

Sending CloudWatch Logs to Sumo Logic

 dev.classmethod.jp/articles/sumo-logic-sendlogs-cloudwatch-logs-kinesis

酒井剛

March 8, 2023

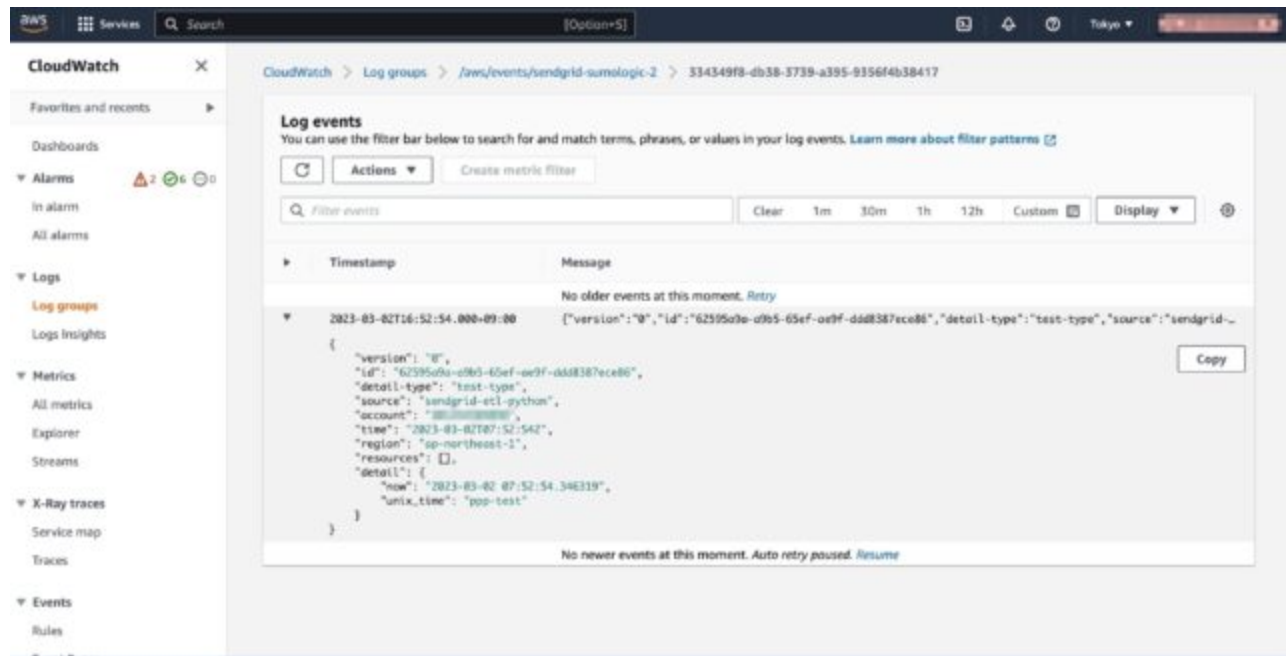
sumo logic

Today I would like to show you how to send logs collected with CloudWatch Logs to Sumo Logic.

Logs are being sent to CloudWatch Logs, so we want to send them to Sumo Logic. There are several ways to send logs from CloudWatch Logs to Sumo Logic, but considering the balance between performance and cost, integration with Firehose Data Kinesis is the best, so we will configure it that way.

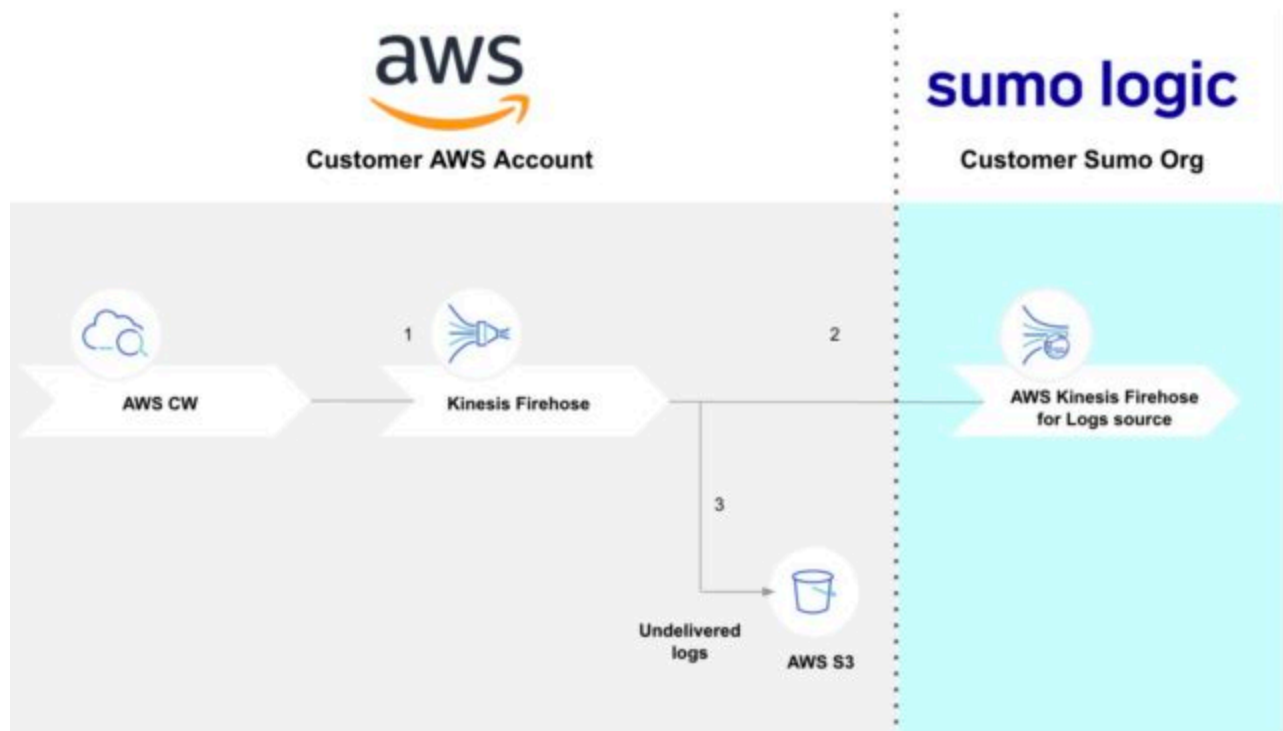
*The content of this blog is basically based on the information in [this official document](#).

