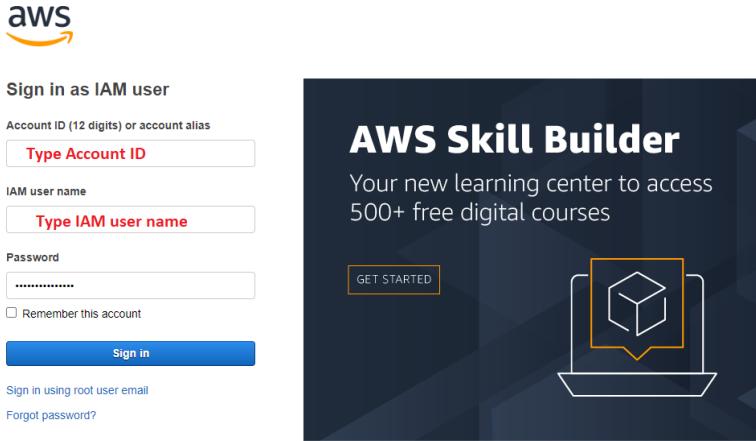
Then, we'll be using our terminal to attempt to SSH into the instance. If the security group allows us SSH access, then the VPC Flow Logs will show a record from the source IP address on port 22 as accepted. If the security group does not allow SSH access, you'll see a record that shows as rejected. These VPC Flow Logs can be configured at the VPC, subnet, or network interface level. We'll configure the flow logs at the VPC level, which means that the flow log data will be tracked for every network interface in the VPC.

By the end of this lab, you'll know how to set up and use VPC Flow Logs, publish to both CloudWatch and an S3 bucket. You'll also know how to create custom metrics and alerts based on CloudWatch logs to receive notifications for potential security issues. Additionally, you'll know how to perform queries against flow logs using CloudWatch Insights.

CloudWatch is great for generating metrics for dashboards and alerts, but S3 offers more cost-effective, long-term storage. For the logs stored in S3, you'll create an Amazon Athena table and run a query based on the data stored on that table.

Athena is a serverless interactive query service to query data in S3 using familiar SQL statements. It's important to note that the VPC Flow Logs can take between 5 and 15 minutes to show up in our S3 or CloudWatch after the flow logs have been created.

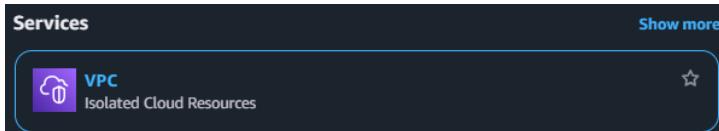
## Log in to your AWS account



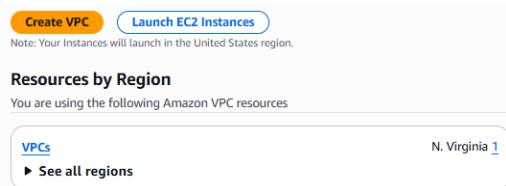
## 1. Create a CloudWatch Log Group and VPC Flow Logs to CloudWatch

### 1.1. Create a VPC Flow Log to S3

- Once you are logged into the AWS Management Console, navigate to **VPC**.



- In the VPC dashboard, select the **VPCs** card. You should see an **A Cloud Guru** VPC pre-provisioned for the lab.



- Check the checkbox next to the **A Cloud Guru** VPC.
- Toward the bottom of the screen, select the **Flow logs** tab.
- On the right, click **Create flow log**.

6. Fill in the flow log details:

- Name:** You can leave this field blank.
- Filter:** Ensure that **All** is selected.
- Maximum aggregation interval:** Select **1 minute**.
- Destination:** Select **Send to an Amazon S3 bucket**.

7. Get the S3 bucket ARN:

- In a new browser tab, navigate to **S3**.
- Select the radio button next to the provided bucket.
- Click **Copy ARN**.

The screenshot shows the Amazon S3 console. On the left, there's a sidebar titled 'Amazon S3' with various bucket categories. The main area is titled 'Account snapshot - updated every 24 hours' and shows 'General purpose buckets (1/1)'. A single bucket, 'cfst-vpcflowlogsbucket-szdpf5udf7ju', is listed. To the right of the bucket name are buttons for 'Bucket ARN copied', 'Copy ARN' (which is highlighted with a red box), 'Empty', 'Delete', and 'Create bucket'.

8. Navigate back to the **VPC Management Console** tab and fill in the rest of the flow log details:

- S3 bucket ARN:** In the text field, paste your copied S3 bucket ARN.
- Log record format:** Ensure that **AWS default format** is selected.

This screenshot shows the 'Log record format' configuration in the VPC Management Console. It includes a note about a resource-based policy being created and attached to the target bucket. The 'AWS default format' radio button is selected and highlighted with a red box.

9. Leave the other fields as the default settings and click **Create flow log**. Your flow log is created.

10. From the **Your VPCs** page, select the **Flow logs** tab.

11. Review the flow log details and verify that it shows an **Active** status.

This screenshot shows the 'Your VPCs' page and the 'Flow logs' tab. The flow log 'fl-091' is selected, and its status is shown as 'Active' (highlighted with a red box). The 'Status' column for this flow log also has a green checkmark and the word 'Active'.

12. Navigate back to the **S3 Management Console** tab.

13. Select your bucket name, and then select the **Permissions** tab.

```

{
    "Version": "2012-10-17",
    "Id": "AWSLogDeliveryWrite20150319",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite1",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:s3:::cfst-3029-  
-vpcflowlogsbucket-ukrsptks26yh/AWSLogs/*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012",
                    "s3:x-amz-acl": "bucket-owner-full-control"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:logs:us-east-1:  
*:*"
                }
            }
        }
    ]
}

```

14. Review the bucket policy and note that it is modified automatically by AWS when you create flow logs so that the flow logs can write to the bucket.

Note: It can take between 5–15 minutes for flow logs to appear. You can continue working through the other lab objectives while you wait for the flow logs to populate.

```

{
    "Version": "2012-10-17",
    "Id": "AWSLogDeliveryWrite20150319",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite1",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:s3:::cfst-3029-  
-vpcflowlogsbucket-ukrsptks26yh/AWSLogs/*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012",
                    "s3:x-amz-acl": "bucket-owner-full-control"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:logs:us-east-1:  
*:*"
                }
            }
        }
    ]
}

```

## 1.2. Create the CloudWatch Log Group and VPC Flow Log

1. In a new browser tab, navigate to **CloudWatch**.

2. In the CloudWatch sidebar menu, navigate to **Logs** and select **Log groups**. A log group is a container for our logs.
3. Click **Create log group**.

**CloudWatch > Log groups**

**Log groups (0)**

By default, we only load up to 10000 log groups.

Filter log groups or try pattern search  Exact match

Log group	Log class	Anomaly d...	Data protection	Sensitive data count	Retention
No log groups					

You have not created any log groups.

Read more about Logs

Create log group

4. In the **Log group name** field, enter **VPCFlowLogs**.

5. Click **Create**.

**CloudWatch > Log groups > Create log group**

**Create log group**

**Log group details** Info

CloudWatch Logs offers two log classes: Standard and Infrequent Access. [Learn more about the features offered by each log class.](#)

**Log group name**

**Retention setting**

**Log class** Info

**KMS key ARN - optional**

**Tags**

A tag is a label that you assign to an Amazon Web Services resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your Amazon Web Services costs.

No tags are associated with this log group.

Add new tag

You can add up to 50 more tag(s).

Cancel **Create**

6. Navigate back to the **VPC Management Console** tab and ensure the **Flow logs** tab is still selected.

7. On the right, click **Create flow log**.

8. Fill in the flow log details:

- Name:** You can leave this field blank.
- Filter:** Ensure that **All** is selected.
- Maximum aggregation interval:** Select **1 minute**.
- Destination:** Ensure that **Send to CloudWatch Logs** is selected.
- Destination log group:** Click into the field and select your **VPCFlowLogs** log group.
- IAM role:** Use the dropdown to select the **DeliverVPCFlowLogsRole** role. This IAM role gives the VPC Flow Logs service permission to write to your CloudWatch Logs group.
- Log record format:** Ensure that **AWS default format** is selected.

