Amazon Web Services
Data Engineering Immersion Day

Lab 1. Real-Time Clickstream Anomaly Detection
July 2021

Table of Contents

Introduction 2

Get Started Using the Lab Environment 3

Set up an Analytics Pipeline Application 5

Connect Lambda as destination to Analytics Pipeline 11

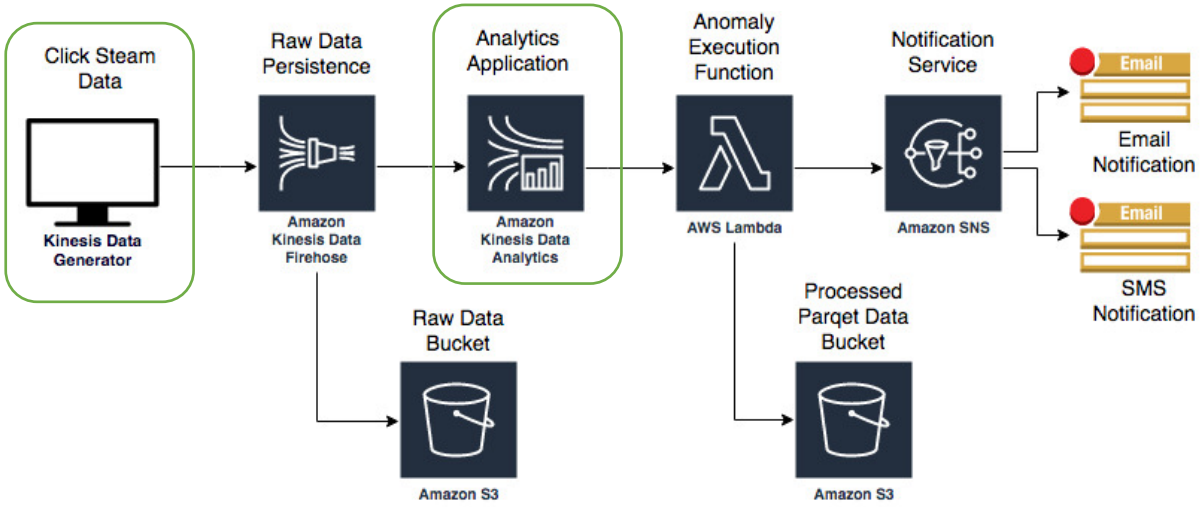
Appendix: Anomaly Detection Scripts 16

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

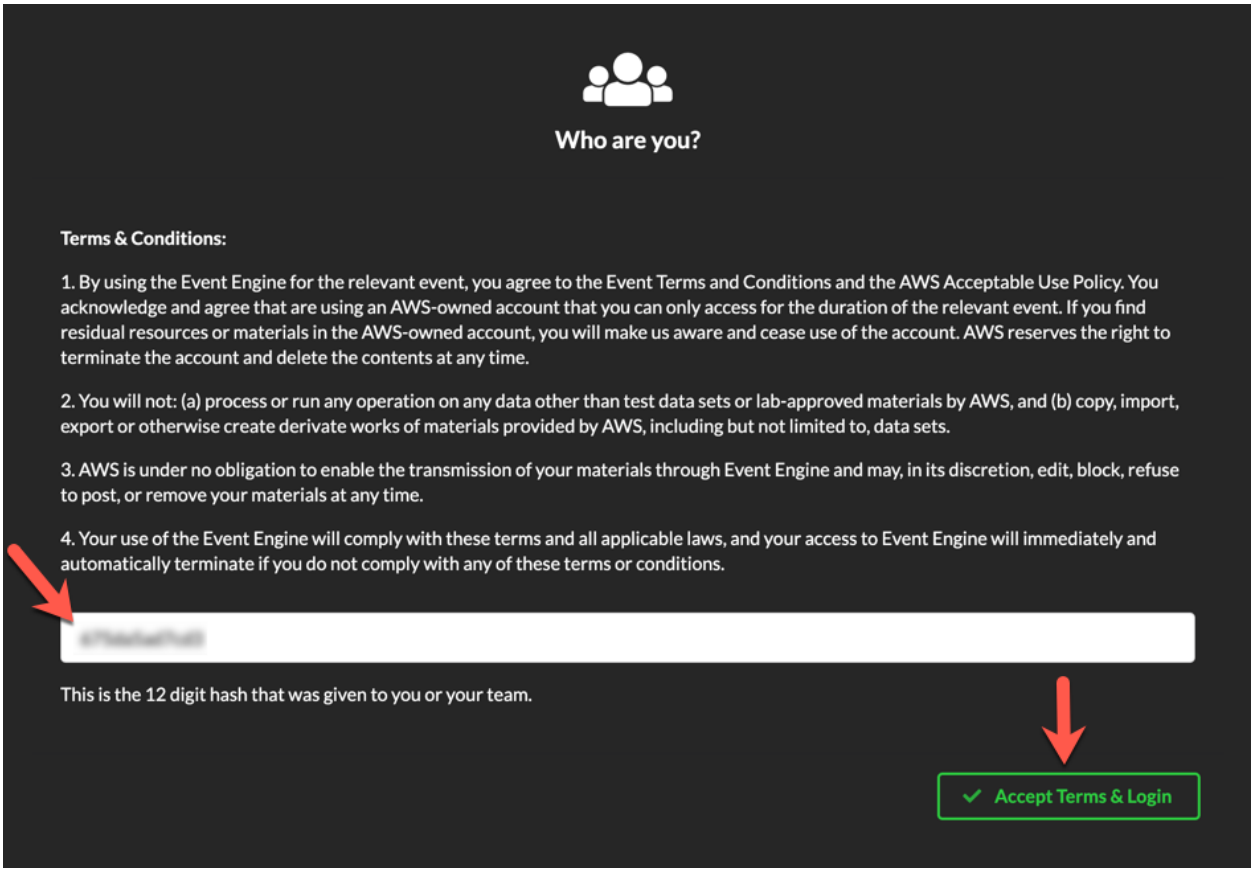
Get Started Using the Lab Environment

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

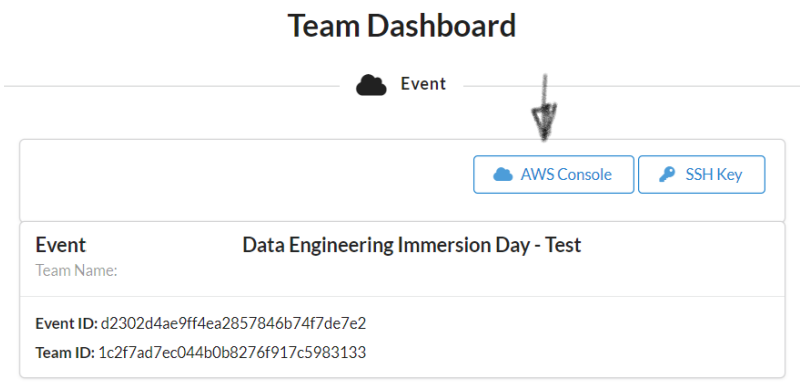
Today, you are attending a formal event and you will have been sent your access details beforehand. If in the future you might want to perform these labs in your own AWS environment by yourself, you can follow instructions on GitHub - https://github.com/aws-samples/data-engineering-for-aws-immersion-day.