First, check the CloudWatch Logs you want to send logs to. We will link the following currently configured logs.



Kinesis Data Firehose data sending image

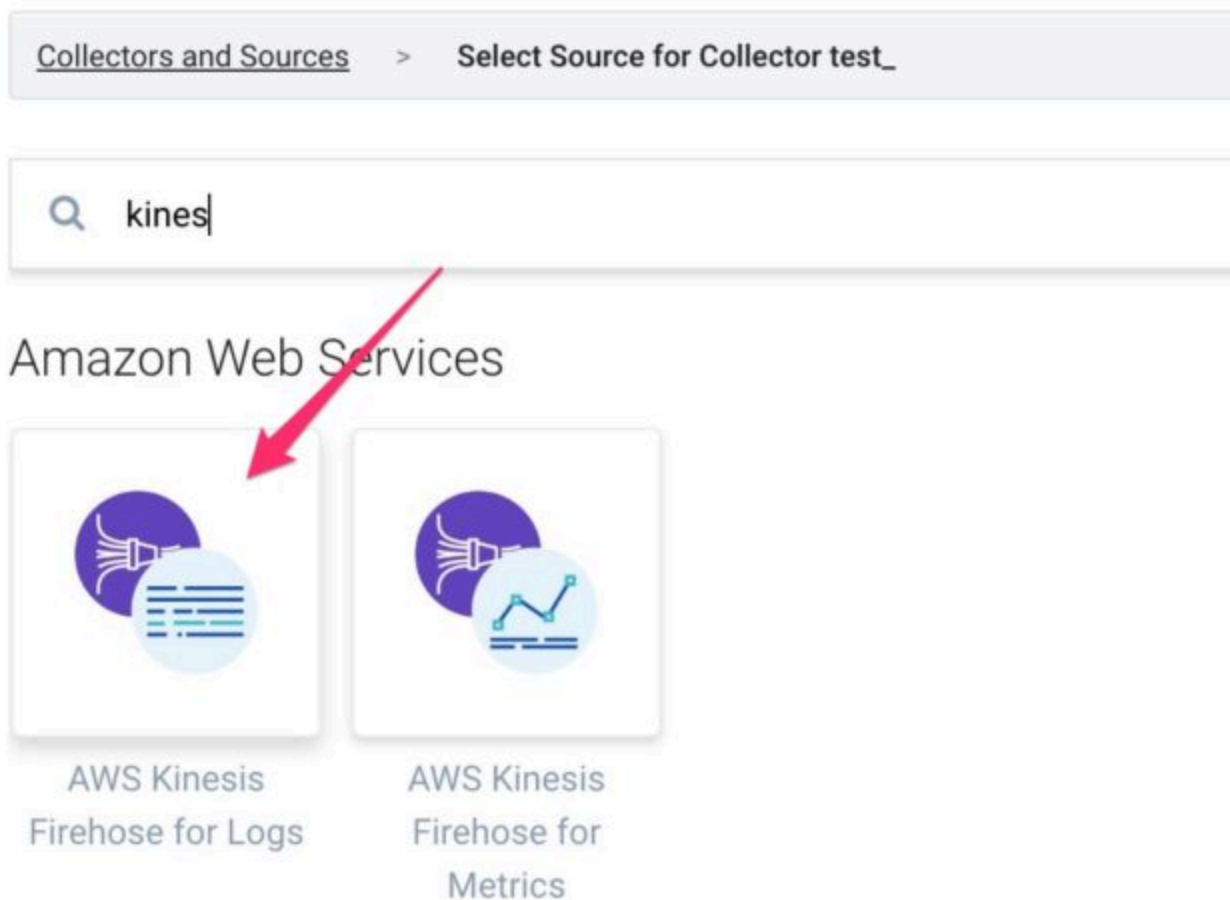
The image of data transmission is as follows:






Configure a Source in the Sumo Logic console

Log in to Sumo Logic and select Add Source on the Hosted Collector from Manage Data > Collection in the left pane. (If you have not yet configured a Hosted Collector, please refer to "1. Creating a Hosted Collector in Sumo Logic" in the blog below.)

When selecting the source, choose AWS Kinesis Firehose for Logs.



