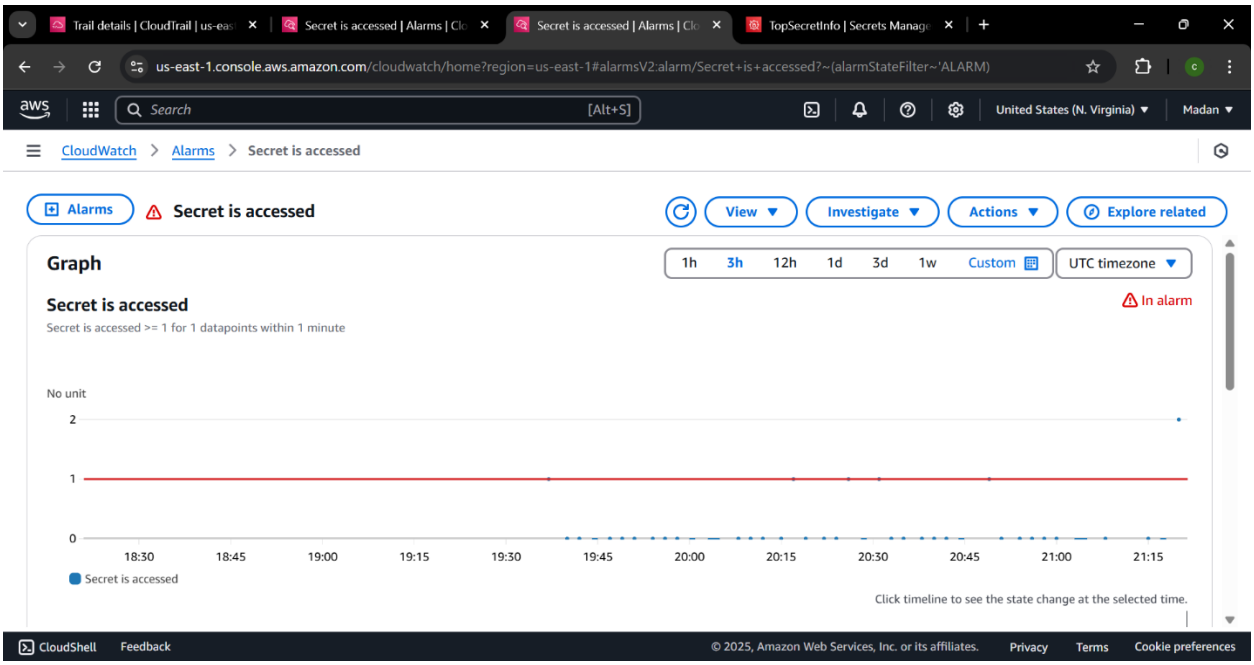
Build a Security Monitoring System



Introducing Today's Project! In this project,
we'll show you how to use CloudTrail to set up a monitoring system in AWS.

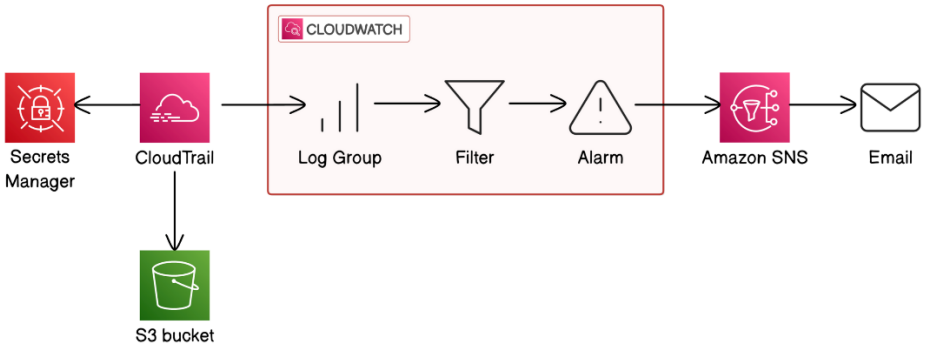
Tools and concepts

Services I used were:

1.CloudTrail
2.S3 Bucket
3.CloudWatch
4.Secrets Manager
5.Amazon SNS

Project reflection

This project took me approximately 2.5 hours The most challenging part was that the alarm changing to 'In Alarm' that took me long like more than 10 minutes and getting the final email ALARM: "Secret Key was accessed"