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.

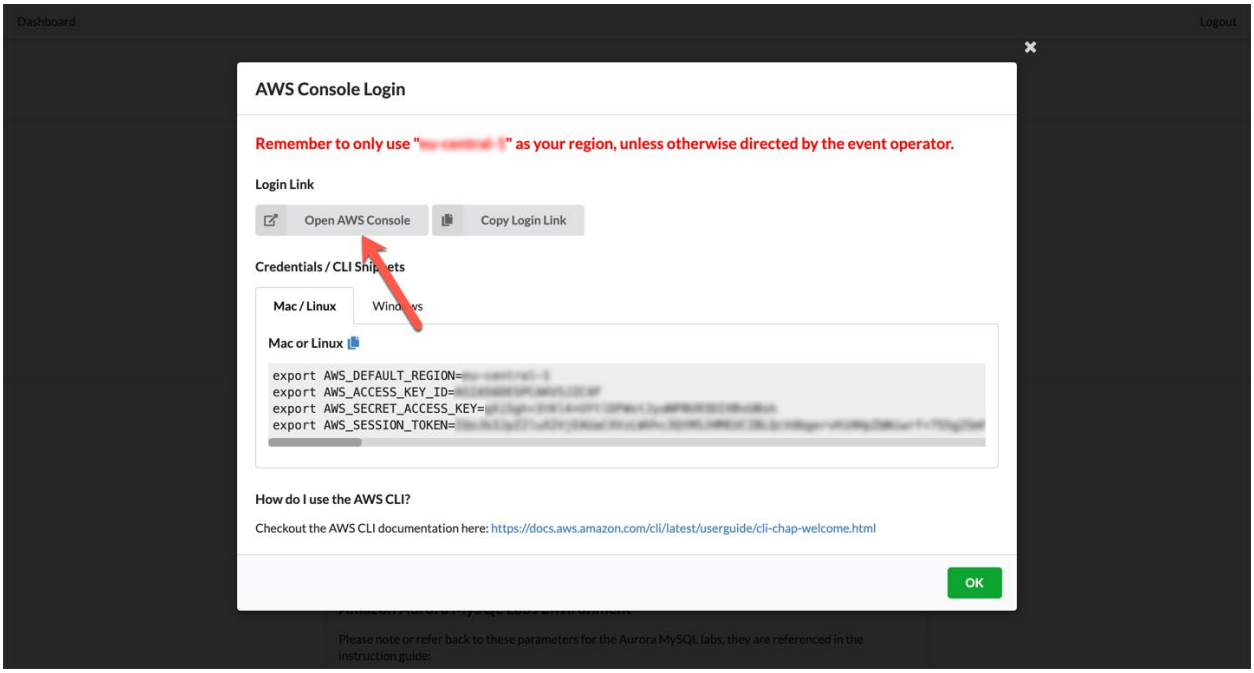
1. Go to https://dashboard.eventengine.run/, enter the access code and click Proceed:



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



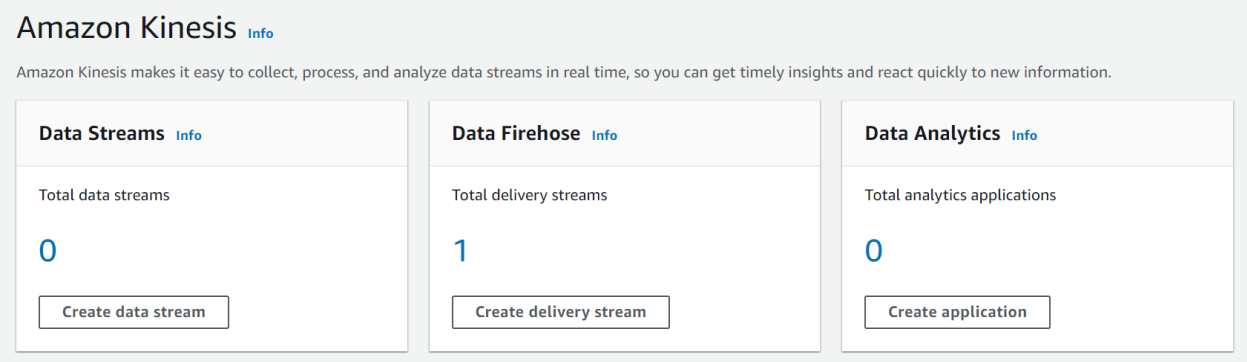
3. Click Open Console. For the purposes of this workshop, you will not need to use command line and API access credentials



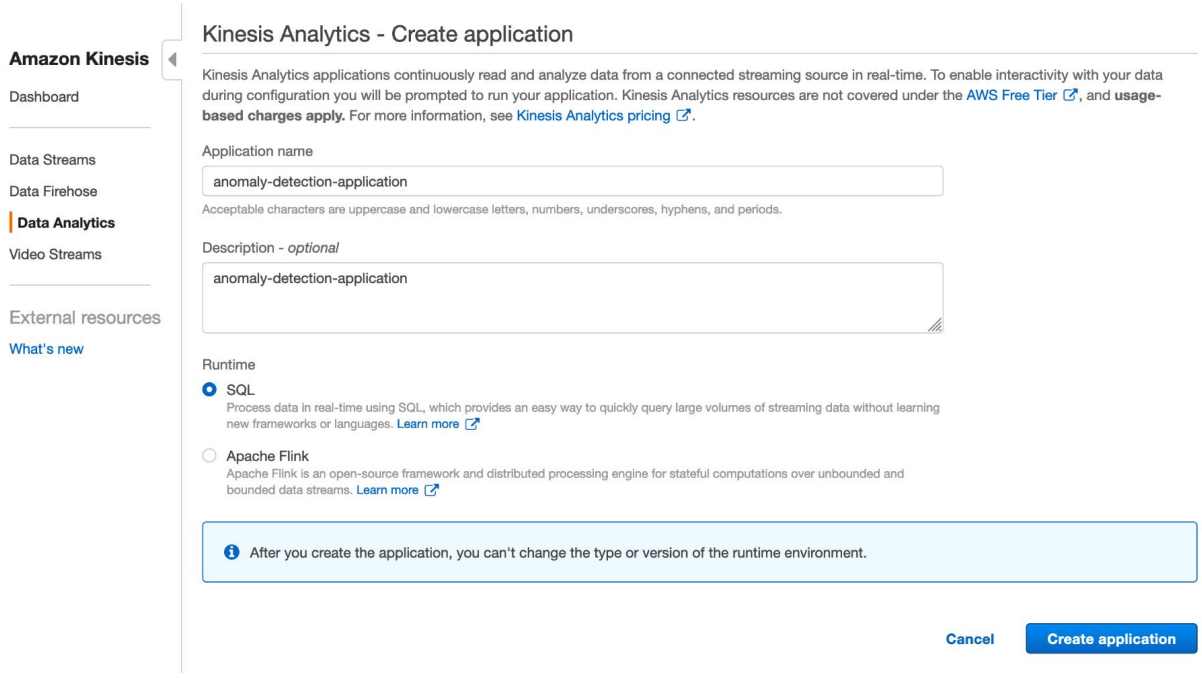
Once you have completed these steps, you can continue with the rest of this lab

