Sending CloudWatch Logs to Sumo Logic

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

酒井剛

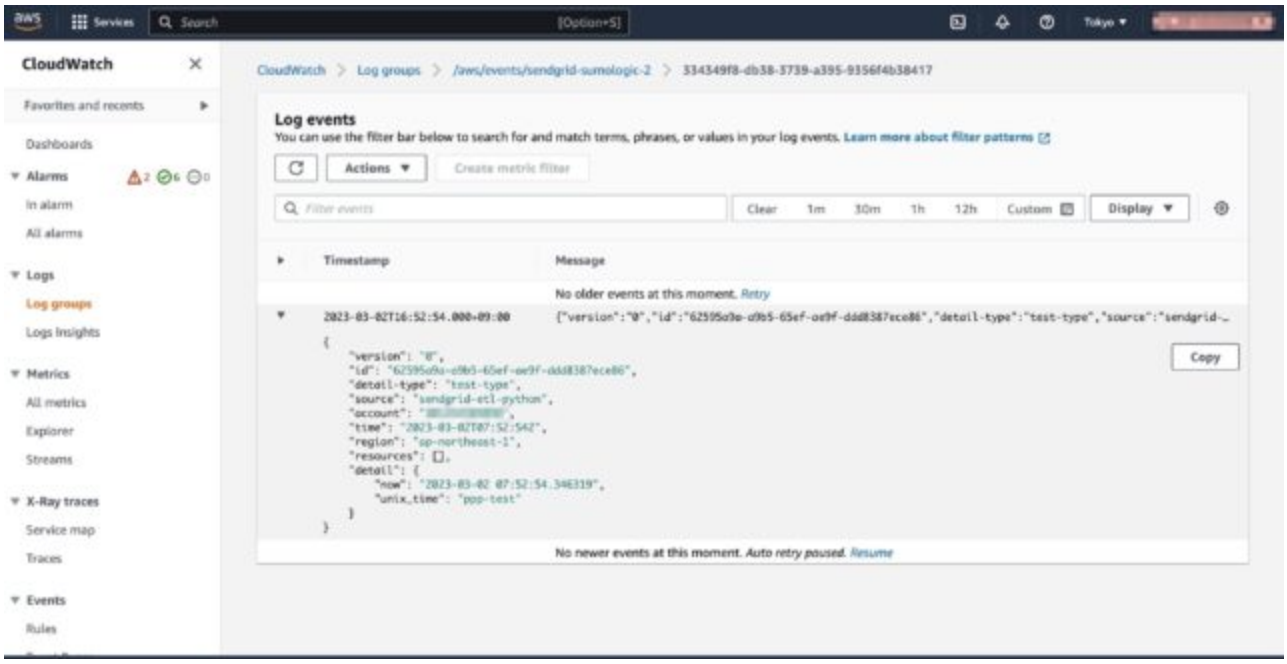
March 8, 2023

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.

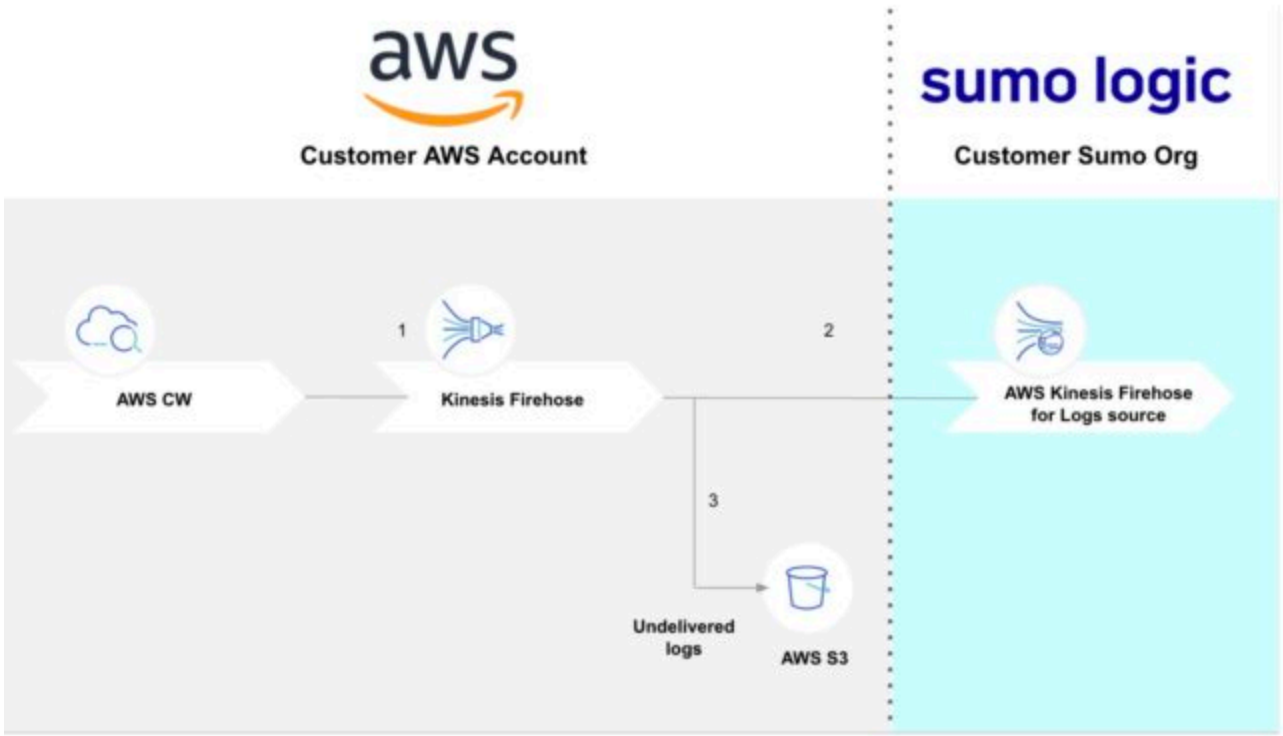
*The content of this blog is basically based on the information in this official document.

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

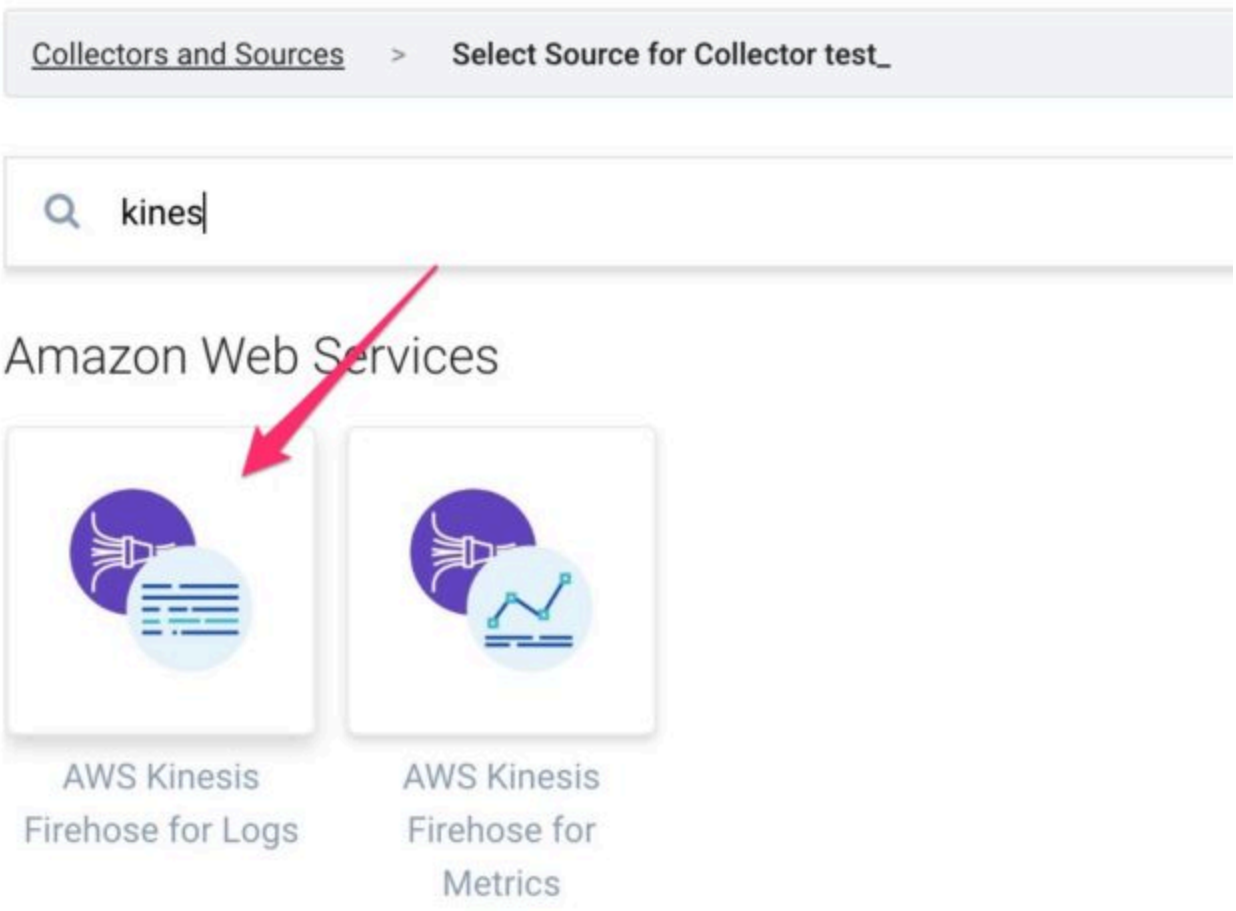
The image of data transmission is as follows:



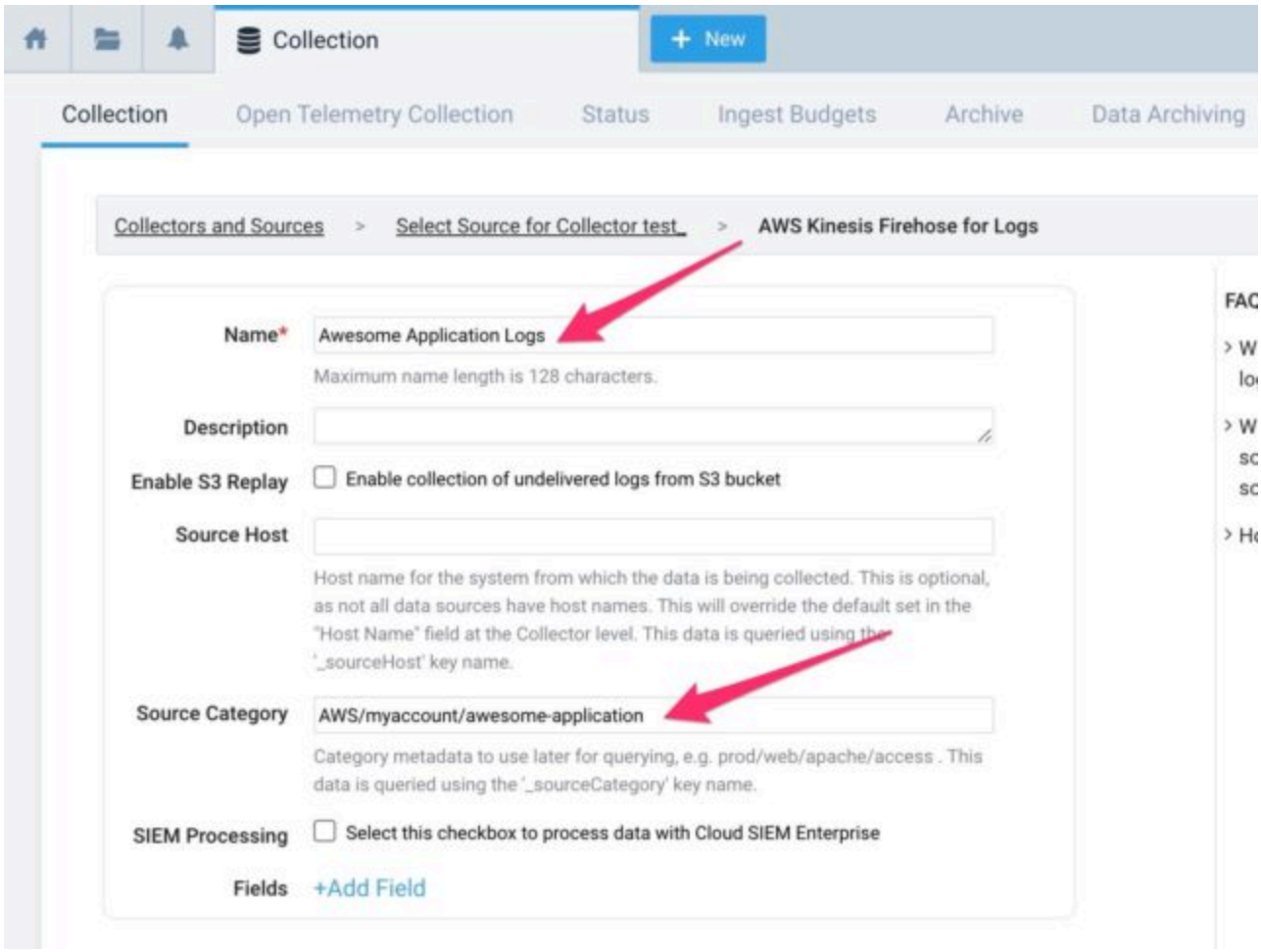
Configure a Source in the Sumo Logic console

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.)

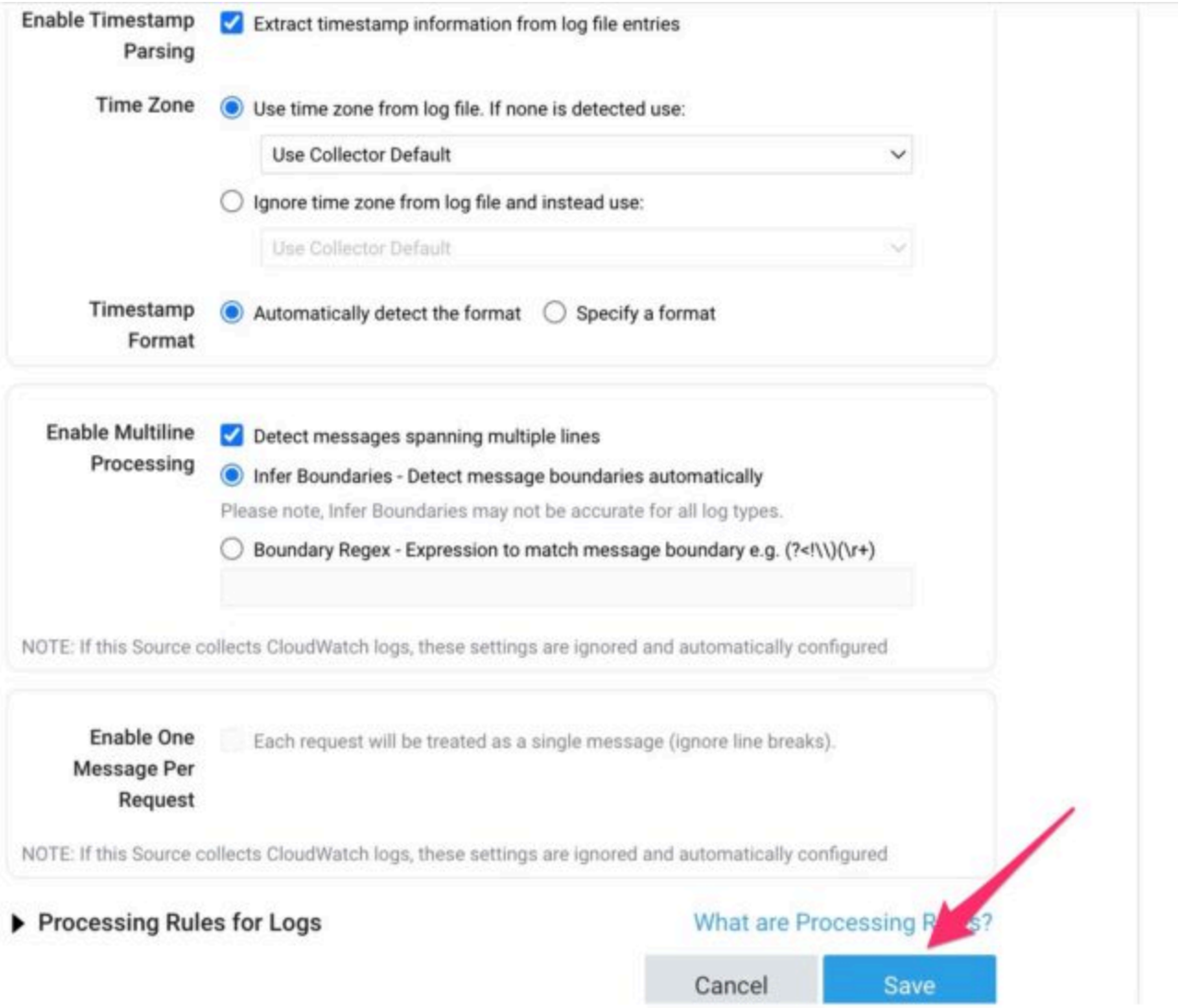
When selecting the source, choose AWS Kinesis Firehose for Logs.



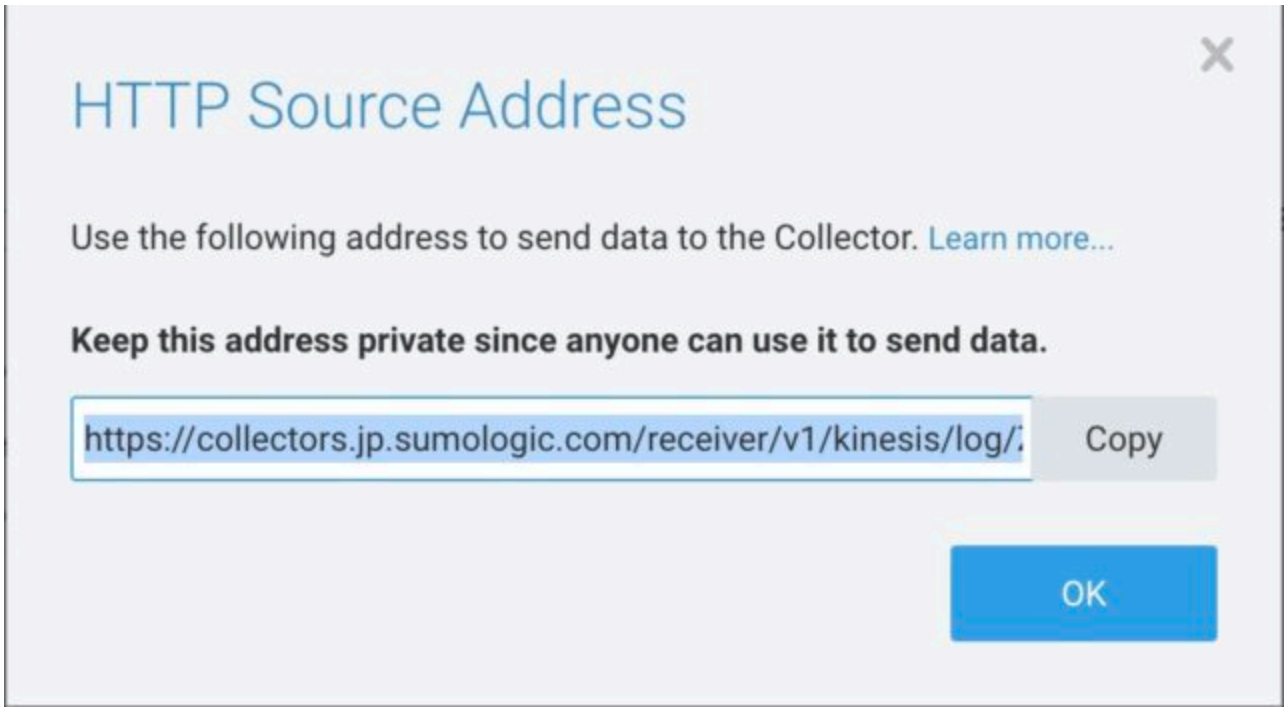
Set the Source name and Source Category. If Kinesis fails to send data, the failed data can be stored in S3. If you want Sumo Logic to poll S3 in the event of a failure, check Enable S3 Replay to enable it. However, this setting will be configured later, so for now, leave it unchecked.


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

Otherwise, leave the default check box and click Save.

