

# Lab 1: Filtering Real-Time data with Amazon Kinesis Platform

## Overview

In this exercise, you will learn how to build a real-time application using the AWS Kinesis platform. As a first step, you'll see how to stream data using Amazon Kinesis, a fully managed cloud-based service for streaming large, distributed data streams in real-time.

To make the exercise more realistic, we will simulate events generated from a game (or IoT device), in this case a stream of events corresponding to the score, game ID, user ID and some data.

We'll create Amazon Kinesis Analytics application to consume data from Amazon Kinesis Stream and write them into an Amazon Firehose for delivery to S3.



## Getting Started

### Launch an Amazon Kinesis Stream

First, you will create an Amazon Kinesis stream to capture the incoming game data, consisting of a single shard.

1. **Open the AWS Management Console** by clicking **Open Console**.
2. Choose the Region: **N. Virginia**

3. From the **Services** menu on the AWS Management Console, **select Amazon Kinesis Stream**.

Amazon Kinesis Create Stream

A stream is composed of multiple shards, each of which provides a fixed unit of capacity. The total capacity of the stream is the sum of the capacities of its shards. Each shard corresponds to 1 MB/s of write capacity and 2 MB/s of read capacity. See the [Amazon Kinesis Developer Guide](#) for more information on estimating number of shards needed for your stream. Note that the cost of the stream is also a function of the number of shards. To learn more about the stream, see the [Amazon Kinesis Pricing Page](#).

Stream Name\* bootcamp

☐ Help me decide how many shards I need

Number of Shards\* 1

Values calculated based on the number of shards entered above:

	Read:	Write:
Total Stream Capacity:	2 MB/s	1 MB/s
Max Transactions/second:	5	1000

\* Required information

Cancel Create

4. On the following screen, click **Create Stream**.
5. For **Stream Name**, use **<user>\_stream** and input **1** for **Number of Shards**. After that, select **Create** in the lower-right corner of the screen.
6. Verify that your Amazon Kinesis stream is in the **CREATING** Status. After a few moments, the status should transition to **ACTIVE**.

Click the stream name itself to review the stream summary in the **Stream Details** page. You should see empty Amazon **Cloud Watch** graphs for **Amazon Kinesis**. The red lines preceding the Write Capacity and Read Capacity graphs indicate the maximum provisioned write capacity of 1 MB/sec and read capacity of 2 MB/sec for **Kinesis Stream** with a single shard.

## Using Amazon Kinesis Streams

Now we'll write a producer application that simulates mobile application events (Similar example can simulate IoT event as well). It will write records to your **Amazon Kinesis** stream containing the **userID**, **applicationID**, **app score**, **current time**, and some **randomly generated example data**.

## Producer Application

1. On an EC2 instance or in your local computer with CLI configured, download the shell script **kinesis\_client.sh** from <https://bit.ly/2zjIbUZ> to start writing some records into Amazon Kinesis stream you created earlier.

2. On the downloaded script `kinesis_client.sh` you should replace your stream name the section in yellow `<user>_stream` with your kinesis stream name –i.e. `user1_stream`.

```
#!/bin/sh

while true
do
    eventTime=$(date +"%Y-%m-%d-%T")
    userId=$(( ( RANDOM % 50 ) + 1 ))
    appId=$(( ( RANDOM % 100 ) + 1 ))
    appScore=$(( ( RANDOM % 100 ) + 1 ))
    appData=SomeTestData

    echo "$eventTime,$userId,$appId,$appScore,$appData"

    aws kinesis put-record --stream-name <user>_stream --data
"$eventTime,$userId,$appId,$appScore,$appData"$'\n' --partition-key $appId --region
us-east-1
done
```

3. Give execution permissions and run script by executing the following commands in your terminal:

```
chmod 700 kinesis_client.sh
```

```
./kinesis_client.sh
```

If you got an access denied error on the Put Record your EC2 instance probably do not have Kinesis permission for your user. Go back to IAM and add it.

