

EXPLORER

OPEN EDITORS

- ! serverless.yml
- JS notify-user.js functions
- JS accept-order.js functions
- JS notify-restaurant.js functio...

BIG-MOUTH

- .serverless
- .vscode
- examples
- functions
- JS accept-order.js
- JS fulfill-order.js
- JS get-index.js
- JS get-restaurants.js
- JS notify-restaurant.js
- JS notify-user.js
- JS place-order.js
- JS retry-notify-restaurant.js
- JS search-restaurants.js
- lib
- node_modules
- static
- tests
- .gitignore
- build.sh
- ! buildspec.yml
- { } package-lock.json
- { } package.json
- README.md
- JS seed-restaurants.js
- ! serverless.yml

BOOKMARKS

! serverless.yml JS notify-user.js JS accept-order.js X JS notify-restaurant.js

```
4 const AWS = require('aws-sdk'); 1.5M (gzipped: 228.2K)
5 const kinesis = new AWS.Kinesis();
6 const streamName = process.env.order_events_stream;
7
8 module.exports.handler = co.wrap(function* (event, context, cb) {
9   let body = JSON.parse(event.body);
10  let restaurantName = body.restaurantName;
11  let orderId = body.orderId;
12  let userEmail = body.userEmail;
13
14  console.log(`restaurant [${restaurantName}] accepted order ID [${orderId}] from user [${userEmail}]`);
15
16  let data = {
17    orderId,
18    userEmail,
19    restaurantName,
20    eventType: 'order_accepted'
21  }
22
23  let req = {
24    Data: JSON.stringify(data), // the SDK would base64 encode this for us
25    PartitionKey: orderId,
26    StreamName: streamName
27  };
28
29  yield kinesis.putRecord(req).promise();
30
31  console.log(`published 'order_accepted' event into Kinesis`);
32
33  let response = {
34    statusCode: 200,
35    body: JSON.stringify({ orderId })
36  }
37
38  cb(null, response);
39});
```



Services ▾

Resource Groups ▾



DynamoDB



Lam



yan.cui @ theburningmonk ▾

N. Virginia ▾

Support ▾

CloudWatch

Dashboards

Alarms

ALARM

1

INSUFFICIENT

9

OK

0

Billing

Events

Rules

Event Buses

Logs

Metrics

Favorites

+ Add a dashboard

CloudWatch > Log Groups > /aws/lambda/big-mouth-dev-accept-order > 2018/02/24/[&LATEST]0c12d3eaccc14ee598d2d5f7507befd9

Expand all



Row



Text

Filter events all 30s 5m 1h 6h 1d 1w custom ▾

Time (UTC +00:00)	Message
2018-02-24	No older events found at the moment. Retry.
▶ 14:32:33	START RequestId: 8cfcebd1-196f-11e8-a058-19dae9188bdf Version: \$LATEST
▼ 14:32:33	2018-02-24T14:32:33.784Z 8cfcebd1-196f-11e8-a058-19dae9188bdf restaurant [Fangtasia] accepted order ID [875ae3c4-c21c-59e6-9057-23694b75f5a2] from user [theburningmonk@gmail.com]
▼ 14:32:33	2018-02-24T14:32:33.916Z 8cfcebd1-196f-11e8-a058-19dae9188bdf published 'order_accepted' event into Kinesis
	2018-02-24T14:32:33.916Z 8cfcebd1-196f-11e8-a058-19dae9188bdf published 'order_accepted' event into Kinesis
▶ 14:32:33	END RequestId: 8cfcebd1-196f-11e8-a058-19dae9188bdf
▶ 14:32:33	REPORT RequestId: 8cfcebd1-196f-11e8-a058-19dae9188bdf Duration: 136.13 ms Billed Duration: 200 ms Memory
	No newer events found at the moment. Retry.

Feedback

English (US)

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

[Privacy Policy](#)[Terms of Use](#)



Services ▾

Resource Groups ▾



DynamoDB



Lam



yan.cui @ theburningmonk

N. Virginia ▾

Support ▾

CloudWatch

Dashboards

Alarms

ALARM

1

INSUFFICIENT

9

OK

0

Billing

Events

Rules

Event Buses

Logs

Metrics

Favorites

+ Add a dashboard

CloudWatch > Log Groups > /aws/lambda/big-mouth-dev-accept-order > 2018/02/24/[&LATEST]0c12d3eaccc14ee598d2d5f7507befd9

Expand all



Row



Text



Filter events

all

30s

5m

1h

6h

1d

1w

custom ▾

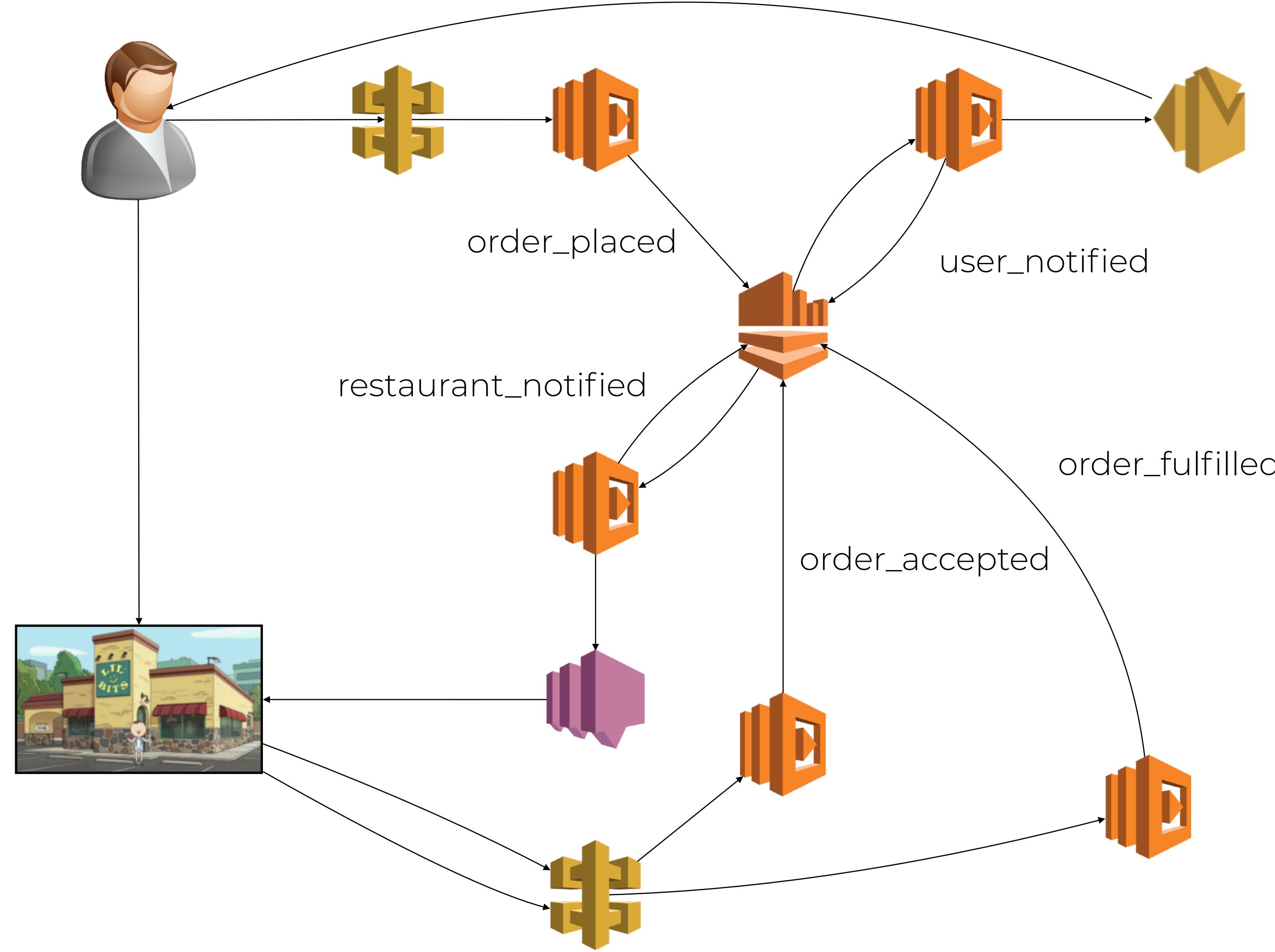
Time (UTC +00:00) Message

2018-02-24

No older events found at the moment. Retry.

