**Project  
TeleMax**

**Project: Data Streaming and Storage using AWS Kinesis and DynamoDB**

**1. Introduction**

In today's fast-paced digital world, businesses rely on real-time data processing to optimize their operations and make informed decisions. This project focuses on creating a data pipeline using AWS Kinesis Data Streams to capture and process streaming data, which will then be stored in an AWS DynamoDB database for future analysis. This implementation will help businesses efficiently manage large-scale data and extract valuable insights.

**2. Motivation**

**Problem Statement:**

The objective of this project is to create a system that captures data from a Kinesis stream and stores it in a DynamoDB database for further processing and analysis.

**Real-World Scenario:**

TELEMAX is a company planning to build networks in rapidly expanding and underserved markets globally. The company specializes in innovative communication hardware that supports cost-effective, high-speed networking solutions. To optimize their operations, TELEMAX needs a real-time NoSQL-based data warehousing solution that continuously processes and analyzes networking data. The company has chosen AWS Cloud services to support their infrastructure and requires a consultation to develop a suitable solution.

**3. Industry Relevance**

The skills utilized in this project are widely applicable in the technology industry. The key technologies involved are:

**1. AWS Console:**

The AWS Management Console is a web application that provides access to various AWS services, including Kinesis, Lambda, and DynamoDB. It is essential for setting up and managing cloud-based resources.

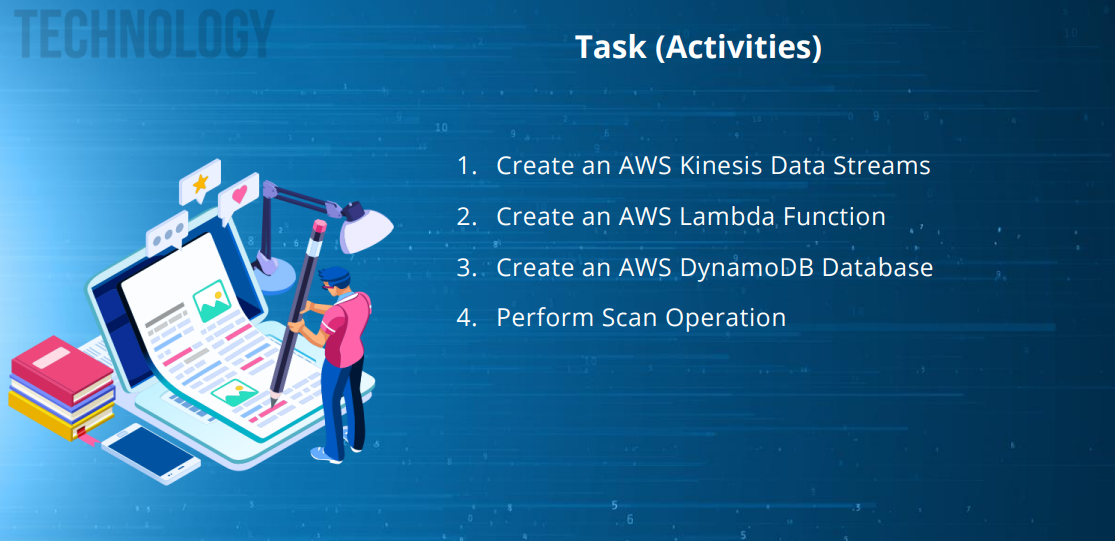
**2. Kinesis Data Streams:**

Amazon Kinesis Data Streams is a scalable, serverless solution for capturing, processing, and storing real-time data streams. It is used by companies to handle large-scale data ingestion from various sources.

**3. AWS Lambda Functions:**

AWS Lambda is a serverless compute service that executes code in response to events. It automates the processing of data from Kinesis and transfers it to DynamoDB without requiring dedicated server management.

**4. Project Tasks**

****

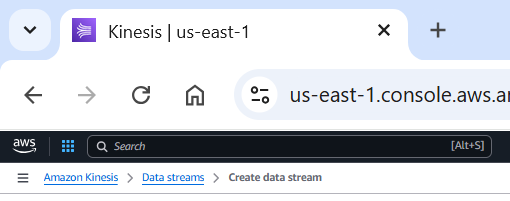
**To successfully implement the solution, the following tasks will be carried out:**

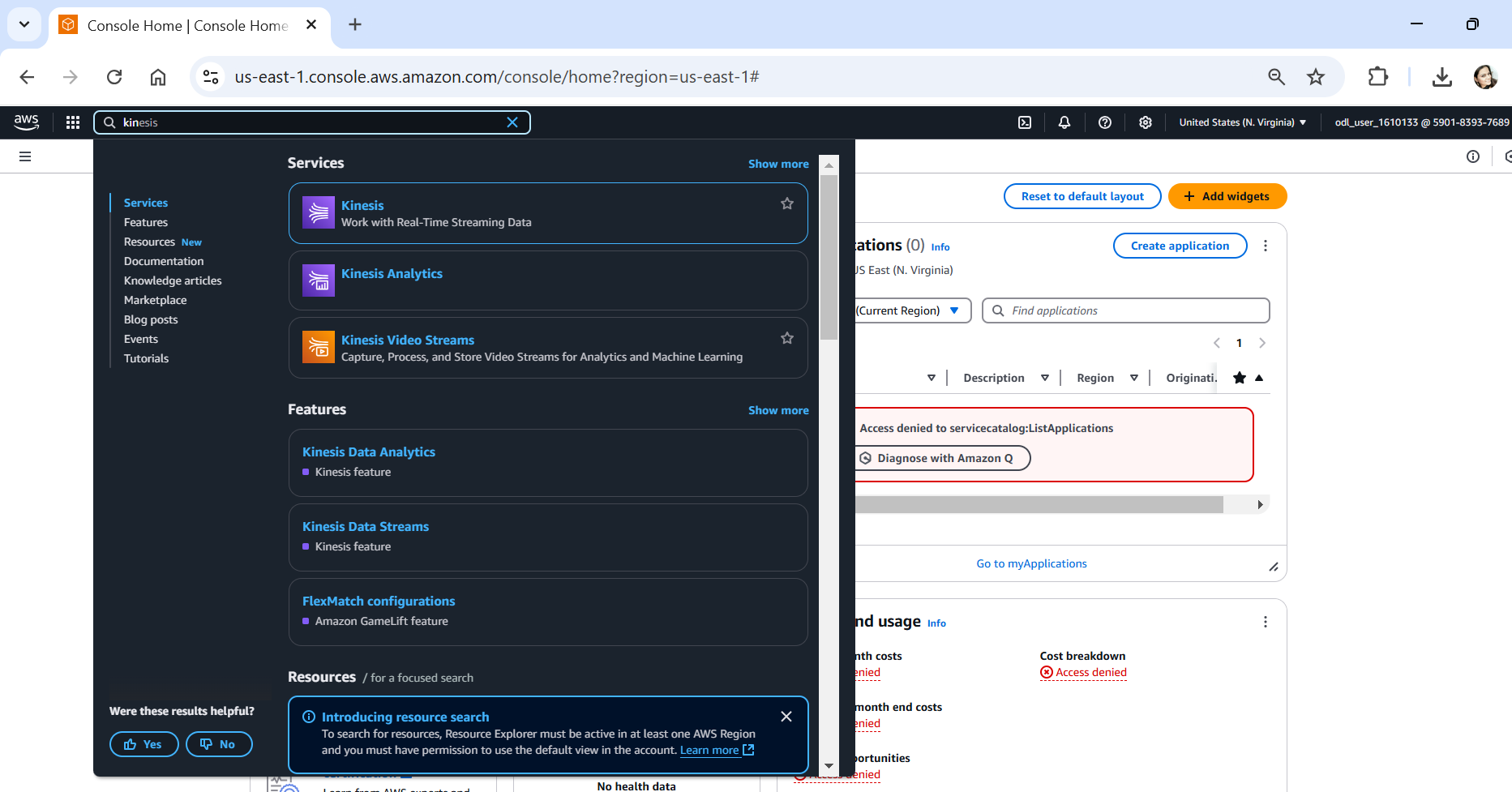
1. **Create an AWS Kinesis Data Stream:**

**Step 1: Set Up a Kinesis Data Stream**

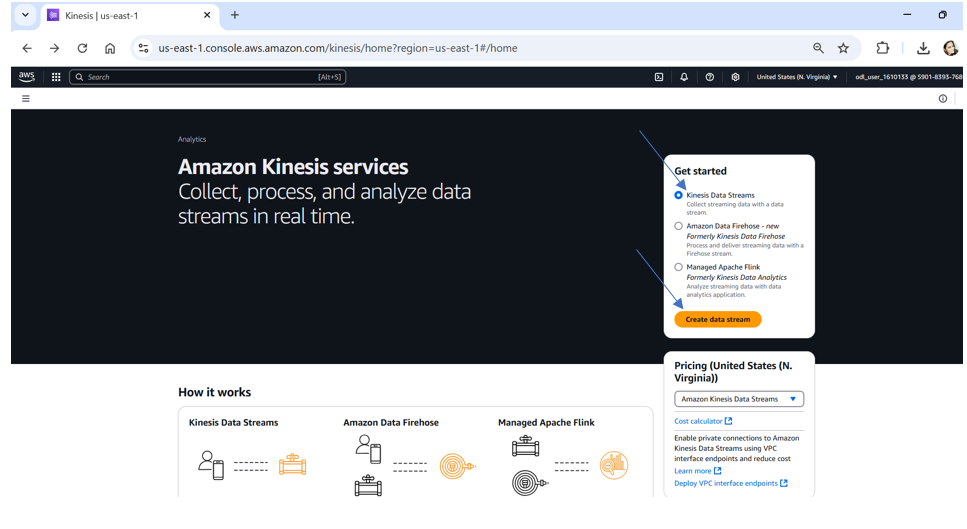
**1.1** Go to the Aws Management Console and log in using the credentials provided in the Practice labs

**1.2** Look up for **Kinesis services** on the management console by searching for services on the search bar in the AWS console and then click on it.

