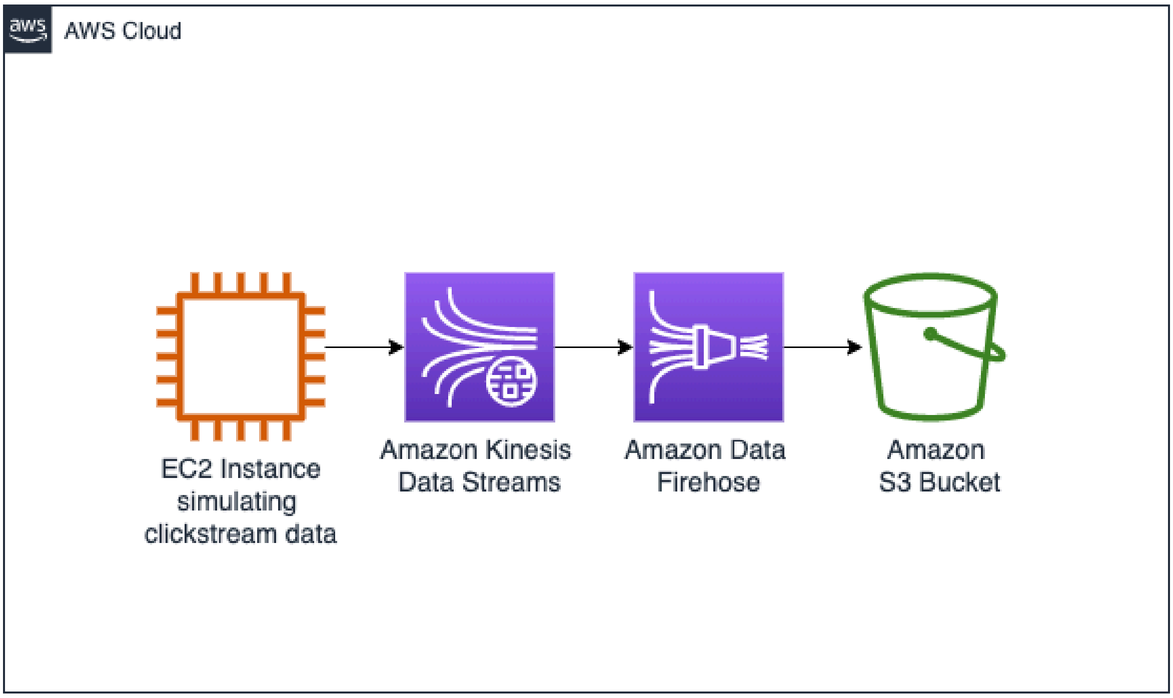
Lab 1 - Setting up a Streaming Delivery Pipeline with Amazon Kinesis

Objectives

1. Create a Kinesis Data Firehose stream and connect the Kinesis data stream to Kinesis Data Firehose.
2. Configure dynamic partitioning on the Kinesis Data Firehose delivery stream.
3. Deliver data to Amazon S3.


AWS Cloud
EC2 Instance simulating clickstream data
Amazon Kinesis Data Streams
Amazon Data Firehose
Amazon S3 Bucket

1. EC2 Instance simulating clickstream data: A virtual server in AWS that generates mock click data to mimic user activity on an e-commerce website.
2. Amazon Kinesis Data Streams: A service that collects and processes real-time data streams, capturing the click data generated by the EC2 instance.

3. Amazon Data Firehose: A service that takes data from Kinesis Data Streams and delivers it to destinations such as Amazon S3, applying any necessary transformations.
4. Amazon S3 Bucket: A storage service where the processed clickstream data is stored, organized, and made available for further analysis.

Task 1: Simulate clickstream data generation

1.1 open the **EC2 Producer terminal** using URL in lab, a terminal opens.

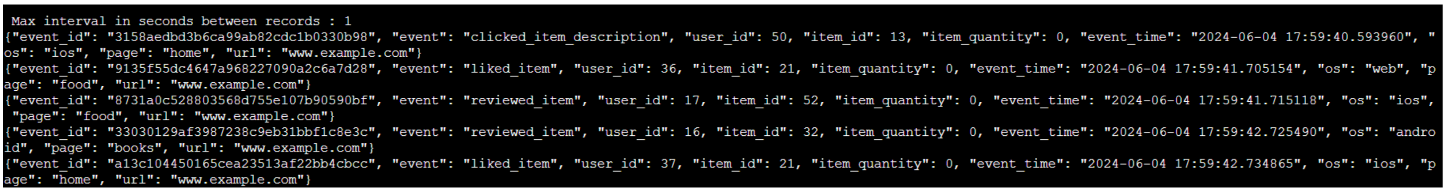
EC2 PT is like a virtual constumer in this case that will imitate clicks.

