



Amazon Web Services Data Engineering Immersion Day

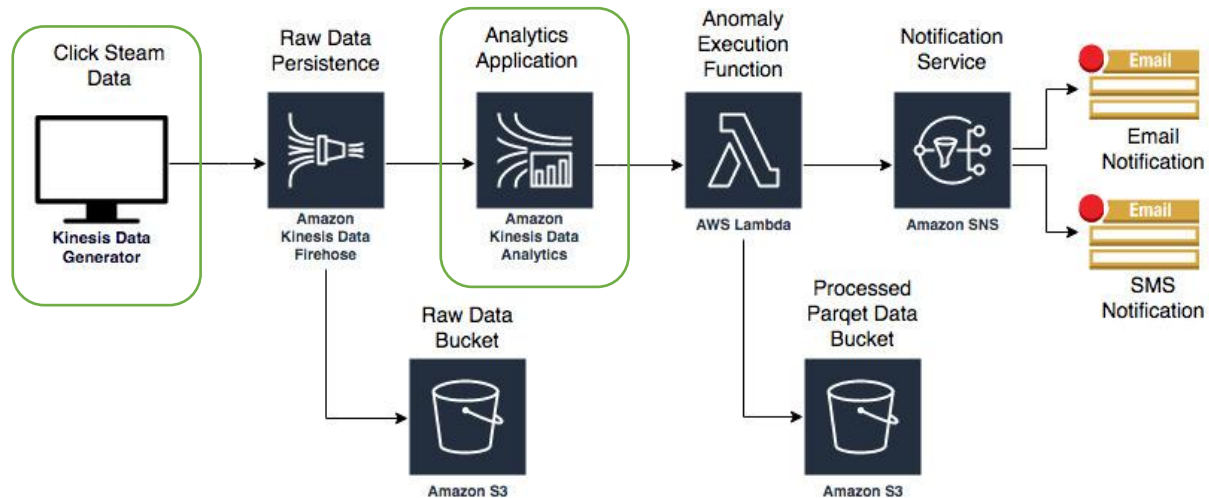
Lab 1 - Prelab. Real-Time Clickstream Anomaly Detection
August 2020

Table of Contents

<i>Introduction.....</i>	2
<i>Get Started Using the Lab Environment.....</i>	3
<i>CloudFormation Stack Deployment.....</i>	5
<i>Set up the Amazon Kinesis Data Generator.....</i>	7
<i>Set up Email and SMS Subscription.....</i>	10
<i>Review AWS Lambda Anomaly function:.....</i>	11

Introduction

This guide will help you set up the pre-lab environment for the Real-Time Clickstream Anomaly Detection Amazon Kinesis Data Analytics lab.



After you deploy the CloudFormation template, sign into your account to view the following resources:

- Two Amazon Simple Storage Service (Amazon S3) buckets: You will use these buckets to persist raw and processed data.
- One AWS Lambda function: This Lambda function will be triggered once an anomaly has been detected.
- Amazon Simple Notification Service (Amazon SNS) topic with an email and phone number subscribed to it: The Lambda function will publish to this topic once an anomaly has been detected.
- Amazon Cognito User credentials: You will use these user credentials to log into the Kinesis Data Generator to send records to our Amazon Kinesis Data Firehose.

Today, you are attending a formal AWS event and we've uploaded the CloudFormation template and the Kinesis Data Generator to an s3 bucket for you. 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/en/300.html>

