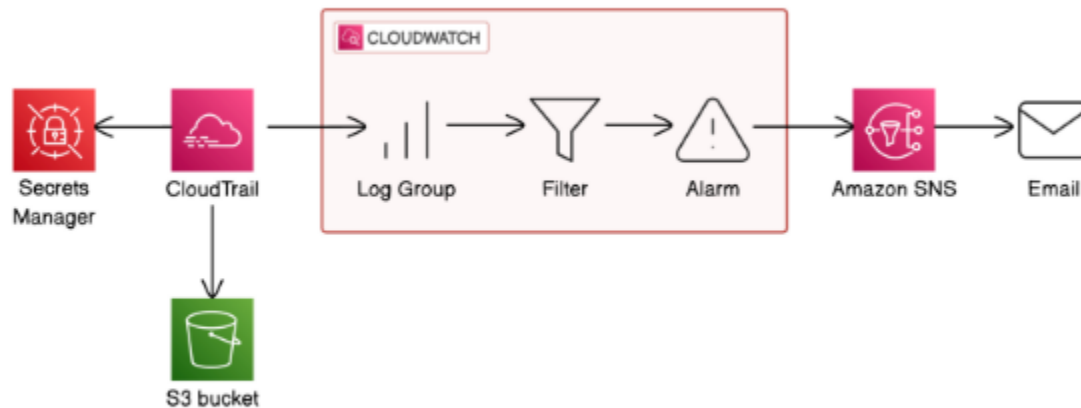


Building an SIEM using AWS

In this project, you'll build your own powerful monitoring system using AWS CloudTrail, CloudWatch and SNS.



Architecture Diagram

Create a New Secret

- Log in to the AWS Management Console [as your IAM Admin user](#).
- In the AWS Management Console search bar at the top, search for Secrets Manager and select **Secrets Manager** from the results.
- Select **Store a new secret** to begin creating your secret.
- Under **Choose secret type**, select **Other type of secret**.

Enter Your Secret

- In the **Key/value** tab, enter The Secret is as the **Key**.
- Enter a random secret or hot take that you have as the **Value**! For example, I need 3 coffees a day to function, or rice is the best carb

Store a new secret

Key/value pairs [Info](#)

Key/value | Plaintext

The Secret Is	I need 3 coffees a day to function
---------------	------------------------------------

[+ Add row](#)

Encryption key [Info](#)


You can encrypt using the KMS key that Secrets Manager creates or a customer-managed KMS key

-
-
- We'll keep the default **Encryption key** setting.
- Select **Next**.
- Welcome to the **Configure secret** page!
- Under **Secret name**, enter *TopSecretInfo*
- Under **Description - optional**, add a description like *Secret created for NextWork's project on Building a Monitoring System*
- Click **Next**.
- Click **Next** again to skip the **Configure rotation - optional** section
 - Let's review your secret set up:
 - Secret type: **Other type of secret**
 - Encryption key: **aws/secretsmanager**
 - Secret name: **TopSecretInfo**
 - Description: **Secret created for NextWork's project on Building a Monitoring System**
 - Secret replication: **Disabled**
 - Automatic rotation: **Disabled**
- Click **Store** at the bottom of the review page.
- You should see a green banner at the top.
- In the green banner, select **View details**
- You can now see your newly created secret *TopSecretInfo*

Now, let's configure CloudTrail to track access to our secret. CloudTrail is a **monitoring** service - it records events that happened in your AWS account, like creating resources, updating a name or setting... and accessing secrets in Secrets Manager

Create a New Trail

- In the AWS Management Console, head to the CloudTrail console.

- From the left hand navigation panel, select **Trails**.
- Select **Create trail** to start setting up a new trail.
- Under **Trail name**, enter secrets-manager-trail
- In the **Storage location** section, select **Create new S3 bucket**.
- Under **Trail log bucket and folder**, enter a unique bucket name:
 - Nextwork-secrets-manager-trail-yourinitials
-  Make sure to uncheck **Log file SSE-KMS encryption** - otherwise, you'll get charged for creating a new customer managed KMS key!
- Keep the other default settings.
- Scroll down and select **Next**.