Keep the script running until the end of this lab.

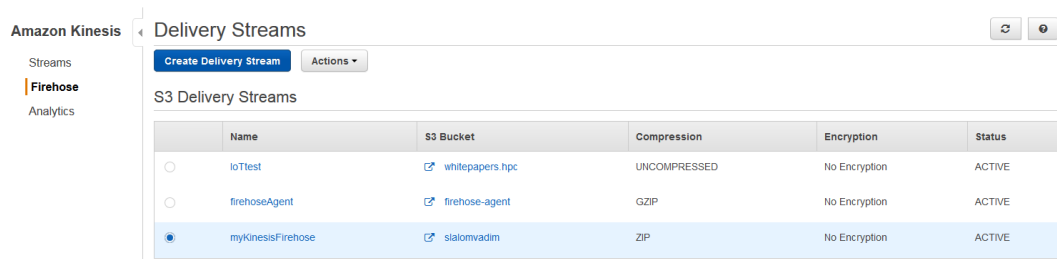
In the AWS Management Console, return to the Amazon Kinesis section as described in steps 1 and 2. Click the stream **Kinesis Stream**. Under **Monitoring**, observe the PUT metrics associated with your stream. It takes a few minutes for **Cloud Watch** to show up metrics.

## Setup up an Amazon Kinesis firehose with delivery to Amazon S3

1. **Navigate to the AWS Management Console and from the Services menu on the AWS Management Console, select Kinesis.**

Ensure that you're using **US East (N. Virginia)** as your region.

2. On the following screen select Kinesis Firehose, and click on **Create Delivery Stream**.



3. Delivery Stream Name: **<user>\_firehose\_stream**

4. Choose Source: **Kinesis Stream**

5. Kinesis stream: **<user>\_stream**

6. Click **Next**

7. Transform Records with Lambda: **Disabled**

**Note:** You can invoke a Lambda function to perform transformations on each stream processed.

8. Click **Next**

9. Destination: **Amazon S3**

**Note:** Kinesis Firehose allows S3, Redshift and Elastic Search as possible destinations.

10. S3 Bucket: **Create New**

11. S3 bucket name: **<user>-bigdata-day**

12. Region: **US East (N. Virginia)**

13. Create S3 Bucket

14. Prefix: **stream**

15. Click **Next**

16. In the Configuration page and provide Buffer size "**5**" and Buffer interval "**60**".

17. S3 compression: **Gzip**

18. S3 encryption: **Disabled**

19. Error logging: **Enabled**
20. In the configuration page, under IAM role, create new. Let the wizard create a new IAM Role with name **<user>\_firehose\_delivery\_role**.
21. Click **Allow**
22. After that, select **Next** in the lower-right corner of the screen to go to the Review page. Verify the details on the Review page, and click on “**Create Delivery Stream**”.

Amazon Kinesis < Create Delivery Stream

Streams  
**Firehose**  
Analytics

Step 1: Destination  
Step 2: Configuration  
**Step 3: Review**

**Review**

Review your destination and configuration before creating your delivery stream.

**Destination** Edit

Destination	Amazon S3
Delivery stream name	myKinesisFirehose
S3 bucket	slalomwadiim
S3 prefix	none

**Configuration** Edit

S3 buffer size	5
S3 buffer interval	300
S3 compression	GZIP
S3 encryption	No Encryption
Error logging	Enabled
IAM role	firehose_delivery_role

Cancel Previous Create Delivery Stream

23. Verify that your Amazon Kinesis Firehose delivery stream is in the **CREATING** Status. After a few moments, the status should transition to **ACTIVE**.

**Kinesis Firehose** will read the streaming data being stored in the **Kinesis stream** and will deliver it into **S3** in files based on batches of either **5MB** or **60secs**. After some minutes, you can check in the **S3** bucket **<user>-bigdata-day** that the files are being created. If you have a look at the produced files in **S3**, you will see that are **csv** files compressed in **.gz** format with the information inserted by the simulator script running into our **EC2** instance into the **Kinesis stream**.

