

# 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.

The screenshot shows the AWS CloudWatch Log Events interface. On the left, there's a sidebar with navigation links for CloudWatch (Favorites and recent, Dashboards, Alarms, Logs, Log groups, Metrics, X-Ray traces, Events), AWS Services (Services, Search, Option+S), and Region (Tokyo). The main area shows a log group path: CloudWatch > Log groups > /aws/events/sendgrid-sumologic-2 > 334349f8-db38-3739-a395-9356f4b38417. The title is "Log events". A message says "You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns". Below is a table with columns: Timestamp and Message. One log entry is shown:

```

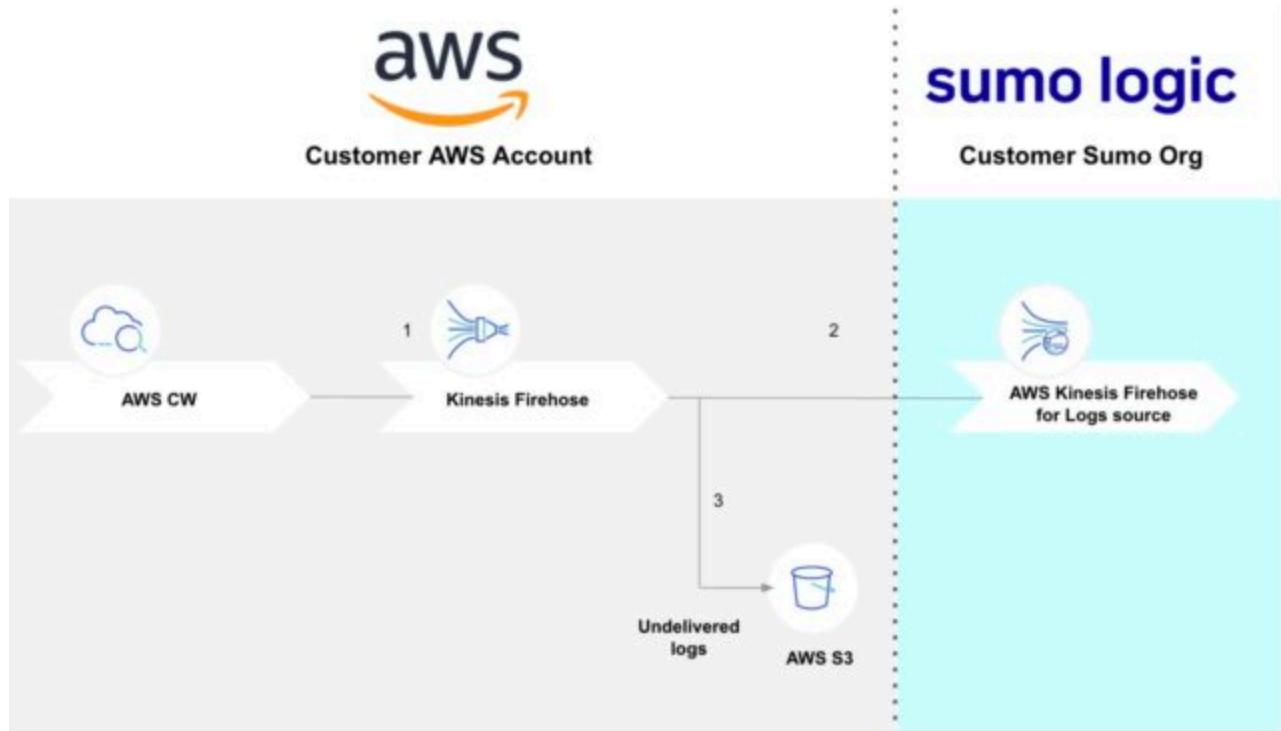
{
  "version": "0",
  "id": "625950a0e-d6b3-65ef-ae9f-ddd8387ece86",
  "detail-type": "test-type",
  "source": "sendgrid-etl-python",
  "account": "123456789012",
  "time": "2023-03-02T07:52:54Z",
  "region": "ap-northeast-1",
  "resources": [],
  "detail": {
    "now": "2023-03-02 07:52:54.346319",
    "unix_time": "1678044774.346319"
  }
}

```

No newer events at this moment. Auto retry paused. Resume.