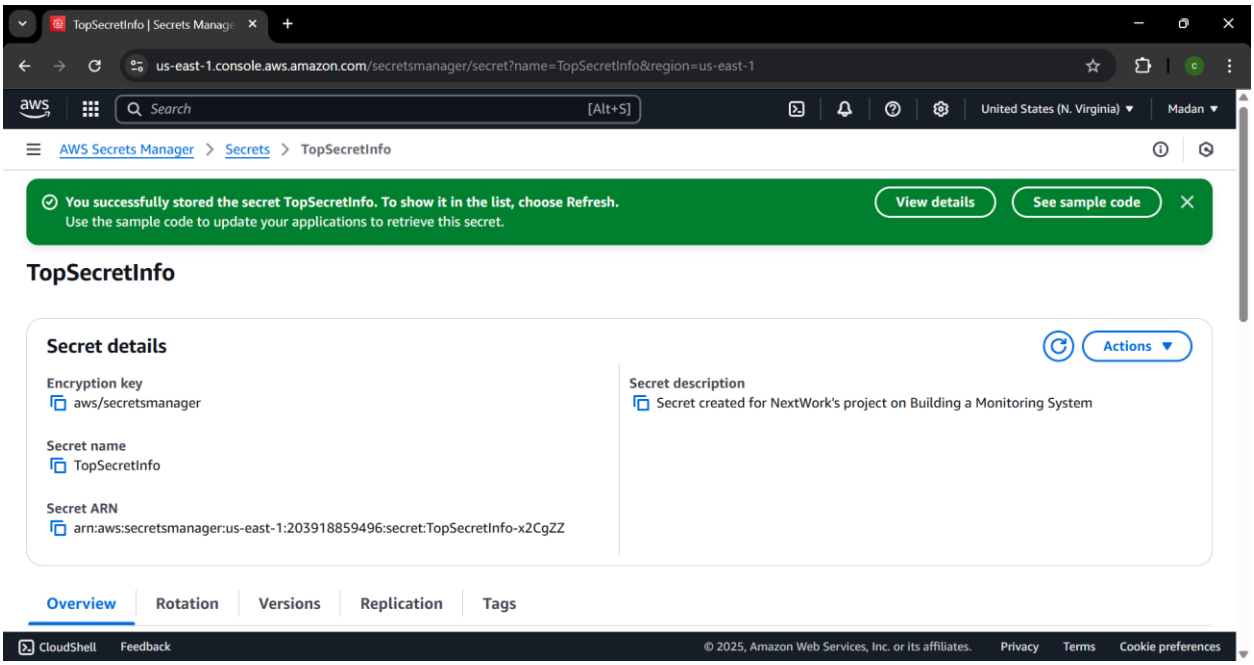


Architecture Diagram

Create a Secret

Secrets Manager helps you protect secrets, which are passwords, API keys, credentials and sensitive information. Instead of storing important credentials in your code or sharing them via email, you can tuck them safely away in Secrets Manager.

To set up for our project, we created a secret called TopSecretInfo in Secrets Manager. This secret contains a hot take from us... we mentioned that we need 3 coffees a day to function.



Set Up CloudTrail

AWS CloudTrail is a monitoring service - think of it as an activity recorder throughout your AWS account. It documents every action taken, like who did what, when they did it, and where they did it from.

CloudTrail events include types like management, data, insights and network activity events. In this project we set up our trail to track management events because accessing a secret falls into that category. It is not a data event (which captures high volume actions performed on resources) because all management events are free to track (and AWS lets us track security operations like this for free)

Read vs Write Activity

Read API activity involves accessing, reading, opening a resource. Write API activity involves creating, deleting, updating a resource. For this project we ticked both to learn about both types of activities, but we only need the Write activity (accessing a secret is considered a Write activity because of its importance.