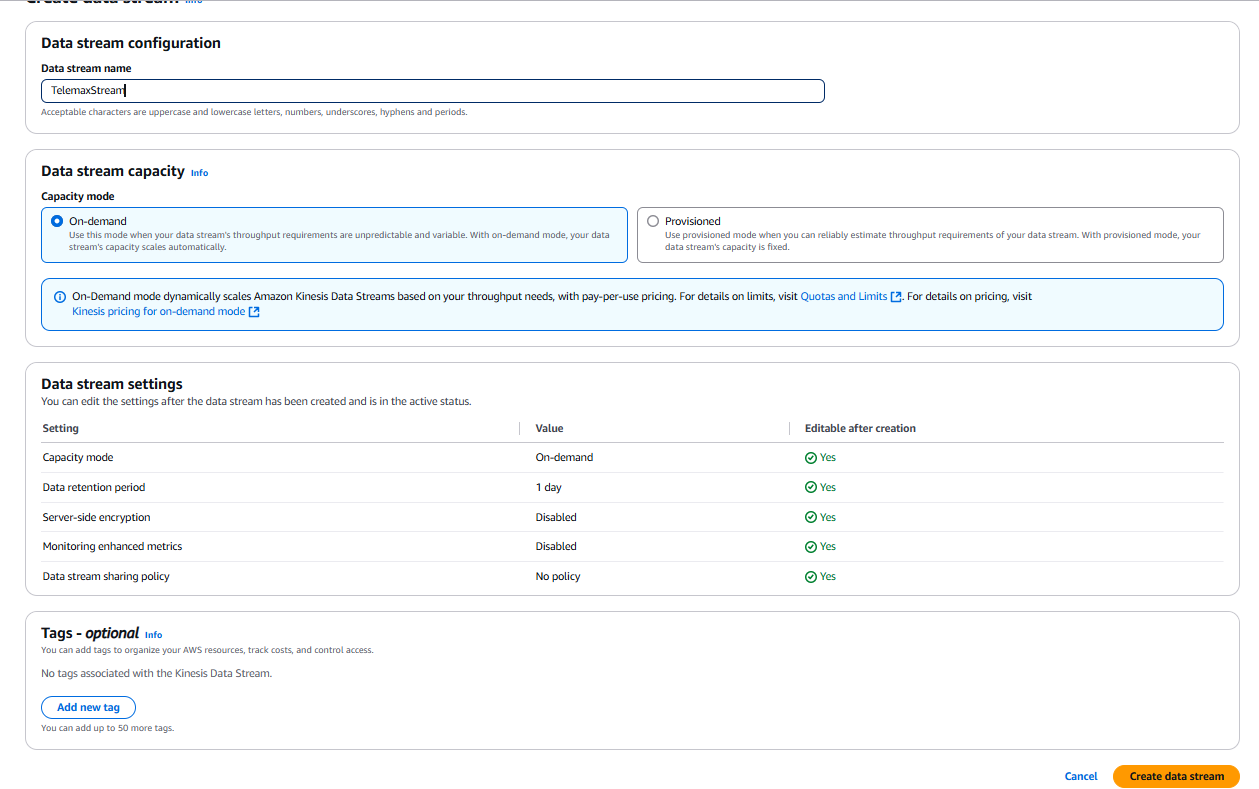
****



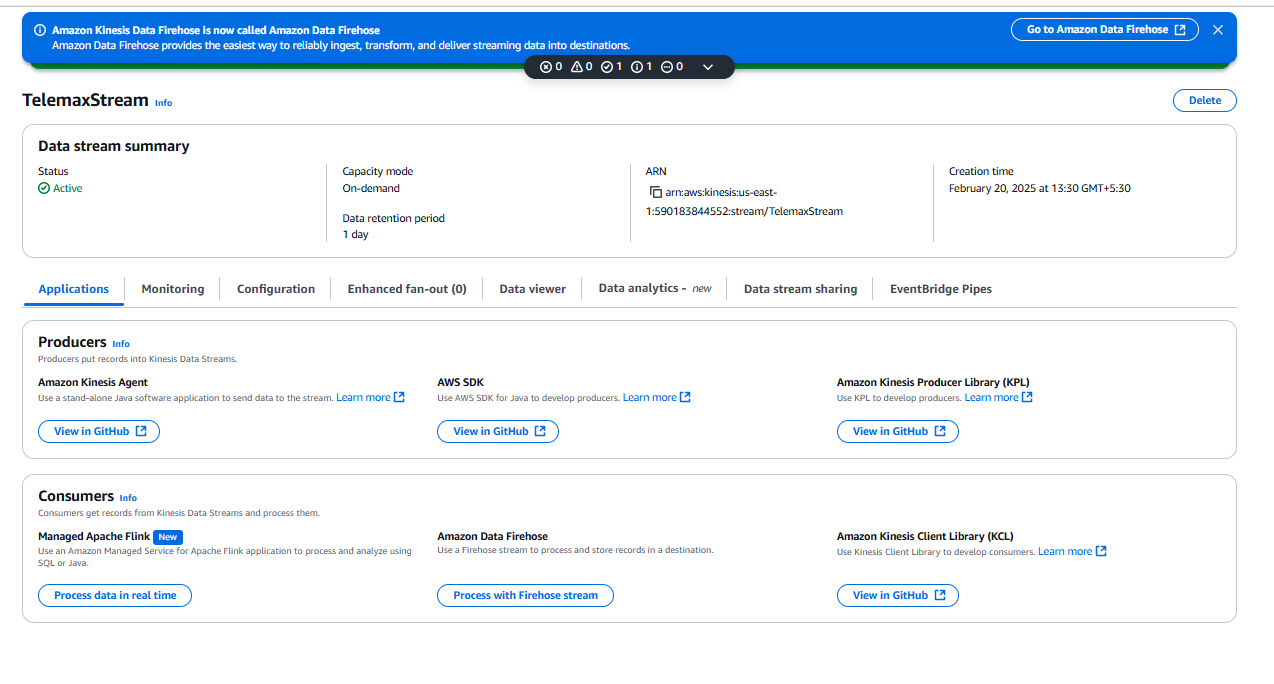
* 1. Once you have selected the Kinesis service it will direct you to **Amazon Kinesis service page**, then click on the **Kinesis Data Stream** and select on **Create Data Streams:**



* 1. Name the data stream as **TelemaxStream** and click on the **create** button to create the stream as shown in the screenshot below:



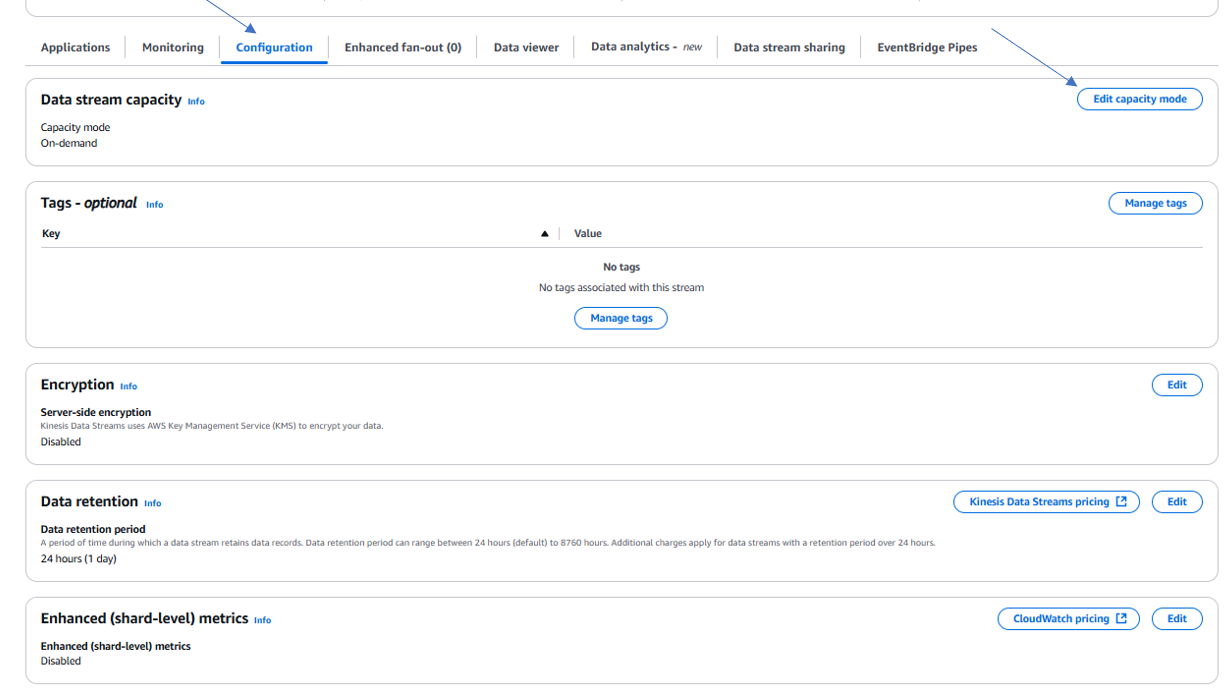
* 1. As the create data stream is clicked it will show the screen as shown in the screenshot below



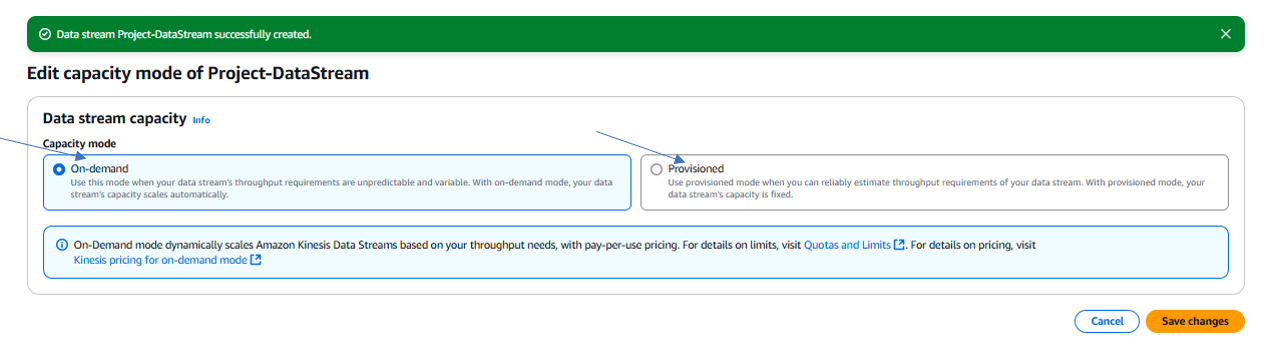
The data stream **TelemaxStream** has been successfully created.



* 1. Now we need to configure the data stream for which we need to go to the configuration tab and click on the edit capacity mode to make changes in the number of shards and capacity mode as shown in the screenshot below:

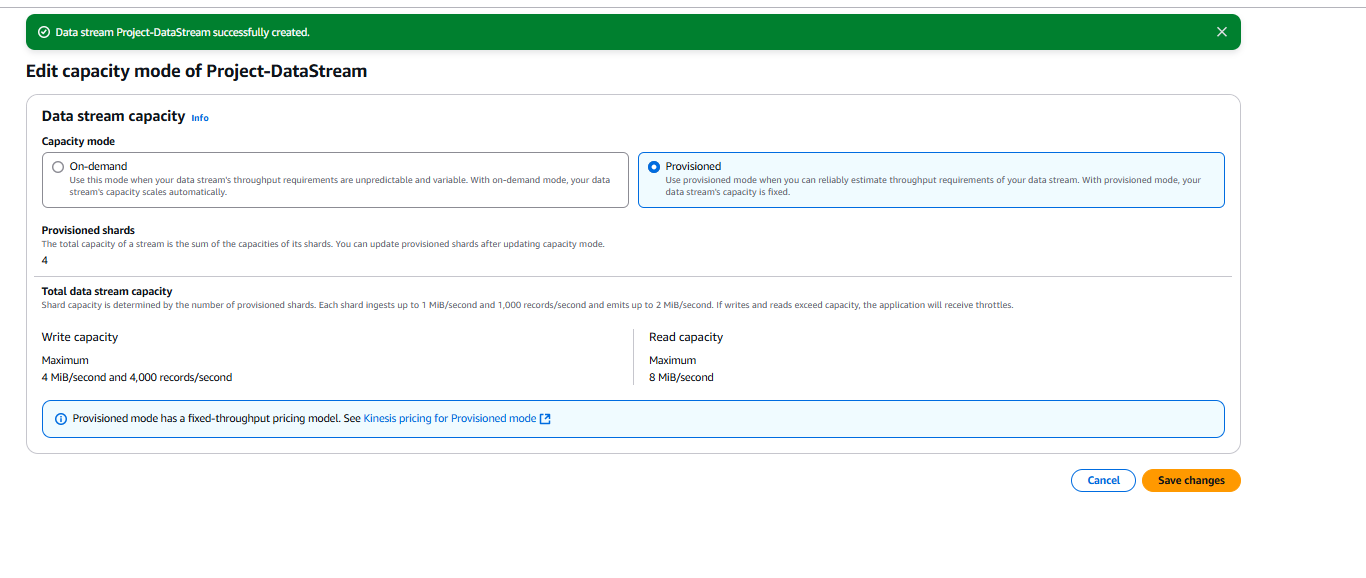


1.8 As you click the edit capacity mode it will take you to the screen below to edit the capacity of Project-Data Stream, the capacity of the data stream could either be set to on-demand or provisioned.

