Linux Academy

Hands-on Lab

Kinesis Streams and Firehose

Contents

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 alloted amount of time finishes

Related Courses

AWS Certified Big Data Specialty

Related Videos

Kinesis Overview

Kinesis Streams

Kinesis Firehose

Need Help?

Linux Academy Community

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

In this lab, we'll learn about Kinesis Streams and Firehose to capture and process streams for big data.

To get started, click the **Start Lab** button on the Hands-on Lab page. Once the lab builds, log in to the AWS console with the credentials provided on the Hands-On Lab page.

Kinesis Firehose

In our first scenario, we'll set up Kinesis Firehose to stream sample data from an EC2 instance to an S3 bucket.

Create a Stream

Once logged into AWS, search for **Kinesis**. From the Kinesis Console, click **Go to the Firehose console**. In the Firehose Console, click **Create Delivery Stream**.

Under the **Destination** dropdown menu, select *Amazon S3*. We must also provide a name for the stream. In this lab, we'll use the name *linuxacademy-courses*.

From the **S3 bucket** dropdown, select *New S3 bucket*. For the purposes of the lab, give it the name *la-courses* followed by a random string of characters. Click **Create Bucket**, then click **Next** on the delivery stream page.

On the next screen, set the **Buffer size** to *1* and the **Buffer interval** to *60*.

We can leave the compression and encryption settings off for this lab.

Under the **Error Logging** section, select *Disable*.

In the **IAM Role** section, choose *Firehose delivery IAM role* from the dropdown menu.

The next page will allow us to configure the role. From the **IAM Role** menu, select the role that's been created for the lab. From the **Policy Name** menu, select *FirehoseDeliveryRole*. Finally, click the **Allow** button.

We'll be returned to the Firehose Console, where we can click **Next**. After reviewing the configuration, click **Create Delivery Stream**.

Start the Stream

While the stream is being created, return to the Linux Academy Hands-on Lab page and copy the IP address for the lab server. In a terminal, connect to the server via SSH using the provided credentials.

Once connected, we'll see that some files have already been created for the lab:

```
[ec2-user@ip] ls -lrt
total 1104
-rwxrwxrwx 1 linuxacademy linuxacademy     914 May 19 21:29 read-
kinesis-stream.py
-rwxrwxrwx 1 linuxacademy linuxacademy    1975 May 19 21:29 write-to-
kinesis-firehose.py
-rwxrwxrwx 1 linuxacademy linuxacademy 1106157 May 19 21:29 apache.log
-rwxrwxrwx 1 linuxacademy linuxacademy     297 May 19 21:29 firehose-
agent.json
-rwxrwxrwx 1 linuxacademy linuxacademy     898 May 19 21:29 firehose-
and-streams-agent.json
-rwxrwxrwx 1 linuxacademy linuxacademy     479 May 19 21:29 firehose-
transform-agent.json
```

Among these, we'll see a script called `write-to-kinesis-firehose.py`. We can execute this script to start sending data to Firehose:

```
[ec2-user@ip] python write-to-kinesis-firehose.py
```

Return to the browser and click on the name of the S3 bucket associated with the stream we created. Refresh the bucket until a folder with the current year as a name appears, and click the folder name. The subdirectories will be named for the current month, day, and hour, respectively. Continue navigating through each directory until one or more files are shown.

Click the file name, then click **Open**. The file will download and we can open it in a text editor to view its contents. We should see a JSON data object that includes courses, videos, users, and timestamps.

Monitoring

Return to the Firehose Console and click on the name of the stream. Select the **Monitoring** tab at the top of the screen. Here, we'll see a summary of various data about our stream.

Update Configuration

Next, click the **Details** tab at the top of the page, then click **Edit**. Here, we'll see the configuration options that we set when we created our stream.

From the **S3 Compression** dropdown menu, select *GZIP*. From the **S3 Encryption** menu, select *aws/s3*. Follow the same process as above to configure the **IAM Role**. Finally, click **Save** and then **Confirm**.

The stream's settings have now been updated with no downtime.

Streaming Apache Logs

Next, we'll walk through a scenario you may encounter in the real world: streaming large log files from an

Apache web server and transforming them into a new format.

Configure Delivery Stream

Return to the Firehose Console and click **Create Delivery Stream**. For the **Destination**, select *Amazon S3*, and enter *firehose-1* as the **Delivery stream name**.

From the **S3 bucket** dropdown menu, select *New S3 bucket*. We'll call this one *apachelogs* followed by a random string to make it unique. Click **Create Bucket**, then **Next**.