We are going to use **Kinesis Analytics** to run queries in real time on the received streaming data.

## Setup up an Amazon Kinesis Analytics application.

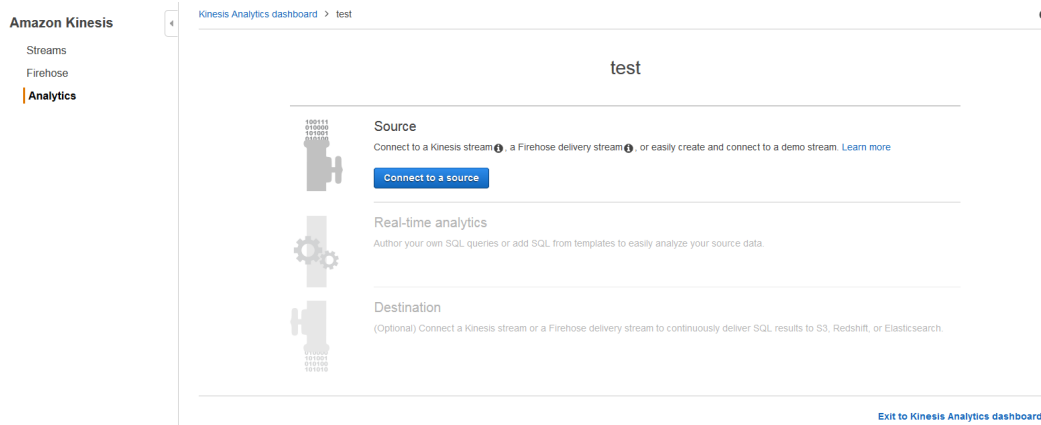
1. **Navigate to the AWS Management Console** and from the **Services** menu on the AWS Management Console, **select Amazon Kinesis Analytics**.

Ensure that you're using **US East (N. Virginia)** as your region.

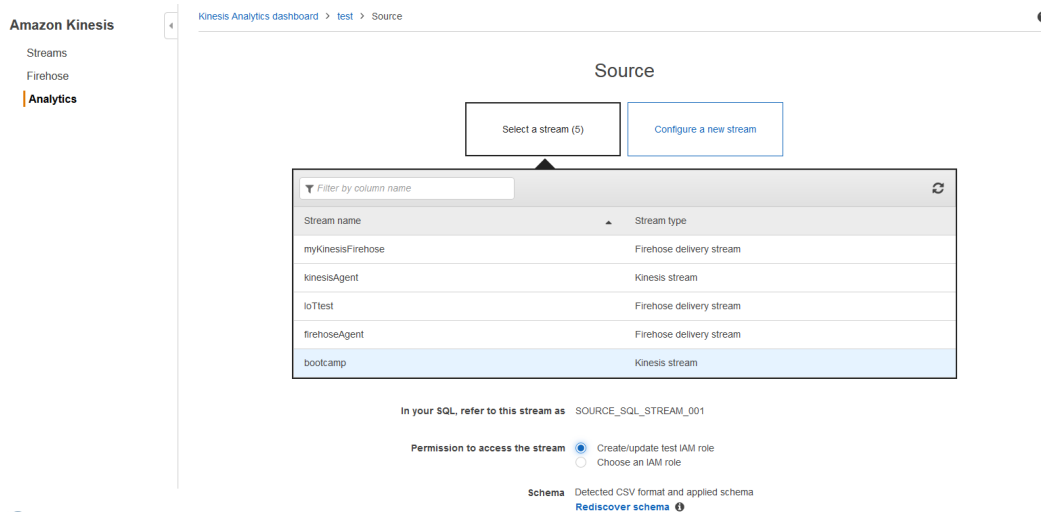
2. On the following screen select Amazon Kinesis Analytics, and click on **Create new application**. Give it a name: **<user>\_kinesis\_analytics**.