## 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.

Collectors and Sources > Select Source for Collector test\_

kines|

Amazon Web Services

The screenshot shows a search bar at the top with the text "kines|". Below it, a section titled "Amazon Web Services" contains two items. A red arrow points from the text "Amazon Web Services" down to the first item. The first item is a card labeled "AWS Kinesis Firehose for Logs" with an icon showing a purple firehose and a blue bar chart. The second item is a card labeled "AWS Kinesis Firehose for Metrics" with an icon showing a purple firehose and a blue line graph.

AWS Kinesis Firehose for Logs	AWS Kinesis Firehose for Metrics
----------------------------------	--

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**   **Open Telemetry Collection**   **Status**   **Ingest Budgets**   **Archive**   **Data Archiving**

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

**Name\***  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**  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](#)

Otherwise, leave the default check box and click Save.

The screenshot shows the configuration interface for a log source. It includes sections for Timestamp Parsing, Time Zone, and Timestamp Format. Below these are sections for Enable Multiline Processing and Enable One Message Per Request. A red arrow points to the 'Save' button at the bottom right.

**Timestamp Parsing**

Extract timestamp information from log file entries

**Time Zone**

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

Ignore time zone from log file and instead use:

**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. (?<!\r\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.**

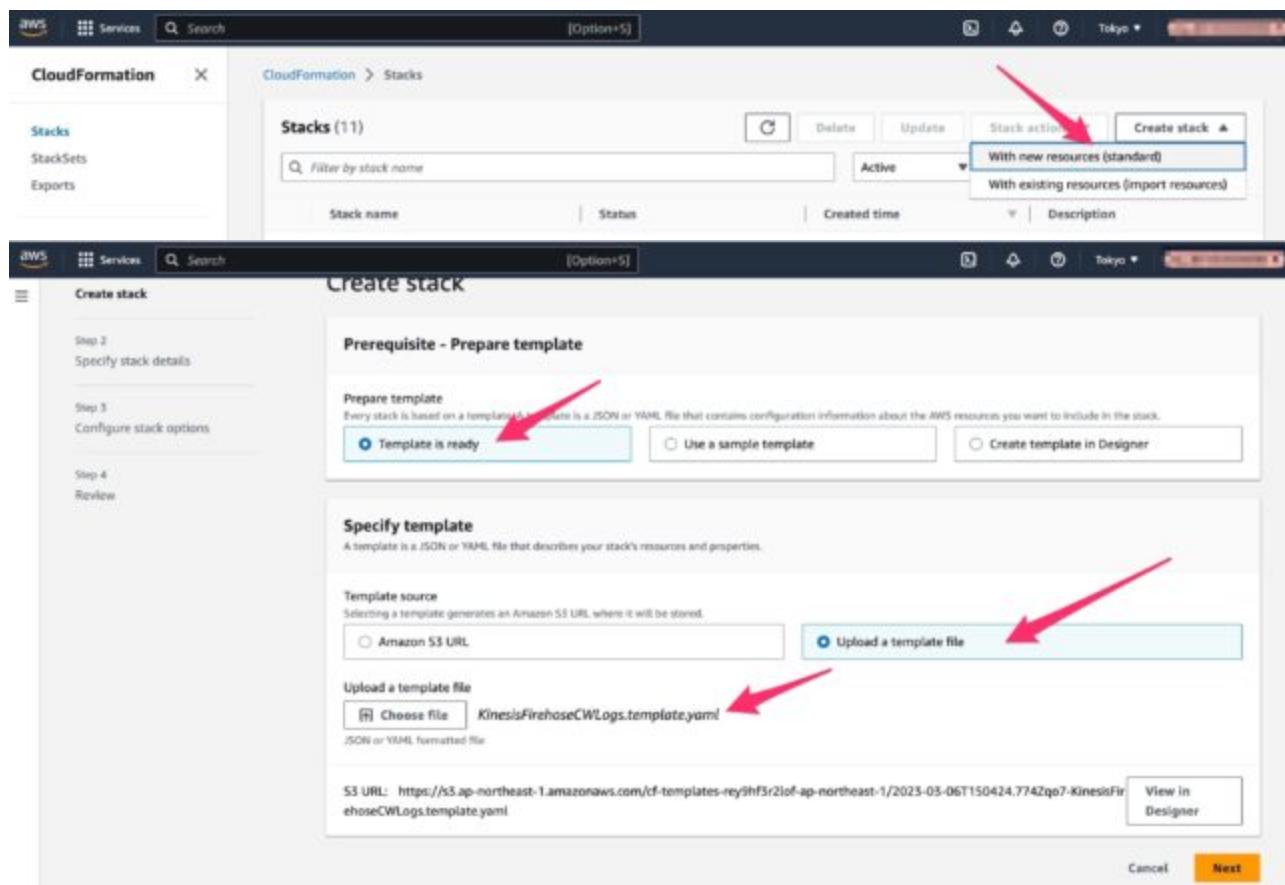
<https://collectors.jp.sumologic.com/receiver/v1/kinesis/log/>