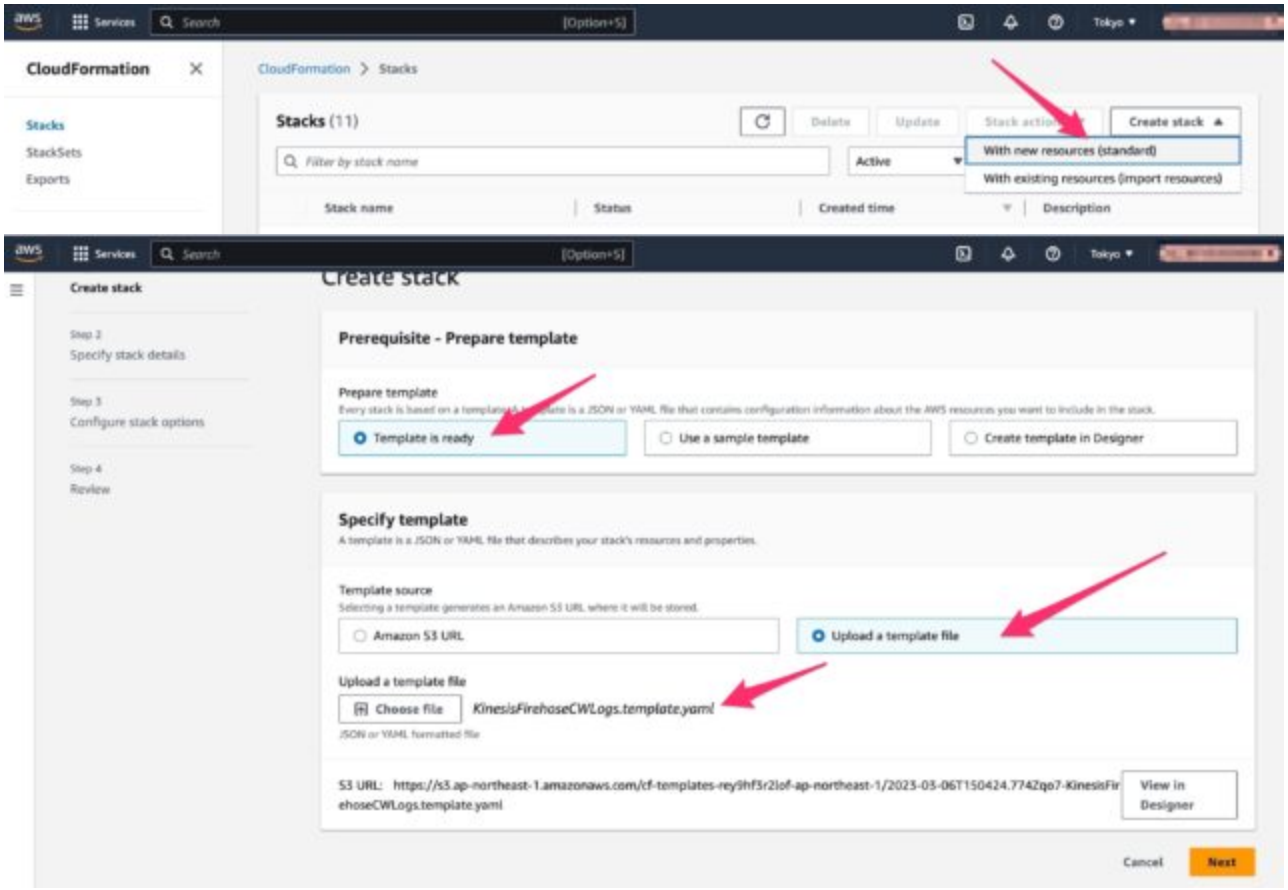


Once you have completed the configuration, the Sumo Logic endpoint to which logs will be sent via Kinesis Data Firehose will be displayed, so copy it.

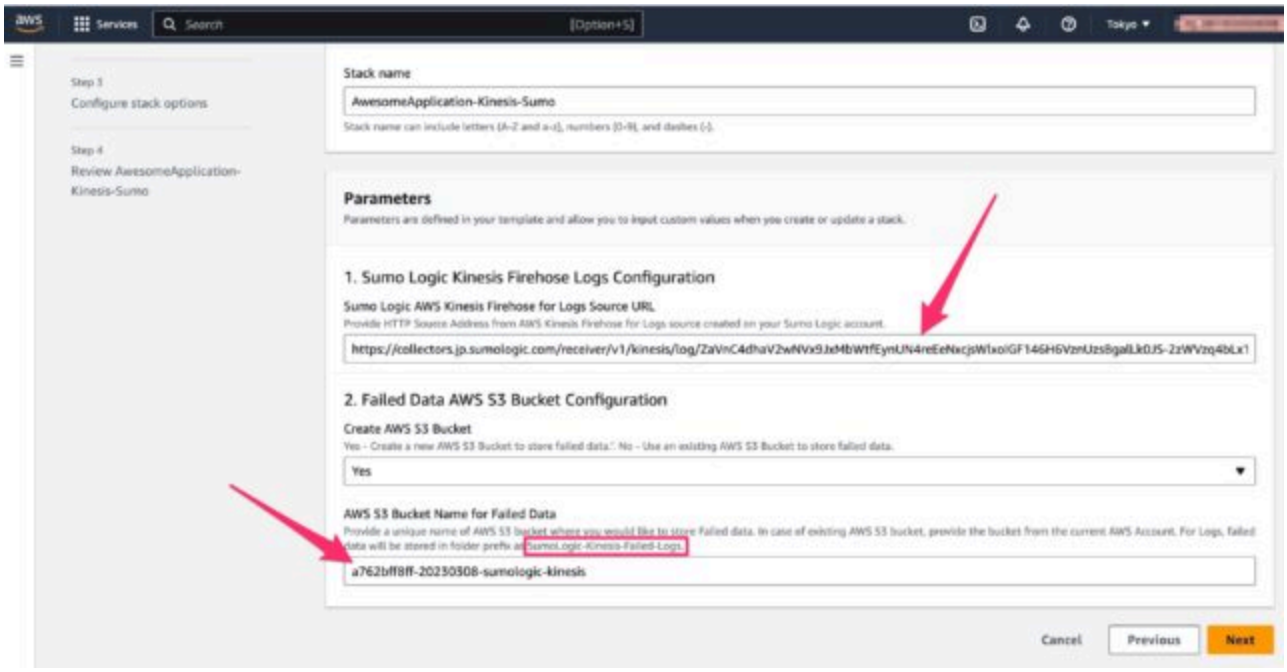


Configuring Kinesis Data Firehose in the AWS Console

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.

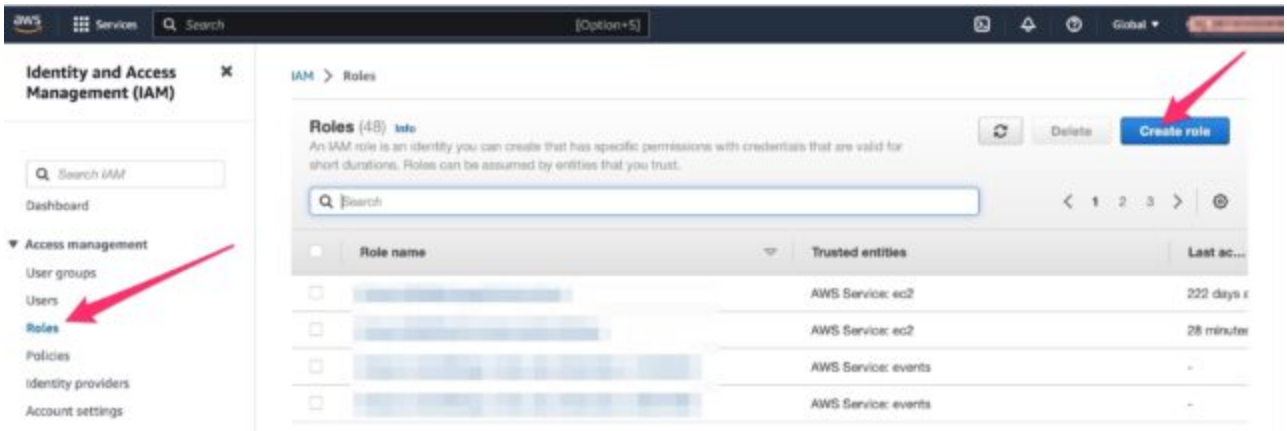


Continue running the stack and verify that it completes successfully.

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.



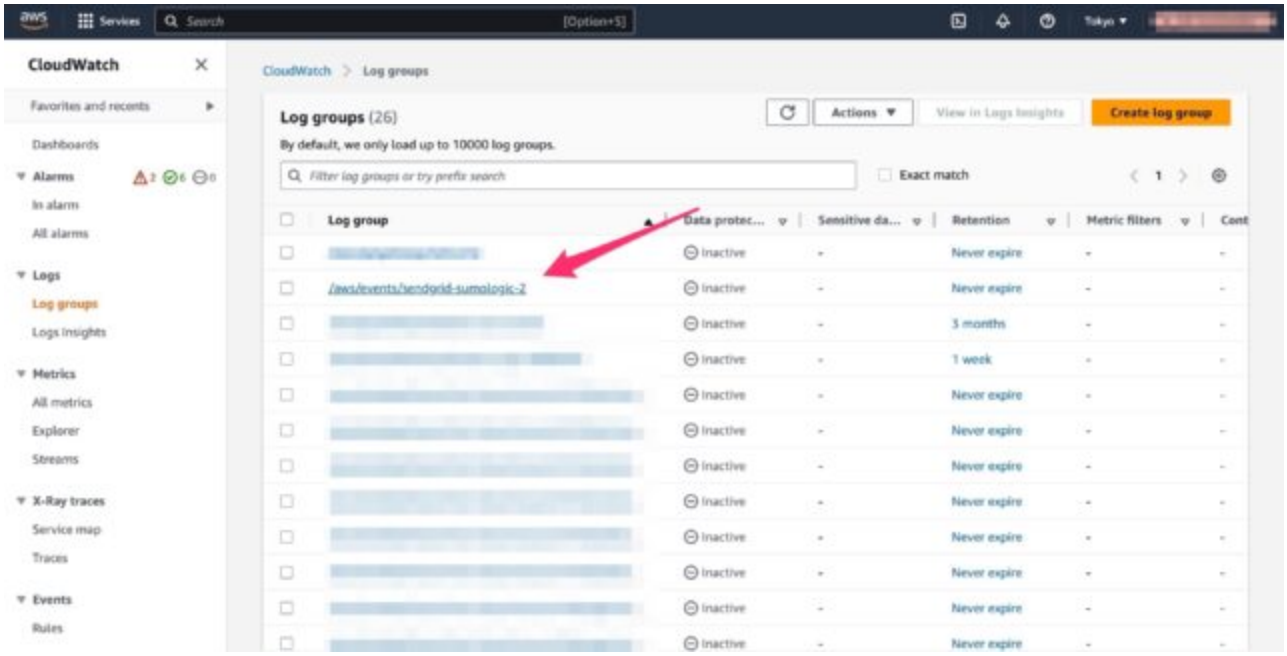
Set up the trust relationship as follows:

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

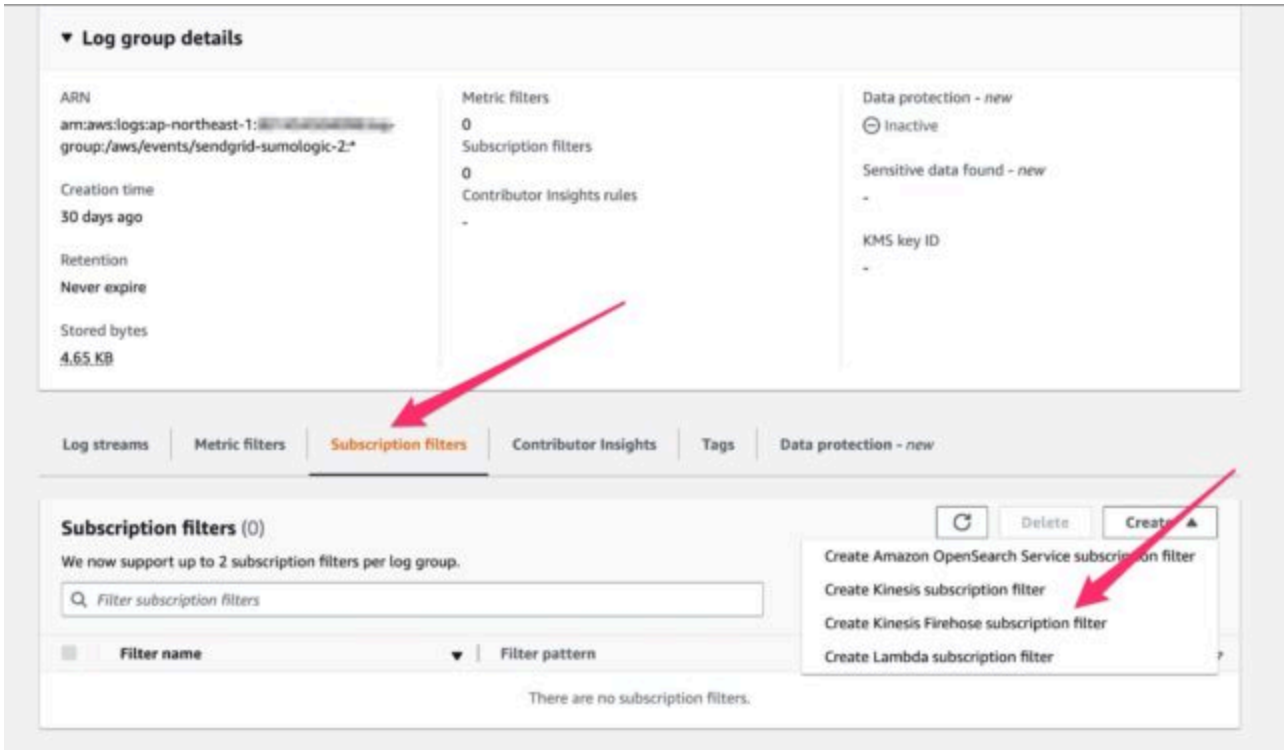
The policy set is the managed policy "AmazonKinesisFirehoseFullAccess".

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

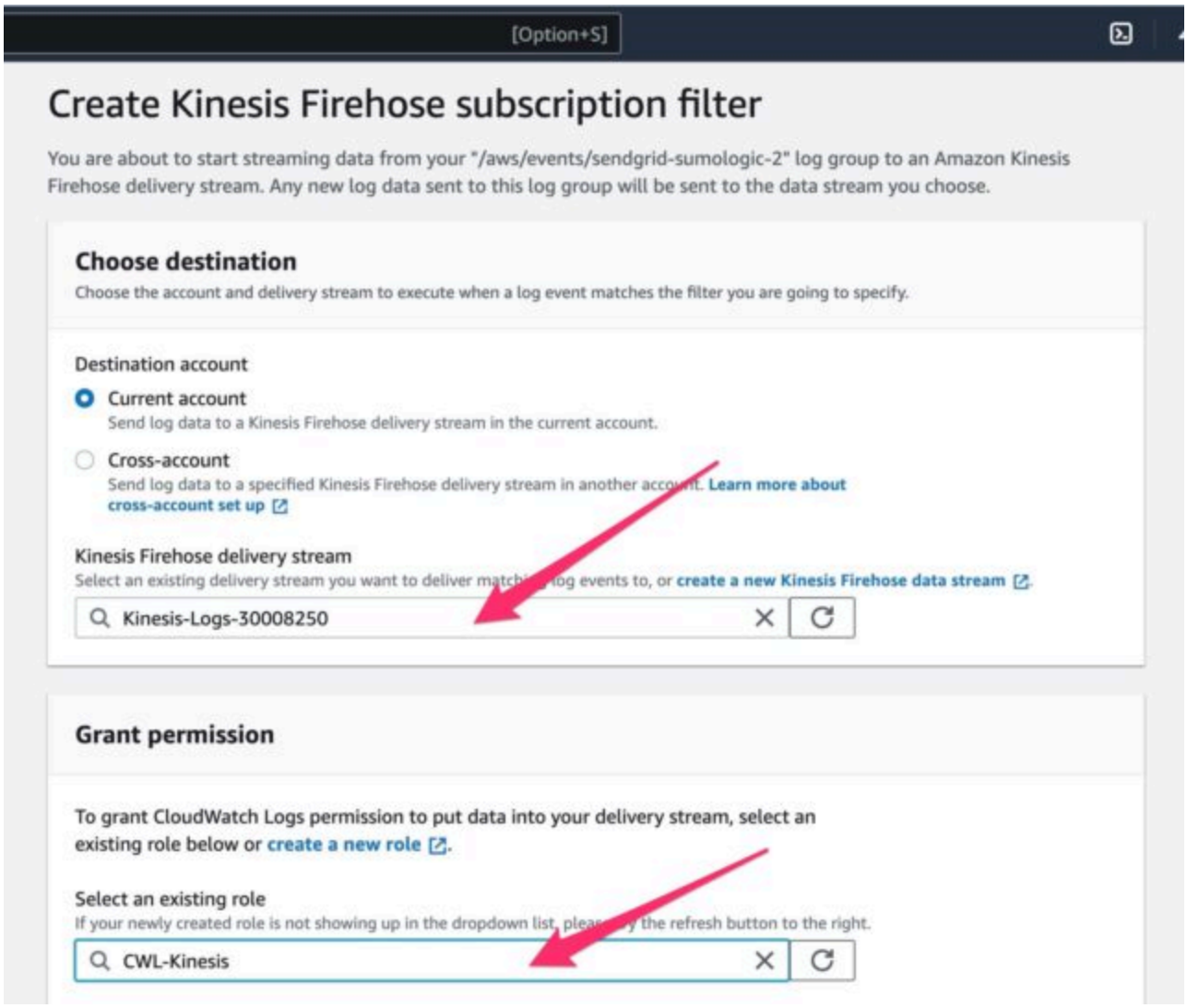
Next, set up a subscription filter for the target Log groups in CloudWatch Logs.

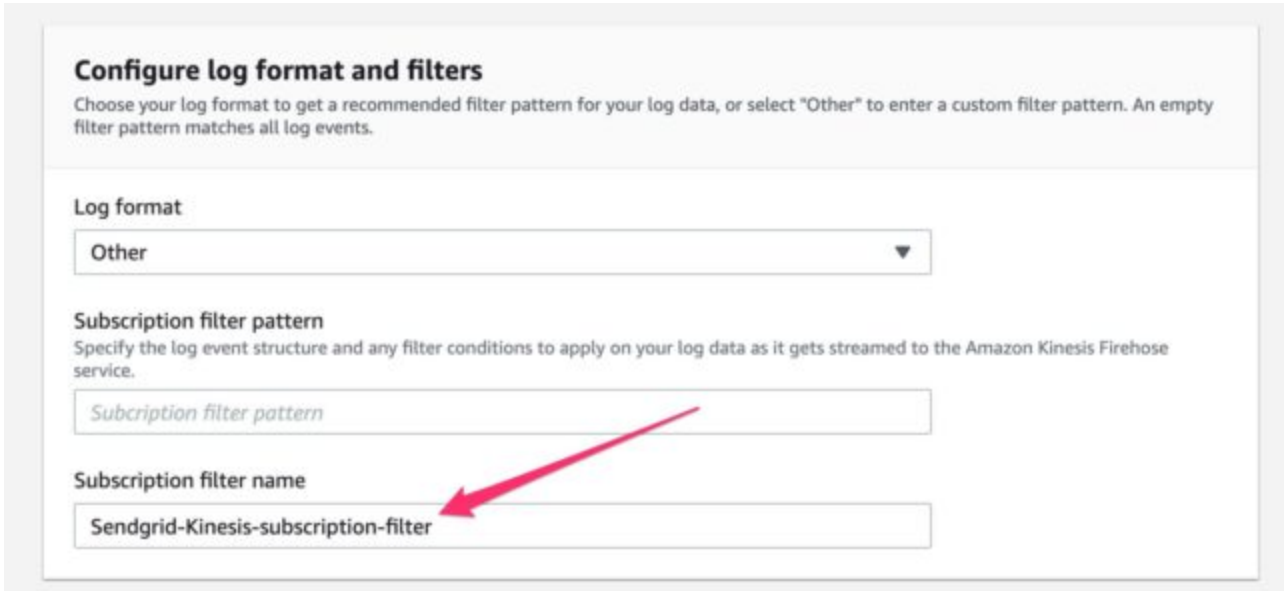


Create a subscription filter for Kinesis Firehose.

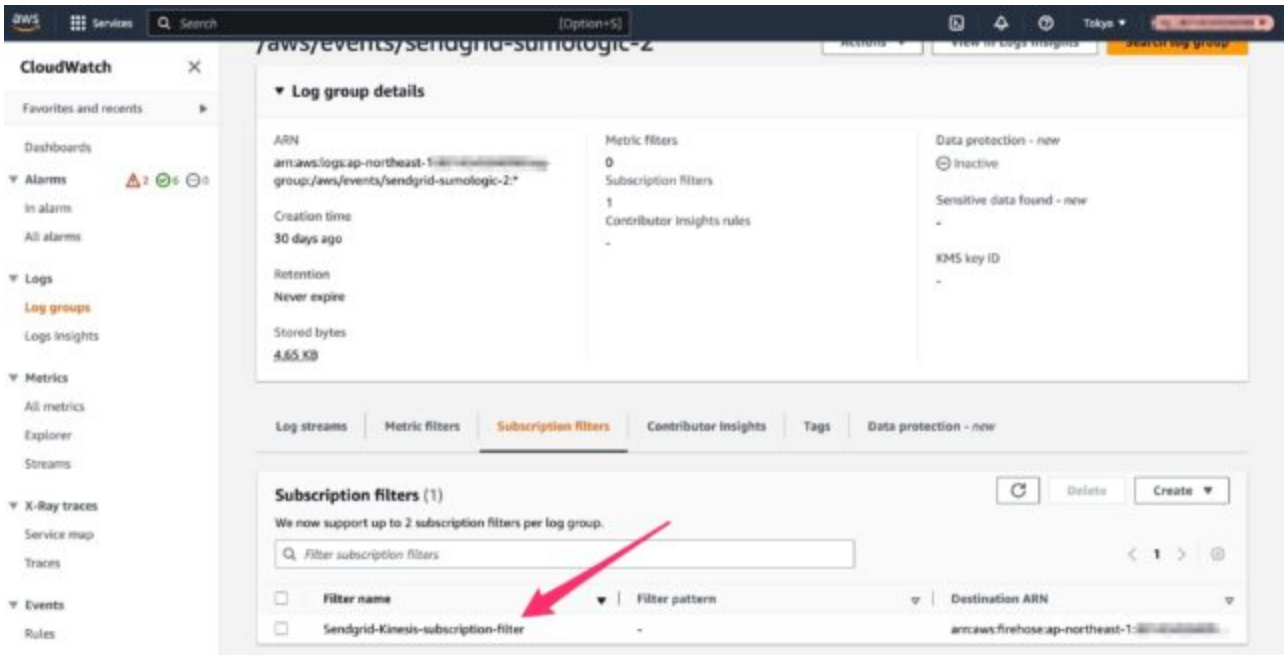


Select the Delivery streams you are creating, the role, and enter a subscription name.