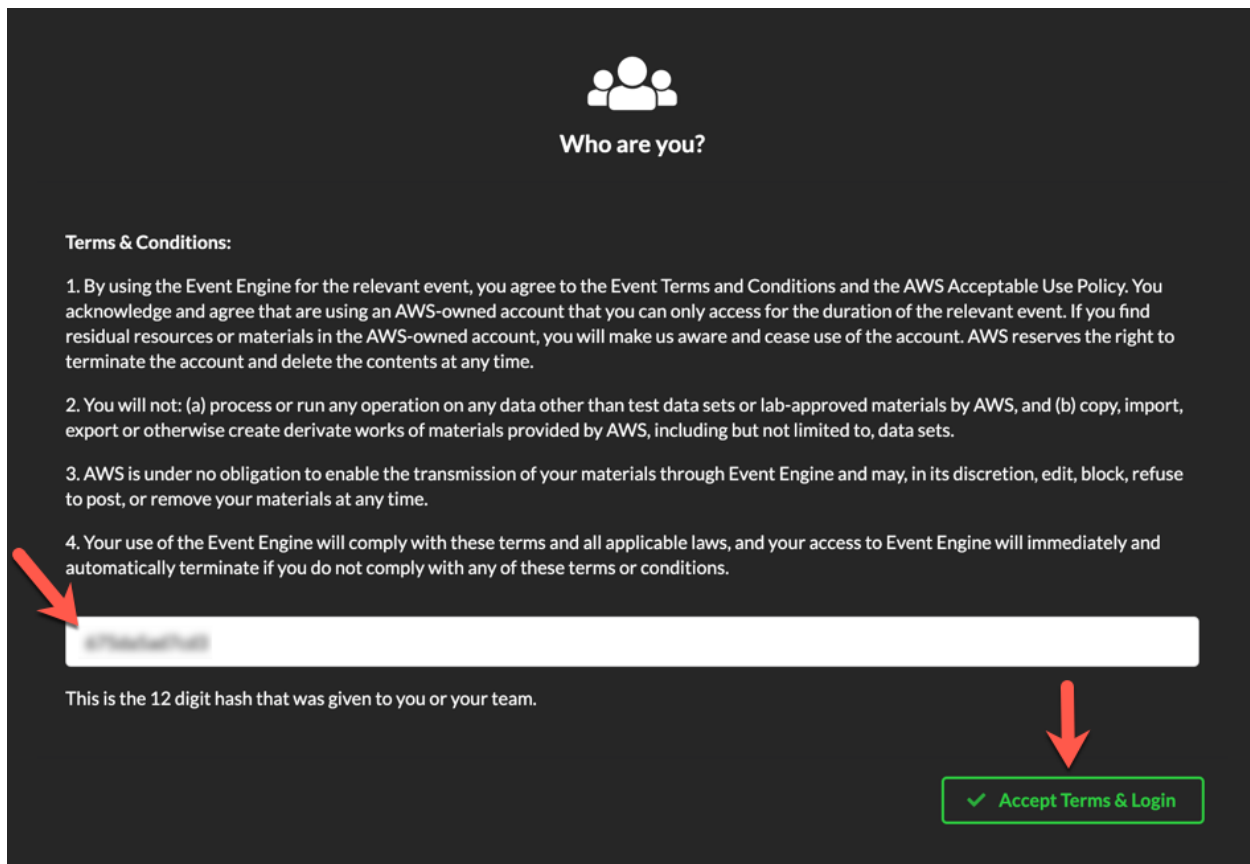
Get Started Using the Lab Environment

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.

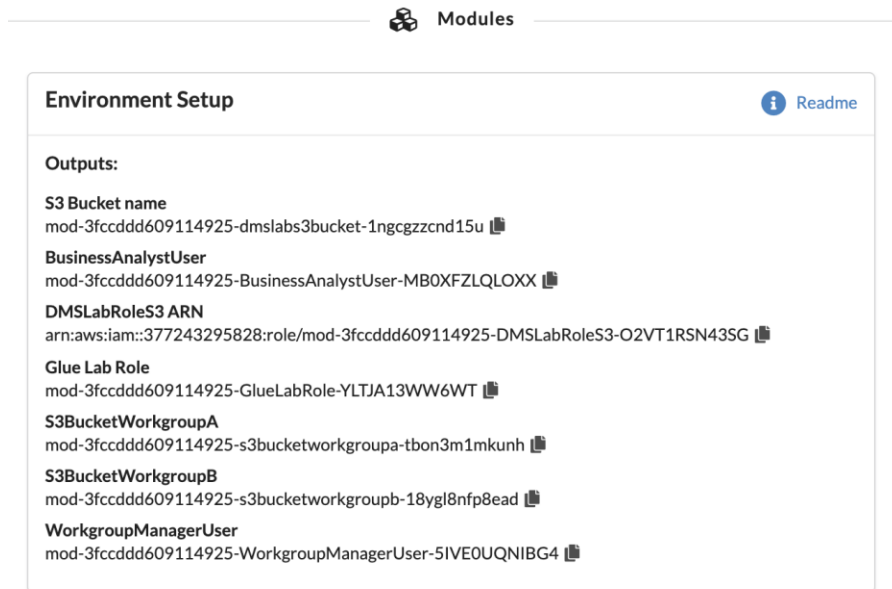
This is the 12 digit hash that was given to you or your team.