Copy

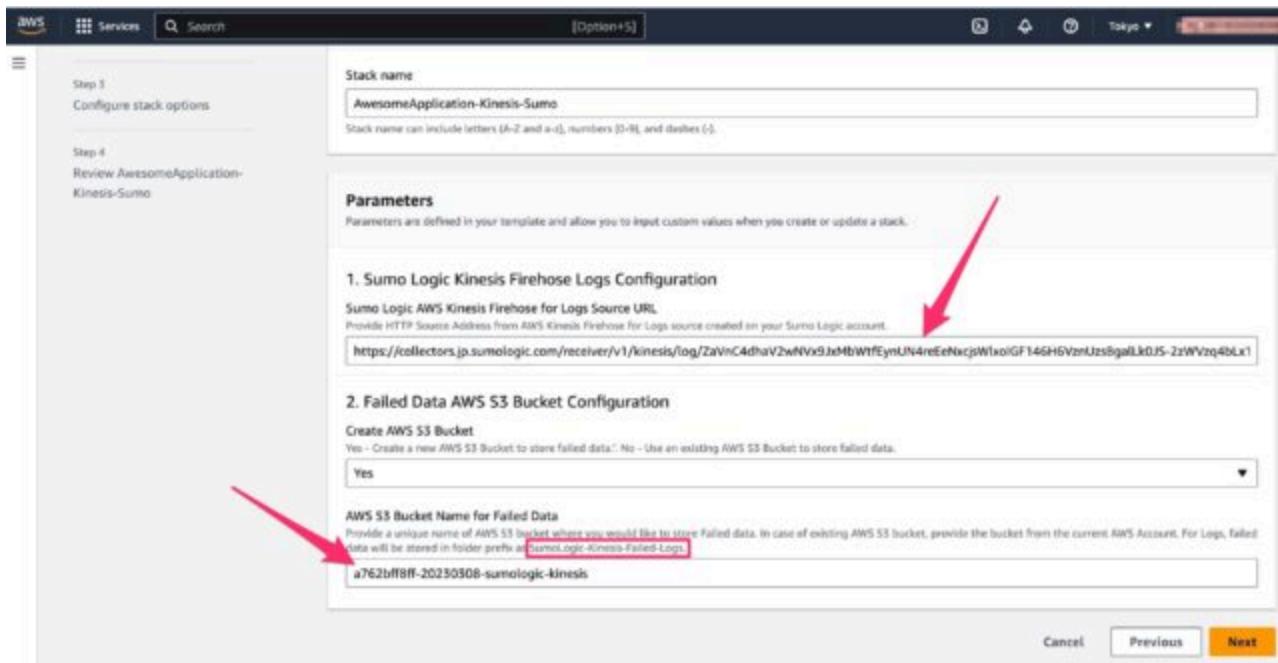
OK

## 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.



There is a section for entering parameters, so enter them **Sumo Logic AWS Kinesis Firehose for Logs Source URL** here. 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.