Configure Log Events

- On the **Choose log events** page, ensure **Management events** is selected under **Event type**.
- Under **API activity**, keep both **Read** and **Write** checked
- Check **Exclude AWS KMS events**.
- Check **Exclude Amazon RDS Data API events**
- Select **Next**.
- Review your trail setup on the **Review and create** page:
 - **Step 1: Choose trail attributes**
 - Trail name: secrets-manager-trail
 - Multi-region trail: **Yes**
 - Apply trail to my organization: **Not enabled**
 - Trail log location:
nextwork-secrets-manager-trail-yourinitials/AWSLogs
 - Log file SSE-KMS encryption: **Not enabled**
 - Log file validation: **Enabled**
 - SNS notification delivery: **Disabled**
 - **Step 2: Choose log events**
 - **API activity: All**
 - **Exclude AWS KMS events: Yes**
 - **Exclude Amazon RDS Data API events: Yes**

Testing if CloudTrail detects SecretsManager Activity

Access Your Secret

- Navigate back to the **Secrets Manager** console.
- Pick your **TopSecretInfo** secret.
- On the secret details page, scroll down to the **Overview** section.
- Select **Retrieve secret value**.

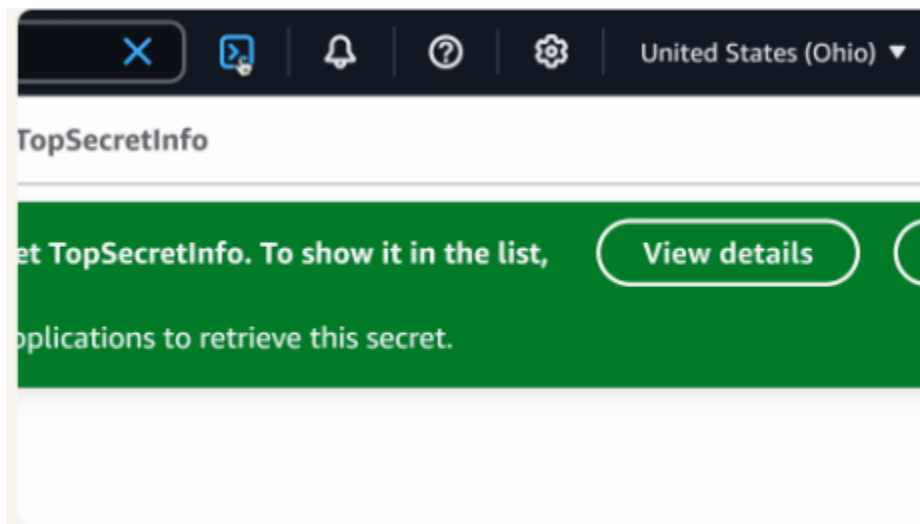
- You should now see the secret value displayed!
- Select Close to close the secret value display.

Access Your Secret Over AWS CLI

Turns out, the console is not the only way to access a secret.

If you'd like to be a little 🙌 extra 🙌, here's your chance at accessing your secret in a second way - the AWS CLI!

- Open AWS CloudShell - click on the CloudShell icon in the AWS Management Console's top navigation bar.



-
- The CloudShell terminal should now be opened and ready (it can take up to 30 seconds).
- In the CloudShell terminal, run the following command.
- Make sure to replace your-region-code at the end of the command. Use the region code you see when you select your Region dropdown (e.g. us-east-2 for the Ohio region):
 - `aws secretsmanager get-secret-value --secret-id "TopSecretInfo" --region us-east-2`
- The command should run successfully and give you the secret's value in JSON format.

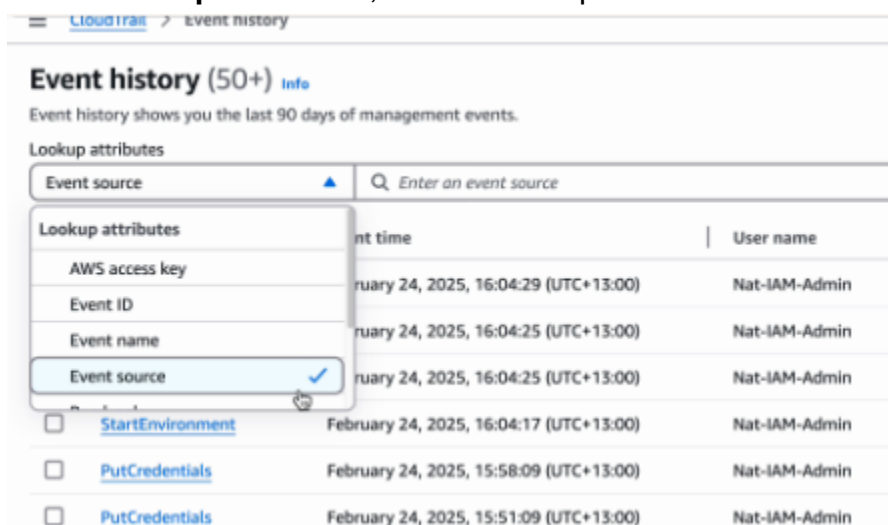
```
CloudShell
us-east-2 | +
~ $ aws secretsmanager get-secret-value --secret-id "TopSecretInfo" --region us-east-2
{
  "ARN": "arn:aws:secretsmanager:us-east-2:022687721466:secret:TopSecretInfo-96fRBn",
  "Name": "TopSecretInfo",
  "VersionId": "5ffffb99f-d369-4b8d-9f09-f249dabff591",
  "SecretString": "{\"The Secret Is\":\"I need three coffees a day to function\"}",
  "VersionStages": [
    "AWSCURRENT"
  ],
  "CreateDate": "2025-05-27T16:51:45.735000+00:00"
}
```