✓ Accept Terms & Login

2. On the Team Dashboard web page you will see a set of parameters that you will need during the labs. Best to save them to a text file locally, alternatively you can always go to this page to review them. Replace the parameters with the corresponding values from here where indicated in subsequent labs:

Lab 1 - Prelab. Real-Time Clickstream Anomaly Detection

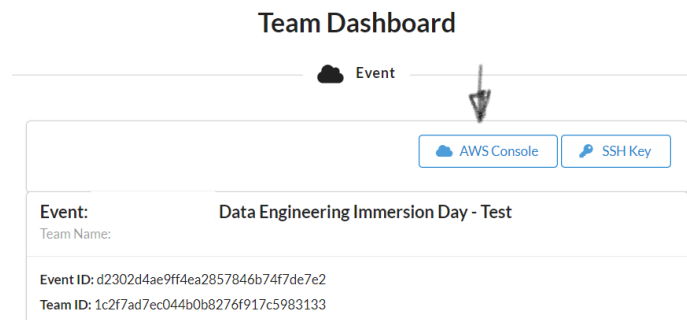
Because you're at a formal event, some AWS resources have been pre-deployed for your convenience, for example you can see a list of parameters on your event dashboard:



The screenshot shows the 'Modules' tab of an event dashboard. Under the 'Environment Setup' section, there is a list of pre-deployed AWS resources:

- S3 Bucket name**: mod-3fccddd609114925-dmslabs3bucket-1ngcgzccnd15u
- BusinessAnalystUser**: mod-3fccddd609114925-BusinessAnalystUser-MB0XFZLQLOXX
- DMSLabRoleS3 ARN**: arn:aws:iam::377243295828:role/mod-3fccddd609114925-DMSLabRoleS3-O2VT1RSN43SG
- Glue Lab Role**: mod-3fccddd609114925-GlueLabRole-YLTJA13WW6WT
- S3BucketWorkgroupA**: mod-3fccddd609114925-s3bucketworkgroupa-tbon3m1mkunh
- S3BucketWorkgroupB**: mod-3fccddd609114925-s3bucketworkgroupb-18ygl8nfp8ead
- WorkgroupManagerUser**: mod-3fccddd609114925-WorkgroupManagerUser-5IVE0UQNIBG4

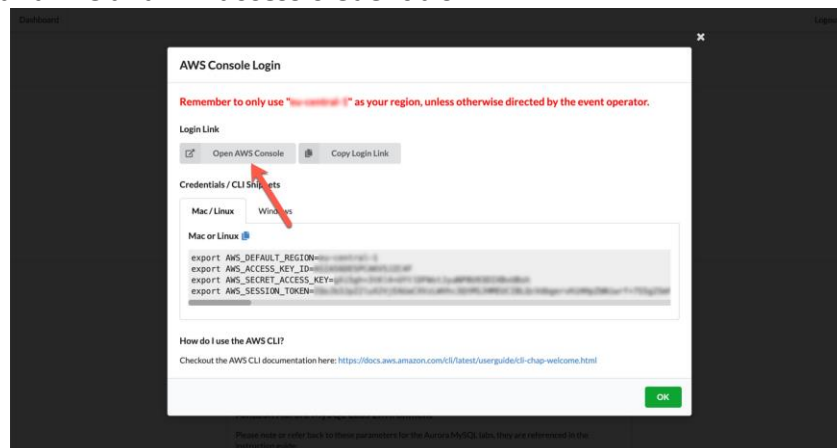
- On the Team Dashboard, please click AWS Console to log into the AWS Management Console:



The screenshot shows the 'Team Dashboard' with a tab labeled 'Event'. Below the tab, there are two buttons: 'AWS Console' and 'SSH Key'. A red arrow points to the 'AWS Console' button. Below the buttons, the event details are displayed:

- Event:** Data Engineering Immersion Day - Test
- Team Name:**
- Event ID:** d2302d4ae9ff4ea2857846b74f7de7e2
- Team ID:** 1c2f7ad7ec044b0b8276f917c5983133

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



The screenshot shows the 'AWS Console Login' dialog box. It contains a warning: 'Remember to only use "us-east-1" as your region, unless otherwise directed by the event operator.' Below this, there is a 'Login Link' section with a button 'Open AWS Console' and a 'Copy Login Link' button. A red arrow points to the 'Open AWS Console' button. Below the 'Login Link' section, there is a 'Credentials / CLI Setup' section with tabs for 'Mac / Linux' and 'Windows'. The 'Mac / Linux' tab is selected, showing a terminal window with the following commands:

```
export AWS_DEFAULT_REGION=us-east-1
export AWS_ACCESS_KEY_ID=mod-3fccddd609114925-BusinessAnalystUser-MB0XFZLQLOXX
export AWS_SECRET_ACCESS_KEY=mod-3fccddd609114925-BusinessAnalystUser-MB0XFZLQLOXX
export AWS_SESSION_TOKEN=mod-3fccddd609114925-BusinessAnalystUser-MB0XFZLQLOXX
```

Below the terminal window, there is a link to the AWS CLI documentation: <https://docs.aws.amazon.com/cli/latest/userguide/cli-welcome.html>. At the bottom right, there is an 'OK' button.

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

CloudFormation Stack Deployment

1. Use this link to create a new CloudFormation Stack:

https://console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/create/template?stackName=kinesis-pre-lab&templateURL=https://aws-dataengineering-day.workshop.aws.s3.amazonaws.com/Kinesis_Pre_Lab_us-east-1.json

Create stack

Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Template is ready ☐ Use a sample template ☐ Create template in Designer

Specify template
A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

☒ Amazon S3 URL ☐ Upload a template file

Amazon S3 URL

Amazon S3 template URL

S3 URL: https://aws-dataengineering-day.workshop.aws.s3.amazonaws.com/Kinesis_Pre_Lab_us-east-1.json [View in Designer](#)

[Cancel](#) [Next](#)

2. Click **Next** at the bottom of the page in as shown in above screenshot.
3. In the **Parameters** section, fill the following fields as shown in screenshot:
 - **Username:** This is your username to login to the Kinesis Data Generator
 - **Password:** This is your password for the Kinesis Data Generator. The password must be at least 6 alpha-numeric characters and contain at least one number and a capital letter.
 - **Email:** Type an email address that you can access. The SNS topic sends a confirmation to this address.
 - **SMS:** Type a phone number (+1XXXXXXXXXX) where you can receive texts from the SNS topic.

Lab 1 - Prelab. Real-Time Clickstream Anomaly Detection

CloudFormation > Stacks > Create stack

Step 1
[Specify template](#)

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review

Specify stack details

Stack name

Stack name

kinesis-pre-lab

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

Kinesis Pre Lab set up

Username
The username of the user you want to create in Amazon Cognito.

tester

Password
The password of the user you want to create in Amazon Cognito. Must be at least 6 alpha-numeric characters, and contain at least one number

email
Email address to send anomaly detection events.

xxxx@real-email.com

SMS
Mobile Phone number to send SMS anomaly detection events. +1XXXXXXXXXX

+61xxxxx

Cancel Previous **Next**

4. In the **Options**, section, keep the default values.
5. In the **Review** section, select the check box marked **I acknowledge that AWS CloudFormation might create IAM resources**.

