



# **Amazon Web Services Data Engineering Immersion Day**

---

Lab 1. Real-Time Clickstream Anomaly Detection  
August 2020

## Table of Contents

<i>Introduction .....</i>	<b>2</b>
<i>Get Started Using the Lab Environment.....</i>	<b>3</b>
<i>Set up an Analytics Pipeline Application .....</i>	<b>5</b>
<i>Connect Lambda as destination to Analytics Pipeline .....</i>	<b>11</b>
<i>Appendix: Anomaly Detection Scripts .....</i>	<b>16</b>

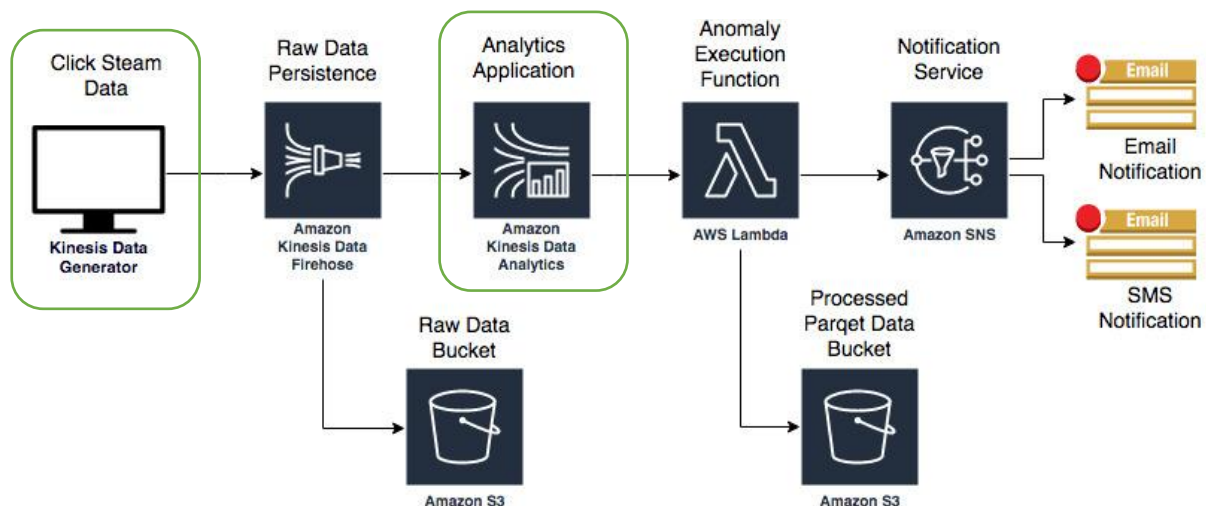
## Lab 1. Real-Time Clickstream Anomaly Detection

### Introduction

This guide helps you complete Real-Time Clickstream Anomaly Detection using Amazon Kinesis Data Analytics.

Analyzing web log traffic to gain insights that drive business decisions has historically been performed using batch processing. Although effective, this approach results in delayed responses to emerging trends and user activities. There are solutions that process data in real time using streaming and micro-batching technologies, but they can be complex to set up and maintain. [Amazon Kinesis Data Analytics](#) is a managed service that makes it easy to identify and respond to changes in data behavior in real-time.

In the prelab, you set up the prerequisites required to complete this lab. Now, you will work to implement the following data pipeline.



Today, you are attending a formal AWS event. If in the future you might want to perform these labs in your own AWS environment by yourself, you can follow instructions here:

<https://aws-dataengineering-day.workshop.aws/300/320-main-lab.html>

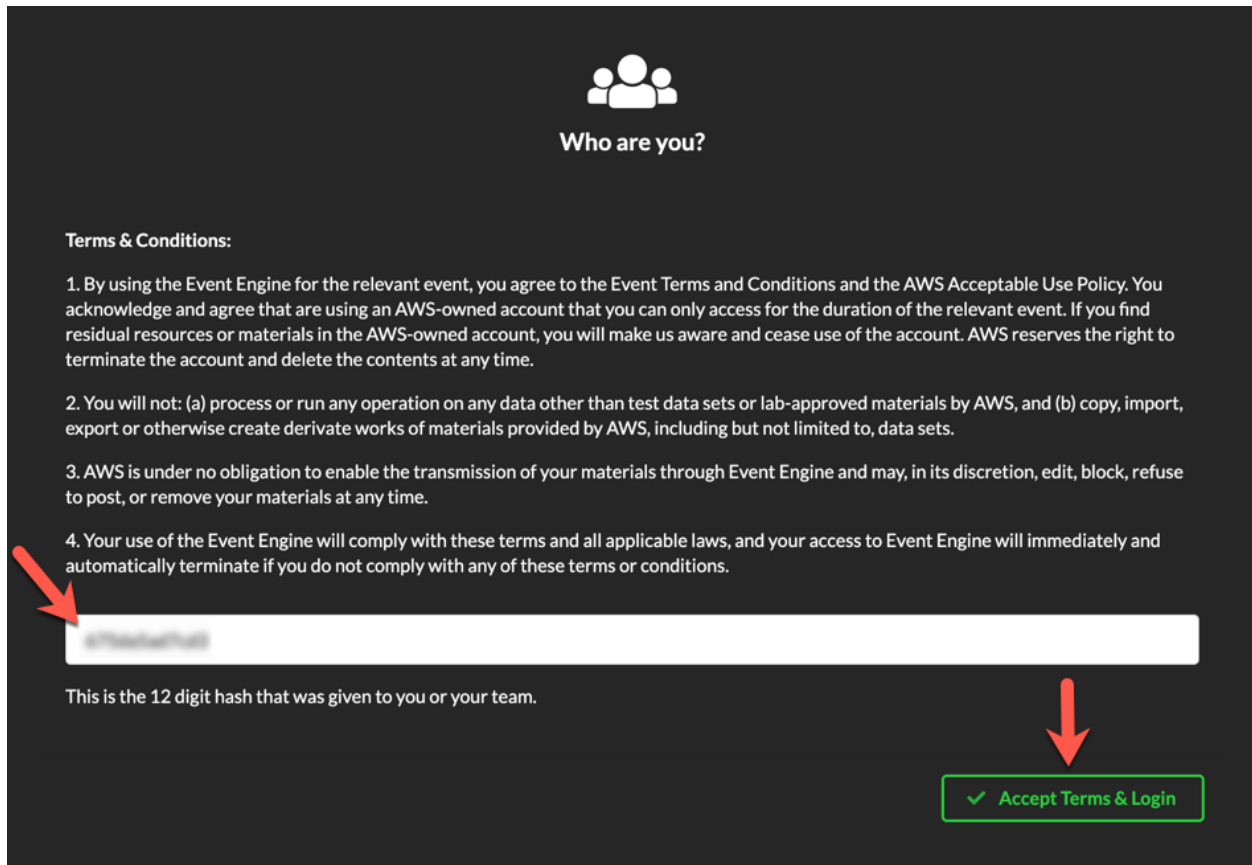
## Get Started Using the Lab Environment

Please skip this section if you are running the lab on your own AWS account.

Today, you are attending a formal event and you will have been sent your access details beforehand. If in the future you might want to perform these labs in your own AWS environment by yourself, you can follow instructions on GitHub - <https://github.com/aws-samples/data-engineering-for-aws-immersion-day>.

A 12-character access code (or 'hash') is the access code that grants you permission to use a dedicated AWS account for the purposes of this workshop.

1. Go to <https://dashboard.eventengine.run/>, enter the access code and click Proceed:



Who are you?

**Terms & Conditions:**

1. By using the Event Engine for the relevant event, you agree to the Event Terms and Conditions and the AWS Acceptable Use Policy. You acknowledge and agree that are using an AWS-owned account that you can only access for the duration of the relevant event. If you find residual resources or materials in the AWS-owned account, you will make us aware and cease use of the account. AWS reserves the right to terminate the account and delete the contents at any time.
2. You will not: (a) process or run any operation on any data other than test data sets or lab-approved materials by AWS, and (b) copy, import, export or otherwise create derivate works of materials provided by AWS, including but not limited to, data sets.
3. AWS is under no obligation to enable the transmission of your materials through Event Engine and may, in its discretion, edit, block, refuse to post, or remove your materials at any time.
4. Your use of the Event Engine will comply with these terms and all applicable laws, and your access to Event Engine will immediately and automatically terminate if you do not comply with any of these terms or conditions.


This is the 12 digit hash that was given to you or your team.