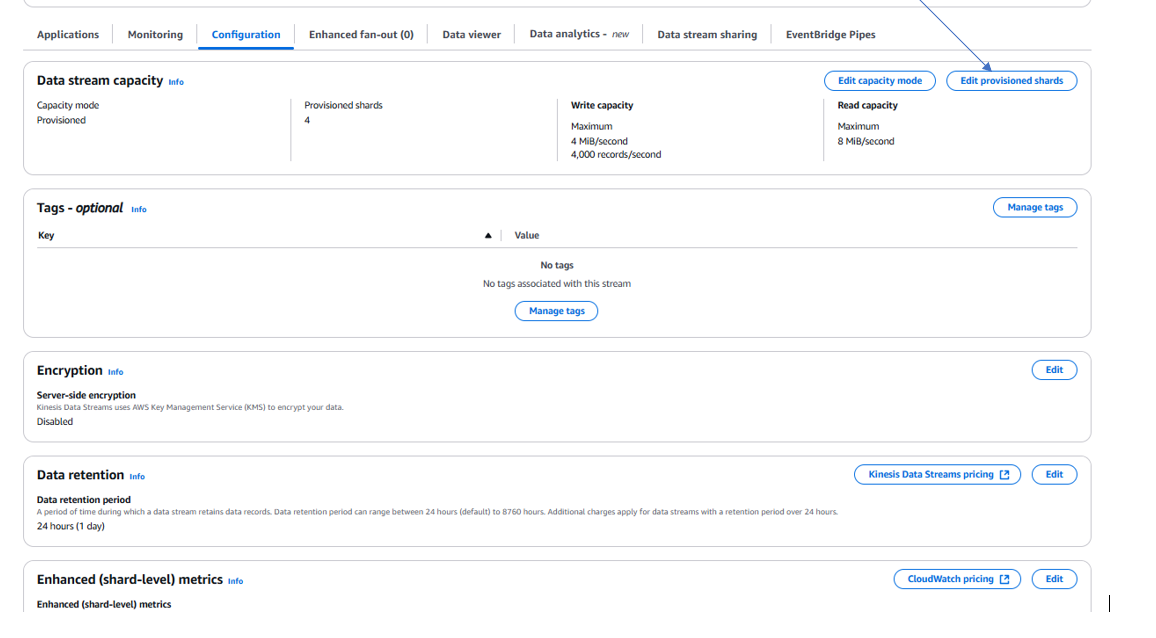
****

1.9 The on-demand capacity stream mode will scale automatically based on demand and the provisioned stream mode will need to be set up manually.

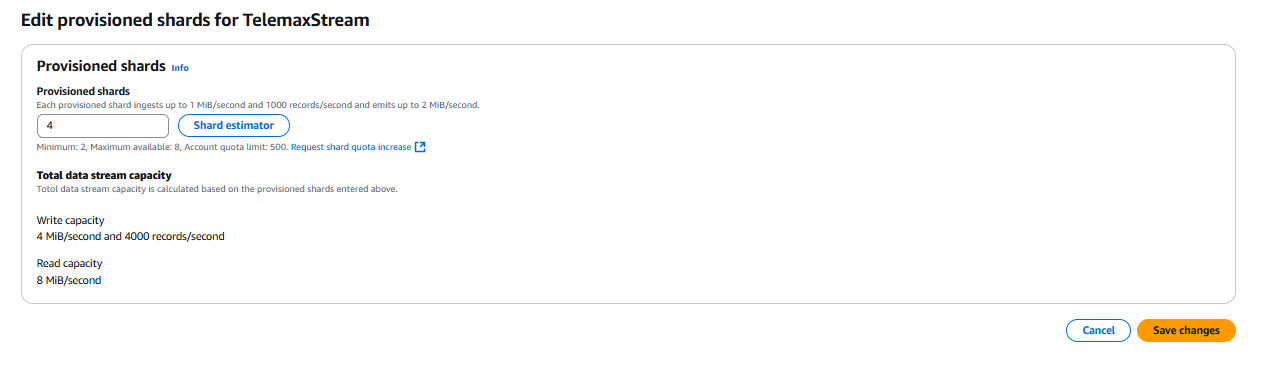
For the project purpose I am using the provisioned mode where I will have to manually enter the number of Provisioned and the number of shards to see the total data stream capacity as shown in the screen shots given below:

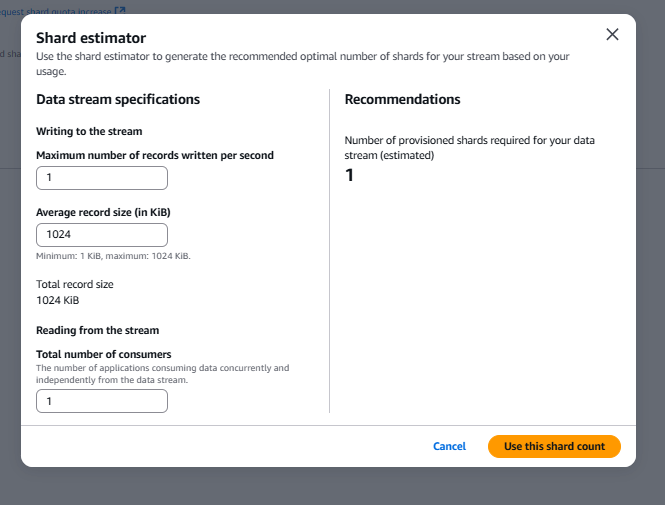
****

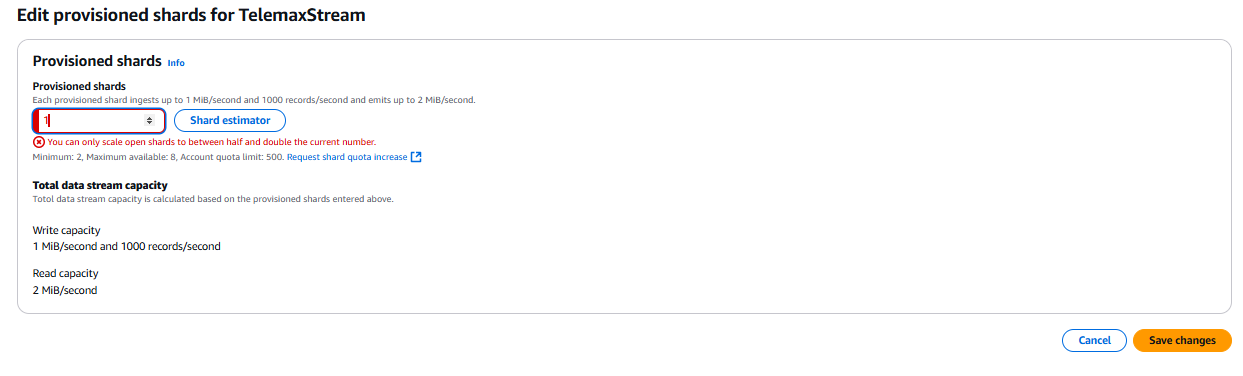
1.10 After the provisioned mode has been selected and saved we need to select the edit provision shards to choose the number of shards

****

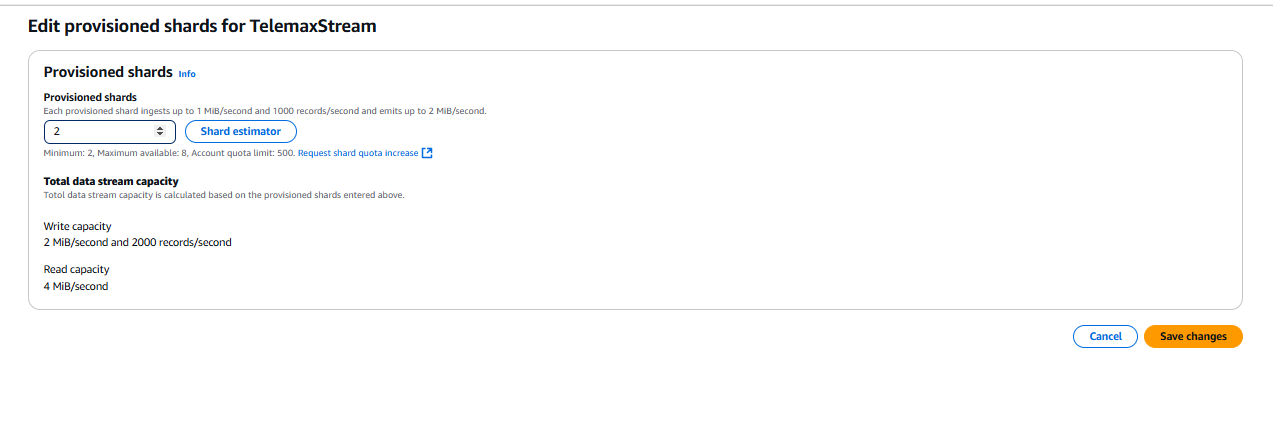
1.11 Now click on the shard estimator as shown in the screenshot below:

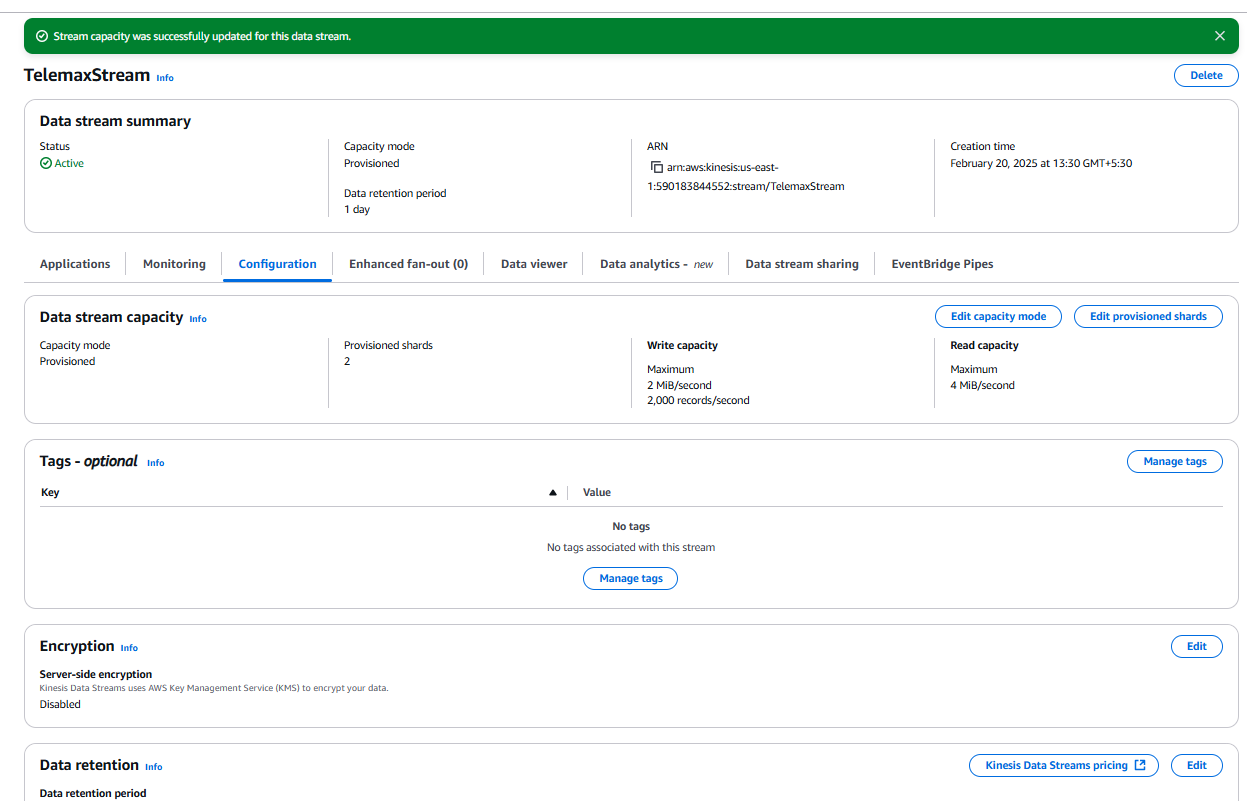






Now, put the provisioned shards accordingly , I have taken 2 shards to begin with

