DATA ENGINEERING:

QUESTION: Stream IoT data using AWS Kinesis Data Streams, apply transformation via AWS Glue, and store in S3.

1. INTRODUCTION

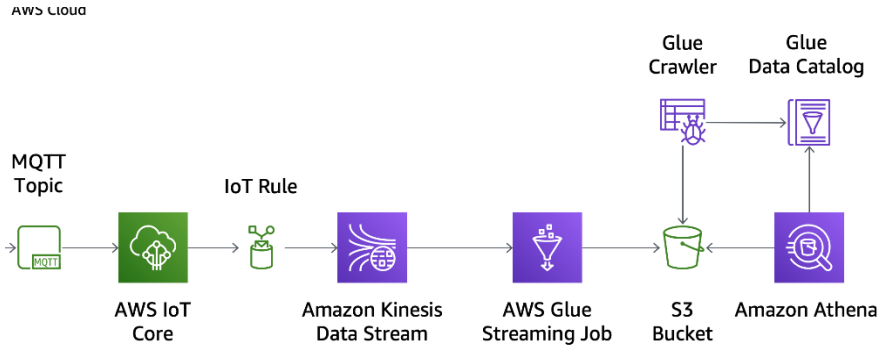
In this experiment, we will design a solution for streaming IoT data using AWS Kinesis Data Streams, applying data transformation with AWS Glue, and finally storing the transformed data in Amazon S3. This solution will enable the real-time processing of IoT data, applying necessary transformations for analysis, and archiving the data for future use. We will outline the necessary components and workflow to achieve this:

- **AWS Kinesis Data Streams**: Used for real-time streaming of IoT sensor data.
- **AWS Glue**: Data transformation service to clean and process the streamed data.
- **Amazon S3**: Cloud storage for storing the transformed IoT data.

2. SYSTEM ARCHITECTURE OVERVIEW

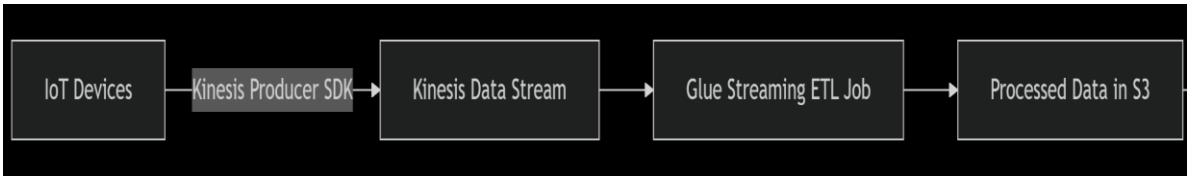
The architecture for this solution involves multiple AWS services working in tandem to handle the streaming, transformation, and storage of IoT data. The architecture can be summarized as follows:

Diagram:



- **IoT Devices**: Devices that generate telemetry data (such as temperature, humidity, pressure).

- **AWS Kinesis Data Streams**: Captures the streaming data from the IoT devices.
- **AWS Glue**: Applies transformation and prepares the data for storage.
- **Amazon S3**: Stores the transformed data for future analysis.



3. AWS KINESIS DATA STREAMS SETUP

AWS Kinesis Data Streams is used to ingest real-time data from IoT devices. Below are the steps to set up Kinesis Data Streams for capturing data:

Steps:

1. **Create Kinesis Data Stream**:
    - Navigate to the AWS Kinesis console.
    - Click on "Create Data Stream."
    - Enter a stream name (e.g., IoTDataStream).
    - Set the number of shards (typically 1 for small streams).
2. **Configure IoT Devices to Stream Data**:
    - Use AWS SDK or MQTT protocol to stream data from IoT devices to Kinesis Data Streams.
3. **Example code to publish data to Kinesis:**