Set up an Analytics Pipeline Application

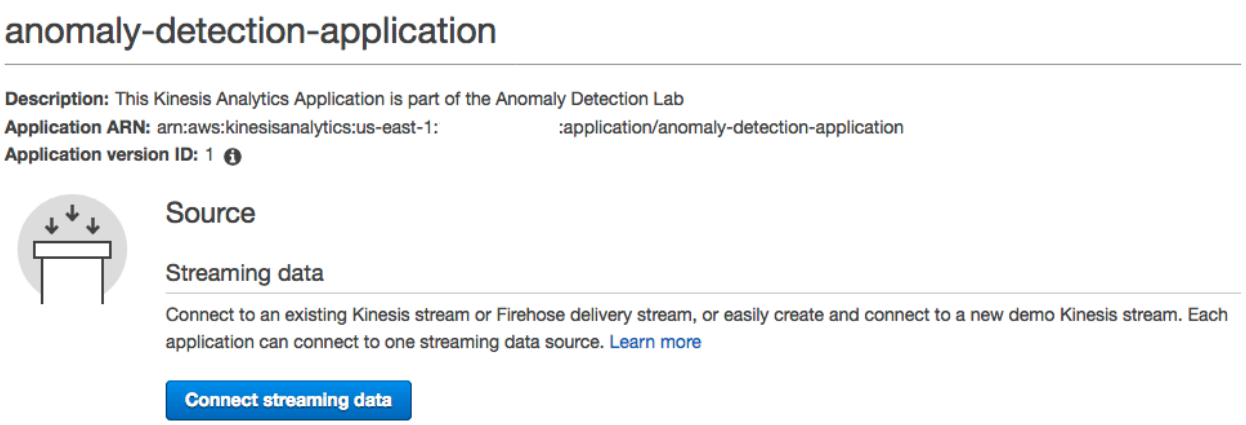
1. Navigate to the **Amazon Kinesis** console by using this link (select appropriate AWS region)**:** https://console.aws.amazon.com/kinesis/home
2. Click **Create application** under **Data Analytics**:



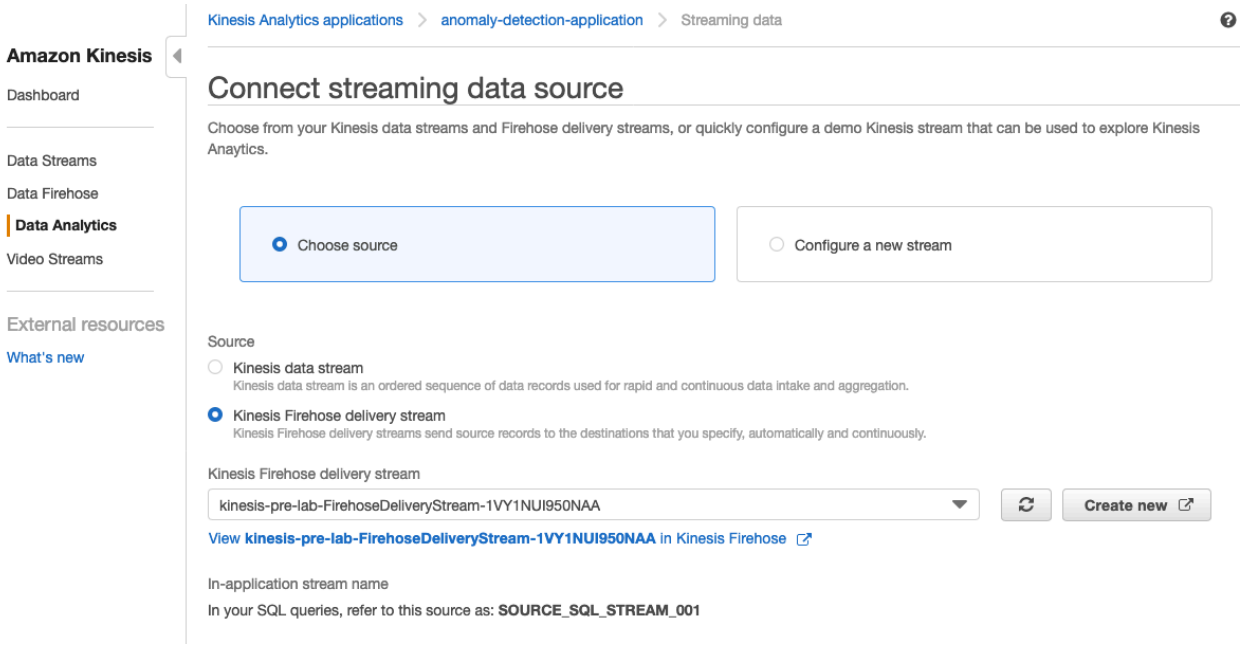
3. On the Create application page, fill the fields as follows:
   a. For **Application name**, type **anomaly-detection-application**
   b. Leave "SQL" selected as Default.