1.2 Use the following code to start the clickstream_generator_items.py script

```
STREAM_NAME=$(aws kinesis list-streams --query
"StreamNames[?contains(@, 'KdsClickstreamData')]" --output text)

echo -e "\n\nThe stream name is : $STREAM_NAME\n\n"

python3 clickstream_generator_items.py $STREAM_NAME 1 1
```

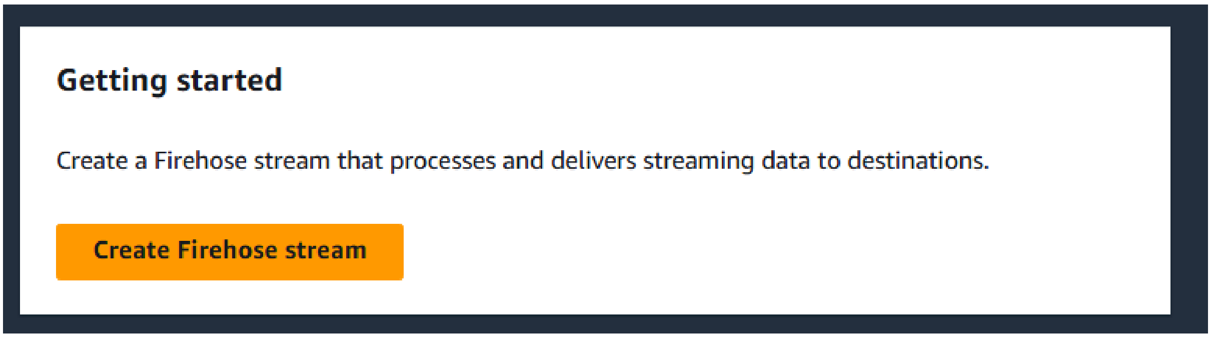


```
 Max interval in seconds between records : 1
{"event_id": "3158aedbd3b6ca99ab82cdc1b0330b98", "event": "clicked_item_description", "user_id": 50, "item_id": 13, "item_quantity": 0, "event_time": "2024-06-04 17:59:40.593960", "
os": "ios", "page": "home", "url": "www.example.com"}
{"event_id": "9135f55dc4647a968227090a2c6a7d28", "event": "liked_item", "user_id": 36, "item_id": 21, "item_quantity": 0, "event_time": "2024-06-04 17:59:41.705154", "os": "web", "p
age": "food", "url": "www.example.com"}
{"event_id": "8731a0c528803568d755e107b90590bf", "event": "reviewed_item", "user_id": 17, "item_id": 52, "item_quantity": 0, "event_time": "2024-06-04 17:59:41.715118", "os": "ios",
 "page": "food", "url": "www.example.com"}
{"event_id": "33030129af3987238c9eb31bbf1c8e3c", "event": "reviewed_item", "user_id": 16, "item_id": 32, "item_quantity": 0, "event_time": "2024-06-04 17:59:42.725490", "os": "andro
id", "page": "books", "url": "www.example.com"}
{"event_id": "a13c104450165cea23513af22bb4cbcc", "event": "liked_item", "user_id": 37, "item_id": 21, "item_quantity": 0, "event_time": "2024-06-04 17:59:42.734865", "os": "ios", "p
age": "home", "url": "www.example.com"}
```

These steps will simulate user activities on a website and send the data to a Kinesis data stream for further analysis.

