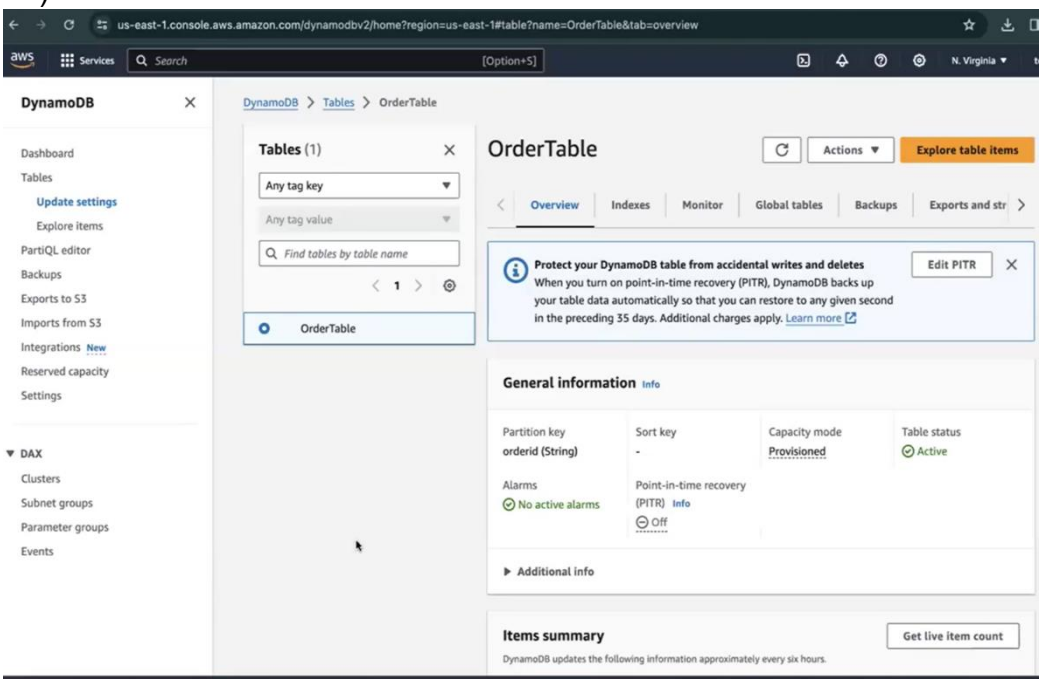
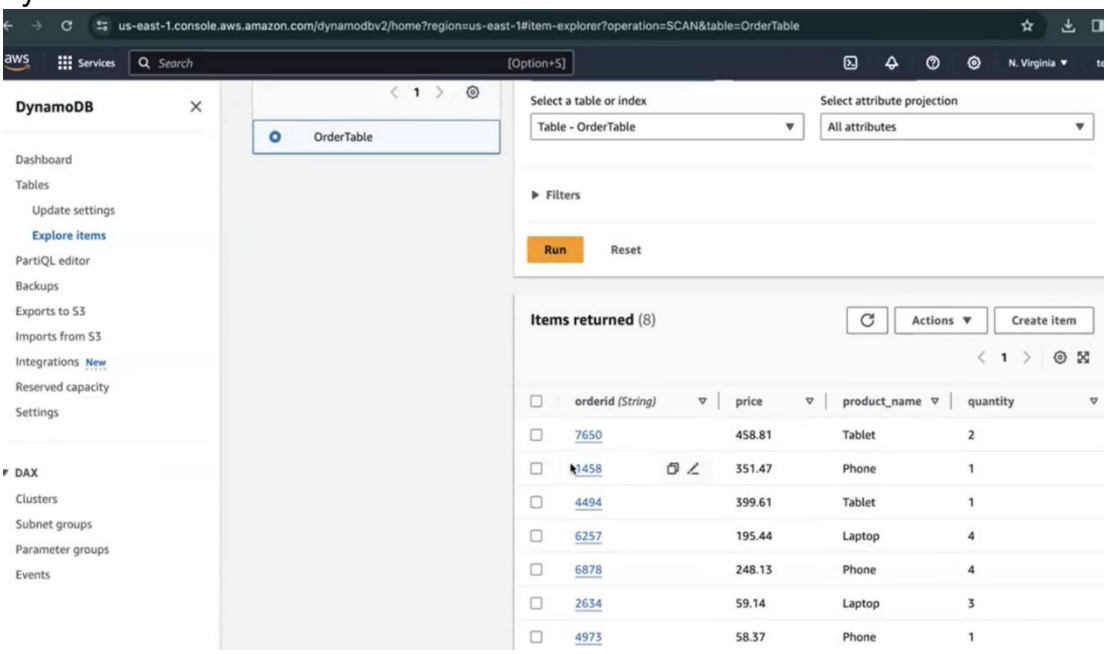
# Sales Data Projection