VPC > Your VPCs > Create flow log

### Create flow log Info

Flow logs can capture IP traffic flow information for the network interfaces associated with your resources. You can create multiple flow logs to send traffic to different destinations.

**Selected resources** Info

Name	Resource ID	State
A Cloud Guru	vpc-1	Available

**Flow log settings**

**Name - optional**

**Filter**  
The type of traffic to capture (accepted traffic only, rejected traffic only, or all traffic).

Accept  
 Reject  
 All

**Maximum aggregation interval** Info  
The maximum interval of time during which a flow of packets is captured and aggregated into a flow log record.

10 minutes  
 1 minute

**Destination**  
The destination to which to publish the flow log data.

Send to CloudWatch Logs  
 Send to an Amazon S3 bucket  
 Send to Amazon Data Firehose in the same account  
 Send to Amazon Data Firehose in a different account

**Destination log group** Info  
The name of an existing log group or the name of a new log group that will be created when you create this flow log. A new log stream is created for each monitored network interface.

VPC > Your VPCs > Create flow log

**Destination log group** [Info](#)  
The name of an existing log group or the name of a new log group that will be created when you create this flow log. A new log stream is created for each monitored network interface.  
 [X](#) [Copy](#)

**Service access**  
VPC flow logs require permissions to create log groups and publish events in CloudWatch.

Use an existing service role  
 Create and use a new service role

**Service role** [Info](#)  
The IAM role that has permission to publish to the Amazon CloudWatch log group.  
 [View this service role in the IAM console](#) [Copy](#)

**Log record format**  
Specify the fields to include in the flow log record.  
 AWS default format  
 Custom format

**Additional metadata**  
Include additional metadata to AWS default log record format.  
 Include Amazon ECS metadata

**Format preview**  
\${version} \${account-id} \${interface-id} \${srcaddr} \${dstaddr} \${srcport} \${dstport} \${protocol} \${packets} \${bytes} \${start} \${end} \${action} \${log-status}

[Add tag](#)  
You can add 50 more tags

[Cancel](#) [Create flow log](#)

IAM > Roles > cfst-3029- [DeliverVPCFlowLogsRole-](#) [Edit policy](#)

Step 1 **Modify permissions in flowlogs-policy** [Info](#)  
Step 2 [Review and save](#)

**Policy editor**

```

1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Action": [
6          "ec2:DeleteFlowLogs",
7          "ec2:CreateFlowLogs",
8          "ec2:DescribeFlowLogs",
9          "logs:CreateLogGroup",
10         "logs:CreateLogStream",
11         "logs:PutLogEvents",
12         "logs:DescribeLogGroups",
13         "logs:DescribeLogStreams",
14         "ec2:CreateFlowLogs"
15       ],
16       "Resource": "*",
17       "Effect": "Allow"
18     }
19   ]
20 }
```

The screenshot shows the AWS IAM Policy Editor. On the left, there are two steps: "Step 1: Modify permissions in flowlogs-policy" and "Step 2: Review and save". Step 2 is currently selected. A large callout box highlights the "Review and save" button and the instruction "Review the permissions, specify details, and tags." Below this, a section titled "Permissions defined in this policy" shows two services with limited permissions: CloudWatch Logs and EC2.

9. Click **Create flow log**. Your flow log is created.

10. From the **Your VPCs** page, ensure the **Flow logs** tab is selected.

11. Review the flow log details and verify that the new flow log shows an **Active** state.

The screenshot shows the AWS VPC Flow Logs page. At the top, a success message says "Successfully created flow log for vpc-0e...". The main table lists two flow logs: "ft-Od" and "ft-Oa", both of which are marked as "Active". The "Status" column for both rows has a red border around the "Active" status indicator.

12. Navigate back to the **CloudWatch Management Console**.

13. Select the **VPCFlowLogs** log group name. You should see there are currently no log streams. Remember, it may take some time before the flow logs start populating data.

The screenshot shows the AWS CloudWatch Log Groups page. A success message at the top says "Log group 'VPCFlowLogs' has been created." The main table shows one log group named "VPCFlowLogs" with a "Standard" log class and "Never expire" retention. The "Status" column for this row has a red border around the "Active" status indicator.

You are going to see a log stream for each elastic network interface (ENI), attached to your EC2 instances. It could take 5-15 minutes to start recording data. Let's generate some network traffic on our own.

## 1.3. Generate Network Traffic

1. In a new browser tab, navigate to **EC2**.



2. In the **Resources** section of the EC2 dashboard, select **Instances (running)**. You should see a **Web Server** instance that was pre-provisioned for the lab.
3. Check the checkbox next to the **Web Server** instance.
4. Click **Connect → Connect**.

The screenshot shows the AWS EC2 Instances page. A red box highlights the 'Connect' button in the top navigation bar. Another red box highlights the 'Connect' button in the 'Connect' dialog box below. The dialog box contains fields for Instance ID (i-0785a2b...), Public IP (3.128.128.128), and Username (root). It also includes a note about the default AMI username being root.

Now that you have connected to the terminal successfully, the VPC flow logs will record for this connection.

```
Amazon Linux 2
AL2 End of Life is 2026-06-30.
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/
[root@ip-172-31-10-1 ~]# sudo su - cloud_user
[cloud_user@ip-172-31-10-1 ~]$ 
```

5. Exit the terminal: **logout**
6. Navigate back to the **EC2 Management Console** tab. We need to change the security group rules such that it will deny access to SSH, ultimately recording a reject record into the VPC Flow Logs.
7. Update the EC2 instance security group:
  - a. Check the checkbox next to the **Web Server** instance, and then use the **Actions** dropdown to select **Security → Change security groups**.

The screenshot shows the AWS EC2 Instances page with one instance listed. The instance is named "Web Server", has an ID of i-078542b..., is running, and is of type t3a.micro. In the "Actions" dropdown, the "Security" option is highlighted with a red box.

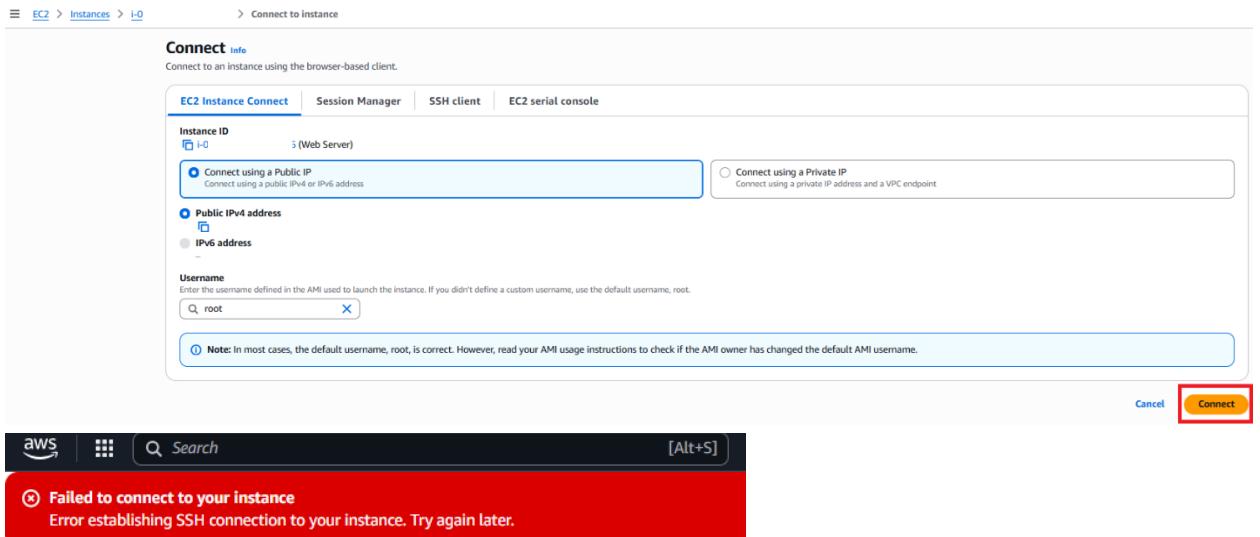
- b. In the **Associated security groups** section, click **Remove** to the right of the security group details to remove the **SecurityGroupHTTPAndSSH** group.

