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

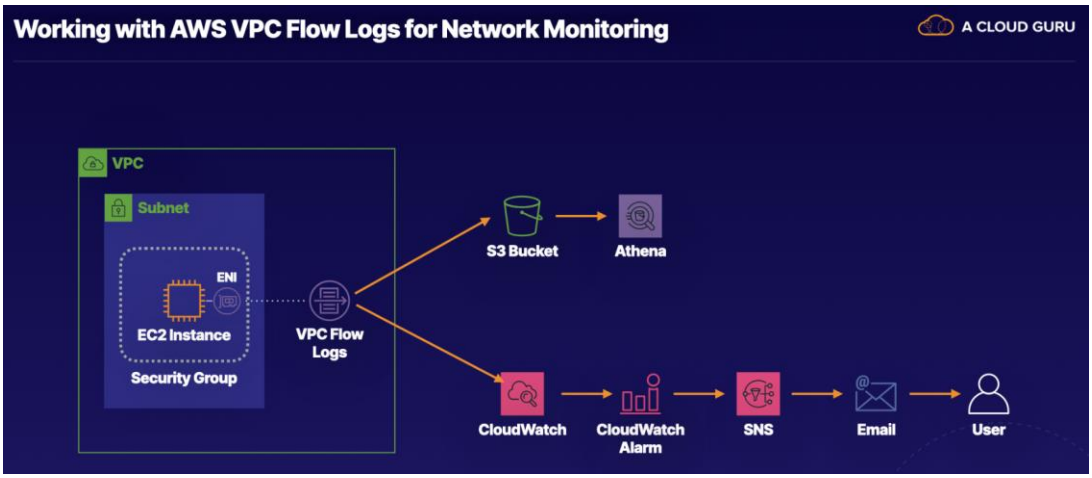
https://aws.amazon.com/athena/faqs/?nc=sn&loc=6#topic-0

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

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

Table of Contents

Lab Diagrams



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.

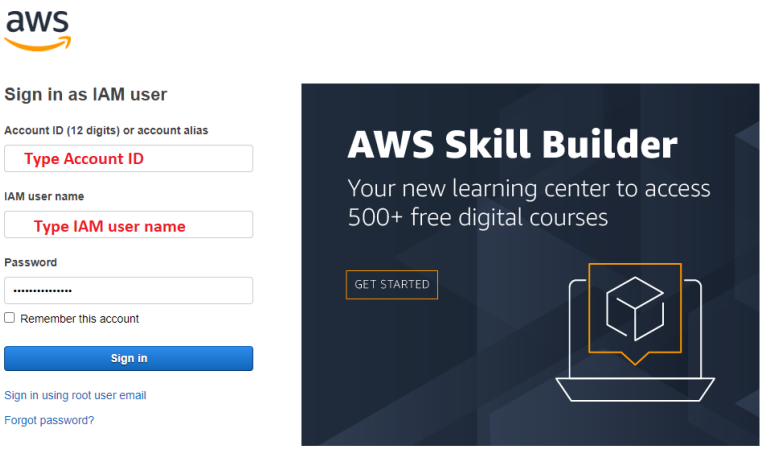
Then, we'll be using our terminal to attempt to SSH into the instance. If the security group allows us SSH access, then the VPC Flow Logs will show a record from the source IP address on port 22 as accepted. If the security group does not allow SSH access, you'll see a record that shows as rejected. 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

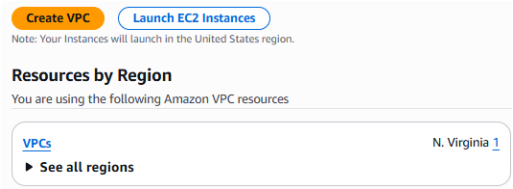


1. Create a CloudWatch Log Group and VPC Flow Logs to CloudWatch

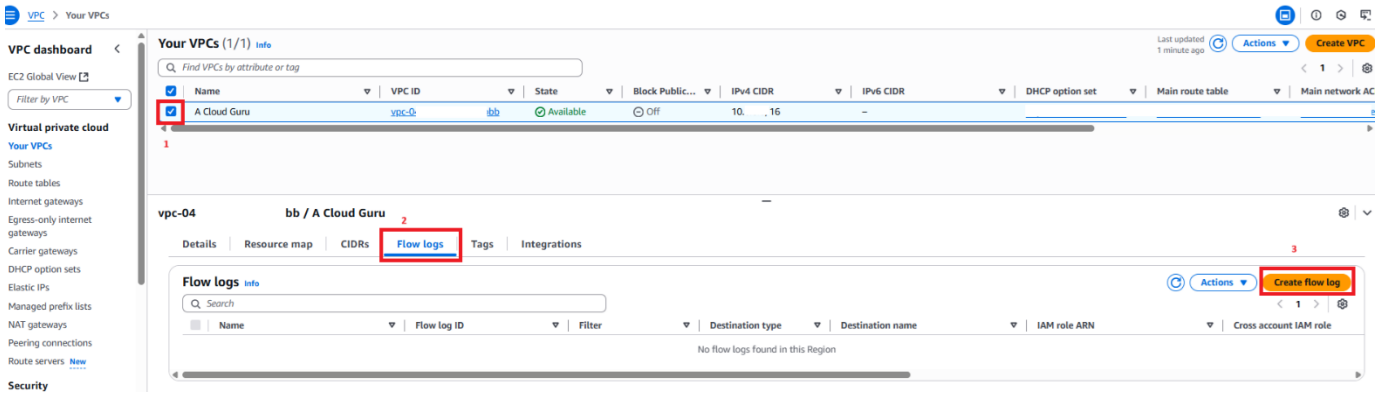
1.1. Create a VPC Flow Log to S3

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

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



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



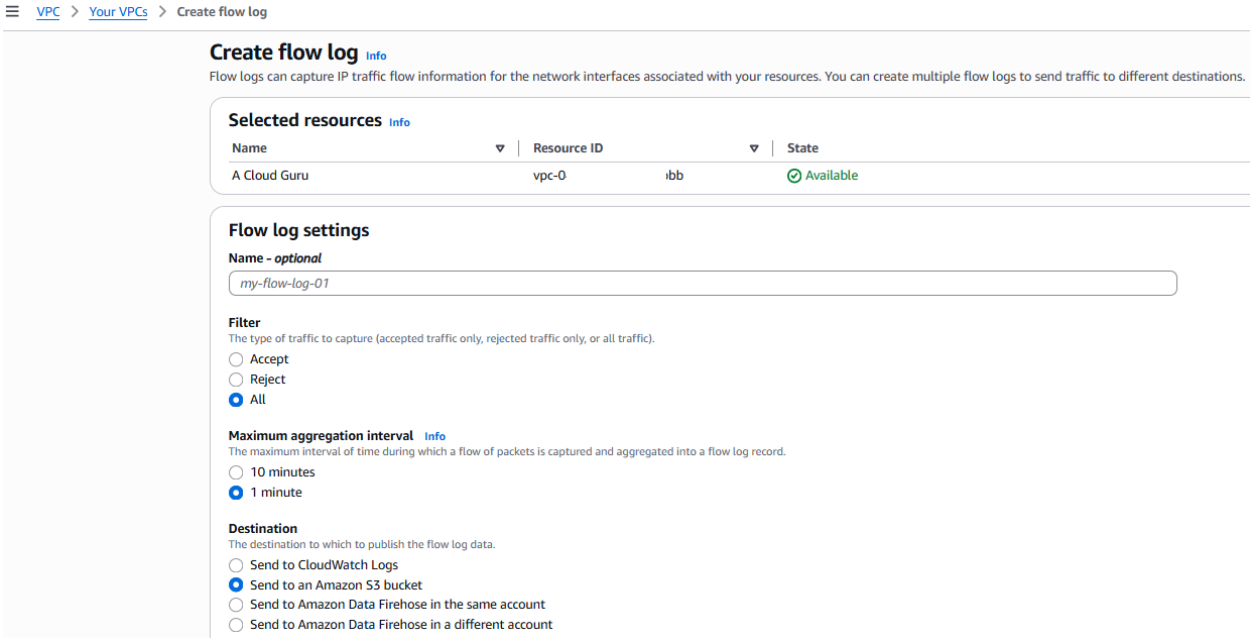
6. Fill in the flow log details:

    a. ***Name***: You can leave this field blank.

    b. ***Filter***: Ensure that **All** is selected.

    c. ***Maximum aggregation interval***: Select **1 minute**.

    d. ***Destination***: Select **Send to an Amazon S3 bucket**.

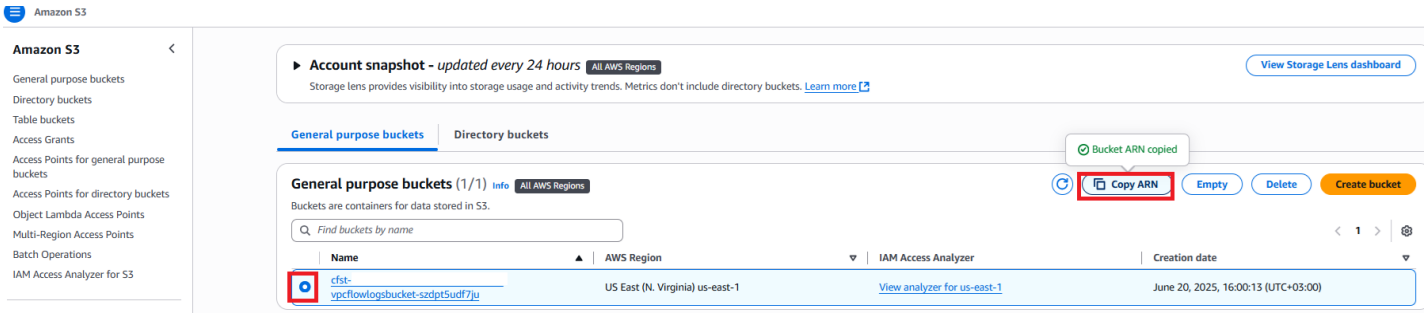


7. Get the S3 bucket ARN:

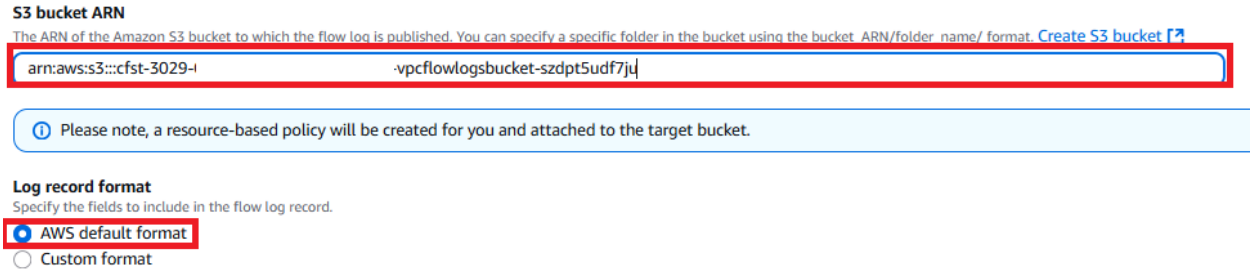
    a. In a new browser tab, navigate to **S3**.

    b. Select the radio button next to the provided bucket.

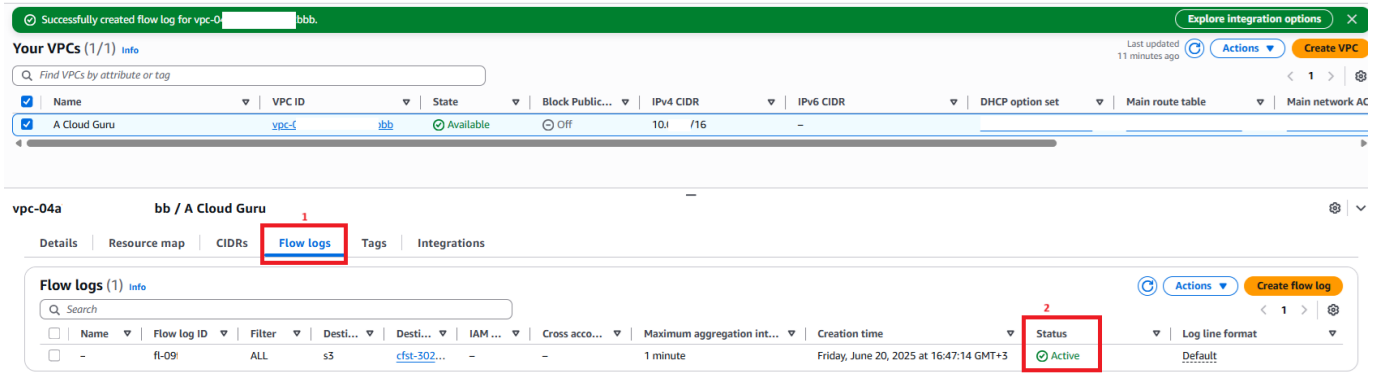
    c. Click **Copy ARN**.



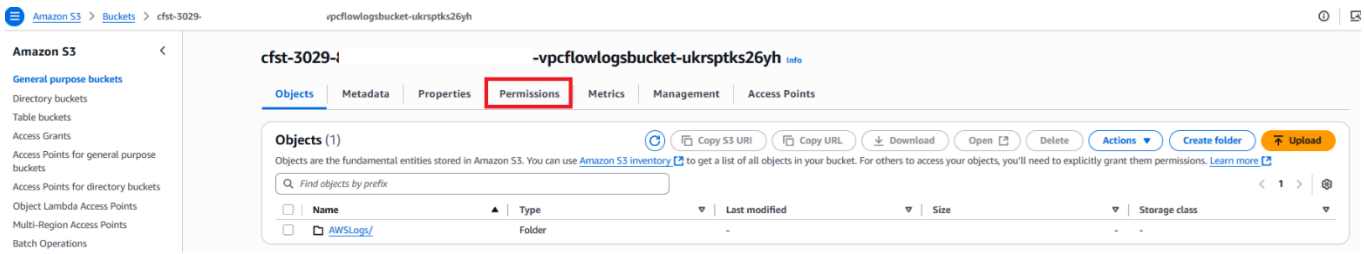
8. Navigate back to the **VPC Management Console** tab and fill in the rest of the flow log details:
    a. ***S3 bucket ARN***: In the text field, paste your copied S3 bucket ARN.
    b. ***Log record format***: Ensure that **AWS default format** is selected.



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.

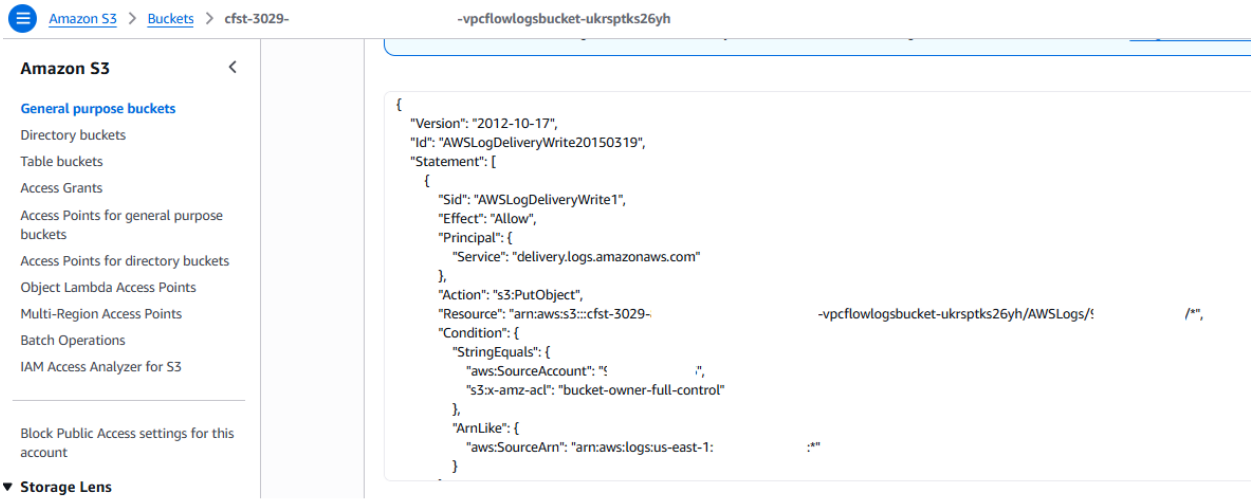


12. Navigate back to the **S3 Management Console** tab.
13. Select your bucket name, and then select the **Permissions** tab.