✓ Accept Terms & Login


2. On the Team Dashboard, please click AWS Console to log into the AWS Management Console:

## Lab 1. Real-Time Clickstream Anomaly Detection

## Team Dashboard

Event

AWS Console

SSH Key

Event

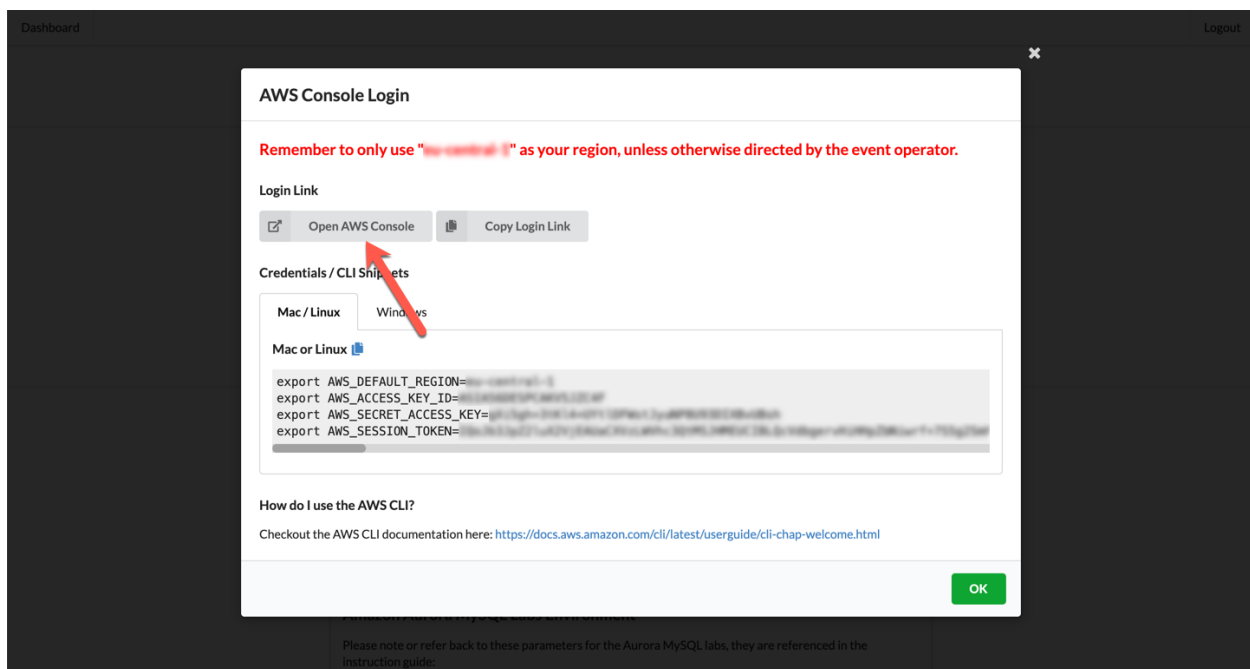
Data Engineering Immersion Day - Test

Team Name:

Event ID: d2302d4ae9ff4ea2857846b74f7de7e2

Team ID: 1c2f7ad7ec044b0b8276f917c5983133

3. Click Open Console. For the purposes of this workshop, you will not need to use command line and API access credentials



Once you have completed these steps, you can continue with the rest of this lab

### Set up an Analytics Pipeline Application

1. Navigate to the **Amazon Kinesis** console by using this link:  
<https://console.aws.amazon.com/kinesis/home?region=us-east-1>
2. Click **Create application** under **Data Analytics**:

**Amazon Kinesis** [Info](#)

Amazon Kinesis makes it easy to collect, process, and analyze data streams in real time, so you can get timely insights and react quickly to new information.

Data Streams <a href="#">Info</a>	Data Firehose <a href="#">Info</a>	Data Analytics <a href="#">Info</a>
Total data streams <b>0</b> <a href="#">Create data stream</a>	Total delivery streams <b>1</b> <a href="#">Create delivery stream</a>	Total analytics applications <b>0</b> <a href="#">Create application</a>

3. On the Create application page, fill the fields as follows:
  - a. For **Application name**, type **anomaly-detection-application**
  - b. Leave "SQL" selected as Default.

**Amazon Kinesis**

- Dashboard
- Data Streams
- Data Firehose
- Data Analytics**
- Video Streams
- External resources
- [What's new](#)

### Kinesis Analytics - Create application

Kinesis Analytics applications continuously read and analyze data from a connected streaming source in real-time. To enable interactivity with your data during configuration you will be prompted to run your application. Kinesis Analytics resources are not covered under the [AWS Free Tier](#), and **usage-based charges apply**. For more information, see [Kinesis Analytics pricing](#).

Application name

Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens, and periods.