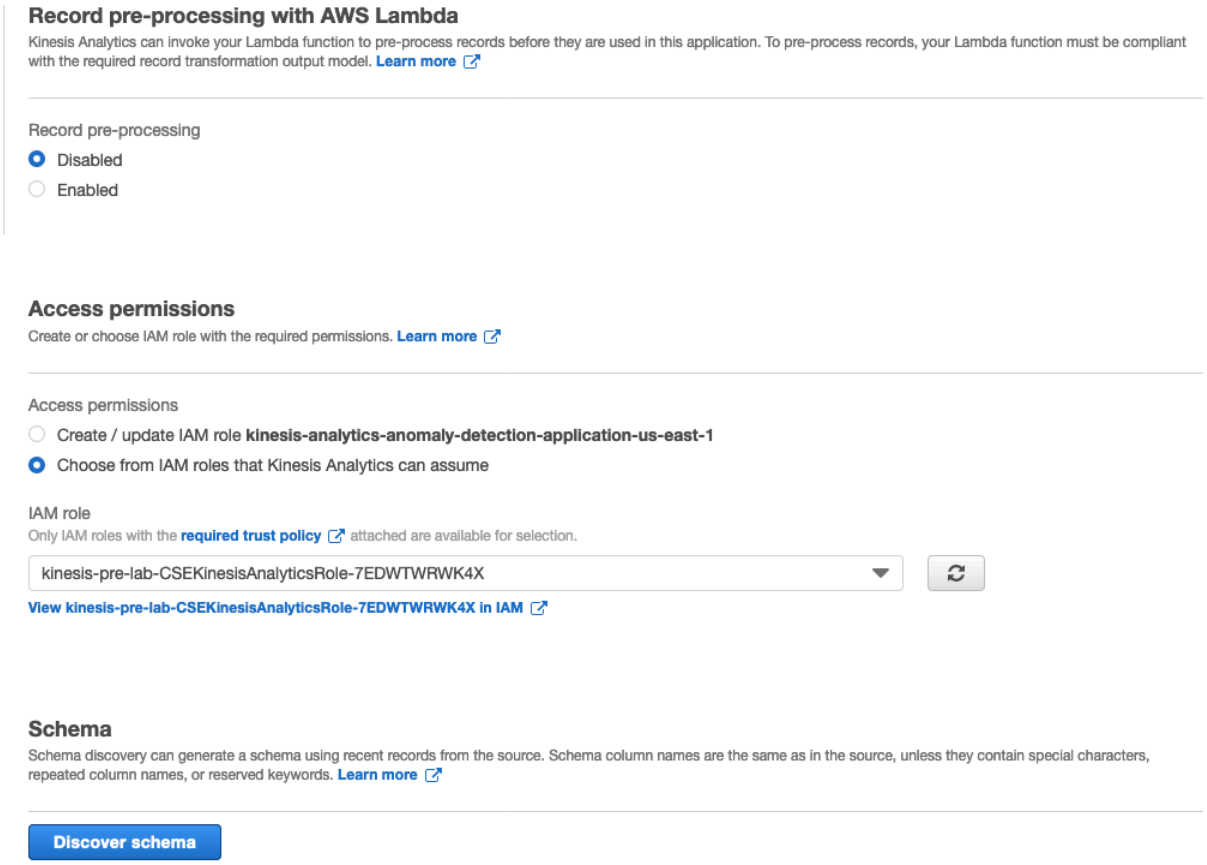
4. Click **Create application**
5. On the application page, click **Connect streaming data**.



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>



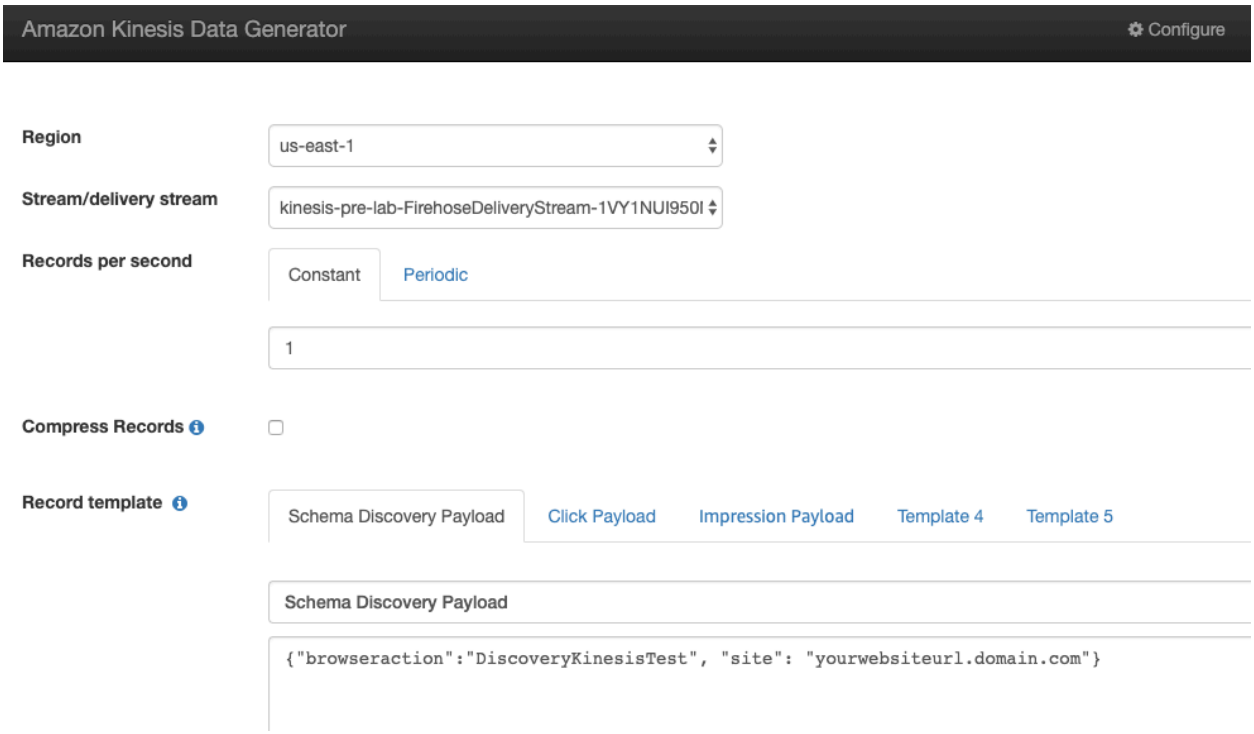
7. In the **Record pre-processing with AWS Lambda** section, choose **Disabled**.
8. In the **Access to chosen resources** section, select **Choose from IAM roles that Kinesis Analytics can assume.**
9. In the **IAM role** box, search for the following role: kinesis-pre-lab-**CSEKinesisAnalyticsRole**-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

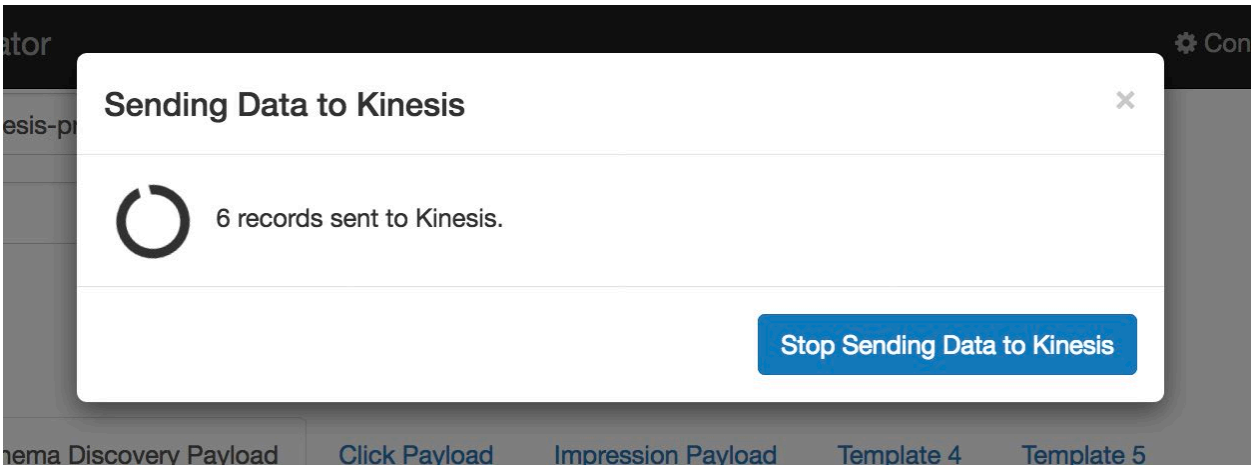


Do not click "**Discover schema" yet.**

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.

