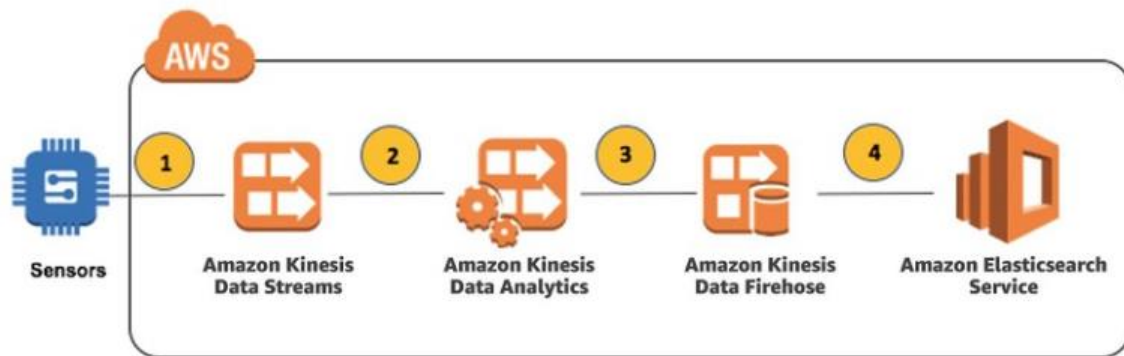


Solution overview

The following diagram depicts a high-level overview of this solution.

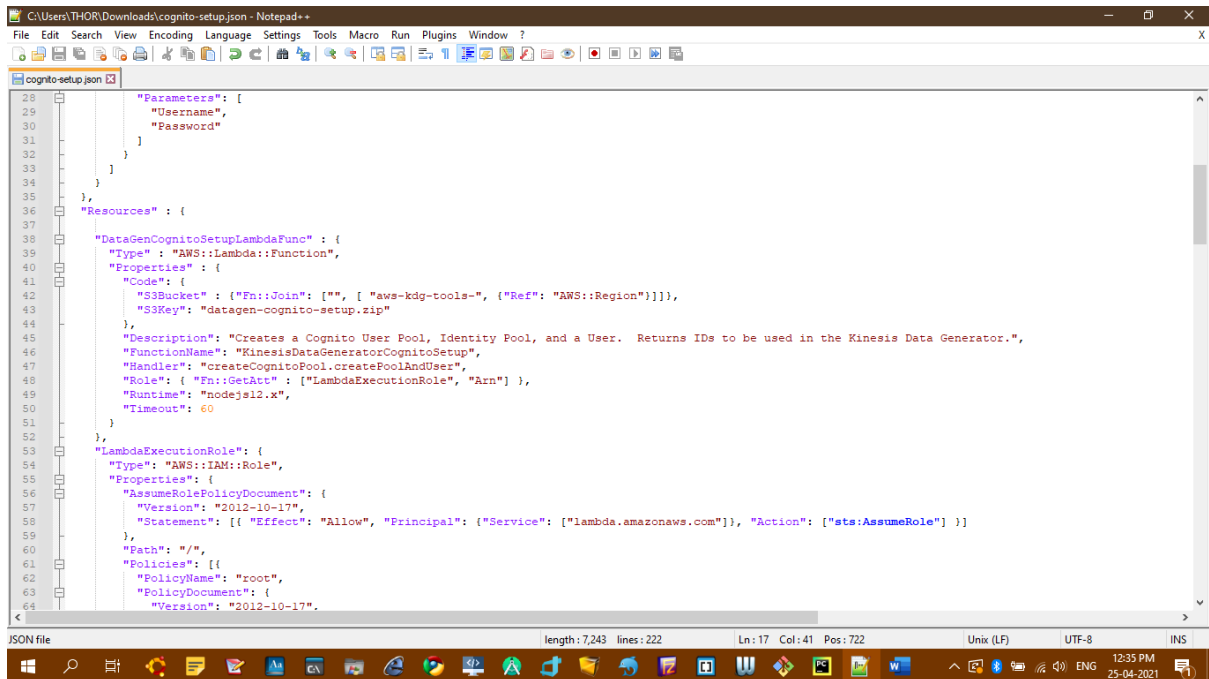


Steps to perform:

1. Create Kinesis delivery stream
2. Simulate streaming application to detect anomalies
3. Open Elasticsearch service and create a new domain
4. Configure Kinesis Firehose to export the results to Amazon ES
5. Update the buffer size and existing IAM role for the process
6. Open the Amazon Kinesis Analytics console and create a new application
7. Connect to the source for further analysis
8. Launch SQL_Editor and start the application
9. Load the processed data into Kinesis Firehose delivery stream
10. Visualize the data using Kibana