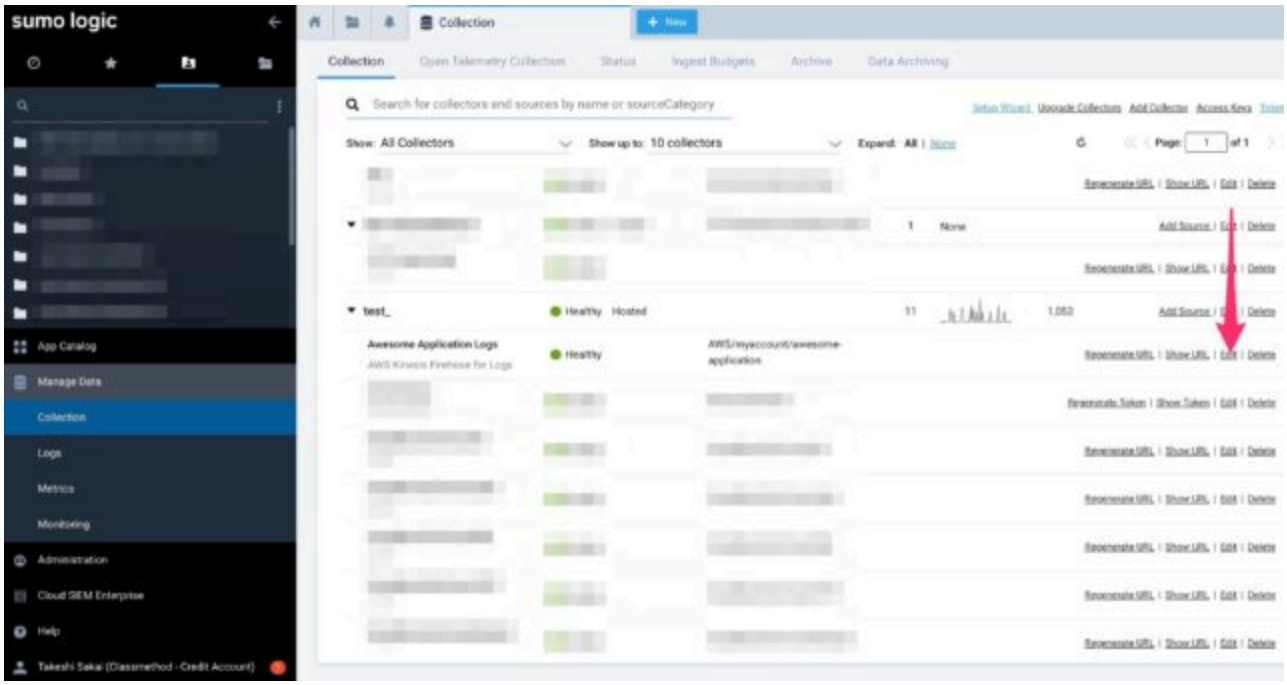


Then check that it's done.

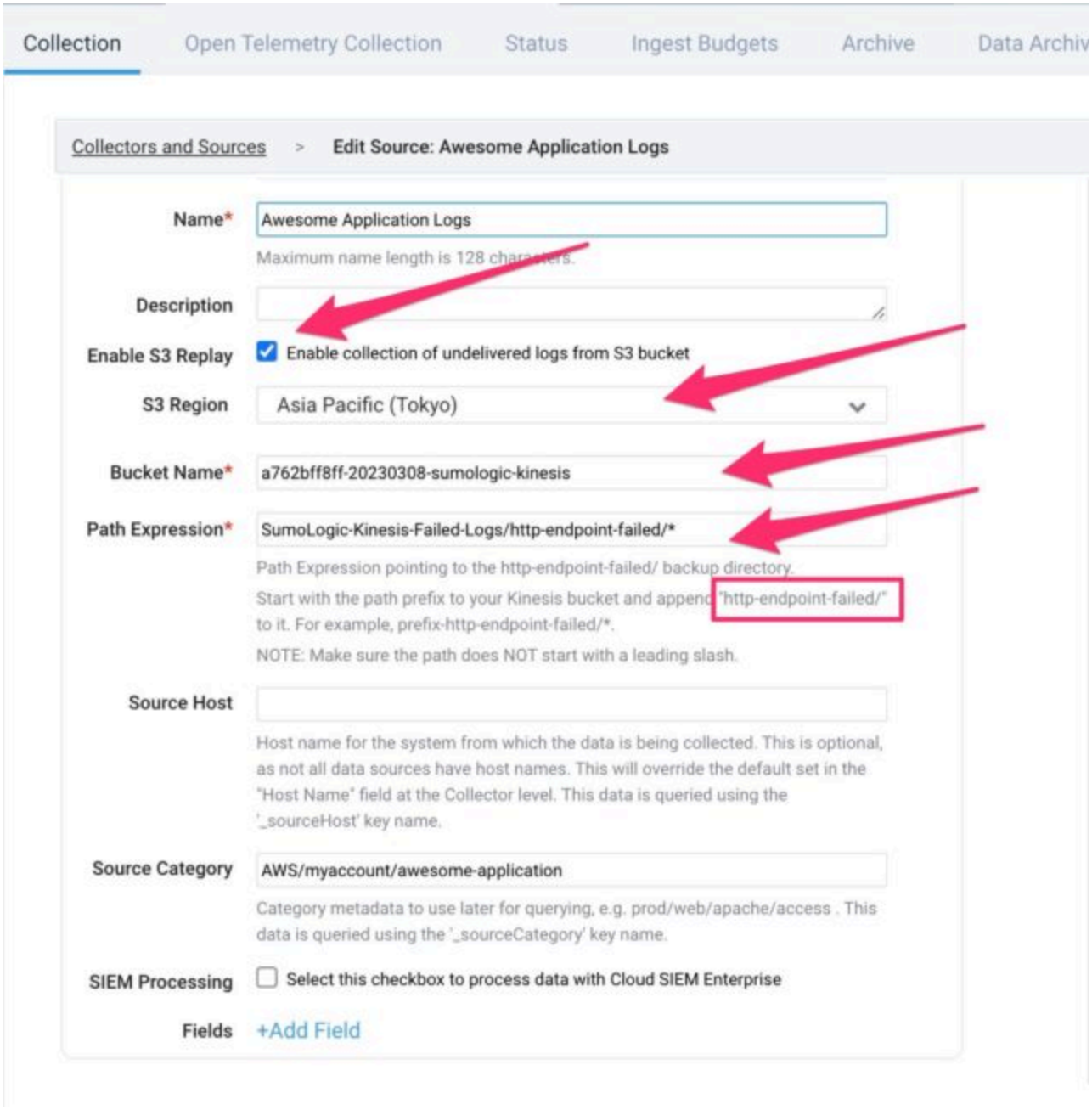


Sumo Logic Source Settings: Kinesis Data Firehose Log Delivery Error Data Acquisition Settings

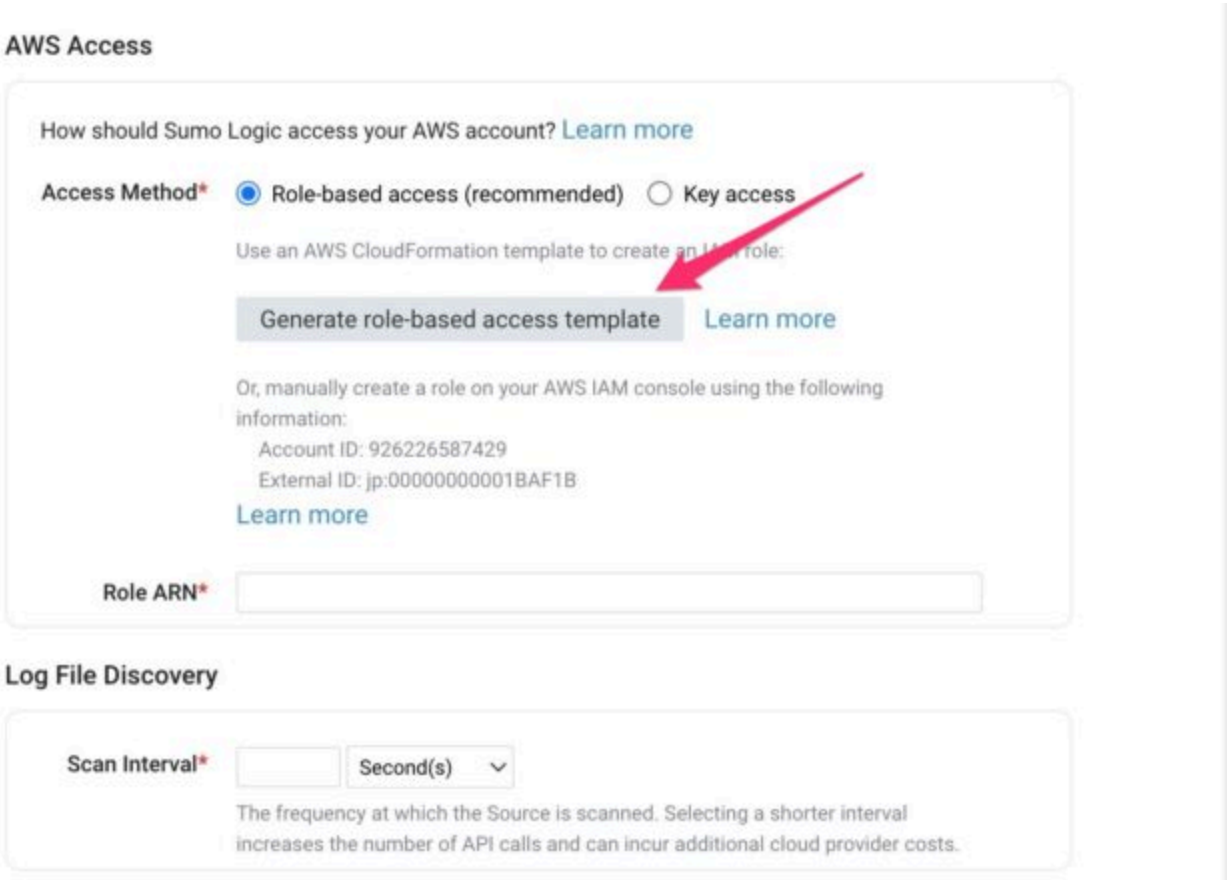
In the Sumo Logic source settings, enable Enable S3 Replay, which was previously unchecked. Edit the source you just created from Manage Data > Collection.