Capabilities

i The following resource(s) require capabilities: [AWS::IAM::Role]

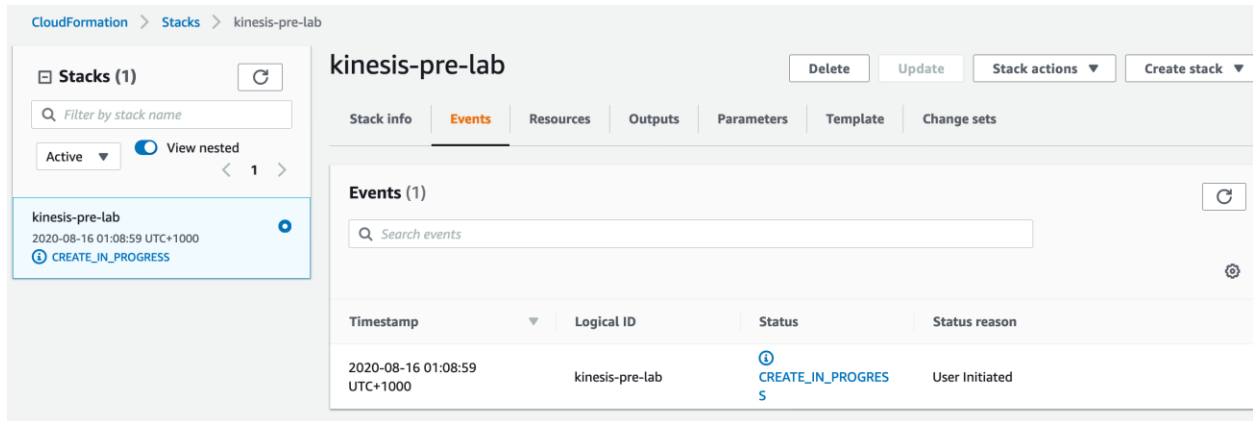
This template contains Identity and Access Management (IAM) resources that might provide entities access to make changes to your AWS account. Check that you want to create each of these resources and that they have the minimum required permissions.
[Learn more](#)

☒ **I acknowledge that AWS CloudFormation might create IAM resources.**

Cancel Previous **Create change set** **Create stack**

6. Click **Create**. CloudFormation redirects you to your existing stacks.

Lab 1 - Prelab. Real-Time Clickstream Anomaly Detection



CloudFormation > Stacks > kinesis-pre-lab

kinesis-pre-lab [Delete] [Update] [Stack actions] [Create stack]

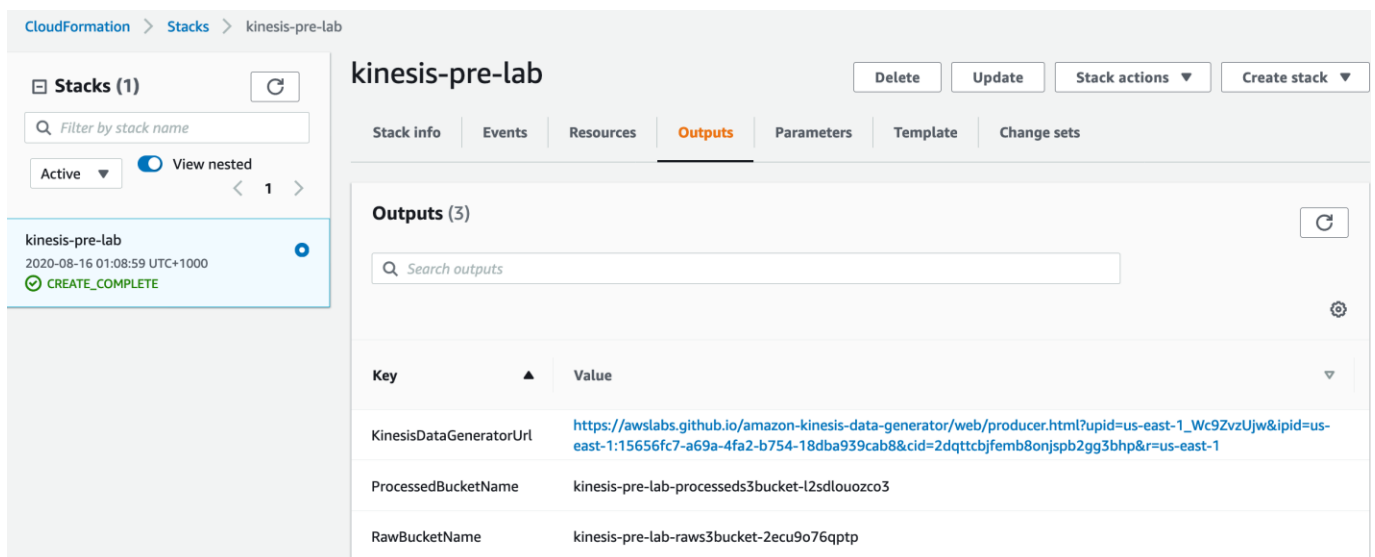
Stack info | **Events** | Resources | Outputs | Parameters | Template | Change sets