3. Choose “**Connect to a source**”



4. Select “**<user>\_stream**” Kinesis Stream created in previous exercise. You will have to “**Create/update IAM role**” to access the stream.



5. Click on **Discover Schema**. Analytics calls the [DiscoverInputSchema](#) API action, which infers a schema by sampling records from your selected input data stream. You can see the applied schema on your data in the formatted sample shown in the browser, as well as

the original sample taken from the raw stream. You can then edit the schema to fine tune it to your needs.

bootcamp	Kinesis stream
firehoseAgent	Firehose delivery stream
IoTTest	Firehose delivery stream
kinesisAgent	Kinesis stream
myKinesisFirehose	Firehose delivery stream
myKinesisFirehoseRS	Firehose delivery stream

In your SQL, refer to this stream as SOURCE\_SQL\_STREAM\_001

Permission to access the stream ☐ Create/update test2 IAM role ☒ Choose an IAM role  
 kinesis-analytics-MyTest

Schema Detected CSV format and applied schema  
[Rediscover schema](#)

Formatted stream sample Raw stream sample

Filter by column name or column type

COL0	COL1	COL2	COL3	COL4
VARCHAR(32)	SMALLINT	SMALLINT	TINYINT	VARCHAR(16)
2016-08-23-04:42:16	21252	651	77	VadimTest
2016-08-23-04:42:17	31516	764	93	VadimTest
2016-08-23-04:42:17	28955	801	25	VadimTest
2016-08-23-04:42:17	26199	758	90	VadimTest

6. Click on **Edit Schema** and name the columns as shown in the following picture:

Filter by column name

	Column order	Column name	Column type	
+ Add column				
x	1	eventTime	VARCHAR	Length: 32
x	2	userId	INTEGER	
x	3	appld	INTEGER	
x	4	appScore	INTEGER	
x	5	appData	VARCHAR	Length: 16

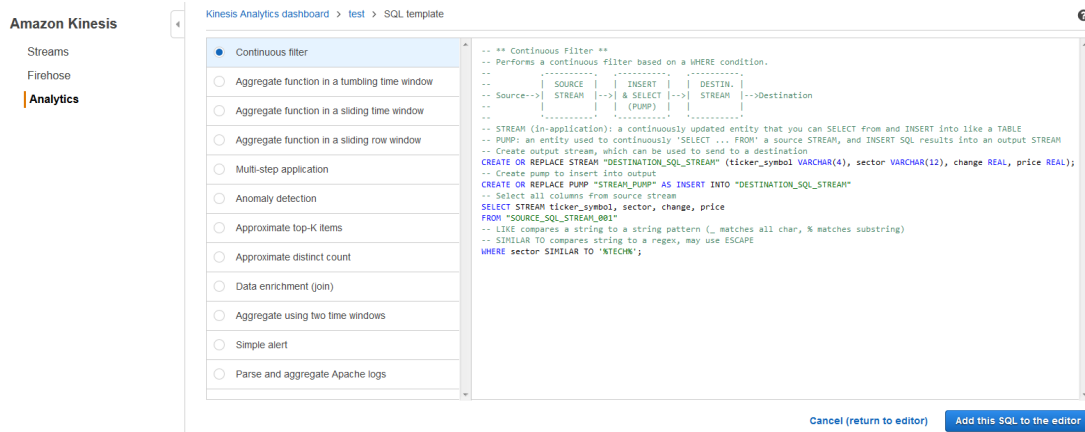
Cancel Save schema and update stream samples

Click **Save schema and update stream samples**. It will take a couple of minutes to restart the application.

