**Joining, Enriching, and Transforming Streaming Data with Amazon Kinesis**

**Introduction**

In this lab, students will set up a Kinesis Data Stream to process streaming orders from customers actively making online orders. The streaming data will then be joined with data hosted in a DynamoDB table to enrich the data. Once the data is enriched, it will be delivered to another Kinesis Data Stream to be further processed by Kinesis Data Analytics. Using Kinesis Data Analytics, students will filter results and store them into S3 using Kinesis Data Firehose.

**Lab Scenario**

You work for an organization that has a mobile application that allows users to place grocery store orders. It’s your job to create a streaming application that helps rewards users who spend more than $100 on grocery orders. In this lab, you’ll create a streaming application that uses Kinesis Stream, Lambda, Kinesis Data Analytics, and Kinesis Firehose to ingest, enrich, filter, and store users’ orders into an S3 data lake.

**Solution**

Log into the AWS Management Console using the credentials provided for the lab. Make sure you're using the *us-east-1* region. You should also log into the Kinesis Live application using the credentials provided for the lab.

**Create a Kinesis Data Stream**

1. From the AWS Management Console, navigate to Kinesis.
2. Create a data stream for incoming orders:
   * On the right, select **Kinesis Data Streams** and click **Create data stream**.
   * In the *Data stream name* field, enter a name for the data stream (e.g. incoming-orders).
   * In the *Number of open shards* field, enter 1.
   * Click **Create data stream**.
3. Create a second data stream for enriched orders:
   * Select **Data streams**, then click **Create data stream**.
   * In the *Data stream name* field, enter a name for the data stream (e.g. enriched-orders).
   * In the *Number of open shards* field, enter 1.
   * Click **Create data stream**.

**Create a Lambda Function to Enrich Records**

**Create a Lambda Function**

1. Navigate to **Lambda** and click **Create function**.
2. Create the function:
   * In the *Function name* field, enter a name for the function (e.g. enrich-orders-function).
   * In the *Runtime* field, use the dropdown to select the latest version of Python (Python 3.8).
   * Expand the *Change default execution role* section and select **Create a new role with basic Lambda permissions**.
   * Click **Create function**.
3. Set the function permissions:
   * After the function is created, select the **Permissions** tab.
   * In a new browser tab, open the *Execution role* link. This opens the function's IAM settings.
   * Click **Attach policies**.
   * Search for the AmazonKinesisFullAccess policy and select it to grant the role full access to Kinesis.
   * Search for the AmazonDynamoDBFullAccess policy and select it to grant the role full access to DynamoDB.
   * Click **Attach policy**.
   * Close out of the IAM browser and navigate back to Lambda.
4. Add a trigger that triggers the Lambda function when orders are put onto the Kinesis Data Stream:
   * Select the **Configuration** tab and click **Add trigger**.
   * Use the search bar to search for and select Kinesis.
   * Use the *Kinesis stream* dropdown to select the incoming-orders data stream.
   * In the *Batch size* field, enter 10. This allows the Lambda function to trigger more often for the lab so you can more easily see data as it comes in.
   * Leave all other defaults and click **Add**. Now, any time incoming traffic comes into the Kinesis Data Stream, the Lambda function is triggered.

**Join the Data from the DynamoDB table**

1. From the *Configuration* tab, select enrich-orders-function.
2. Scroll down to the *Function code* to input the function you want to run when events come in from the Kinesis Data Stream.
3. Replace the existing lambda\_function code with the enrich data [code snippet](https://das-c01-data-analytics-specialty.s3.amazonaws.com/Labs/enrich-data-lambda-function.py) provided in the lab resources.
4. Update the <OUTPUT\_STREAM\_NAME> on line 4 to the enriched orders data stream (e.g. enriched-orders). As records are enriched, they will be output onto this output stream.
5. Click **Deploy** to save the Lambda function.

**Start Streaming Data**

**Stream Data**

1. Navigate to the Kinesis Live application using the Kinesis IP credentials provided for the lab.
2. Fill in the streaming details:
   * In the *Kinesis Stream Name* field, enter incoming-orders.
   * In the *AWS Region* field, enter us-east-1.
   * Fill in the *Access Key* and *Secret Access Key* fields using the credentials provided for the lab.
3. Click **Start Streaming Data** and let the stream run.
4. Navigate back to Kinesis in the AWS Management Console to verify that you can see incoming data in the Kinesis Data Stream.
5. Select the streams link in the *Data Streams* section.
6. Select the incoming-orders data stream, then select the **Monitoring** tab to view the stream metrics. You may need to adjust the time frame (for example, to 1h) to see the data more clearly.
7. In the sidebar menu, select **Data streams**.
8. Select the enriched-orders data stream to verify you have incoming records, then select the **Monitoring** tab to view the stream metrics. You can see that records are coming in for the *Incoming data*, which means orders are being placed, streamed through the Kinesis Data Stream, joined with data from DynamoDB, and the results are output onto the enriched data stream.

**View Data Joined from DynamoDB**

1. To see the data being joined from DynamoDB, navigate to CloudWatch.
2. In the sidebar menu, select **Logs**, then click **View log groups**.
3. Select the /aws/lambda/enrich-orders-function log group.
4. Select the log stream to view the records being output onto the Kinesis Data Stream and orders being output to the enriched data stream.

