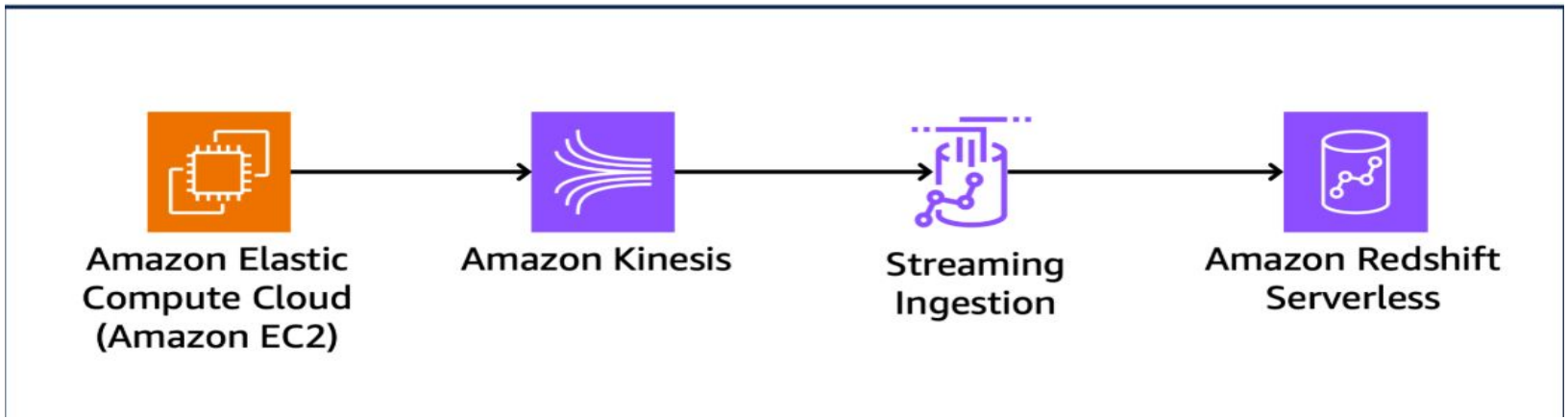
# Populating the Data Warehouse From Amazon Kinesis

streaming data source (Amazon Kinesis) and its structure. Next, you create an external schema and a materialized view with Auto Refresh to populate the Redshift data warehouse. Finally, you query the materialized view.

## Objectives

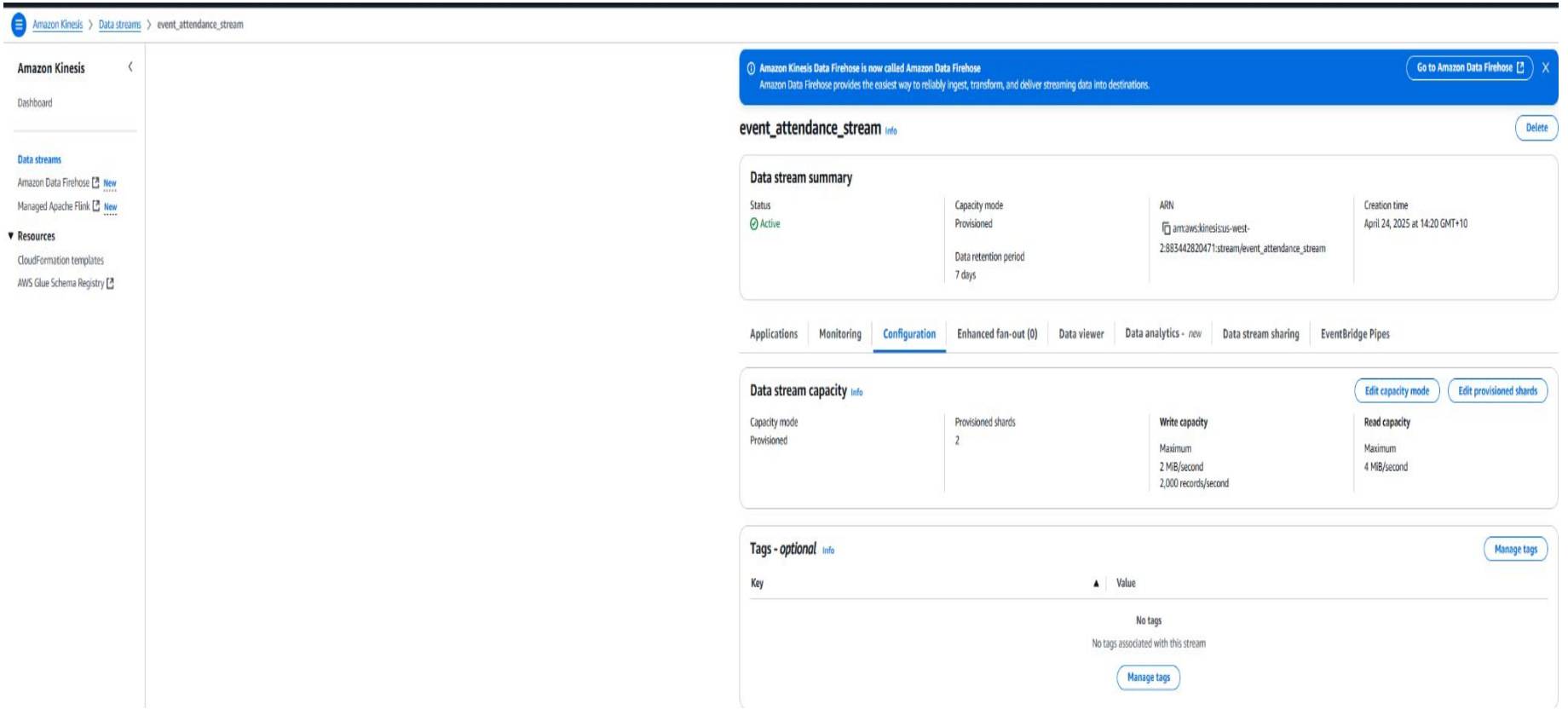
By the end of this lab, you should be able to do the following:

- Create an external schema for streaming data.
- Create a materialized view with Auto Refresh.
- Query the materialized view.