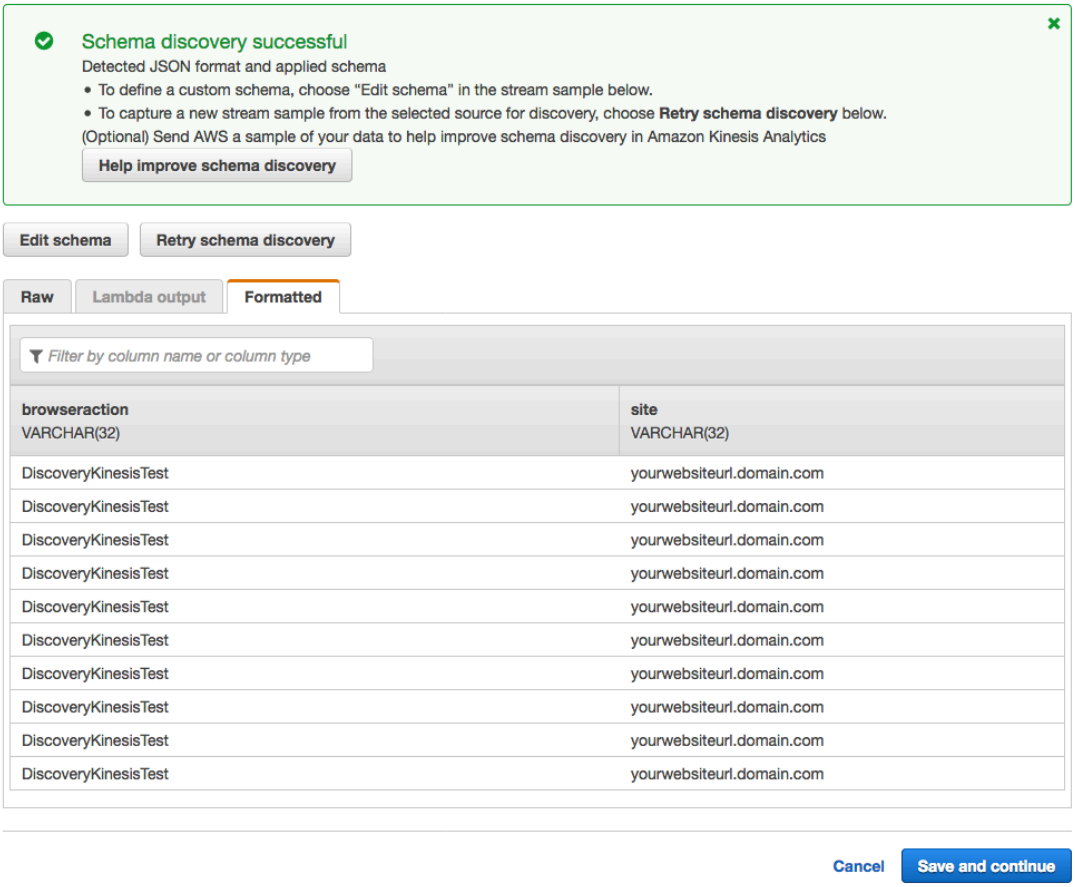
Navigate to the Amazon Kinesis Data Generator (Amazon KDG) which you setup in prelab and start sending the **Schema Discovery Payload** at **1 record per second** by clicking on Send data button. Make sure to select the appropriate region where you deployed the kinesis-pre-lab CloudFormation stack.



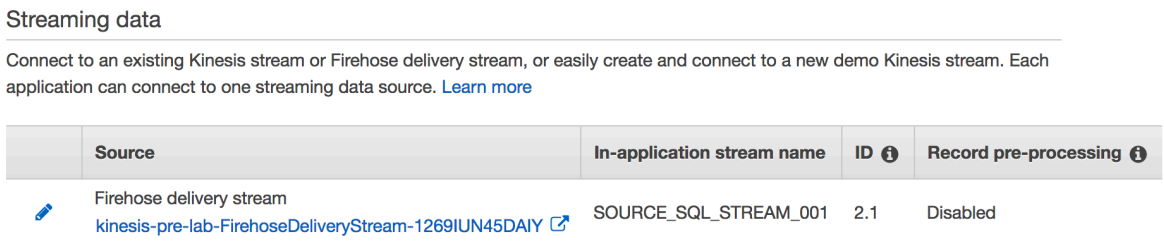


Now that your Kinesis Data Firehose is receiving data, you can continue configuring the Kinesis Data Analytics Application.

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



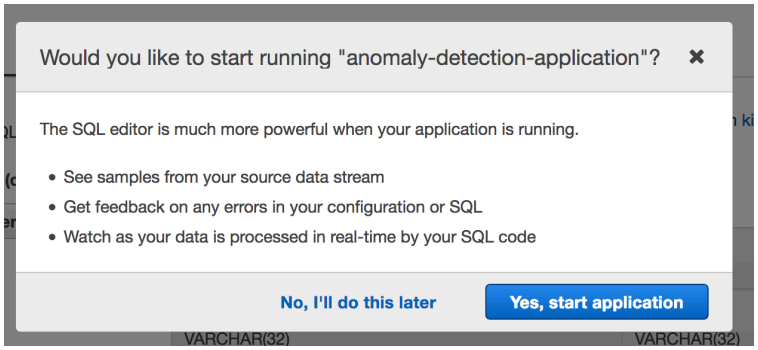
11. Click **Save and continue**. Your Kinesis Data Analytics Application is created with an input stream.