Set the Source name and Source Category. If Kinesis fails to send data, the failed data can be stored in S3. If you want Sumo Logic to poll S3 in the event of a failure, check Enable S3 Replay to enable it. However, this setting will be configured later, so for now, leave it unchecked.



Collection

+ New

Collection

Open Telemetry Collection

Status

Ingest Budgets

Archive

Data Archiving

Collectors and Sources > Select Source for Collector test > AWS Kinesis Firehose for Logs

Name*

Awesome Application Logs

Maximum name length is 128 characters.

Description

Enable S3 Replay

☐ Enable collection of undelivered logs from S3 bucket

Source Host

Host name for the system from which the data is being collected. This is optional, as not all data sources have host names. This will override the default set in the "Host Name" field at the Collector level. This data is queried using the '_sourceHost' key name.

Source Category

AWS/myaccount/awesome-application

Category metadata to use later for querying, e.g. prod/web/apache/access . This data is queried using the '_sourceCategory' key name.

SIEM Processing

☐ Select this checkbox to process data with Cloud SIEM Enterprise

Fields

+Add Field

FAC

> W

lo

> W

sc

sc

> H

Otherwise, leave the default check box and click Save.

Enable Timestamp Parsing

☒ Extract timestamp information from log file entries

Time Zone

☒ Use time zone from log file. If none is detected use:

Use Collector Default

☐ Ignore time zone from log file and instead use:

Use Collector Default

Timestamp Format

☒ Automatically detect the format ☐ Specify a format

Enable Multiline Processing

☒ Detect messages spanning multiple lines

☒ Infer Boundaries - Detect message boundaries automatically
Please note, Infer Boundaries may not be accurate for all log types.

☐ Boundary Regex - Expression to match message boundary e.g. (?<!\n)(\r+)

Enable One Message Per Request

☐ Each request will be treated as a single message (ignore line breaks).

NOTE: If this Source collects CloudWatch logs, these settings are ignored and automatically configured

▶ Processing Rules for Logs

[What are Processing Rules?](#)

Cancel

Save

Once you have completed the configuration, the Sumo Logic endpoint to which logs will be sent via Kinesis Data Firehose will be displayed, so copy it.

HTTP Source Address

Use the following address to send data to the Collector. [Learn more...](#)

Keep this address private since anyone can use it to send data.

https://collectors.jp.sumologic.com/receiver/v1/kinesis/log/

Copy

OK

Configuring Kinesis Data Firehose in the AWS Console

Log in to the AWS console and create and configure Kinesis Data Firehose resources. Sumo Logic provides a CloudFormation template for this configuration, so we will use [this](#) . Download the template and configure it using AWS CloudFormation.

CloudFormation

Stacks (11)

Filter by stack name

Active

Stack name

Status

Created time

Description

Stack actions

Create stack

With new resources (standard)

With existing resources (import resources)

Create stack

Step 2: Specify stack details

Step 3: Configure stack options

Step 4: Review

Prerequisite - Prepare template

Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Template is ready

Use a sample template

Create template in Designer

Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source

Selecting a template generates an Amazon S3 URL, where it will be stored.

Amazon S3 URL

Upload a template file

Upload a template file

Choose file

KinesisFirehoseCWLogs.template.yaml

JSON or YAML formatted file

S3 URL: https://s3-ap-northeast-1.amazonaws.com/cf-templates-ey5hf3r2lof-ap-northeast-1/2023-03-06T150424.774Zqo7-KinesisFirehoseCWLogs.template.yaml

View in Designer

Cancel

Next

6/19

There is a section for entering parameters, so enter them

Sumo Logic AWS Kinesis Firehose for Logs Source URLhere. Enter the endpoint URL you copied in the previous settings. **AWS S3 Bucket Name for Failed Data**This is the S3 bucket where Kinesis Data Firehose will store the logs as a backup if it fails to transfer them. Bucket names must be unique, so set them to be unique by including an appropriate hash value, date, purpose, etc. It also **SumoLogic-Kinesis-Failed-Logs/**says that logs will be output to the folder.

Stack name: AwesomeApplication-Kinesis-Sumo

Parameters

1. Sumo Logic Kinesis Firehose Logs Configuration

Sumo Logic AWS Kinesis Firehose for Logs Source URL

Provide HTTP Source Address from AWS Kinesis Firehose for Logs source created on your Sumo Logic account.

https://collectors.jp.sumologic.com/receiver/v1/kinesis/log/ZaVnC4dhvZwNVx9JxMbWtFEynUN4reEeNcjpWxoIGF146H6VznUzs8gallkDJS-2zWVzq4bLx1

2. Failed Data AWS S3 Bucket Configuration

Create AWS S3 Bucket

Yes - Create a new AWS S3 Bucket to store failed data. No - Use an existing AWS S3 Bucket to store failed data.

Yes

AWS S3 Bucket Name for Failed Data

Provide a unique name of AWS S3 bucket where you would like to store Failed data. In case of existing AWS S3 bucket, provide the bucket from the current AWS Account. For Logs, failed data will be stored in folder prefix as **SumoLogic-Kinesis-Failed-Logs/**

a762bfbff-20230308-sumologic-kinesis

Cancel Previous Next

Continue running the stack and verify that it completes successfully.

Setting up subscription filters in CloudWatch Logs

Configure a role for CloudWatch Logs to access Kinesis Data Firehose.

Create a role on the IAM page.