Task 2: Create and configure a Kinesis Data Firehose delivery stream

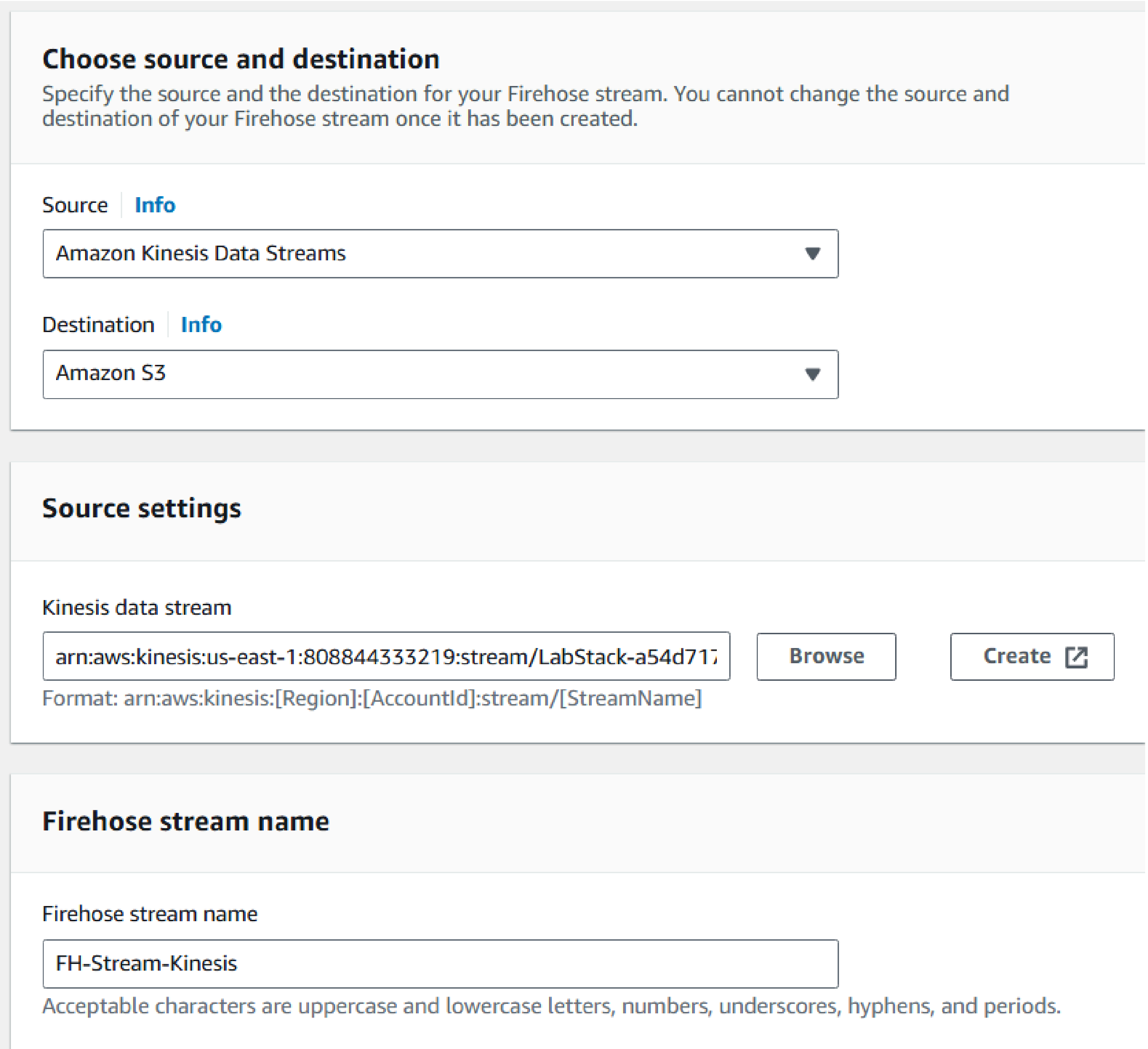
2.1 Open FIreHose in AWS console, and click on



Getting started

Create a Firehose stream that processes and delivers streaming data to destinations.

2.2 Setting up the stream



Choose source and destination

Specify the source and the destination for your Firehose stream. You cannot change the source and destination of your Firehose stream once it has been created.

Source | Info

Amazon Kinesis Data Streams ▼

Destination | Info

Amazon S3 ▼

Source settings

Kinesis data stream

arn:aws:kinesis:us-east-1:808844333219:stream/LabStack-a54d71

Format: arn:aws:kinesis:[Region]:[AccountId]:stream/[StreamName]

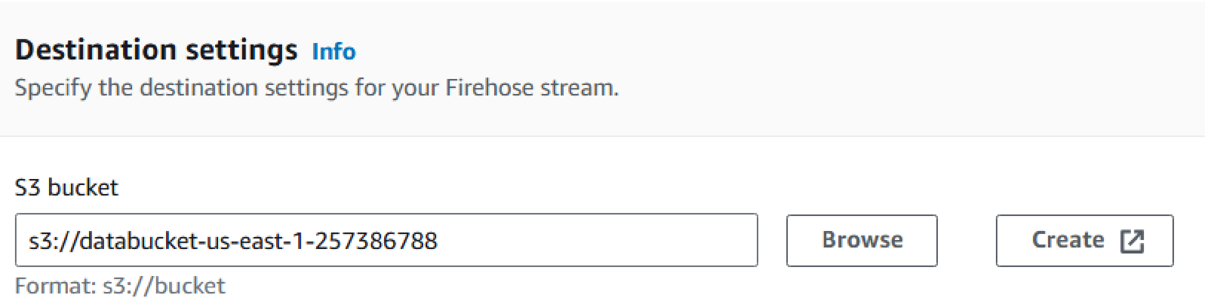
Firehose stream name

Firehose stream name

FH-Stream-Kinesis

Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens, and periods.