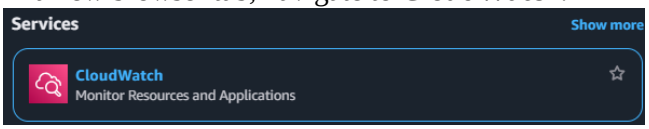
14. Review the bucket policy and note that it is modified automatically by AWS when you create flow logs so that the flow logs can write to the bucket.

    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.



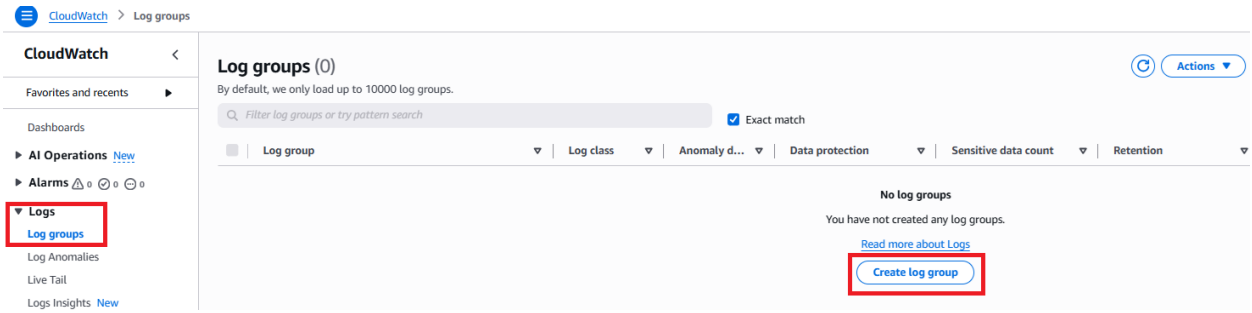
1.2. Create the CloudWatch Log Group and VPC Flow Log

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



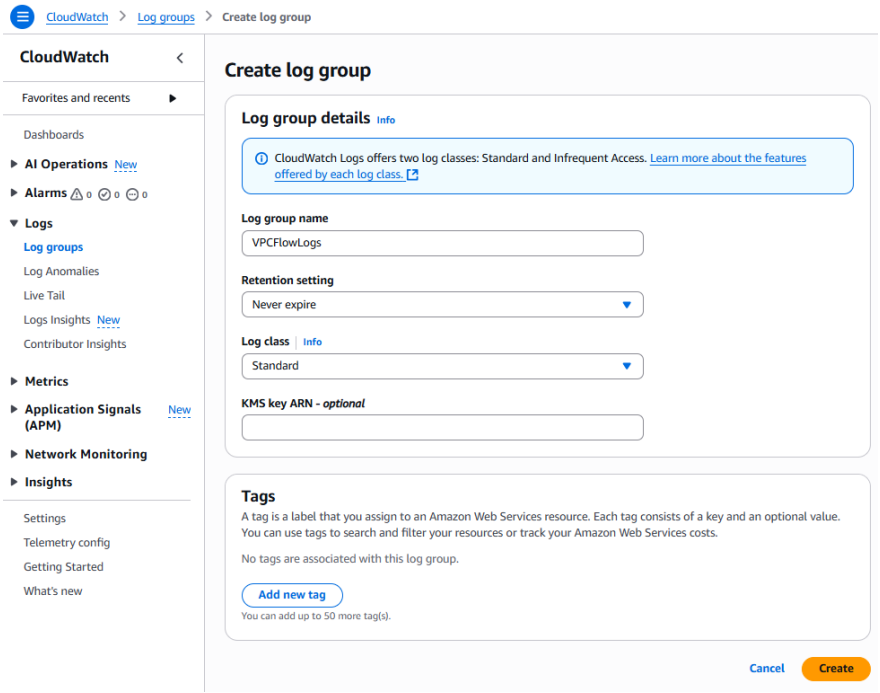
2. In the CloudWatch sidebar menu, navigate to **Logs** and select **Log groups**. A log group is a container for our logs.

3. Click **Create log group**.



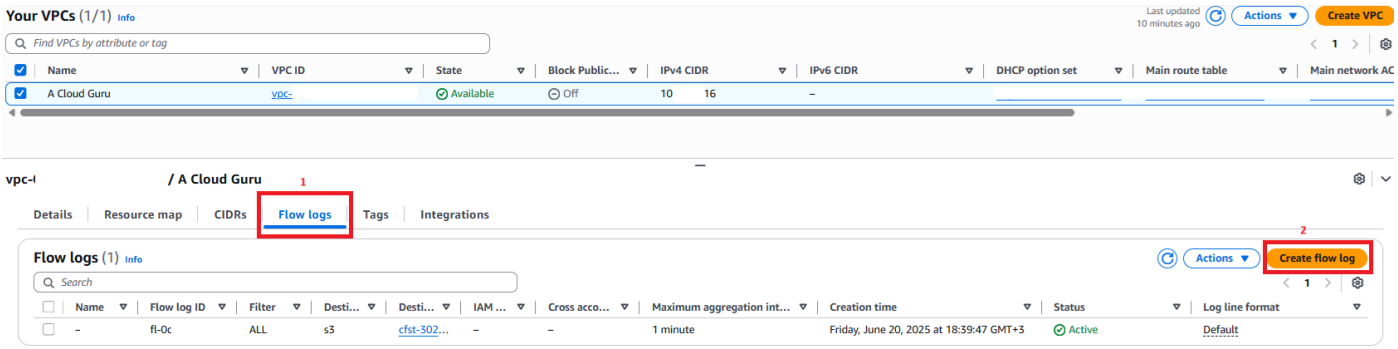
4. In the **Log group name** field, enter ***VPCFlowLogs***.

5. Click **Create**.

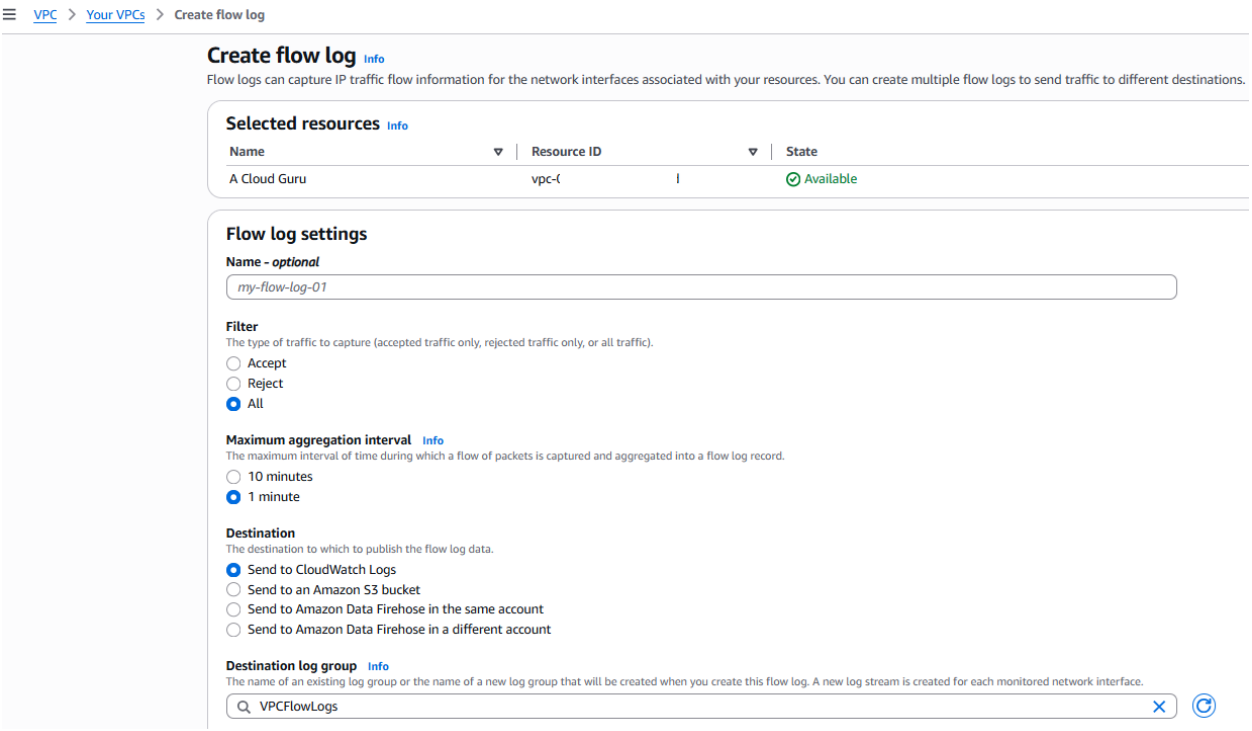


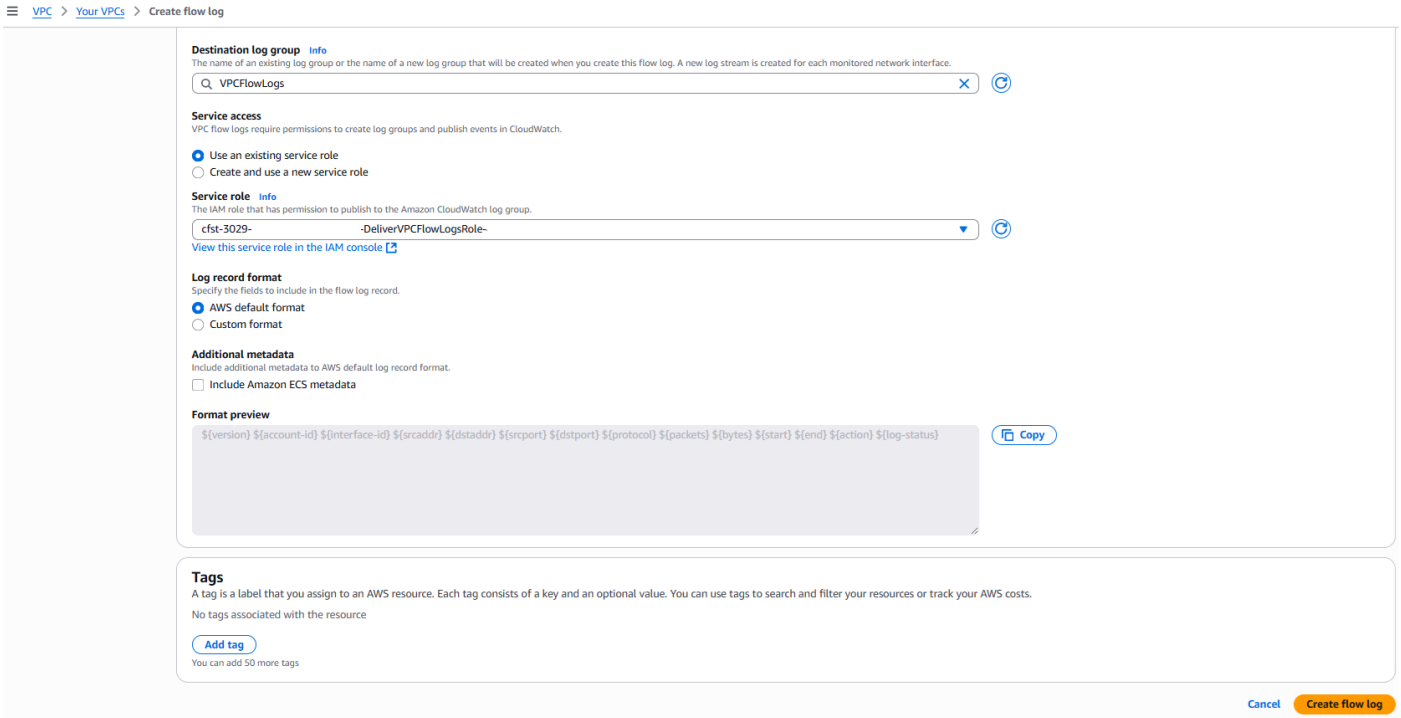
6. Navigate back to the **VPC Management Console** tab and ensure the **Flow logs** tab is still selected.

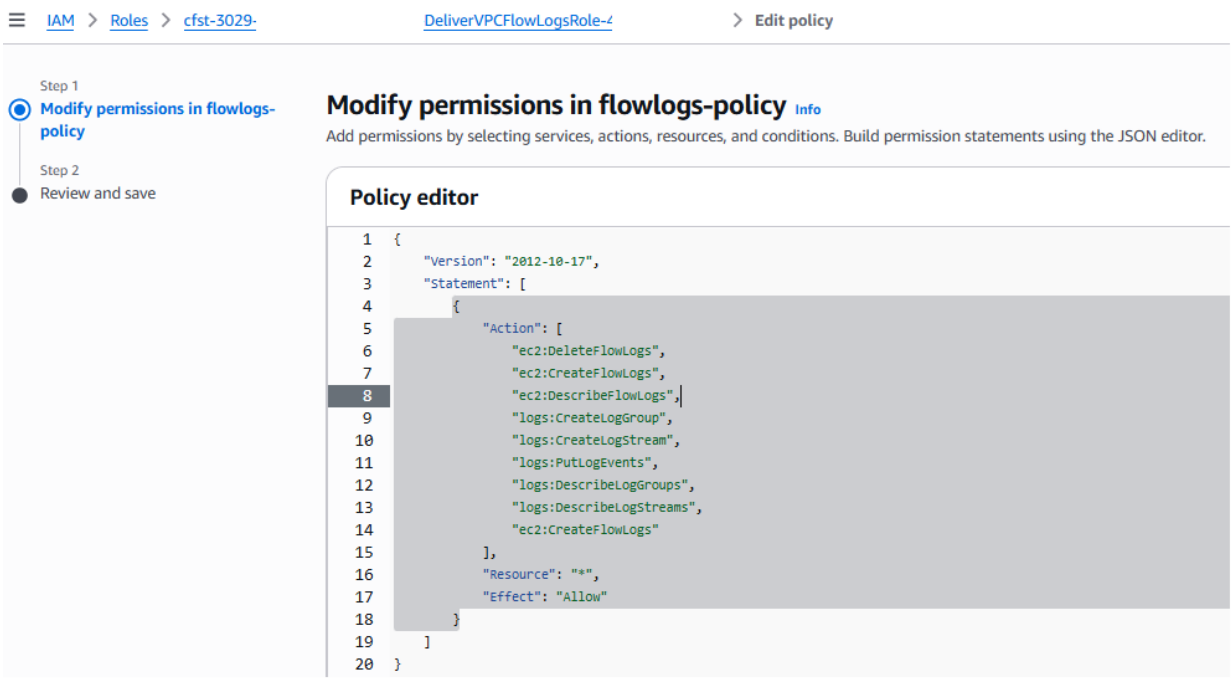
7. On the right, click **Create flow log**.

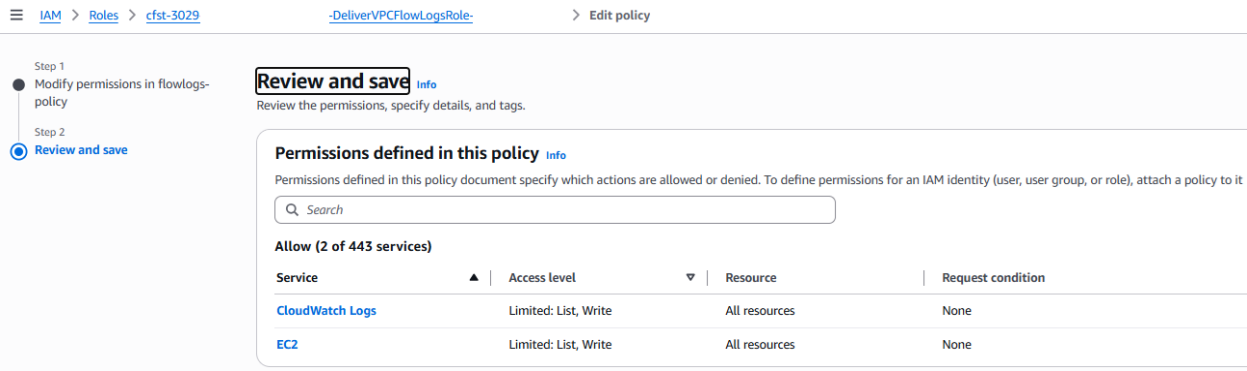


8. Fill in the flow log details:
    a. ***Name***: You can leave this field blank.
    b. ***Filter***: Ensure that **All** is selected.
    c. ***Maximum aggregation interval***: Select **1 minute**.
    d. ***Destination***: Ensure that **Send to CloudWatch Logs** is selected.
    e. ***Destination log group***: Click into the field and select your **VPCFlowLogs** log group.
    f. ***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.
    g. ***Log record format***: Ensure that **AWS default format** is selected.