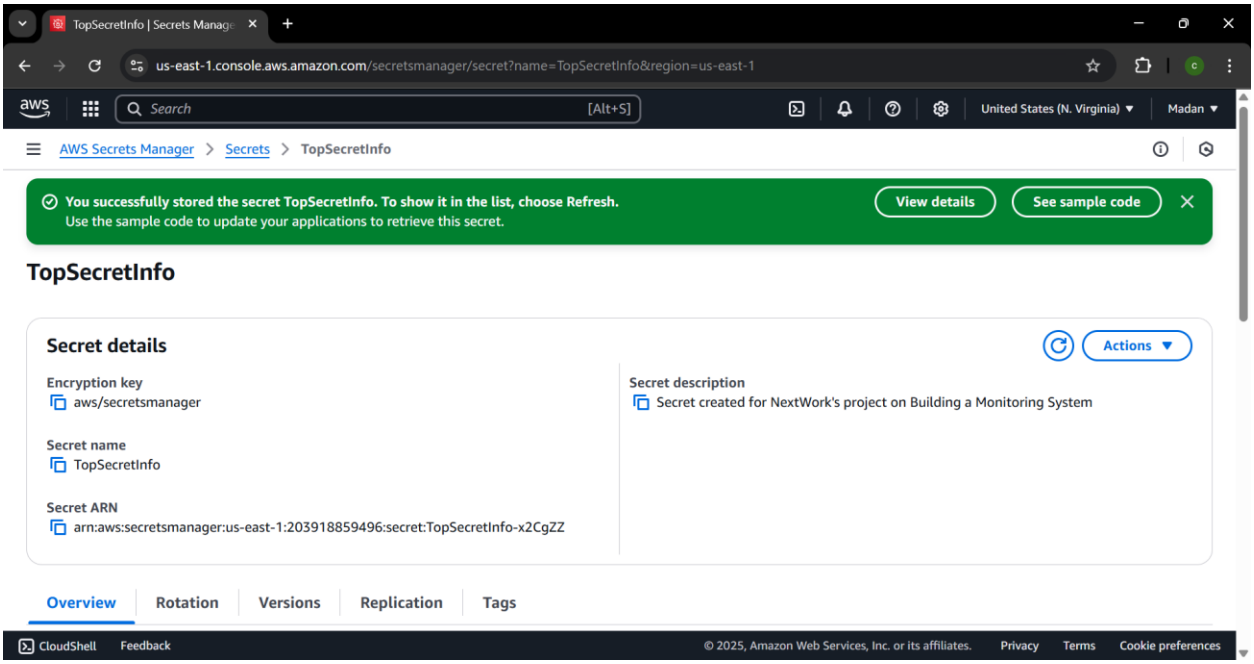
Verifying CloudTrail

We retrieved our secret in two ways:

First through the secrets manager console where we could easily just select a "Return secret value" button.

Second way was using the AWS CLI i.e. running a get-secret-value in CloudShell.

To analyze my CloudTrail events i.e. see the event where we get our secret's value, we visited the event history in CloudTrail. We found that there is a GetSecretValue event tracked regardless of whether we did it over the CLI or the console. This tells us that CloudTrail can see and track when we open our Secrets Manager key.