We'll use the same settings for this stream as our previous one. Data transformation should be disabled. The **Buffer Size** should be set to *1* and the **Buffer interval** should be set to *60*. Compression and encryption should be left as their default values, and error logging should be disabled.

From the **IAM role** dropdown menu, select *Firehose delivery IAM role*. On the **Role Summary** page, select the IAM role that has been created for the lab, and select *FirehoseDeliveryRole* for its policy name. Click **Allow**, then **Next**, then **Create Delivery Stream**.

Install and Configure Kinesis Agent

We should still be logged into our EC2 instance lab server. While the stream is being created, let's take another look at the files in our home directory on the server:

```
[ec2-user@ip] ls -lrt
total 1104
-rwxrwxrwx 1 linuxacademy linuxacademy     914 May 19 21:29 read-
kinesis-stream.py
-rwxrwxrwx 1 linuxacademy linuxacademy    1975 May 19 21:29 write-to-
kinesis-firehose.py
-rwxrwxrwx 1 linuxacademy linuxacademy 1106157 May 19 21:29 apache.log
-rwxrwxrwx 1 linuxacademy linuxacademy     297 May 19 21:29 firehose-
agent.json
-rwxrwxrwx 1 linuxacademy linuxacademy     898 May 19 21:29 firehose-
and-streams-agent.json
-rwxrwxrwx 1 linuxacademy linuxacademy     479 May 19 21:29 firehose-
transform-agent.json
```

This time, we'll be looking at the `apache.log` file, which will be used as a pattern for the data we're processing:

```
[ec2-user@ip] tail apache.log
```

To process the logs, we'll be using the Kinesis agent. We can install it using `yum`:

```
[ec2-user@ip] sudo yum install aws-kinesis-agent
```

Next, we'll configure the Kinesis agent. The configuration file can be found at `/etc/aws-kinesis/agent.json`. The lab server has also been provisioned with a sample configuration file that we can use, `firehose-agent.json`.

For the purposes of this lab, we'll use these settings for the Kinesis agent:

```
[ec2-user@ip] sudo cp firehose-agent.json /etc/aws-kinesis/agent.json
```

With these settings in place, we can start the service:

```
[ec2-user@ip] sudo service aws-kinesis-agent start
```

View Logs

Now that the Kinesis agent has started, we can view the Apache logs that we've streamed to our S3.

Return to the S3 Console, and select the bucket we created with our stream. We may need to refresh a few times in order to see the new directories created by the stream. Click through the subdirectories until an Apache log file appears. Click **Open** to download the file, and open it in a text editor. The file will be an Apache log, streamed to the bucket by our Firehose stream.

Transform Data

On the lab server, we have another agent configuration, `firehose-transform-agent.json`. This file contains options that will allow us to transform the Apache log data.

```
[ec2-user@ip] sudo cp firehose-transform-agent.json /etc/aws-kinesis/
agent.json
```

To put these changes into effect, restart the service:

```
[ec2-user@ip] sudo service aws-kinesis-agent restart
```

Now if we return to the S3 Console and check the most recent file in the bucket, we'll see the log data in JSON format.

Kinesis Stream

In the final scenario, we'll use a KCL application to read from our Kinesis stream.

Create a Stream

Return to the Kinesis Console in the browser and click **Go to the Streams console**. On the Streams Console, click **Create Kinesis Stream**.

Enter a **Stream name**. For the purposes of the lab, we can call it *kinesis-1*. For **Number of shards**, enter *1* and click **Create Kinesis stream**.

Configure Kinesis Agent

On the lab server, we'll see another Kinesis agent configuration file, `firehose-and-streams-agent.json`. This configuration will allow us to send the same data to multiple streams:

```
[ec2-user@ip] sudo cp firehose-and-streams-agent.json /etc/aws-kinesis/
agent.json
[ec2-user@ip] sudo service aws-kinesis restart
```

Read the Kinesis Stream

Finally, we'll run the `read-kinesis-stream.py` script to read from our newly created Kinesis Stream:

```
[ec2-user@ip] python read-kinesis-stream.py
```

In the standard output, we'll see the realtime results of the Stream in JSON format.

Review

In this lab, we learned how to use Kinesis Firehose to deliver data streams to an S3 bucket, as well as how we can apply this operation to real world data like Apache log files. We also used Kinesis Streams to read a stream of data with a custom KCL application.

Congratulations! You've completed the lab on Kinesis Streams and Firehose!