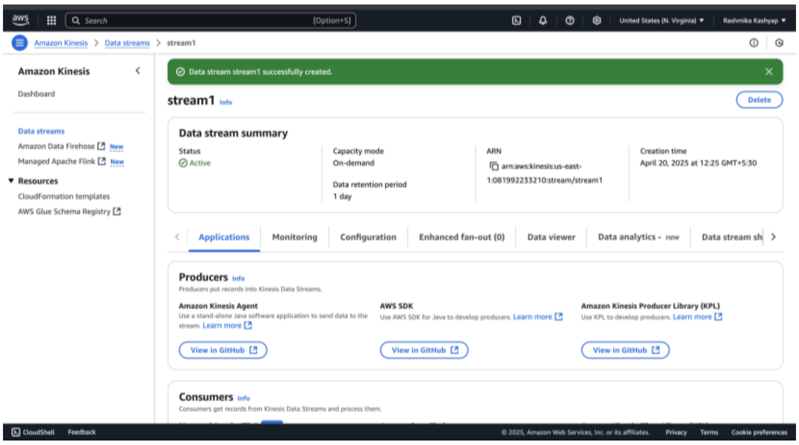
CLOUD ASSIGNMENT 10

E22CSEU0550

RASHMIKA KASHYAP

Creating kinesis stream



Data stream stream1 successfully created.

stream1 Info

Data stream summary

Status: Active

Capacity mode: On-demand

Data retention period: 1 day

ARN: arn:aws:kinesis:us-east-1:081992233210:stream/stream1

Creation time: April 20, 2025 at 12:25 GMT+5:30

Producers Info

Producers put records into Kinesis Data Streams.

Amazon Kinesis Agent
Use a stand-alone Java software application to send data to the stream. Learn more

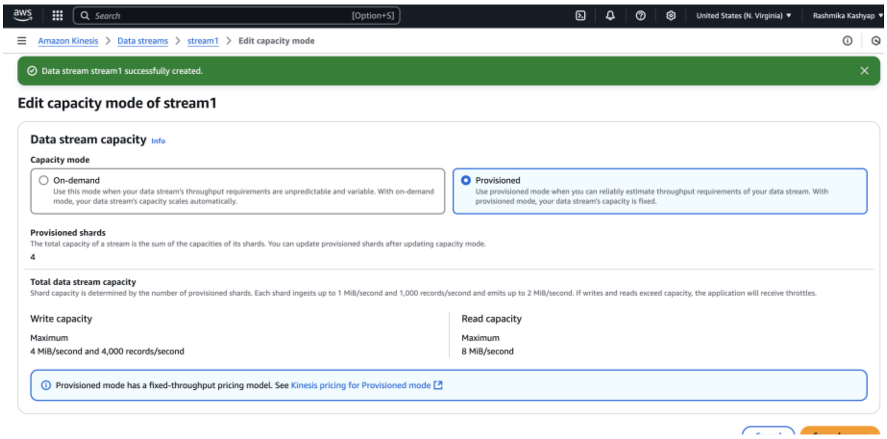
AWS SDK
Use AWS SDK for Java to develop producers. Learn more

Amazon Kinesis Producer Library (KPL)
Use KPL to develop producers. Learn more

Consumers Info

Consumers get records from Kinesis Data Streams and process them.

Creating provisioned capacity


aws
Search [Option+S]
United States (N. Virginia) ▾
Rashmika Kashyap ▾
Amazon Kinesis > Data streams > stream1 > Edit capacity mode

Data stream stream1 successfully created.

Edit capacity mode of stream1

Data stream capacity Info

Capacity mode

○ On-demand
Use this mode when your data stream's throughput requirements are unpredictable and variable. With on-demand mode, your data stream's capacity scales automatically.

● Provisioned
Use provisioned mode when you can reliably estimate throughput requirements of your data stream. With provisioned mode, your data stream's capacity is fixed.