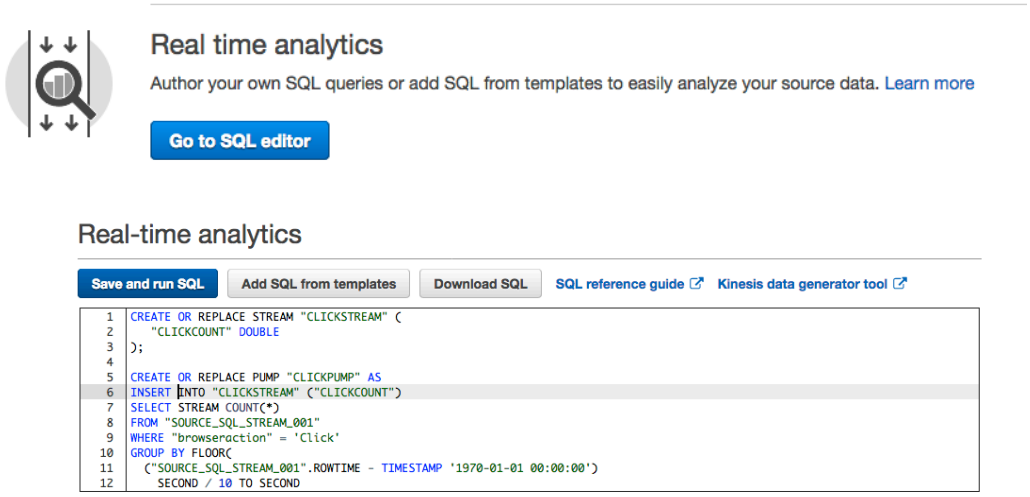
Now, you can add some SQL queries to easily analyze the data that is being fed into the stream.

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

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



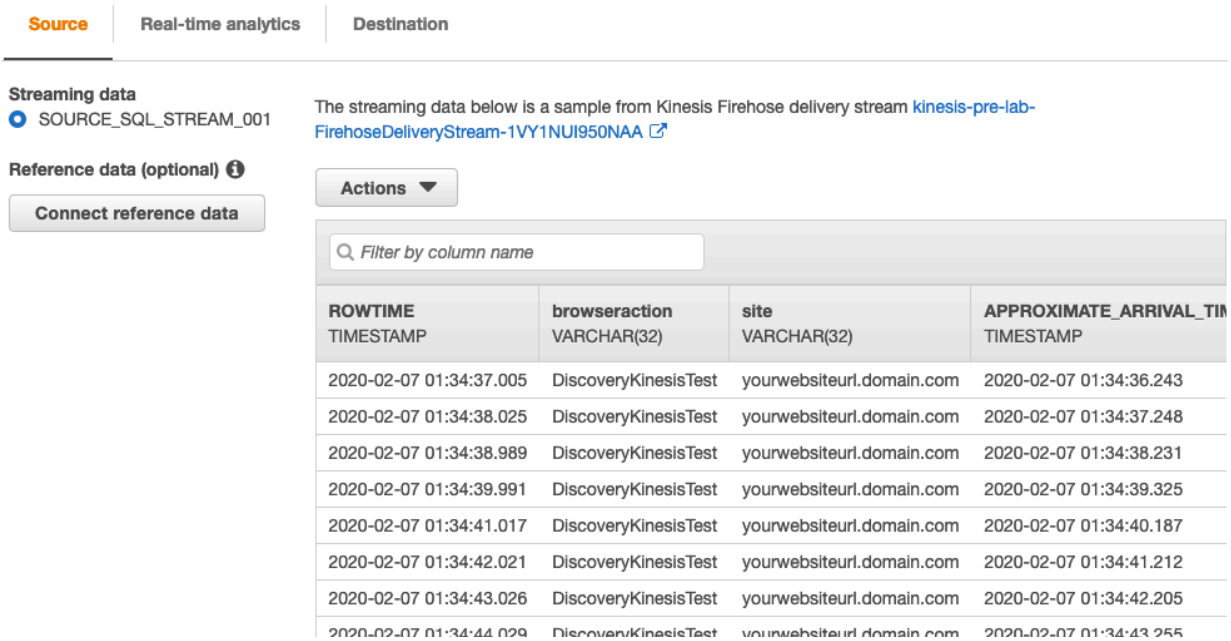
14. Click on this link, grab the SQL script and paste it into the SQL editor. (You can also find the code in Appendix)



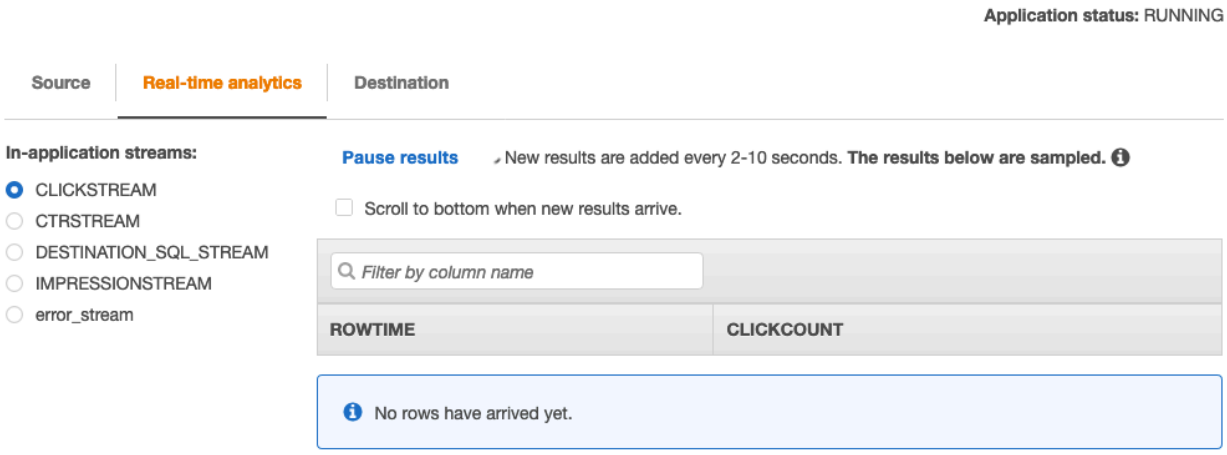
15. Click **Save and run SQL**. The analytics application starts and runs your SQL query. (You can find the SQL query in Appendix A.)

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".



