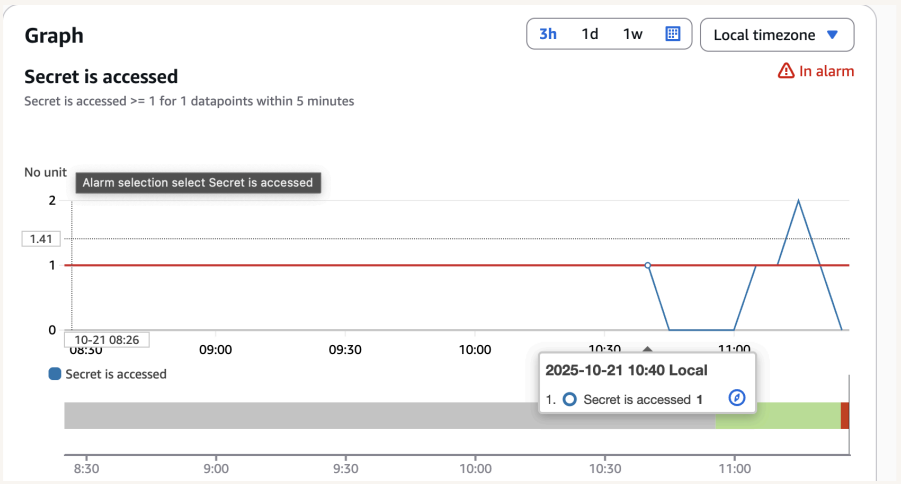
nextwork.org

Build a Security Monitoring System

harshit katheria



Introducing Today's Project!

In this project, I will be working and building a Security Monitoring System using AWS core monitoring services like Cloudtrail to log all the api access events, Cloudwatch to log all the logs whenever a action is performed on the AWS account and finally the SNS service, which notifies about the same.

Tools and concepts

Services I used were AWS [SecretsManager, CloudTrail, CloudWatch, SNS]. Key concepts I learnt include how to put in use of secretsManager, how to setup cloudtrail to record activity being done in an aws account, setting up cloudwatch alarms to get notified about the activity through SNS service, which delivers the notification to me through mail.

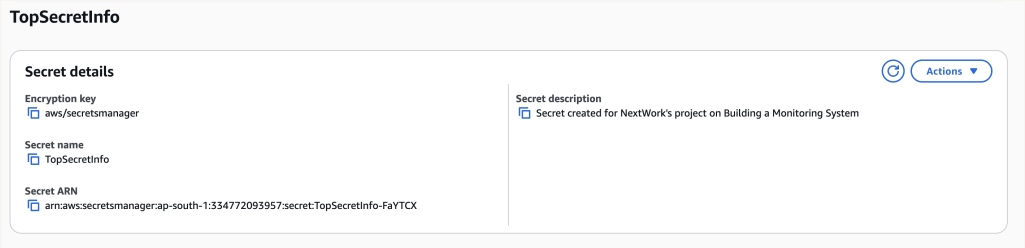
Project reflection

This project took me approximately 2-3 hours to complete. The most challenging part was debugging why i was not receiving the notification after the secret was accessed. It was most rewarding to finally receive the notification after debugging and setting everything up correctly.

Create a Secret

Secrets Manager is a AWS service, which helps in storing the private info like DB's password, API Keys, or Access and Secret Access Keys completely secure and isolated. Instead of storing them directly in our repos or in the code, we use Secrets manager service, which is a more viable option.

To set up for my project, I created a secret called... that contains...To set up for my project, I created a secret called "TopSecretInfo" which for now contains a top secret about nutritional information of a certain food. I also gave it a brief description explaining whats the secret is for. I used a Key-Value pair to store my secret, where the the Key is "The Secret Is" and Value hold the secret.



Set Up CloudTrail

Cloudtrail is used to monitor as well to log/record/track each and every api call happening in the AWS accounts. A trail on the other hand, is telling CloudTrail that hey, i want to record this activity happening here and then store the same on this particular location.

Different types of CloudTrail events and API activities are : 1) Management Events 2) Data Events 3) Insights Events 4) Network Activity Events

Read vs Write Activity

Read API activity happens when I view information but don't make any changes. For example, if I'm just listing my S3 buckets, describing my EC2 instances, or viewing the metadata about a secret (like its name or description) without revealing its value, that's a Read API activity. Write API activity occurs when I make changes. This includes creating, deleting, or modifying resources. What's particularly interesting for my project is that retrieving the actual value of a secret is also considered a Write API activity. Even though I'm not changing the secret's content when I retrieve its value, AWS classifies this as a 'Write' operation for security monitoring purposes. This is because accessing and decrypting a sensitive secret is a critical security event that needs to be tracked.