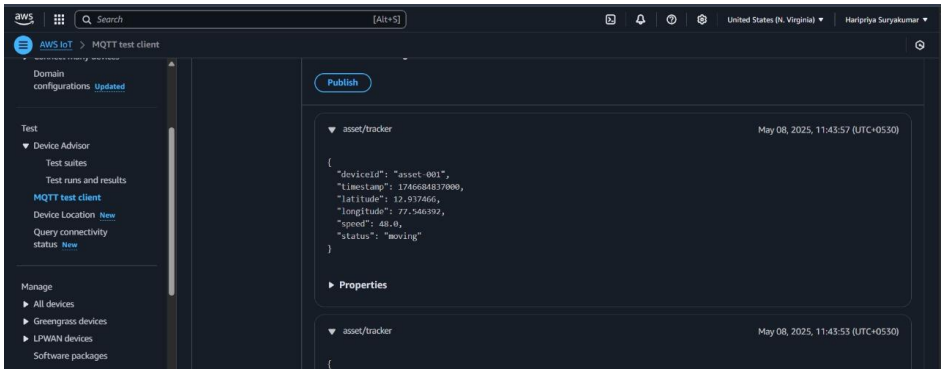
```
import boto3
import json
from datetime import datetime

kinesis = boto3.client('kinesis', region_name='us-west-2')
```

```
data = {
    "latitude": 37.7749,
    "longitude": -122.4194,
    "timestamp": datetime.utcnow().isoformat(),
    "device_id": "device123",
    "status": "active"
}

kinesis.put_record(
    StreamName='IoTDataStream',
    Data=json.dumps(data),
    PartitionKey="device123"
)
```

4. PYTHON TELEMETRARY

```
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient

import time

import json

import random

import datetime

client = AWSIoTMQTTClient("AssetTracker")

client.configureEndpoint("a3dmb30l73pi7o-ats.iot.us-east-1.amazonaws.com", 8883)

client.configureCredentials("AmazonRootCA1.pem",
"private.pem.key","certificate.pem.crt")

cliet.configureOfflinePublishQueueing(-1)

clint.configureDrainingFrequency(2)

client.configureConnectDisconnectTimeout(10)
```

```
client.configureMQTTOperationTimeout(5)

client.connect()

def iso_to_unix_ms(iso_timestamp):    dt = datetime.datetime.strptime(iso_timestamp, '%Y-%m-%dT%H:%M:%SZ')unix_timestamp = nt(dt.timestamp())  # Convert to Unix timestamp (seconds)

        unix_timestamp_ms = unix_timestamp * 1000    # Convert to milliseconds
        return unix_timestamp_ms
    def simulate_asset_data():
        data = {
            "deviceId": "asset-001",
            "timestamp": time.strftime("%Y-%m-%dT%H:%M:%SZ"),
            "latitude": round(random.uniform(12.90, 13.00), 6),
            "longitude": round(random.uniform(77.50, 77.60), 6),
            "speed": round(random.uniform(0, 80), 2),
            "status": random.choice(["moving", "stopped"])
        }
            data["timestamp"] = iso_to_unix_ms(data["timestamp"])
        return data

while True:

        data = simulate_asset_data()
        print("Publishing:", data)
        client.publish("asset/tracker", json.dumps(data), 1)
        time.sleep(5)
```



4. AWS GLUE SETUP

AWS Glue is used for data transformation. Here, we will use Glue to process and clean the data streaming through Kinesis.

Steps to Set Up AWS Glue:

1. **Create a Glue Crawler**:

- Navigate to AWS Glue console.
  - Create a Crawler to catalog the raw data from Kinesis Data Streams.
  - Define the data source as Kinesis Data Stream and set up the target as Glue Data Catalog.

2. **Create a Glue ETL Job**: Define a Glue job to transform the data. This job will read from the Kinesis Data Stream, apply the necessary transformations (e.g., data cleaning, schema adjustment), and then write the processed data to an S3 bucket.