Description - optional

Runtime

☒ **SQL**  
Process data in real-time using SQL, which provides an easy way to quickly query large volumes of streaming data without learning new frameworks or languages. [Learn more](#)

☐ **Apache Flink**  
Apache Flink is an open-source framework and distributed processing engine for stateful computations over unbounded and bounded data streams. [Learn more](#)

**After you create the application, you can't change the type or version of the runtime environment.**

[Cancel](#) [Create application](#)

4. Click **Create application**
5. On the application page, click **Connect streaming data**.

## Lab 1. Real-Time Clickstream Anomaly Detection

### anomaly-detection-application

**Description:** This Kinesis Analytics Application is part of the Anomaly Detection Lab

**Application ARN:** arn:aws:kinesisanalytics:us-east-1: :application/anomaly-detection-application

**Application version ID:** 1 ⓘ



#### Source

##### Streaming data

Connect to an existing Kinesis stream or Firehose delivery stream, or easily create and connect to a new demo Kinesis stream. Each application can connect to one streaming data source. [Learn more](#)

[Connect streaming data](#)

6. Select **Choose source**, and make the following selections:
  - a. For **Source**, choose **Kinesis Firehose delivery stream**.
  - b. For **Kinesis Firehose delivery stream**, select **FirehoseDeliveryStream-`<random string>`**

Kinesis Analytics applications > anomaly-detection-application > Streaming data ⓘ

### Connect streaming data source

Choose from your Kinesis data streams and Firehose delivery streams, or quickly configure a demo Kinesis stream that can be used to explore Kinesis Analytics.

☒ Choose source

☐ Configure a new stream

**Source**

☐ Kinesis data stream  
Kinesis data stream is an ordered sequence of data records used for rapid and continuous data intake and aggregation.

☒ Kinesis Firehose delivery stream  
Kinesis Firehose delivery streams send source records to the destinations that you specify, automatically and continuously.

Kinesis Firehose delivery stream

kinesis-pre-lab-FirehoseDeliveryStream-1VY1NUI950NAA

[View kinesis-pre-lab-FirehoseDeliveryStream-1VY1NUI950NAA in Kinesis Firehose](#)

In-application stream name

In your SQL queries, refer to this source as: **SOURCE\_SQL\_STREAM\_001**

7. In the **Record pre-processing with AWS Lambda** section, choose **Disabled**.
8. In the **Access to chosen resources** section, select **Choose from IAM roles that Kinesis Analytics can assume**.
9. In the **IAM role** box, search for the following role:  
kinesis-pre-lab-**CSEKinesisAnalyticsRole-`<random string>`**

## Lab 1. Real-Time Clickstream Anomaly Detection

### Record pre-processing with AWS Lambda

Kinesis Analytics can invoke your Lambda function to pre-process records before they are used in this application. To pre-process records, your Lambda function must be compliant with the required record transformation output model. [Learn more](#)

Record pre-processing

- ☒ Disabled  
☐ Enabled

### Access permissions

Create or choose IAM role with the required permissions. [Learn more](#)

Access permissions

- ☐ Create / update IAM role **kinesis-analytics-anomaly-detection-application-us-east-1**  
☒ Choose from IAM roles that Kinesis Analytics can assume

IAM role

Only IAM roles with the [required trust policy](#) attached are available for selection.

kinesis-pre-lab-CSEKinesisAnalyticsRole-7EDWTWRWK4X



[View kinesis-pre-lab-CSEKinesisAnalyticsRole-7EDWTWRWK4X in IAM](#)

### Schema

Schema discovery can generate a schema using recent records from the source. Schema column names are the same as in the source, unless they contain special characters, repeated column names, or reserved keywords. [Learn more](#)

Discover schema

Do not click **“Discover schema”** yet.

You have set up the Kinesis Data Analytics application to receive data from a Kinesis Data Firehose and to use an IAM role from the pre-lab. However, you need to start sending some data to the Kinesis Data Firehose before you click **Discover schema** in your application.