Source - Kinesis
Destination - S3



Destination settings Info

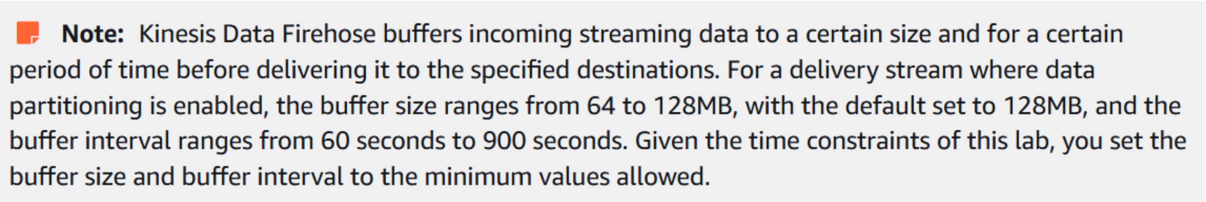
Specify the destination settings for your Firehose stream.

S3 bucket

s3://databucket-us-east-1-257386788

Format: s3://bucket

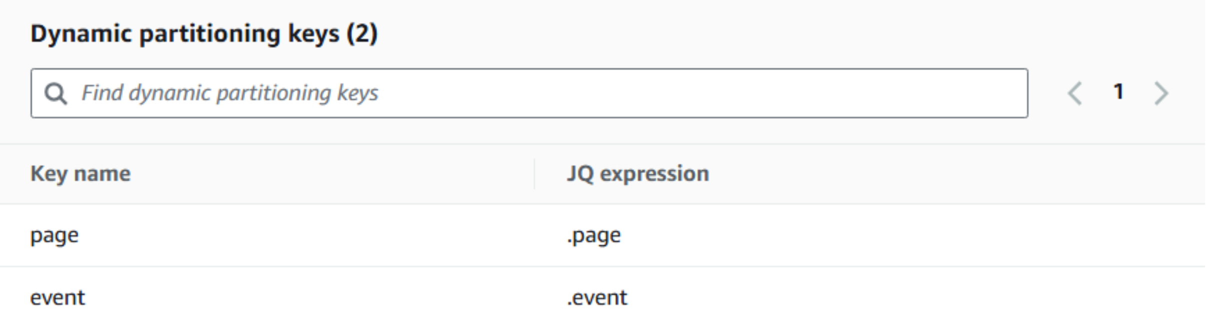
NOTE - There were bunch of other technical specifications in building the data stream, that I did not note here.
Like, we gave values for buffer size and buffer time



Note: Kinesis Data Firehose buffers incoming streaming data to a certain size and for a certain period of time before delivering it to the specified destinations. For a delivery stream where data partitioning is enabled, the buffer size ranges from 64 to 128MB, with the default set to 128MB, and the buffer interval ranges from 60 seconds to 900 seconds. Given the time constraints of this lab, you set the buffer size and buffer interval to the minimum values allowed.

Also we setup IAM role that Kinesis Data Firehose uses to access your S3 bucket.