The screenshot shows the "Change security groups" dialog box. In the "Associated security groups" section, there is a table with one row. The row contains the security group name "SecurityGroupHTTPAndSSH", its description "HTTP and SSH Access", and a "Remove" button which is highlighted with a red box. The "Add security group" button is also highlighted with a red box.

- c. Use the search bar in the **Associated security groups** section to select the **SecurityGroupHTTPOnly** security group.
- d. Click **Add security group**, and then click **Save**.

The screenshot shows the "Change security groups" dialog box. In the "Associated security groups" section, there is a table with one row. The row contains the security group name "SecurityGroupHTTPOnly", its description "-HTTPOnly-K2srUHnkHTn1 ( sg-1 )", and a "Save" button which is highlighted with a red box. The "Add security group" button is also highlighted with a red box.

8. Navigate back to your terminal session and reconnect to the EC2 instance.



This time, your connection should time out because you removed SSH access with the security group change. This will be recorded in VPC Flow Logs as a reject record.

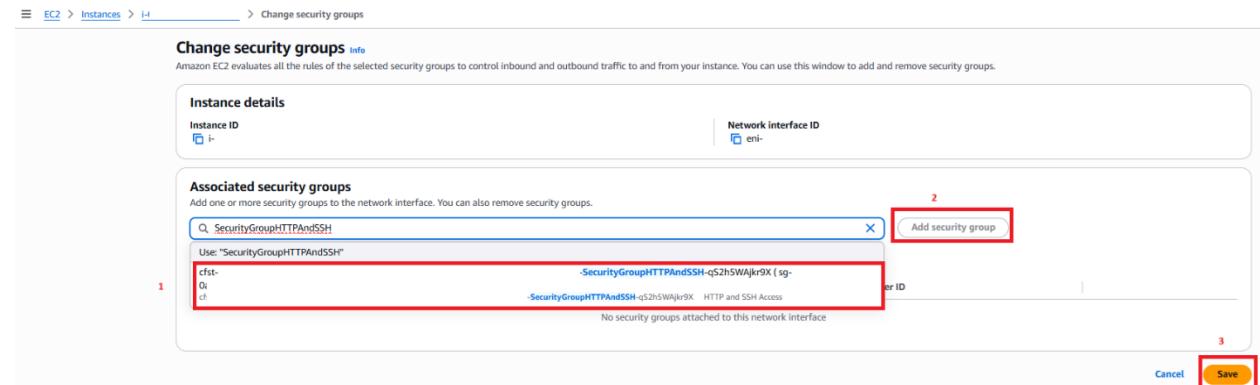
9. Navigate back to the **EC2 Management Console** tab.
10. Revert the EC2 security group back to **SecurityGroupHTTPAndSSH**:
  - a. Ensure the **Web Server** instance is selected, and then use the **Actions** dropdown to select **Security → Change security groups**.

The screenshot shows the 'Instances (1/1)' page for the 'Web Server' instance. The 'Actions' dropdown menu is open, showing options like 'Change security groups', 'Get Windows password', and 'Modify IAM role'. The 'Change security groups' option is highlighted with a red box.

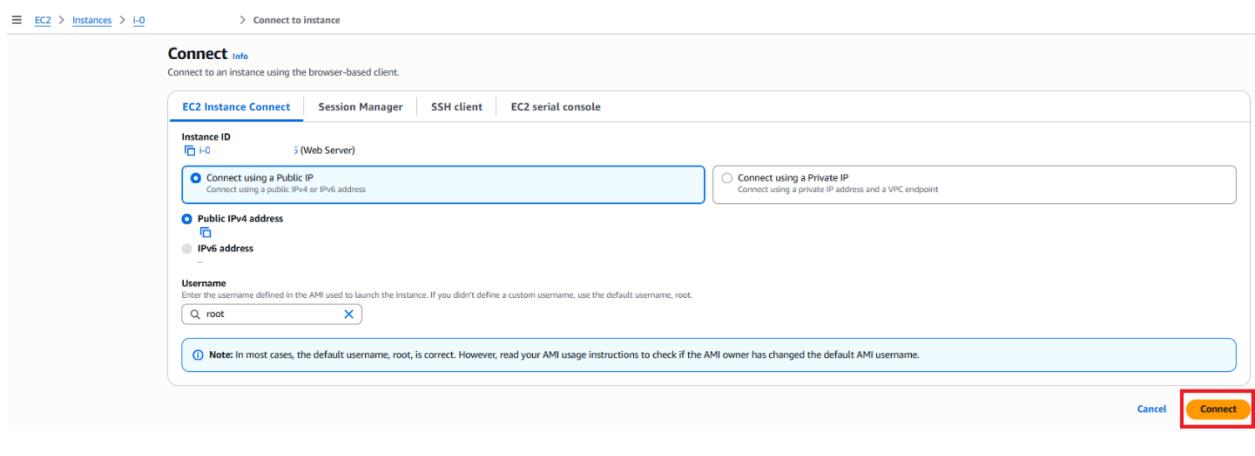
- b. In the **Associated security groups** section, click **Remove** to the right of the security group details to remove the **SecurityGroupHTTPOnly** group.

The screenshot shows the 'Change security groups' page for the 'Web Server' instance. In the 'Associated security groups' section, a table lists a single row for 'SecurityGroupHTTPOnly'. The 'Remove' button in the last column of this row is highlighted with a red box.

- c. Use the search bar in the **Associated security groups** section to select the **SecurityGroupHTTPAndSSH** security group.
- d. Click **Add security group**, and then click **Save**.



11. Navigate back to your terminal session and reconnect to the EC2 instance. This time, the connection should be accepted.



```
aws | ■■■ | Q Search [Alt+S]
[redacted]@compute-1.amazonaws.com ~]# [redacted]

last login: Fri Jun 20 16:11:05 2025 from [redacted] compute-1.amazonaws.com
[redacted]# Amazon Linux 2
[redacted]# AL2 End of Life is 2026-06-30.
[redacted]# A newer version of Amazon Linux is available!
[redacted]# Amazon Linux 2023, GA and supported until 2028-03-15.
[redacted]# https://aws.amazon.com/linux/amazon-linux-2023/
[redacted]# [redacted]
```

So, in our VPC, we configured two flow logs, one to an S3 bucket and the other to CloudWatch Logs. We generated traffic to an EC2 instance in our VPC, and we blocked access to port 22, so we should see reject messages in our flow logs, as well as accept messages.

## 2. Create CloudWatch Filters and Alerts

### 2.1. Create a CloudWatch Log Metric Filter

We'll set up a CloudWatch metric and alarm based on the VPC log data stored in CloudWatch. Next, we'll run a SQL query against that same data using CloudWatch Insights. Finally, we'll create an Amazon Athena table and run a SQL query against that using the VPC Flow Log data stored in S3.

1. Navigate back to the **CloudWatch Management Console** tab.
2. In the CloudWatch sidebar menu, navigate to **Logs** and select **Log groups**.
3. Select the **VPCFlowLogs** log group name. You should now see a log stream. If you don't see a log stream listed yet, wait a few more minutes and refresh the page until the data appears.
4. Select the listed log stream name and review the data.

The screenshot shows the CloudWatch Logs console. On the left, the sidebar has sections for Dashboards, AI Operations, Alarms, Logs (with Log groups selected), Log Anomalies, Live Tail, Logs Insights, Contributor Insights, Metrics, Application Signals (APM), Network Monitoring, and Insights. The 'Logs' section is highlighted with a red box. The main area shows the 'VPCFlowLogs' log group details. The ARN is arn:aws:logs:us-east-1:123456789012:log-group:VPCFlowLogs:. The log class is Info. There are 0 metric filters, 0 subscription filters, and no contributor insights rules. The retention is never expire, and stored bytes are 0. Under 'Log streams', there is one entry: eni-031 10c-all, which is highlighted with a red box. The last event time is 2025-06-20 16:29:13 (UTC). There are tabs for Log streams, Tags, Anomaly detection, Metric filters, Subscription filters, Contributor Insights, Data protection, and Field indexes.