1. this was a sales data streaming project for processing the data from Dynamo db to destination S3 target path.
2. there was some source from where records(orderid, product_name, quantity, price) were being written into the nosql dynamodb(which is a key value based db).

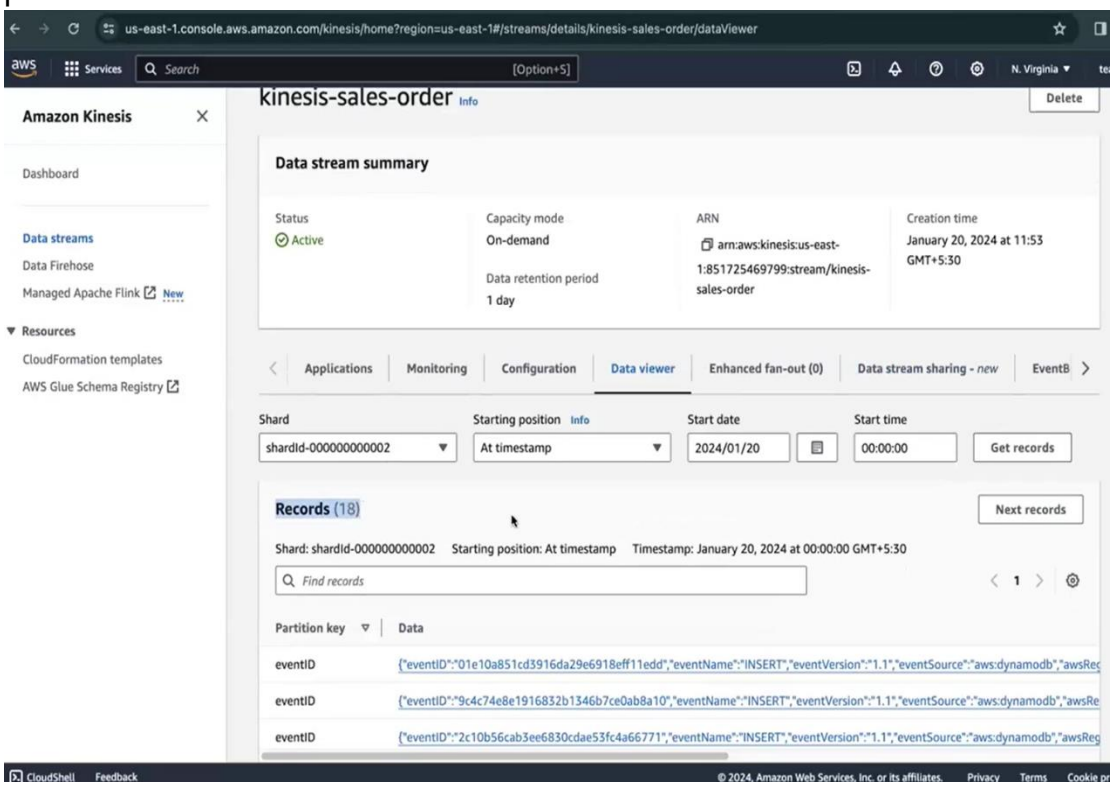


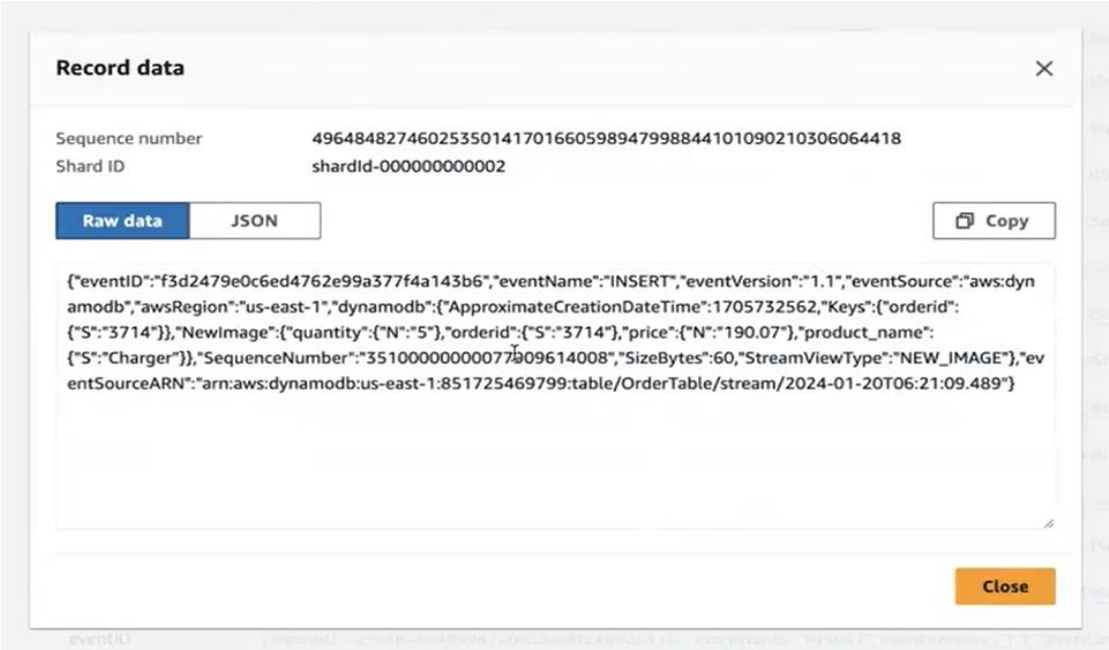

3. we created a dynamodb table where we targeted particular partition key(orderid) and the entire data(orderid, product_name, quantity, price) used to get ingested as a value there. That's how we were publishing a record in dynamodb.