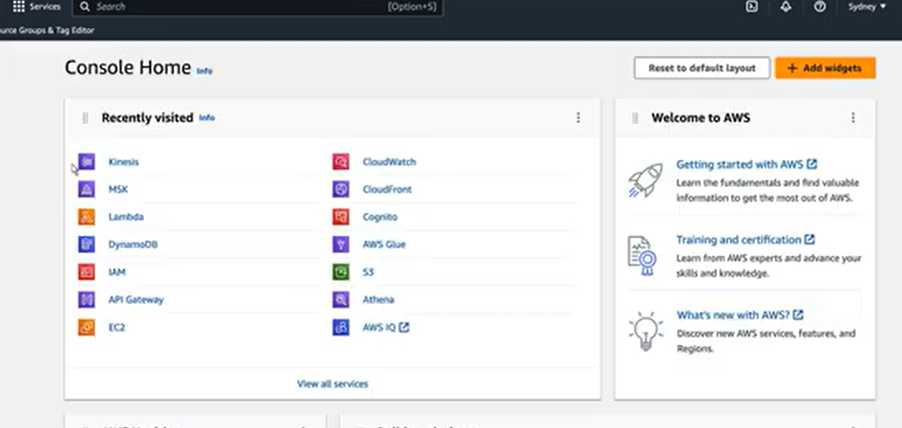
The total capacity of a Kinesis Data Stream is determined by adding up the capacities of its individual shards. By specifying the number of provisioned shards, we can understand how data stream capacity works.

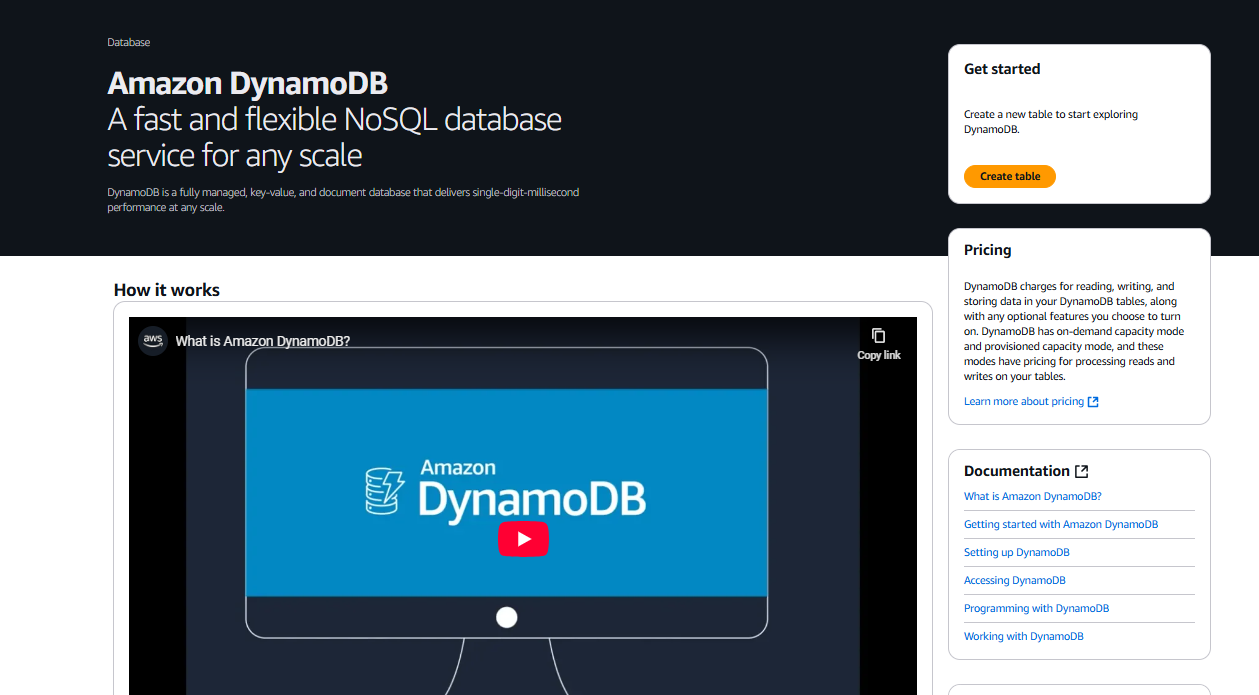


**By following the above steps, we have successfully created the kinesis data stream named TelemaxStream which is ready to use.**

1. **Create an AWS DynamoDB Database(Task3):**

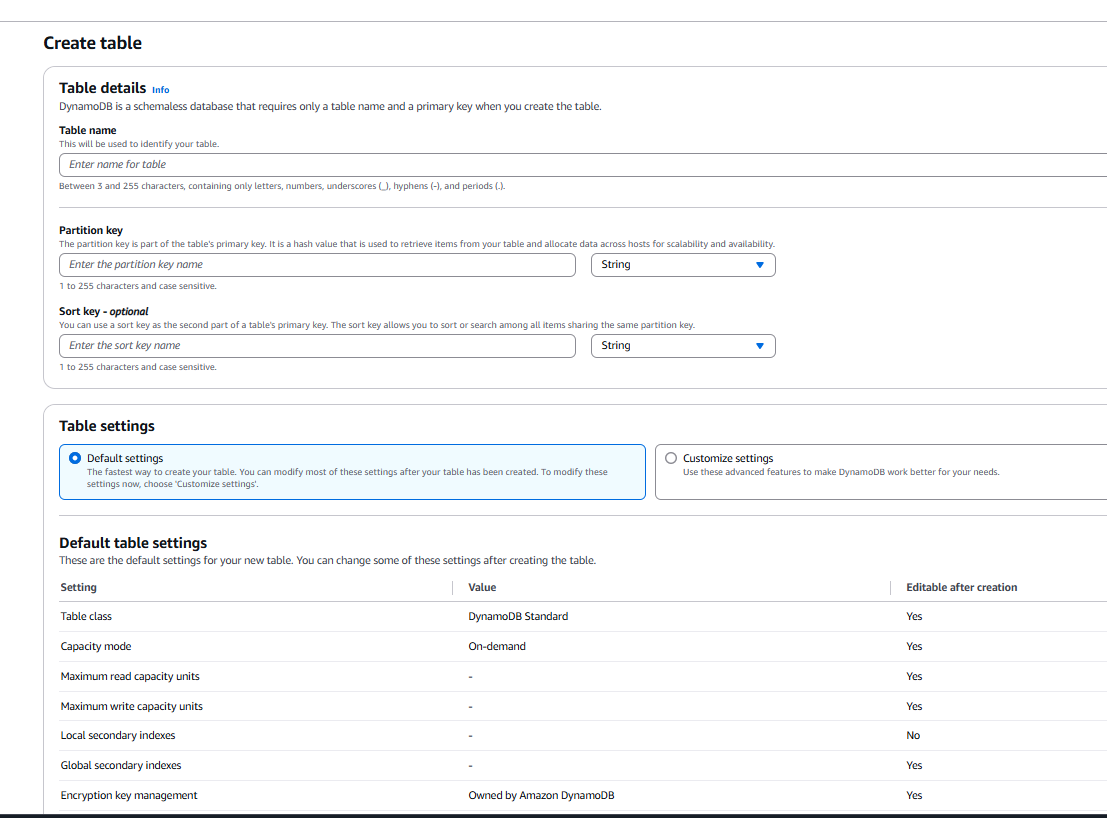
2.1 Go to Dynamo DB in the management console and select the Dynamo Db, the page will redirect to the following screenshot:



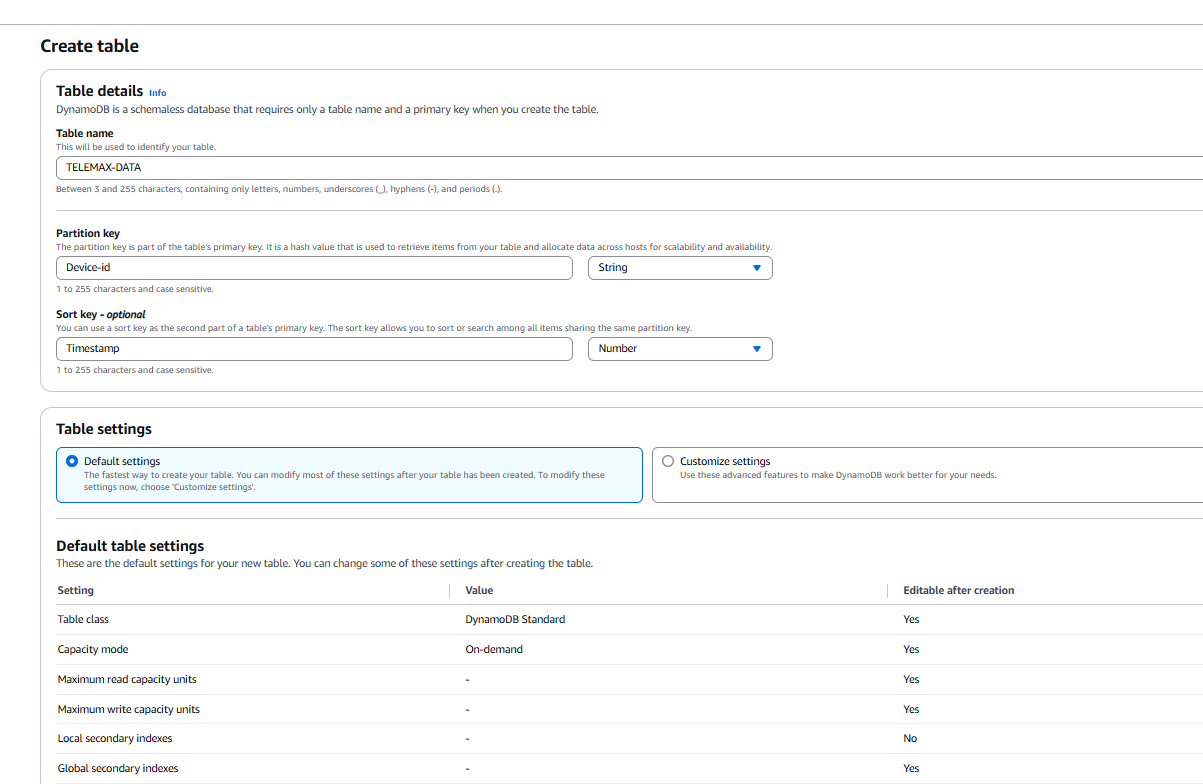


Now click on create table

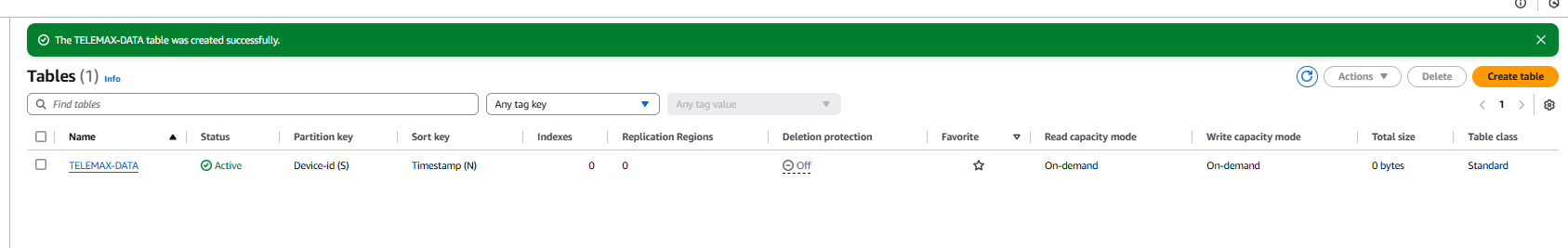
The following screen will appear



Now put in the following details Table name as TELEMAX-DATA, set Primary - Key: Device-id (string) and Sort-Key: Timestamp(number)