Events (1)

Search events

Timestamp	Logical ID	Status	Status reason
2020-08-16 01:08:59 UTC+1000	kinesis-pre-lab	CREATE_IN_PROGRESS	User Initiated

7. Once your stack is deployed, click the **Outputs** tab to view more information:
- **KinesisDataGeneratorUrl**: This value is the Kinesis Data Generator (KDG) URL.
 - **RawBucketName** – Store raw data coming from KDG.
 - **ProcessedBucketName** – Store transformed data



CloudFormation > Stacks > kinesis-pre-lab

kinesis-pre-lab [Delete] [Update] [Stack actions] [Create stack]

Stack info | Events | Resources | **Outputs** | Parameters | Template | Change sets

Outputs (3)

Search outputs

Key	Value
KinesisDataGeneratorUrl	https://awslabs.github.io/amazon-kinesis-data-generator/web/producer.html?upid=us-east-1_Wc9ZvzUjw&ipid=us-east-1:15656fc7-a69a-4fa2-b754-18dba939cab8&cid=2dqttcbjfb8onjpsb2gg3bhp&r=us-east-1
ProcessedBucketName	kinesis-pre-lab-processed3bucket-l2sdlouozco3
RawBucketName	kinesis-pre-lab-raws3bucket-2ecu9o76qptp

Congratulations! You are all done with the CloudFormation deployment.

Set up the Amazon Kinesis Data Generator

On the **Outputs** tab, notice the **Kinesis Data Generator URL**. Navigate to this URL to login into the Amazon Kinesis Data Generator (Amazon KDG).

Lab 1 - Prelab. Real-Time Clickstream Anomaly Detection

The KDG simplifies the task of generating data and sending it to Amazon Kinesis. The tool provides a user-friendly UI that runs directly in your browser. With the KDG, you can do the following tasks:

- Create templates that represent records for your specific use cases
- Populate the templates with fixed data or random data
- Save the templates for future use
- Continuously send thousands of records per second to your Amazon Kinesis stream or Firehose delivery stream

Let's test your Cognito user in the Kinesis Data Generator.

1. On the **Outputs** tab, click the **KinesisDataGeneratorUrl**.

The screenshot shows the AWS CloudFormation console for the 'kinesis-pre-lab' stack. The 'Outputs' tab is active, showing a table of three outputs:

Key	Value
KinesisDataGeneratorUrl	https://awslabs.github.io/amazon-kinesis-data-generator/web/producer.html?upid=us-east-1_Wc9ZvzUjw&ipid=us-east-1:15656fc7-a69a-4fa2-b754-18dba939cab8&cid=2dqttcbjfemb8onjspb2gg3bhp&r=us-east-1
ProcessedBucketName	kinesis-pre-lab-processed3bucket-l2sdlouozco3
RawBucketName	kinesis-pre-lab-raws3bucket-2ecu9o76qptp

2. Sign in using the **username** and **password** you entered in the CloudFormation console.

The screenshot shows the Amazon Kinesis Data Generator (KDG) login page. The 'Sign In' button is highlighted with a red box. The page title is 'Amazon Kinesis Data Generator'.

