Steps-

Create a DynamoDB Table

name- OrderTable

partition key- orderid

Create Table

Mock Generator Script-

To publish the mock data into DynamoDB, here we are using the boto3 library, with which we can connect to AWS service (in this case DynamoDB)

Below lines of code has to be reviewd and change accordingly while creating the project

session = boto3.Session(profile\_name='default', region\_name='ap-south-1') # Change the region name according to the dynamodb table's region

dynamodb = session.resource('dynamodb')# from the boto3 session which AWS resource we want to access, in this case dynamodb

table = dynamodb.Table('OrderTable') # give the name of our dynamodb table's name, in this case 'OrderTable'

table.put\_item(Item=data) # here we are inserting the item in the table. to put the data in the table we must have the authorisation.

Change Data Capture-

As soon as any change(Insert, Delete, Update) happens in our DynamoDB Table we want to capture it. We will use DynamoDB Stream.

And with the help of DynamoDB Stream we can send it to any DownStream system (in our case Kinesis)

Enabling DynamoDB Stream-

Go to our DynamoDB table -> Exports and Streams -> Turn on "DynamoDB stream details" -> Select "New image"

Creating Kinesis Stream -

Create Kinesis Data Stream -> Data stream name - kinesis-sales-data

Creating EventBridge Pipe - to connect DynamoDB Stream to Kinesis Stream

Create EventBridge Pipes -> pipe name - Dynamo-cdc-to-kinesis

Source - DynamoDB Stream -> select DynamoDB Stream name

Target - Kinesis Stream -> select Kinesis Stream name -> Partition key - eventID

Create pipe

After creation change the permission of pipe

add below permissions:

AmazonKinesisFullAccess

AmazonDynamoDBFullAccess

Testing if the EventBridge pipe is working or not

start our mock script

In kinesis Stream - Under Data viewer - in one the shard our data is landing

Creating Kinesis Firehose - data coming from kinesis stream will be collected as a batch of 10-15 and will be dumped into the S3 there we can use athena on top of it.

Kinesis Data FIrehose - Create Firehose Stream

Source : Kinesis Data Stream

Destination : S3

Select our kinesis stream under "Kinesis data stream"

Firehose Stream Name : "Kinesis-to-s3-delivery"

We will be accumulating certain number of records and will prepare a mini batch file. using Lambda we can do transformation on that mini batch file before dumping it into the S3 destination

for this we will be creating a lambda function:

name : stream\_transformation

after creating change the permissions and attach policies

AWSLambdaKinesisExecutionRole

AmazonS3FullAccess

AmazonKinesisFullAccess

AmazonKinesisFirehoseFullAccess

AmazonDynamoDBFullAccess

select "Turn on data transformation" and select the lambda function created above

change "Buffer Interval" to 15sec

Destination- Create S3 bucket "amazon-black-friday-sales-data-projection" and then select the bucket

Under S3 buffer hits:

S3 buffer hints : 5 mb

Buffer interval : 60 secs #how long the buffer should wait to accumulate the records before dumping into the S3

Create Firehose Stream

After Creation we have to update the iam role of it

attach below policies:

AmazonKinesisFirehoseFullAccess

AmazonKinesisFullAccess

AmazonS3FullAccess

AWSLambda\_FullAccess

AWSLambdaKinesisExecutionRole

Run the Mock script and see the magic . Data will be dumped into the S3

Creating Crawler to crawl the S3 Data

crawler name : "amazon-sales-data-crawler"

Source : our destination s3 bucket "s3://amazon-black-friday-sales-data-projection"

as our data is Json we have to create the custom classifier

Create new classifier

Classifier name : json\_classifier

Classifier type and properties : json

json path : $.orderid,$.product\_name, $.quantity, $.price # it will be according the json data

Attach the created classifier to the crawler

Attach the Glue IAM role

Create the glue catalog database -> name : amazon\_sales\_database

Attach the Database

table name prefix : "projection\_"

After Creating the crawler , RUN the crawler

To View the data , go to athena and we can query our data directly from there

With the help of Athena we can do the near real time analysis of our live streaming data