

Ingesting Any Log from AWS Cloudwatch Logs via Firehose

Setup instructions

We can follow most of what we need to setup Firehose and Splunk from [this blog](#) – this will walk through most of the steps needed.

Note the important changes below:

We will first start to follow the “Walkthrough” guidance in the blog to set up HEC, noting the following changes:

When setting up HEC on Splunk, set the sourcetype to `aws:cloudwatchlogs` (although this is optional as we over-ride this with the Environment variable in the transform later)

Continue with following the steps to create the splashback S3 Bucket and IAM Role.

Before creating the Firehose Stream, we will create a new Lambda function to transform the log into a Splunk JSON event.

In the AWS Console, navigate to Lambda, and create a new Function. Name it **FirehoseSplunkTransform**, change the runtime to Python 2.7, and then select the Existing role *LambdaBasicRole* created from the blog.

The screenshot shows the 'Create function' page in the AWS Lambda console. At the top, there are two tabs: 'Author from scratch' (selected) and 'Blueprints'. Below the tabs, the 'Author from scratch' section is active, showing a form to create a new function. The form includes the following fields:

- Name:** A text input field containing 'FirehoseSplunkTransform'.
- Runtime:** A dropdown menu showing 'Python 2.7'.
- Role:** A dropdown menu showing 'Choose an existing role'.
- Existing role:** A dropdown menu showing 'LambdaBasicRole'.

Below the form, there is a section for 'Existing role' with a note: 'You can use an existing role with this function. Lambda must be able to assume this role, and the role must have Amazon CloudWatch Logs permissions.'

Click “Create Function”, and on the next page, scroll down to the Lambda code window. Open up the [lambda code available here](#), and copy all of the python code into your new function (replacing all the default content). This code is essentially taken from the AWS Lambda Blueprint, and updated to do some transformations (see below).



Scroll down to the Environment Variables, and type in `SPLUNK_SOURCETYPE` in the key, and the sourcetype of the cloudwatch log into the value. For example, this could simply be `aws:cloudwatchlogs`. Save the function.


Now continue with the previous blog setup with the “Create Firehose Stream”. Note here however that on the Firehose setup, instead of creating a new function, select the Function **FirehoseSplunkTransform** which we created earlier.


Transform source records with AWS Lambda

To return records from AWS Lambda to Kinesis Firehose after transformation, the Lambda function you invoke must be compliant with the required record transformation output model. [Learn more](#)

Record transformation* ☐ Disabled ☒ Enabled

Lambda function* FirehoseSplunkTransform  

[View FirehoseSplunkTransform in Lambda](#) 

Lambda function version* \$LATEST 

Runtime python2.7

Another important change is when selecting Splunk as the destination, select the endpoint type as **Event** vs **Raw**. This is a key change, as the event sent to Splunk will be in JSON format.


Splunk destination

Firehose accesses Splunk instances through an endpoint and an authentication token. Generate the endpoint and authentication token by enabling the HTTP Event Collector (HEC) on your Splunk instances. [Learn more](#)

To grant Firehose access to an on-premises data center and Splunk instance, ensure [proper network configurations](#).

Splunk cluster endpoint*

Splunk endpoint type ☐ Raw endpoint
Capable of parsing most common log formats. [View supported log formats.](#)

 ☒ Event endpoint
Requires specific JSON formatting. Use the Firehose data transformation feature to properly format source data.

Authentication token*

☐ Show token

HEC acknowledgement timeout* seconds ⓘ
Specify a timeout duration from 180 - 600 seconds

Retry duration* seconds ⓘ
Specify a retry duration from 0 - 7200 seconds

Follow the remainder of the blog until you reach “Create a VPC Flow Log”.

At this point, you can now create a Cloudwatch Log group that you wish to use if you haven’t already a Log Group (or just follow the setup for VPC Flow logs if you wish to test with VPC flow logs).

If you already have a log group, then you can jump straight to “Publish CloudWatch to Kinesis Data Firehose”.

You should be able to follow the remainder of the blog instructions to create a subscription filter for the Cloudwatch Log group.

Once this is done, data should start flowing from the Cloudwatch Log into Splunk!

You can now create further subscription filters to this Firehose Stream from other CloudWatch logs.