5. Use the breadcrumb along the top of the page to select **VPCFlowLogs**.

Log events							
	Timestamp	Message					
You can use the filter bar below to search for and match terms, phrases, or values in your log events. <a href="#">Learn more about filter patterns</a>							
	Filter events - press enter to search						
▶	2025-06-20T16:51:17.000Z	2 9:	16 eni-03:	210c 8.2	175 10.0.1.44	17 1 32 1:	1750438294 REJECT OK
▶	2025-06-20T16:51:17.000Z	2 9:	16 eni-03:	210c 18.	87.28 10.0.1.	6 4 344 17:	1750438294 ACCEPT OK
▶	2025-06-20T16:51:17.000Z	2 9:	16 eni-03:	210c 10.i	4 18.206.107.	6 2 176 17:	1750438294 ACCEPT OK
▶	2025-06-20T16:51:51.000Z	2 9:	16 eni-03:	210c 206	34.171 10.0.1	86 6 1 60 :	1 1750438333 REJECT OK
▶	2025-06-20T16:51:51.000Z	2 9:	16 eni-03:	210c 95.	.144 10.0.1.4	6 1 40 17:	1750438333 REJECT OK
▶	2025-06-20T16:51:51.000Z	2 9:	16 eni-03:	210c 112	27.2 10.0.1.4	6 1 40 17:	1750438333 REJECT OK
▶	2025-06-20T16:51:51.000Z	2 9:	16 eni-03:	210c 147	133.10 10.0.1	89 6 1 44 :	1 1750438333 REJECT OK
▶	2025-06-20T16:51:51.000Z	2 9:	16 eni-03:	210c 124	135.70 10.0.1	6 1 40 17:	1750438333 REJECT OK
▶	2025-06-20T16:51:51.000Z	2 9:	16 eni-03:	210c 64.i	7.82 10.0.1.4	6 1 40 175:	1750438333 REJECT OK
▶	2025-06-20T16:51:51.000Z	2 9:	16 eni-03:	210c 89.	65.165 10.0.1	635 6 1 40	11 1750438333 REJECT OK
▶	2025-06-20T16:52:15.000Z	2 9:	16 eni-03:	210c 198	24.5 10.0.1.4	1 44 1750:	50438360 REJECT OK
▶	2025-06-20T16:52:15.000Z	2 9:	16 eni-03:	210c 117	63.247 10.0.1	5 6 1 52 1:	1750438360 REJECT OK
▶	2025-06-20T16:52:15.000Z	2 9:	16 eni-03:	210c 199	54.184 10.0.1	808 17 1 4:	335 1750438360 REJECT OK
▶	2025-06-20T16:52:15.000Z	2 9:	16 eni-03:	210c 162	149.77 10.0.1	631 6 1 44	35 1750438360 REJECT OK
▶	2025-06-20T16:52:15.000Z	2 9:	16 eni-03:	210c 18.	87.28 10.0.1.	6 4 344 17:	1750438360 ACCEPT OK
▶	2025-06-20T16:52:15.000Z	2 9:	16 eni-03:	210c 10.i	4 18.206.107.	6 2 176 17:	1750438360 ACCEPT OK
▶	2025-06-20T16:52:58.000Z	2 9:	16 eni-03:	210c 52.	98.91 10.0.1.	6 24 7370	78 1750438387 ACCEPT OK

6. Select the **Metric filters** tab and then click **Create metric filter**. We can use this metric filter to create a CloudWatch alarm.

CloudWatch > Log groups > VPCFlowLogs

**VPCFlowLogs**

**Log group details**

- Log class: Standard
- ARN: arn:aws:logs:us-east-1: :log-group:VPCFlowLogs\*
- Creation time: 1 hour ago
- Retention: Never expire
- Stored bytes: -

**Metric filters** (0)

No metric filters

There are no metric filters defined for this log group.

**Create metric filter**

7. In the **Filter pattern** field, enter the following pattern to track failed SSH attempts on port 22, protocol 6 represents TCP:

[version, account, eni, source, destination, srcport, destport="22", protocol="6", packets, bytes, windowstart, windowend, action="REJECT", flowlogstatus]

8. Use the **Select log data to test** dropdown to select **Custom log data**.

windowstart&gt;windowend&amp;action==\"REJECT\",flowid. A note says: 'You can use metric filters to monitor events in a log group as they are sent to CloudWatch Logs. You can monitor and count specific terms or extract values from log events and associate the results with a metric. Learn more about pattern syntax.' Below it is a 'Test pattern' section with a scrollable log messages area showing several log entries."/&gt;

9. In the **Log event messages** field, replace the existing log data with the following example data:

2 086112738802 eni-0d5d75b41f9bef9e 61.177.172.128 172.31.83.158 39611 22 6 1 40  
1563108188 1563108227 REJECT OK

2 086112738802 eni-0d5d75b41f9bef9e 182.68.238.8 172.31.83.158 42227 22 6 1 44  
1563109030 1563109067 REJECT OK

2 086112738802 eni-0d5d75b41f9bef9e 42.171.23.181 172.31.83.158 52417 22 6 24 4065  
1563191069 1563191121 ACCEPT OK

2 086112738802 eni-0d5d75b41f9bef9e 61.177.172.128 172.31.83.158 39611 80 6 1 40  
1563108188 1563108227 REJECT OK

10. Click **Test pattern** and then review the results.

Event number	\$account	\$action	\$bytes	\$destination	\$destport	\$eni	\$flowlogst
1	086112738802	REJECT	40	172.31.83.158	22	eni-0d5d75b41f9bef9e	OK
2	086112738802	REJECT	44	172.31.83.158	22	eni-0d5d75b41f9bef9e	OK

You will see we have 2 matches out of 4 events, so this is what we're looking for → so it works.

11. Click **Next**.

12. Fill in the metric details:

- a. **Filter name:** In the text field, enter **dest-port-22-reject**.
- b. **Metric namespace:** In the text field, enter a name (e.g., **vpcflowlogs**).
- c. **Metric name:** In the text field, enter **SSH Rejects**.
- d. **Metric value:** In the text field, enter **1**. This will register 1 metric every time there's an SSH reject message in our VPC Flow Log.

The screenshot shows the 'Assign metric' step of the 'Create metric filter' wizard. On the left, the navigation bar includes 'CloudWatch', 'Log groups', 'VPCFlowLogs', and 'Create metric filter'. The sidebar lists 'Log groups', 'Metrics', and 'Insights'. The main panel has three tabs: 'Step 1 Define pattern', 'Step 2 Assign metric' (which is selected), and 'Step 3 Review and create'. Under 'Assign metric', there are fields for 'Filter name' (containing 'dest-port-22-reject'), 'Metric namespace' (containing 'vpcflowlogs'), 'Metric name' (containing 'SSH Rejects'), and 'Metric value' (containing '1'). A note at the bottom says 'Valid metric values are: floating point number (1, 99.9, etc.), numeric field identifiers (\$1, \$2, etc.), or named field identifiers (e.g. \${requestSize} for delimited filter pattern or \${status} for JSON-based filter pattern - dollar (\$) or dollar dot (\$) followed by alphanumeric and/or underscore (\_) characters.)'.

13. Leave the other fields blank and click **Next**.

14. Review the metric details and then click **Create metric filter**.

The screenshot shows the 'Review and create' step of the 'Create metric filter' wizard. The left sidebar and top navigation are identical to the previous screenshot. The main panel shows the 'Review and create' summary. It includes tabs for 'Step 1: Pattern' (with a 'Edit' button) and 'Step 2: Metric' (with a 'Edit' button). Under 'Step 1: Pattern', the 'Create filter pattern' section shows a 'Filter pattern' field with the value '[version, account, eni, source, destination, srcport, destport="22", protocol="6", packets, bytes, windowstart, windowend, action="REJECT", flowlogstatus]'. Under 'Step 2: Metric', the 'Assign metric' section shows the following details:

- Filter name: dest-port-22-reject
- Metric namespace: vpcflowlogs
- Metric name: SSH Rejects
- Applied on transformed logs: -
- Metric value: 1
- Default value: -
- Unit: -

A 'Create metric filter' button is located at the bottom right of the review panel.

Metric filter "dest-port-22-reject" has been created.