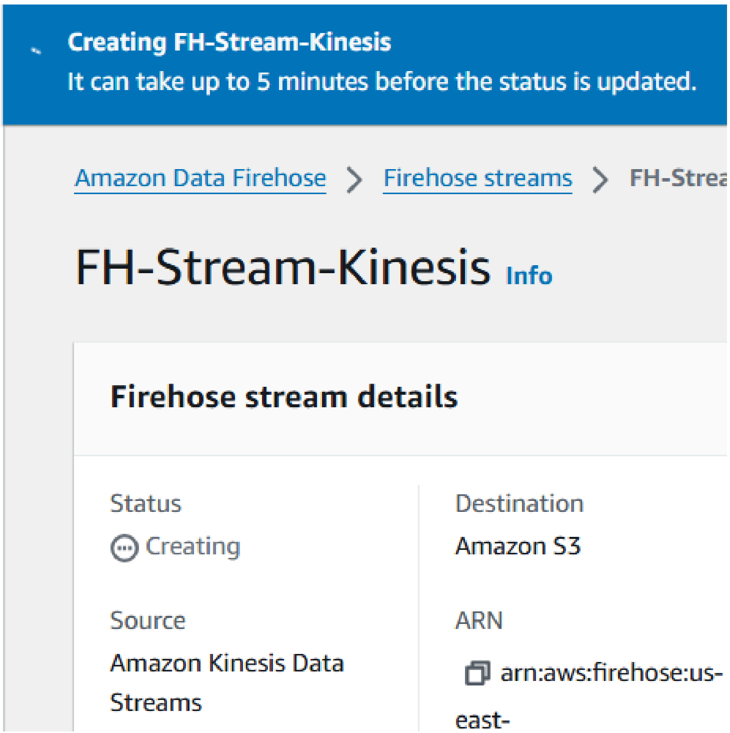
We also setup dynamic partitioning, which means that data stored in S3 will be categorized based on events (type of product in this example).



Dynamic partitioning keys (2)

Key name	JQ expression
page	.page
event	.event

2.3 FH stream creating



Creating FH-Stream-Kinesis

It can take up to 5 minutes before the status is updated.

Amazon Data Firehose > Firehose streams > FH-Strea

FH-Stream-Kinesis Info

Firehose stream details

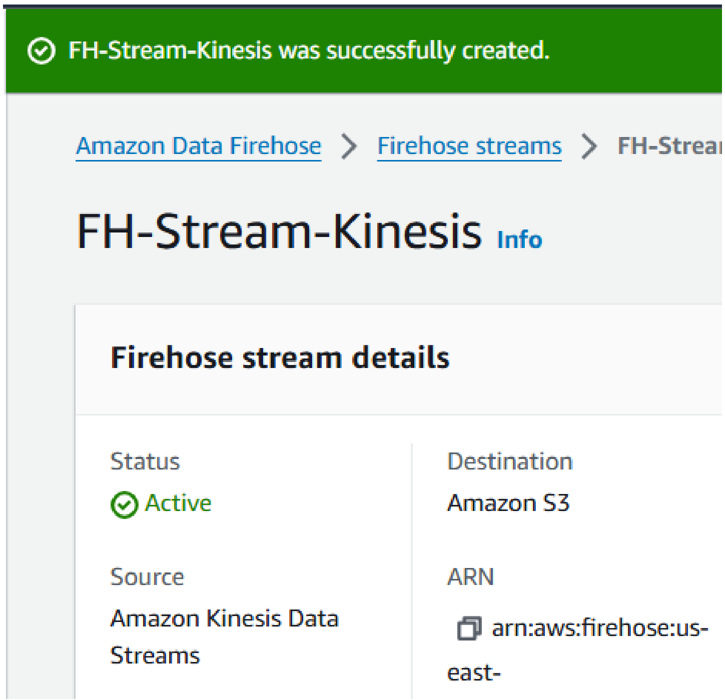
Status: Creating

Destination: Amazon S3

Source: Amazon Kinesis Data Streams

ARN: arn:aws:firehose:us-east-

2.4 Created



FH-Stream-Kinesis was successfully created.

Amazon Data Firehose > Firehose streams > FH-Strea

FH-Stream-Kinesis Info

Firehose stream details

Status: Active

Destination: Amazon S3

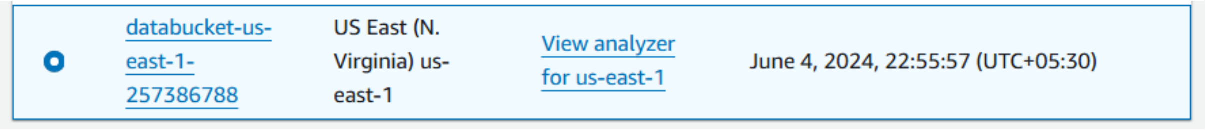
Source: Amazon Kinesis Data Streams

ARN: arn:aws:firehose:us-east-

The output will be stored in Amazon S3 and will be partitioned into page and further into events.