Oooo... You've generated **secret access events** through the console and the AWS CLI. Consider this our secret accessed. Do you think CloudTrail knows what you just did?

Let's see whether CloudTrail captured the events 🕵️

Analyse Your CloudTrail Events

- Head to the CloudTrail console again.
- You should now be on the **CloudTrail** dashboard.
- In the left navigation pane, select **Event history**.
- Under **Lookup attributes**, select the dropdown and choose **Event source**.



- In the search bar next to **Event source**, enter `secretsmanager.amazonaws.com`
- You should now see events related to `secretsmanager.amazonaws.com`.
- Scroll through the event list - check for an event called `GetSecretValue`. This events means your secret's value was retrieved or used. Woah!

Event history (32) Info

Event history shows you the last 90 days of management events.

Lookup attributes

Event source < 1 >

<input type="checkbox"/>	Event name	Event time	User name	Event source	Resource type	Resource name
<input type="checkbox"/>	GetSecretValue	May 27, 2025, 13:49:32 (UTC-05...	root	secretsmanager.amazonaws.com	AWS::SecretsManager::...	TopSecretInfo
<input type="checkbox"/>	GetSecretValue	May 27, 2025, 13:45:54 (UTC-05...	root	secretsmanager.amazonaws.com	AWS::SecretsManager::...	arn:aws:secretsma

Track Secrets Access Using CloudWatch Metrics

Alright! We know CloudTrail can track events for us, next up is to figure out how we can get alerts when your secret *does* get accessed. We will use CloudWatch.

Analyse Your CloudWatch Logs

Let's start this step by telling CloudTrail that we want a copy of all logs sent to another service - CloudWatch!

- Still in your CloudTrail console, select **Trails** in the left navigation pane.
- Select your trail secrets-manager-trail
- You should now be seeing the details of your trail.
- Scroll down to the **CloudWatch Logs** section.
- Select **Edit**.
- Check the **Enabled** checkbox for **CloudWatch Logs**.
- Select **New** log group.
- Under **Log group name**, enter nextwork-secretsmanager-loggroup
- Under **IAM Role**, select **New**.
- Under **Role name**, enter
CloudTrailRoleForCloudWatchLogs_secrets-manager-trail
- Select **Save changes** to save the new CloudWatch Logs setup.

Verify Your CloudWatch Logs

- Head to the **CloudWatch** console. Let's verify that CloudTrail is really passing the logs to a new log group.
- In the left navigation pane, expand **Logs** and select **Log groups**.
- In the **Log groups** page, search for and select
nextwork-secretsmanager-loggroup
- You might see multiple **Log streams** (i.e. subfolders of log groups). Pick any one of them. If you only see one, that's fine too!

- You should now see *heaps* of logs inside the stream. If you don't see rows and rows of logs straight away, you might need to wait a few minutes and refresh your page first.

<	Log streams	Tags	Anomaly detection	Metric filters	Subscription filters	Contributor Insights
---	--------------------	------	-------------------	----------------	----------------------	----------------------

Log streams (3)			Delete	Create log stream	
<input type="text"/> <small>Filter log streams or try prefix search</small>		<input type="checkbox"/> Exact match <input type="checkbox"/> Show expired Info			
<input type="checkbox"/>	Log stream	Last event time			
<input type="checkbox"/>	022687721466_CloudTrail_us-east-2	2025-05-27 19:57:08 (UTC)			
<input type="checkbox"/>	022687721466_CloudTrail_us-east-2_3	2025-05-27 19:55:12 (UTC)			
<input type="checkbox"/>	022687721466_CloudTrail_us-east-2_2	2025-05-27 19:54:58 (UTC)			