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles
- Policies
- Identity providers
- Account settings

Roles (48)

Create role

Role name	Trusted entities	Last ac...
	AWS Service: ec2	222 days i
	AWS Service: ec2	28 minutes
	AWS Service: events	-
	AWS Service: events	-

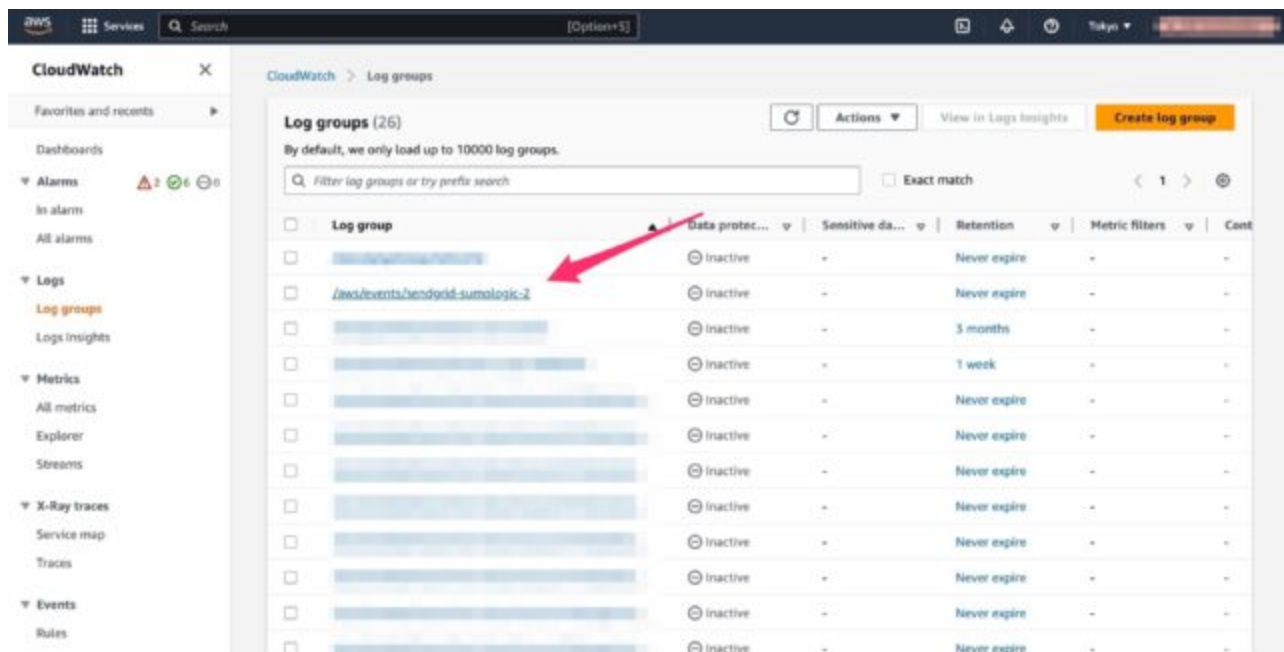
Set up the trust relationship as follows:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Principal": {
        "Service": "logs.ap-northeast-1.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

The policy set is the managed policy "AmazonKinesisFirehoseFullAccess".

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "firehose:*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Next, set up a subscription filter for the target Log groups in CloudWatch Logs.



Create a subscription filter for Kinesis Firehose.

Log group details

ARN
arn:aws:logs:ap-northeast-1:123456789012:log-group:/aws/events/sendgrid-sumologic-2*

Creation time
30 days ago

Retention
Never expire

Stored bytes
4.65 KB

Metric filters
0

Subscription filters
0

Contributor Insights rules
-

Data protection - new
Inactive

Sensitive data found - new
-

KMS key ID
-

Log streams | Metric filters | **Subscription filters** | Contributor Insights | Tags | Data protection - new

Subscription filters (0)

We now support up to 2 subscription filters per log group.

Filter subscription filters

Filter name | Filter pattern

There are no subscription filters.

Create Amazon OpenSearch Service subscription filter
Create Kinesis subscription filter
Create Kinesis Firehose subscription filter
Create Lambda subscription filter

Select the Delivery streams you are creating, the role, and enter a subscription name.

Create Kinesis Firehose subscription filter

You are about to start streaming data from your `"/aws/events/sendgrid-sumologic-2"` log group to an Amazon Kinesis Firehose delivery stream. Any new log data sent to this log group will be sent to the data stream you choose.

Choose destination

Choose the account and delivery stream to execute when a log event matches the filter you are going to specify.

Destination account

☒ **Current account**

Send log data to a Kinesis Firehose delivery stream in the current account.

☐ **Cross-account**

Send log data to a specified Kinesis Firehose delivery stream in another account. [Learn more about cross-account set up](#)

Kinesis Firehose delivery stream

Select an existing delivery stream you want to deliver matching log events to, or [create a new Kinesis Firehose data stream](#) .

Kinesis-Logs-30008250

Grant permission

To grant CloudWatch Logs permission to put data into your delivery stream, select an existing role below or [create a new role](#) .

Select an existing role

If your newly created role is not showing up in the dropdown list, please try the refresh button to the right.

CWL-Kinesis

Configure log format and filters

Choose your log format to get a recommended filter pattern for your log data, or select "Other" to enter a custom filter pattern. An empty filter pattern matches all log events.