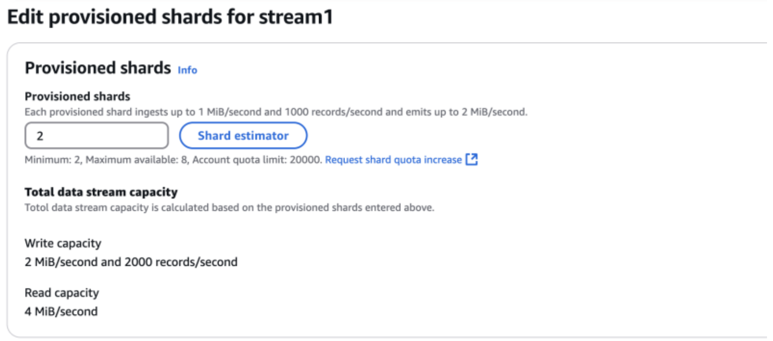
Provisioned shards
The total capacity of a stream is the sum of the capacities of its shards. You can update provisioned shards after updating capacity mode.
4

Total data stream capacity
Shard capacity is determined by the number of provisioned shards. Each shard ingests up to 1 MiB/second and 1,000 records/second and emits up to 2 MiB/second. If writes and reads exceed capacity, the application will receive throttles.

Write capacity
Maximum
4 MiB/second and 4,000 records/second

Read capacity
Maximum
8 MiB/second

Provisioned mode has a fixed-throughput pricing model. See Kinesis pricing for Provisioned mode



Edit provisioned shards for stream1

Provisioned shards Info

Provisioned shards
Each provisioned shard ingests up to 1 MiB/second and 1000 records/second and emits up to 2 MiB/second.

2

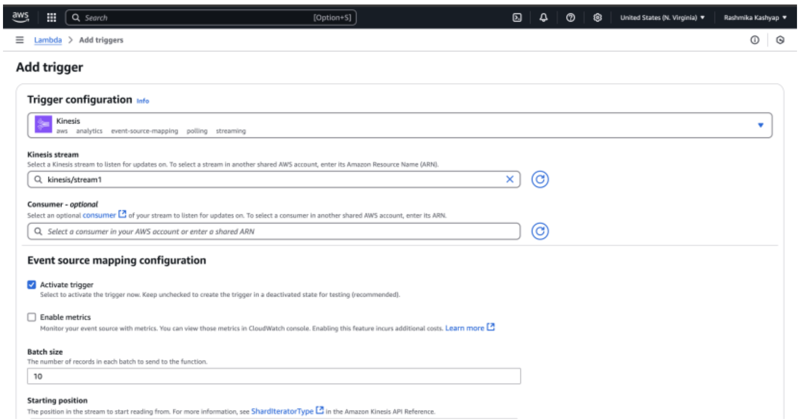
Minimum: 2, Maximum available: 8, Account quota limit: 20000. Request shard quota increase

Total data stream capacity
Totol data stream capacity is calculated based on the provisioned shards entered above.

Write capacity
2 MiB/second and 2000 records/second

Read capacity
4 MiB/second

Adding kinesis as trigger to lambdaConsumer



Add trigger

Trigger configuration Info

Kinesis
aws analytics event-source-mapping polling streaming

Kinesis stream
Select a Kinesis stream to listen for updates on. To select a stream in another shared AWS account, enter its Amazon Resource Name (ARN).
kinesis/stream1

Consumer - *optional*
Select an optional consumer of your stream to listen for updates on. To select a consumer in another shared AWS account, enter its ARN.
Select a consumer in your AWS account or enter a shared ARN

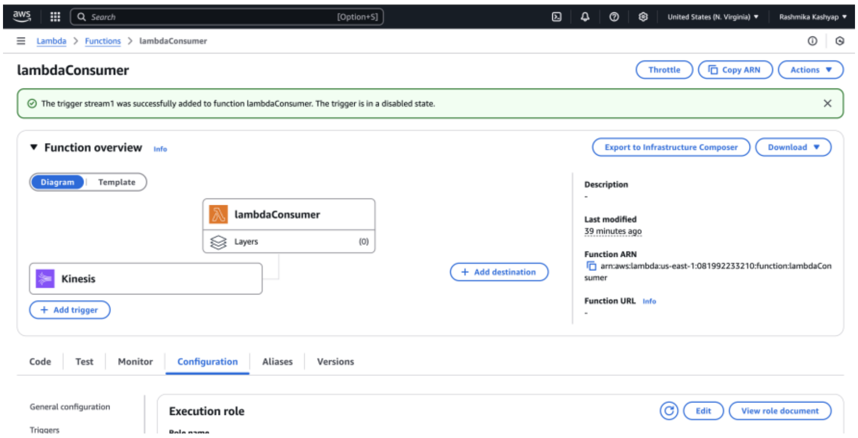
Event source mapping configuration

☑ Activate trigger
Select to activate the trigger now. Keep unchecked to create the trigger in a deactivated state for testing (recommended).