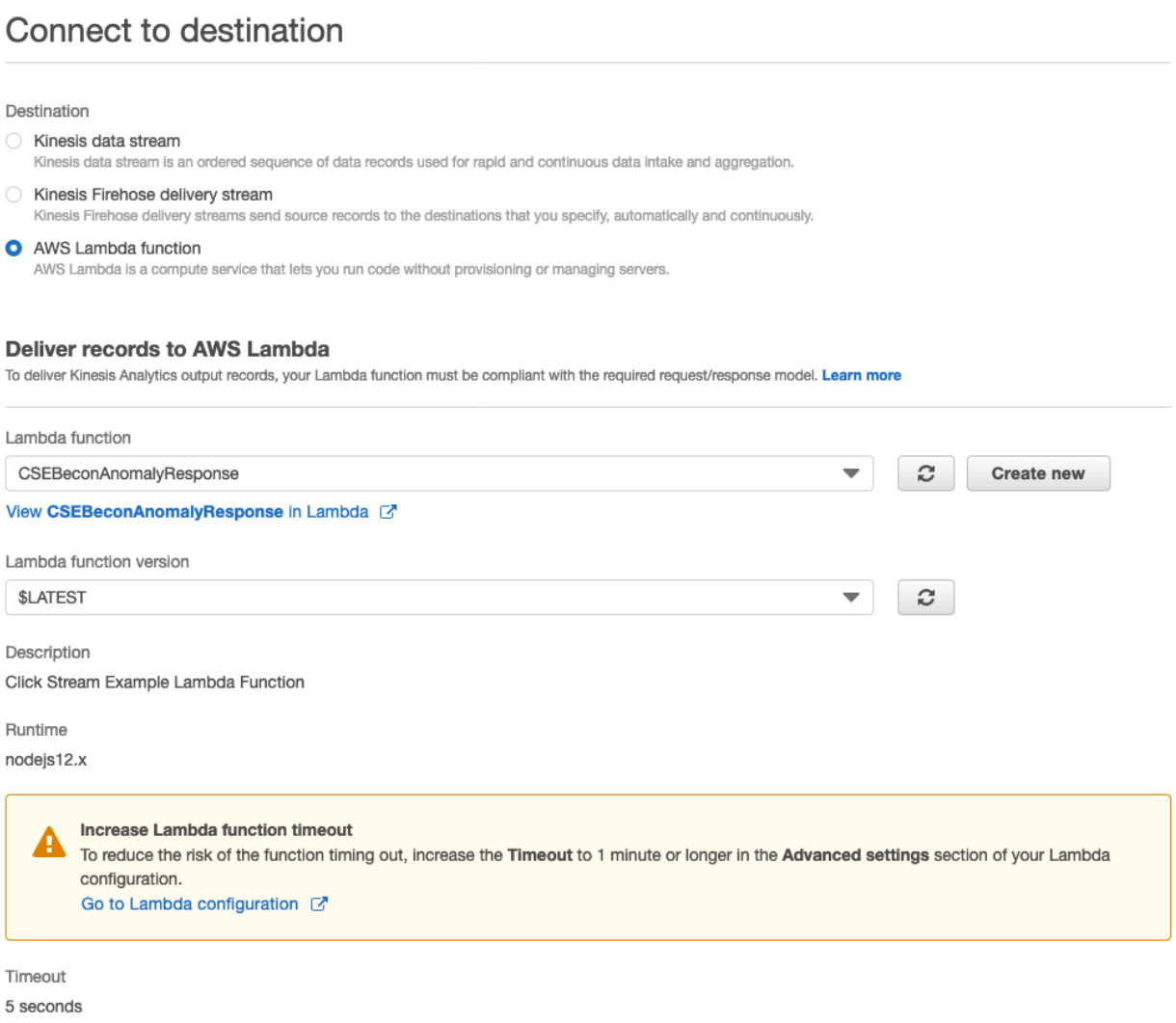
If you click the **Real-time analytics** tab, you will notice multiple in-application streams You will populate data in these streams later in the lab.



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**.

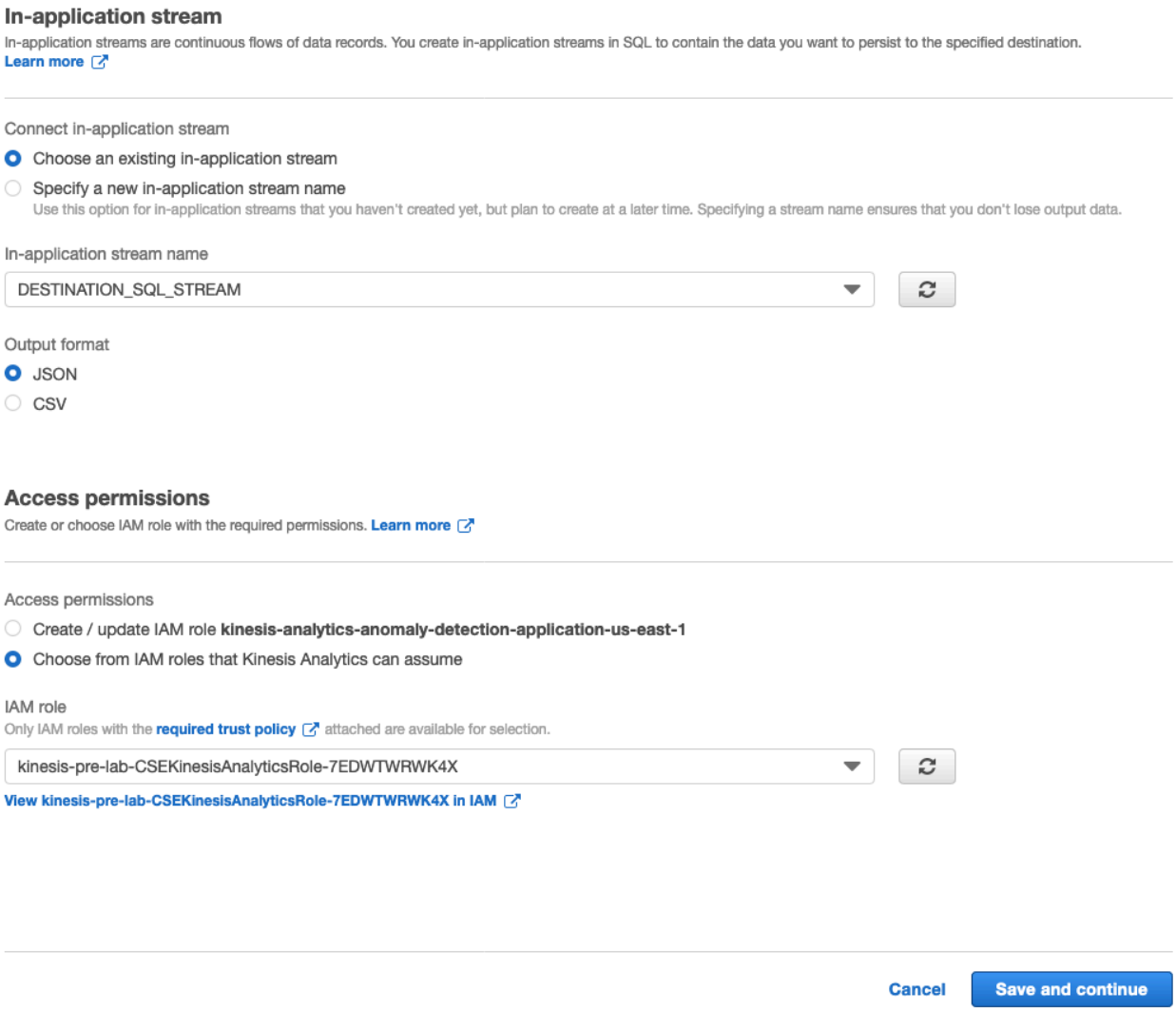


3. In the Deliver records to AWS Lambda section, make the following selections:
    a. For **Lambda function**, choose **CSEBeconAnomalyResponse**.
    b. For **Lambda function version**, choose **$LATEST**.