**VPCFlowLogs**

**Log group details**

- Log class: Info
- Standard
- ARN: arn:aws:logs:us-east-1: :log-group:VPCFlowLogs:\*
- Creation time: 1 hour ago
- Retention: Never expire
- Stored bytes:

**Metric filters** 1

**Subscription filters** 0

**Contributor Insights rules**

**KMS key ID**

**Anomaly detection** Configure

**Data protection**

**Sensitive data count**

**Field indexes** Configure

**Transformer** Configure

Actions View in Logs Insights Start tailing Search log group

Log streams Tags Anomaly detection Metric filters Subscription filters Contributor Insights Data protection Field indexes Transformer

**Metric filters (1)**

dest-port-22-reject

Filter pattern [version, account, eni, source, destination, srctype, destport="22", protocol="6", packets, bytes, windowstart, windowend, action="REJECT", flowlogstatus]

So that's been created, and you can see our filter here.

## 2.2. Create an Alarm Based on the Metric Filter

- After the metric filter is created, ensure that the **Metric filters** tab is selected.
- In the **Metric filter** details, check the checkbox to the right of the **dest-port-22-reject** filter.
- On the right, click **Create alarm**. The **Alarms** page opens in a new browser tab automatically.

Metric filter "dest-port-22-reject" has been created.

**VPCFlowLogs**

**Log group details**

- Log class: Info
- Standard
- ARN: arn:aws:logs:us-east-1: :log-group:VPCFlowLogs:\*
- Creation time: 6 minutes ago
- Retention: Never expire
- Stored bytes:

**Metric filters** 1

**Subscription filters** 0

**Contributor Insights rules**

**KMS key ID**

**Anomaly detection** Configure

**Data protection**

**Sensitive data count**

**Field indexes** Configure

**Transformer** Configure

Actions View in Logs Insights Start tailing Search log group

Log streams Tags Anomaly detection Metric filters Subscription filters Contributor Insights Data protection Field indexes Transformer

**Metric filters (1/1)**

dest-port-22-reject

Filter pattern [version, account, eni, source, destination, srctype, destport="22", protocol="6", packets, bytes, windowstart, windowend, action="REJECT", flowlogstatus]

Edit Delete Create alarm Create metric filter

- Specify the metric conditions:
  - Period:** Use the dropdown to select **1 minute**.
  - Threshold type:** Ensure that **Static** is selected.

c. **Whenever SSH Rejects is...:** Select **Greater/Equal**.

d. **than...:** In the text field, enter **1**.

Note: The metric will trigger an alarm whenever there is one or more reject messages within a one-minute period.

5. Click **Next**.

The screenshot shows the 'Specify metric and conditions' step of the CloudWatch Create alarm wizard. On the left, a navigation bar lists 'Step 1 Specify metric and conditions', 'Step 2 Configure actions', 'Step 3 Add alarm details', and 'Step 4 Preview and create'. The main area is titled 'Specify metric and conditions' and contains two tabs: 'Metric' and 'Conditions'.  
**Metric Tab:** Shows a graph of 'SSH Rejects' over time from 16:00 to 18:30. A red line represents the metric value, which stays below a blue horizontal threshold at '1'. The graph has a legend indicating 'SSH Rejects'. Below the graph, the 'Namespace' is set to 'vpcflowlogs' and the 'Metric name' is 'SSH Rejects'. The 'Statistic' is set to 'Sum' and the 'Period' is '1 minute'.  
**Conditions Tab:** Shows the configuration for the alarm condition. Under 'Threshold type', 'Static' is selected. Under 'Whenever SSH Rejects is...', 'Greater/Equal' is selected and its threshold value is set to '1'. Other options like 'Anomaly detection' and 'Lower/Equal' are also shown.  
At the bottom right of the page are 'Cancel' and 'Next' buttons.

6. Configure the alarm actions:

- a. **Alarm state trigger:** Ensure that **In alarm** is selected.
- b. **Send a notification to the following SNS topic:** Select **Create a new topic**.
- c. **Create a new topic...:** Leave the default topic name.
- d. **Email endpoints that will receive the notification...:** In the text field, enter an email address (this can be your real email address or a sample address like user@example.com), and then click **Create topic**.

Note: If you enter your real email address, open your email inbox and click the **Confirm Subscription** link that you receive in the SNS email. If not, the alarm will work anyway, you just won't receive email notifications.

7. Click **Next**.

CloudWatch > Alarms > Create alarm

Step 1: Specify metric and conditions  
Step 2: Configure actions  
 Step 3  
 Add alarm details  
 Step 4  
 Preview and create

### Configure actions

**Notification**

Alarm state trigger  
Define the alarm state that will trigger this action.

In alarm  
The metric or expression is outside of the defined threshold.

OK  
The metric or expression is within the defined threshold.

Insufficient data  
The alarm has just started or not enough data is available.

**Send a notification to the following SNS topic**  
Define the SNS (Simple Notification Service) topic that will receive the notification.

Select an existing SNS topic

Create new topic

Use topic ARN to notify other accounts

**Send a notification to...**  
Default\_CloudWatch\_Alarms\_Topic

Only topics belonging to this account are listed here. All persons and applications subscribed to the selected topic will receive notifications.

**Email (endpoints)**  
user@example.com - View in SNS Console

**Add notification**

8. In the **Alarm name** field, enter **SSH rejects**.

9. Click **Next**.

CloudWatch > Alarms > Create alarm

Step 1: Specify metric and conditions  
Step 2: Configure actions  
 Step 3: Add alarm details  
 Step 4: Preview and create

### Add alarm details

**Name and description**

**Alarm name**  
SSH rejects

**Alarm description - optional** [View formatting guidelines](#)

**Edit** **Preview**

# This is an H1  
\*\*double asterisks will produce strong character\*\*  
This is [an example](https://example.com/) inline link.

Up to 1024 characters (0/1024)

**Markdown formatting** is only applied when viewing your alarm in the console. The description will remain in plain text in the alarm notifications.

**Cancel** **Previous** **Next**

10. Review the alarm details and then click **Create alarm**. The alarm is created but will take some time to start populating data.

CloudWatch > Alarms > Create alarm

16:00	16:30	17:00	17:30	18:00	18:30
<input checked="" type="radio"/> SSH Rejects					

**Conditions**

Threshold type  
Static

Whenever SSH Rejects is  
Greater/Equal ( $\geq$ )

than...  
1

**Additional configuration**

**Step 2: Configure actions**

**Actions**

Notification  
When in alarm, send a notification to "Default\_CloudWatch\_Alarms\_Topic"

**Step 3: Add alarm details**

**Alarm details**

Name  
SSH rejects

Description  
-

**Create alarm**

You will see immediately that the State is “Insufficient data”, and that simply means it’s still collecting data because the alarm is newly defined.

The screenshot shows the AWS CloudWatch Alarms interface. At the top, there are two notifications: one indicating a successful alarm creation and another about pending SNS subscriptions. Below these, the 'Alarms (1)' section is displayed. The single alarm, 'SSH rejects', is listed with the following details:

- Name:** SSH rejects
- Last state update (UTC):** 2025-06-20 18:58:47
- Condition:** SSH Rejects >= 1 for 1 datapoints within 1 minute
- Status:** Actions enabled (Warning)

Now, we’re going to generate some more traffic so we can see this alert in action.

### 2.3. Generate Traffic for Alerts

1. Navigate back to the terminal session and reconnect to the EC2 instance.

The screenshot shows the 'EC2 Instance Connect' interface for instance 'i-0'. The 'Connect' step is selected. The 'Public IPv4 address' option is chosen, and the 'root' username is entered in the 'Username' field. A note at the bottom states: 'Note: In most cases, the default username, root, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.' The 'Connect' button is highlighted with a red box.