▶ 14:32:33	START RequestId: 8cfcebd1-196f-11e8-a058-19dae9188bdf Version: \$LATEST
▼ 14:32:33	2018-02-24T14:32:33.784Z 8cfcebd1-196f-11e8-a058-19dae9188bdf restaurant [Fangtasia] accepted order ID [875ae3c4-c21c-59e6-9057-23694b75f5a2] from user [theburningmonk@gmail.com]
▼ 14:32:33	2018-02-24T14:32:33.784Z 8cfcebd1-196f-11e8-a058-19dae9188bdf restaurant [Fangtasia] accepted order ID [875ae3c4-c21c-59e6-9057-23694b75f5a2] from user [theburningmonk@gmail.com]
▼ 14:32:33	2018-02-24T14:32:33.916Z 8cfcebd1-196f-11e8-a058-19dae9188bdf published 'order_accepted' event into Kinesis
2018-02-24T14:32:33.916Z 8cfcebd1-196f-11e8-a058-19dae9188bdf	published 'order_accepted' event into Kinesis
▶ 14:32:33	END RequestId: 8cfcebd1-196f-11e8-a058-19dae9188bdf
▶ 14:32:33	REPORT RequestId: 8cfcebd1-196f-11e8-a058-19dae9188bdf Duration: 136.13 ms Billed Duration: 200 ms Memory
No newer events found at the moment. Retry.	

no way to search across log groups





Services ▾

Resource Groups ▾



DynamoDB



Lam



yan.cui @ theburningmonk ▾

N. Virginia ▾

Support ▾

CloudWatch

Dashboards

Alarms

ALARM

1

INSUFFICIENT

9

OK

0

Billing

Events

Rules

Event Buses

Logs

Metrics

Favorites

+ Add a dashboard

CloudWatch > Log Groups

Create Metric Filter

Actions ▾



Log Groups	Expire Events After	Metric Filters	Subscriptions
<input type="radio"/> /aws/lambda/big-mouth-dev-accept-order	Never Expire	0 filters	None
<input type="radio"/> /aws/lambda/big-mouth-dev-fulfill-order	Never Expire	0 filters	None
<input checked="" type="radio"/> /aws/lambda/big-mouth-dev-get-index	Never Expire	0 filters	None
<input type="radio"/> /aws/lambda/big-mouth-dev-get-restaurants	Never Expire	0 filters	None
<input type="radio"/> /aws/lambda/big-mouth-dev-notify-restaurant	Never Expire	0 filters	None
<input type="radio"/> /aws/lambda/big-mouth-dev-notify-user	Never Expire	0 filters	None
<input type="radio"/> /aws/lambda/big-mouth-dev-place-order	Never Expire	0 filters	None
<input type="radio"/> /aws/lambda/big-mouth-dev-retry-notify-restaurant	Never Expire	0 filters	None
<input type="radio"/> /aws/lambda/big-mouth-dev-retry-notify-user	Never Expire	0 filters	None
<input type="radio"/> /aws/lambda/big-mouth-dev-search-restaurants	Never Expire	0 filters	None



Services

Resource Groups



DynamoDB



Lam



yan.cui @ theburningmonk

N. Virginia

Support

CloudWatch

Dashboards

Alarms

ALARM

1

INSUFFICIENT

9

OK

0

Billing

Events

Rules

Event Buses

Logs

Metrics

Favorites

+ Add a dashboard

CloudWatch > Log Groups

Create Metric Filter

Actions



Filter: /aws/lambda/bi

Log Groups

 /aws/lambda/big-m /aws/lambda/big-m /aws/lambda/big-m /aws/lambda/big-m /aws/lambda/big-m /aws/lambda/big-m /aws/lambda/big-m /aws/lambda/big-m /aws/lambda/big-m /aws/lambda/big-mouth-dev-retry-notify-restaurant /aws/lambda/big-mouth-dev-retry-notify-user /aws/lambda/big-mouth-dev-search-restaurants

Create log group

Delete log group

Export

Export data to Amazon S3

View all exports to Amazon S3

Subscriptions

Stream to AWS Lambda

Stream to Amazon Elasticsearch Service

Remove Subscription Filter

Expire Events After

Never Expire

Metric Filters

0 filters

Applause from Ben Kehoe, Robin Taylor, and 447 others



Liz Bennett [Follow](#)
Engineer working on data platforms.
Jun 17, 2017 · 5 min read

Some things you should know before using Amazon's Elasticsearch Service on AWS

Elasticsearch is a powerful but fragile piece of infrastructure with a ton of things that can cause the AWS service to become unstable



<http://bit.ly/things-to-know-about-aws-elk>



Services ▾

Resource Groups ▾



DynamoDB



Lam



yan.cui @ theburningmonk ▾

N. Virginia ▾

Support ▾

CloudWatch

Dashboards

Alarms

ALARM

1

INSUFFICIENT

9

OK

0

Billing

Events

Rules

Event Buses

Logs

Metrics

Favorites

+ Add a dashboard

CloudWatch > Log Groups

Create Metric Filter

Actions ▾



Filter: /aws/lambda/big-

Log Groups

/aws/lambda/big-m

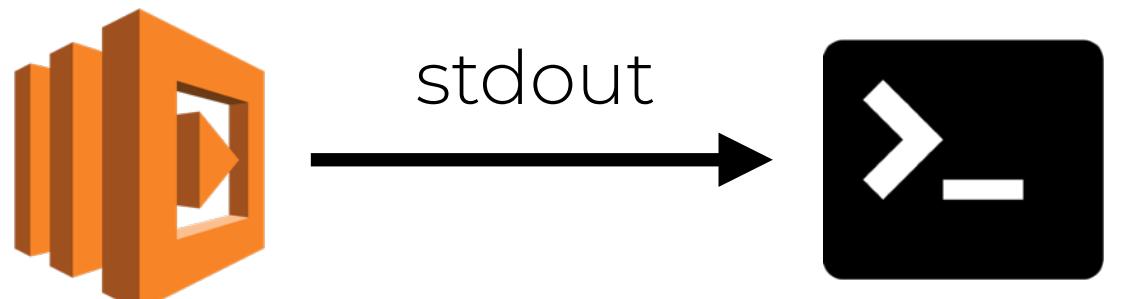
/aws/lambda/big-mouth-dev-retry-notify-restaurant

/aws/lambda/big-mouth-dev-retry-notify-user

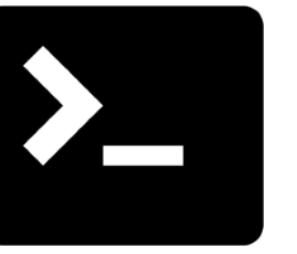
/aws/lambda/big-mouth-dev-search-restaurants