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>**.

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.



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.

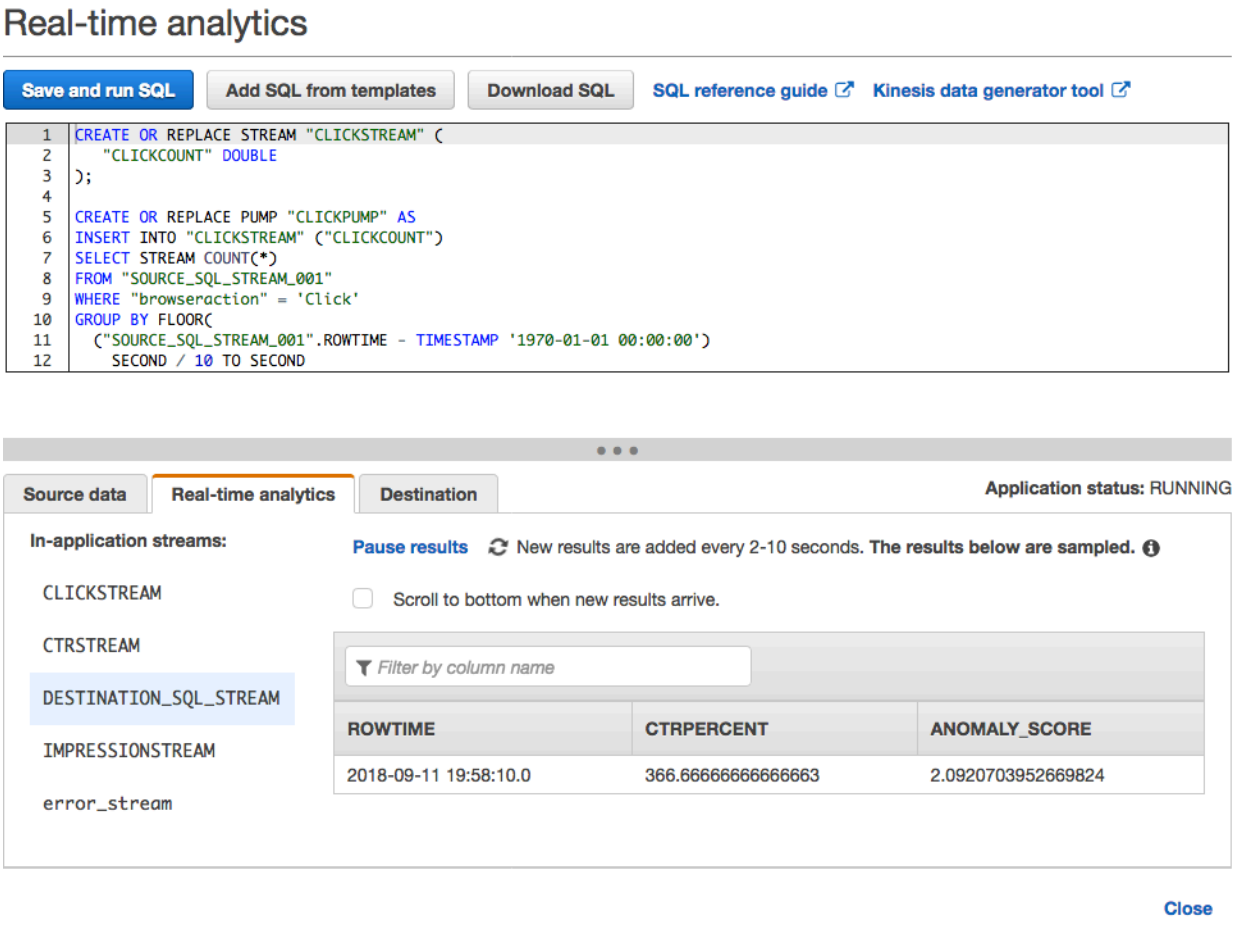
1. Open your KDG in five separate browser windows and sign in as the same user.
Note: Make sure to select the appropriate AWS region.

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

3. On another browser window, start sending the **Click payload** at a rate of 1 record per second **(keep this running)**.

4. 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.



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:

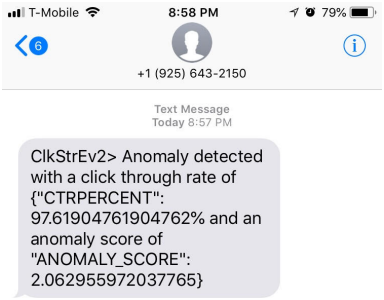
ClkStrEv2
Anomaly Detected
To:

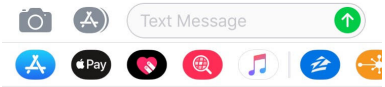
Anomaly detected with a click through rate of 3231.818181818182% and an anomaly score of 2.1384996930666254

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-west-2.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns: :ClickStreamEvent2:56ce971a-a67f-465c-8e12-d1dc099e499f&Endpoint=

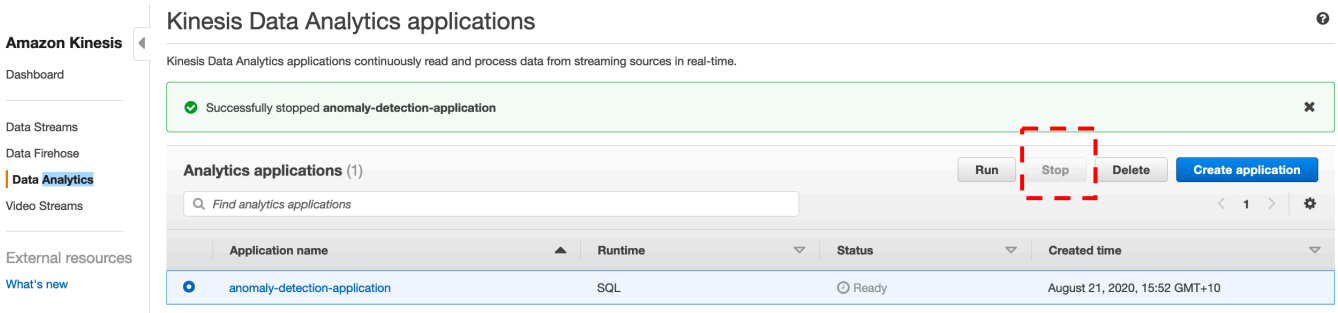
Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support

SMS Snapshot:





After you have completed the lab, click **Actions > Stop Application** to stop your application and avoid receiving a flood of SMS and e-mails messages.



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
);

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;
```