2. Exit the terminal: **logout**

```
Last login: Fri Jun 20 18:46:20 2025 from [REDACTED].compute-1.amazonaws.com
[REDACTED]# 
[REDACTED]### 
[REDACTED]###\ Amazon Linux 2
[REDACTED]\### AL2 End of Life is 2026-06-30.
[REDACTED]#/ V~'-->
[REDACTED]~~ / A newer version of Amazon Linux is available!
[REDACTED]~.~/ Amazon Linux 2023, GA and supported until 2028-03-15.
[REDACTED]/m/* https://aws.amazon.com/linux/amazon-linux-2023/
[root@ip-[REDACTED] ~]# logout
```

3. Navigate back to the **EC2 Management Console** tab.
4. Update the EC2 instance security group:

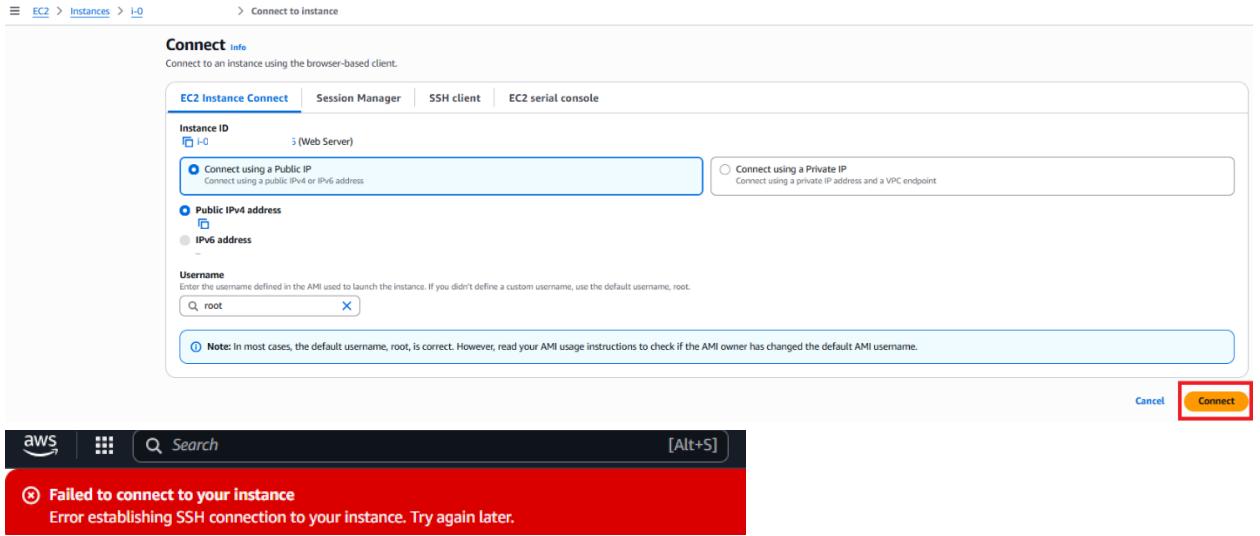
- a. Check the checkbox next to the **Web Server** instance, and then use the **Actions** dropdown to select **Security → Change security groups**.

- b. In the **Associated security groups** section, click **Remove** to the right of the security group details to remove the **SecurityGroupHTTPAndSSH** group.

- c. Use the search bar in the **Associated security groups** section to select the **SecurityGroupHTTPOnly** security group.  
d. Click **Add security group**, and then click **Save**.

5. Navigate back to your terminal session and reconnect to the EC2 instance.

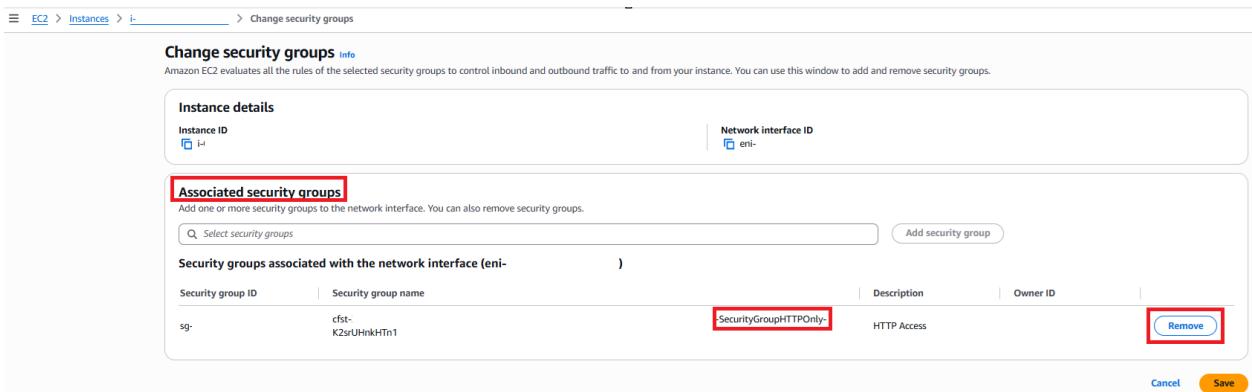
Again, this will be recorded as a rejected record, since you no longer have SSH access.



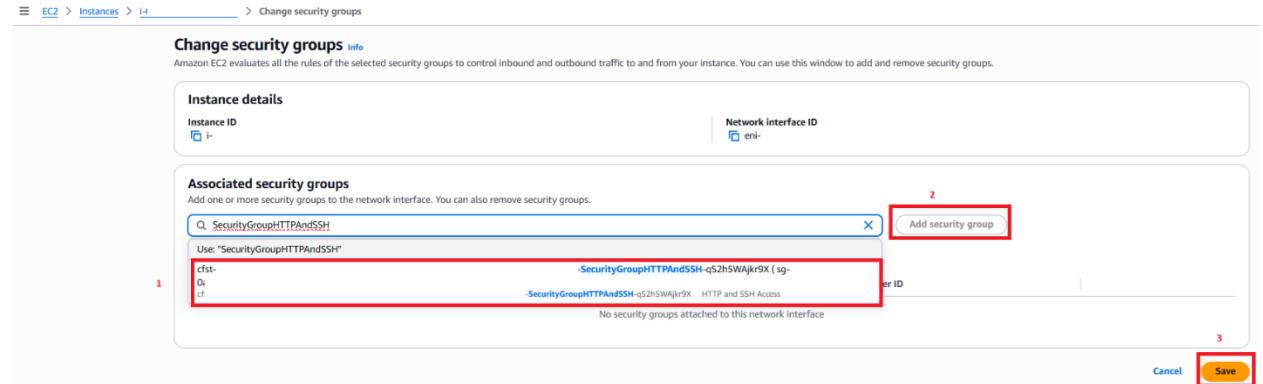
6. Navigate back to the **EC2 Management Console** tab.
7. Revert the EC2 security group back to **SecurityGroupHTTPAndSSH**:
  - a. Ensure the **Web Server** instance is selected, and then use the **Actions** dropdown to select **Security → Change security groups**.



- b. In the **Associated security groups** section, click **Remove** to the right of the security group details to remove the **SecurityGroupHTTPOnly** group.



- c. Use the search bar in the **Associated security groups** section to select the **SecurityGroupHTTPAndSSH** security group.
- d. Click **Add security group**, and then click **Save**.



8. Navigate back to the **CloudWatch Alarms** tab and refresh the alarms details. You should see that the alarm state is now **In alarm**. If you attached the alarm to your email address, you should receive a notification about this alarm.

Name	Type	Threshold	Namespace	Outpoints to alarm
SSH rejects	Metric alarm	SSH Rejects >= 1 for 1 datapoints within 1 minute	vpcflowlogs	1 out of 1
	Description	No description	Metric name	Missing data treatment
		Last state update	Statistic	Treat missing data as missing
				Percentiles with low samples evaluate

**Note:** If the alarm state still shows **Insufficient data**, wait another moment or two and then refresh the alarms details again.

Now we need to do a little bit of analysis on our flow logs data.

### 3. Use CloudWatch Logs Insights

1. In the CloudWatch sidebar menu, navigate to **Logs** and select **Logs Insights**.
2. Use the **Select log group(s)** search bar to select **VPCFlowLogs**.
3. In the right-hand pane, select **Queries**.

The screenshot shows the CloudWatch Logs Insights interface. On the left, the sidebar includes sections for AI Operations, Metrics, Application Signals, Network Monitoring, and Insights. The 'Logs Insights' section is selected and highlighted with a red box. The main area shows a 'Logs Insights Info' panel with a 'Logs Insights QL' tab selected. A 'Selection criteria' dropdown is open, showing 'VPCFlowLogs' selected. Below it is a 'Query generator' with a code editor containing:

```

1 fields @timestamp, @message, @logStream, @Log
2 | sort @timestamp desc
3 | limit 10000
  
```

Buttons for 'Run query', 'Cancel', 'Save', and 'History' are at the bottom of the query editor. To the right, a 'Saved and sample queries' section is visible, also highlighted with a red box. The right sidebar contains sections for 'Queries', 'Saved queries', and 'Sample queries'.