Expire Events After

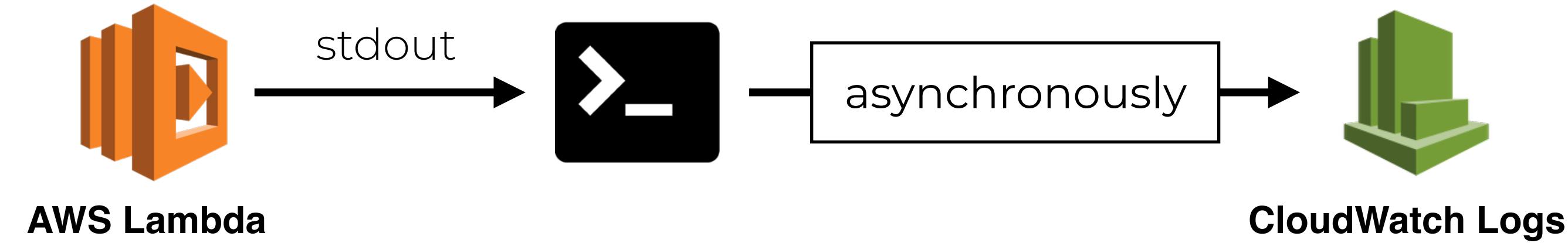
Never Expire

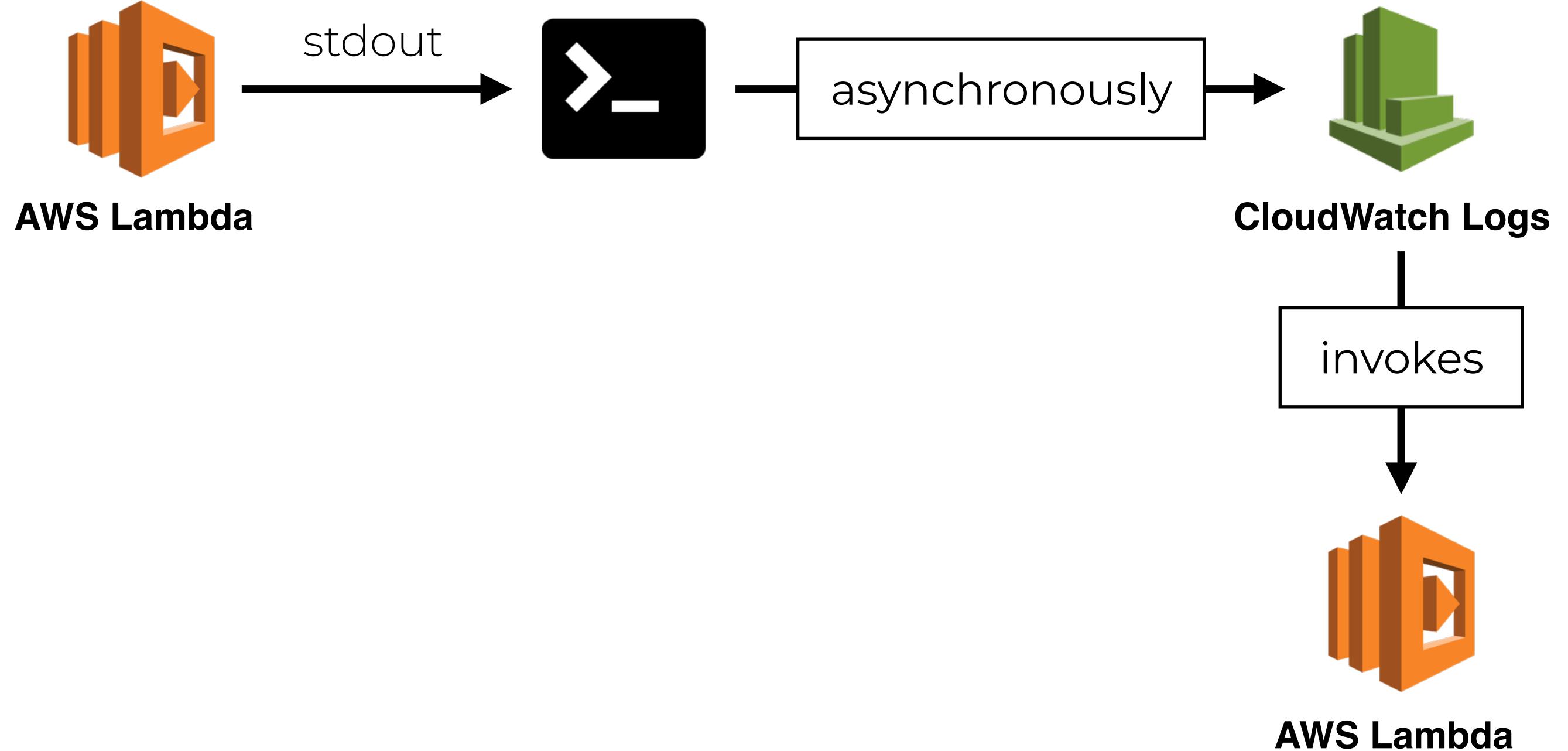


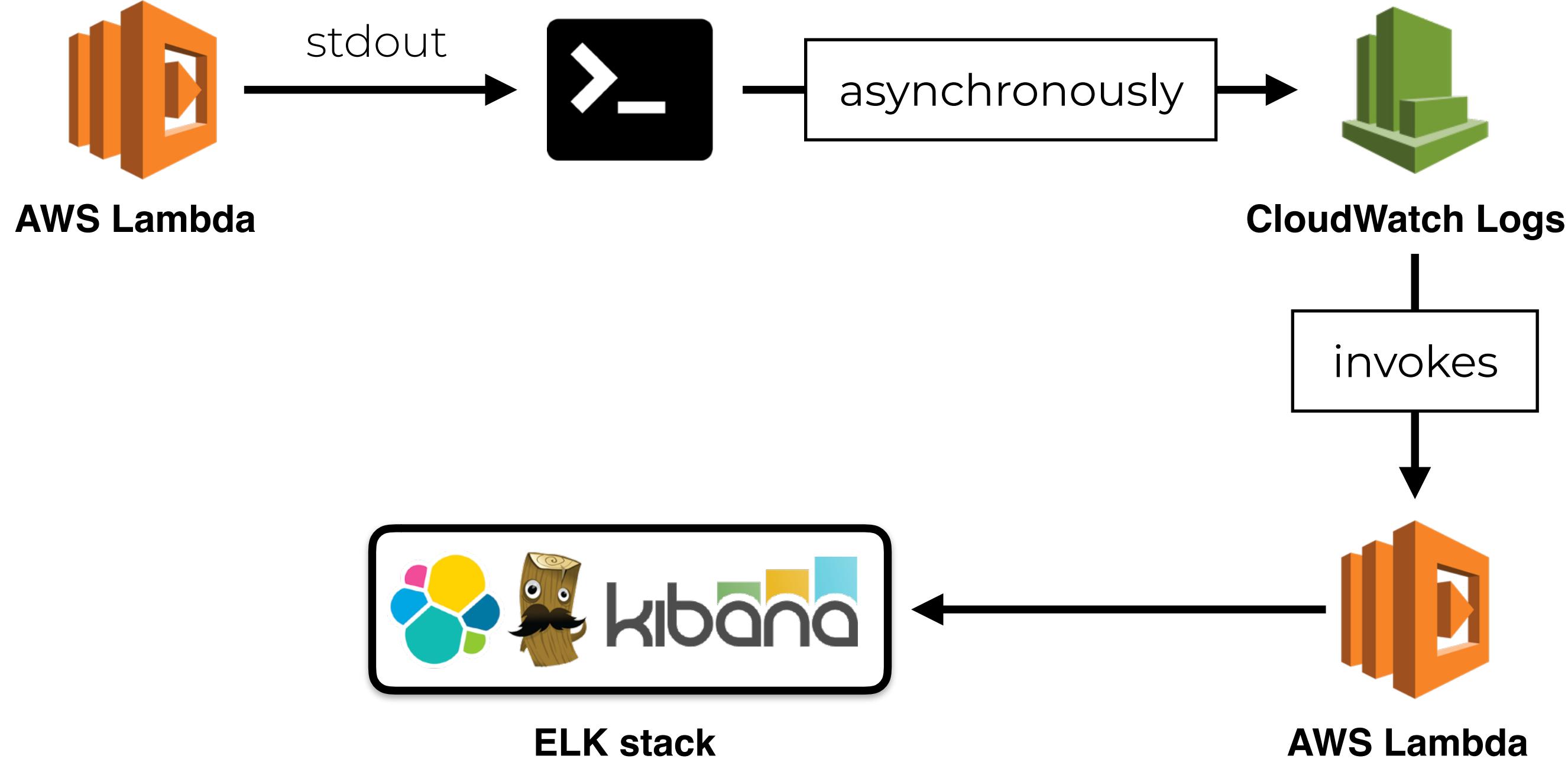