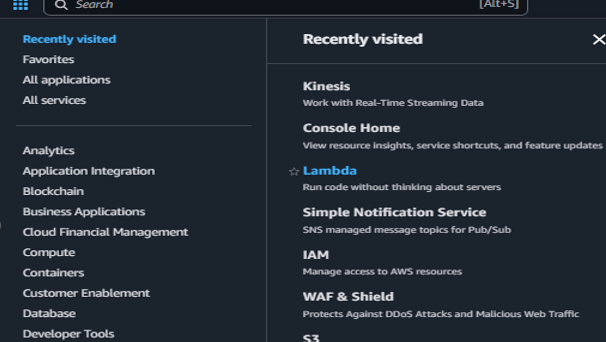
After providing these details create the table

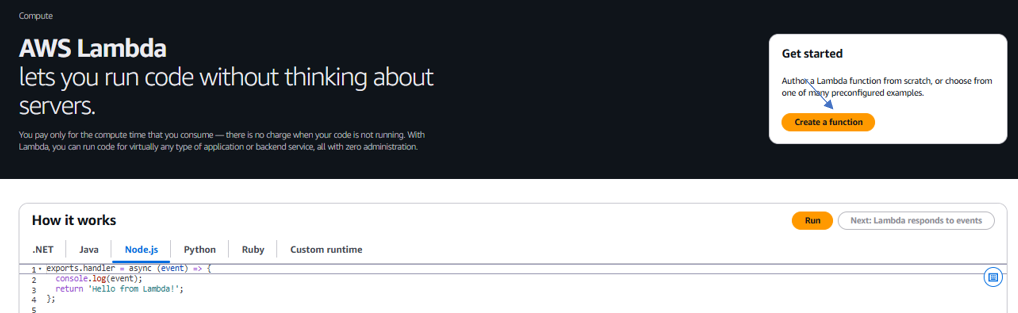


**Create an AWS Lambda Function(Task2):   
Purpose:** AWS Lambda will process incoming records from Kinesis and insert them into **DynamoDB**.

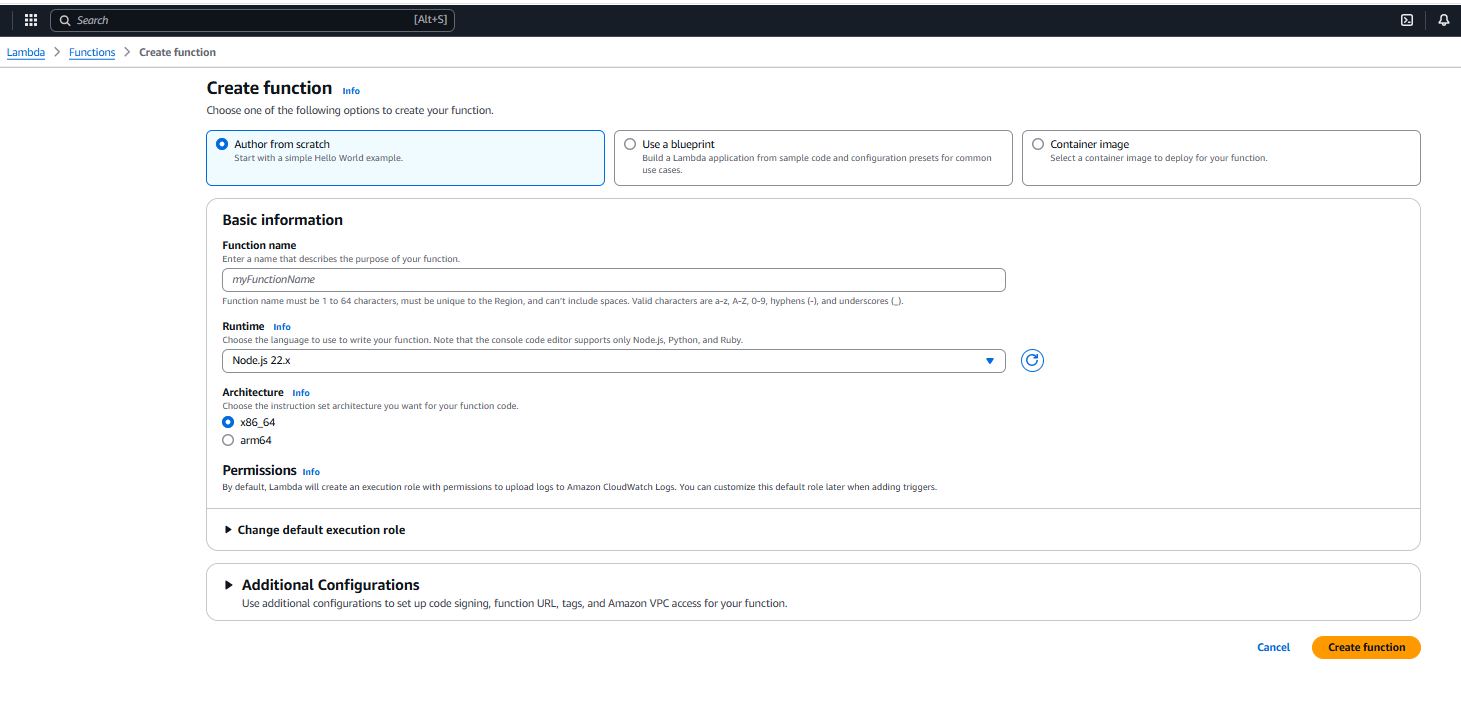
* 1. **Go to AWS Lambda Console** → Click on **Create Function**.

3.2 In the Management console search for Lambda and click on the Lambda service



* 1. Now click on the create function as shown below:  
     

2.3 To create and configure the function follow the steps listed below:

2.4 As we click the create function the following screen will appear  


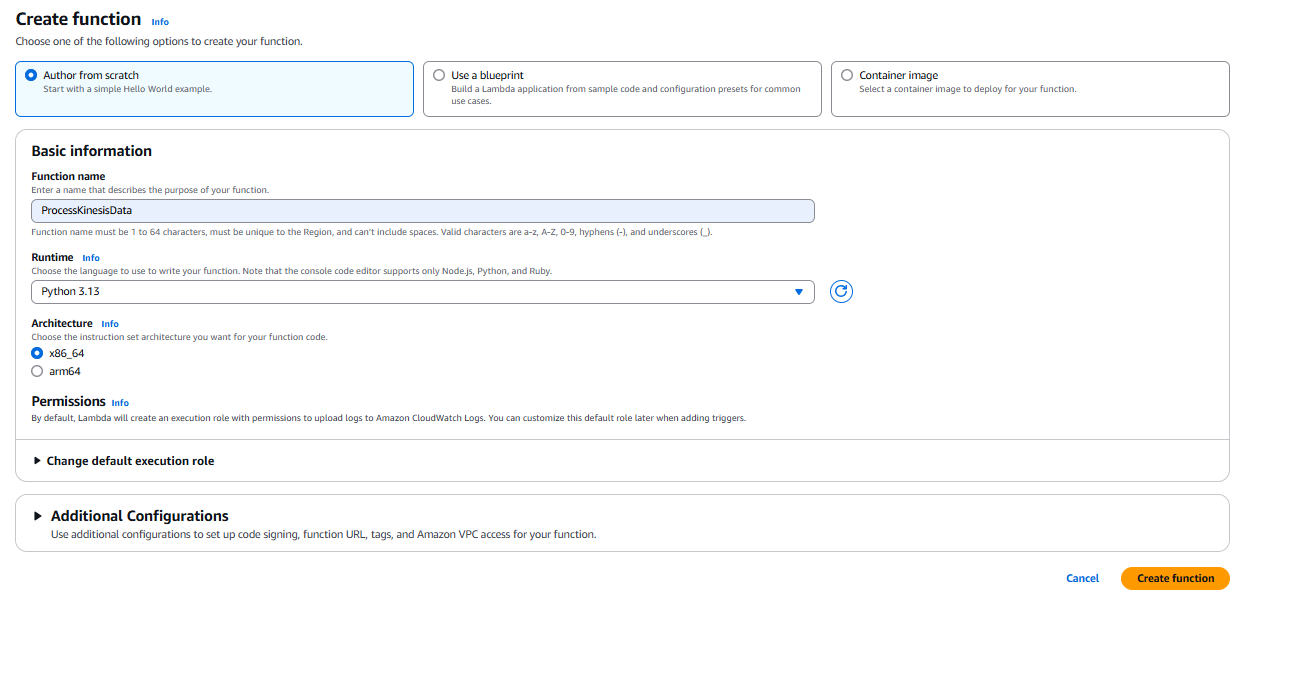
Now put in the following details:

a. Enter function name as **ProcessKinesisData**

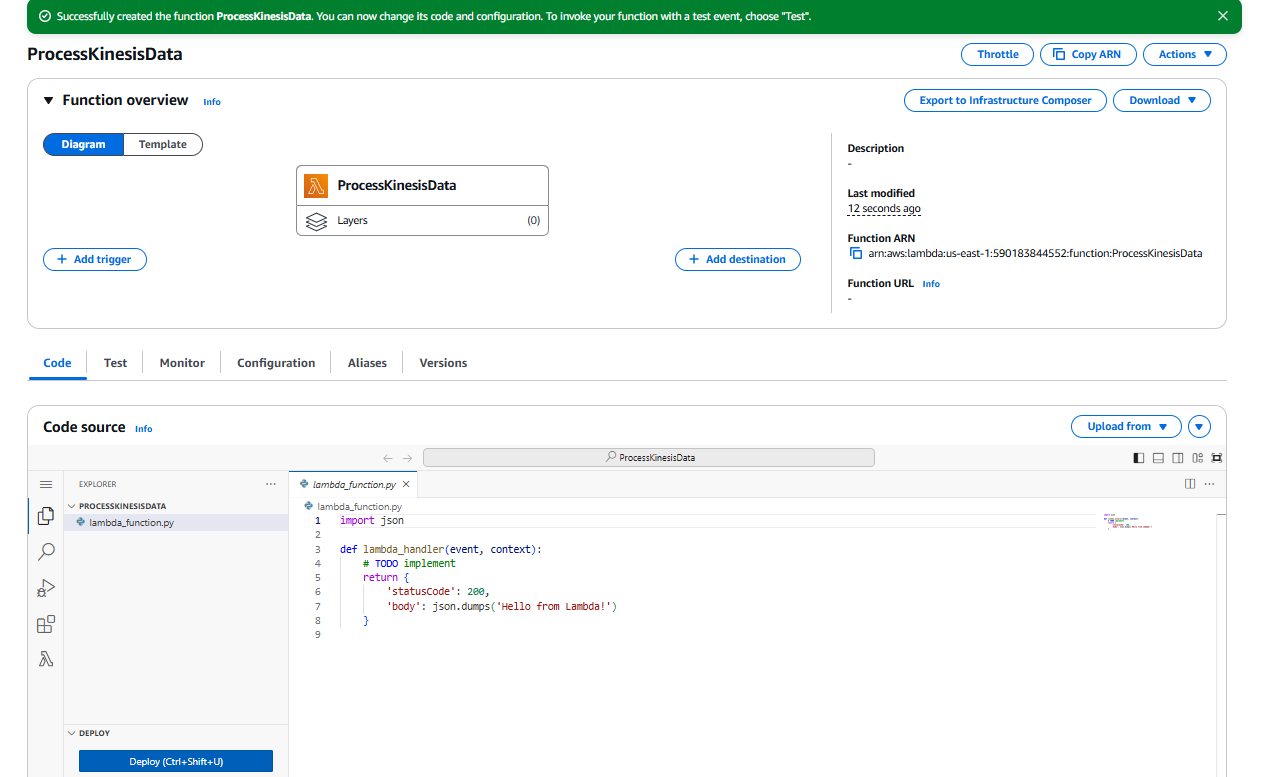
b. Choose an Author from scratch

c. Set **Runtime** to **Python 3.13**

d. Click **Create Function**.