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.
VPCFlowLogs
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.
cfst-3029- -DeliverVPCFlowLogsRole-
View this service role in the IAM console
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}
Copy
Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.
No tags associated with the resource
Add tag
You can add 50 more tags
Cancel
Create flow log

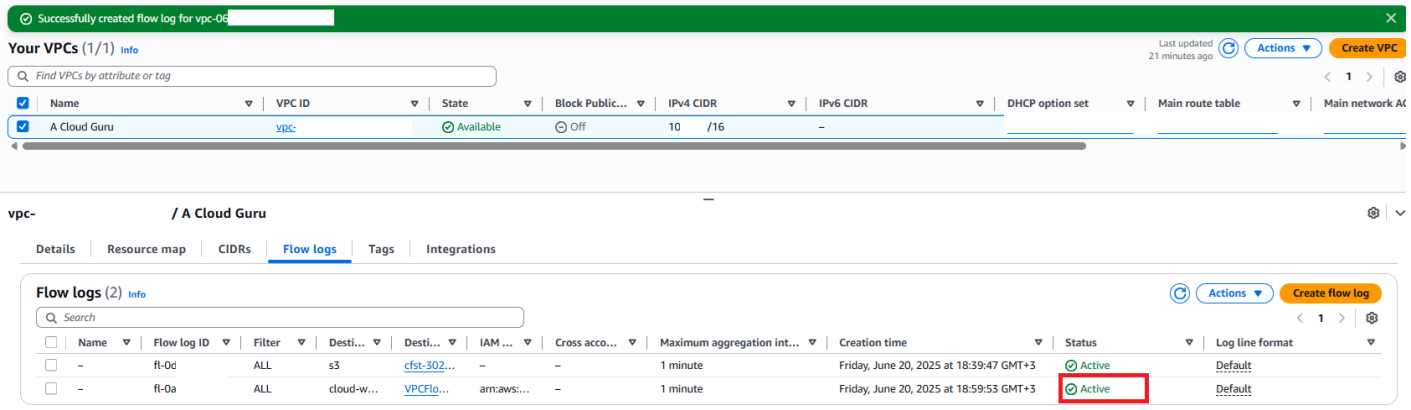

IAM > Roles > cfst-3029- DeliverVPCFlowLogsRole-4 > Edit policy
Step 1
Modify permissions in flowlogs-policy
Step 2
Review and save
Modify permissions in flowlogs-policy Info
Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.
Policy editor
{
"Version": "2012-10-17",
"Statement": [
{
"Action": [
"ec2:DeleteFlowLogs",
"ec2:CreateFlowLogs",
"ec2:DescribeFlowLogs",
"logs:CreateLogGroup",
"logs:CreateLogStream",
"logs:PutLogEvents",
"logs:DescribeLogGroups",
"logs:DescribeLogStreams",
"ec2:CreateFlowLogs"
],
"Resource": "*",
"Effect": "Allow"
}
]
}



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

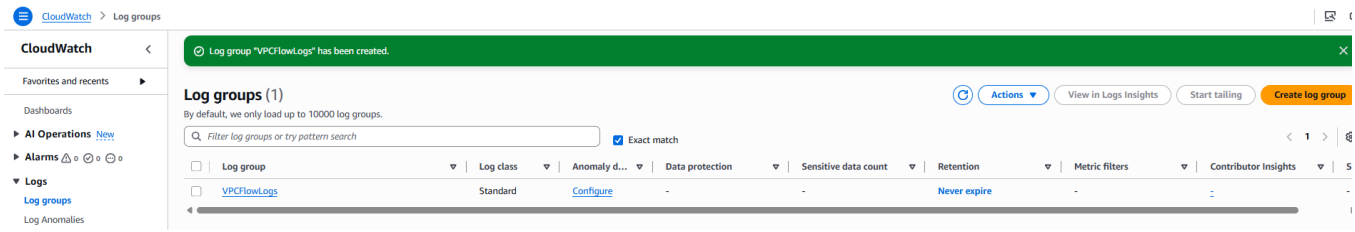
10. From the **Your VPCs** page, ensure the **Flow logs** tab is selected.

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



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

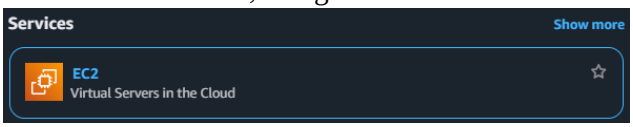
13. Select the **VPCFlowLogs** log group name. You should see there are currently no log streams. Remember, it may take some time before the flow logs start populating data.



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

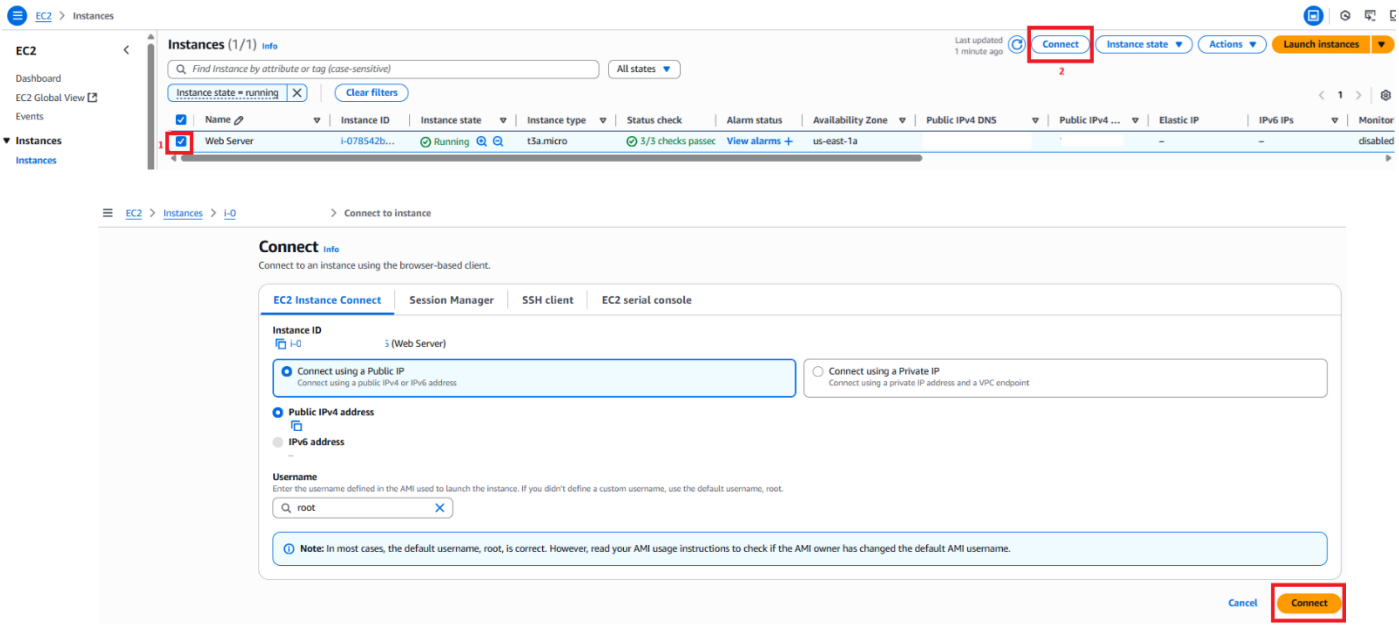
1. In a new browser tab, navigate to **EC2**.



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.**



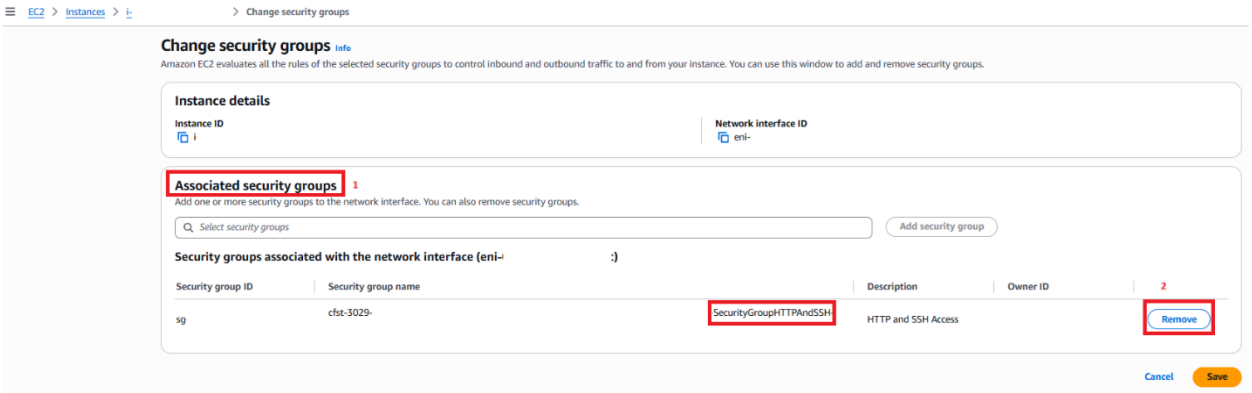
Now that you have connected to the terminal successfully, the VPC flow logs will record for this connection.

5. Exit the terminal: ***logout***

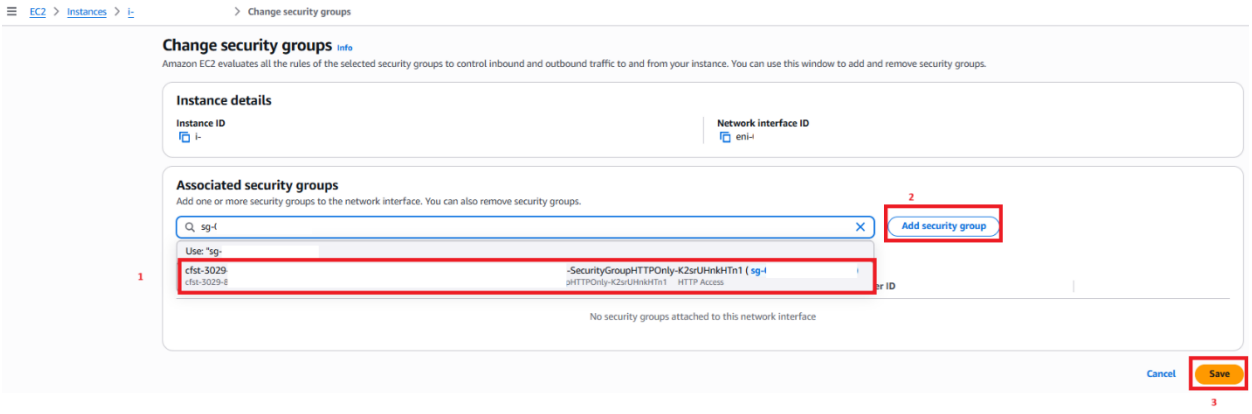
6. Navigate back to the **EC2 Management Console** tab. We need to change the security group rules such that it will deny access to SSH, ultimately recording a reject record into the VPC Flow Logs.

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**.

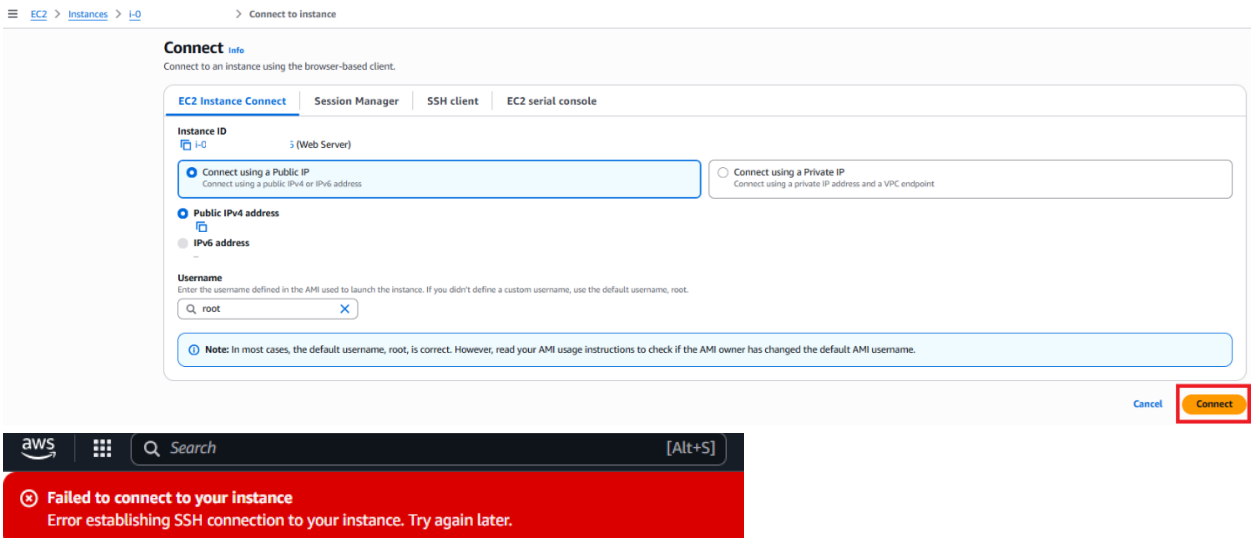
    b. In the **Associated security groups** section, click **Remove** to the right of the security group details to remove the **SecurityGroupHTTPAndSSH** group.



    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**.



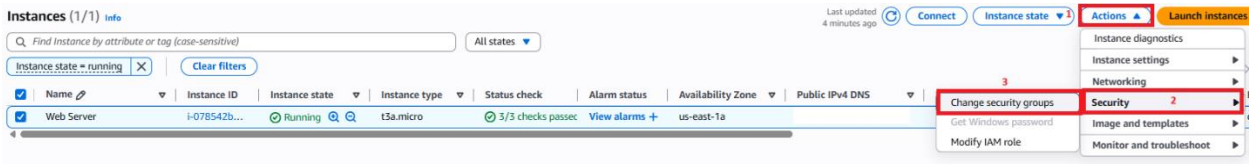
8. Navigate back to your terminal session and reconnect to the EC2 instance.



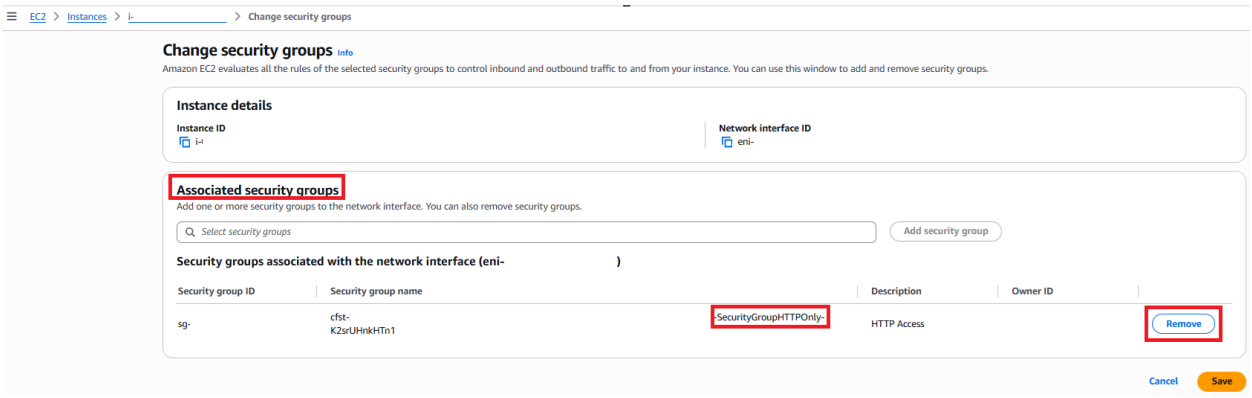
This time, your connection should time out because you removed SSH access with the security group change. This will be recorded in VPC Flow Logs as a reject record.