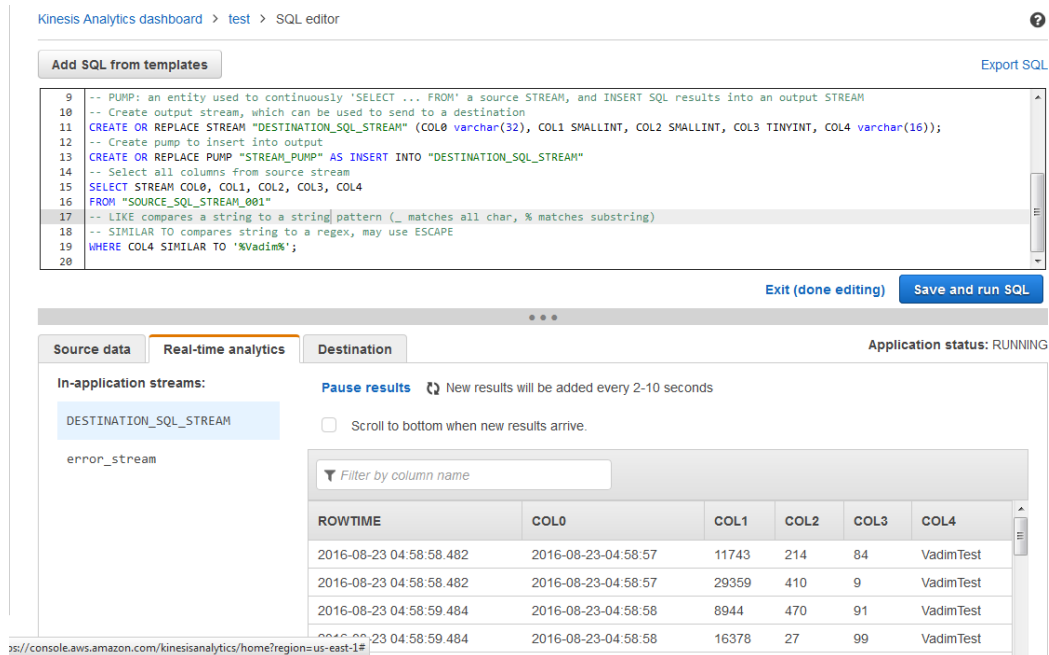
7. Go to **Kinesis Analytics applications** -> **<user>\_kinesis\_analytics** on top of the window and go to the **Real-Time analytics** section. Click on **Go to SQL Editor**. This SQL editor is the development environment for **Amazon Kinesis Analytics**. On the top portion of the screen, there is a text editor with syntax highlighting and intelligent auto-complete, as well as a number of SQL templates to help you get started. On the bottom portion of the screen, there is an area for you to explore your source data, your intermediate SQL results,

and the data you are writing to your destinations. You can view the entire flow of your application here, end-to-end.

8. “Add SQL Template” on the top left and pick up “Continuous filter”. Click on Add this SQL to the editor.



9. Replace (copy & paste) the template with the test query that you can download from <https://bit.ly/2A6SXgV>
10. The SELECT statement in this SQL template performs a count over a 10-second tumbling window. A window is used to group rows together relative to the current row that the Amazon Kinesis Analytics application is processing. Choose Save and run SQL.





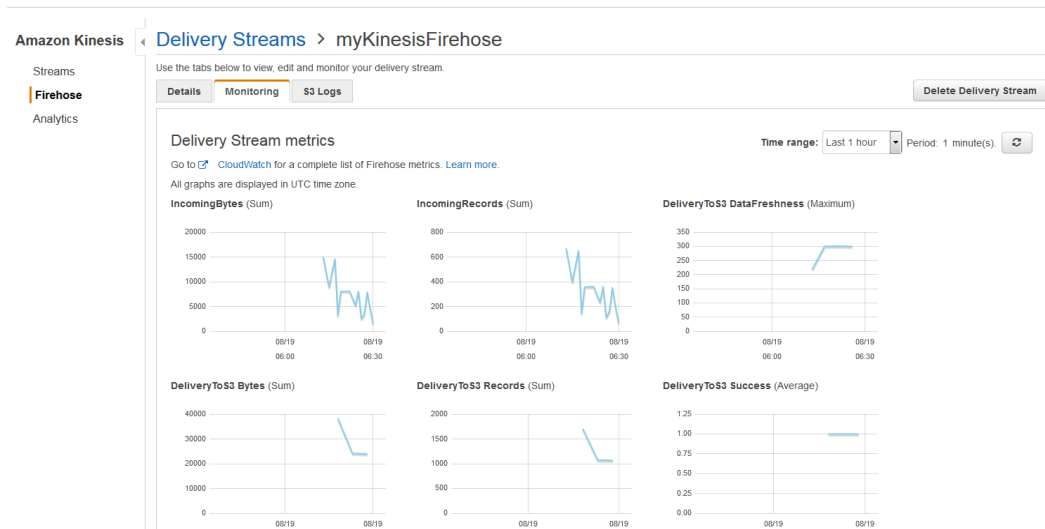
In the source data tab, you can see the streaming data being read from the Kinesis stream (**SOURCE\_SQL\_STREAM\_001**).