Create Metric Filter

- Head back to your log group.
- At the top of your log group, select **Actions** and then **Create metric filter** from the dropdown menu
- In the **Filter pattern** field, enter "GetSecretValue"
- We'll get to know the **Test pattern** section in Step #6 - for now, let's focus on creating the metric filter!
- Select **Next**.
- For the **Filter name**, let's use GetSecretsValue
- Under **Metric details**, name the **Metric namespace**: SecurityMetrics
- **Metric name**: Enter *Secret is accessed*
- **Metric value**: Enter 1.
- **Default value**: Enter 0
 - **Review and create** page for the metric filter.
 - Filter pattern: GetSecretValue
 - Filter name: GetSecretValue
 - Metric name: Secret is accessed
 - Metric namespace: SecurityMetrics
 - Applied on transformed logs: -
 - Metric value: 1
 - Default value: 0
 - Unit: -
- Select **Create metric filter** to finalize and create your metric filter.
- **Fantastic! You've created a metric filter and metric to track secret access.** Next, we'll create a CloudWatch Alarm to notify us when the secret is accessed.

Create CloudWatch Alarm and SNS Topic

- Now, let's create a CloudWatch Alarm that triggers when our `SecretIsAccessed` metric exceeds a threshold. We'll also set up an SNS topic to receive email notifications when the alarm is triggered.

Create CloudWatch Alarm

- Still in your CloudWatch Alarm's page, select the **Metric filters** tab.
- Scroll down and check the box next to the `GetSecretValue` metric filter.
- Select **Create alarm**.
- Now, welcome to the CloudWatch alarm setup!
- Under **Metric**, let's use the following values:
 - Namespace: **SecurityMetrics**
 - Metric name: Secret is accessed
 - Statistic: Average
 - Period: 5 minutes
- Under **Conditions**, set **Threshold type** to **Static**.
- Set **Whenever SecretIsAccessed is...** to **Greater/Equal**.
- Set **than...** to 1
- Select **Next**

We're now on the Configure actions page! This is *how* we tell CloudWatch what to do when we want to be alerted.

- Under **Notification**, keep the default setting **In alarm**
- Under the heading **Select a notification to the following SNS topic**, Select **Create new topic**.
- Under **Topic name**, enter **SecurityAlarms**
- Under **Email endpoints**, enter your email address that you can access. In the next step, you'll check this inbox for emails (when the alarm goes off)!
- Select **Create topic**.
- **The SNS topic should now be created!**
- **Select Next** - we'll head to add name and description for the alarm.
- Under **Alarm name**, enter **Secret is accessed**
- Under **Alarm description**, enter a description like **This alarm goes off whenever a secret in Secrets Manager is accessed**.
- **Select Next to review and create the alarm.**
 - **Let's review your alarm set up:**
 - **Namespace: SecurityMetrics**
 - **Metric name: Secret is accessed**
 - **Statistic: Average**

- **Period: 5 minutes**
- **Threshold type: Static**
- **Whenever Secret is accessed is: Greater/Equal (>=)**
- **than...: 1**
- **Notification: When In alarm, send a notification to "SecurityAlarms"**

Select Create alarm.

- **You should see a green banner at the top confirming that your alarm was created!**
- **There's also blue banner below it, telling us that a subscription is pending confirmation.**

Confirm SNS Subscription

- **Ooo check your created alarm's row too - there's a Warning sign under the Actions heading.**
- **Select the warning sign.**
- **Check your email inbox for an email from AWS Notifications with the subject AWS notification - Subscription Confirmation.**
- **This email is to confirm your subscription to the SNS topic you created!**
- **Select Confirm subscription.**
- **You should now see a Subscription confirmed! page in your browser.**

Test Email Notification

Let's test if our email notification system works as expected. We'll trigger the alarm by accessing the secret again and check if we receive an email notification.

Trigger Alarm

- **Head back to the Secrets Manager console. Let's try to trigger our own alarm by retrieving the secret value again!**
- **Head to your TopSecretInfo secret again.**
- **Select Retrieve secret value.**
- **Check your email inbox after a few minutes (it might take up to 5 minutes for the alarm to trigger and the email to arrive).**
- **Looks like we're not receiving the email after we trigger the alarm! This is actually a common scenario in real-world cloud environments.**
- **Setting up a monitoring system is one thing, but making sure all the parts work together correctly often requires some troubleshooting.**

You've done this before in 🐱 Step #3
and we'll do it again! Do you remember how to check your CloudTrail logs?