Log format

Other

Subscription filter pattern

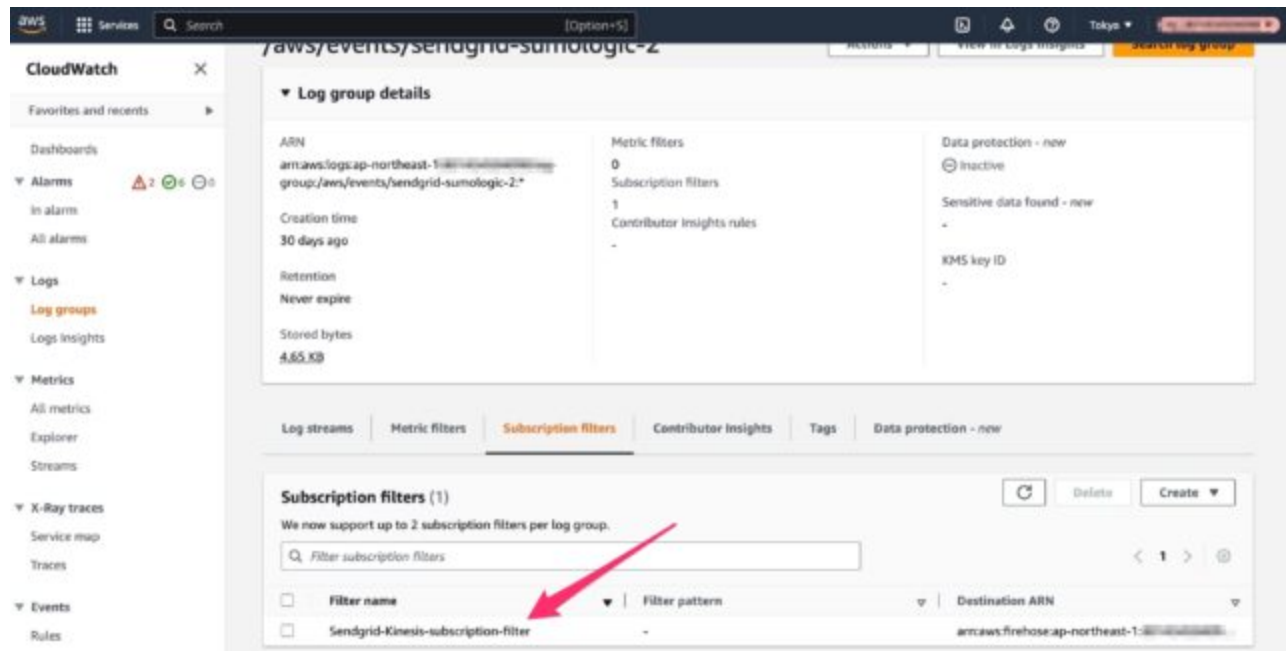
Specify the log event structure and any filter conditions to apply on your log data as it gets streamed to the Amazon Kinesis Firehose service.

Subscription filter pattern

Subscription filter name

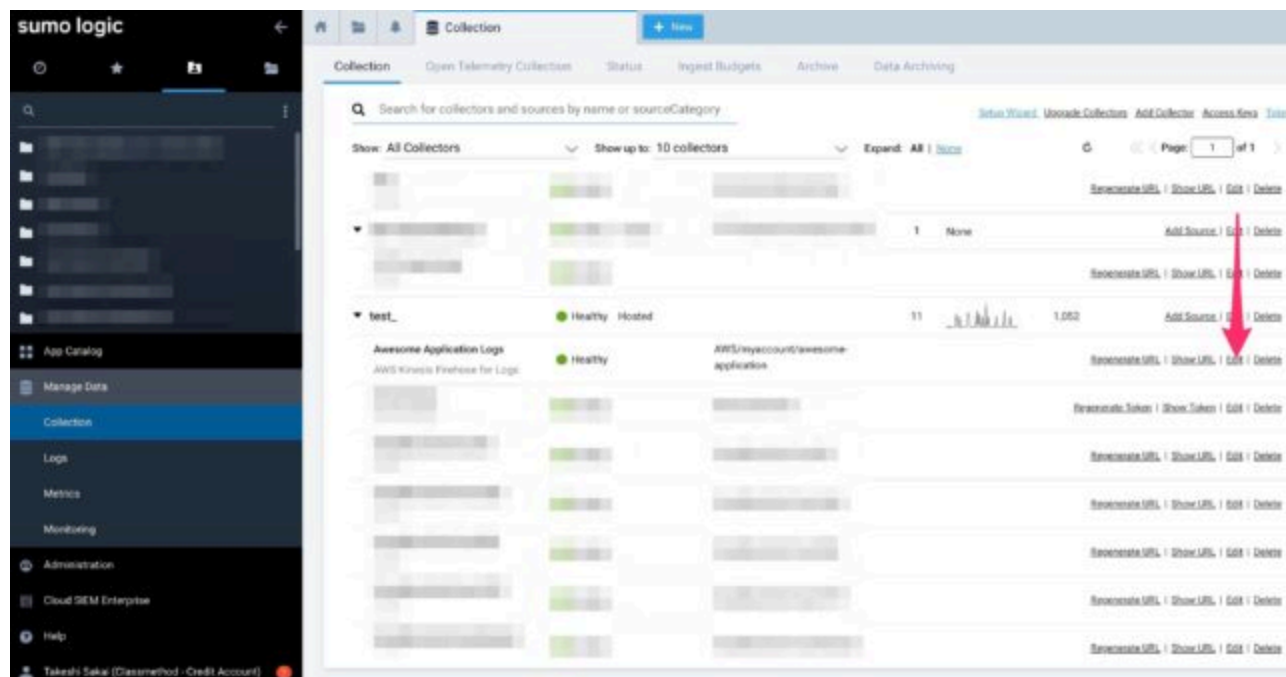
Sendgrid-Kinesis-subscription-filter

Then check that it's done.