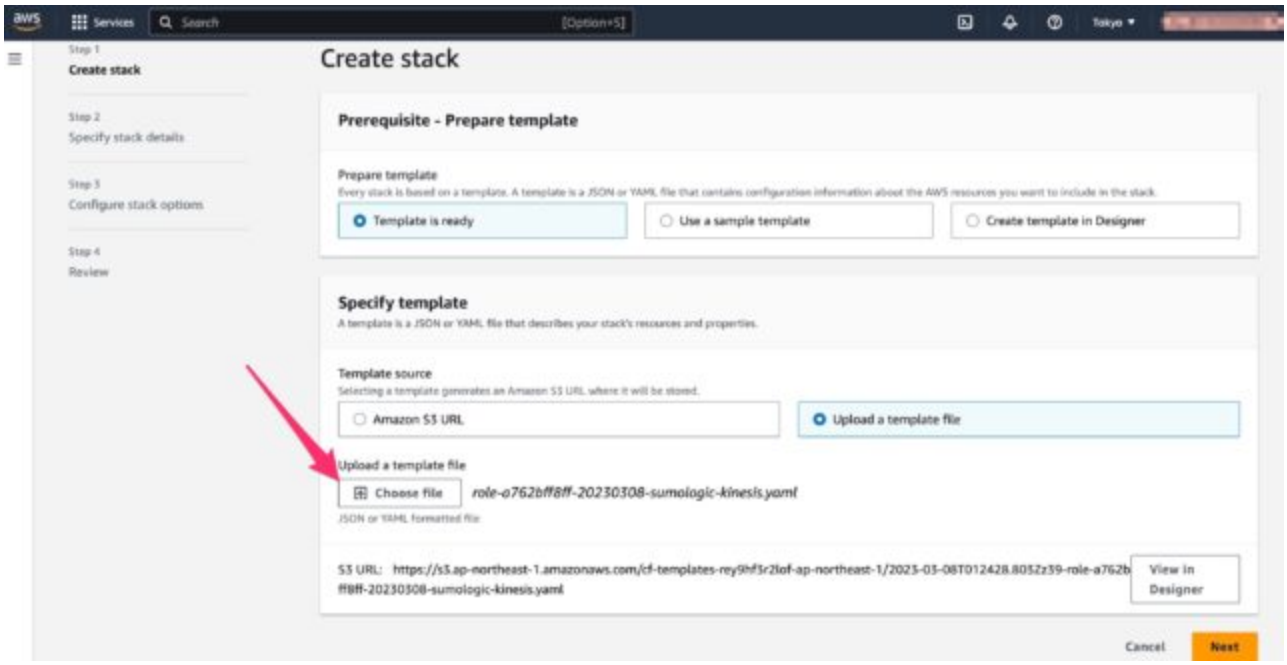
Check Enable S3 Replay to enable it, and enter the S3 region, bucket name, and path. The path is `http-endpoint-failed/`written to include the S3 folder name in the event of a delivery error when creating a Kinesis Data Firehose resource in CloudFormation, so specify it accordingly.

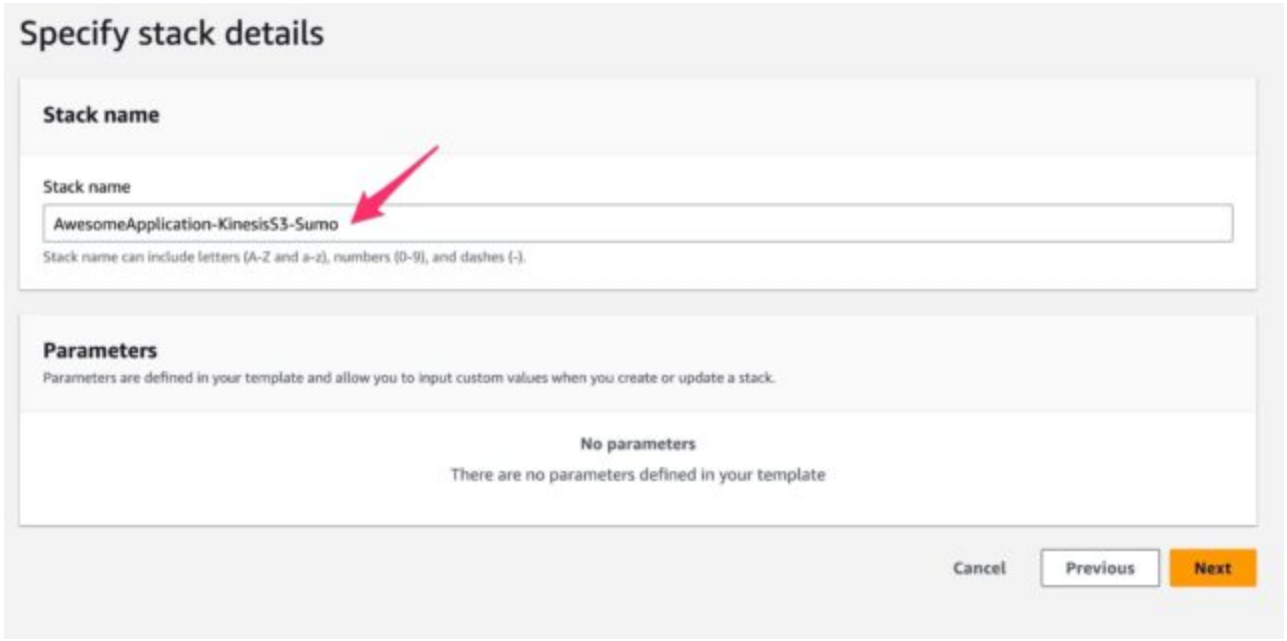


When you enable Enable S3 Replay, a setting item called AWS Access appears. Here, you can configure a role for Sumo Logic to access S3. `Generate role-based access template`Click to download the CloudFormation template.

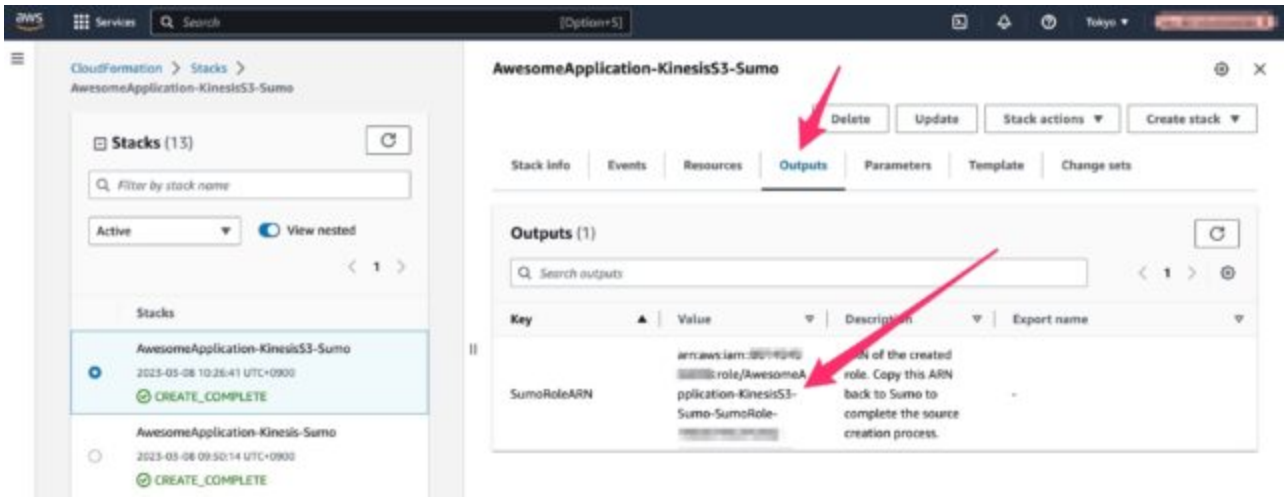


Log in to the AWS console again and create a CloudFormation stack. Specify the template you downloaded earlier, enter a stack name, and then run it by default.

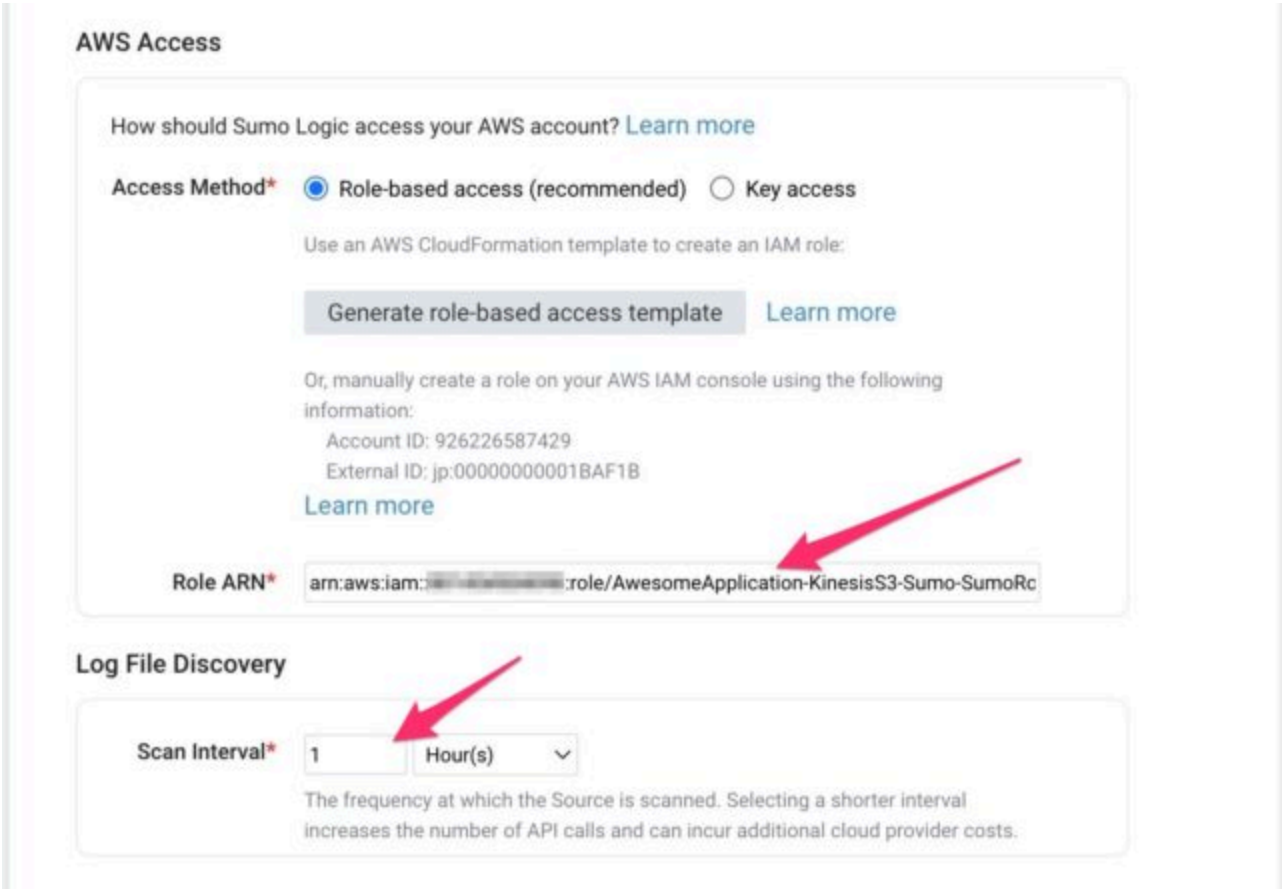




Verify that the AWS CloudFormation execution completes successfully, and copy the ARN of the role that was created.