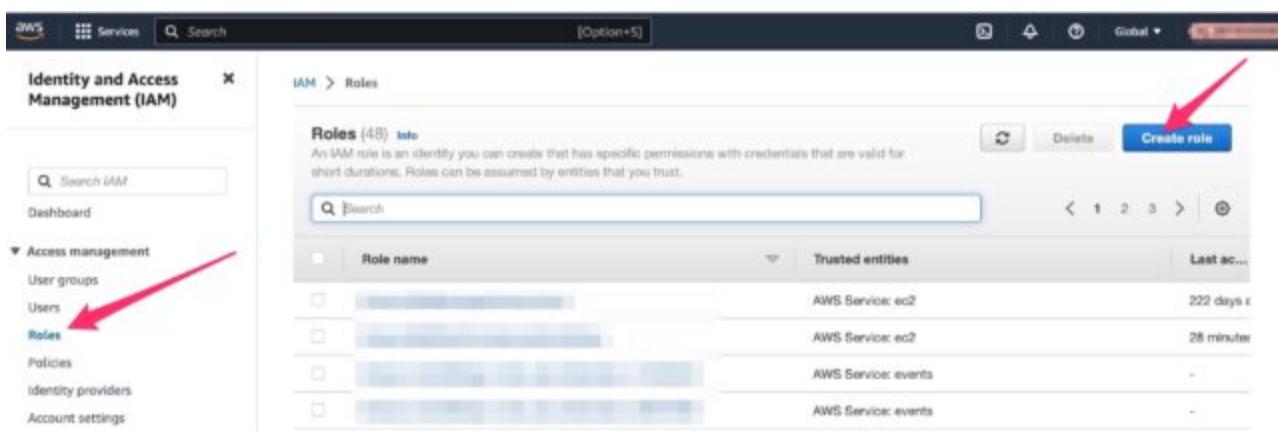


Continue running the stack and verify that it completes successfully.

## 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.



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.

The screenshot shows the AWS CloudWatch Logs interface. On the left, there's a sidebar with navigation links for Services, CloudWatch (selected), Alarms, Logs (selected), Metrics, X-Ray traces, and Events. Under Logs, 'Log groups' is also selected. The main area displays a table titled 'Log groups (26)'. The table has columns for Log group, Data protection, Sensitive data, Retention, Metric filters, and Cont. A red arrow points to the 'Log group' column for the entry '/aws/events/sendgrid-sumologic-2'.

Log group	Data protec...	Sensitive da...	Retention	Metric filters	Cont
[REDACTED]	Inactive	-	Never expire	-	-
[REDACTED]	Inactive	-	Never expire	-	-
[REDACTED]	Inactive	-	3 months	-	-
[REDACTED]	Inactive	-	1 week	-	-
[REDACTED]	Inactive	-	Never expire	-	-
[REDACTED]	Inactive	-	Never expire	-	-
[REDACTED]	Inactive	-	Never expire	-	-
[REDACTED]	Inactive	-	Never expire	-	-
[REDACTED]	Inactive	-	Never expire	-	-
[REDACTED]	Inactive	-	Never expire	-	-
[REDACTED]	Inactive	-	Never expire	-	-
[REDACTED]	Inactive	-	Never expire	-	-
[REDACTED]	Inactive	-	Never expire	-	-
[REDACTED]	Inactive	-	Never expire	-	-
[REDACTED]	Inactive	-	Never expire	-	-
[REDACTED]	Inactive	-	Never expire	-	-
[REDACTED]	Inactive	-	Never expire	-	-
[REDACTED]	Inactive	-	Never expire	-	-
[REDACTED]	Inactive	-	Never expire	-	-
[REDACTED]	Inactive	-	Never expire	-	-
[REDACTED]	Inactive	-	Never expire	-	-
/aws/events/sendgrid-sumologic-2	Inactive	-	Never expire	-	-