Sumo Logic Source Settings: Kinesis Data Firehose Log Delivery Error Data Acquisition Settings

In the Sumo Logic source settings, enable Enable S3 Replay, which was previously unchecked. Edit the source you just created from Manage Data > Collection.



Check Enable S3 Replay to enable it, and enter the S3 region, bucket name, and path. The path is **http-endpoint-failed/**written to include the S3 folder name in the event of a delivery error when creating a Kinesis Data Firehose resource in CloudFormation, so specify it accordingly.

CollectionOpen Telemetry CollectionStatusIngest BudgetsArchiveData Archiv

Collectors and Sources > Edit Source: Awesome Application Logs

Name*

Awesome Application Logs

Maximum name length is 128 characters.

Description

Enable S3 Replay

☒ Enable collection of undelivered logs from S3 bucket

S3 Region

Asia Pacific (Tokyo)

Bucket Name*

a762bff8ff-20230308-sumologic-kinesis

Path Expression*

SumoLogic-Kinesis-Failed-Logs/http-endpoint-failed/*

Path Expression pointing to the http-endpoint-failed/ backup directory.
Start with the path prefix to your Kinesis bucket and append "http-endpoint-failed/" to it. For example, prefix-http-endpoint-failed/*.
NOTE: Make sure the path does NOT start with a leading slash.

Source Host

Host name for the system from which the data is being collected. This is optional, as not all data sources have host names. This will override the default set in the "Host Name" field at the Collector level. This data is queried using the '_sourceHost' key name.

Source Category

AWS/myaccount/awesome-application

Category metadata to use later for querying, e.g. prod/web/apache/access . This data is queried using the '_sourceCategory' key name.

SIEM Processing

☐ Select this checkbox to process data with Cloud SIEM Enterprise

Fields

+Add Field

When you enable Enable S3 Replay, a setting item called AWS Access appears. Here, you can configure a role for Sumo Logic to access S3. [Generate role-based access template](#)Click to download the CloudFormation template.

AWS Access

How should Sumo Logic access your AWS account? [Learn more](#)

Access Method* ☒ Role-based access (recommended) ☐ Key access

Use an AWS CloudFormation template to create an IAM role:

[Generate role-based access template](#) [Learn more](#)

Or, manually create a role on your AWS IAM console using the following information:

Account ID: 926226587429

External ID: jp:00000000001BAF1B

[Learn more](#)

Role ARN*

Log File Discovery

Scan Interval* Second(s)

The frequency at which the Source is scanned. Selecting a shorter interval increases the number of API calls and can incur additional cloud provider costs.

Log in to the AWS console again and create a CloudFormation stack. Specify the template you downloaded earlier, enter a stack name, and then run it by default.

The screenshot shows the AWS Management Console interface for creating a new CloudFormation stack. The left sidebar indicates the current step is 'Step 1: Create stack'. The main content area is titled 'Create stack' and includes a 'Prerequisite - Prepare template' section with options for 'Template is ready', 'Use a sample template', and 'Create template in Designer'. Below this is the 'Specify template' section, which includes a 'Template source' section with options for 'Amazon S3 URL' and 'Upload a template file'. The 'Upload a template file' option is selected, and a red arrow points to the 'Choose file' button. The file name 'role-a762bfbf-20230308-sumologic-kinesis.yaml' is displayed. At the bottom, the 'S3 URL' is shown, and there are 'Cancel' and 'Next' buttons.

Specify stack details

Stack name

Stack name

AwesomeApplication-KinesisS3-Sumo

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

No parameters

There are no parameters defined in your template

Cancel Previous Next

Verify that the AWS CloudFormation execution completes successfully, and copy the ARN of the role that was created.

The screenshot shows the AWS CloudFormation console. On the left, a list of stacks is shown, with 'AwesomeApplication-KinesisS3-Sumo' selected. The main panel displays the 'Outputs' tab for this stack. A table lists the outputs, with 'SumoRoleARN' highlighted. The value of 'SumoRoleARN' is copied to the clipboard, and a red arrow points to the 'Copy' button next to it.

Key	Value	Description	Export name
SumoRoleARN	arn:aws:iam::301119140101:role/AwesomeApplication-KinesisS3-Sumo-SumoRole	ARN of the created role. Copy this ARN back to Sumo to complete the source creation process.	

Return to the Sumo Logic console and continue configuring the source. Enter the Role ARN and set the scan interval to check the Kinesis Data Firehose delivery error bucket. This time, set it to one hour and click Save to confirm the settings.

AWS Access

How should Sumo Logic access your AWS account? [Learn more](#)


Access Method* ☒ Role-based access (recommended) ☐ Key access

Use an AWS CloudFormation template to create an IAM role:


[Generate role-based access template](#) [Learn more](#)

Or, manually create a role on your AWS IAM console using the following information:

Account ID: 926226587429
External ID: jp:00000000001BAF1B
[Learn more](#)

Role ARN* 

Log File Discovery

Scan Interval* 

The frequency at which the Source is scanned. Selecting a shorter interval increases the number of API calls and can incur additional cloud provider costs.