Navigate to the Amazon Kinesis Data Generator (Amazon KDG) which you setup in prelab and start sending the **Schema Discovery Payload** at **1 record per second** by clicking on Send data button. Make sure to select the region **“us-east-1”**



## Lab 1. Real-Time Clickstream Anomaly Detection

Amazon Kinesis Data Generator

Configure

Region

us-east-1

Stream/delivery stream

kinesis-pre-lab-FirehoseDeliveryStream-1VY1NUI950I

Records per second

Constant

Periodic

1

Compress Records

☐

Record template

Schema Discovery Payload

Click Payload

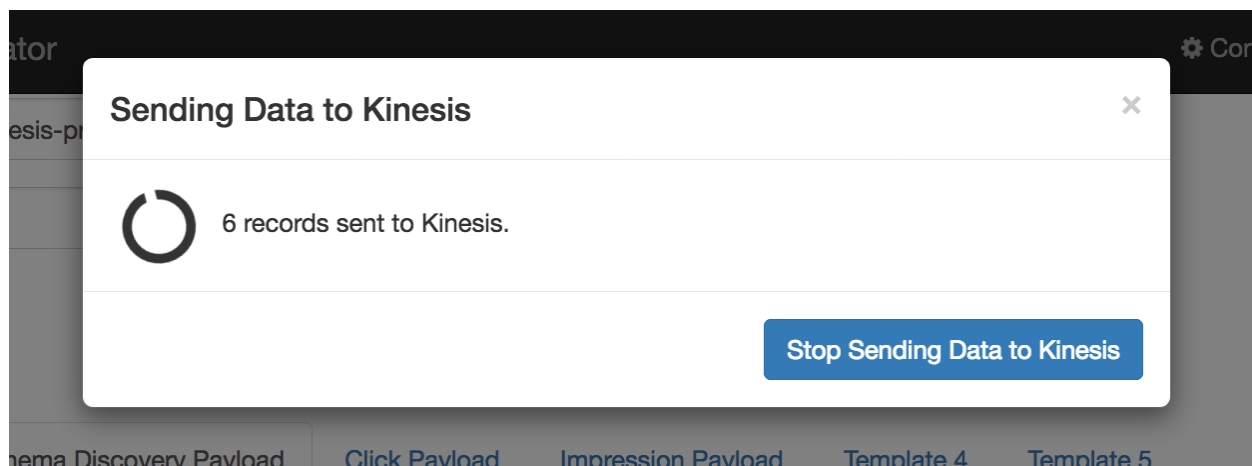
Impression Payload

Template 4

Template 5

Schema Discovery Payload

```
{"browseraction": "DiscoveryKinesisTest", "site": "yourwebsiteurl.domain.com"}
```



Now that your Kinesis Data Firehose is receiving data, you can continue configuring the Kinesis Data Analytics Application.

10. Go back to the AWS console, Now click **Discover Schema**.

## Lab 1. Real-Time Clickstream Anomaly Detection

✔

**Schema discovery successful**  
Detected JSON format and applied schema

- To define a custom schema, choose "Edit schema" in the stream sample below.
- To capture a new stream sample from the selected source for discovery, choose **Retry schema discovery** below.

(Optional) Send AWS a sample of your data to help improve schema discovery in Amazon Kinesis Analytics  
[Help improve schema discovery](#)

✕

Edit schema

Retry schema discovery

Raw

Lambda output

Formatted

Filter by column name or column type

browseraction VARCHAR(32)	site VARCHAR(32)
DiscoveryKinesisTest	yourwebsiteurl.domain.com
DiscoveryKinesisTest	yourwebsiteurl.domain.com
DiscoveryKinesisTest	yourwebsiteurl.domain.com
DiscoveryKinesisTest	yourwebsiteurl.domain.com
DiscoveryKinesisTest	yourwebsiteurl.domain.com
DiscoveryKinesisTest	yourwebsiteurl.domain.com
DiscoveryKinesisTest	yourwebsiteurl.domain.com
DiscoveryKinesisTest	yourwebsiteurl.domain.com
DiscoveryKinesisTest	yourwebsiteurl.domain.com
DiscoveryKinesisTest	yourwebsiteurl.domain.com

Cancel

Save and continue

11. Click **Save and continue**. Your Kinesis Data Analytics Application is created with an input stream.

### Source

#### Streaming data

Connect to an existing Kinesis stream or Firehose delivery stream, or easily create and connect to a new demo Kinesis stream. Each application can connect to one streaming data source. [Learn more](#)

	Source	In-application stream name	ID ⓘ	Record pre-processing ⓘ
	Firehose delivery stream <a href="#">kinesis-pre-lab-FirehoseDeliveryStream-1269IUN45DAIY</a>	SOURCE_SQL_STREAM_001	2.1	Disabled

Now, you can add some SQL queries to easily analyze the data that is being fed into the stream.