3. After you sign in, you should see the KDG console. You need to set up some templates to mimic the clickstream web payload.
- Create the following three templates. Copy the tab name highlight in bold letter and value as json string, refer screenshot:

Lab 1 - Prelab. Real-Time Clickstream Anomaly Detection

Schema Discovery Payload

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

Click Payload

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

Impression Payload

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

- Change Region to **US-EAST-1** and select a created Firehose Delivery Stream from the dropdown.
- Set **Records per second** to **1**

Your Amazon Kinesis Data Generator console should look similar to this example.

The screenshot shows the Amazon Kinesis Data Generator console interface. At the top, there's a header bar with "Amazon Kinesis Data Generator" on the left and a gear icon with "Con" on the right. Below the header, the configuration is organized into sections:

- Region:** A dropdown menu showing "us-east-1".
- Stream/delivery stream:** A dropdown menu showing "Kinesis-Pre-Lab-FirehoseDeliveryStream-1XMH0FAX1".
- Records per second:** Two tabs, "Constant" and "Periodic", with "Periodic" selected. Below the tabs is a text input field containing the value "1".
- Compress Records:** A checkbox that is currently unchecked.
- Record template:** A section with a title and an information icon. It contains a row of tabs: "Schema Discovery Payload", "Click Payload", "Impression Payload", "Template 4", and "Template 5". The "Schema Discovery Payload" tab is active, showing a text area with the JSON payload:

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

Don't click on Send Data yet, leave this browser tab open, we will do that during the main lab.

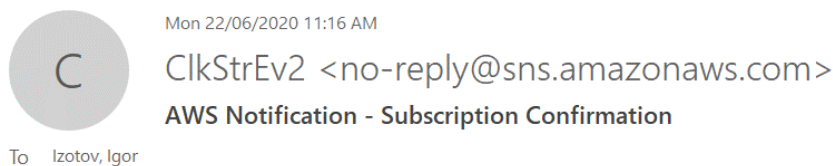
Set up Email and SMS Subscription

1. Navigate to Amazon SNS Topics by following this link:
<https://console.aws.amazon.com/sns/v3/home?region=us-east-1#/topics>
2. Click the topic name. The Topic details screen appears listing the e-mail/SMS subscription as pending or confirmed.

The screenshot shows the Amazon SNS console interface. On the left is a navigation menu with options like Dashboard, Topics, Subscriptions, Mobile, Push notifications, and Text messaging (SMS). The main panel displays the details for a topic named 'ClickStreamEvent'. It includes fields for Name, ARN, Display name, and Topic owner. Below this, there are tabs for Subscriptions, Access policy, Delivery retry policy (HTTP/S), Delivery status logging, Encryption, and Tags. The 'Subscriptions' tab is active, showing a table with two subscriptions, both in a 'Confirmed' status. The first subscription is for EMAIL and the second is for SMS.

ID	Endpoint	Status	Protocol
13ed5fe6-6caa-49f4-8efc-af801997879a	[redacted].com	Confirmed	EMAIL
276d59b2-b91c-4f83-ab9f-7cdf3acc6f8b	+1 [redacted]	Confirmed	SMS

3. Check your inbox for a subscription confirmation email from no-reply@sns.amazonaws.com, click **Confirm subscription** to confirm



You have chosen to subscribe to the topic:
arn:aws:sns:us-east-1:222752441477:ClickStreamEvent2

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
[Confirm subscription](#)