4. In the **Sample queries** section, expand **VPC Flow Logs** and then expand **Top 20 source IP addresses with highest number of rejected requests**.
5. Click **Apply** and note the changes applied in the query editor.
6. Click **Run query**. After a few moments, you'll see some data start to populate.

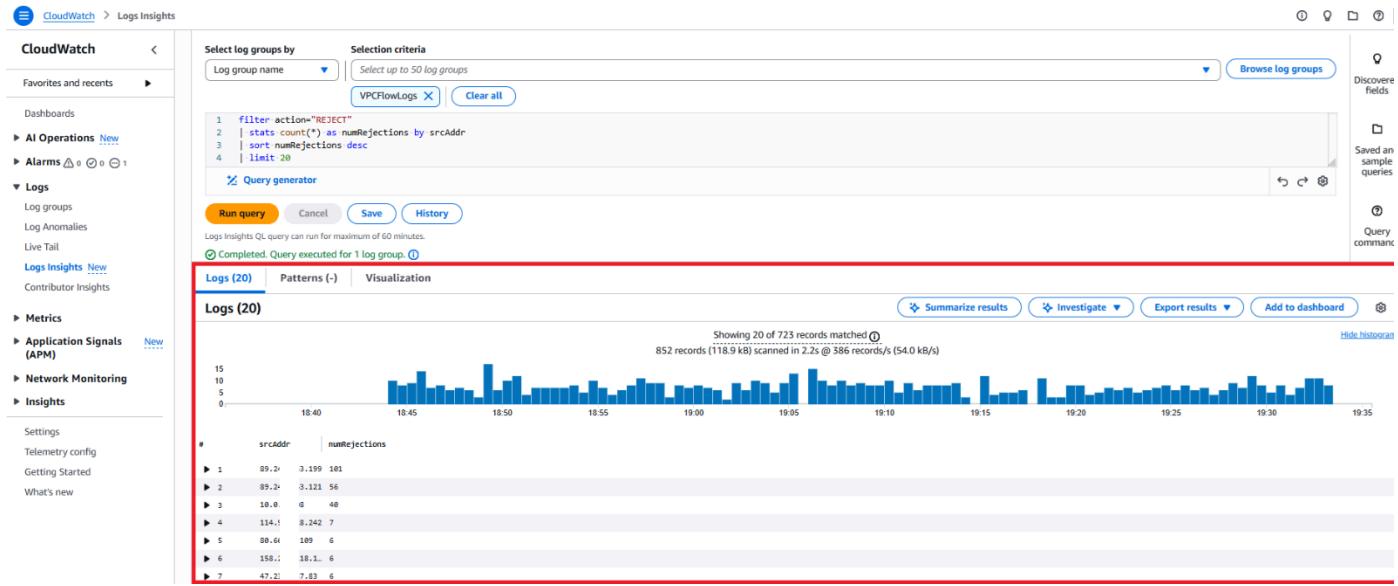
This screenshot shows the same interface after performing steps 4-6. The 'Saved and sample queries' section now has a red box around the 'VPC Flow Logs' section, which is expanded. Under it, the 'Top 20 source IP addresses with highest number of rejected requests' section is also expanded, with a red box highlighting it. The 'Run query' button is highlighted with a red box. The query code in the editor is now:

```

1 filter action="REJECT"
2 | stats count(*) as numRejections by srcAddr
3 | sort numRejections desc
4 | limit 20
  
```

A message 'The code has changed' is displayed above the code editor. The right sidebar remains the same.

If we scroll down a bit, you'll see for each source IP address we have a certain number of rejections.



So, in CloudWatch we created a metric filter. We created a CloudWatch alarm, which provided notifications to a SNS topic that sent us an email when that condition was met. We also performed this ad hoc query based on the data in CloudWatch.

CloudWatch is great for generating metrics for dashboards and alerts like this, but S3 offers much more cost-effective, long-term storage. For the logs stored in S3, you'll now create an Athena table to run a query based on the data stored on that table.

## 4. Analyze VPC Flow Logs Data in Athena

### 4.1. Create the Athena Table

1. Navigate back to the **S3** browser tab and then navigate to your **Buckets**.
2. Select the provisioned bucket name to open it.
3. Select the **AWSLogs/** folder, and then continue opening the subfolders until you reach the **<DAY>** folder containing the logs.
4. In the top right, click **Copy S3 URI**.

Amazon S3 > Buckets > cfst-3029-> vpcflowlogsbucket-oh6931m4krsv > AWSLogs/ > 0 > vpcflowlogs/ > us-east-1/ > 2025/ > 06/ > 20/

**Objects (25)**

Name	Type	Last modified	Size	Storage class
6_vpcflowlogs_us-east-1_fl-145_20250620T18402_20854b3e.log.gz	gz	June 20, 2025, 21:47:52 (UTC+03:00)	694.0 B	Standard
6_vpcflowlogs_us-east-1_fl-145_20250620T18402_e6bd029a.log.gz	gz	June 20, 2025, 21:42:52 (UTC+03:00)	350.0 B	Standard
6_vpcflowlogs_us-east-1_fl-145_20250620T18452_1bbdf3e1.log.gz	gz	June 20, 2025, 21:52:52 (UTC+03:00)	866.0 B	Standard
6_vpcflowlogs_us-east-1_fl-145_20250620T18452_7141cd8d.log.gz	gz	June 20, 2025, 21:47:52 (UTC+03:00)	958.0 B	Standard
6_vpcflowlogs_us-east-1_fl-145_20250620T18502_a597fe2e.log.gz	gz	June 20, 2025, 21:52:52 (UTC+03:00)	822.0 B	Standard
6_vpcflowlogs_us-east-1_fl-145_20250620T18502_bbf787cb.log.gz	gz	June 20, 2025, 21:57:52 (UTC+03:00)	729.0 B	Standard
6_vpcflowlogs_us-east-1_fl-145_20250620T18552_40114ac9.log.gz	gz	June 20, 2025, 21:57:52 (UTC+03:00)	802.0 B	Standard
6_vpcflowlogs_us-east-1_fl-145_20250620T18552_968bb95ca.log.gz	gz	June 20, 2025, 22:02:52 (UTC+03:00)	891.0 B	Standard
6_vpcflowlogs_us-east-1_fl-145_20250620T19002_8ed9d027.log.gz	gz	June 20, 2025, 22:02:51 (UTC+03:00)	769.0 B	Standard

5. Paste the URI into a text file, as you'll need it shortly.

s3://cfst-3029-> -vpcflowlogsbucket-oh6931m4krsv/AWSLogs/us-east-1/2025/06/20/

6. In a new browser tab, navigate to **Athena**.

Services

**Athena** Serverless interactive analytics service

7. On the right, click **Launch query editor**.

Analytics

## Amazon Athena

Start querying data instantly.

Amazon Athena is an interactive query service that makes it easy to analyze data in Amazon S3 and other federated data sources using standard SQL.

**Get started**

- Query your data with Trino
 

Use Query editor to analyze data on S3, on-premises, or on other clouds.
- Analyze your data using PySpark and Spark SQL
 

Use notebooks to build interactive Spark applications.

**Launch query editor**

8. Select the **Settings** tab and then click **Manage**.

Amazon Athena > Query editor

**Settings**

**Query result encryption**

1

Query result location	Encrypt query results	Expected bucket owner	Assign bucket owner full control over query results
-	-	-	Turned off

**Workgroup** primary

**Manage**

9. In the **Location of query result** field, paste your copied S3 URI.

10. Click **Save**.

The screenshot shows the 'Manage settings' page for Amazon Athena. In the 'Query result location and encryption' section, the 'Query result location - optional' field contains the URI `s3://cfst-3029--vpcflowlogsbucket-oh6931m4krs/AWSLogs/1/`. Below it, there's a note about creating lifecycle rules and a link to 'Lifecycle configuration'. Under 'Expected bucket owner - optional', there's a field to 'Enter AWS account ID'. There are also checkboxes for 'Assign bucket owner full control over query results' and 'Encrypt query results'. At the bottom right are 'Cancel' and 'Save' buttons, with 'Save' being highlighted.

**Query result location and encryption**

Location of query result - optional  
Enter an S3 prefix in the current region where the query result will be saved as an object.

