**Job-Ready Project: Kafka Real-time Data Processing**

**Integrate AWS Lambda**

***with***

**Amazon MSK**

**1. Setup: Launching the MSK Cluster**

1. **Create a VPC and Subnets**
   * For MSK’s high availability, set up a VPC with at least two subnets in different availability zones.
2. **Launch MSK Cluster in AWS Console**
   * Go to **Amazon MSK > Create Cluster**.
   * Choose **Kafka Version** (e.g., 2.8.1).
   * Configure networking, ensuring it’s in the VPC and subnets created.
   * For a basic POC, select **Plaintext** for unauthenticated access.
   * **Save the Kafka Broker Endpoints** from the configuration, which will be used later.

**2. Launch an EC2 Instance as a Kafka Client**

1. **Launch EC2 Instance (Ubuntu)**
   * Go to **EC2 > Launch Instance**.
   * Select **Ubuntu** as the AMI and a suitable instance type (e.g., t2.micro).
   * In **Network Settings**, choose the VPC and a public subnet, and enable **Auto-assign Public IP**.
2. **Configure Security Groups**
   * Create security groups allowing all traffic between MSK and EC2 instance.
   * Add inbound/outbound rules to allow traffic on Kafka broker ports (9092, 9094).

**3. Connect to EC2 Instance and Install Kafka Tools**

1. **SSH into EC2 Instance**

ssh -i "your-key.pem" ubuntu@your-ec2-public-ip

1. **Install Java**  
   Kafka requires Java. Install it with:

**sudo apt update**

**sudo apt install -y openjdk-11-jdk**

1. **Download Kafka and Set Up Kafka Tools**

wget https://archive.apache.org/dist/kafka/2.8.1/kafka\_2.13-2.8.1.tgz

tar -xvf kafka\_2.13-2.8.1.tgz

cd kafka\_2.13-2.8.1

**4. Create Kafka Topics**

1. **Create a Topic for Testing**: In your EC2 instance’s Kafka directory:

**bin/kafka-topics.sh --create --topic test-topic --bootstrap-server <MSK-broker-endpoints> --replication-factor 2 --partitions 2**

**5. Configure Lambda to Publish Messages to MSK**

1. **Create a Lambda Function**
   * Go to **AWS Lambda > Create Function**.
   * Choose **Python 3.x** as the runtime.
2. **Set Up Lambda Permissions**
   * Ensure Lambda has VPC access to the MSK VPC and subnets.
   * Attach the necessary IAM role with access to MSK and SQS.
3. **Add Kafka Producer Code to Lambda**
   * Use the kafka-python library to connect Lambda with Kafka.
   * The following sample function publishes a message to Kafka:

**import json**

**from kafka import KafkaProducer**

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

**producer = KafkaProducer(**

**bootstrap\_servers=['<MSK-broker-endpoints>'],**

**value\_serializer=lambda v: json.dumps(v).encode('utf-8')**

**)**

**message = {'key': 'value'}**

**producer.send('test-topic', value=message)**

**producer.flush()**

**return {**

**'statusCode': 200,**

**'body': json.dumps('Message sent to Kafka!')**

**}**

1. **Deploy Lambda Function**

**6. Set Up SQS Queue for Load Balancing**

1. **Create an SQS Queue**
   * Go to **SQS > Create Queue** and choose **Standard Queue**.
2. **Set Up SQS to Trigger Lambda**
   * Go to **AWS Lambda > your-function-name > Triggers** and add SQS as a trigger.
3. **Test SQS and Lambda Workflow**
   * Send a message to SQS and verify it triggers Lambda.

**7. Test Kafka Consumer on EC2**

1. **Install Kafka Python Library**

pip3 install kafka-python

1. **Write and Run Kafka Consumer in Python**

**from kafka import KafkaConsumer**

**consumer = KafkaConsumer(**

**'test-topic',**

**bootstrap\_servers=['<MSK-broker-endpoints>'],**

**auto\_offset\_reset='earliest',**

**enable\_auto\_commit=True,**

**group\_id='test-group'**

**)**

**for message in consumer:**

**print(f"Received message: {message.value.decode('utf-8')}")**

* + Run this consumer script on the EC2 instance to receive messages from the Kafka topic.

**8. Test the Complete Flow**

1. **Send a Message via Lambda**
   * **Manually Invoke Lambda:**
     + Go to **AWS Lambda > your-function-name > Test**.
     + Select or create a test event with a simple payload (e.g., {"test\_key": "test\_value"}).
     + Click **Test** to manually invoke the function and send a message to Kafka.
   * **Send a Message via SQS to Trigger Lambda:**
     + Go to **Amazon SQS > your-queue-name > Send and receive messages**.
     + Enter a test message and click **Send message**.
     + This will trigger Lambda automatically if SQS is set as a trigger.
2. **Verify Kafka Consumer Output**
   * On the EC2 instance, run the Kafka consumer script from Step 7.
   * You should see the message from Lambda printed in the console, confirming successful integration and message flow through SQS, Lambda, and Kafka