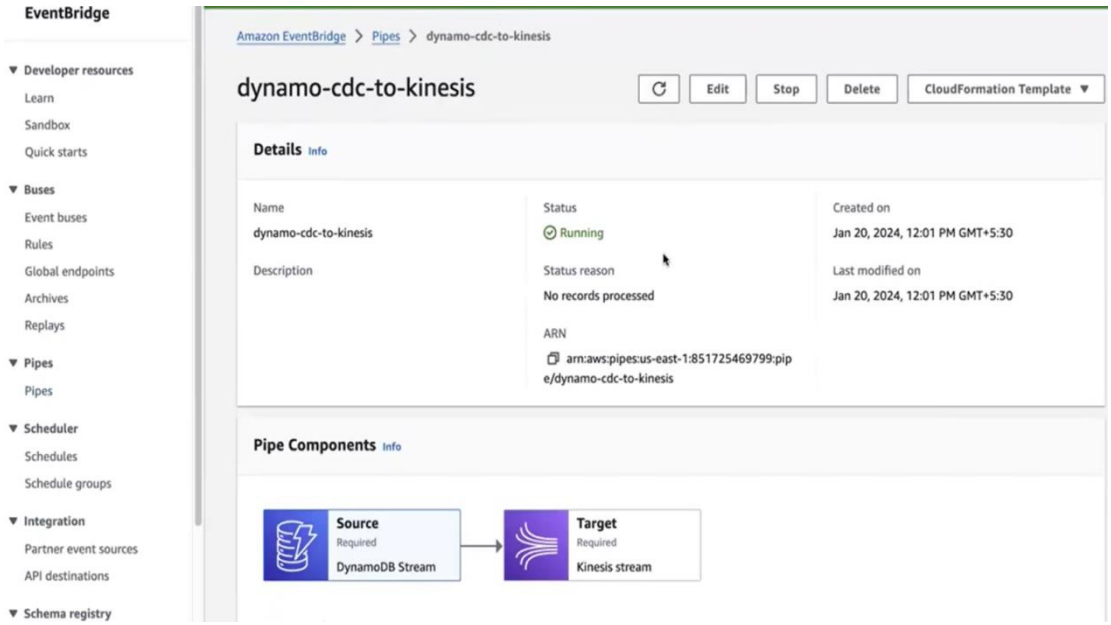
4. next our task was to capture changes happening in table. Records were getting inserted, updated/deleted to table in real time. CDC is whatever changes are happening in real time we capture those changes. That's what CDC is. So we needed to enable dynamodb streams from 'Exports & Streams' option to perform CDC. As we were able to consume the CDC changes, hence created CDC pipeline.

5. then we set up a kinesis stream which is a queuing mechanism to hold our data. Kinesis stream has a concept of shards just like publishing data to partitions. Data here arrives at kinesis stream in different shard.