As you create the function, the following screen will appear:



The screen confirms that the Lambda function has been suitably created.

Once the function is created, go to the code section and copy the following code to access the Dynamo db table, as shown in the screen below:

**import json**

**import base64**

**import boto3**

**# Initialize AWS clients**

**dynamodb = boto3.resource('dynamodb')**

**table = dynamodb.Table('Telemax-Data')**

**def lambda\_handler(event, context):**

**for record in event['Records']:**

**# Decode Kinesis data**

**payload = json.loads(base64.b64decode(record['kinesis']['data']).decode("utf-8"))**

**# Extract fields**

**device\_id = payload.get('Device-id')**

**timestamp = payload.get('Timestamp')**

**# Ensure data is valid**

**if not device\_id or not timestamp:**

**print("Invalid data format:", payload)**

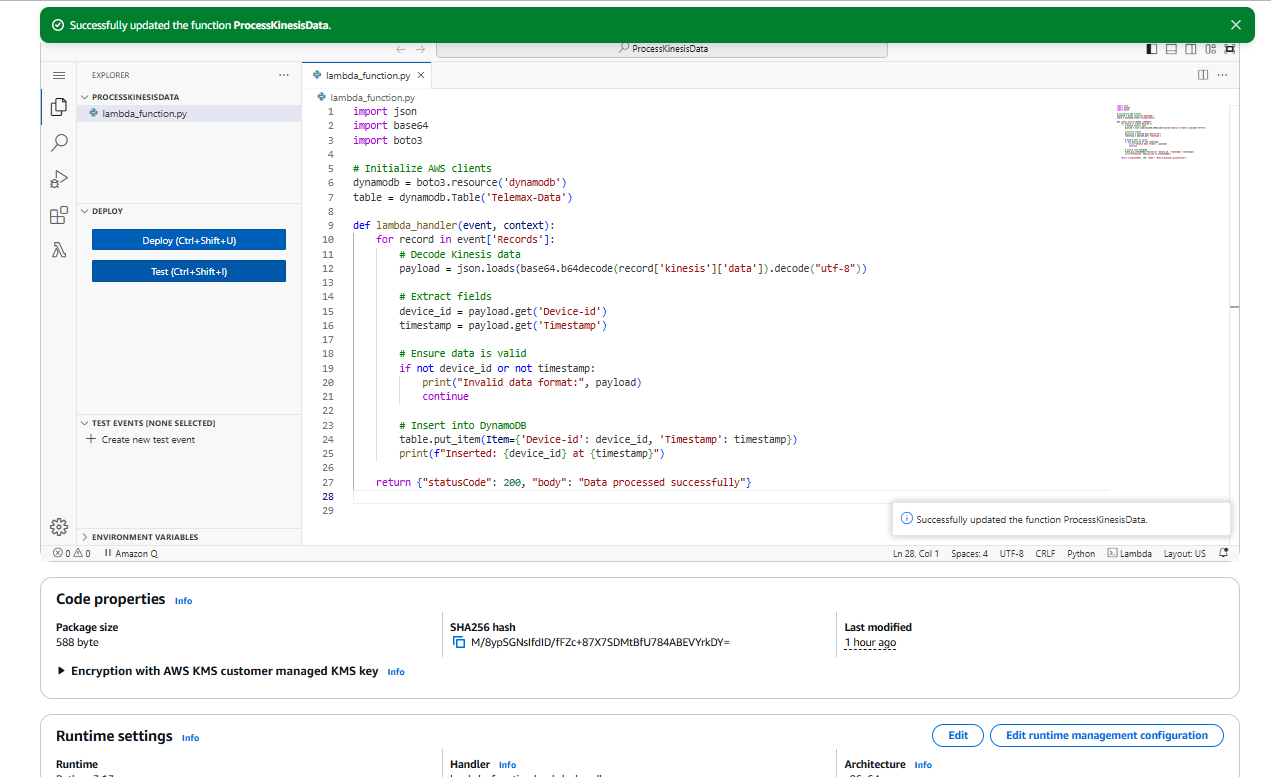
**continue**

**# Insert into DynamoDB**

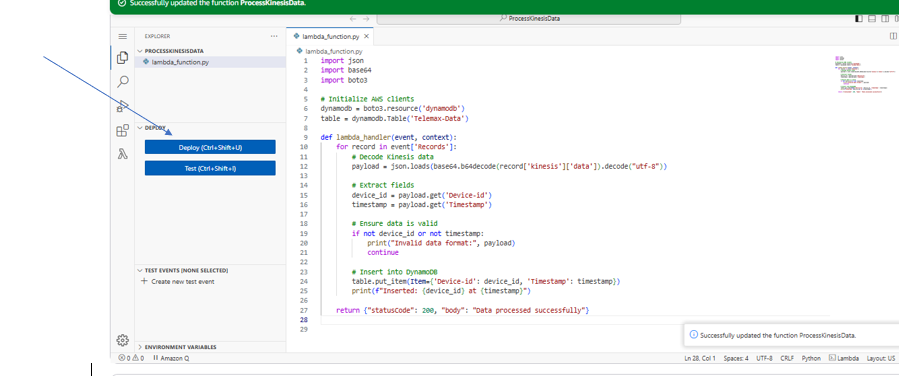
**table.put\_item(Item={'Device-id': device\_id, 'Timestamp': timestamp})**

**print(f"Inserted: {device\_id} at {timestamp}")**

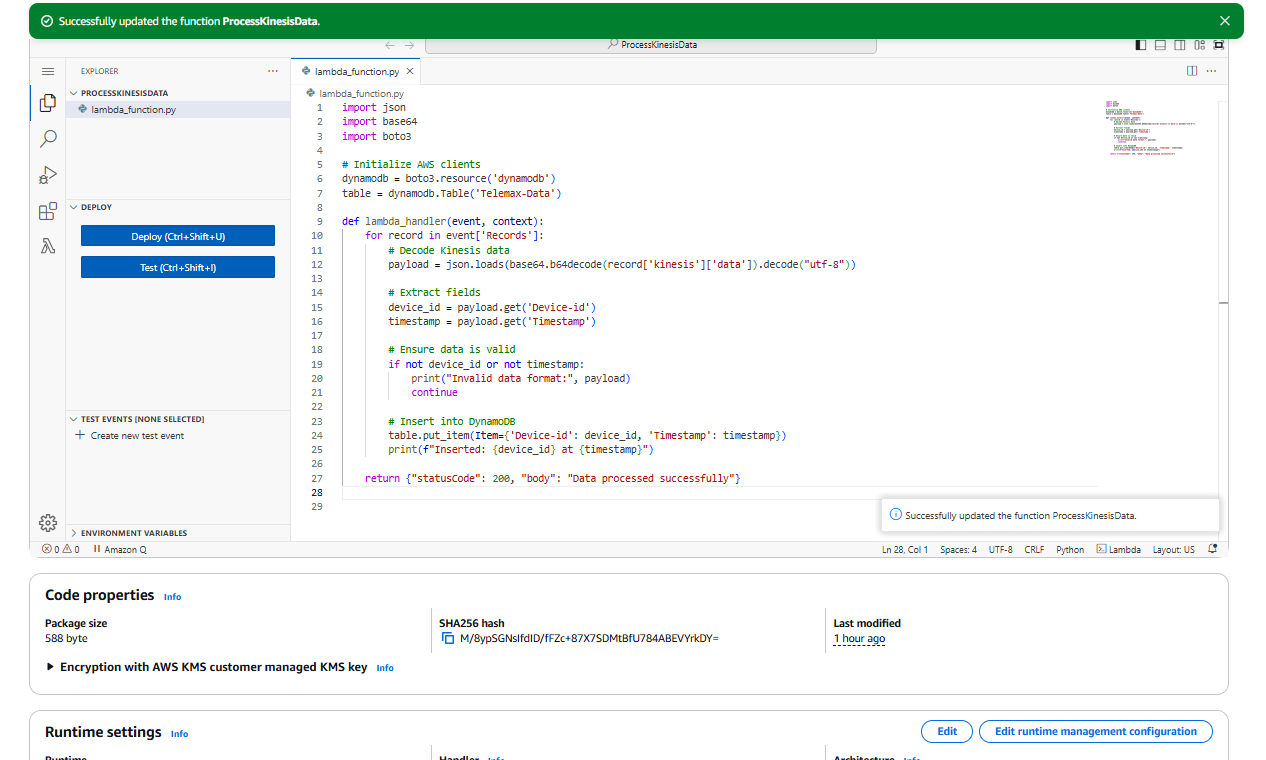
**return {"statusCode": 200, "body": "Data processed successfully"}**

****

After the code is pasted in the code section, click on deploy to deploy the code.

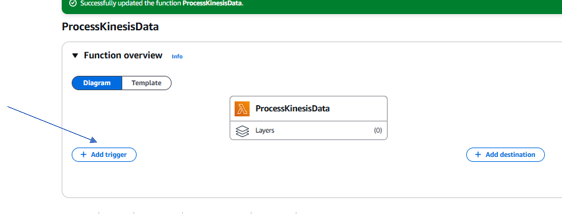


Now the following screen shows that the code is successfully deployed.



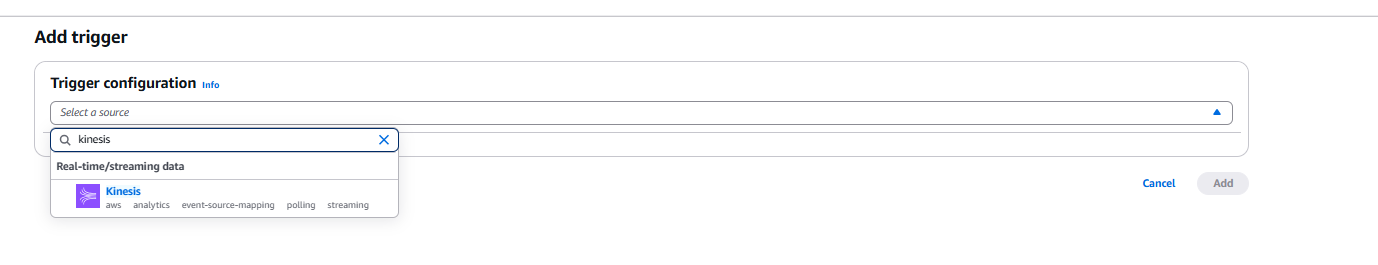
**Now we will need to Add Kinesis Trigger, for which we will need the following steps to be followed:**

* Go to the **Lambda function** → **Add Trigger**.