**Filter Streaming Data with Kinesis Data Analytics**

1. Navigate to Kinesis and use the sidebar menu to select **Analytics applications**.
2. Click **Create application**.
3. In the *Application name* field, enter a name for the application (e.g. filter-top-orders).
4. Click **Create application**, then click **Connect streaming data**.
5. Select **Choose source**, then select **Kinesis data stream** as the source.
6. In the *Kinesis data stream* field, use the search bar to search for and select enriched-orders.
7. Leave all other defaults and click **Discover schema**. You can see the records coming in from the enriched orders, which shows order details like first name, last name, and total cost.
8. After reviewing the data, click **Save and continue**.
9. In the *Real time analytics* section, click **Go to SQL editor**. A dialog box displays and prompts you to run the application.
10. On the dialog box, click **Yes, start application**. This may take 30-90 seconds.
11. Replace the existing comments in the editor with the [SQL query](https://das-c01-data-analytics-specialty.s3.amazonaws.com/Labs/filter-top-orders.sql) provided in the lab resources. This query allows you to filter the enriched records and only return the orders that have a total\_cost of $100 or more.
12. Click **Save and run SQL**. The query runs and displays the results below the editor.

**Create a Kinesis Data Firehose to Transform and Deliver the Final Results**

**Set the Firehose Data Stream Name and Source**

1. Click **Close** below the SQL query results to return to your Kinesis Analytics application.
2. In the *Destination* section, click **Connect to a Destination**.
3. Select **Kinesis Firehose delivery stream**, then click **Create new** to create a Firehose stream. The *Amazon Kinesis Firehose* page opens in a new browser tab.
4. In the *Delivery stream name* field, enter a name for the Firehose stream (e.g. top-orders-delivery-stream).
5. Leave the default *Source* as *Direct PUT or other sources* and click **Next**.
6. In the *Data transformation* field, select **Enabled**.

**Configure the Firehose Data Stream Record Processing**

1. In a new browser, open the AWS Management Console and navigate to *Lambda*.
2. Click **Create function**, then ensure *Author from scratch* is selected.
3. In the *Function name* field, enter a name for the function (e.g add-new-line). This will add a new line to each record to ensure it is in the right format if you want to query it at a later time.
4. Use the *Runtime* dropdown to select the latest version of Python (3.8).
5. Expand the *Change default execution role* section and select **Use an existing role**.
6. Use the *Existing role* dropdown to select the enrich-orders-function role you created earlier.
7. Click **Create function**. The function is created and the function's *Configuration* tab opens.
8. In the *Function code* section, select lambda\_function and replace the existing code with the new line [code snippet](https://das-c01-data-analytics-specialty.s3.amazonaws.com/Labs/new-line-function.py) provided in the lab resources. This function takes the incoming records and adds a new line to the payload before the records are sent to S3.
9. Click **Deploy** and close out of the Lambda tab.
10. Navigate back to the Kinesis Firehose set up and click the **Refresh** icon to the right of the *Lambda function* field.
11. Use the *Lambda function* dropdown to select the add-new-line function you just created.
12. Leave all other defaults and click **Next**.

**Choose a Destination for the Firehose Data Stream**

1. In the *Destination* field, select **Amazon S3**.
2. To the right of the *S3 bucket* field, select **Create new**.
3. Fill in the *Create S3 bucket* pop-up window:
   * In the *S3 bucket name* field, enter a unique name for the bucket (ideally something related to top-orders-bucket).
   * Ensure the *Region* field is set to US East (N. Virginia).
   * Click **Create S3 bucket**. The *S3 bucket* field selects this new bucket automatically.
4. Leave all other defaults and click **Next**.

**Configure and Review the Firehose Data Stream Settings**

1. Change the *Buffer size* to 1 and the *Buffer interval* to 60. These are the minimum required values.
2. In the *S3 compression* field, select **Zip** to zip the files and save space on the data stored in S3.
3. Leave all other defaults and click **Next**.
4. Review the Firehose configuration details, then click **Create delivery stream**.
5. Ensure the Firehose data stream's status is *Active*, then close out of the *Amazon Kinesis Firehose* tab.

**Add the Firehose Data Stream to the Kinesis Analytics Application**

1. Navigate back to the Kinesis Analytics application.
2. Below the *Kinesis Firehose delivery stream* field, click the **Refresh** icon.
3. Use the *Kinesis Firehose delivery stream* dropdown to select the Firehose you just created.
4. Use the *In-application stream name* dropdown to select DESTINATION\_USER\_DATA.
5. Leave all other defaults and click **Save and continue**.

The Kinesis Analytics application is now complete. The application will filter for orders that are $100 or more and then deliver that data to a Kinesis Firehose delivery stream, where it is transformed by adding a new line to each record before finally being zipped and sent to S3.

**Review the Results in the S3 Bucket**

1. In a new browser tab, navigate to S3. After 5-10 minutes, you should see data being populated in the S3 bucket.
2. Select the top-orders bucket and continue selecting the available folder links to view the data results. The results are delivered as .zip files.
3. Download a .zip file to review it:
   * Select a .zip file.
   * Use the *Actions* dropdown to select **Download**.
   * Unzip the file and open it in a text editor to review the results. You can use these results to create an S3 data lake containing information on your top orders.

**Conclusion**

Congratulations — you've completed this hands-on lab!