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.

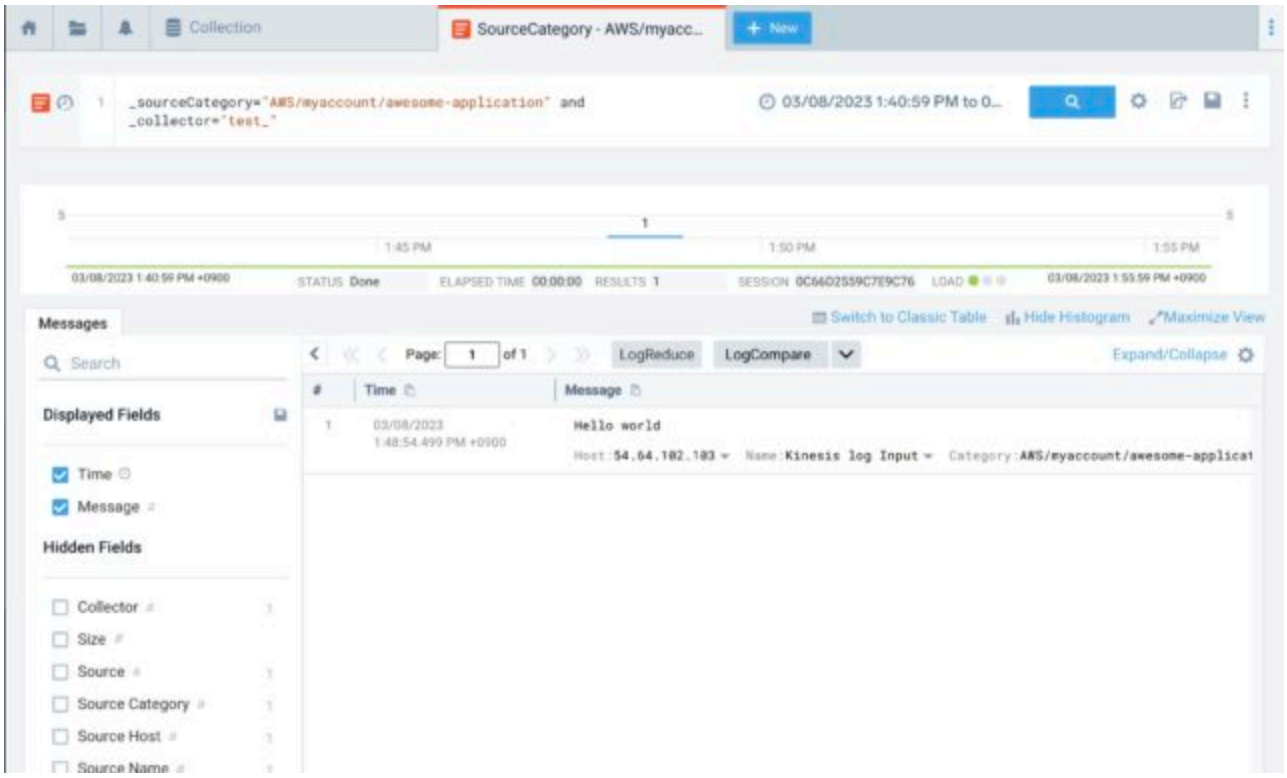


Verifying data transfer with Kinesis Data Firehose

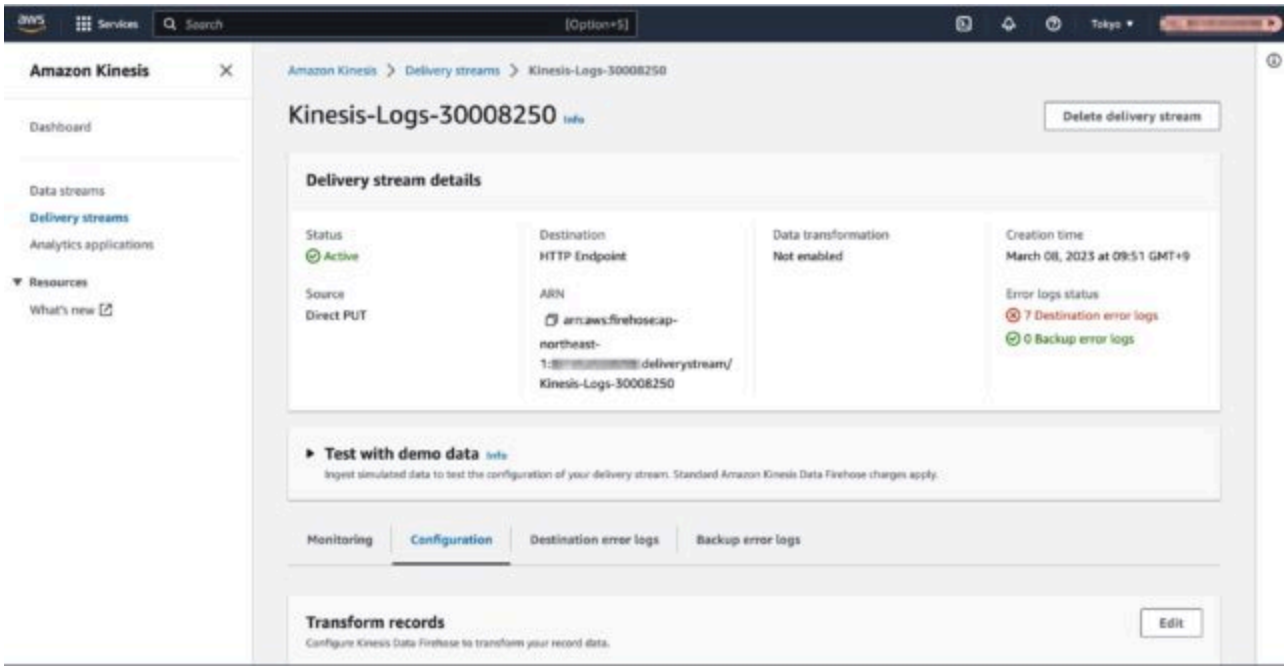
Now that the configuration on Sumo Logic and AWS is complete, let's check if the data is being sent correctly.
Let's try sending data from the CLI.

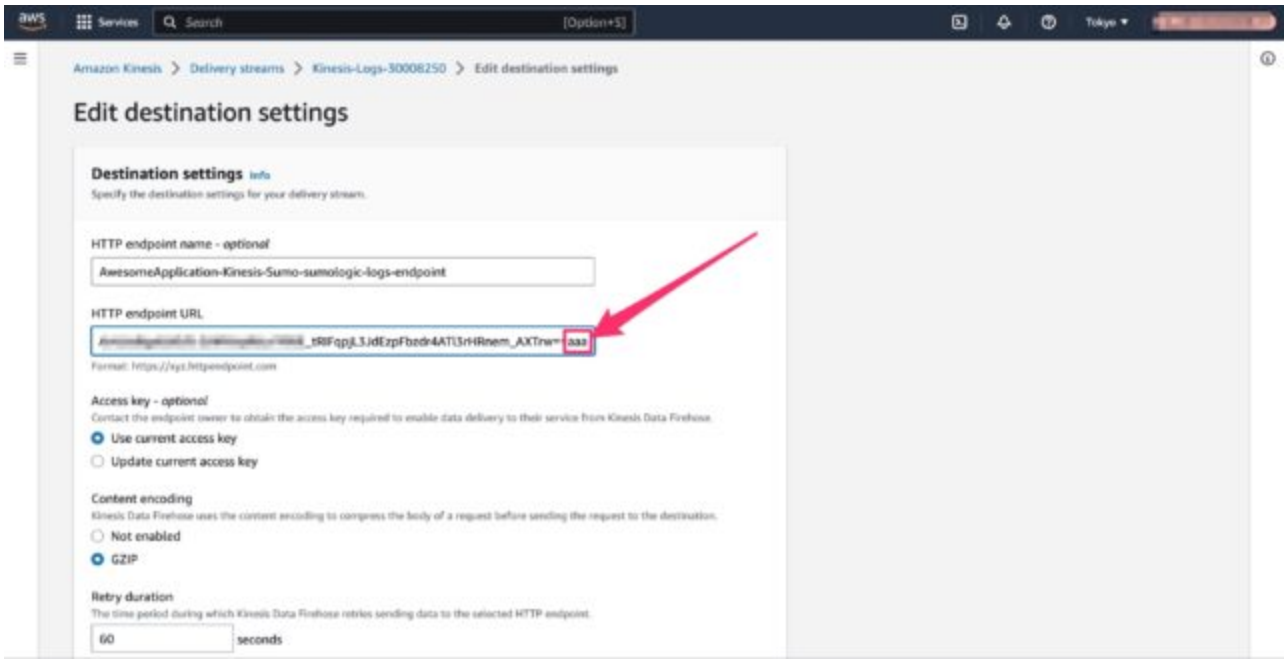
```
aws firehose put-record --delivery-stream-name Kinesis-Logs-30008250 \
--record '{"Data":"SGVsbG8gd29ybGQ="}'
```