Verifying CloudTrail

I retrieved the secret in two ways: First through by going to the SecretsManager in the console itself, and Second by using AWS CloudShell and running the command "aws secretsmanager get-secret-value --secret-id "TopSecretInfo" --region ap-south-1", which returned the secret in a JSON body.

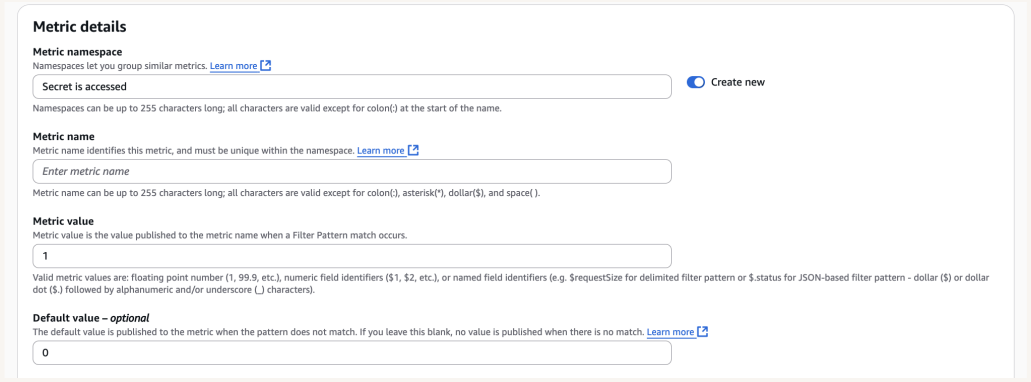
To analyze my CloudTrail events, I navigated to the 'Event History' tab. There, I filtered by 'Event Source' and entered 'secretsmanager.amazonaws.com' to focus specifically on Secrets Manager activities. I then observed multiple 'GetSecretValue' events listed under 'Event Name'. This confirmed that my secret was indeed accessed and its value retrieved.

CloudWatch Metrics

Amazon CloudWatch Logs is an AWS service that acts as a centralized hub for all my logs from various AWS services, including CloudTrail, and even my own applications. It helps me gather, monitor, and analyze these logs from a single converging point. It's incredibly important for my monitoring because once my logs are in CloudWatch, I can do several powerful things: Look up and analyze logs: I can easily search through my logs to troubleshoot issues or understand activity patterns. Set up events and alarms: I can create metric filters and alarms based on specific log patterns (like someone accessing my secret, as I'm doing in this project). Real-time monitoring: I get immediate insights into the health and performance of my applications and infrastructure Automated responses: I can trigger actions or notifications based on alarm states

The Key difference b/w storing logs in cloudTrail vs cloudWatch is that cloudWatch allows us to set custom alerts and alarms based on specific events, and cloudwatch also allows to retain the logs for as long as we want whereas cloudtrail only stores them for 90days.

The purpose of Metric value is that it helps in knowing how many times a value which is present in our metric filter gets recorded, i've set up it to 1, so that whenver a new match is found it increments it by 1. In the same way a default value is what gets recorded when out metric filter doesnt find any matches. Setting it to 0 so that time periods with no secret access show up as zero on our charts, rather than not showing up at all.


Metric details
Metric namespace
Namespaces let you group similar metrics. Learn more
Secret is accessed
Create new
Namespaces can be up to 255 characters long; all characters are valid except for colon(:) at the start of the name.
Metric name
Metric name identifies this metric, and must be unique within the namespace. Learn more
Enter metric name
Metric name can be up to 255 characters long; all characters are valid except for colon(:), asterisk(*), dollar($), and space( ).
Metric value
Metric value is the value published to the metric name when a Filter Pattern match occurs.
1
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).
Default value – optional
The default value is published to the metric when the pattern does not match. If you leave this blank, no value is published when there is no match. Learn more
0

CloudWatch Alarm

The threshold setting in my cloudwatch alarm is set to >=1, meaning it will scan through the metric filter in the span of 5 mins, and if the count is >=1, then it will trigger a said alarm about it.

