Building Streaming Data Analytics Solutions on AWS

Lab 2: Streaming Analytics with Amazon Managed Service for Apache Flink

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at *AWS Training and Certification*.

Lab overview

You are a data engineer at AnyCompany Online Shopping. AnyCompany Online Shopping sells a variety of consumer goods through e-commerce transactions in five categories - books, food, apparel, electronics, and home goods.

Your company wants a low latency solution in real-time to analyze web clickstream events from its online customers who interact with its retail website. Your executive team wants to see top sales by category updated in near real time. It wants a solution with no setup costs and without having to manage servers. The team decided to build its solution on Amazon Web Services (AWS) using Amazon Kinesis and Amazon Managed Service for Apache Flink services.

You are responsible for building a real-time streaming analytics pipeline in Managed Apache Flink Studio to meet the requirements. Managed Apache Flink Studio processes data with sub-second latencies from Amazon Kinesis Data Streams and responds to events in real time. It can run Apache Flink applications continuously and scale automatically with no setup cost and without you having to manage servers. It can analyze streaming data interactively using managed Apache Zeppelin notebooks. Zeppelin notebooks provide a complete suite of analytics tools for data visualization, exporting data to files, and controlling the output format for easier analysis.

In this lab, you build a stream processing pipeline by ingesting clickstream data and enriching the clickstream data with catalog data stored in Amazon Simple Storage Service (Amazon S3). You perform analysis on the enriched data to identify the sales per category in real time and visualize the output.

OBJECTIVES

By the end of this lab, you will be able to do the following:

- Build a real-time streaming analytics pipeline in Managed Apache Flink Studio using Apache Flink and Apache Zeppelin to ingest, enrich, and analyze the clickstream data.

- Perform interactive data analytics and visualize using Apache Zeppelin notebooks with Managed Apache Flink Studio.

ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. The following list explains the purpose for each icon:

- **Expected output:** A sample output that you can use to verify the output of a command or edited file.
- **Note:** A hint, tip, or important guidance.
- **Task complete:** A conclusion or summary point in the lab.

Start lab

1. To launch the lab, at the top of the page, choose Start lab.

Caution: You must wait for the provisioned AWS services to be ready before you can continue.

2. To open the lab, choose Open Console.

You are automatically signed in to the AWS Management Console in a new web browser tab.

WARNING: Do not change the Region unless instructed.

COMMON SIGN-IN ERRORS

Error: You must first sign out

If you see the message, **You must first log out before logging into a different AWS account:**

- Choose the **click here** link.
- Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.
- Choose Open Console again.

Error: Choosing Start Lab has no effect

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

- Add the lab domain name to your pop-up or script blocker's allow list or turn it off.
- Refresh the page and try again.

LAB ENVIRONMENT

The following diagram shows the basic architecture of the lab environment:

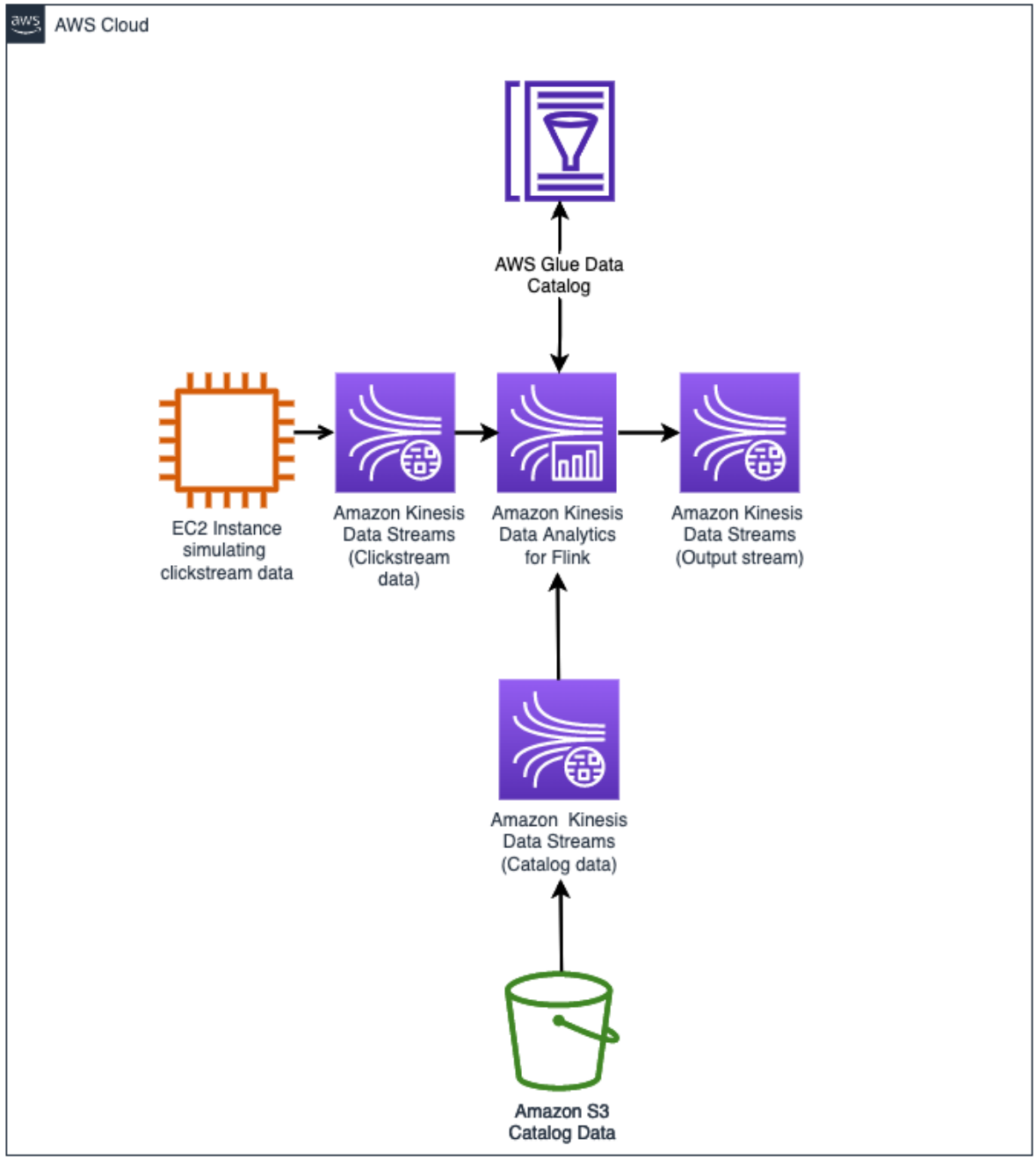


Image: The following list details the major resources in the diagram that have been pre-provisioned for this lab:

- An **Amazon Elastic Compute Cloud (Amazon EC2)instance** that functions as a clickstream generator.
- Three **Kinesis data streams**.
    - A data stream to ingest clickstream data

    - A data stream to ingest user data from a S3 bucket
    - A data stream to ingest alert data from an Amazon Managed Service for Apache Flink application
- A **Managed Apache Flink studio** for developing the analytics workload.
- An **S3 bucket** where catalog data (item_id, item_quantity, item_price, email and page) is stored.

AWS SERVICES NOT USED IN THIS LAB

AWS service capabilities used in this lab are limited to what the lab requires. Expect errors when accessing other services or performing actions beyond those provided in this lab guide.

Task 1: Setting up Zeppelin notebook environment

In this task, you setup a Managed Apache Flink Studio notebook for Amazon Managed Service for Apache Flink to interactively query data streams in real time.

TASK 1.1: INITIALIZE THE KINESIS ANALYTICS APPLICATION

In this step, you navigate to the created Kinesis Analytics application and initialize the application.

3. At the top of the console, in the search bar, search for and choose

   Kinesis. Open the link in a new tab.

Note: You navigate back the management console many times during this lab to launch multiple services. Open (right-click) the new services by choosing Open Link in New Ta.

4. In the left navigation pane, choose **Managed Apache Flink**.
5. In the left navigation pane, choose **Studio notebooks**.
6. Choose the option button next to the Studio notebook name starting with **KdaStudio-** and, at the top of the page, choose **Run**.
7. To confirm, choose **Run**.

Note:

- You can't perform any operations on the application while it is starting.
- It can take up to 5 minutes for the application to get to the *Running* state. Notice that the Managed Apache Flink Studio notebook status changes to *Starting*.

While waiting for the application to get to the *Running* state, download Zeppelin notebook (represented as a .zpln file called Lab2_Kinesis_Analytics.zpln). You open this file in Managed Apache Flink Studio.

TASK 1.2: DOWNLOAD THE ZEPPELIN FILE FROM AMAZON S3

8. Save the Lab2_Kinesis_Analytics.zpln file to your local machine.

Task complete: You have successfully setup the Zeppelin notebook environment.