12. In the **Real time analytics** section, click **Go to SQL editor**.



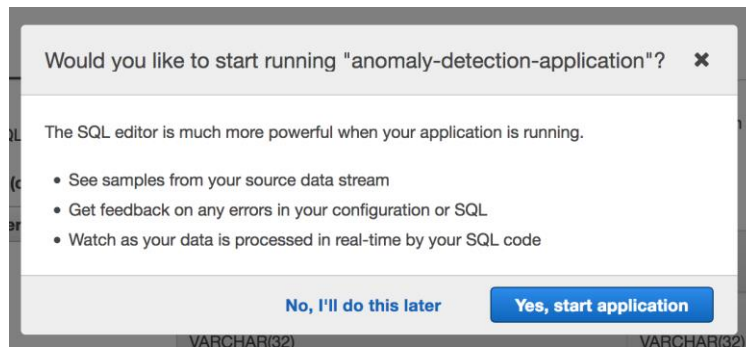
### Real time analytics

Author your own SQL queries or add SQL from templates to easily analyze your source data. [Learn more](#)

[Go to SQL editor](#)

## Lab 1. Real-Time Clickstream Anomaly Detection

13. Click on **"Yes, start application"** to start your kinesis analytics application.



14. Click on this [link](#), grab the SQL script and paste it into the SQL editor. (You can also find the code in Appendix)



### Real time analytics

Author your own SQL queries or add SQL from templates to easily analyze your source data. [Learn more](#)

[Go to SQL editor](#)

### Real-time analytics



15. Click **Save and run SQL**. The analytics application starts and runs your SQL query. (You can find the SQL query in [Appendix A](#).)

To learn more about the SQL logic, see the **Analytics application** section in the following blog post:

<https://aws.amazon.com/blogs/big-data/real-time-clickstream-anomaly-detection-with-amazon-kinesis-analytics/>

16. On the **Source data** tab, observe the input stream data named "SOURCE\_SQL\_STREAM\_001".

## Lab 1. Real-Time Clickstream Anomaly Detection

**Source** | Real-time analytics | Destination

**Streaming data**  
☒ SOURCE\_SQL\_STREAM\_001

The streaming data below is a sample from Kinesis Firehose delivery stream [kinesis-pre-lab-FirehoseDeliveryStream-1VY1NUI950NAA](#) [↗](#)

**Reference data (optional)** ⓘ  
Connect reference data

**Actions** ▼

ROWTIME TIMESTAMP	browseraction VARCHAR(32)	site VARCHAR(32)	APPROXIMATE_ARRIVAL_TIME TIMESTAMP
2020-02-07 01:34:37.005	DiscoveryKinesisTest	yourwebsiteurl.domain.com	2020-02-07 01:34:36.243
2020-02-07 01:34:38.025	DiscoveryKinesisTest	yourwebsiteurl.domain.com	2020-02-07 01:34:37.248
2020-02-07 01:34:38.989	DiscoveryKinesisTest	yourwebsiteurl.domain.com	2020-02-07 01:34:38.231
2020-02-07 01:34:39.991	DiscoveryKinesisTest	yourwebsiteurl.domain.com	2020-02-07 01:34:39.325
2020-02-07 01:34:41.017	DiscoveryKinesisTest	yourwebsiteurl.domain.com	2020-02-07 01:34:40.187
2020-02-07 01:34:42.021	DiscoveryKinesisTest	yourwebsiteurl.domain.com	2020-02-07 01:34:41.212
2020-02-07 01:34:43.026	DiscoveryKinesisTest	yourwebsiteurl.domain.com	2020-02-07 01:34:42.205
2020-02-07 01:34:44.029	DiscoveryKinesisTest	yourwebsiteurl.domain.com	2020-02-07 01:34:43.255

If you click the **Real-time analytics** tab, you will notice multiple in-application streams. You will populate data in these streams later in the lab.

Application status: RUNNING

Source | **Real-time analytics** | Destination

**In-application streams:**  
☒ CLICKSTREAM  
☐ CTRSTREAM  
☐ DESTINATION\_SQL\_STREAM  
☐ IMPRESSIONSTREAM  
☐ error\_stream

**Pause results** ⓘ New results are added every 2-10 seconds. The results below are sampled. ⓘ  
☐ Scroll to bottom when new results arrive.

ROWTIME	CLICKCOUNT
---------	------------

ⓘ No rows have arrived yet.

## Connect Lambda as destination to Analytics Pipeline

Now that the logic to detect anomalies is in the Kinesis Data Firehose, you can connect it to a destination (AWS Lambda function) to notify you when there is an anomaly.