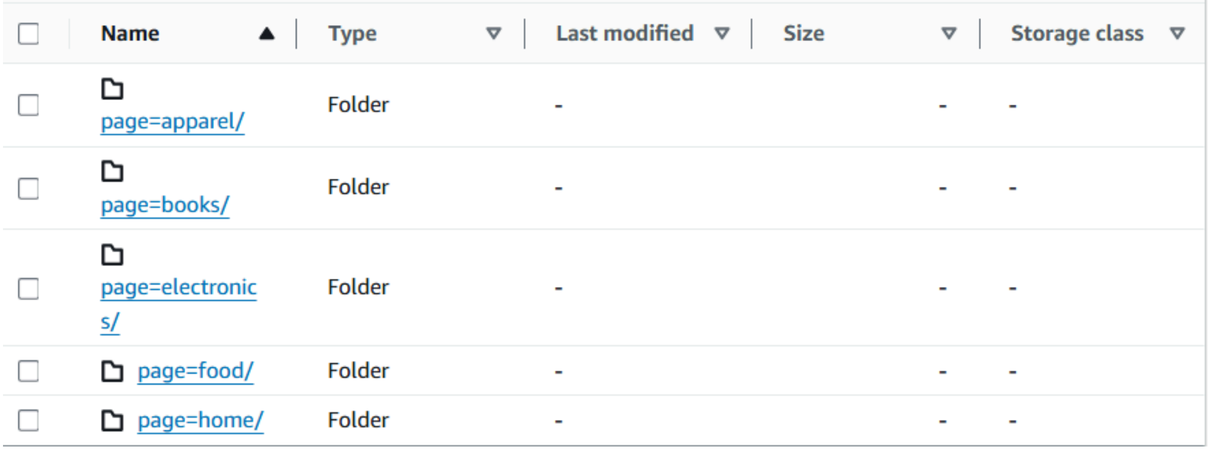
Task 3: Verify your output in Amazon S3

3.1 Open S3 and open this bucket



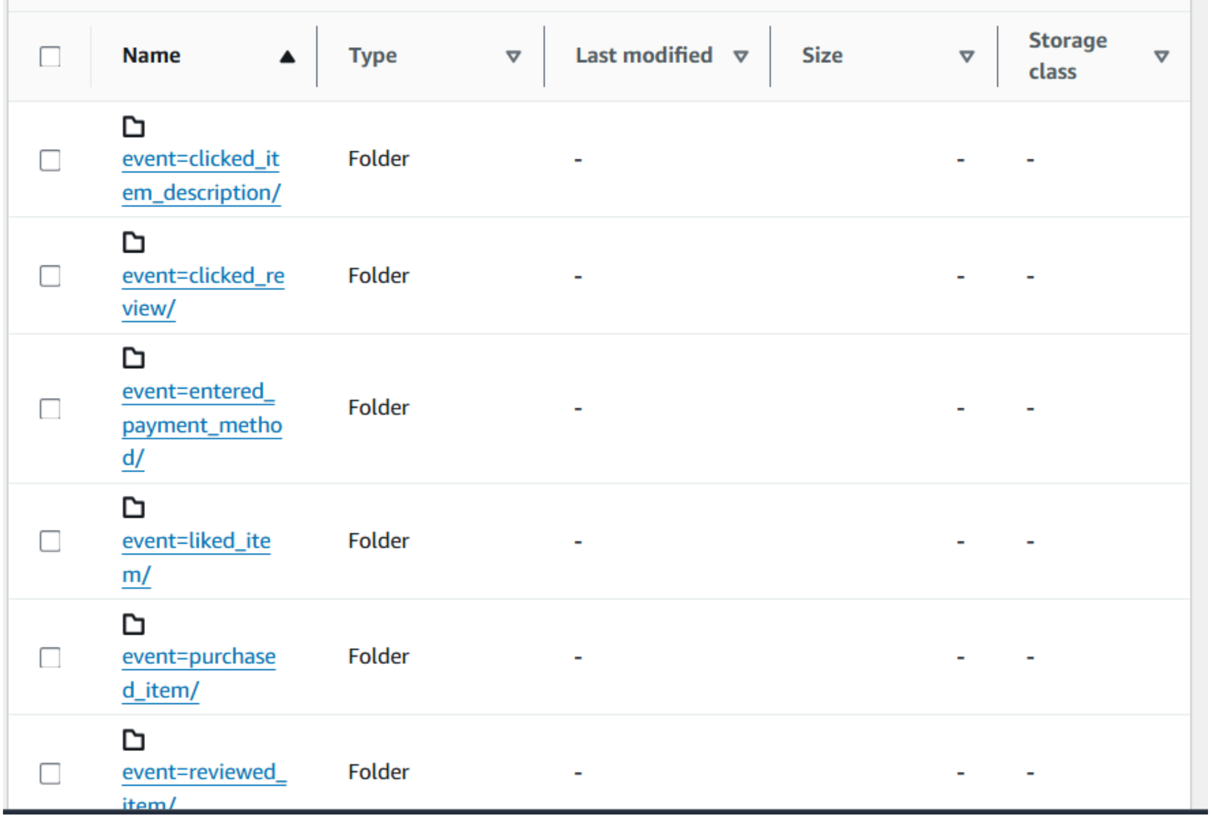
●	databucket-us-east-1-257386788	US East (N. Virginia) us-east-1	View analyzer for us-east-1	June 4, 2024, 22:55:57 (UTC+05:30)