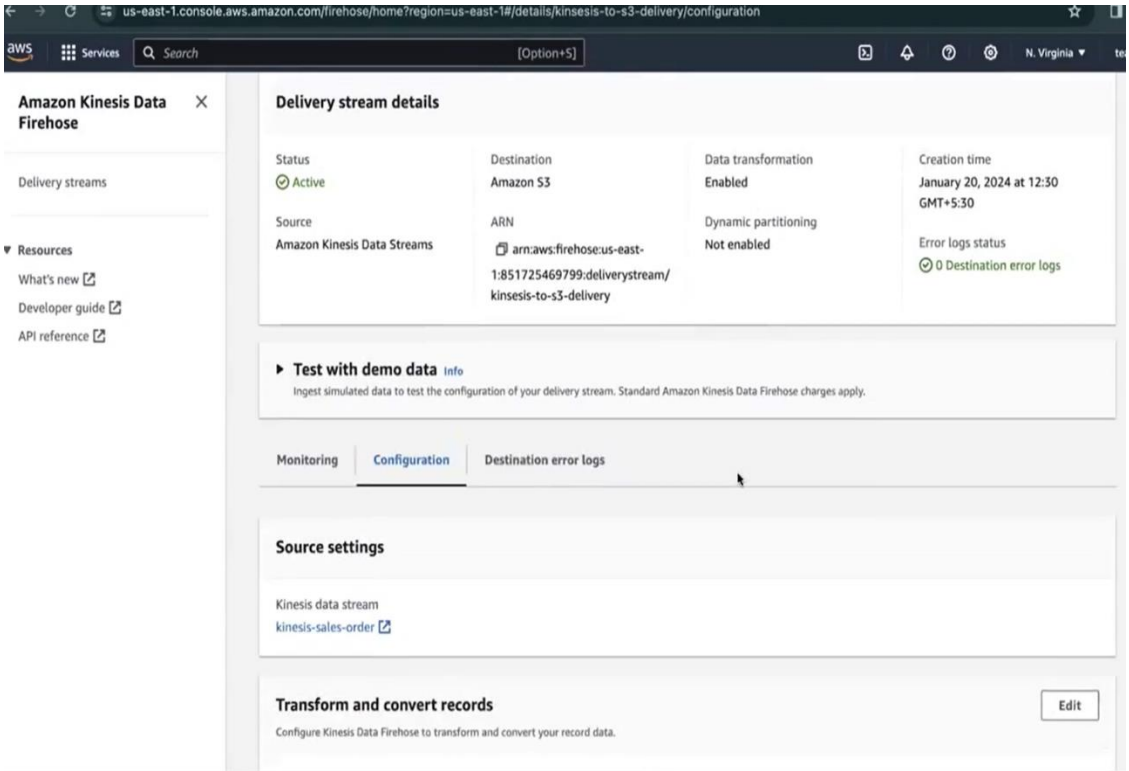


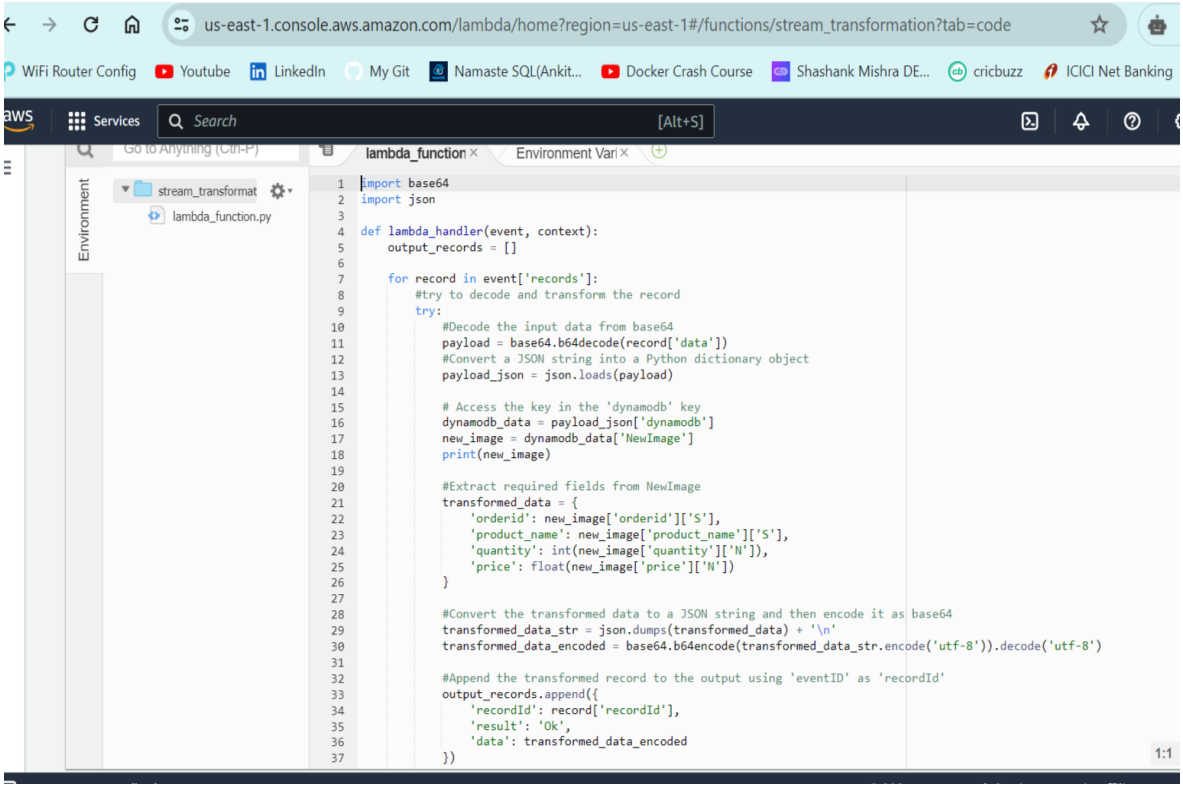
6. then we had a event bridge pipe in between dynamodb stream and kinesis stream. We set a batch size consisting of no of message to be sent to any kinesis stream shard. Set up required IAM role to eventbridge pipe to access dynamo stream & kinesis stream. Using event bridge we used to flow data to kinesis stream.