9. Navigate back to the **EC2 Management Console** tab.

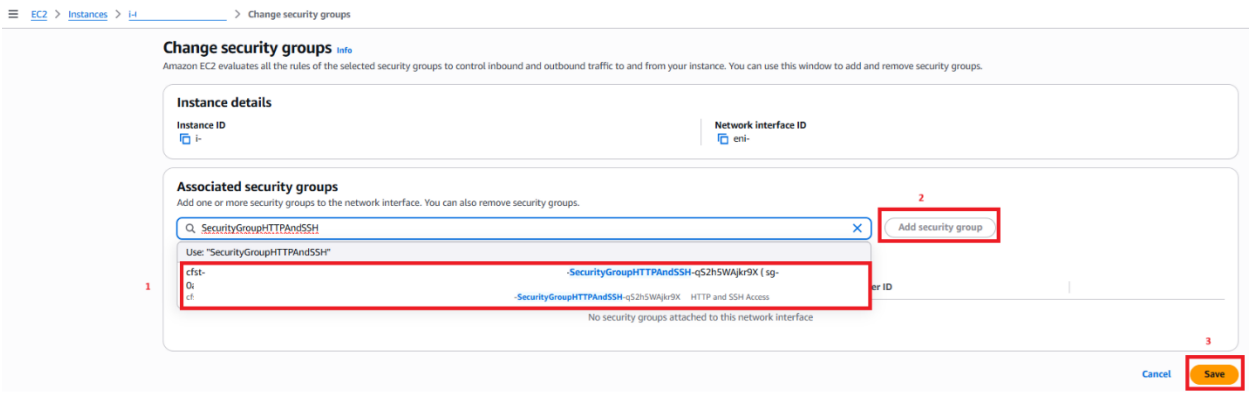
10. Revert the EC2 security group back to **SecurityGroupHTTPAndSSH**:
    a. Ensure the **Web Server** instance is selected, and then use the **Actions** dropdown to select **Security → Change security groups**.



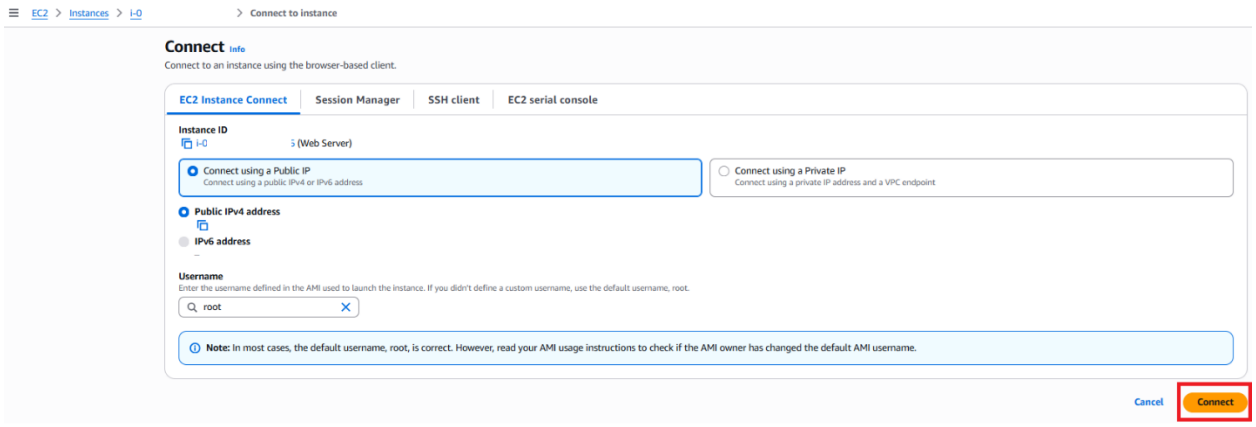
    b. In the **Associated security groups** section, click **Remove** to the right of the security group details to remove the **SecurityGroupHTTPOnly** group.

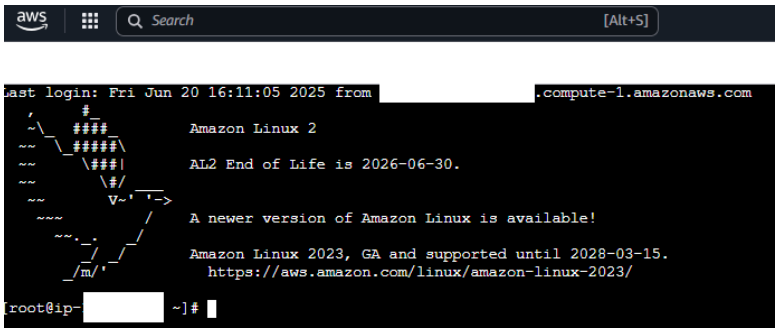


    c. Use the search bar in the **Associated security groups** section to select the **SecurityGroupHTTPAndSSH** security group.
    d. Click **Add security group**, and then click **Save**.



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





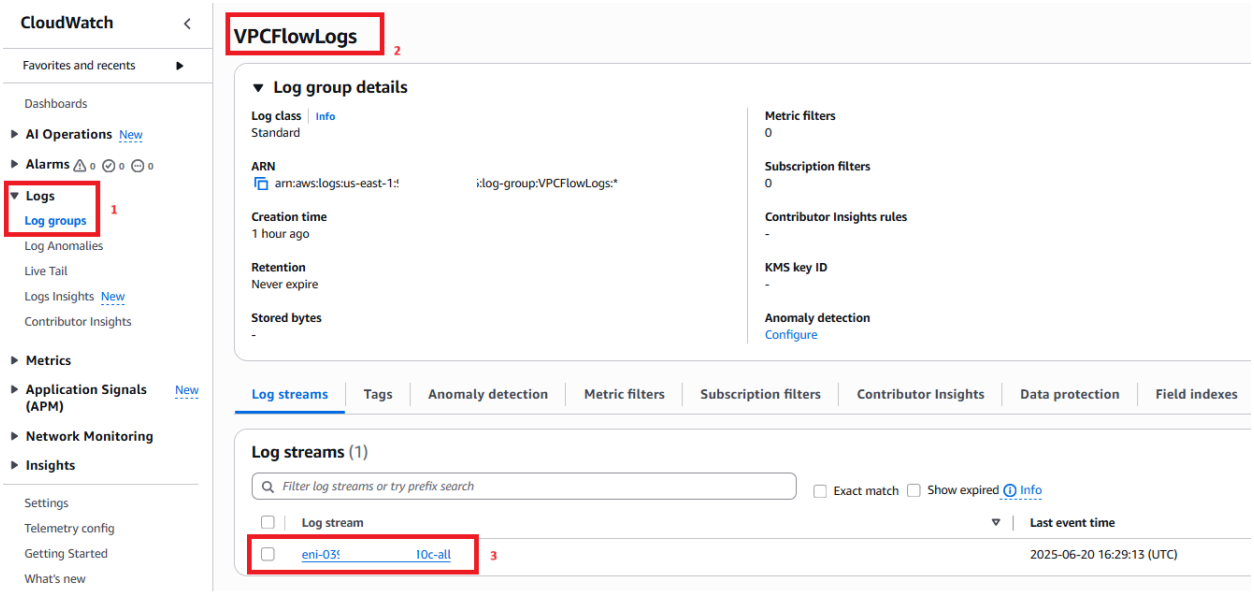
So, in our VPC, we configured two flow logs, one to an S3 bucket and the other to CloudWatch Logs. We generated traffic to an EC2 instance in our VPC, and we blocked access to port 22, so we should see reject messages in our flow logs, as well as accept messages.

2. Create CloudWatch Filters and Alerts

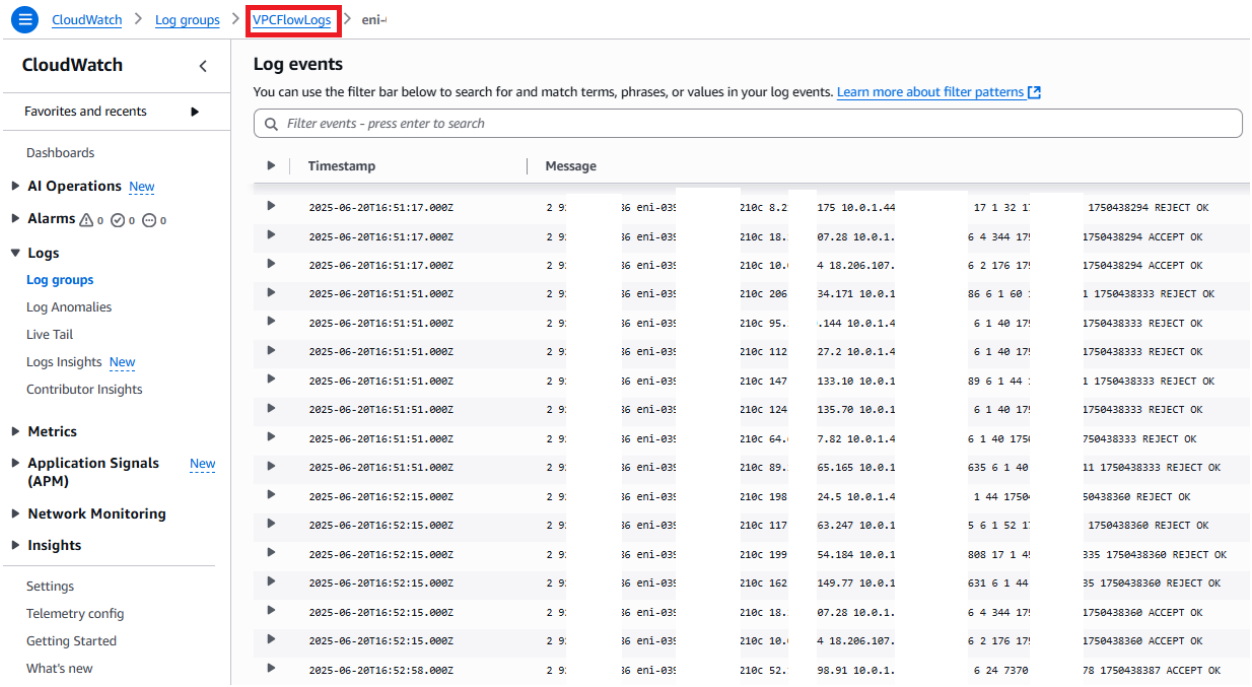
2.1. Create a CloudWatch Log Metric Filter

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.

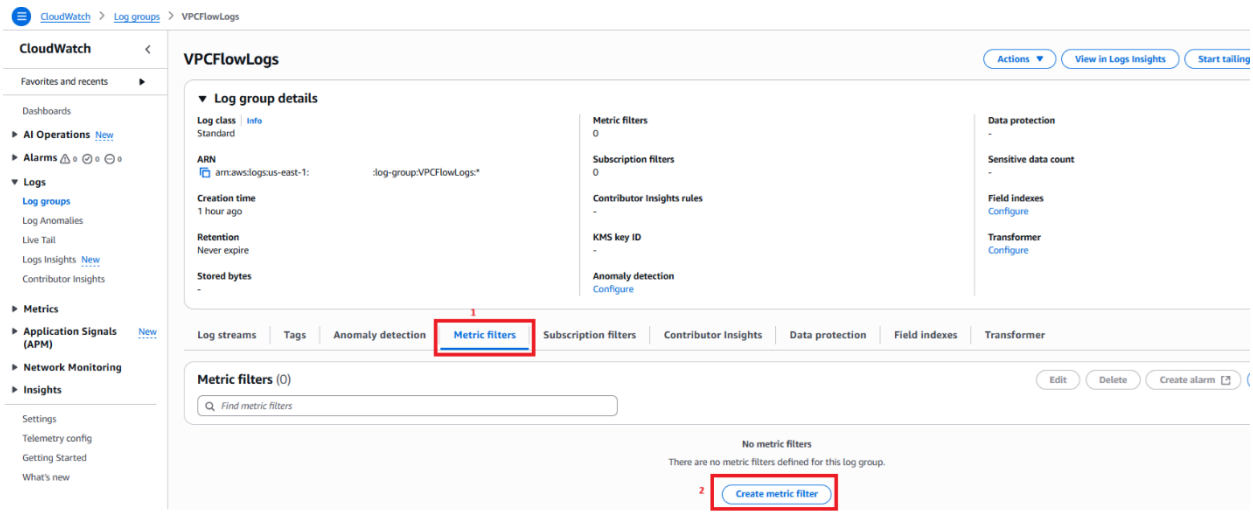
1. Navigate back to the **CloudWatch Management Console** tab.
2. In the CloudWatch sidebar menu, navigate to **Logs** and select **Log groups**.
3. Select the **VPCFlowLogs** log group name. You should now see a log stream. If you don't see a log stream listed yet, wait a few more minutes and refresh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



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



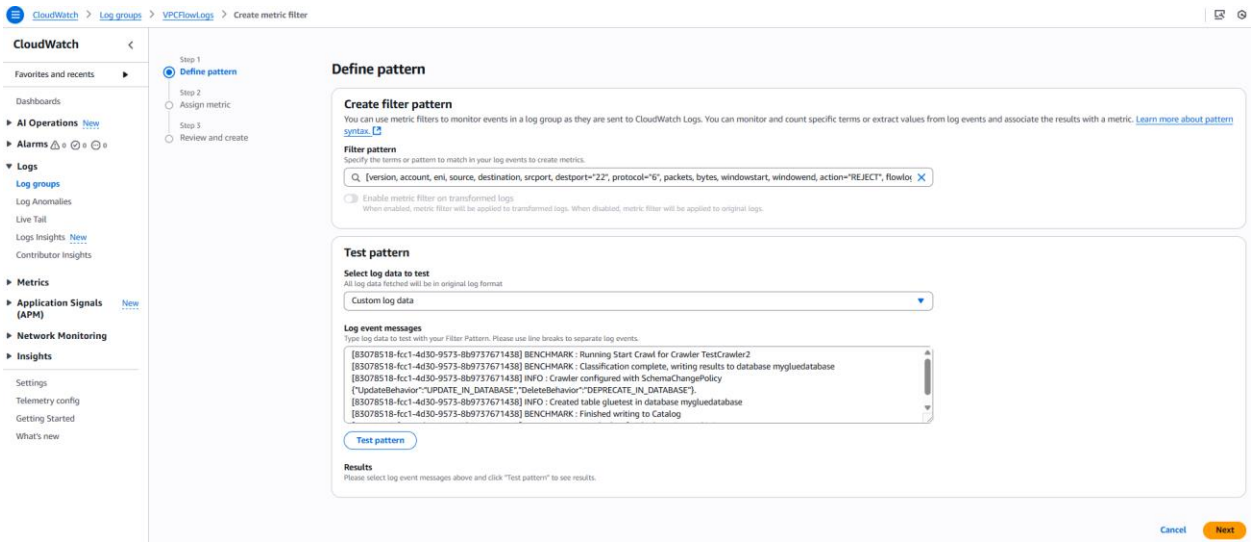
6. Select the **Metric filters** tab and then click **Create metric filter**. We can use this metric filter to create a CloudWatch alarm.



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**.