Verifying data transfer with Kinesis Data Firehose

Now that the configuration on Sumo Logic and AWS is complete, let's check if the data is being sent correctly.

Let's try sending data from the CLI.

```
aws firehose put-record --delivery-stream-name Kinesis-Logs-30008250 \  
--record '{"Data":"SGVsbG8gd29ybGQ="}'
```

When I searched for the data in Sumo Logic, I was able to confirm that the data was being sent properly.

The screenshot shows the Amazon Kinesis console interface. At the top, there's a breadcrumb trail: "SourceCategory - AWS/myacc...". Below it, a search bar contains the query: `_sourceCategory="AWS/myaccount/awesome-application" and _collector="test_"`. The search results show a single message at 03/08/2023 1:40:59 PM to 0... The message details are as follows:

#	Time	Message
1	03/08/2023 1:48:54.499 PM +0900	Hello world Host: 54.64.182.183 Name: Kinesis log Input Category: AWS/myaccount/awesome-applica

On the left, the "Displayed Fields" section shows "Time" and "Message" checked. The "Hidden Fields" section lists "Collector", "Size", "Source", "Source Category", "Source Host", and "Source Name".

Next, let's check the status of the delivery error by changing the destination endpoint in Kinesis Delivery streams to an invalid one.

The screenshot shows the Amazon Kinesis console interface for the "Kinesis-Logs-30008250" delivery stream. The "Configuration" tab is selected. The delivery stream details are as follows:

Field	Value
Status	Active
Destination	HTTP Endpoint
Data transformation	Not enabled
Creation time	March 08, 2023 at 09:51 GMT+9
Source	Direct PUT
ARN	arn:aws:firehose:ap-northeast-1:30008250:deliverystream/Kinesis-Logs-30008250
Error logs status	7 Destination error logs 0 Backup error logs

Below the details, there's a "Test with demo data" section with a link to "Test with demo data". The "Monitoring" tab is also visible, showing "Destination error logs" and "Backup error logs".

Edit destination settings

Destination settings Info
Specify the destination settings for your delivery stream.

HTTP endpoint name - optional
AwesomeApplication-Kinesis-Sumo-sumologic-logs-endpoint

HTTP endpoint URL
https://api.sumologic.com/api/v1/_source/_search?format=json&sourceType=kinesis-logs&sourceId=42ce-9f2b-12d6895a2489 **aaa**
Format: https://api.httpendpoint.com

Access key - optional
Contact the endpoint owner to obtain the access key required to enable data delivery to their service from Kinesis Data Firehose.

☒ Use current access key
☐ Update current access key

Content encoding
Kinesis Data Firehose uses the content encoding to compress the body of a request before sending the request to the destination.

☐ Not enabled
☒ GZIP

Retry duration
The time period during which Kinesis Data Firehose retries sending data to the selected HTTP endpoint.

60 seconds

Send the data again from the CLI.

```
aws firehose put-record --delivery-stream-name Kinesis-Logs-30008250 \
--record '{"Data":"SGVsbG8gd29ybGQ="}'
```

Then, data that was not delivered to S3 will flow in.

Amazon S3

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight

AWS Marketplace for S3

Amazon S3 > Buckets > a762bff9f-20230308-sumologic-kinesis > SumoLogic-Kinesis-Failed-Logs/ > http-endpoint-failed/ > 2023/ > 03/ > 04/

04/

Copy S3 URI

Objects Properties

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 Inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

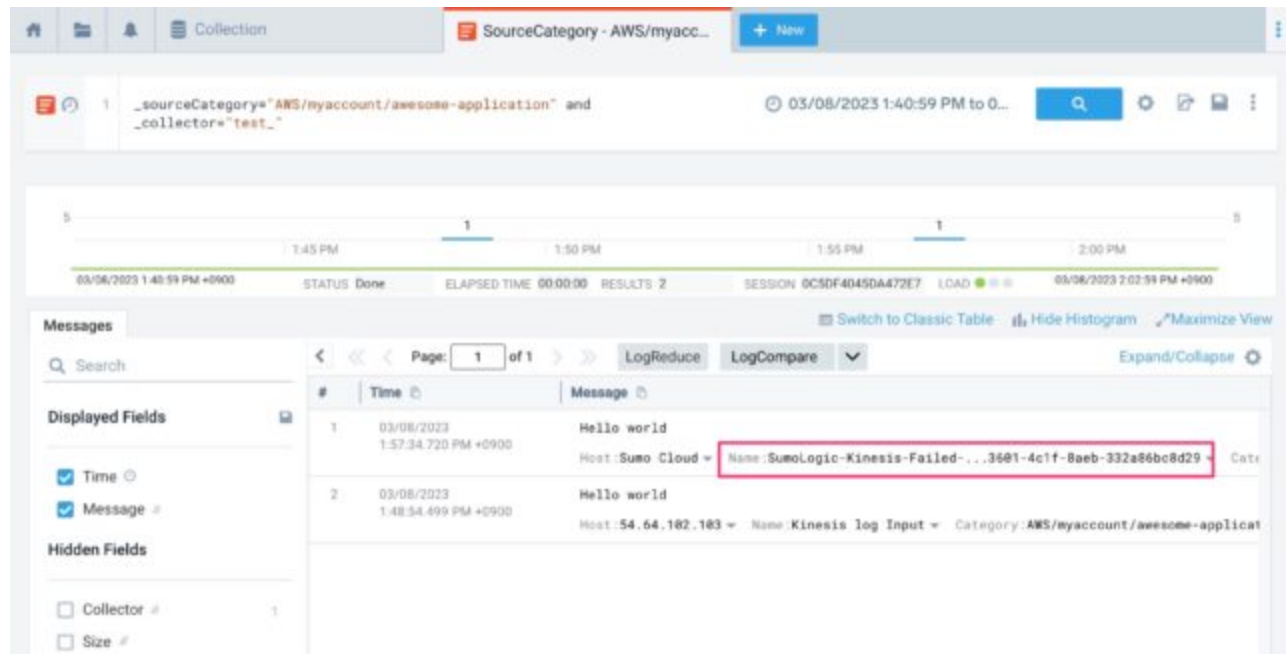
Refresh Copy S3 URI Copy URL Download Open Delete Actions Create folder