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*	Metric filters 0	Data protection - new Inactive
Creation time 30 days ago	Subscription filters 0	Sensitive data found - new -
Retention Never expire	Contributor Insights rules -	KMS key ID -
Stored bytes 4.65 KB		

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** ▾

- 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

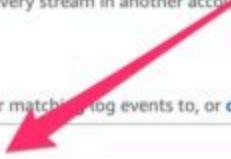
Send log data to a Kinesis Firehose delivery stream in the current account.

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](#)



### 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.



### 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



#### 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 name



Then check that it's done.

The screenshot shows the AWS CloudWatch Log Group details page for a log group named 'arn:aws:logs:ap-northeast-1:.../group:/aws/events/sendgrid-sumologic-2+'. The 'Subscription Filters' section displays a single filter named 'Sendgrid-Kinesis-subscription-filter'. A red arrow points to this filter name.

## 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.

The screenshot shows the Sumo Logic Collection management interface. A collection named 'test\_' is selected. In the list of sources, there is one entry for 'AWS Kinesis Firehose for Logs' with the status 'Healthy'. A red arrow points to the 'Edit' link next to this source entry.

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.

[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\*

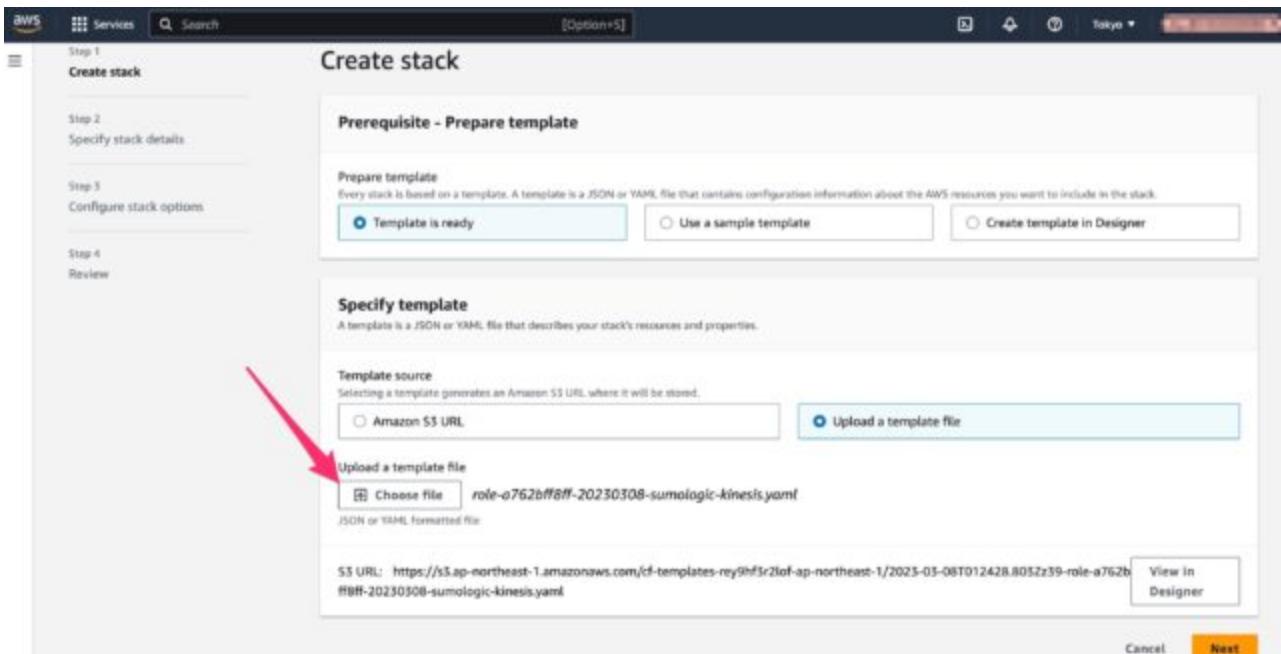
## Log File Discovery

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.