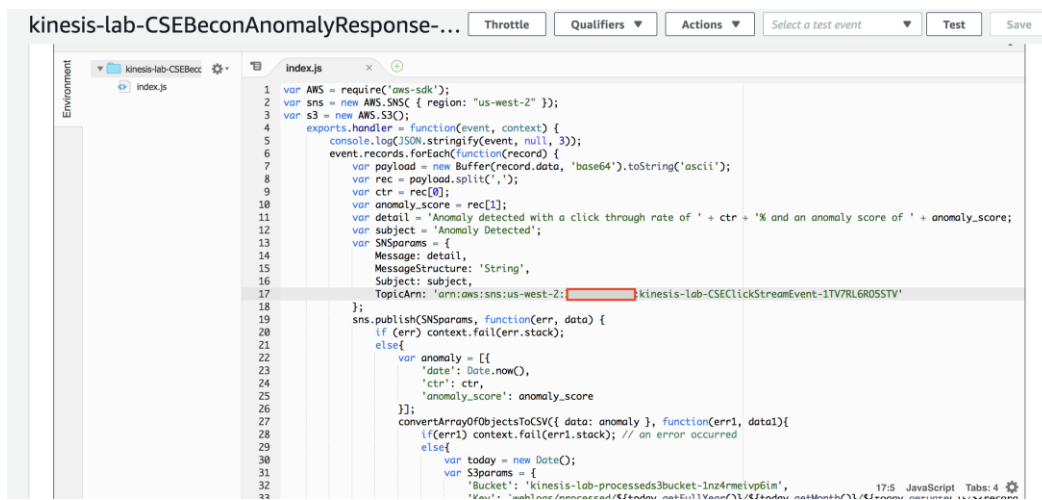
Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#)

Note: If you can't locate the request confirmation email, make sure to check your email junk folder.

Review AWS Lambda Anomaly function:

CloudFormation template already deployed this Lambda function. You just need to spend few minutes to observe code and understand the action behind the lambda trigger:

1. In the console, navigate to **CSEBeconAnomalyResponse** AWS Lambda function by following the link: <https://console.aws.amazon.com/lambda/home?region=us-east-1&#/functions/CSEBeconAnomalyResponse?tab=configuration>
2. Scroll down to code section.



```
1 var AWS = require('aws-sdk');
2 var sns = new AWS.SNS({ region: 'us-west-2' });
3 var s3 = new AWS.S3();
4 exports.handler = function(event, context) {
5   console.log(JSON.stringify(event, null, 3));
6   event.records.forEach(function(record) {
7     var payload = new Buffer(record.data, 'base64').toString('ascii');
8     var rec = payload.split(',');
9     var ctr = rec[0];
10    var anomaly_score = rec[1];
11    var detail = 'Anomaly detected with a click through rate of ' + ctr + '% and an anomaly score of ' + anomaly_score;
12    var subject = 'Anomaly Detected';
13    var SNSparams = {
14      Message: detail,
15      MessageStructure: 'String',
16      Subject: subject,
17      TopicArn: 'arn:aws:sns:us-west-2:kinesis-lab-CSEClickStreamEvent-1TV7RL6R05STV'
18    };
19    sns.publish(SNSparams, function(err, data) {
20      if (err) context.fail(err.stack);
21    } else {
22      var anomaly = [{
23        'date': Date.now(),
24        'ctr': ctr,
25        'anomaly_score': anomaly_score
26      }];
27      convertArrayOfObjectsToCSV({ data: anomaly }, function(err1, data1){
28        if(err1) context.fail(err1.stack); // an error occurred
29      } else {
30        var today = new Date();
31        var S3params = {
32          'Bucket': 'kinesis-lab-processed3bucket-1nz4rme1vp6im',
33          'Key': 'upload/processed/ETtoday_nz4r1Yanp7Y/Etoday_nz4rme1vp6im/processed/processed.csv'
34        };
35        s3.upload(S3params, function(err2, data2) {
36          if(err2) context.fail(err2.stack);
37        } else {
38          console.log('Upload successful');
39        }
40      }
41    }
42  });
43 }
```

3. Review the code in the Lambda code editor. Notice the TopicArn value matches the SNS topic ARN from the previous step.

You've completed the pre-lab instructions. Please proceed to lab 3.