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

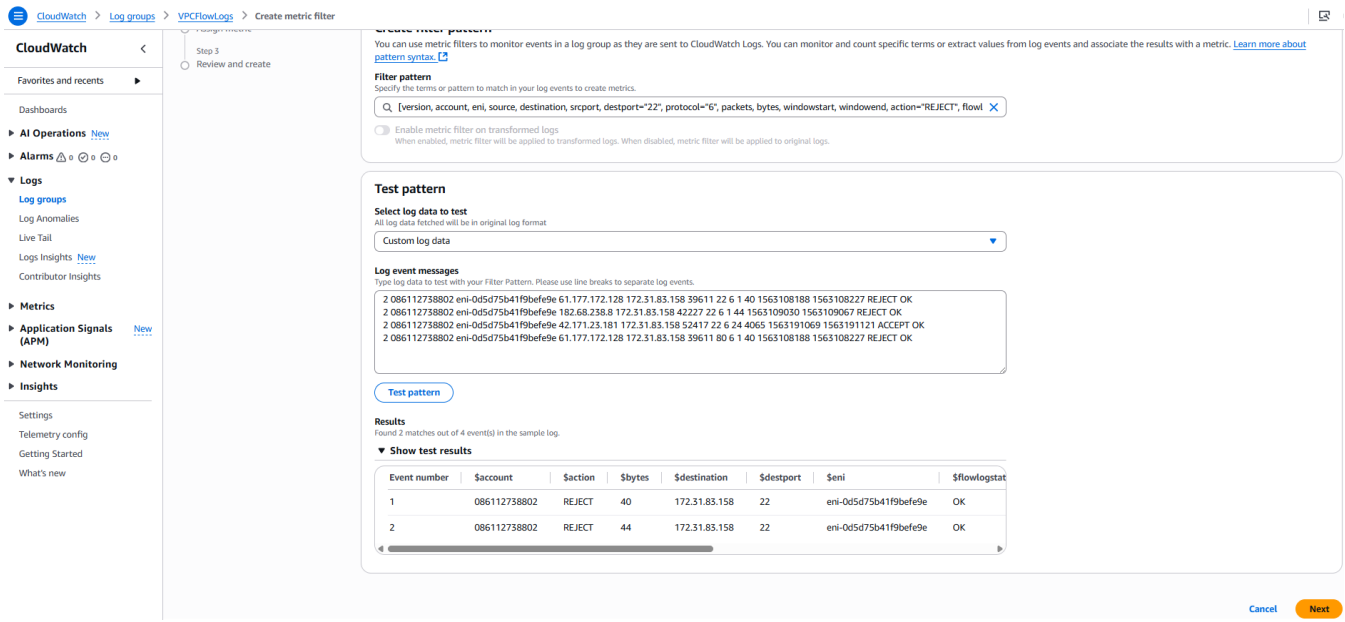
2 086112738802 eni-0d5d75b41f9befe9e 61.177.172.128 172.31.83.158 39611 22 6 1 40 1563108188 1563108227 REJECT OK

2 086112738802 eni-0d5d75b41f9befe9e 182.68.238.8 172.31.83.158 42227 22 6 1 44 1563109030 1563109067 REJECT OK

2 086112738802 eni-0d5d75b41f9befe9e 42.171.23.181 172.31.83.158 52417 22 6 24 4065 1563191069 1563191121 ACCEPT OK

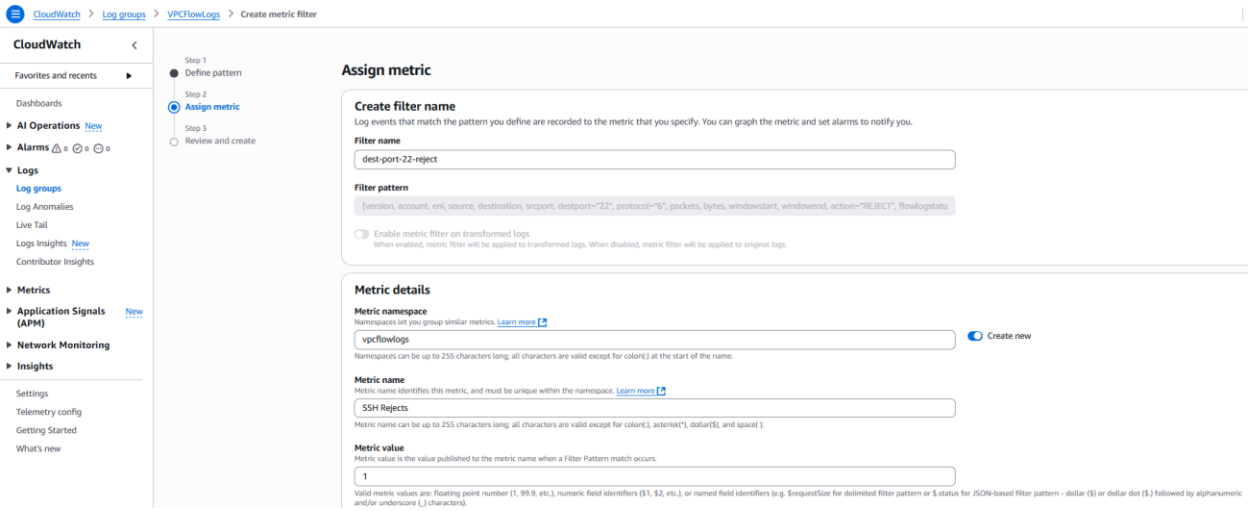
2 086112738802 eni-0d5d75b41f9befe9e 61.177.172.128 172.31.83.158 39611 80 6 1 40 1563108188 1563108227 REJECT OK

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

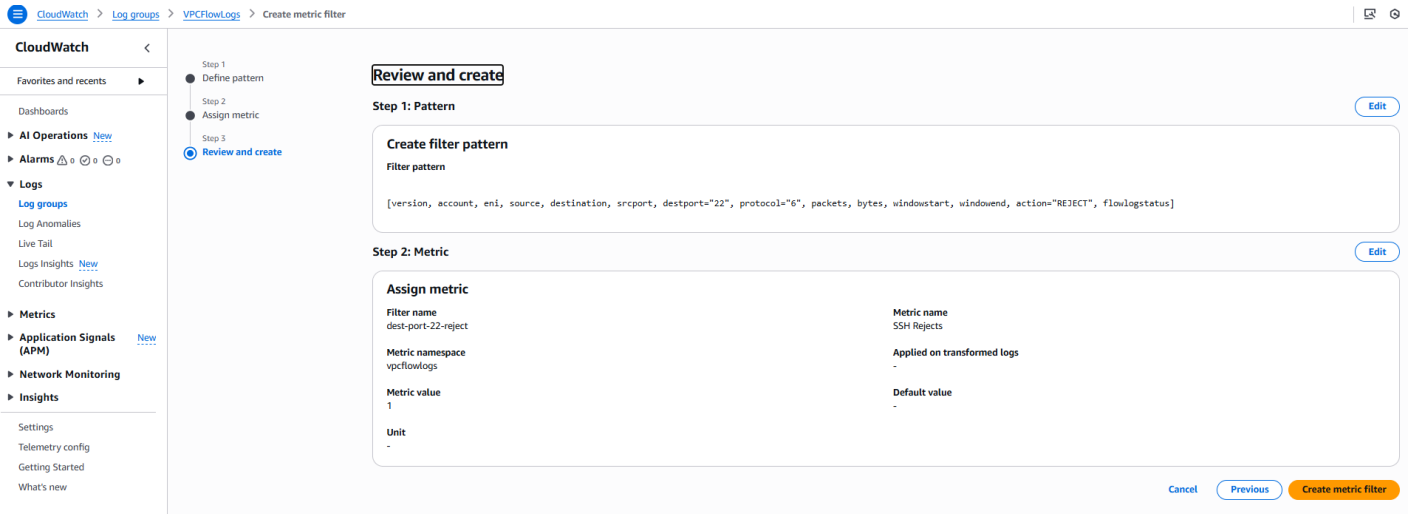


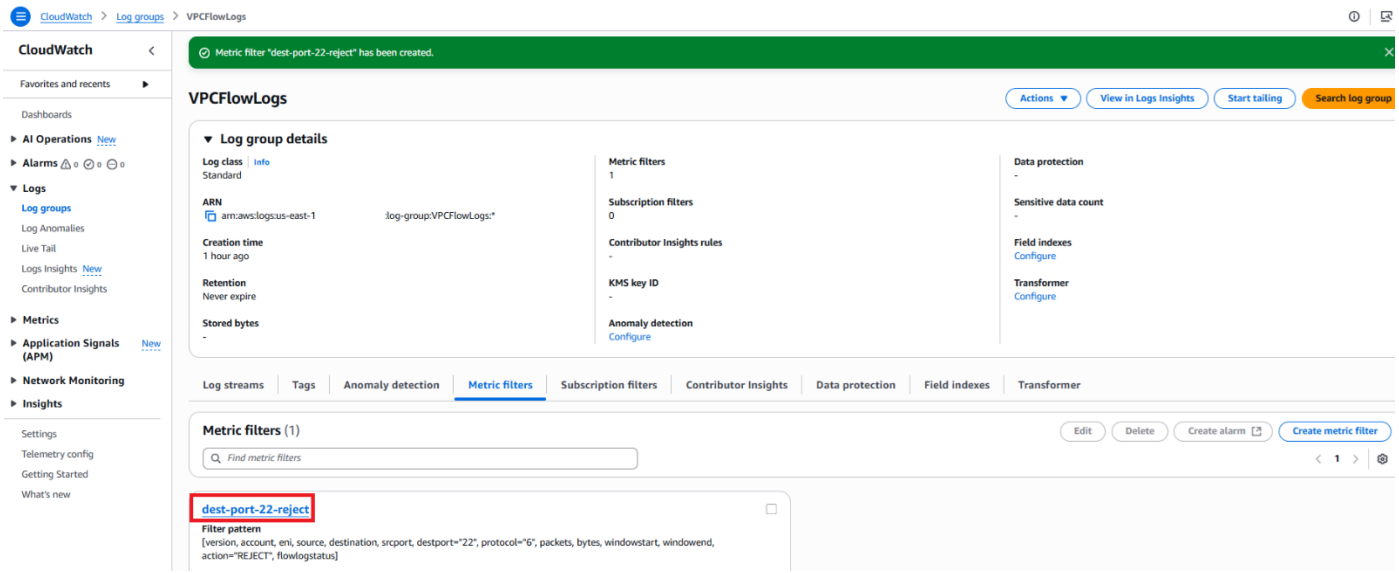
You will see we have 2 matches out of 4 events, so this is what we're looking for → so it works.

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.



13. Leave the other fields blank and click **Next**.
14. Review the metric details and then click **Create metric filter**.

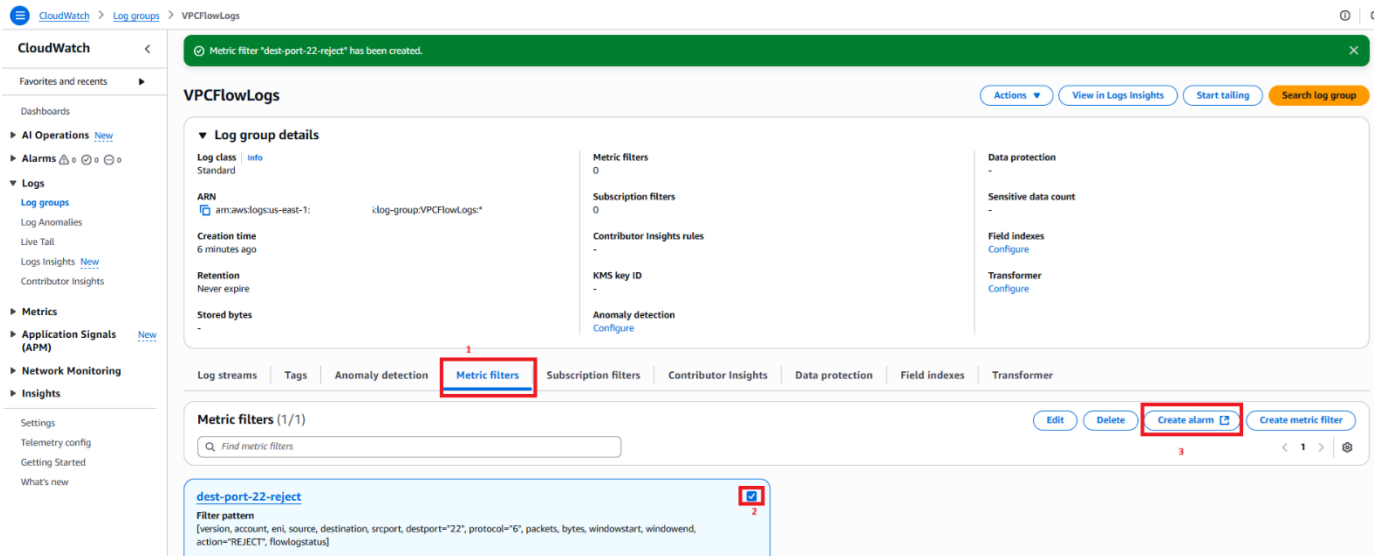




So that's been created, and you can see our filter here.

2.2. Create an Alarm Based on the Metric Filter

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



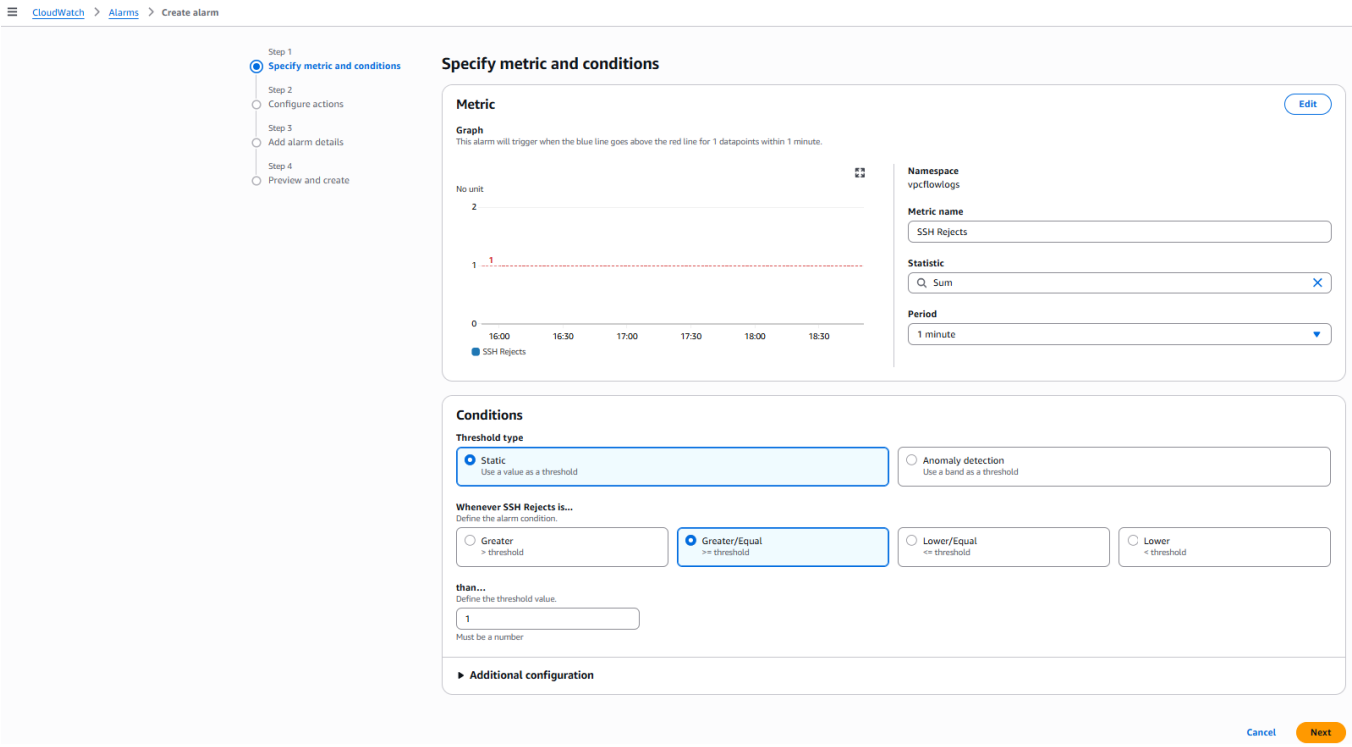
4. Specify the metric conditions:
    a. ***Period***: Use the dropdown to select **1 minute**.
    b. ***Threshold type***: Ensure that **Static** is selected.

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**.



6. Configure the alarm actions:

a. ***Alarm state trigger***: Ensure that **In alarm** is selected.

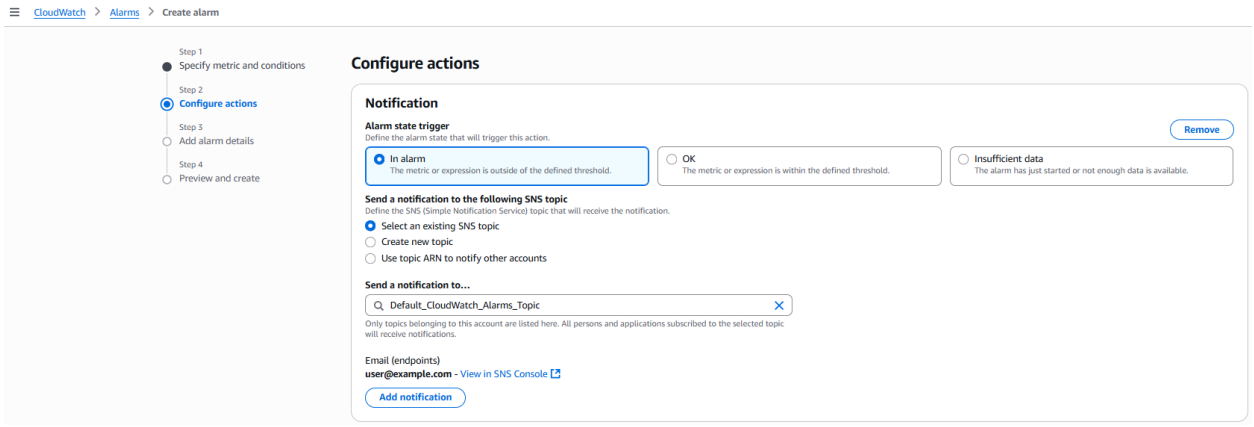
b. ***Send a notification to the following SNS topic***: Select **Create a new topic**.

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**.