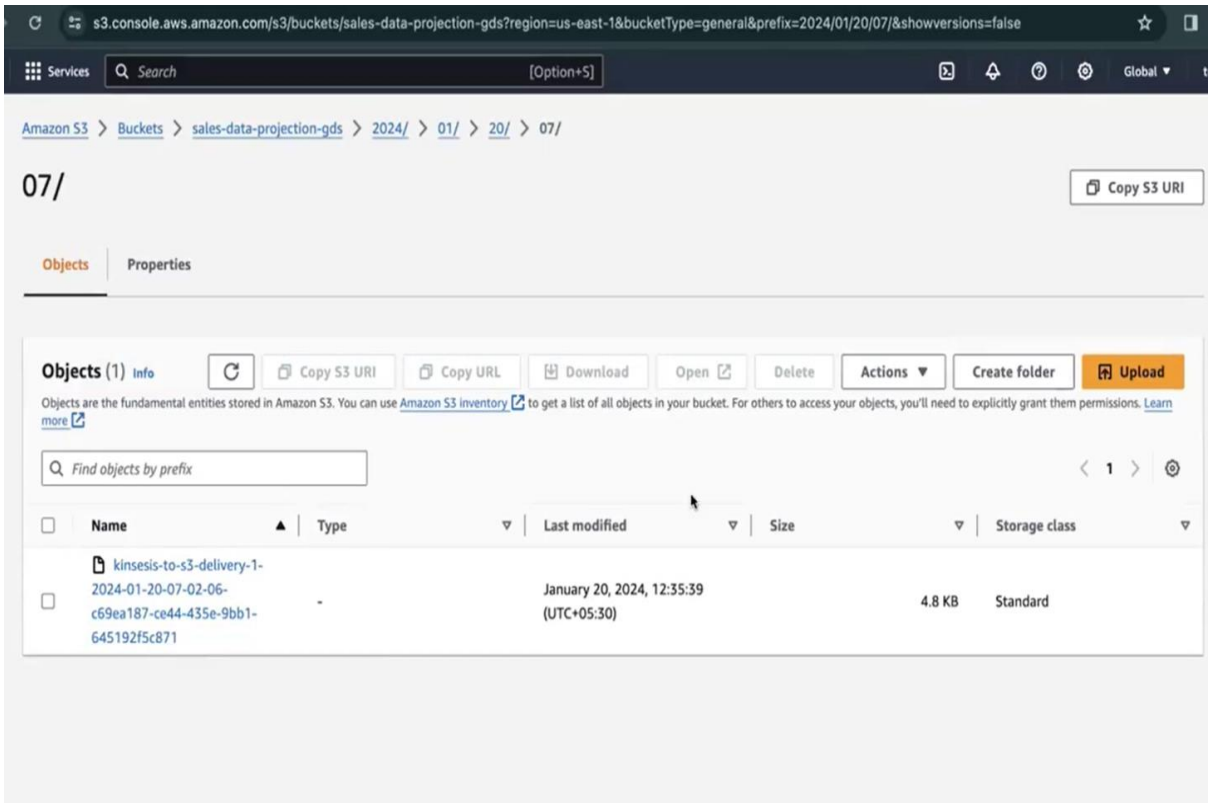

7. next we created a firehose stream which is a delivery stream. It processed data(like ingest, transform) and delivered streaming data to destinations(like data lakes, data warehouses) in batches based on buffer size or buffer time factor. Here source is kinesis stream and destination is s3.