Upload

Find objects by prefix

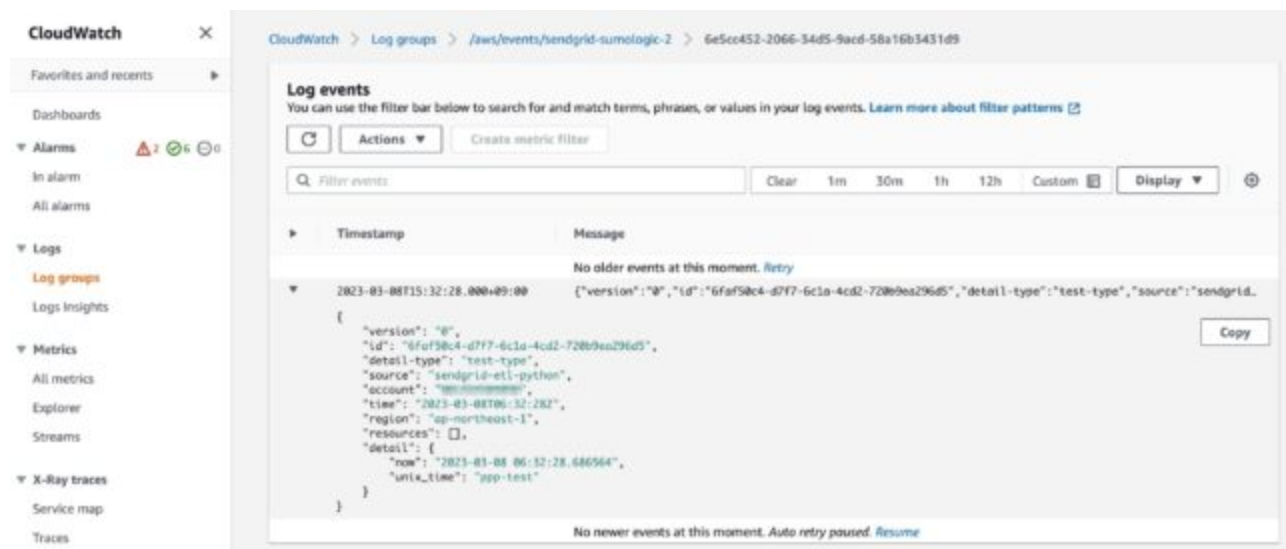
<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	Kinesis-Logs-4c1f-8aeb-352a86bc8d29	-	March 8, 2023, 13:56:05 (UTC+09:00)	1.1 KB	Standard
<input type="checkbox"/>	Kinesis-Logs-42ce-9f2b-12d6895a2489	-	March 8, 2023, 13:59:25 (UTC+09:00)	2.3 KB	Standard

If you check in Sumo Logic, you will see that another line of log has arrived. This is the log for which delivery failed, retrieved by polling S3.



Verify that CloudWatch Logs data is being ingested

Finally, we will verify that the data output to CloudWatch Logs is integrated with Sumo Logic. Now that the logs have been written to CloudWatch Logs, we will verify that they can be imported into Sumo Logic.



I was able to confirm that it was successfully captured.

The screenshot shows the AWS CloudWatch Logs console interface. At the top, the breadcrumb navigation indicates the path: **SourceCategory - AWS/myacc...**. Below this, a search filter is applied: `sourceCategory="AWS/myaccount/awesome-application" and collector="test_"`. The console displays a timeline of log events. The selected event is from 03/08/2023 3:32:28.000 PM +0900, with a status of **Done**. The message content is as follows:

```

{
  version: "0",
  id: "6faf50c4-d7f7-6c1a-4cd2-720b9ea296d5",
  detail-type: "test-type",
  source: "sendgrid-etl-python",
  account: "123456789012",
  time: "2023-03-08T06:32:28Z",
  region: "ap-northeast-1",
  resources: [ ],
  detail: {
    now: "2023-03-08 06:32:28.686564",
    unix_time: "ppp-test"
  }
}

```

On the left sidebar, under **Displayed Fields**, **Time** and **Message** are checked. Under **Hidden Fields**, **Collector**, **Size**, **Source**, **Source Category**, **Source Host**, and **Source Name** are listed with checkboxes.

summary

I hope this article was helpful to someone. I used Kinesis Data Firehose to integrate CloudWatch Logs with Sumo Logic to deliver logs.