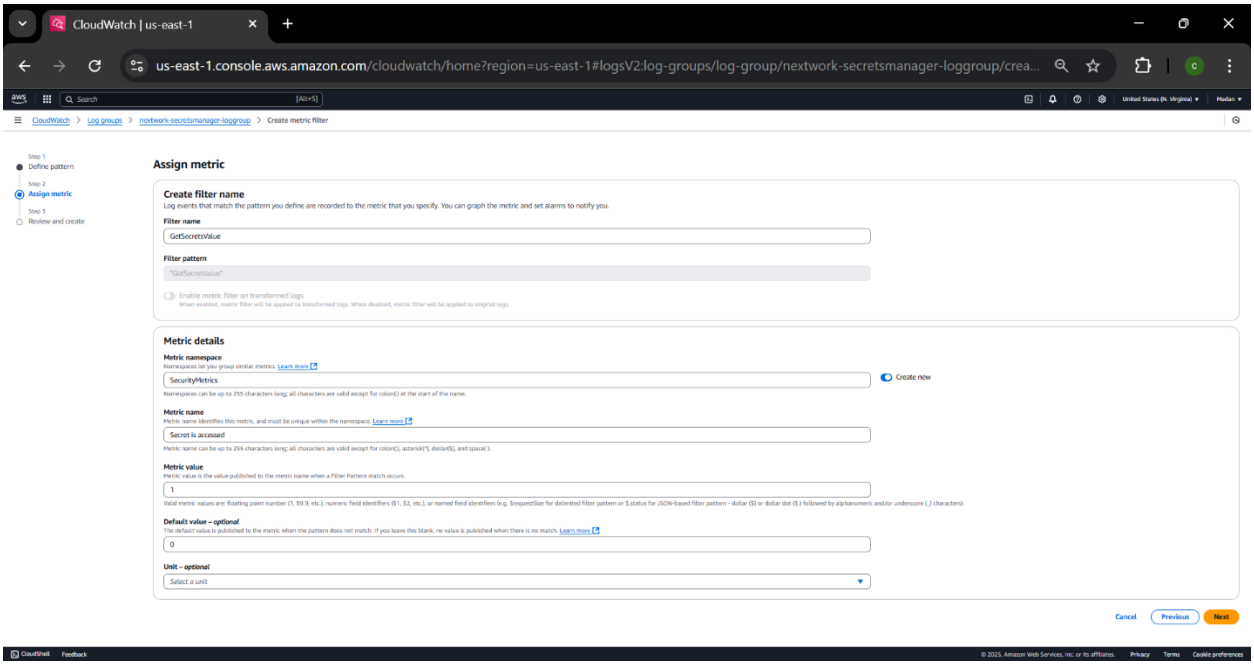


CloudWatch Metrics

CloudWatch Logs is a monitoring service that brings together logs from other AWS services (including CloudTrail) to help us analyze and create alarm for. It's important for monitoring because you get to create insights and get alerted based on events that happen in your account.

CloudTrail's Event History is useful for quickly reading(management) events that happened in the last 90 days, while CloudWatch logs are better for combining and analyzing logs from different sources, accessing logs for longer than 90 days, and advanced filtering.

A CloudWatch metric is a specific way that we count or track events that happen in. When setting up a metric the metric value represents how we increment or 'count' an event when it passes our filters (in our case, we want to increment our metric value by 1 whenever our secret is accessed). Default value is used when the event that we are tracking that does not occur.

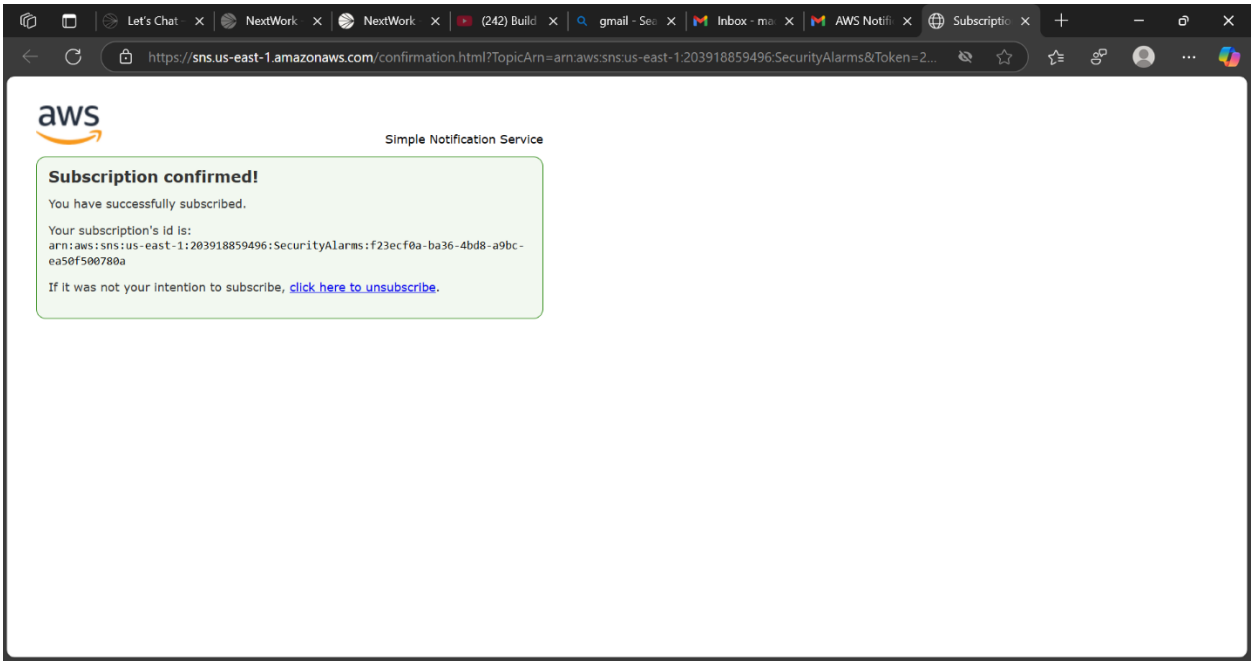


CloudWatch Alarm

A CloudWatch alarm is a feature and alert system in CloudWatch that's designed to "go off" i.e. Indicate when certain conditions have been met in our log group. We set my CloudWatch alarm threshold to be about how many times the GreatSecretValue event happen in a 5-minute period so the alarm will trigger when the average number of times is above 1.