3.2 Folders based on dynamic partitioning appear in the bucket



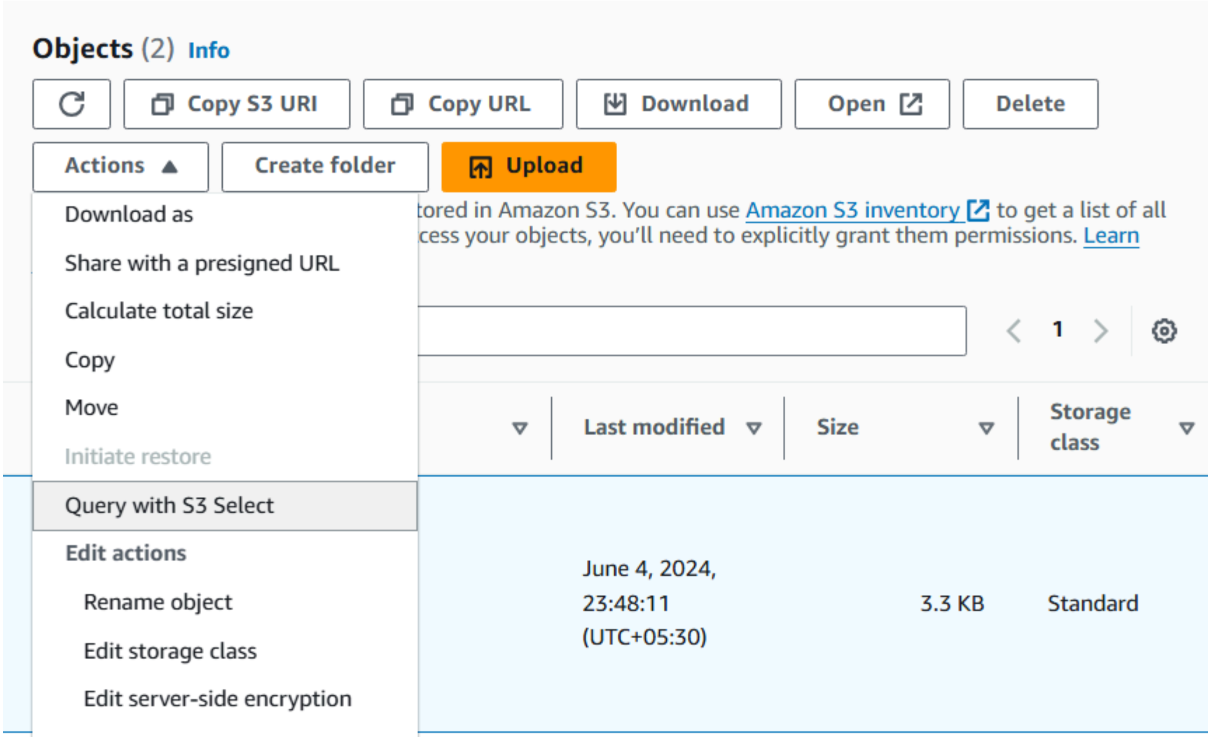
□	Name	Type	Last modified	Size	Storage class
□	page=apparel/	Folder	-	-	-
□	page=books/	Folder	-	-	-
□	page=electronics/	Folder	-	-	-
□	page=food/	Folder	-	-	-
□	page=home/	Folder	-	-	-

3.3 Inside every page (catoegory in this example), there are following events



□	Name	Type	Last modified	Size	Storage class
□	event=clicked_item_description/	Folder	-	-	-
□	event=clicked_review/	Folder	-	-	-
□	event=entered_payment_method/	Folder	-	-	-
□	event=liked_item/	Folder	-	-	-
□	event=purchased_item/	Folder	-	-	-
□	event=reviewed_item/	Folder	-	-	-