Task 2: Connect to the Amazon EC2 producer and start the clickstream generator

In this task, you connect to a custom producer running on an EC2 instance using Session Manager, a capability of AWS Systems Manager, and follow the steps to start the clickstream generator.

Imagine owning a website or a web app. Users find their way to your site and perform various actions. Visiting a page, clicking a button, submitting information, and purchasing a product are examples of such activities. The users could also be using different devices when performing these operations. The simulator tries to imitate this payload structure across five sample categories (food,electronics,apparel,books and home goods).

Note:

- Session Manager provides secure instance management without needing to open inbound ports, maintain bastion hosts, or manage Secure Shell (SSH) keys.
- If the command shell is unresponsive when you type a command, to return to the shell prompt, press *Ctrl+C*.

9. To access **Session Manager**, in the list to the left of these instructions, copy the **CommandHostSessionManagementURL**.
10. To display the **EC2 Producer terminal** window, paste the **URL** into a new browser tab and press **Enter**.

The Kinesis stream name is needed for the clickstream data ingestion. This value is dynamically obtained and set as a variable in the commands below. It displays the name so you can ensure the value is set. Then it initiates the script that ingests the data.

11. **Command:** To start the **clickstream_generator_items.py** script, run the following command:

```
STREAM_NAME=$(aws kinesis list-streams --query "StreamNames[?contains(@,
'ClickstreamDataStream')]" --output text)

echo -e "\n\nThe stream name is : $STREAM_NAME\n\n"

python3 clickstream_generator_items.py $STREAM_NAME 1 1
```

Expected output: Output has been shortened as it just keeps printing.

```
******************************
**** This is OUTPUT ONLY. ****
******************************


The stream name is : LabStack-jfoekb-3o57AGNcJdbHE3CcGfwQ6J-0-
KdsClickstreamData-8VgwXqv9PuRr


 Number of arguments: 4 arguments
```

```
 Argument List: ['clickstream_generator_items.py', 'LabStack-jfoekb-
3o57AGNcJdbHE3CcGfwQ6J-0-KdsClickstreamData-8VgwXqv9PuRr', '1', '1']

 The program name is: clickstream_generator_items.py

 The kinesis data stream name is : LabStack-jfoekb-3o57AGNcJdbHE3CcGfwQ6J-0-
KdsClickstreamData-8VgwXqv9PuRr

 Max interval in seconds between records : 1
The current region is set to us-west-2.
```


```
{"event_id": "6136066c482ddd74a268451d43454187", "event": "reviewed_item",
"user_id": 1, "item_id": 31, "item_quantity": 0, "event_time": "2023-09-27
16:35:26.025579", "os": "ios", "page": "apparel", "url": "www.example.com"}
{"event_id": "8200e635814059ec209e1ca4d41d36e8", "event": "liked_item",
"user_id": 30, "item_id": 13, "item_quantity": 0, "event_time": "2023-09-27
16:35:27.143534", "os": "android", "page": "home", "url": "www.example.com"}
{"event_id": "444837c2003ade27d0fc3fba140901bb", "event": "reviewed_item",
"user_id": 2, "item_id": 52, "item_quantity": 0, "event_time": "2023-09-27
16:35:28.154368", "os": "ios", "page": "electronics", "url":"www.example.com"}
{"event_id": "d027166efbd8107f9c87e35ec6896276", "event": "reviewed_item",
"user_id": 20, "item_id": 13, "item_quantity": 0, "event_time": "2023-09-26
17:38:27.620738", "os": "web", "page": "books", "url": "www.example.com"}
{"event_id": "9e05835423cd2f2ce11794890a16d006", "event": "purchased_item",
"user_id": 36, "item_id": 33, "item_quantity": 4, "event_time": "2023-09-26
17:38:27.631147", "os": "android", "page": "books", "url":"www.example.com"}
{"event_id": "b5e17e03743ac37b04e697e6ab6b6796", "event":
"clicked_item_description", "user_id": 48, "item_id": 23, "item_quantity": 0,
"event_time": "2023-09-26 17:38:28.642577", "os": "web", "page": "books",
"url": "www.example.com"}
{"event_id": "0faf5f8a5931d80b6ff46c9486c4d230", "event":
"clicked_item_description", "user_id": 44, "item_id": 33, "item_quantity": 0,
"event_time": "2023-09-26 17:38:29.652591", "os": "android", "page": "food",
"url": "www.example.com"}
{"event_id": "0433fe4d36d01c3b94523119b632c68f", "event":
"clicked_item_description", "user_id": 34, "item_id": 23, "item_quantity": 0,
"event_time": "2023-09-26 17:38:30.662130", "os": "android", "page": "food",
"url": "www.example.com"}
```


Optional: The response from Kinesis Data Stream can be observed by appending `--verbose` to the end of the command.