An SNS (Simple Notification Service) topic is like a broadcast channel for your notifications. First, you create the channel (topic), then you invite subscribers (such as your email), and finally, you send messages to the topic. SNS automatically delivers that message to all subscribers.

AWS requires email confirmation because it would not automatically start emailing addresses that we subscribe to an SNS topic. This helps prevent any unwanted subscriptions for recipients (i.e. people who are receiving those emails).



Troubleshooting Notification Errors

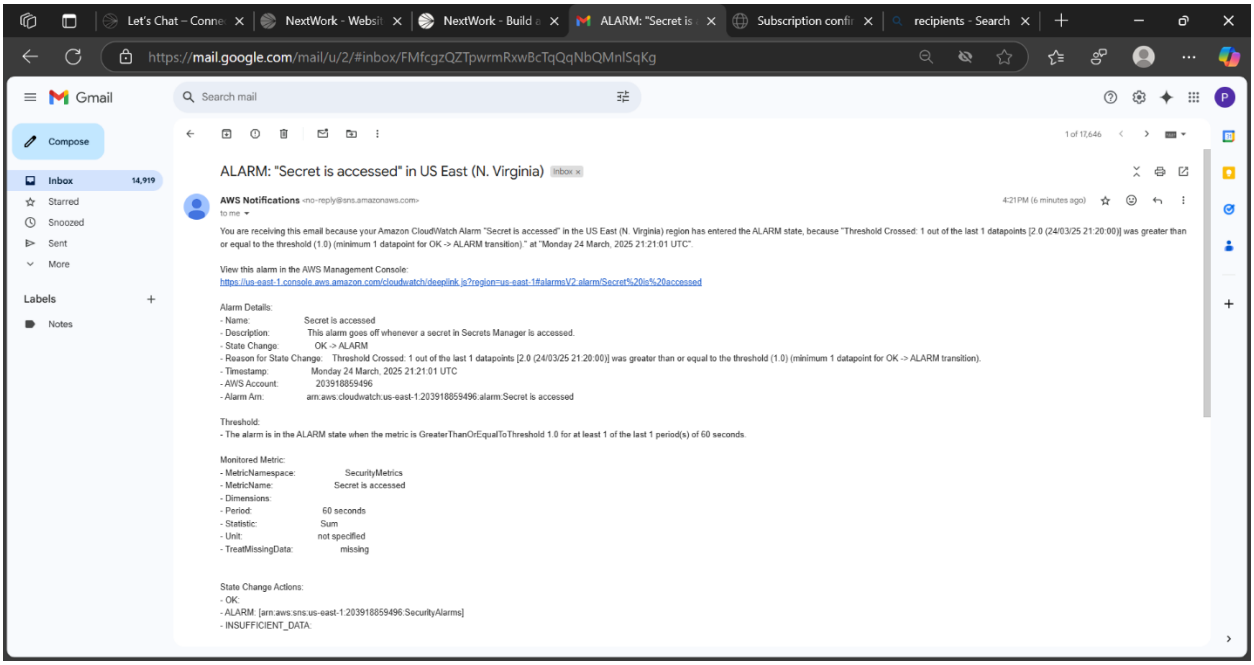
To test our monitoring system, we opened and accessed our secret again! The results weren't successful- we didn't get any emails/notifications about our secret getting accessed!

When troubleshooting the notification issues, we investigated every single part of our monitoring system -whether CloudTrail is picking up on events that are happening when we access our secret, whether CloudTrail is sending logs to CloudWatch, whether the filter is accidentally rejecting the correct events, whether the alarm gets triggered, whether the triggering the alarm sends an email.

We initially didn't receive an email because CloudWatch was configured to use the wrong threshold instead of calculating the AVERAGE number of times a secret was accessed in a time, it should've been the SUM.

Success!

To validate that our monitoring system can successfully detect and alert when my secret is accessed. I have checked my secrets value one more time. We received an email within 1-2 minutes of the event. Our alarm in CloudWatch is also in the 'in alarm' state.



Comparing CloudWatch with CloudTrail Notifications

In a project extension, we enabled SNS notification delivery in CloudTrail because this lets us evaluate CloudTrail vs CloudWatch for notifying us about events like our secret getting access.

After enabling CloudTrail SNS notifications, my inbox was very quickly filled with new emails from SNS (as if we notified by CloudTrail). In terms of the usefulness of these emails we thought that we are receiving LOTS (it's a little over whelming) and the logs themselves don't show what happened in terms of management events that occurred. We only see that only new logs have been stored in our bucket.