□ Enable metrics
Monitor your event source with metrics. You can view those metrics in CloudWatch console. Enabling this feature incurs additional costs. Learn more

Batch size
The number of records in each batch to send to the function.
10

Starting position
The position in the stream to start reading from. For more information, see ShardIteratorType in the Amazon Kinesis API Reference.



lambdaConsumer

The trigger stream1 was successfully added to function lambdaConsumer. The trigger is in a disabled state.

Function overview Info

Description
-

Last modified
39 minutes ago

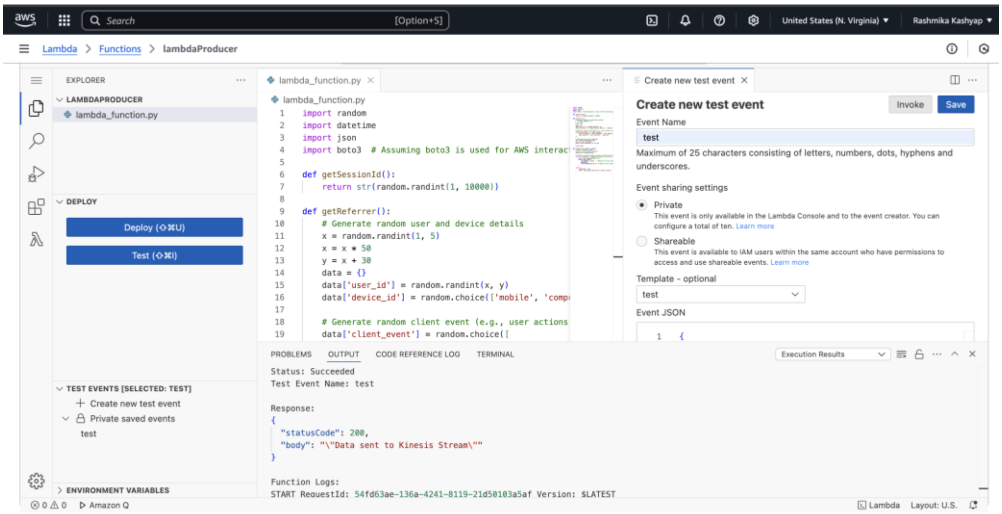
Function ARN
arn:aws:lambda:us-east-1:081992233210:function:lambdaConsumer

Function URL Info
-

Execution role

Role name

Successfully executing lambdaProducer which has permissions to forward data to kinesis which triggers lambdaConsumer



```
import random
import datetime
import json
import boto3  # Assuming boto3 is used for AWS interac

def getSessionId():
    return str(random.randint(1, 10000))

def getReferrer():
    # Generate random user and device details
    x = random.randint(1, 5)
    x = x * 50
    y = x + 30
    data = {}
    data['user_id'] = random.randint(x, y)
    data['device_id'] = random.choice(['mobile', 'comp

    # Generate random client event (e.g., user actions
    data['client_event'] = random.choice([
```

Create new test event

Event Name
test
Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

● Private
This event is only available in the Lambda Console and to the event creator. You can configure a total of ten. Learn more

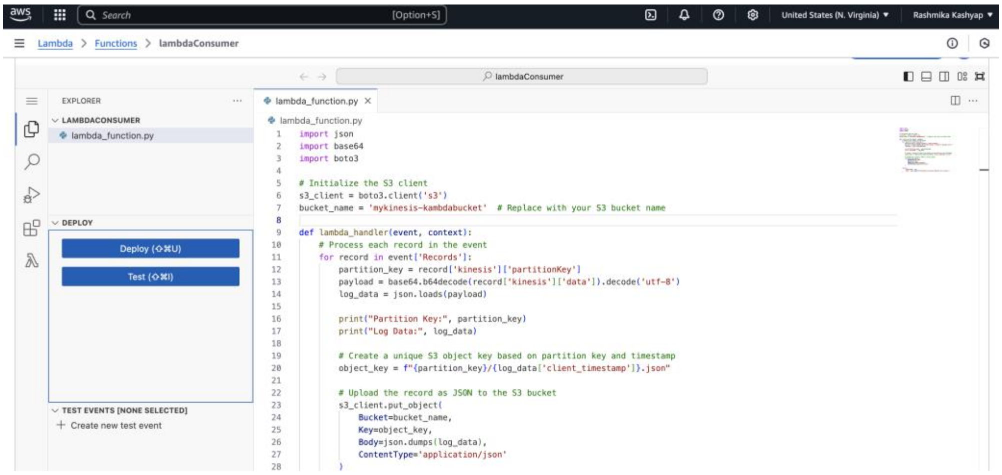
○ Shareable
This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more