Paste the S3 URI copied

`s3://cfst-3029--vpcflowlogsbucket-oh6931m4krs/AWSLogs/1/`

You can create and manage lifecycle rules for this bucket  
Use Amazon S3 lifecycle rules to store your query results and metadata cost effectively or to delete them after a period of time. [Learn more](#)

Expected bucket owner - optional  
Specify the AWS account ID that you expect to be the owner of your query results output location bucket.

Enter AWS account ID

Assign bucket owner full control over query results  
Enabling this option grants the owner of the S3 query results bucket full control over the query results. This means that if your query result location is owned by another account, you grant full control over your query results to the other account.

Encrypt query results

Cancel Save

Amazon Athena > Query editor

Settings successfully updated.

Editor | Recent queries | Saved queries | **Settings**

Workgroup primary

Query result encryption

Query result location  
`s3://cfst-30--vpcflowlogsbucket-oh6931m4krs/AWSLogs/1/`

Encrypt query results

Expected bucket owner

Assign bucket owner full control over query results  
Turned off

Manage

If you get an error message when you hit Save, it just means you probably forgot a trailing slash at the end of your S3 bucket.

In order to read the data in the table, we need to partition it.

## 4.2. Create Partitions and Analyze the Data

1. Select the query editor's **Editor** tab.
2. In the **Query 1** editor, paste the following query, replacing `{your_log_bucket}` and `{account_id}` with your log bucket and account ID details (you can pull these from the S3 URI path you copied):

```
CREATE EXTERNAL TABLE IF NOT EXISTS default.vpc_flow_logs (
version int,
account string,
interfaceid string,
sourceaddress string,
destinationaddress string,
sourceport int,
destinationport int,
protocol int,
```

```

    numpackets int,
    numbytes bigint,
    starttime int,
    endtime int,
    action string,
    logstatus string
)
PARTITIONED BY (dt string)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ''
LOCATION 's3://[your_log_bucket]/AWSLogs/{account_id}/vpcflowlogs/us-east-1/'
TBLPROPERTIES ('skip.header.line.count'='1');

```

- Click **Run**. You should see a message indicating that the query was successful.

The screenshot shows the Amazon Athena Query Editor interface. The top navigation bar includes 'Amazon Athena > Query editor', 'Workgroup primary', and a 'Edit preferences' button. The left sidebar has tabs for 'Editor' (which is selected and highlighted in red), 'Recent queries', 'Saved queries', and 'Settings'. A status message at the top says 'Athena now supports typeahead code suggestions to speed up SQL query development. Typeahead suggestions are turned on by default. You can change this setting in query editor preferences.' Below the status message is a 'Query 1' editor area containing the provided SQL code. The code is highlighted with a red box. At the bottom of the editor are buttons for 'Run' (highlighted in red), 'Explain', 'Cancel', 'Clear', and 'Create'. To the right of the editor is a results pane titled 'Query results' with a 'Results' tab. It displays the message 'No results' and 'Run a query to view results'. There are also 'Copy' and 'Download results CSV' buttons. The bottom right corner of the interface shows a timestamp: 'Reuse query results up to 60 minutes ago'.

- On the right, click the + icon to open a new query editor.

The screenshot shows the Amazon Athena Query Editor interface. A red box highlights the '+' icon in the top right corner. The query editor contains the following SQL code:

```

8 destinationport int,
9 protocol int,
10 numpackets int,
11 numbytes bigint,
12 starttime int,
13 endtime int,
14 action string,
15 logstatus string
16 )
17 PARTITIONED BY (dt string)
18 ROW FORMAT DELIMITED
19 FIELDS TERMINATED BY ''
20 LOCATION 's3://cfst- -vpcflowlogsbucket-oh6931m4krs/AWSLogs/0 /vpcflowlogs/us-east-1/2025/06/20/';
21 TBLPROPERTIES ('skip.header.line.count'='1');
22
SQL Ln 20, Col 134

```

Below the code, there are buttons for 'Run again', 'Explain', 'Cancel', 'Clear', and 'Create'. To the right, there's a 'Reuse query results up to 60 minutes ago' checkbox. The status bar at the bottom shows 'Time in queue: 54 ms', 'Run time: 615 ms', and 'Data scanned: -'.

- In the editor, paste the following query, replacing **YYYY-MM-DD** with the current date, and replacing the existing location with your copied S3 URI:

```
ALTER TABLE default.vpc_flow_logs
ADD PARTITION (dt='YYYY-MM-DD')
location 's3://{your_log_bucket}/AWSLogs/{account_id}/vpcflowlogs/us-east-1/YYYY/MM/DD/';
```

- Click **Run**. You should see a message indicating that the query was successful.

The screenshot shows the Amazon Athena Query Editor interface with a new query editor tab open. A red box highlights the '+ icon' in the top right corner of the new tab. The new tab contains the following SQL code:

```

1 ALTER TABLE default.vpc_flow_logs
2 ADD PARTITION (dt='2025-06-20')
3 location 's3://cfst-3029- -vpcflowlogsbucket-oh6931m4krs/AWSLogs/0 /vpcflowlogs/us-east-1/2025/06/20/';
4

```

The left sidebar shows the data source is 'AwsDataCatalog', catalog is 'None', and database is 'default'. The tables section shows 'vpc\_flow\_logs' is selected. The 'Run' button is highlighted with a red box. The status bar at the bottom shows 'Time in queue: 54 ms', 'Run time: 615 ms', and 'Data scanned: -'.

- On the right, click the + icon to open a new query editor.
- In the editor, paste the following query:

```
SELECT day_of_week(from_iso8601_timestamp(dt)) AS
day,
dt,
```

```


interfaceid,
sourceaddress,
destinationport,
action,
protocol
FROM vpc_flow_logs
WHERE action = 'REJECT' AND protocol = 6
ORDER BY sourceaddress
LIMIT 100;


```

- Click **Run**. Your partitioned data should display in the query results.

The screenshot shows the Amazon Athena Query Editor interface. The top navigation bar includes 'Amazon Athena > Query editor', 'Workgroup primary', and 'Edit preferences'. Below the navigation is a toolbar with 'Editor', 'Recent queries', 'Saved queries', and 'Settings'. A message通知 says 'Athena now supports typeahead code suggestions to speed up SQL query development. Typeahead suggestions are turned on by default. You can change this setting in query editor preferences.' The main area has three tabs: 'Query 1' (selected), 'Query 2', and 'Query 3'. The 'Query 1' tab contains the following SQL code:

```


1 SELECT day,dt,interfaceid,
2      sourceaddress,
3      destinationport,
4      action,
5      protocol
6 FROM vpc_flow_logs
7 WHERE action = 'REJECT' AND protocol = 6
8 ORDER BY sourceaddress
9 LIMIT 100;
10
11
12
13


```

Below the code, there are buttons for 'Run again', 'Explain', 'Cancel', 'Clear', and 'Create'. The 'Results' tab is selected, showing a table with 7 rows of data. The table has columns: #, day, dt, interfaceid, sourceaddress, destinationport, action, and protocol. The data is as follows:

#	day	dt	interfaceid	sourceaddress	destinationport	action	protocol
1	5	2025-06-20	eni-0a1	100.192.1	20	REJECT	6
2	5	2025-06-20	eni-0a1	100.192.4	1723	REJECT	6
3	5	2025-06-20	eni-0a1	100.192.7	21	REJECT	6
4	5	2025-06-20	eni-0a1	105.2.250.	8728	REJECT	6
5	5	2025-06-20	eni-0a1	105.2.250.	8728	REJECT	6
6	5	2025-06-20	eni-0a1	105.2.250.	8728	REJECT	6
7	5	2025-06-20	eni-0a1	105.2.250.4	8728	REJECT	6

So, we created VPC flow logs to both CloudWatch and a S3 bucket, we created a CloudWatch metric based on failed attempts to access port 22, and we created a CloudWatch alarm based on that metric and configured automatic email notifications using Amazon SNS.

Then, we used CloudWatch Insights to run ad hoc SQL queries against the log data stored in CloudWatch Logs. We created a table in Amazon Athena for the S3 log data from the VPC flow logs. Finally, we ran an ad hoc SQL query against Amazon Athena for the log data stored in S3.