stdout

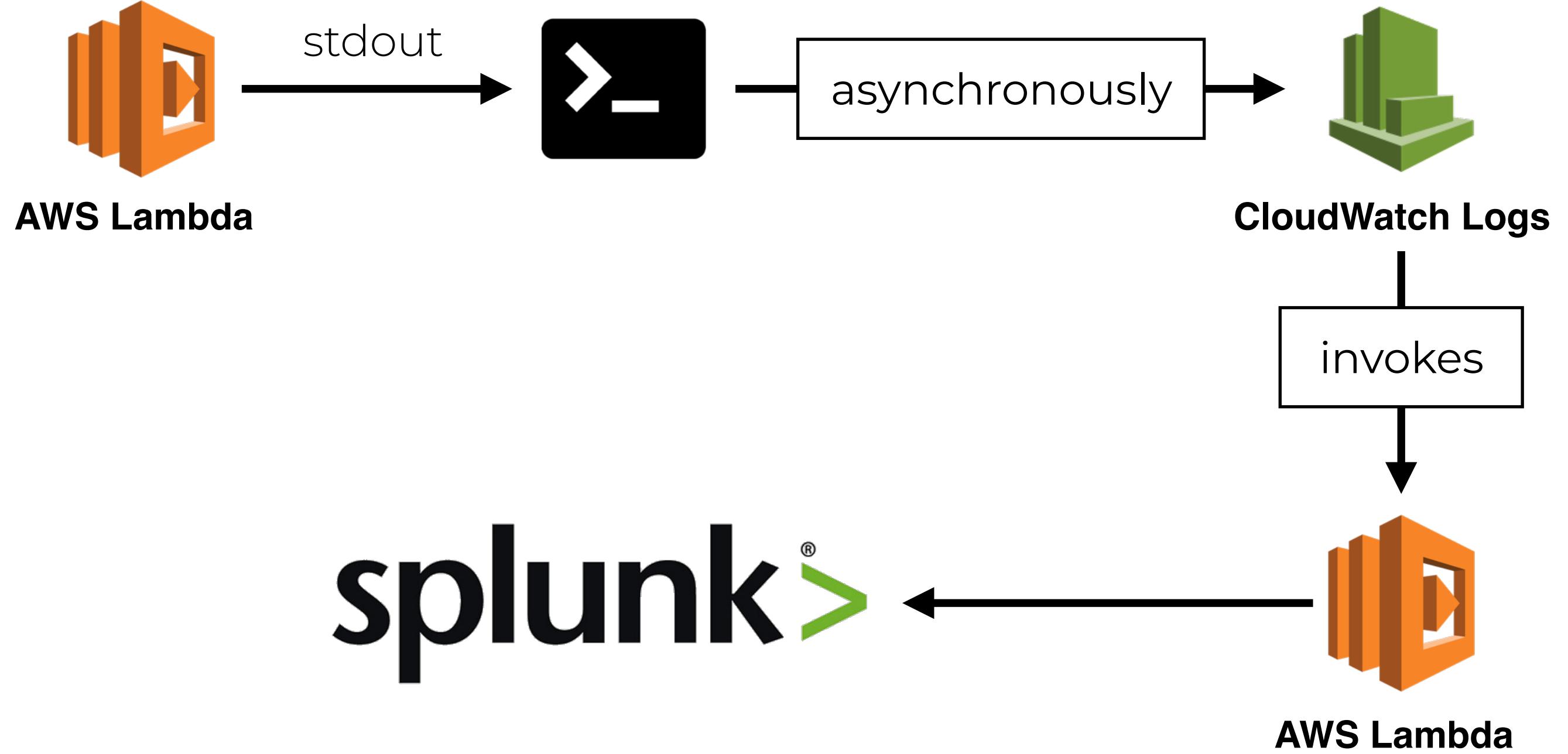


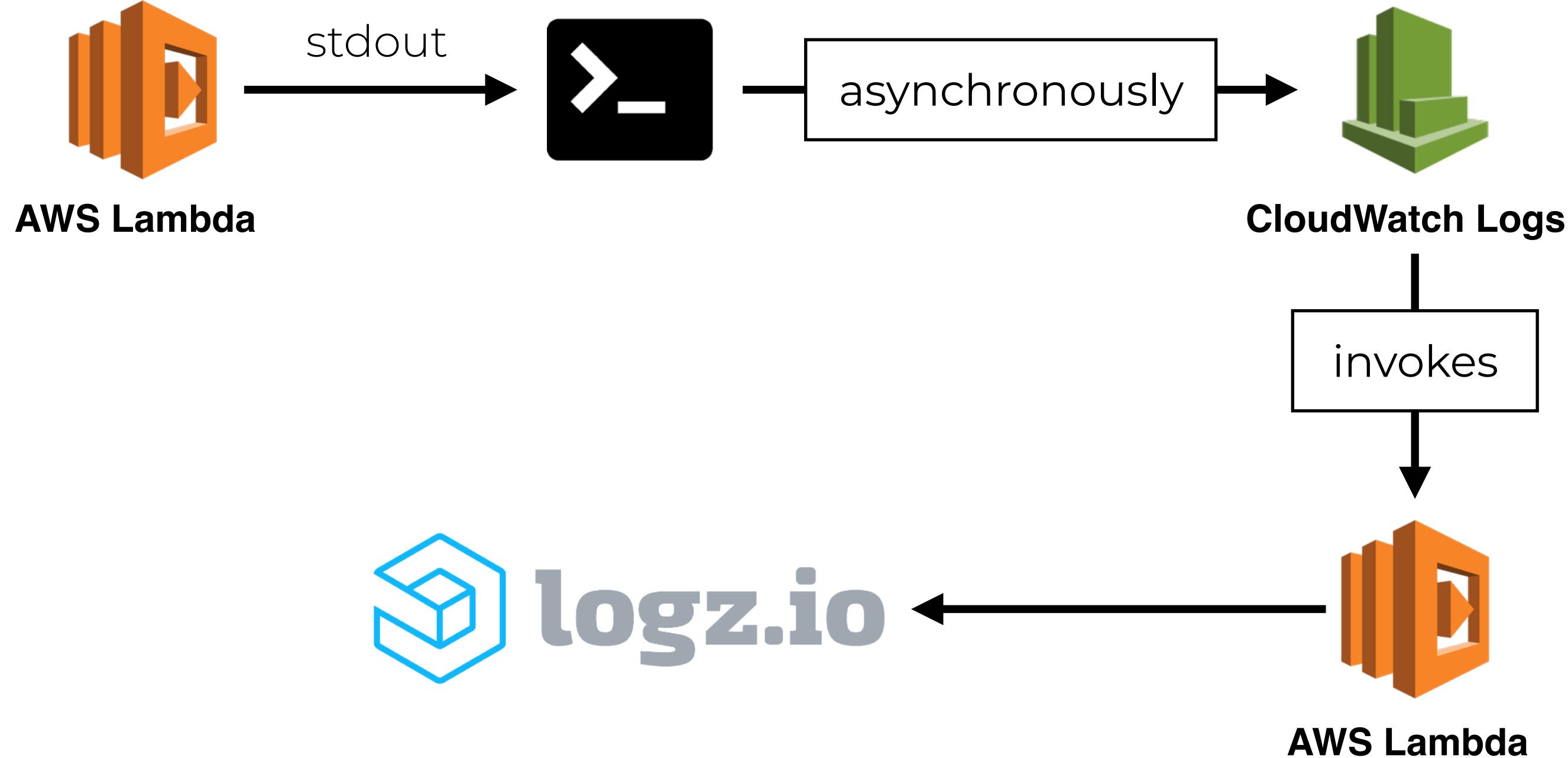
AWS Lambda

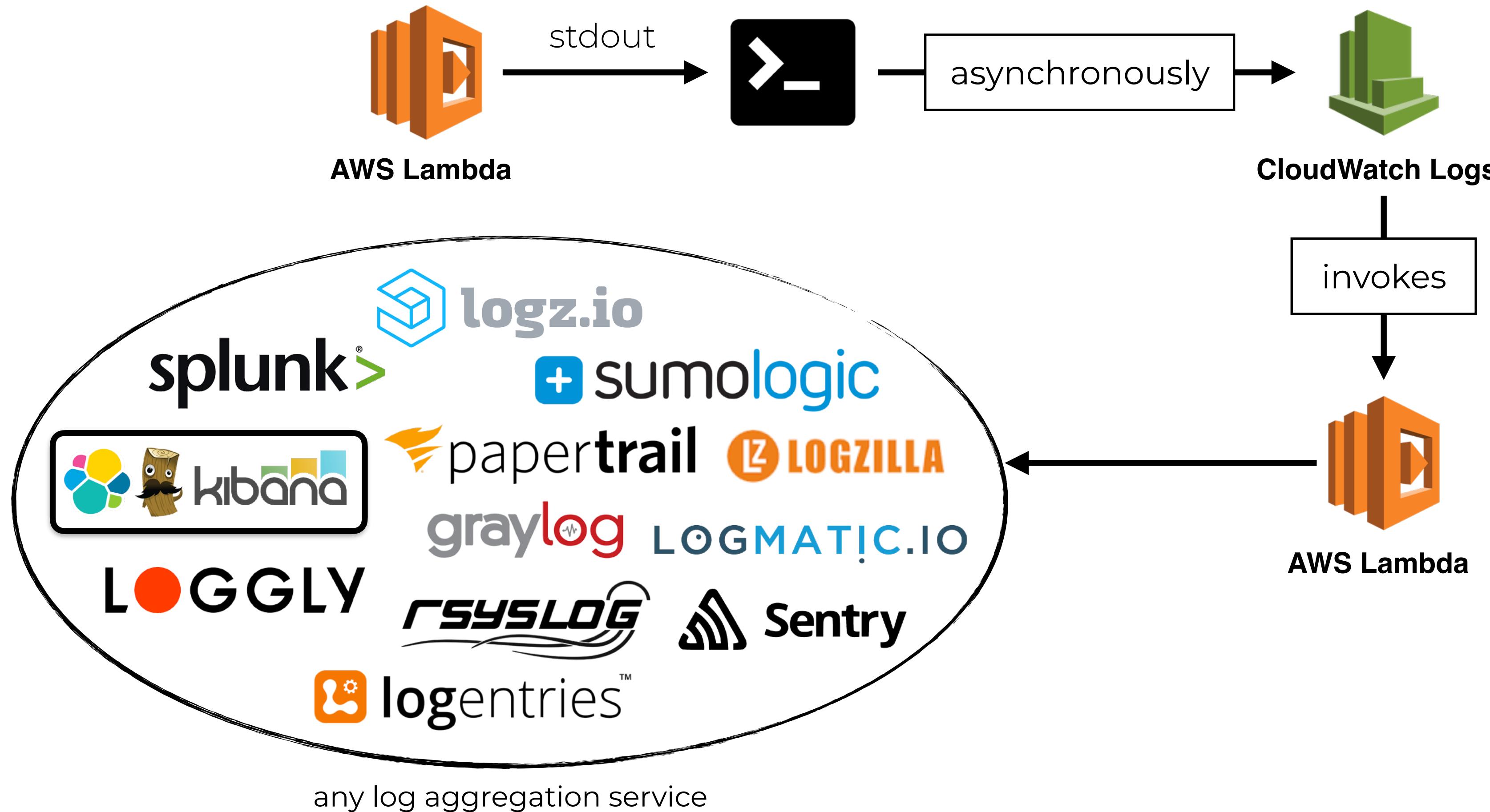














New!