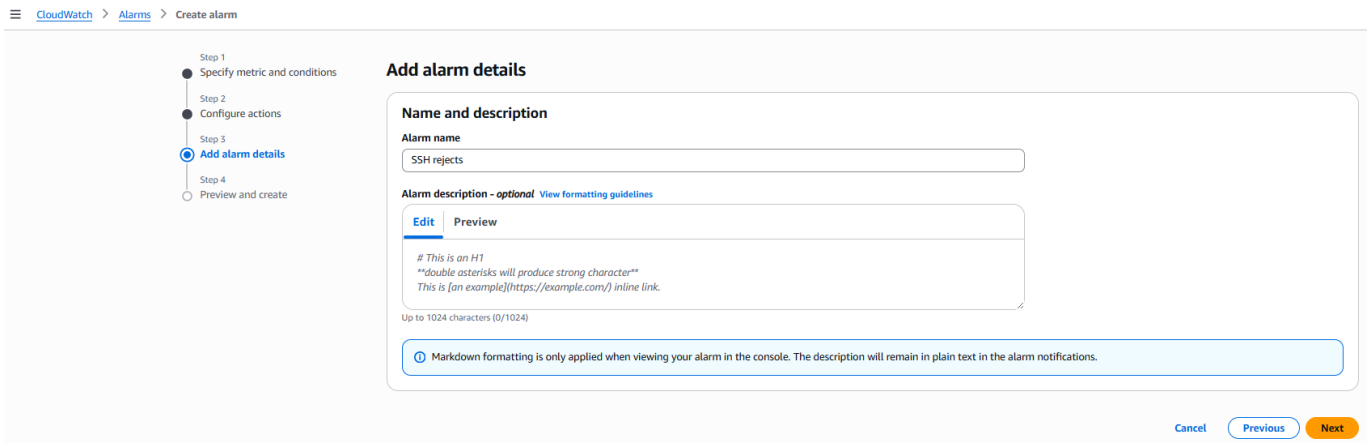
Note: If you enter your real email address, open your email inbox and click the **Confirm Subscription** link that you receive in the SNS email. If not, the alarm will work anyway, you just won't receive email notifications.

7. Click **Next**.

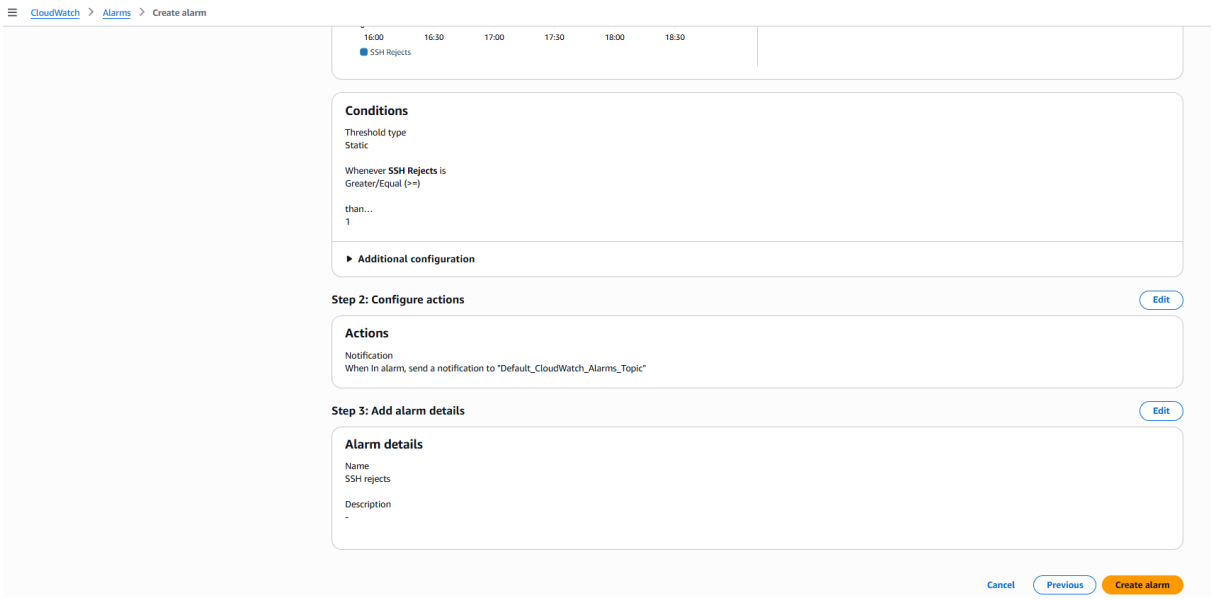


8. In the **Alarm name** field, enter **SSH rejects**.

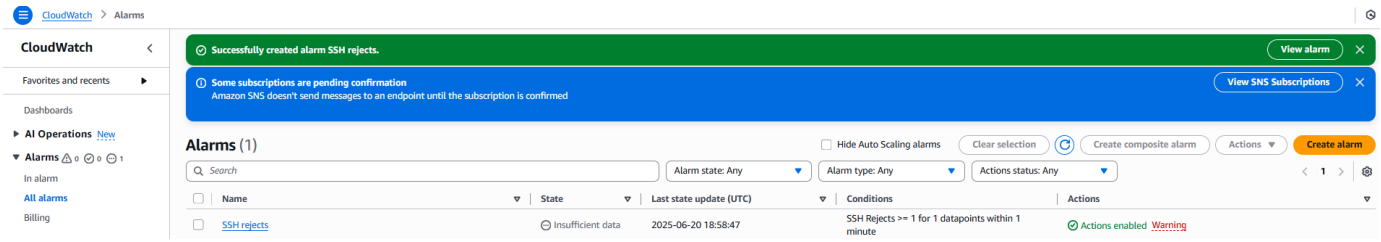
9. Click **Next**.



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



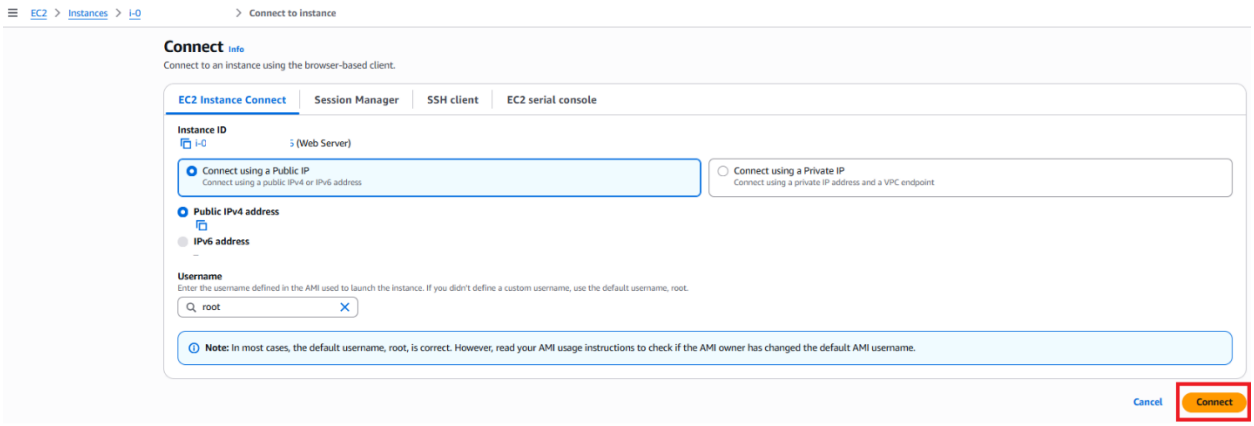
You will see immediately that the State is “Insufficient data”, and that simply means it’s still collecting data because the alarm is newly defined.



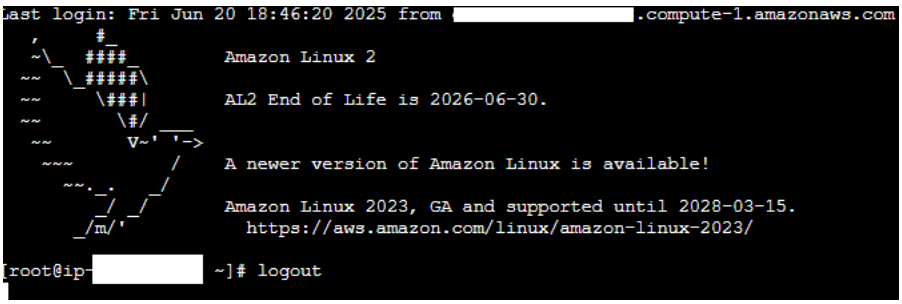
Now, we’re going to generate some more traffic so we can see this alert in action.

2.3. Generate Traffic for Alerts

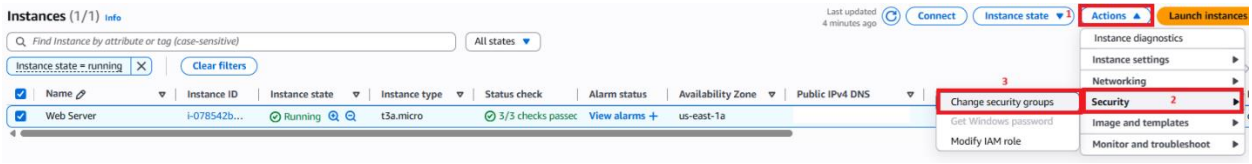
1. Navigate back to the terminal session and reconnect to the EC2 instance.



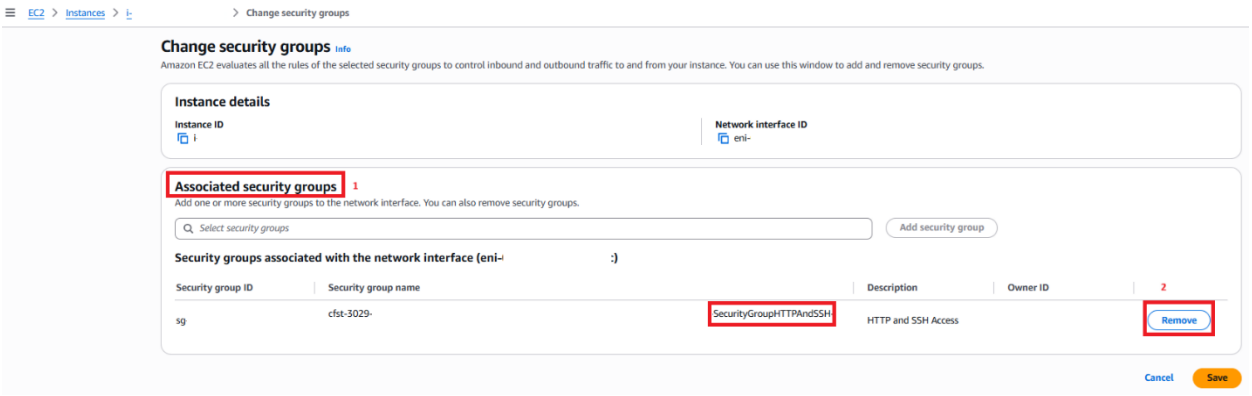
2. Exit the terminal: ***logout***



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**.

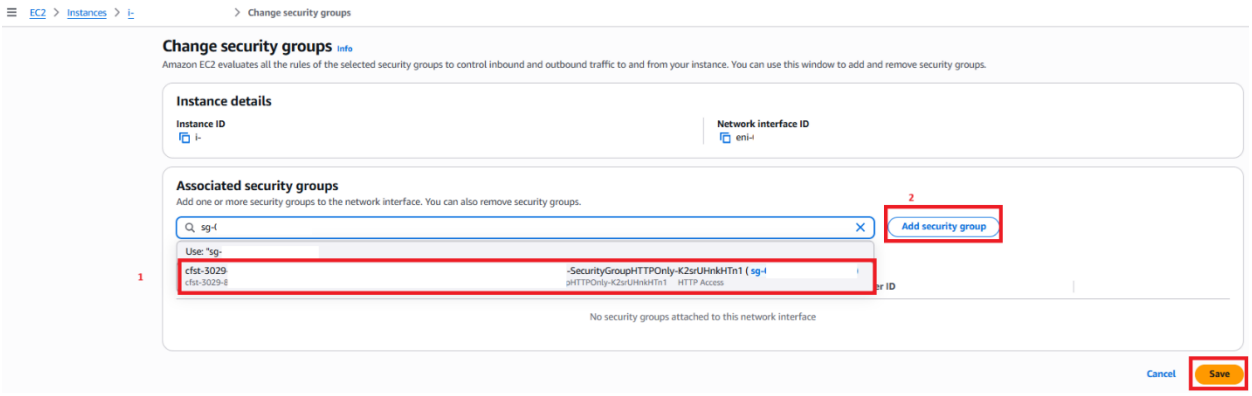


b. In the **Associated security groups** section, click **Remove** to the right of the security group details to remove the **SecurityGroupHTTPAndSSH** group.



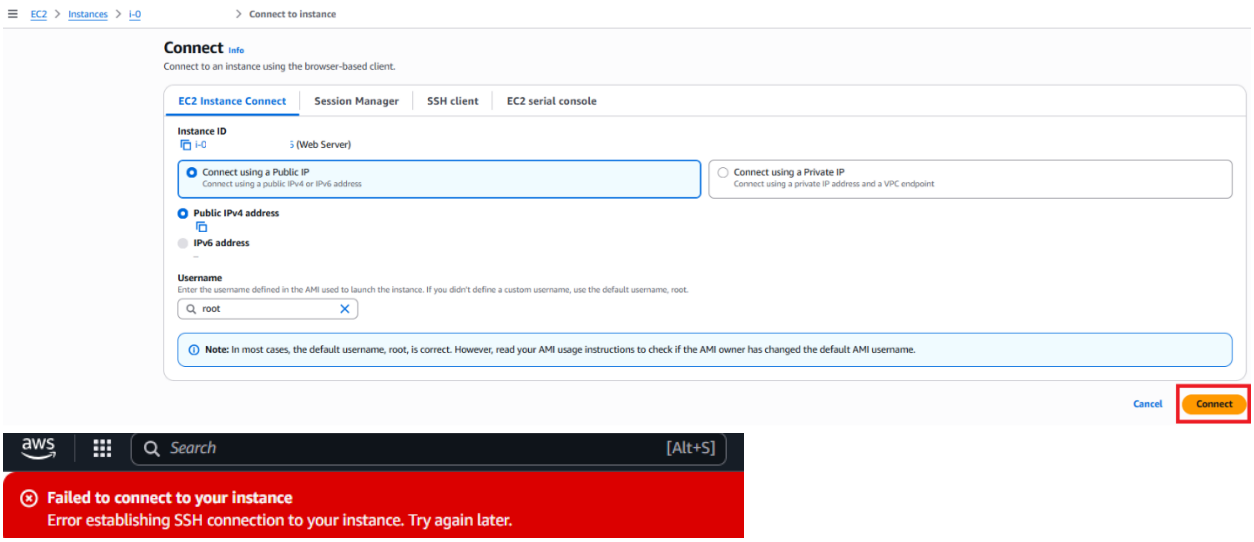
c. Use the search bar in the **Associated security groups** section to select the **SecurityGroupHTTPOnly** security group.

d. Click **Add security group**, and then click **Save**.



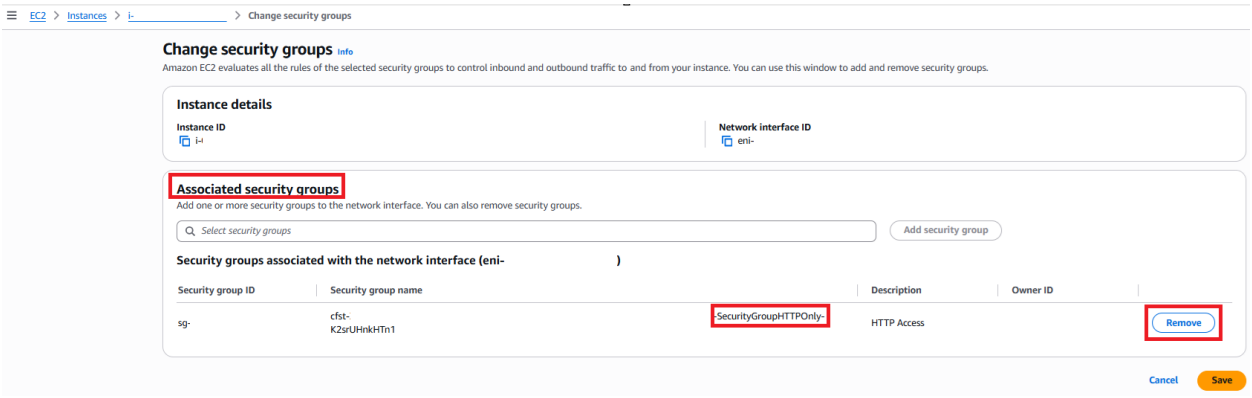
5. Navigate back to your terminal session and reconnect to the EC2 instance.