1. Click the **Destination** tab and click **Connect to a Destination**.
2. For **Destination**, choose **AWS Lambda function**.

## Lab 1. Real-Time Clickstream Anomaly Detection

### Connect to destination

#### Destination

- ☐ **Kinesis data stream**  
Kinesis data stream is an ordered sequence of data records used for rapid and continuous data intake and aggregation.
- ☐ **Kinesis Firehose delivery stream**  
Kinesis Firehose delivery streams send source records to the destinations that you specify, automatically and continuously.
- ☒ **AWS Lambda function**  
AWS Lambda is a compute service that lets you run code without provisioning or managing servers.

#### Deliver records to AWS Lambda

To deliver Kinesis Analytics output records, your Lambda function must be compliant with the required request/response model. [Learn more](#)

#### Lambda function

CSEBeconAnomalyResponse



Create new

[View CSEBeconAnomalyResponse in Lambda](#)

#### Lambda function version

\$LATEST



#### Description

Click Stream Example Lambda Function

#### Runtime

nodejs12.x



#### Increase Lambda function timeout

To reduce the risk of the function timing out, increase the **Timeout** to 1 minute or longer in the **Advanced settings** section of your Lambda configuration.

[Go to Lambda configuration](#)

#### Timeout

5 seconds

3. In the Deliver records to AWS Lambda section, make the following selections:
  - a. For **Lambda function**, choose **CSEBeconAnomalyResponse**.
  - b. For **Lambda function version**, choose **\$LATEST**.
4. In the **In-application stream** section, make the following selections:
  - a. Select **Choose an existing in-application stream**.
  - b. For **In-application stream name**, choose **DESTINATION\_SQL\_STREAM**.
  - c. For **Output format**, choose: **JSON**.
5. In the Access to chosen resources section, make the following selections:
  - a. Select **Choose from IAM roles that Kinesis Analytics can assume**.
  - b. For **IAM role**, choose **kinesis-pre-lab-CSEKinesisAnalyticsRole-  
<random string>**.

## Lab 1. Real-Time Clickstream Anomaly Detection

Your parameters should look like the following image. This configuration allows your Kinesis Data Analytics Application to invoke your anomaly Lambda function and notify you when any anomalies are detected.

### In-application stream

In-application streams are continuous flows of data records. You create in-application streams in SQL to contain the data you want to persist to the specified destination.

[Learn more](#)

Connect in-application stream

☒ Choose an existing in-application stream

☐ Specify a new in-application stream name

Use this option for in-application streams that you haven't created yet, but plan to create at a later time. Specifying a stream name ensures that you don't lose output data.

In-application stream name

DESTINATION\_SQL\_STREAM

Output format

☒ JSON

☐ CSV

### Access permissions

Create or choose IAM role with the required permissions. [Learn more](#)

Access permissions

☐ Create / update IAM role **kinesis-analytics-anomaly-detection-application-us-east-1**

☒ Choose from IAM roles that Kinesis Analytics can assume

IAM role

Only IAM roles with the [required trust policy](#) attached are available for selection.

kinesis-pre-lab-CSEKinesisAnalyticsRole-7EDWTWRWK4X

[View kinesis-pre-lab-CSEKinesisAnalyticsRole-7EDWTWRWK4X in IAM](#)

[Cancel](#)

[Save and continue](#)

Now that all of the components are in place, you can test your analytics application. For this part of the lab, you will need to use your Kinesis Data Generator in five separate browser windows. There will be one window sending normal impression payload, one window sending normal click payload, and three windows sending extra click payload.

1. Open your KDG in five separate browser windows and sign in as the same user.

**Note:** Make sure to select the **us-east-1** region. Do not accept the default region.

2. In one of your browser windows, start sending the **Impression payload** at a rate of 1 record per second (**keep this running**).

## Lab 1. Real-Time Clickstream Anomaly Detection

- On another browser window, start sending the **Click payload** at a rate of 1 record per second (**keep this running**).
- On your last three browser windows, start sending the **Click payload** at a rate of 1 record per second for a period of about **20 seconds** before stopping them.  
**\*\*If you did not receive an anomaly email, open another KDG window and send additional concurrent Click payloads.** Make sure to not allow these functions to run for more than 10 to 20 seconds at a time. This could cause AWS Lambda to send you multiple emails and SMS messages due to the number of anomalies you are creating.

You can monitor anomalies on the **Real-time analytics** tab in the **DESTINATION\_SQL\_STREAM** table. If an anomaly is detected, it displays in that table.