Product

Solutions

Pricing

Company

Resources

Free Trial

Request Demo



Login

AI-Powered ELK as a Service

Easily ship, analyze and extract insights from your data!

 Enter your work Email

Free Trial

**TURNER****ORACLE®****BRITISH AIRWAYS****asurion** 

ELK that works at scale

Logz.io offers the world's most popular open-



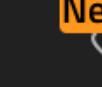
Resolve issues faster

Logz.io applies advanced machine learning



Cost efficiency as you grow

Logz.io provides a suite of analytics and



13 days left

Upgrade

theburningmonk limited (the...)



GENERAL >

MANAGE USERS >

MANAGE ACCOUNTS >

USAGE & BILLING >

SHARED TOKENS >

S3 ARCHIVING >

FIELD MAPPING >

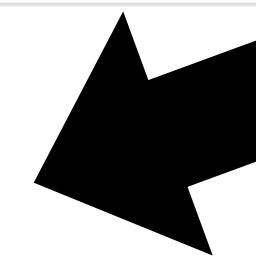
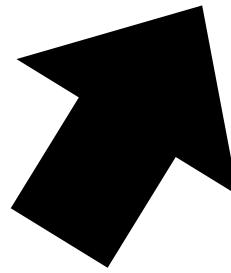
SYSTEM STATUS >

GENERAL

USER SETTINGS Yan Cui (theburningmonk@gmail.com)

Password [Change](#)Dashboard theme [Dark](#) [Light](#)

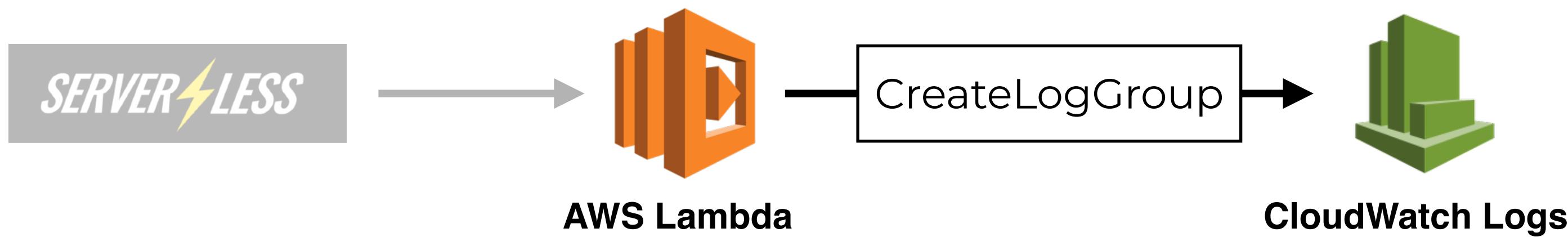
ACCOUNT SETTINGS theburningmonk limited (31112)

Account's token Audit trail [View](#)Support access [Enabled](#) [Disabled](#) Never expires [Change](#)[Refresh Kibana mapping](#)

SERVERLESS



AWS Lambda





Services ▾

Resource Groups ▾



DynamoDB



Lam



yan.cui @ theburningmonk ▾

N. Virginia ▾

Support ▾

CloudWatch
Dashboards
Alarms

ALARM

1

INSUFFICIENT

9

OK

0

Billing

Events

Rules

Event Buses

Logs

Metrics

Favorites

+ Add a dashboard

Step 1: Create rule

Create rules to invoke Targets based on Events happening in your AWS environment.

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

Event Pattern  Schedule 

Build event pattern to match events by service

Service Name

CloudWatch Logs

Event Type

AWS API Call via CloudTrail

For AWS API call events, CloudWatch Events supports the same read/write APIs as CloudTrail does. Read-only APIs, such as those that begin with **List**, **Get**, or **Describe** are not supported by CloudWatch Events. [See more details](#) about which services are supported by CloudTrail.

 Any operation Specific operation(s)

CreateLogGroup



Event Pattern Preview

Copy to clipboard Edit

```
{  
  "source": [  
    "aws.logs"]}
```

Targets

Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.

Lambda function

Function*

cloudwatch-logs-to-logzio-dev-subscribe

▶ Configure version/alias

▶ Configure input

+ Add target*

Creating a Trail in the Console

You can configure your trail for the following:

- Specify if you want the trail to apply to all regions or a single region.
- Specify an Amazon S3 bucket to receive log files.
- For management and data events, specify if you want to log read-only, write-only, or all events.

On this page:

Creating a Trail in the Console

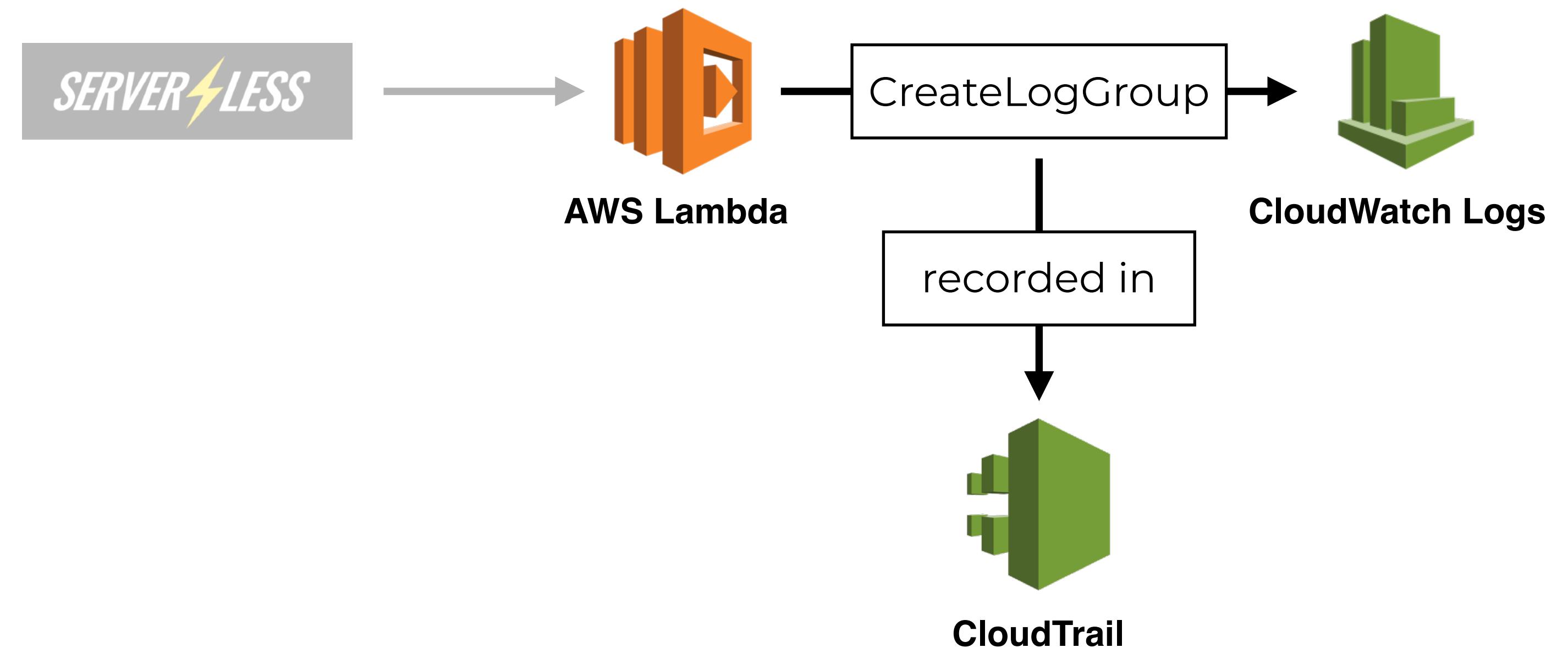
[Configuring Advanced Settings for Your Trail](#)