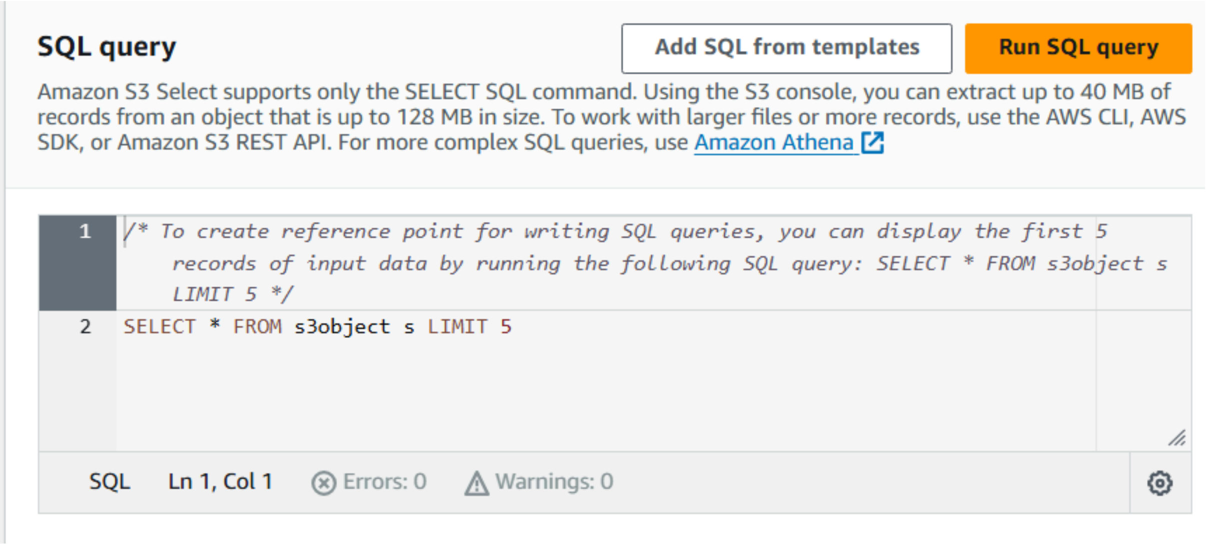
3.4 Select any item inside an event, and choose:-



tored in Amazon S3. You can use Amazon S3 inventory to get a list of all
cess your objects, you'll need to explicitly grant them permissions. Learn

Last modified	Size	Storage class
June 4, 2024, 23:48:11 (UTC+05:30)	3.3 KB	Standard

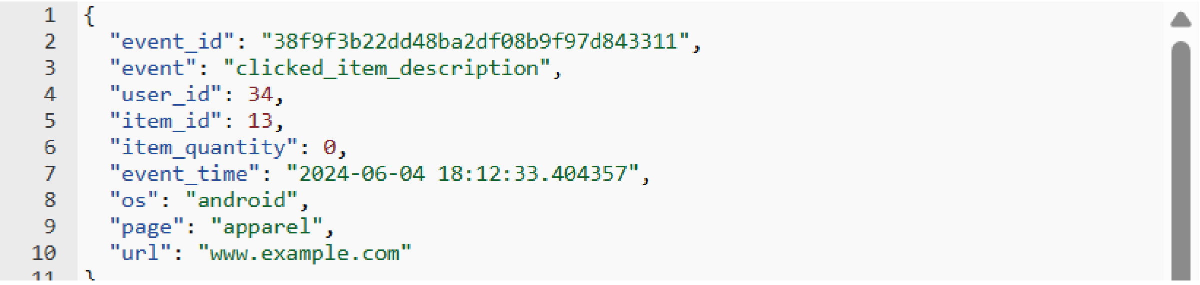
3.5 Run the following SQL query



Amazon S3 Select supports only the SELECT SQL command. Using the S3 console, you can extract up to 40 MB of records from an object that is up to 128 MB in size. To work with larger files or more records, use the AWS CLI, AWS SDK, or Amazon S3 REST API. For more complex SQL queries, use Amazon Athena

```sql
/* To create reference point for writing SQL queries, you can display the first 5
    records of input data by running the following SQL query: SELECT * FROM s3object s
    LIMIT 5 */
SELECT * FROM s3object s LIMIT 5
```

3.6 Example output



```
{
  "event_id": "38f9f3b22dd48ba2df08b9f97d843311",
  "event": "clicked_item_description",
  "user_id": 34,
  "item_id": 13,
  "item_quantity": 0,
  "event_time": "2024-06-04 18:12:33.404357",
  "os": "android",
  "page": "apparel",
  "url": "www.example.com"
}
```