# Amazon Kinesis

Create -Data stream- event_attendance_stream.



policy attached to the Redshift IAM role, which provides permission for communication with the Amazon Kinesis data stream.

```
{
    "Version": "2012-10-17",
    "Statement": [        {
            "Sid": "ReadStream",
            "Effect": "Allow",
            "Action": [
                "kinesis:DescribeStreamSummary",
                "kinesis:GetShardIterator",
                "kinesis:GetRecords",
                "kinesis:DescribeStream"            ],
            "Resource": "arn:aws:kinesis:*:0123456789:stream/event_attendance_stream"        },
        {    "Sid": "ListStream",
            "Effect": "Allow",
            "Action": [   "kinesis:ListStreams",    "kinesis:ListShards"
            ],
            "Resource": "*"        }    ]
}
```

At the top of the AWS Management Console, in the search bar, search for and choose `Amazon Redshift`.

On the Serverless dashboard page, choose **Query data**.

in the new browser tab, you might get the following caution in a red color banner: User information couldn't be retrieved.

To create an account to use Redshift Query Editor V2, on the AWS KMS encryption page, leave all the values at the default settings and choose **Configure account**.
The query editor now opens in a new browser tab.

In the Redshift query editor v2 pane, choose the vertical ellipses (three dots) at Serverless:labworkgrp.

Choose Create connection.

On the Connect to labworkgroup window prompt, choose AWS Secrets Manager.

Select Choose a secret drop-down menu and choose redshift!labnamespace-dbadmin.

Choose **Create connection**.

On the top right, choose the newly created tickitdb database instead of the *dev* database.

To create an external schema to map the data from Kinesis to a schema, run the following query:

```
CREATE EXTERNAL SCHEMA event_kds

FROM KINESIS

    IAM_ROLE default;
```

o list the external schemas that you created in Redshift, including the Kinesis external schema that was created specifically to access data from an Amazon Kinesis data stream, run the following query:

```
    SELECT * FROM SVV_EXTERNAL_SCHEMAS;
```

## Create a materialized view to consume the stream data

In this task, you start the custom producer that runs on an EC2 instance and create a materialized view to consume the stream data.

20. Copy the CommandHostUrlRedShift value found in the left pane of these instructions into a new browser tab to access the command host terminal.

Consider: AnyCompany Events sells online tickets to concerts. Event attendance tracking is one of the keys to optimizing their future plans and remaining flexible on event day. They want to read the attendance information in real-time as part of a powerful data-driven approach to shape the future of their events and deliver exceptional experiences for the event attendees. The simulator tries to imitate the event attendance tracking system (eventid, attendanceid, checkintime) that tracks the attendance at an event in real-time.

Command: To generate the event attendance data, run the following query:

```
cd ~
```

21. `python3 eventattenddatagenerator.py event_attendance_stream`

21. Now, return to the browser tab that was connected to Redshift query editor v2 in Task 2 to create a materialized view.
Note: Streaming ingestion provides low-latency, high-speed ingestion of stream data from Amazon Kinesis Data Streams into an Amazon Redshift Serverless materialized view. A materialized view is the landing area for data read from the stream, which is processed as it arrives. Amazon Redshift Serverless workgroup is the stream consumer.

Command: To create a materialized view that automatically refreshes when the base data changes by selecting and parsing JSON data from the source table, run the following query:
Note: The following view also validates that the data is a valid JSON source.

```
CREATE MATERIALIZED VIEW event_attendance_view AUTO REFRESH YES AS

SELECT approximate_arrival_timestamp,

JSON_PARSE(kinesis_data) as Data

FROM event_kds.event_attendance_stream

      WHERE CAN_JSON_PARSE(kinesis_data);
```

To refresh the view, run the following query:

Note: Refresh the materialized view command invokes Redshift to read from the stream and load data into the materialized view.

```
REFRESH MATERIALIZED VIEW event_attendance_view;
```

To return the top 5 records ordered by the approximate_arrival_timestamp column in descending order, run the following query on the materialized view:

```
select * from event_attendance_view order by approximate_arrival_timestamp desc limit 5;
```

Python script to gefnerte and send the data to Kenisis data stream

import datetime import json import random import boto3 import sys

def get_data():

return {

'attendanceid': random.randint(1,100),

'checkintime': datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S.%f"),

'eventid': random.randint(1,8000)}

def generate(stream_name, kinesis_client):

while True: data = get_data() print(data) kinesis_client.put_record( StreamName=stream_name, Data=json.dumps(data), PartitionKey="partitionkey") if __name__ == '__main__': my_session = boto3.session.Session() my_region = my_session.region_name STREAM_NAME=str(sys.argv[1]) generate(STREAM_NAME, boto3.client('kinesis', my_region))

- Streaming ingestion for Kinesis Data Streams doesn’t require an authentication type. It uses the IAM role defined in the CREATE EXTERNAL SCHEMA statement for making Kinesis Data Streams requests.