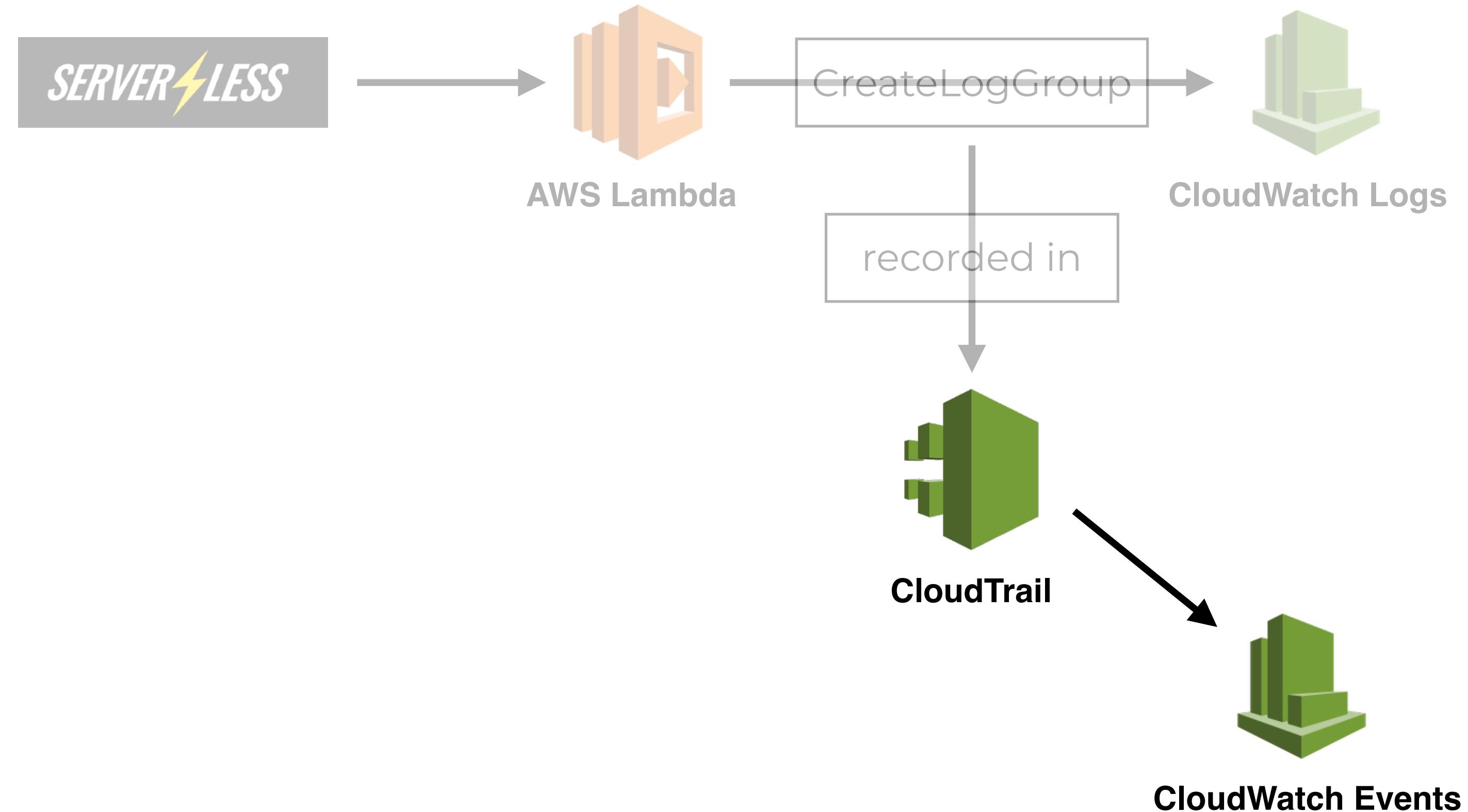
[Next Steps](#)

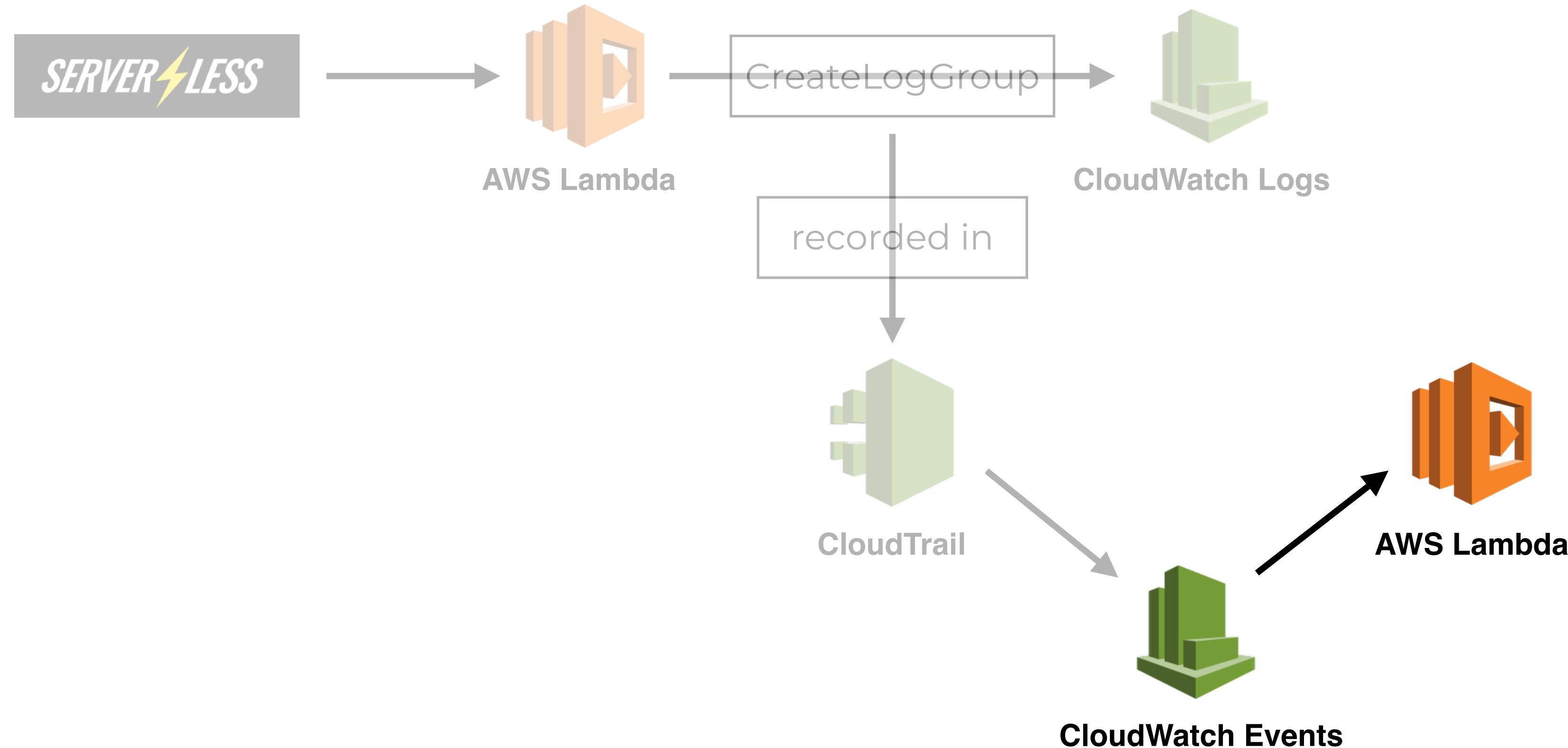
To create a CloudTrail trail with the AWS Management Console