Expected output: This is output for one shardId.

```
**************************************************
**** This is the Kinesis response OUTPUT ONLY. ****
**************************************************
```


```
{
 "event_id": "9a75230c6e99eb57c6d5a39595474291",
 "event": "clicked_review",
 "user_id": 25,
 "item_id": 12,
 "item_quantity": 0,
 "event_time": "2023-09-26 17:46:03.970616",
 "os": "web",
```

```
 "page": "electronics",
 "url": "www.example.com"
}
Message written to the Kinesis Data Stream
{
 "ShardId": "shardId-000000000001",
 "SequenceNumber": "49644893285493347387758872903216807650192741278821646354",
 "EncryptionType": "KMS",
 "ResponseMetadata": {
 "RequestId": "f918f3c1-3253-7cfa-a495-e8da46c1bf61",
 "HTTPStatusCode": 200,
 "HTTPHeaders": {
  "x-amzn-requestid": "f918f3c1-3253-7cfa-a495-e8da46c1bf61",
  "x-amz-id-2":
"buizxhEv/aaPHfVa9SuG5sHE4V2Xbl3YDh8/FyIJhzpZpl+eeK/39vOjeCOqLkg6wapDcA+lTjkBd
YpLbd5VVv9KhXJZDA9/",
  "date": "Tue, 26 Sep 2023 17:46:03 GMT",
  "content-type": "application/x-amz-json-1.1",
  "content-length": "133"
 },
 "RetryAttempts": 0
 }
}
shardId-000000000001
SequenceNumber:
49644893285493347387758872903216807650192741278821646354
HTTStatusCode:
200
```


Note:

- Keep the web browser tab open for the duration of this lab running the data generator script.
- If the Session Manager window times out, repeat this task to start producing the clickstream data again.

Task complete: You have successfully simulated clickstream data generation by running the clickstream generator Python script on the EC2 Producer terminal.

Task 3: Import the Zeppelin notebook

12. Navigate back to the Managed Apache Flink Studio tab. Ensure that the Managed Apache Flink Studio notebook status is **Running**.

Note: You might need to refresh the page and navigate back to the **Studio notebooks** page to see the status update.

13. Choose the option button next to the Studio notebook name starting with **KdaStudio-** and choose **Open in Apache Zeppelin**.

Note: You might need to turn off pop-up blockers on your browser.

This will open a web-based notebook that you can use for interactive analytics. All of the existing notes are listed on the left side of the page. You can also create a new note or import a note.

14. Choose **Import Note**.
15. Choose **Select JSON File/IPYNB File** from your local machine, and select **Lab2_Kinesis_Analytics.zpln**.
16. To open the notebook, choose **Lab2_Kinesis_Analytics**.

ZEPPELIN NOTEBOOK EXPLAINED

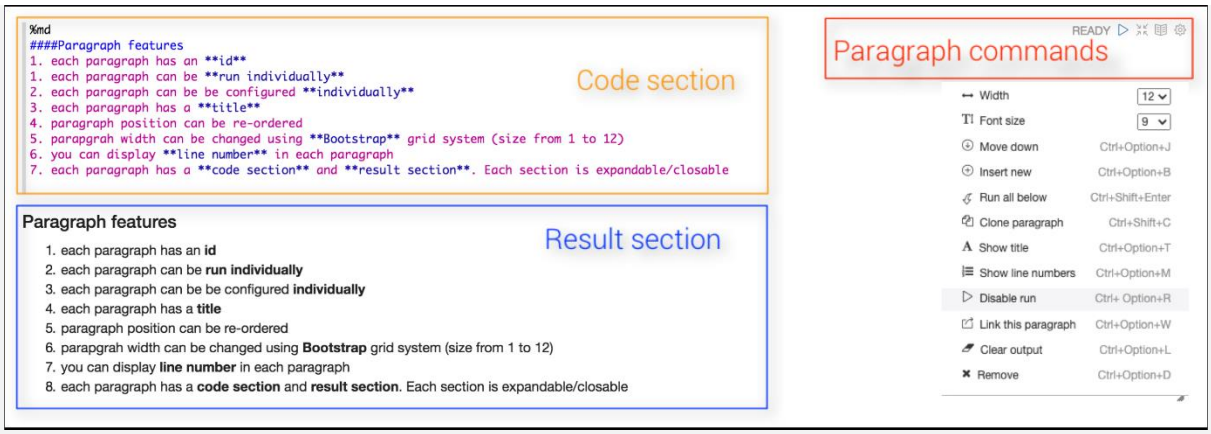
Here is a brief overview of the Zeppelin notebook UI as it will be used in this lab.