In the Real-time analytics tab, you can see the results of our query, which counts how many requests are received from a specific user in 10secs interval.

In the destination tab, we could click on **Connect to Destination**. We are going to configure a new Firehose stream to store the results of our query in **S3**

11. Click **Add Destination**
12. Select **Kinesis Firehose** and click on **Create New**
13. You are redirected to the **Kinesis Firehose** web page. Click on **Create Delivery Streams**.
14. Name your firehose stream: **<user>\_firehose\_analytics**
15. Source: **Direct PUT or other sources**.
16. Click **Next**
17. Record transformation: **Disabled**.
18. Click **Next**
19. Destination: **S3**
20. S3 bucket: **<user>-bigdata-day**
21. Prefix: **analytics**
22. Click: **Next**
23. Buffer size: **5 MB**
24. Buffer Interval: **60 seconds**
25. S3 compression: **Gzip**
26. S3 encryption: **Disabled**
27. IAM Role: Choose the role created before: **<user>\_firehose\_delivery\_role**.
28. Policy Name: Choose: **oneClick\_firehose\_delivery\_role**
29. Click **Allow**. Click **Next**
30. **Create Delivery Stream**.
31. Go back to the **Kinesis Analytics** section. Expand your **Kinesis Analytics** app **<user>\_kinesis\_analytics** and click on **Application Details**.
32. Click Destination -> **Connect to a destination**.
33. Select **Kinesis Firehose delivery Stream** -> **<user>\_firehose\_analytics**
34. Select for Connect in-application stream: **Choose an existing in-application stream -> DESTINATION\_SQL\_STREAM**
35. Output format: **CSV**
36. Permission to access the stream -> **Select a IAM role** that Kinesis Analytics can assume. Choose an existing Kinesis Analytics role.
37. Click **Save and Continue**.

38. In a few minutes check that **Firehose** start picking up data and storing them on **S3**



### 39. Congratulations, you just wrote your first SQL query on streaming data and saved result on S3 !!!

Upload

Create Folder

Actions

Search by prefix

None

Properties

Transfers

All Buckets

/ slalomvadm / 2016 / 08 / 19 / 06

Name	Storage Class	Size	Last Modified
 myKinesisFirehose-3-2016-08-19-06-23-154f9502-2798-4908-a751-971cd082331.zip	Standard	9.4 KB	Thu Aug 18 23:18:24 GMT+700 2016
 myKinesisFirehose-3-2016-08-19-06-23-244c29747-3b0a-442c-9391-9e6961dc338.zip	Standard	7.2 KB	Thu Aug 18 23:23:25 GMT+700 2016

40. You can check the contents of the S3 bucket: **<user>-bigdata-day**. You should see files been created under **analytics<YEAR> (results of Kinesis Analytics queries)** and **streams<YEAR> (data streamed into the Kinesis stream)** directories.

41. Use **CTRL-Z** to exit the script on your **EC2** instance. Do not delete the created resources since we will use some in next labs.