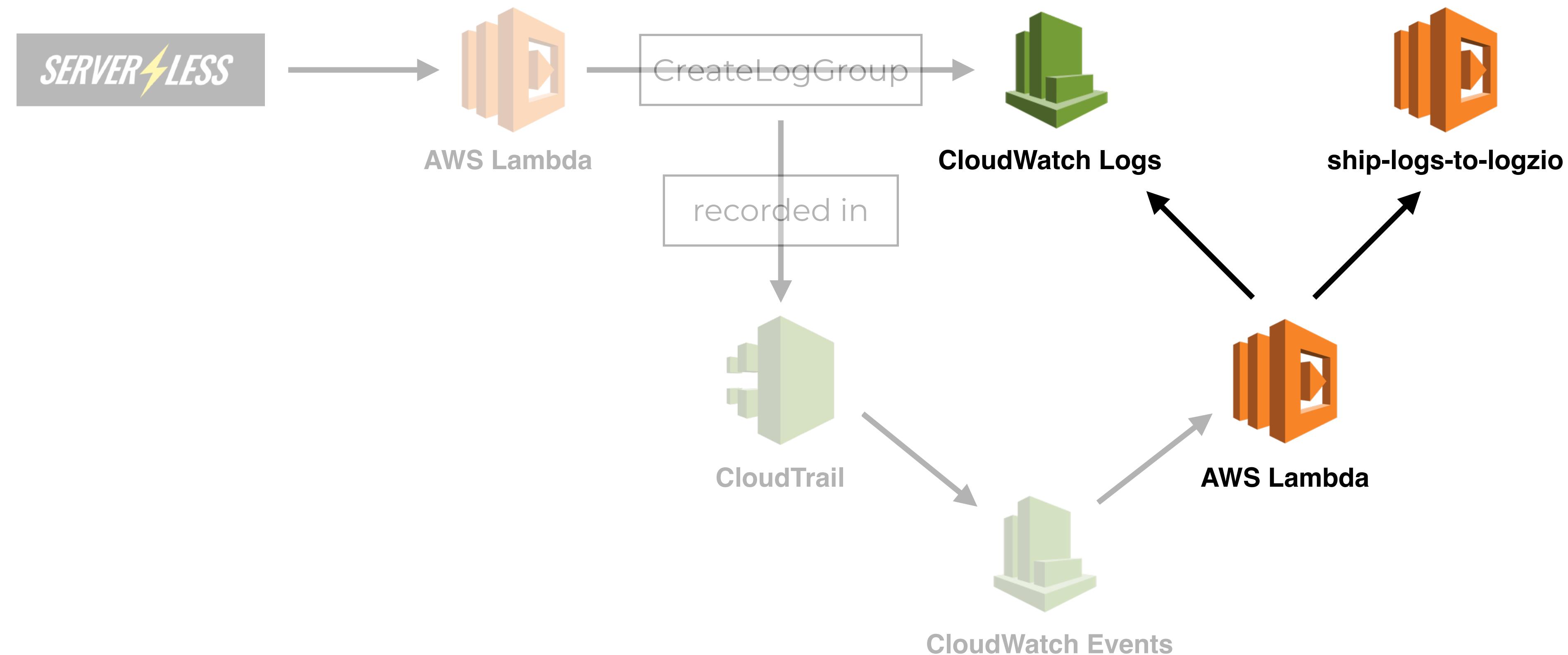
1. Sign in to the AWS Management Console and open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
2. Choose the region where you want the trail to be created.
3. Choose **Get Started Now**.
4. On the **Create Trail** page, for **Trail name**, type a name for your trail. For more information, see [CloudTrail Trail Naming Requirements](#).
5. For **Apply trail to all regions**, choose **Yes** to receive log files from all regions. This is the default and recommended setting. If you choose **No**, the trail logs files only from the region in which you create the trail.
6. For **Management events**, for **Read/Write events**, choose if you want your trail to log **All**, **Read-only**, **Write-only**, or **None**, and then choose **Save**. By default, trails log all management events. For more information, see [Management Events](#).
7. For **Data events**, you can specify logging data events for Amazon S3 buckets, for