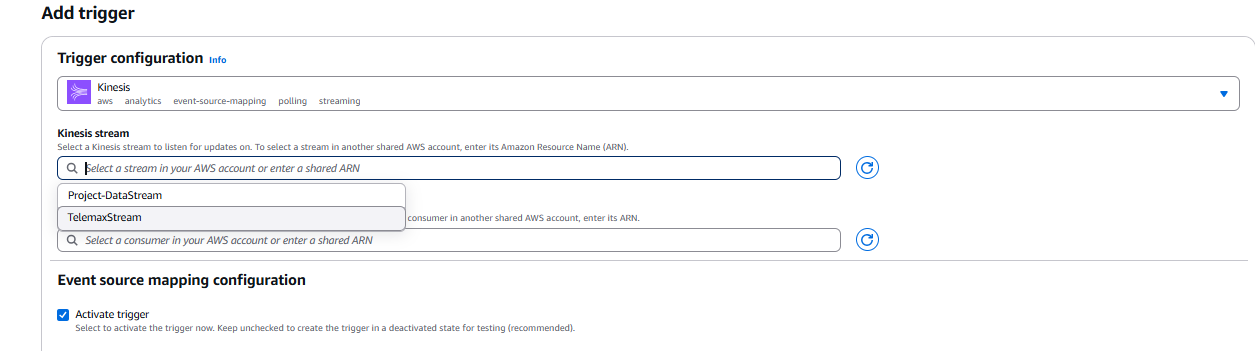


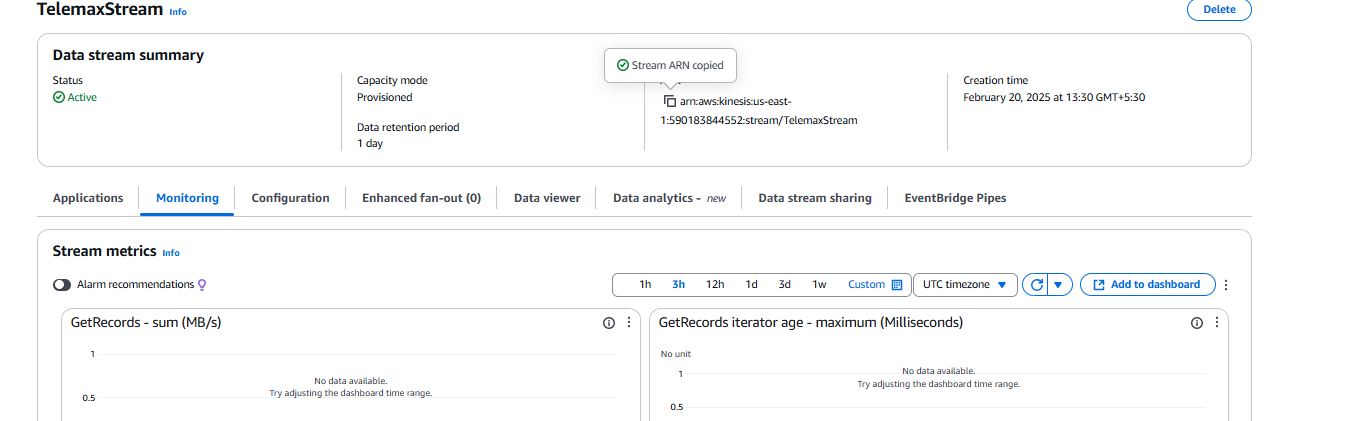


* To add a trigger select the arrow to choose the source for the trigger configuration as shown in the screenshot above

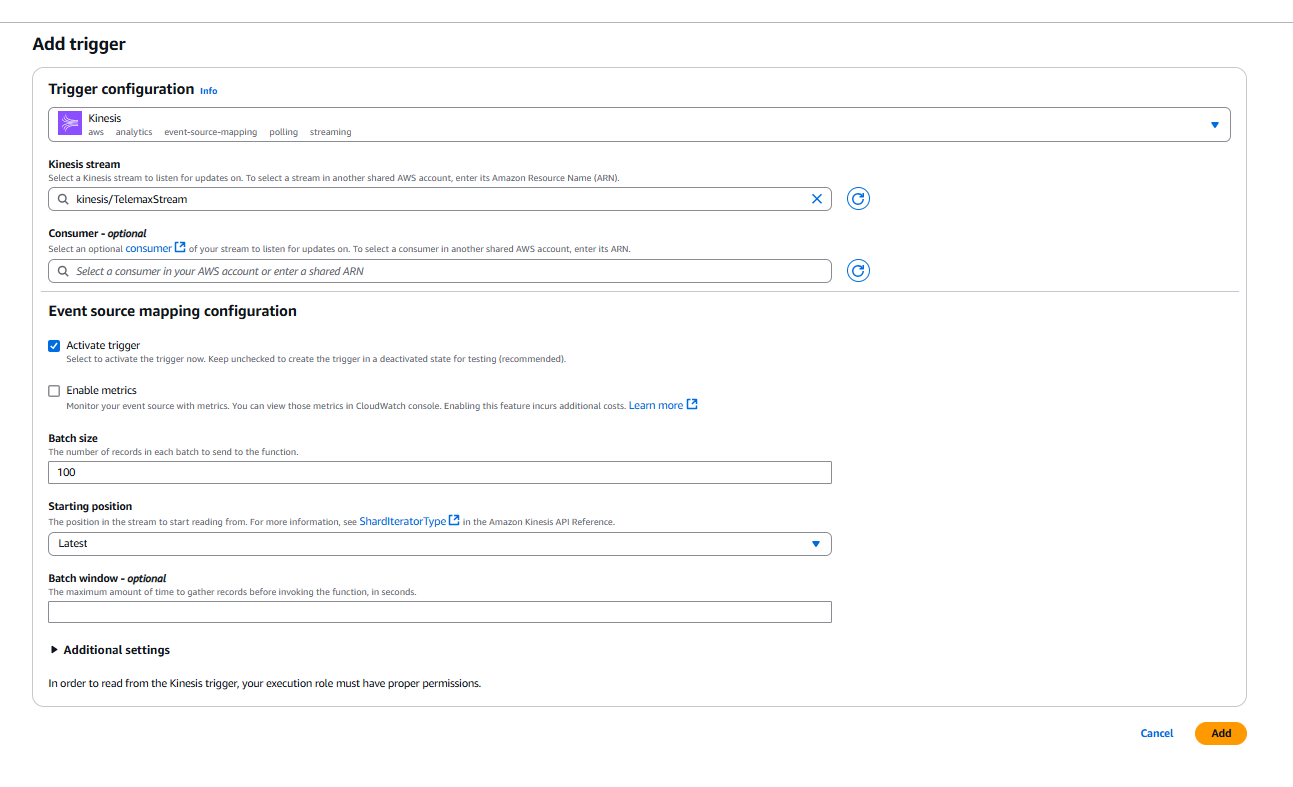
And Choose **Kinesis** as shown in the following screenshot  


Now, Select the created **Kinesis Data Stream** from the source, alternatively, you could explore the Kinesis service to copy the **ARN** of the created stream to be pasted) as shown in the two screenshots below:



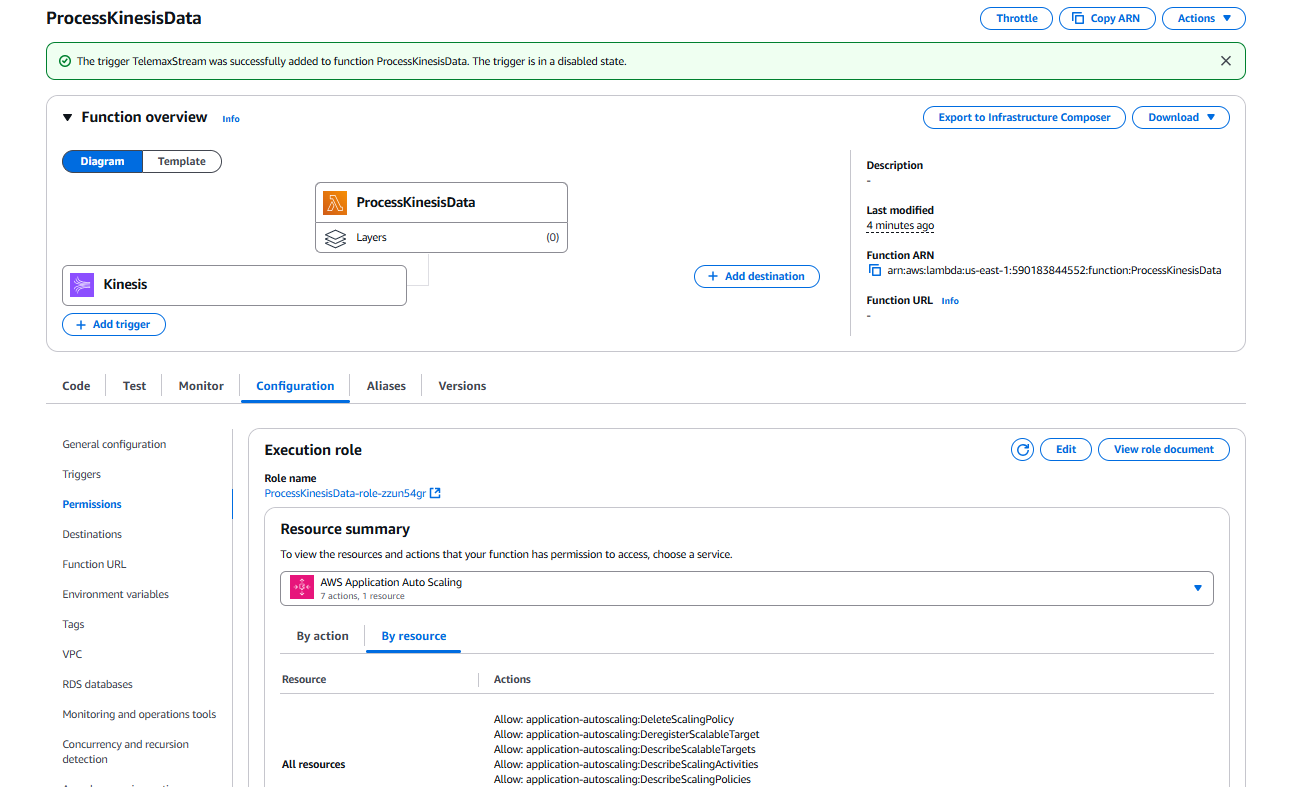


I have selected the Kinesis stream from the drop-down menu to proceed with my project.



Now Configure batch size (default: **100** records) and Click **Add**.

Not able to add the trigger as the role has no defined set of permissions to perform the following policies, otherwise the trigger would have been created successfully.



Hence, the Kinesis Stream is successfully linked as Trigger to Lambda Function

Also to Test The Lambda Function we just created we will follow the following Steps:

Now Go To Test in the Lambda Function named ProcessKinesisData.