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



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

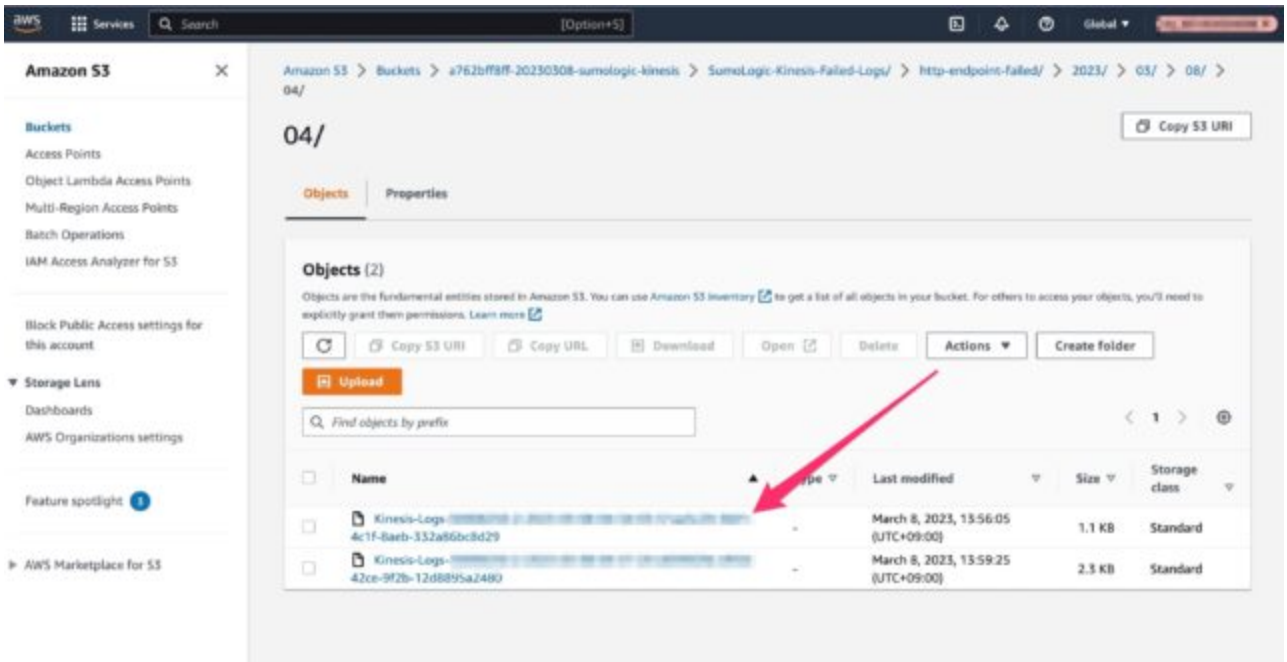




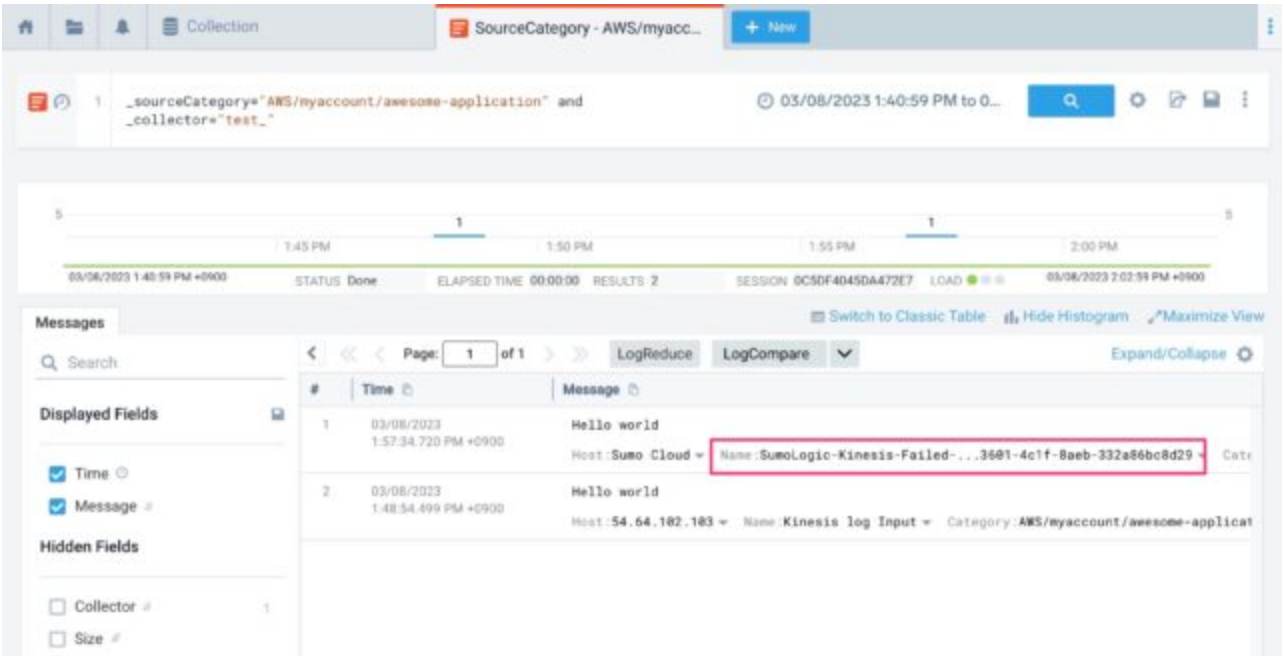
Send the data again from the CLI.

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

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

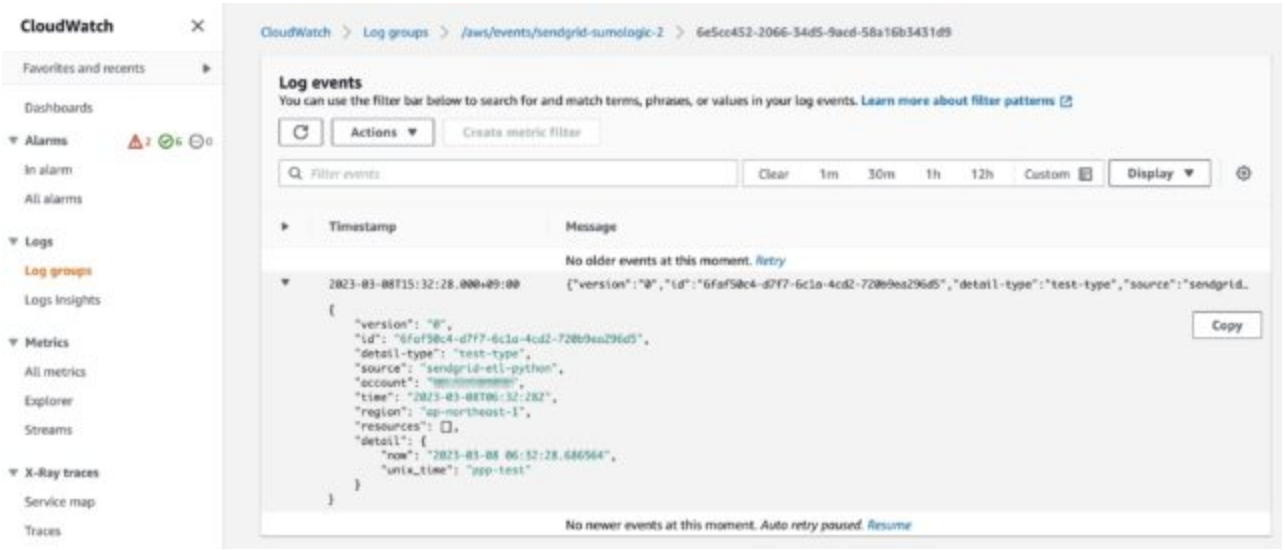


If you check in Sumo Logic, you will see that another line of log has arrived. This is the log for which delivery failed, retrieved by polling S3.

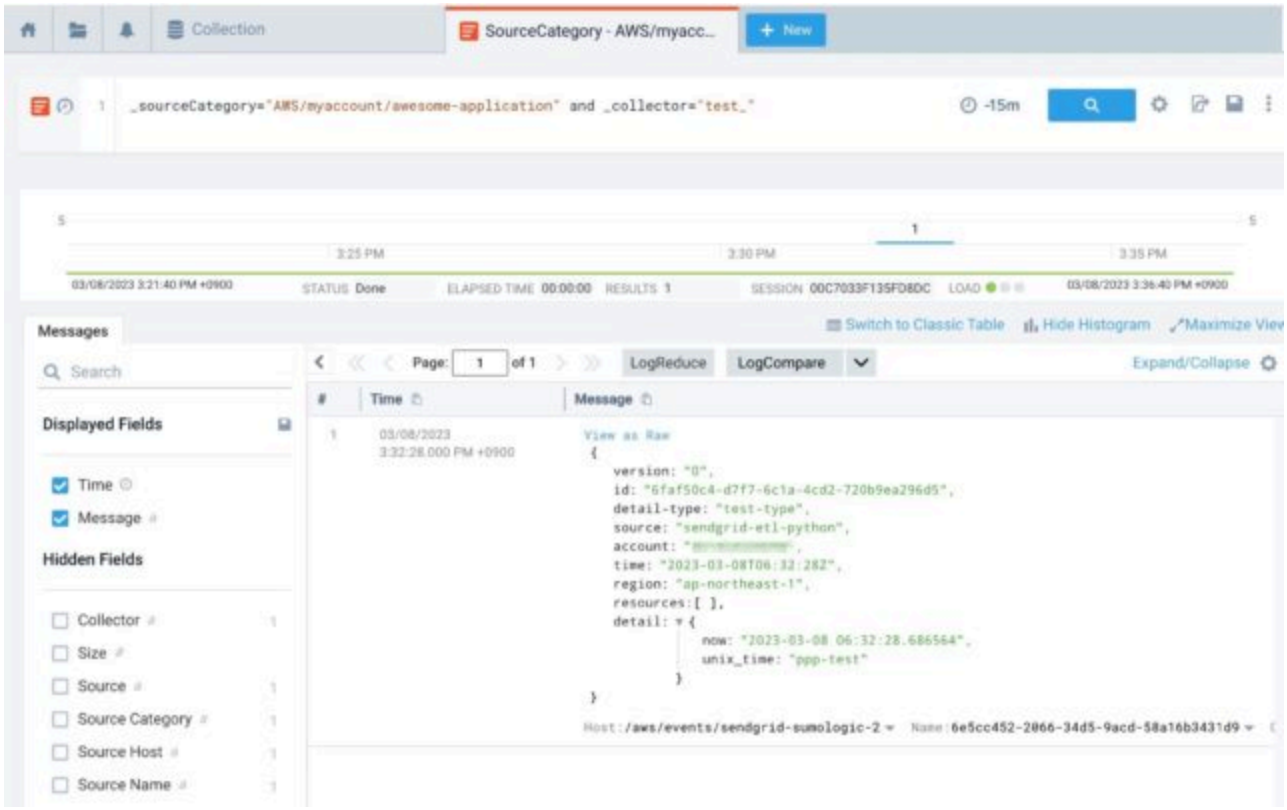


Verify that CloudWatch Logs data is being ingested

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.



summary

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