Experienced Zeppelin notebook users can skip this and proceed to the next task.

Each Zeppelin note is composed of paragraphs. The note can be viewed as a paragraph container.

Each paragraph consists of the following two sections:

- **Code section** where you put your source code. Locate the code section in your notebook UI.
- **Result section** where you can view the result of the code run. Locate the result section in your notebook UI.
- To the right of these sections, the **Paragraph commands** run the paragraph code. Locate the paragraph commands in your notebook UI.



The image identifies the layout for the Code section, Result section, and the Paragraph commands.

Managed Apache Flink Studio and the Zeppelin notebook environment is now set up for analysis.

Task complete: You have successfully imported the Zeppelin notebook.

Task 4: Analytics development in Managed Apache Flink Studio with the Zeppelin notebook

Now that you have clickstream data flowing through your Kinesis data streams, you configure Managed Apache Flink as a consumer to query and analyze the streaming data as you build the streaming pipeline. You use the Managed Apache Flink Studio notebook to interactively query data

streams in real time as you build the streaming pipeline. You use an Apache Flink interpreter to interact with your streaming data within the Zeppelin notebook environment.

Note:

- Managed Apache Flink Studio notebooks uses notebooks powered by Zeppelin and uses Apache Flink as the stream processing engine.
- Apache Flink is a framework and distributed processing engine for stateful computations over unbounded and bounded data streams. Flink has been designed to run in all common cluster environments and, perform computations at in-memory speed and at any scale.
- Zeppelin provides your Managed Apache Flink Studio notebook with a complete suite of analytics tools for data visualization, exporting data to files and controlling the output format for easier analysis.

17. Follow the instructions in the Zeppelin notebook environment and return for the next optional task or to complete the lab.

Task complete:

Task 5: Understanding in-memory table creation in AWS Glue Data Catalog

In this task, you navigate to AWS Glue Data Catalog to view the tables created and their metadata.

In this lab, you use *AWS Glue* to understand your data assets. AWS Glue is a fully managed ETL service that is designed to make it easier and more cost-effective to categorize your data, clean it, enrich it, and move it reliably between various data stores. AWS Glue automatically stores metadata in a central data catalog. It can create table definitions for many common data stores, including S3 buckets, web logs, and AWS databases.

While performing the analytics development in the Zeppelin notebook, you create many in-memory tables. These tables are stored in *AWS Glue Data Catalog*. You maintain a unified view of your data using the AWS Glue Data Catalog. You can view the Data Catalog to quickly search and discover the datasets that you own, and maintain the relevant metadata in one central repository.

18. Navigate back to the AWS Management Console tab.
19. At the top of the console, in the search bar, search for and choose

    AWS Glue and open link in a new tab.
20. In the left navigation pane, under *Data Catalog*, choose **Tables**.
21. Notice the four tables that were created from Managed Apache Flink Studio Zeppelin notebook.
22. Choose any of the tables with the text link and you will be directed to the page with

    Table details and

    Advanced properties.
23. Choose **Advanced properties**.

The **Table Properties** contain the schema and data type defined in Managed Apache Flink notebook while creating the in-memory table.

You have successfully accessed AWS Glue to view the tables created and their metadata.

Task complete:

Conclusion

You have done the following:

- Built a real-time streaming analytics pipeline with Kinesis data streams in Managed Apache Flink Studio using Apache Flink and Apache Zeppelin by ingesting clickstream data and enriching the clickstream data with catalog data stored in Amazon S3. You performed analysis on the enriched data to identify the sales per category in real time.
- Visualized the analysis.
- Output the data to a Kinesis data stream for further downstream processing depending on operational needs.

End lab

Follow these steps to close the console and end your lab.

24. Return to the **AWS Management Console**.
25. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
26. Choose End lab and then confirm that you want to end your lab.