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

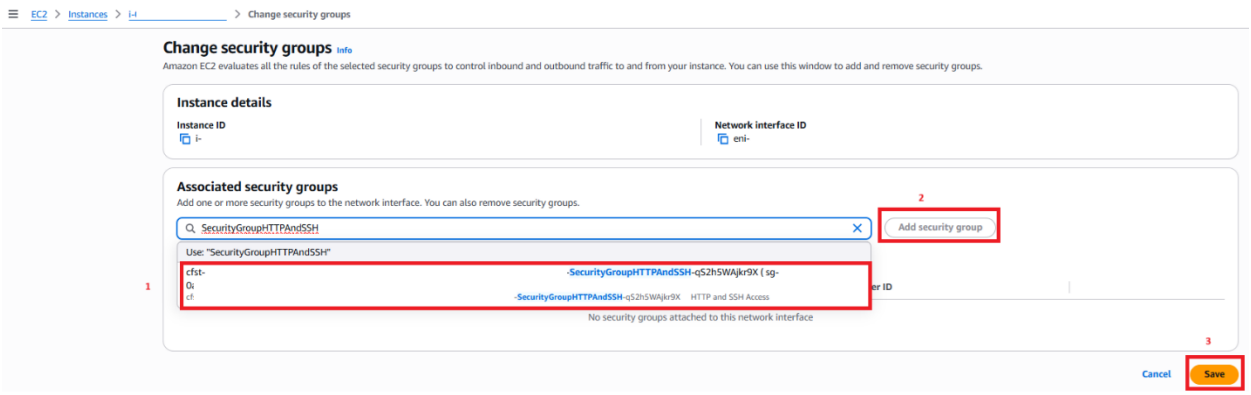


6. Navigate back to the **EC2 Management Console** tab.
7. Revert the EC2 security group back to **SecurityGroupHTTPAndSSH**:
    a. Ensure the **Web Server** instance is selected, and then use the **Actions** dropdown to select **Security → Change security groups**.

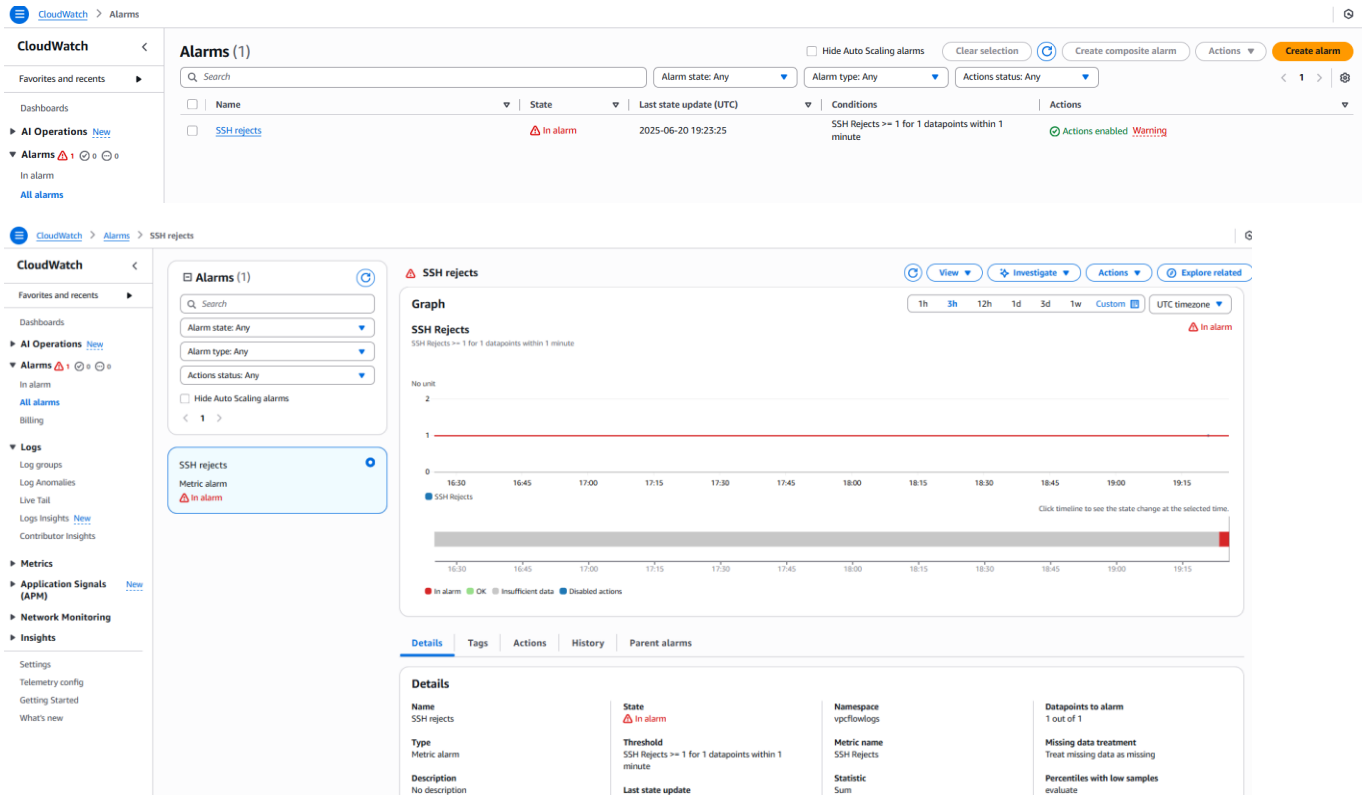
    b. In the **Associated security groups** section, click **Remove** to the right of the security group details to remove the **SecurityGroupHTTPOnly** group.



    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.

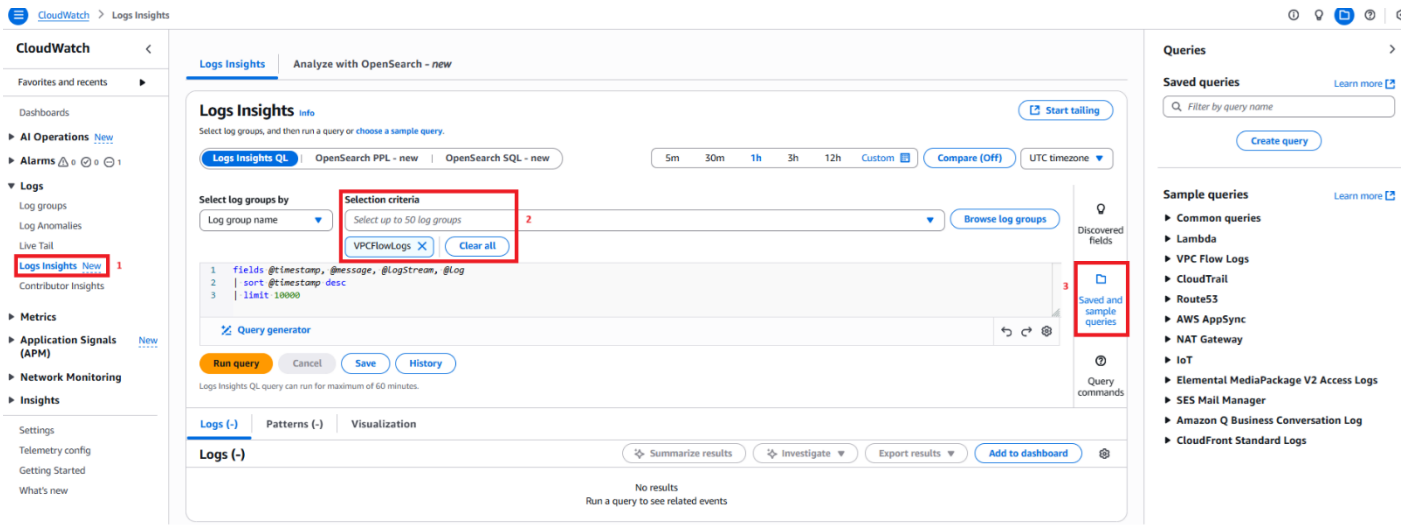


Note: If the alarm state still shows **Insufficient data**, wait another moment or two and then refresh the alarms details again.

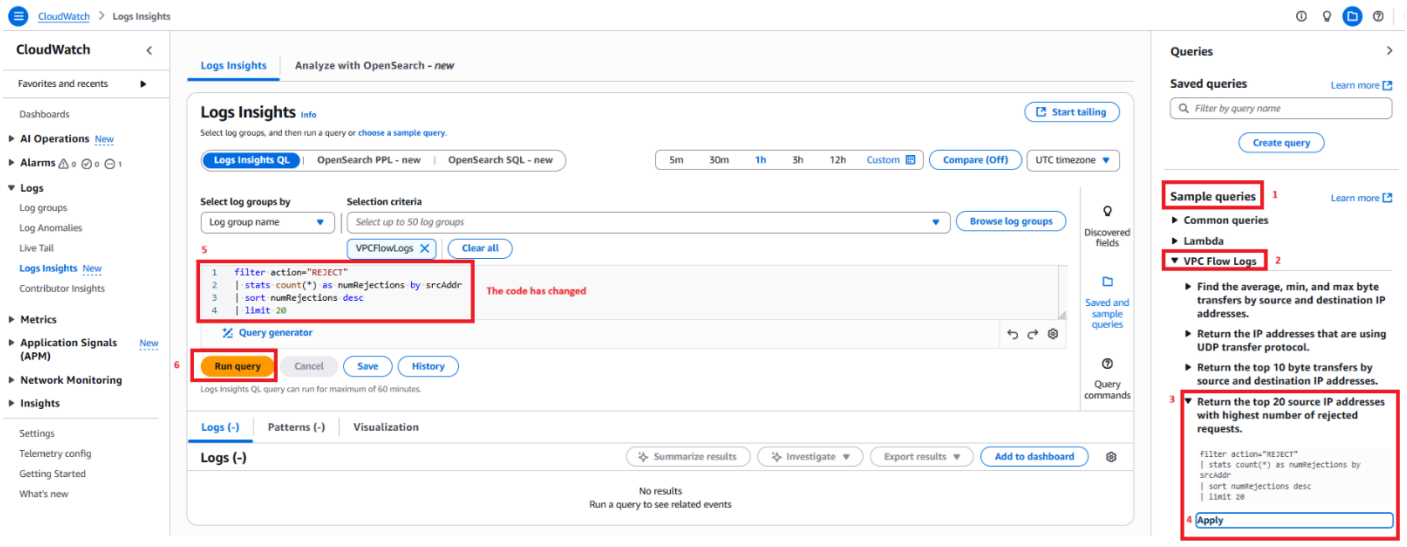
Now we need to do a little bit of analysis on our flow logs data.

3. Use CloudWatch Logs Insights

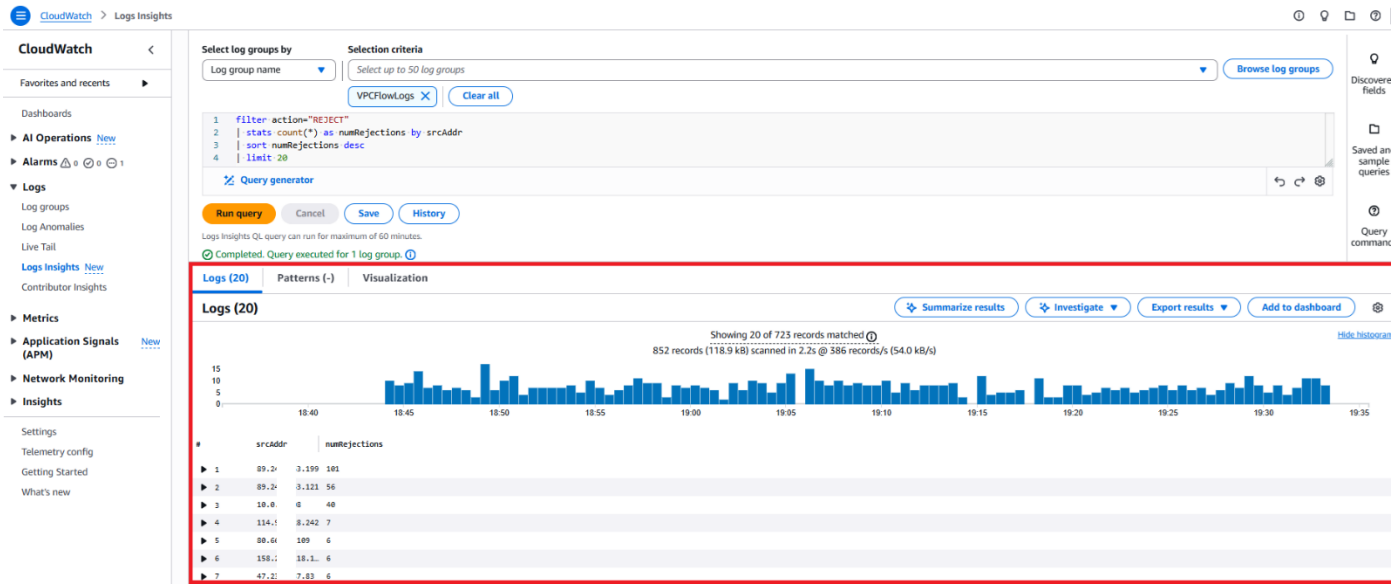
1. In the CloudWatch sidebar menu, navigate to **Logs** and select **Logs Insights**.
2. Use the **Select log group(s)** search bar to select **VPCFlowLogs**.
3. In the right-hand pane, select **Queries**.



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.



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



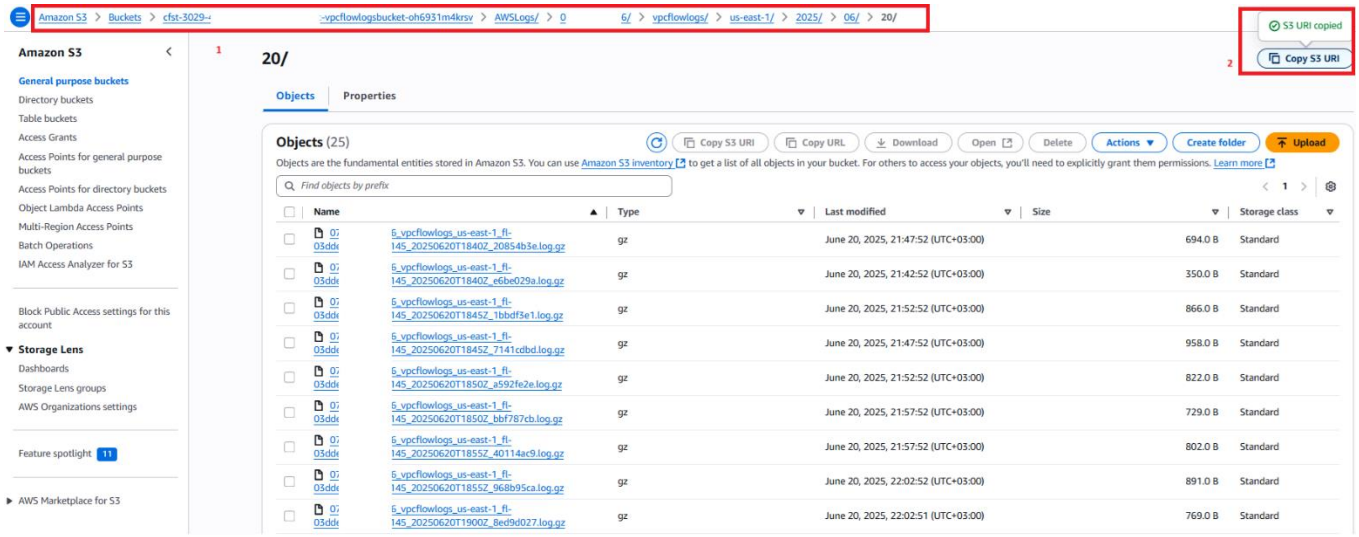
So, in CloudWatch we created a metric filter. We created a CloudWatch alarm, which provided notifications to a SNS topic that sent us an email when that condition was met. We also performed this ad hoc query based on the data in CloudWatch.