Give a Name to the test as ‘Event’ and

Then Paste the following code in the Json section to test the function:

{

"Records": [

{

"kinesis": {

"kinesisSchemaVersion": "1.0",

"partitionKey": "device-1",

"sequenceNumber": "123456789",

"data": "eyJkZXZpY2VJZCI6ICJhIiwgInRpbWVzdGFtcCI6IDV9",

"approximateArrivalTimestamp": 1700000000

},

"eventSource": "aws:kinesis",

"eventVersion": "1.0",

"eventID": "shardId-000000000001:123456789",

"eventName": "aws:kinesis:record",

"invokeIdentityArn": "arn:aws:iam::123456789012:role/LambdaExecutionRole",

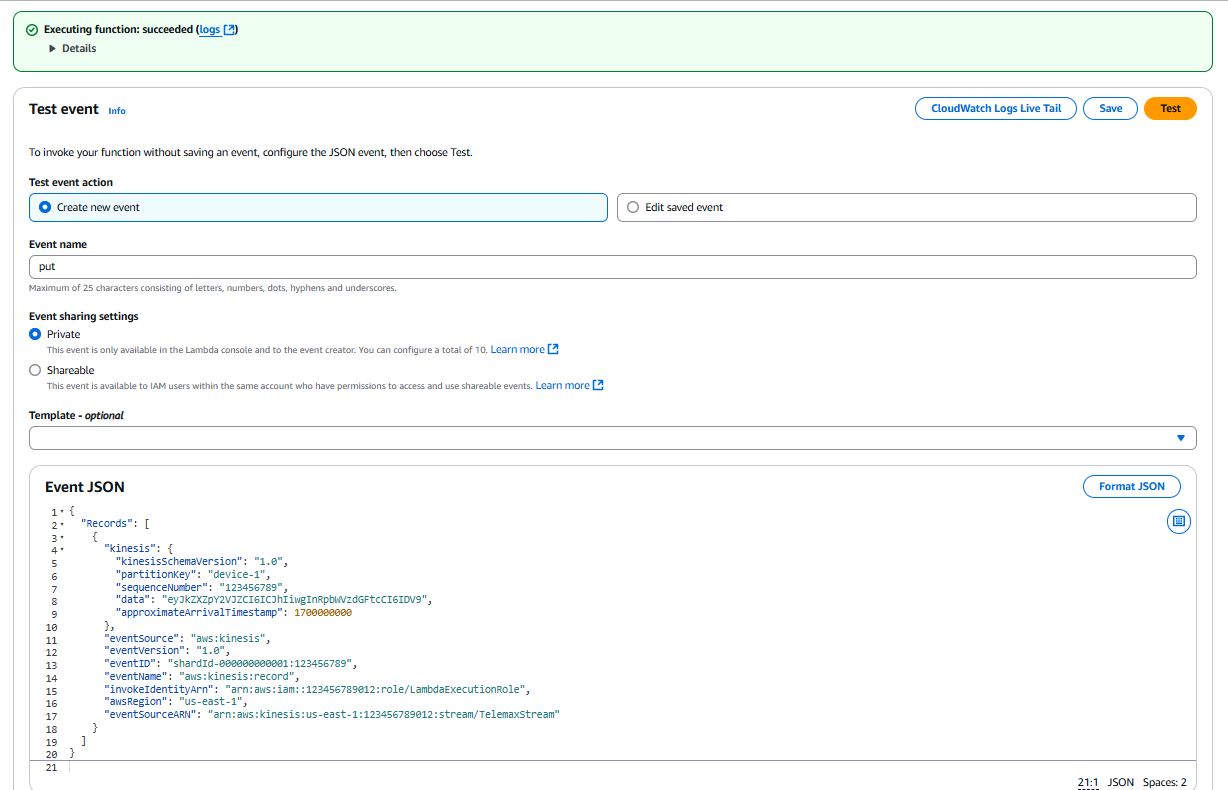
"awsRegion": "us-east-1",

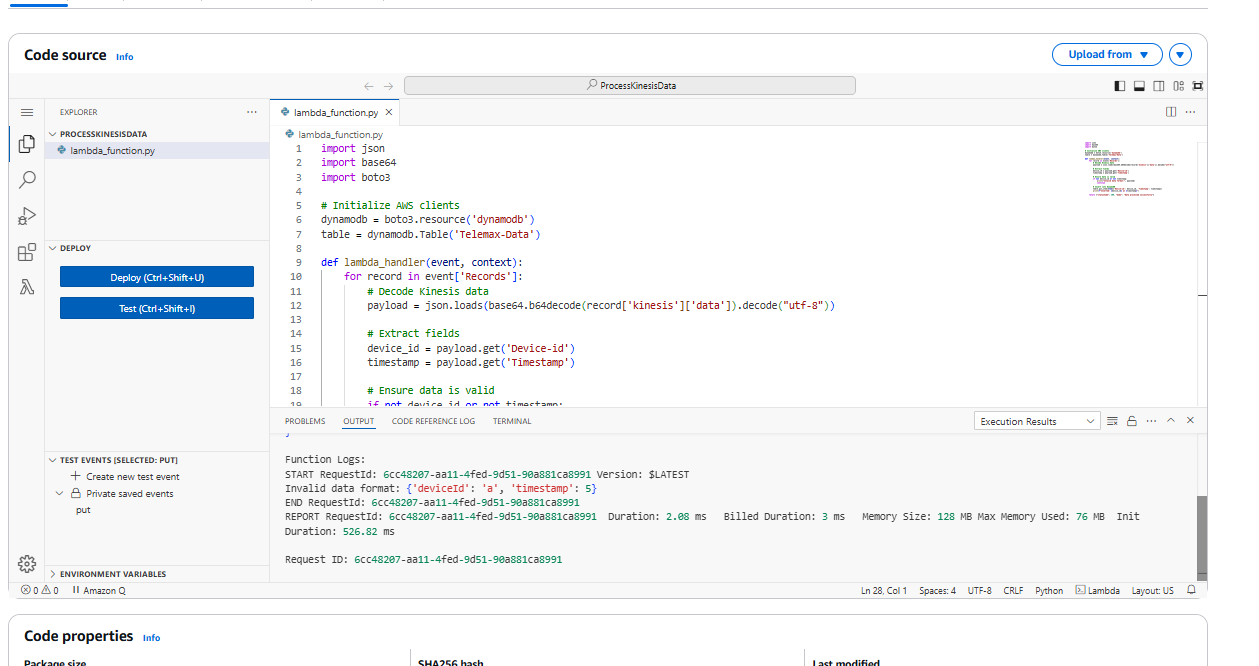
"eventSourceARN": "arn:aws:kinesis:us-east-1:123456789012:stream/TelemaxStream"

}

]

}





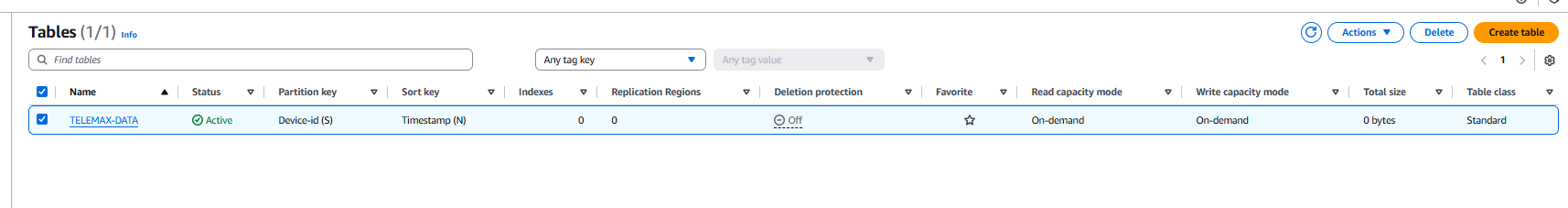
The above screen confirms that the test has been successfully conducted.

1. **Perform a Scan Operation on the Dynamo Db table:**

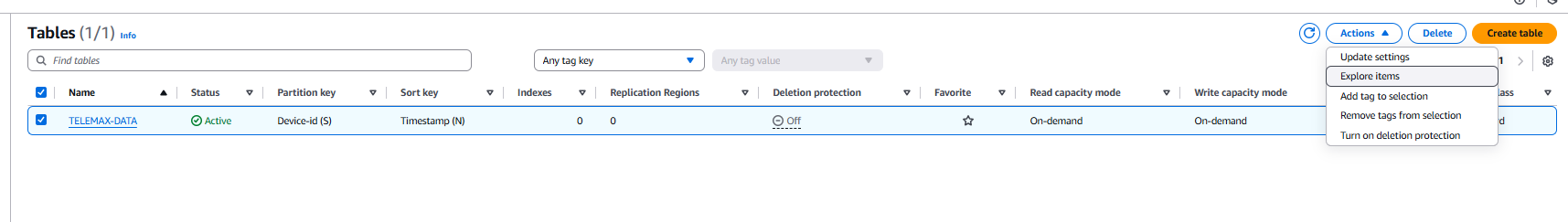
**Purpose:**  
The **Scan Operation** retrieves all records from the table for analysis.

**Steps to Perform Scan Operation:**

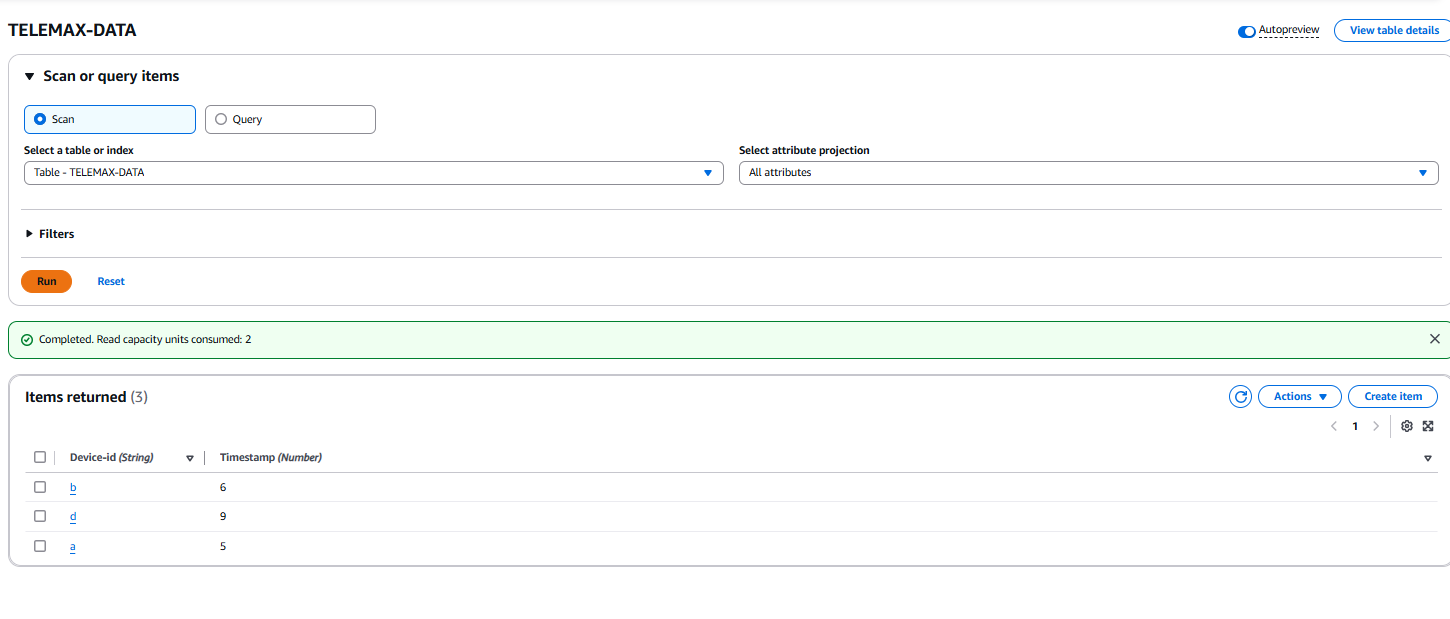
1. Go to **DynamoDB Console** → Select the TELEMAX-Data table.



1. Click **Explore Table Items**



1. Click **Scan** to view all stored records.



**5. Conclusion**

This project demonstrates the integration of AWS services for real-time data streaming and storage. By implementing Kinesis Data Streams, AWS Lambda, and DynamoDB, businesses can efficiently manage and analyze their data. TELEMAX, as a case study, can leverage this solution to enhance its networking infrastructure, optimize performance, and make data-driven decisions for future growth.