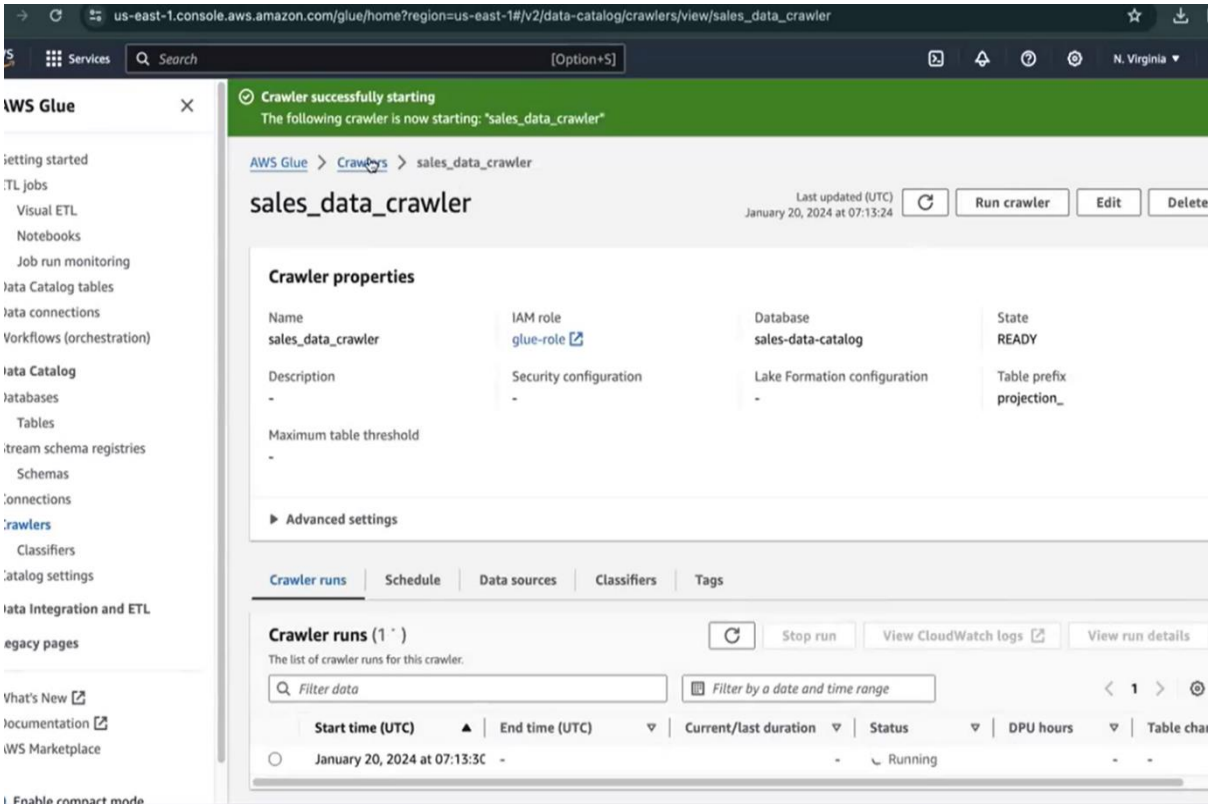
8. So Kinesis firehose delivery stream was loading data from source Kinesis stream, holding that streaming data until the buffer size/buffer time is reached & then data is sent to lambda function in a batch. Following this set up **Firehose** to invoke our **Lambda function** to **transform** incoming batch records this way.