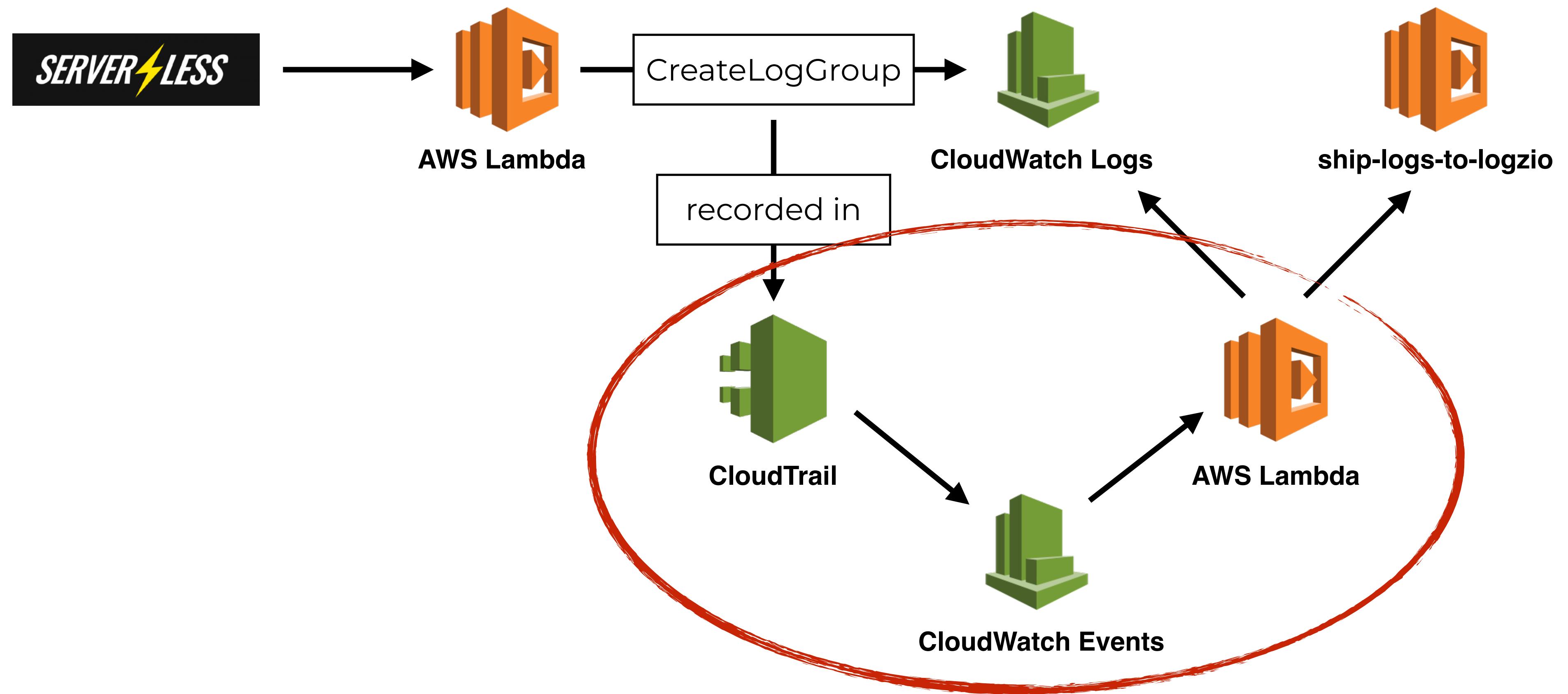
<https://amzn.to/2G5MRPH>









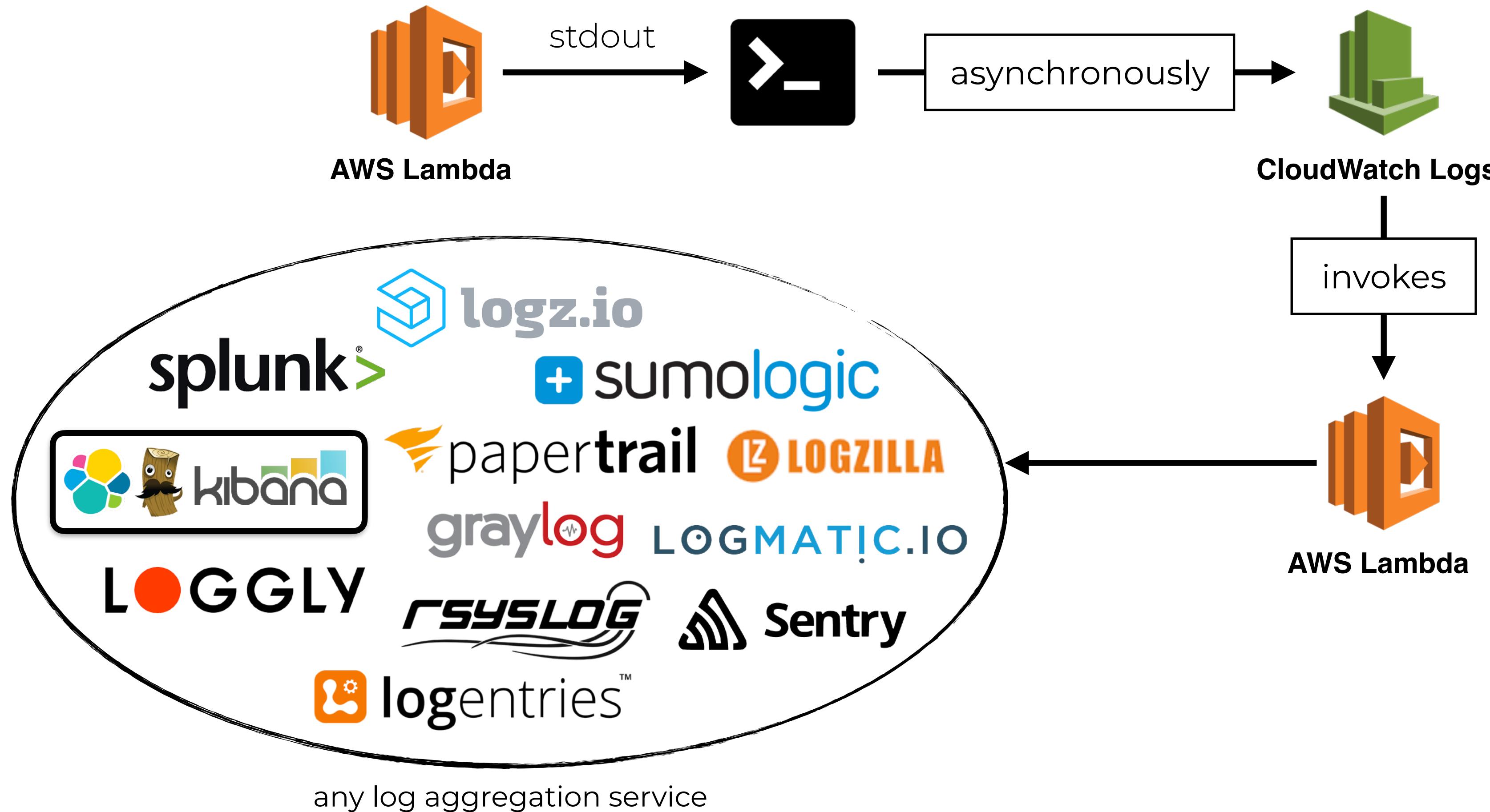


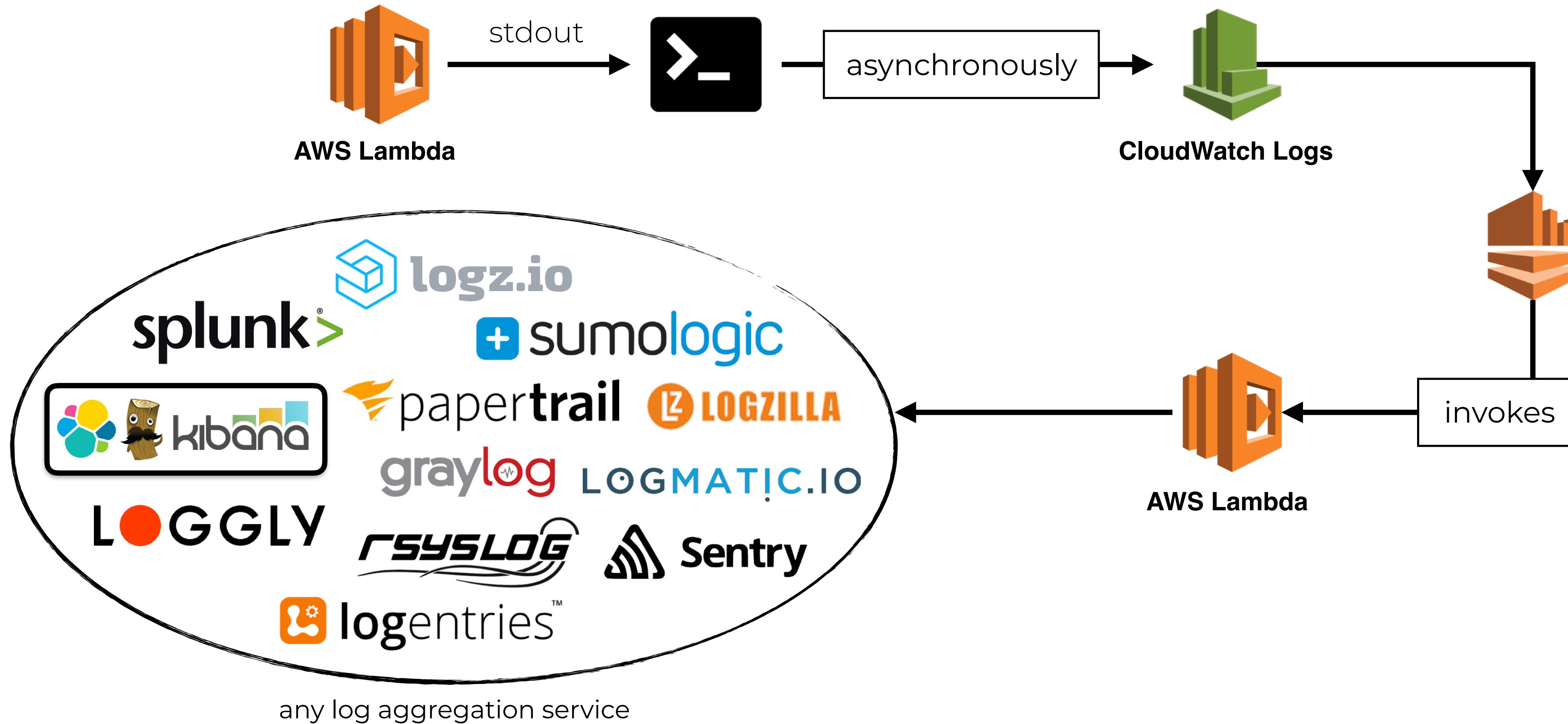
Create Metric Filter Actions ▾

Filter: Log Group Name Prefix x

Log Groups 1-15

Log Groups	Expire Events After	Metric Filters	Subscriptions
/aws/apigateway/welcome	Never Expire	0 filters	None
/aws/codebuild/big-mouth	Never Expire	0 filters	None
/aws/lambda/big-mouth-dev-accept-order	Never Expire	0 filters	None
/aws/lambda/big-mouth-dev-fulfill-order	Never Expire	0 filters	None
/aws/lambda/big-mouth-dev-get-index	Never Expire	0 filters	None
/aws/lambda/big-mouth-dev-get-restaurants	Never Expire	0 filters	None
/aws/lambda/big-mouth-dev-notify-restaurant	Never Expire	0 filters	None
/aws/lambda/big-mouth-dev-notify-user	Never Expire	0 filters	None
/aws/lambda/big-mouth-dev-place-order	Never Expire	0 filters	Lambda (cloudwatch-logs-to-logzio-dev-ship-logs-to-logzio)
/aws/lambda/big-mouth-dev-retry-notify-restaurant	Never Expire	0 filters	None
/aws/lambda/big-mouth-dev-search-restaurants	Never Expire	0 filters	None
/aws/lambda/cloudwatch-logs-to-logzio-dev-set-retention	Never Expire	0 filters	None
/aws/lambda/cloudwatch-logs-to-logzio-dev-ship-logs-to-logzio	Never Expire	0 filters	None
/aws/lambda/cloudwatch-logs-to-logzio-dev-subscribe	Never Expire	0 filters	None
API-Gateway-Execution-Logs_0ny6d5kfjj/test	Never Expire	0 filters	None





auto-subscribe-log-group-to-arn

 Creates custom IAM roles or resource policies

Lumigo

 <https://github.com/lumigo/SAR-Logging>

 3 stars

 Deploy

23 deployments

Subscribes new and existing CloudWatch log groups to Lambda/Kinesis/Firehose by ARN.

[Readme](#)

[License](#)

[Permissions](#)

cloudwatch-logs-auto-subscribe

Subscribes new and existing CloudWatch log groups to Lambda/Kinesis/Firehose by ARN.

Parameters

DestinationArn: The ARN of the Lambda function or Kinesis stream to subscribe a newly created CloudWatch log group to.

Prefix: (Optional) if specified then only log groups with the prefix will be subscribed. E.g. '/aws/lambda/' will subscribe only Lambda function logs

FilterName: (Optional) if specified, will override the filter name for the subscription.

FilterPattern: (Optional) if specified, will override the filter pattern used to create the subscription.

<https://amzn.to/2DnREgn>

auto-set-log-group-retention

 Creates custom IAM roles or resource policies

Lumigo

 <https://github.com/lumigo/SAR-Logging>

 3 stars

 Deploy

5 deployments

Updates the retention policy for new and existing CloudWatch log groups to the specified number of days.

[Readme](#)

[License](#)

[Permissions](#)

cloudwatch-logs-auto-set-retention

Updates the retention policy for new and existing CloudWatch log groups to the specified number of days.

Parameters

RetentionDays: The number of days to retain logs in CloudWatch Logs for.

<https://amzn.to/2uZYmEw>