



Linux Academy
Hands-on Lab

Analyzing Internet Data with Kinesis

Contents

Create a Firehose Stream.....	1
Stream Data from EC2.....	1
Kinesis Analytics.....	1
Set Destinations.....	2
Review.....	3

Related Courses

[AWS Certified Big Data Specialty](#)

Related Videos

[S3 Core Concepts](#)

[Kinesis Firehose](#)

Need Help?

[Linux Academy Community](#)

*... and you can
always send in a
support ticket on
our website to talk
to an instructor!*

Lab Connection Information

- Labs may take up to five minutes to build
- Access to an AWS Console is provided on the Hands-on Lab page, along with your login credentials
- Ensure you are using the N. Virginia region
- Labs will automatically end once the allotted amount of time finishes

In this lab, we'll use a combination of AWS Kinesis services to analyze sample viewer data for Linux Academy course videos.

To get started, click the **Start Lab** button on the Hands-on Lab page.

Create a Firehose Stream

First, log in to the AWS console using the credentials provided on the Hands-on Lab page. From the services list, select Kinesis.

From the Kinesis console, click **Go to Firehose**, then click **Create Delivery Stream**. Select *Amazon S3* as the destination, and set its name to *linuxacademy-courses*. Select *New Bucket* from the **S3 bucket** menu, and name it *linuxacademy-courses* followed by a random string to make it globally unique. Click **Create bucket**, then click **Next**.

Next, set the buffer size to *1* and the buffer interval to *60*, and disable error logging. From the **IAM Role** menu, select *Firehose delivery IAM role*. On the next page, select the IAM role that has been configured for this lab (it should contain your Linux Academy username). Select *FirehoseDeliveryRole* as the policy name and click **Allow**.

Finally, click **Next** and then **Create Delivery Stream**.

Stream Data from EC2

In a terminal window, connect to the EC2 lab server using the credentials from the Hands-on Lab page.

We'll find a number of scripts and files for the lab in the *linuxacademy* user's home directory:

```
[linuxacademy@ip] ls -lrt
```

Begin streaming data to the Firehose delivery stream:

```
[linuxacademy@ip] python write-to-kinesis-firehose.py
```

Note that this script has been specifically configured to use a stream named *linuxacademy-courses*.

Kinesis Analytics

Back in the browser, select **Analytics** from the menu on the left of the Kinesis console. Click **Create new application**.

We'll provide a name of *la-courses* and leave the description empty, then click **Save and Continue**.

On the next page, click the **Connect to a source** button. From the menu, select the *linuxacademy-courses* Firehose delivery stream, and for its permissions, choose *Select from IAM roles that Kinesis Analytics can consume*. From the IAM role menu, select the existing IAM role that has been preconfigured for this lab.

Below, we'll see a table with a sample of the streaming data. However, there are some additional configuration steps needed, so click **Edit schema**. We'll notice that two columns use reserved SQL keywords, so we can change *user* to *user_name*, and *timestamp* to *timestamp_unix*. Click **Save schema and update stream samples**. After a few moments, the table will reload with these new values.

Click **Exit (done)**, then select the application name (*la-courses*) from the top of the page to return to the main console. This time, we'll click **Go to SQL editor**.

At the top of the page, click the **Add SQL from templates** button. From the menu on the template page, select *Approximate top-K items* and click **Add this SQL to the editor**.

In the SQL editor, modify the following line:

```
'course', -- name of column in single quotes
```

This allows us to monitor the top courses by checking occurrences of the course names. Change the next line as well:

```
5, -- number of top items
```

Finally, click **Save and run SQL**. Once the query finishes running, select the **Real-time analytics** tab in the bottom pane and then select *DESTINATION_SQL_STREAM*. After a few moments, data will populate showing five rows of the most popular Linux Academy courses. We can leave it open for 60 seconds to watch it update in real time as the stream continues to be processed.

Set Destinations

We can also send our results for further processing or storage. In the bottom pane, select the **Destination** tab and click **Add a destination**. On the next page, click **Go to Kinesis Firehose** and select **Create Delivery Stream**.

From the destination menu, select *Amazon S3* and set the name to *la-courses-top5*. From the S3 bucket dropdown menu, select the existing bucket we created previously. This time, we'll set the prefix as *summary* before clicking **Next**.

Next, set the buffer size to *1* and the buffer interval to *60*. From the IAM role menu, select *Firehose delivery IAM role*. From the next IAM role dropdown menu, select *firehose_delivery_role* and set the policy name to *oneClick_firehose_delivery_role*, then click **Allow**.

Click **Next** and then **Create Delivery Stream**.

Return to the **Destinations** screen within the Analytics console, and refresh the page. The new destination should now show up, and we can choose it by clicking **Select a stream** near the top of the page.

From the menu, select the *la-courses-top5* Firehose stream we just created. Set the output format to *CSV* and select the permission option *Select from IAM roles that Kinesis Analytics can assume*. Finally, select the existing IAM role from the menu. Click **Save and continue**.

The results of the analytics process are now being dumped to the S3 bucket we specified.

Review

In this lab, we created a stream of data from an EC2 instance to a Firehose delivery stream, set up real-time SQL querying capabilities, and learned how to direct the results to a destination of our choosing. The Kinesis suite of services allows for simple but powerful analysis of large datasets in real time, and is incredibly useful when handling big data.

Congratulations! You've completed the lab on analyzing internet data with Kinesis!