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

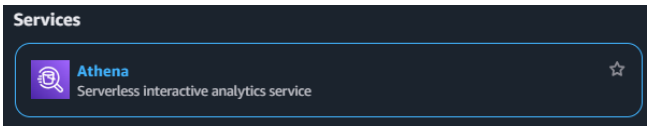
1. Navigate back to the **S3** browser tab and then navigate to your **Buckets**.
2. Select the provisioned bucket name to open it.
3. Select the **AWSLogs/** folder, and then continue opening the subfolders until you reach the **<DAY>** folder containing the logs.
4. In the top right, click **Copy S3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



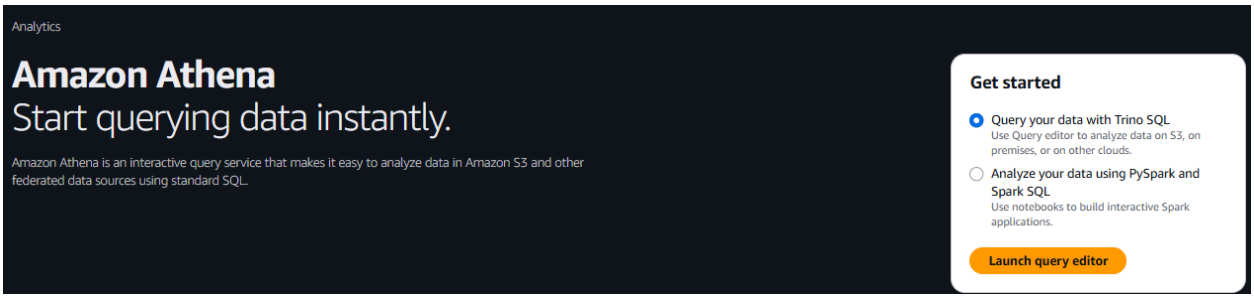
5. Paste the URI into a text file, as you'll need it shortly.

```
s3://cfst-3029-                    -vpcflowlogsbucket-oh6931m4krsv/AWSLogs/6                    /vpcflowlogs/us-
east-1/2025/06/20/
```

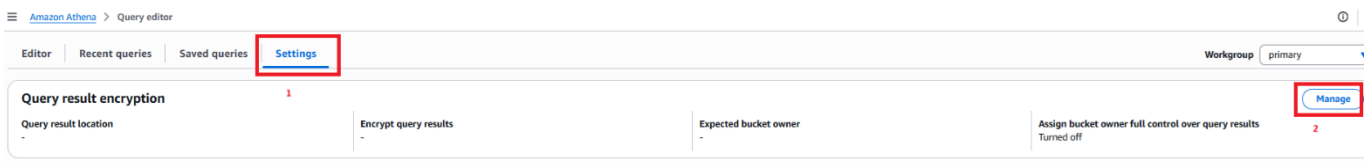
6. In a new browser tab, navigate to **Athena**.



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

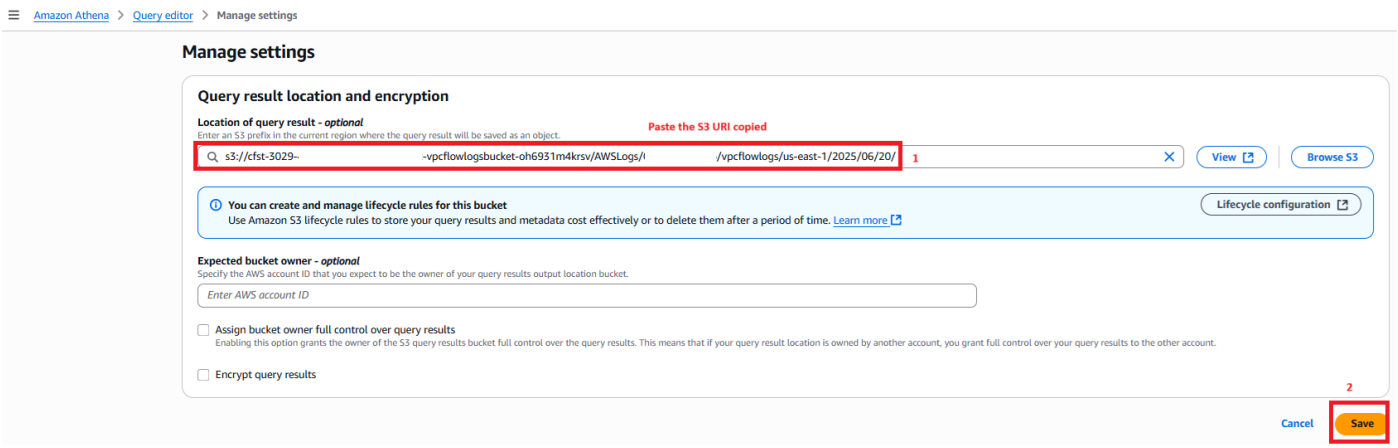


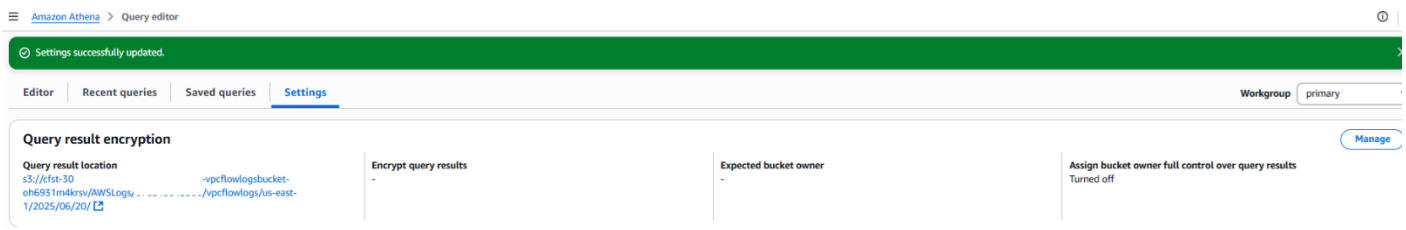
8. Select the **Settings** tab and then click **Manage**.



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

10. Click **Save**.





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)

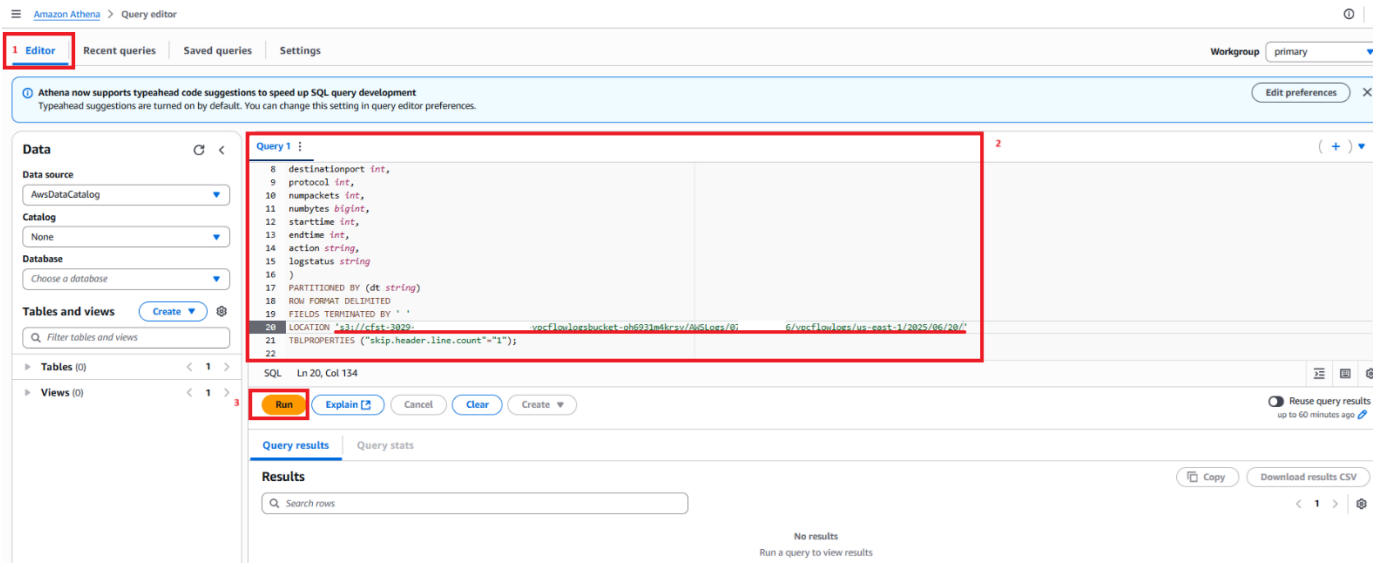
ROW FORMAT DELIMITED

FIELDS TERMINATED BY ' '

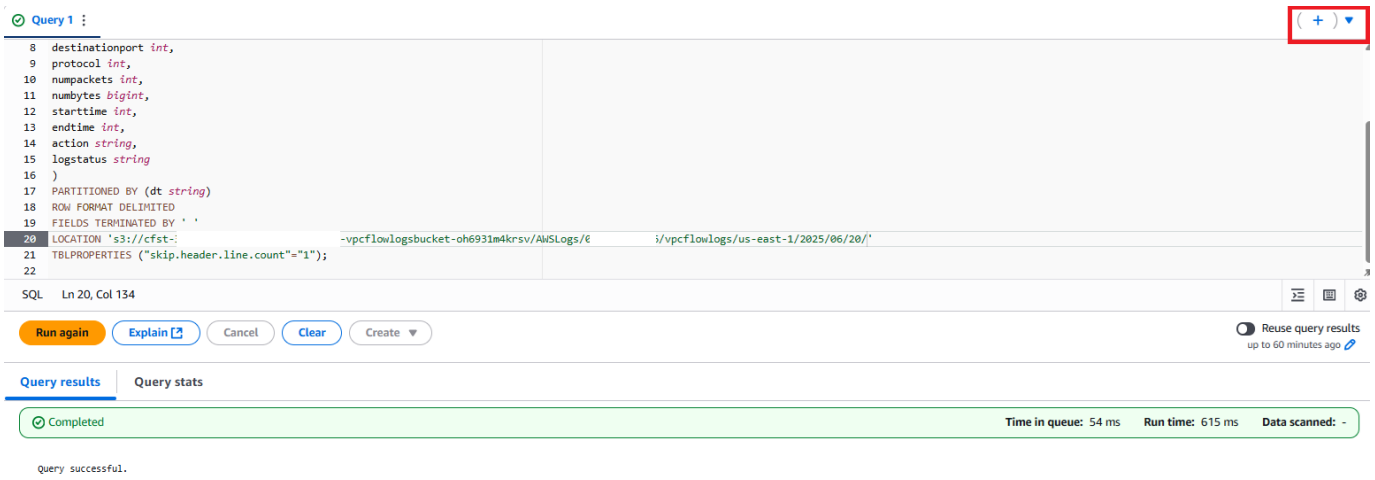
LOCATION 's3://{your_log_bucket}/AWSLogs/{account_id}/vpcflowlogs/us-east-1/'

TBLPROPERTIES ("skip.header.line.count"="1");

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



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



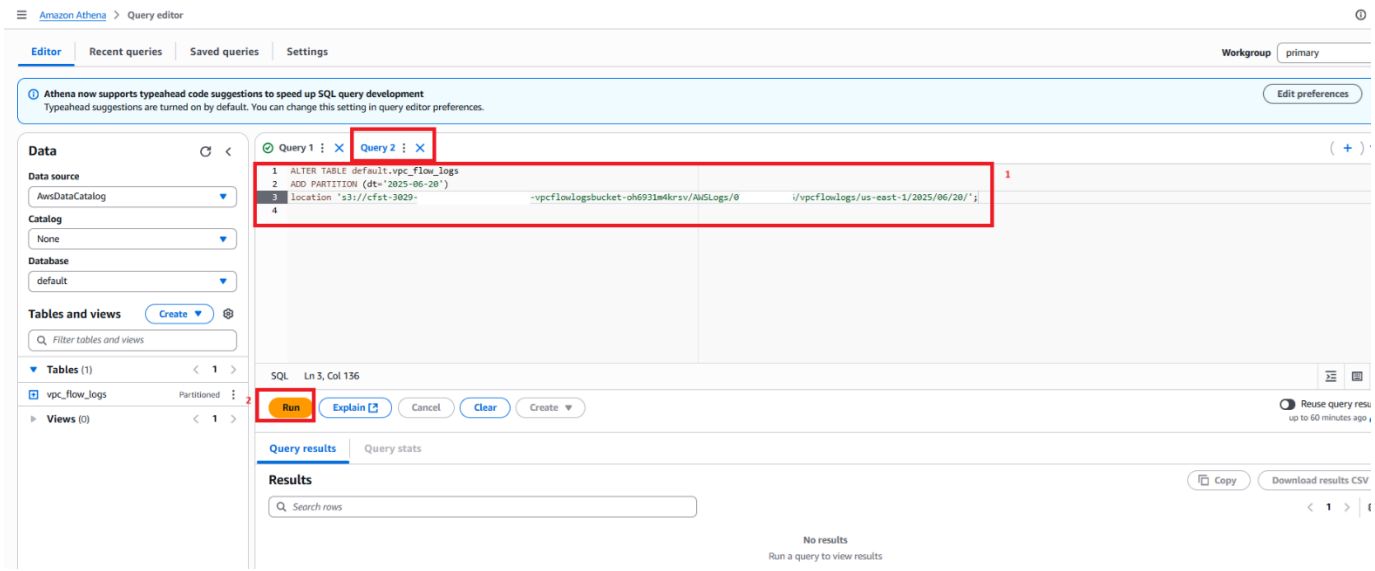
5. In the editor, paste the following query, replacing **YYYY-MM-DD** with the current date, and replacing the existing location with your copied S3 URI:

   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/';

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



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

8. In the editor, paste the following query:

   SELECT day_of_week(from_iso8601_timestamp(dt)) AS

   day,

   dt,

interfaceid,

sourceaddress,

destinationport,

action,

protocol

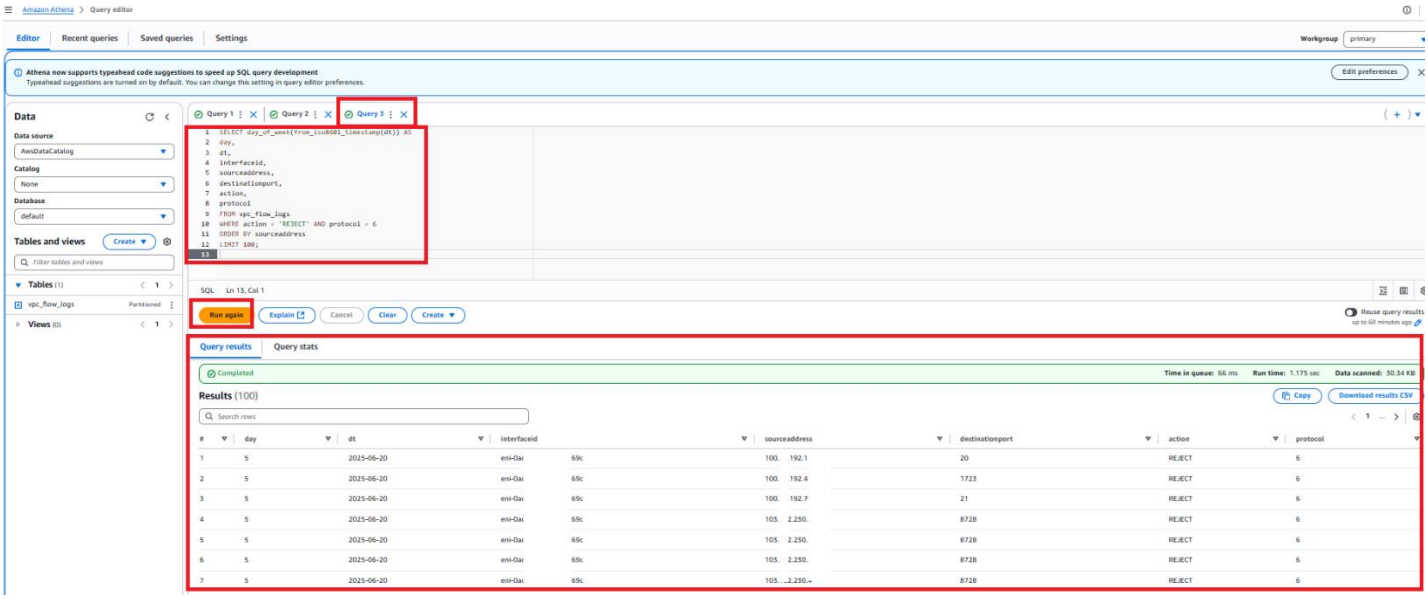
FROM vpc_flow_logs

WHERE action = 'REJECT' AND protocol = 6

ORDER BY sourceaddress

LIMIT 100;

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



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.