## Specify stack details

**Stack name**

Stack name  (arrow pointing to this field)  
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 with the 'Stacks' page open. On the left, there's a list of stacks, including 'AwesomeApplication-KinesisS3-Sumo' which is highlighted. On the right, the details for the selected stack are shown. The 'Outputs' tab is active, displaying one output named 'SumoRoleARN' with the value: arn:aws:iam::00010242:role/AwesomeApplication-KinesisS3-SumoS3-Role.

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\*

arn:aws:iam::██████████:role/AwesomeApplication-KinesisS3-Sumo-SumoRc



## Log File Discovery

Scan Interval\*

1 Hour(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.

## 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 Logstash interface with the following details:

- Collection:** SourceCategory - AWS/myacc...
- Search Query:** \_sourceCategory="AWS/myaccount/awesome-application" and \_collector="test\_"
- Time Range:** 03/08/2023 1:40:59 PM to 03/08/2023 1:53:59 PM
- Status:** Done
- Elapsed Time:** 00:00:00
- Results:** 1
- Session:** 0C66D2559C7E9C76
- Load:** Low
- Messages:** 1
- Displayed Fields:** Time, Message (checked)
- Hidden Fields:** Collector, Size, Source, Source Category, Source Host, Source Name

#	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-application

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 AWS Amazon Kinesis Data Firehose console. On the left, there's a sidebar with navigation links: Dashboard, Data streams (Delivery streams is selected), Analytics applications, Resources (Resources is expanded), and What's new. The main content area shows the details for a delivery stream named "Kinesis-Logs-30008250". The stream is active, using Direct PUT as the source and an HTTP Endpoint as the destination (Amazon CloudWatch Logs). Data transformation is not enabled. The creation time is March 08, 2023 at 09:51 GMT+9. There are 7 destination error logs and 0 backup error logs. A "Test with demo data" button is available for testing the stream's configuration. At the bottom, tabs for Monitoring, Configuration (which is selected), Destination error logs, and Backup error logs are present. A "Transform records" section with an "Edit" button is also visible.

AWS Services Search [Option+S] Tokyo

Amazon Kinesis > Delivery streams > Kinesis-Logs-30008250 > Edit destination settings

## 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://aws-kinesis-sumologic-test-x509qj3idExPfbed4ATl3rHrem\\_AXTnw.saa](https://aws-kinesis-sumologic-test-x509qj3idExPfbed4ATl3rHrem_AXTnw.saa)

Format: <https://xyz.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 > a762bff8ff-20230308-sumologic-kinesis > SumoLogic-Kinesis-Failed-Logs/ > http-endpoint-failed/ > 2023/ > 03/ > 08/ > 04/

**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

04/

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

Copy S3 URI  Copy URL  Download  Open  Delete  Actions  Create folder

Upload

Find objects by prefix

Name	Type	Last modified	Size	Storage class
Kinesis-Logs-4c1f-8aeb-332a86bc8d29	File	March 8, 2023, 15:56:05 (UTC+09:00)	1.1 KB	Standard
Kinesis-Logs-42ce-9f2b-12d8b95a2480	File	March 8, 2023, 15:59:25 (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 Sumo Logic interface with a search results page. The search query is: `_sourceCategory="AWS/myaccount/awesome-application" and _collector="test_"`. The results show one log entry from 03/08/2023 at 3:32:28 PM +0900. The log message is:

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

## summary

---

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