- Head to the CloudTrail console again.
- You should now be on the CloudTrail dashboard.
- In the left navigation pane, select Event history.
- Under Lookup attributes, select the dropdown and choose Event source.
- In the search bar next to **Event source**, enter **secretsmanager.amazonaws.com**
- You should see at least one **GetSecretValue** event at the top of the list, which matches the time you viewed your secret. There should be more than one row, since it's now your second time retrieving a secret.

Adjust Your Alarm Settings

Hmm... so we know the alarm *can* trigger an email. Since it's not triggering an email at the moment, we're down to the final investigation. Maybe your CloudWatch alarm isn't being triggered when it should!

Let's check a few critical settings:

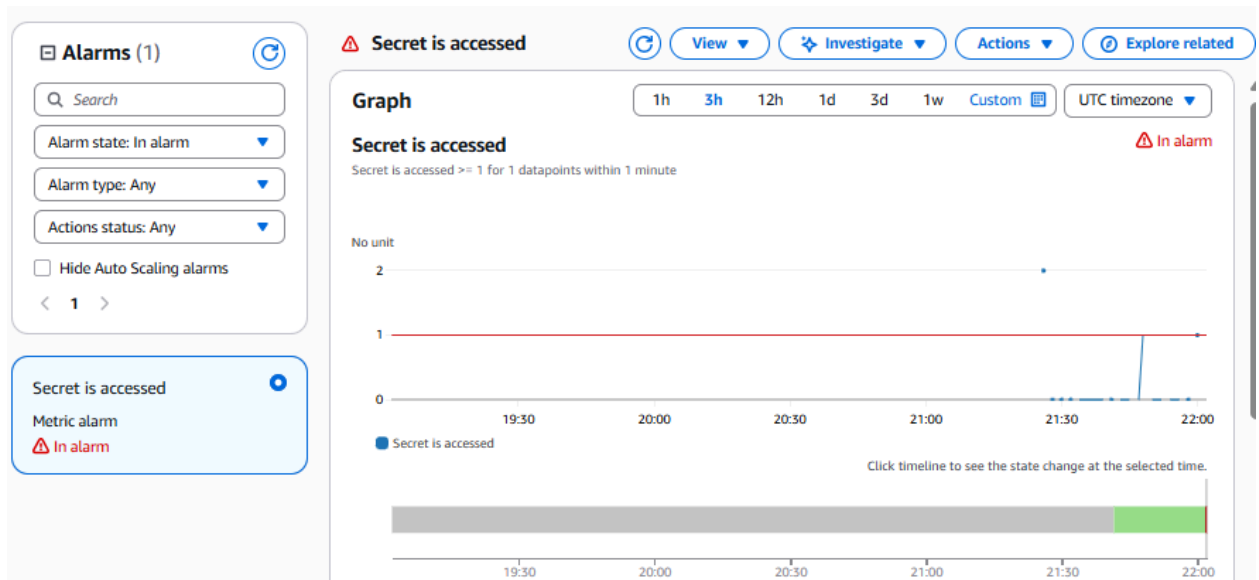
- Head back to the CloudWatch console.
- Select All alarms from the left hand navigation panel.
- Check the checkbox for your alarm.
- Select the Actions dropdown, and select Edit.
- On the alarm details page, look at the Statistic field.
- Aha - maybe the statistic should be set to Sum, not Average or any other statistic!
- This is crucial because:
 - Sum adds up all occurrences of secret access in the period (what we want).
 - Average would calculate an average rate (i.e. the average number of times our secret was accessed *per second* over the 5 minute period), which might never cross our threshold.
- Change the Statistic dropdown from Average to Sum.
- You can also update the Period from 5 minutes to 1 minute, so we can trigger the alarm even faster when we see the Secret's value.
- Check that the Threshold type is set to Static.
- Confirm that the condition is Greater/Equal than 1.
- Select Skip to Preview and create.
- In the review page, make sure Statistic is now Sum.
- Select Update alarm at the bottom of the page.

Access Your Secret Again

- Now that we have direct CloudTrail SNS notifications set up, let's generate another secret access event.
- Head back to the Secrets Manager console.
- Navigate to your TopSecretInfo secret again.
- Select Retrieve secret value to access the secret.
- This will trigger another secret access event that should now be captured by both our CloudWatch Alarm and the direct CloudTrail notifications.

Verify Your Alarm

- Head back to the CloudWatch console.
- Refresh your Secret is accessed alarm.
- It should be in the In alarm state!



Is this a success?!

- Now that you see your alarm in alarm state, head back to your inbox!
- Look for an email from AWS Notifications with the subject ALARM: "SecretIsAccessedAlarm".
- YESSSSSS - that's your monitoring system at work!

●

○