I created an SNS topic along the way. An SNS (Simple Notification Service) topic is like a broadcast channel for the notifications. First, i created the channel (topic), then invited subscribers (such as my email), and finally, send messages to the topic. SNS automatically delivers that message to all subscribers.

AWS requires email confirmation to confirm if its actually us that have set up the SNS and alarm

aws

Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:
`arn:aws:sns:ap-south-1:334772093957:SecurityAlarms:a3bd094f-88b1-46b6-a71c-9fe385c90ce1`

If it was not your intention to subscribe, click here to unsubscribe.

Troubleshooting Notification Errors

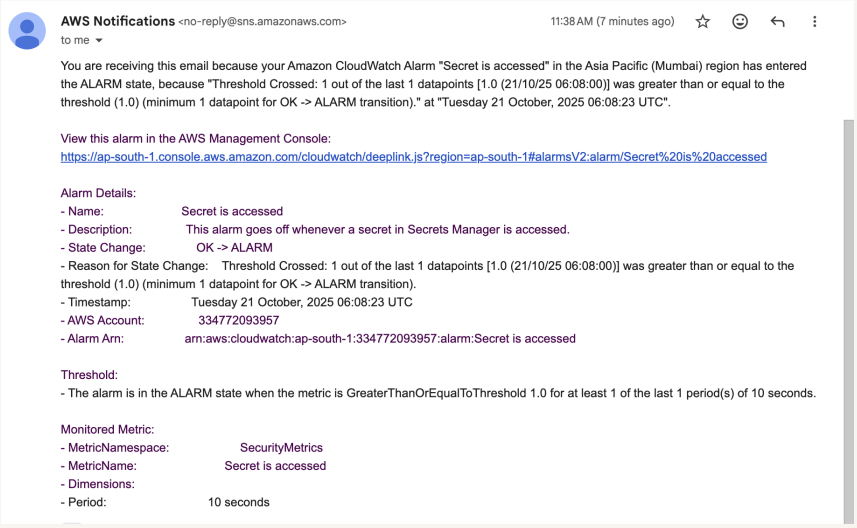
To test my monitoring system, I access my secret in the secretsManager and was hoping to receive the notification about the same, but didnt. Gonna check it further to find the root cause of the issue.

I troubleshooted the error by: Checking CloudTrail Event History to ensure the GetSecretValue event was recorded. Verifying CloudTrail Log Delivery to confirm logs were sent to CloudWatch. Testing the CloudWatch Metric Filter with sample logs to ensure it correctly identified the pattern. Reviewing the CloudWatch Alarm Configuration, including the statistic and threshold. Confirming the Amazon SNS Subscription was active and receiving notifications. It turned out my SNS notification service wasn't enabled, and the statistic in the CloudWatch alarm needed to be set to Average.

I initially didn't receive an email before because SNS notification service wasn't enabled, and the statistic in the CloudWatch alarm needed to be set to Average

Success!

To validate that my monitoring system works, is by accessing the secret and then checking the cloudtrail to see if the log activity has been captured or not. Once i confirmed it, i went to look cloudwatch if my metric filter was in alarm state or not, when i confirmed this as well, i checked my inbox to find that yes, i've received the notification about the same.



AWS Notifications <no-reply@sns.amazonaws.com> 11:38 AM (7 minutes ago)
to me

You are receiving this email because your Amazon CloudWatch Alarm "Secret is accessed" in the Asia Pacific (Mumbai) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [1.0 (21/10/25 06:08:00)] was greater than or equal to the threshold (1.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Tuesday 21 October, 2025 06:08:23 UTC".

View this alarm in the AWS Management Console:
https://ap-south-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=ap-south-1#alarmsV2:alarm/Secret%20is%20accessed

Alarm Details:
- Name: Secret is accessed
- Description: This alarm goes off whenever a secret in Secrets Manager is accessed.
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [1.0 (21/10/25 06:08:00)] was greater than or equal to the threshold (1.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Tuesday 21 October, 2025 06:08:23 UTC
- AWS Account: 334772093957
- Alarm Arn: arn:aws:cloudwatch:ap-south-1:334772093957:alarm:Secret is accessed

Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanOrEqualToThreshold 1.0 for at least 1 of the last 1 period(s) of 10 seconds.

Monitored Metric:
- MetricNamespace: SecurityMetrics
- MetricName: Secret is accessed
- Dimensions:
- Period: 10 seconds

The place to learn & showcase your skills

Check out nextwork.org for more projects