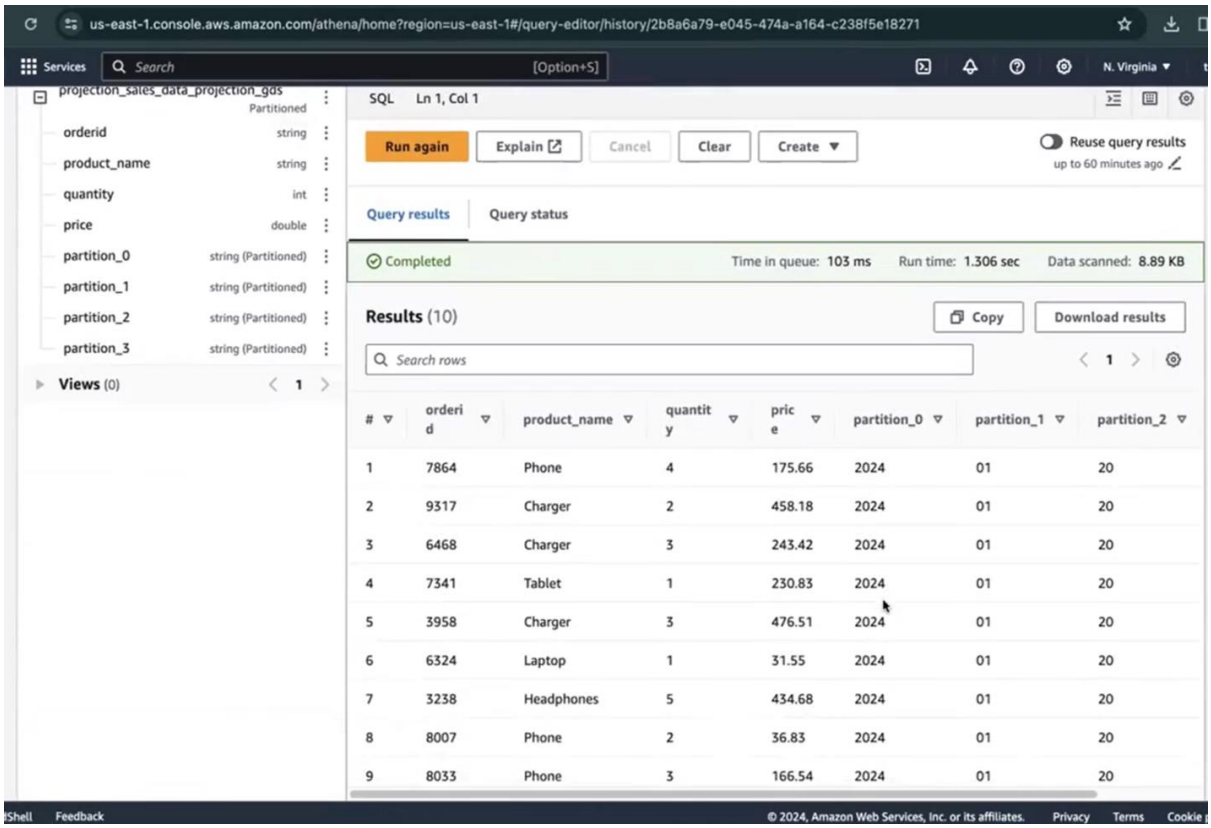

9. Once this is done, then delivering the transformed batch records to S3 destination.



10. Lastly, we created crawler over s3 target path which resulting in catalog table creation. Also, needed to add classifier for json format(i.e. $.columns).




11. using athena query we then did analysis




| # | orderid | product_name | quantity | price | partition_0 | partition_1 | partition_2 |
|---|---|---|---|---|---|---|---|
| 1 | 7864 | Phone | 4 | 175.66 | 2024 | 01 | 20 |
| 2 | 9317 | Charger | 2 | 458.18 | 2024 | 01 | 20 |
| 3 | 6468 | Charger | 3 | 243.42 | 2024 | 01 | 20 |
| 4 | 7341 | Tablet | 1 | 230.83 | 2024 | 01 | 20 |
| 5 | 3958 | Charger | 3 | 476.51 | 2024 | 01 | 20 |
| 6 | 6324 | Laptop | 1 | 31.55 | 2024 | 01 | 20 |
| 7 | 3238 | Headphones | 5 | 434.68 | 2024 | 01 | 20 |
| 8 | 8007 | Phone | 2 | 36.83 | 2024 | 01 | 20 |
| 9 | 8033 | Phone | 3 | 166.54 | 2024 | 01 | 20 |