### Real-time analytics

[Save and run SQL](#) [Add SQL from templates](#) [Download SQL](#) [SQL reference guide](#) [Kinesis data generator tool](#)

```
1 CREATE OR REPLACE STREAM "CLICKSTREAM" (  
2   "CLICKCOUNT" DOUBLE  
3 );  
4  
5 CREATE OR REPLACE PUMP "CLICKPUMP" AS  
6 INSERT INTO "CLICKSTREAM" ("CLICKCOUNT")  
7 SELECT STREAM COUNT(*)  
8 FROM "SOURCE_SQL_STREAM_001"  
9 WHERE "browseraction" = 'Click'  
10 GROUP BY FLOOR(  
11   ("SOURCE_SQL_STREAM_001".ROWTIME - TIMESTAMP '1970-01-01 00:00:00')  
12   SECOND / 10 TO SECOND
```

**Source data** **Real-time analytics** **Destination**

Application status: RUNNING

In-application streams:  

CLICKSTREAM

CTRSTREAM

**DESTINATION\_SQL\_STREAM**

IMPRESSIONSTREAM

error\_stream

[Pause results](#) [New results are added every 2-10 seconds. The results below are sampled.](#)

☐ Scroll to bottom when new results arrive.

Filter by column name

ROWTIME	CTRPERCENT	ANOMALY_SCORE
2018-09-11 19:58:10.0	366.66666666666663	2.0920703952669824

[Close](#)

Make sure to click other streams and review the data.

Once an anomaly has been detected in your application and you will receive an email and text message to the specified accounts.

### Email Snapshot:





## Appendix: Anomaly Detection Scripts

```
CREATE OR REPLACE STREAM "CLICKSTREAM" (  
    "CLICKCOUNT" DOUBLE  
);
```

```
CREATE OR REPLACE PUMP "CLICKPUMP" AS  
INSERT INTO "CLICKSTREAM" ("CLICKCOUNT")  
SELECT STREAM COUNT(*)  
FROM "SOURCE_SQL_STREAM_001"  
WHERE "browseraction" = 'Click'  
GROUP BY FLOOR(  
    ("SOURCE_SQL_STREAM_001".ROWTIME - TIMESTAMP '1970-01-01 00:00:00')  
    SECOND / 10 TO SECOND  
);
```

```
CREATE OR REPLACE STREAM "IMPRESSIONSTREAM" (  
    "IMPRESSIONCOUNT" DOUBLE  
);
```

```
CREATE OR REPLACE PUMP "IMPRESSIONPUMP" AS  
INSERT INTO "IMPRESSIONSTREAM" ("IMPRESSIONCOUNT")  
SELECT STREAM COUNT(*)  
FROM "SOURCE_SQL_STREAM_001"  
WHERE "browseraction" = 'Impression'  
GROUP BY FLOOR(  
    ("SOURCE_SQL_STREAM_001".ROWTIME - TIMESTAMP '1970-01-01 00:00:00')  
    SECOND / 10 TO SECOND  
);
```

```
CREATE OR REPLACE STREAM "CTRSTREAM" (  
    "CTR" DOUBLE  
);
```

```
CREATE OR REPLACE PUMP "CTRPUMP" AS  
INSERT INTO "CTRSTREAM" ("CTR")  
SELECT STREAM "CLICKCOUNT" / "IMPRESSIONCOUNT" * 100.000 as "CTR"  
FROM "IMPRESSIONSTREAM",  
    "CLICKSTREAM"  
WHERE "IMPRESSIONSTREAM".ROWTIME = "CLICKSTREAM".ROWTIME;
```

```
CREATE OR REPLACE STREAM "DESTINATION_SQL_STREAM" (  
    "CTRPERCENT" DOUBLE,  
    "ANOMALY_SCORE" DOUBLE  
);
```

## Lab 1. Real-Time Clickstream Anomaly Detection

```
CREATE OR REPLACE PUMP "OUTPUT_PUMP" AS
INSERT INTO "DESTINATION_SQL_STREAM"
SELECT STREAM * FROM
TABLE (RANDOM_CUT_FOREST(
    CURSOR(SELECT STREAM "CTR" FROM "CTRSTREAM"), --
    inputStream
    100, --numberOfTrees (default)
    12, --subSampleSize
    100000, --timeDecay (default)
    1) --shingleSize (default)
)
WHERE ANOMALY_SCORE > 2;
```