Template - optional
test

Event JSON

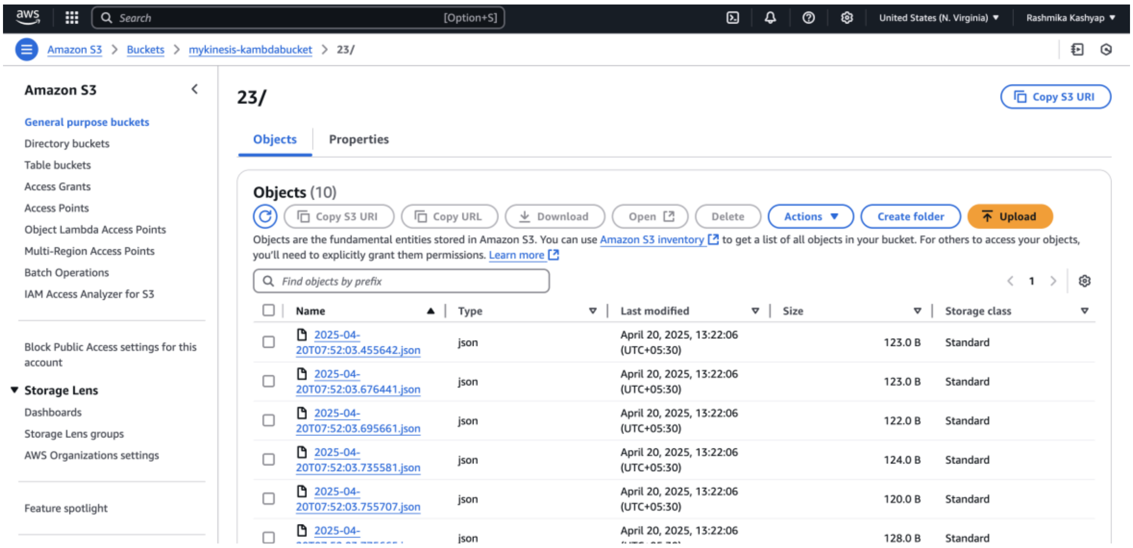
```
1  {
```

```
Status: Succeeded
Test Event Name: test

Response:
{
  "statusCode": 200,
  "body": "\"Data sent to Kinesis Stream\""
}

Function Logs:
START RequestId: 54fd63ae-136a-4241-8119-21d50103a5af Version: $LATEST
```



```python
import json
import base64
import boto3

# Initialize the S3 client
s3_client = boto3.client('s3')
bucket_name = 'mykinesis-kambdabucket'  # Replace with your S3 bucket name

def lambda_handler(event, context):
    # Process each record in the event
    for record in event['Records']:
        partition_key = record['kinesis']['partitionKey']
        payload = base64.b64decode(record['kinesis']['data']).decode('utf-8')
        log_data = json.loads(payload)

        print("Partition Key:", partition_key)
        print("Log Data:", log_data)

        # Create a unique S3 object key based on partition key and timestamp
        object_key = f"{partition_key}/{log_data['client_timestamp']}.json"

        # Upload the record as JSON to the S3 bucket
        s3_client.put_object(
            Bucket=bucket_name,
            Key=object_key,
            Body=json.dumps(log_data),
            ContentType='application/json'
        )
```

Therefore records saved as .json in S3



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	2025-04-20T07:52:03.455642.json	json	April 20, 2025, 13:22:06 (UTC+05:30)	123.0 B	Standard
□	2025-04-20T07:52:03.676441.json	json	April 20, 2025, 13:22:06 (UTC+05:30)	123.0 B	Standard
□	2025-04-20T07:52:03.695661.json	json	April 20, 2025, 13:22:06 (UTC+05:30)	122.0 B	Standard
□	2025-04-20T07:52:03.735581.json	json	April 20, 2025, 13:22:06 (UTC+05:30)	124.0 B	Standard
□	2025-04-20T07:52:03.755707.json	json	April 20, 2025, 13:22:06 (UTC+05:30)	120.0 B	Standard
□	2025-04-	json	April 20, 2025, 13:22:06	128.0 B	Standard