3. **Transformation Script**:

```
import boto3

import json

from awsglue.context import GlueContext

from pyspark.context import SparkContext

sc = SparkContext()

glueContext = GlueContext(sc)

dynamic_frame = glueContext.create_dynamic_frame.from_catalog(

database="iot_data_db", table_name="raw_data"

)
```

transformed_data = dynamic_frame.drop_nulls()

glueContext.write_dynamic_frame.from_options(

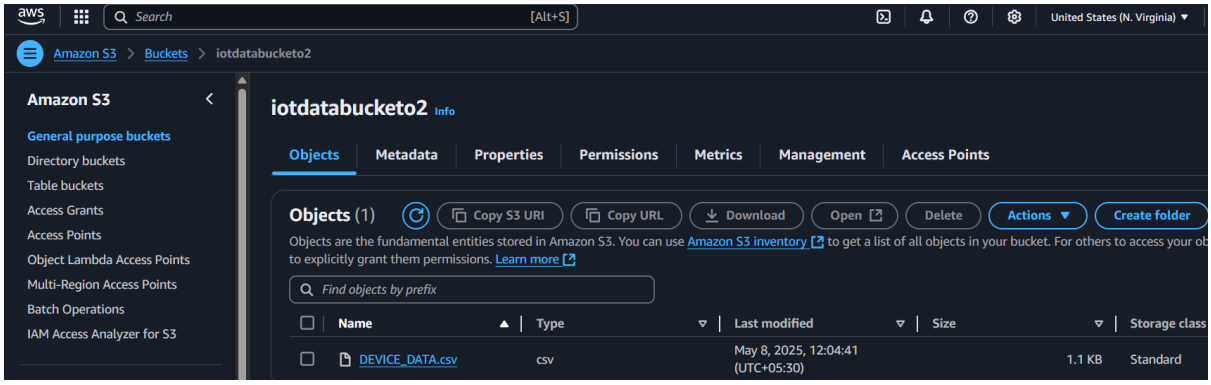
transformed_data, connection_type="s3", connection_options={"path": "s3://my-bucket/transformed-data/"}

5. STORING DATA IN S3

After applying transformations using AWS Glue, the cleaned data is stored in Amazon S3 for long-term storage and further analysis.

Steps:

1. **Create an S3 Bucket**:
    - Go to the S3 console and create a new bucket (e.g., iot-transformed-data).
    - Define permissions to allow access for Glue and other AWS services.
2. **Store Transformed Data**:
    - Data is written to the S3 bucket by the Glue job after the transformation.



6. DATA FLOW AND INTEGRATION

The data flow for this solution is designed to efficiently manage and process IoT telemetry data in real time. The flow can be described in the following steps:

1. **Real-time Data Collection from IoT Devices**: IoT devices continuously send telemetry data, such as device status, location, and environmental conditions,

to **Kinesis Data Streams**. Kinesis acts as a high-throughput, low-latency data stream service capable of handling large amounts of data from various devices.

2. **Data Transformation with AWS Glue**: The data is then ingested into **AWS Glue**, which serves as an ETL (Extract, Transform, Load) service. Glue transforms the incoming data by performing necessary operations, such as:
    - **Data Cleaning**: Removing any invalid or null values that might hinder the analysis.
    - **Schema Adjustments**: Reformatting the data into a standardized structure or correcting inconsistent schema formats.
    - **Enrichment**: Optionally, adding additional metadata or enhancing the dataset with derived insights.
3. **Data Storage in Amazon S3**: After the transformation, the data is written to **Amazon S3**, a scalable and durable storage service. The data is stored in a structured or partitioned format, ensuring it is easily accessible for later analysis, reporting, or visualizations. This stage makes the data ready for downstream analytics, business intelligence tools like Amazon QuickSight, or machine learning processes.

7. CONCLUSION

This solution demonstrates an end-to-end process of streaming IoT data using AWS Kinesis Data Streams, applying necessary transformations with AWS Glue, and storing the processed data in Amazon S3 for future use. By utilizing the power of AWS services, this architecture provides an efficient, scalable, and cost-effective approach to handle vast amounts of real-time IoT data.

The system ensures that:

- IoT data is collected in real time with minimal latency.
- The data is properly transformed to meet business or analytical needs.
- The transformed data is stored efficiently, making it available for various applications such as real-time dashboards, historical analysis, or machine learning models.