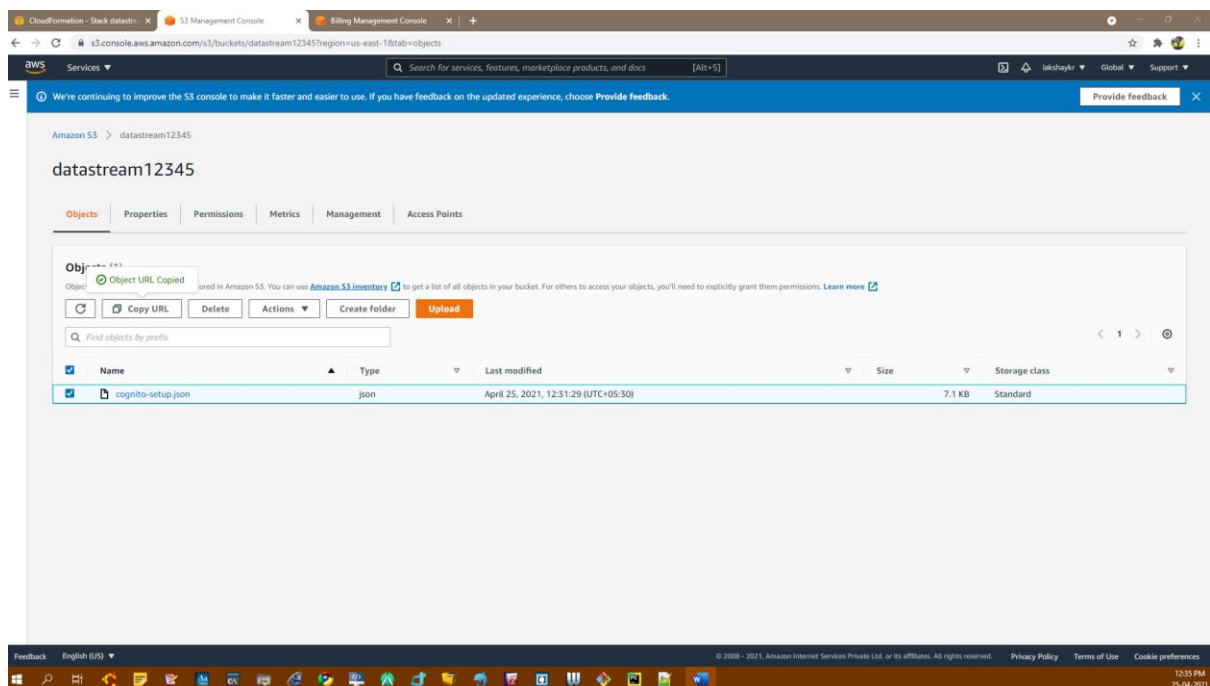
Note: All the below steps performed in my Personal AWS Account .

1. Below in the .json script which I used to create the KDG "Kinesis Data Generator" and Congnito(Kinesis Data Generator-Usrs) /Identities pool



```
28      "Parameters": [
29        "Username",
30        "Password"
31      ],
32    },
33  },
34  },
35  },
36  "Resources": {
37    "DataGenCognitoSetupLambdaFunc": {
38      "Type": "AWS::Lambda::Function",
39      "Properties": {
40        "Code": {
41          "S3Bucket": { "Fn::Join": [ "", [ { "aws-kdg-tools-", { "Ref": "AWS::Region" } ] ] ],
42          "S3Key": "datagen-cognito-setup.zip"
43        },
44        "Description": "Creates a Cognito User Pool, Identity Pool, and a User. Returns IDs to be used in the Kinesis Data Generator.",
45        "FunctionName": "KinesisDataGeneratorCognitoSetup",
46        "Handler": "createCognitoPool.createPoolAndUser",
47        "Role": { "Fn::GetAtt": [ "LambdaExecutionRole", "Arn" ] },
48        "Runtime": "nodejs12.x",
49        "Timeout": 60
50      },
51    },
52    "LambdaExecutionRole": {
53      "Type": "AWS::IAM::Role",
54      "Properties": {
55        "AssumeRolePolicyDocument": {
56          "Version": "2012-10-17",
57          "Statement": [ { "Effect": "Allow", "Principal": { "Service": [ "lambda.amazonaws.com" ] }, "Action": [ "sts:AssumeRole" ] } ]
58        },
59        "Path": "/",
60        "Policies": [ {
61          "PolicyName": "root",
62          "PolicyDocument": {
63            "Version": "2012-10-17",
```

- 2.Uploaded .json script in the S3 Account and make the that Object Public accessible



- 3.Upload the .json template from the S3 to Cloud Formation and created the Stack to setup the KDG Account and Congito account with users

Stacks (1)

datastream
2021-04-25 12:32:11 UTC+0530
CREATE_COMPLETE

datastream

Stack info | **Events** | Resources | Outputs | Parameters | Template | Change sets

Events (17)

Timestamp	Logical ID	Status	Status reason
2021-04-25 12:32:58 UTC+0530	datastream	CREATE_COMPLETE	-
2021-04-25 12:32:56 UTC+0530	SetupCognitoCustom	CREATE_COMPLETE	-
2021-04-25 12:32:56 UTC+0530	SetupCognitoCustom	CREATE_IN_PROGRESS	Resource creation Initiated
2021-04-25 12:32:51 UTC+0530	SetupCognitoCustom	CREATE_IN_PROGRESS	-
2021-04-25 12:32:49 UTC+0530	DataGenCognitoSetupLambdaFunc	CREATE_COMPLETE	-
2021-04-25 12:32:49 UTC+0530	DataGenCognitoSetupLambdaFunc	CREATE_IN_PROGRESS	Resource creation Initiated
2021-04-25 12:32:48 UTC+0530	DataGenCognitoSetupLambdaFunc	CREATE_IN_PROGRESS	-
2021-04-25 12:32:46 UTC+0530	LambdaExecutionRole	CREATE_COMPLETE	-
2021-04-25 12:32:31 UTC+0530	LambdaExecutionRole	CREATE_IN_PROGRESS	Resource creation Initiated
2021-04-25 12:32:31 UTC+0530	LambdaExecutionRole	CREATE_IN_PROGRESS	-
2021-04-25 12:32:29 UTC+0530	UnauthenticatedUserRole	CREATE_COMPLETE	-
2021-04-25 12:32:29 UTC+0530	AuthenticatedUserRole	CREATE_COMPLETE	-
2021-04-25 12:32:15 UTC+0530	UnauthenticatedUserRole	CREATE_IN_PROGRESS	Resource creation Initiated
2021-04-25 12:32:15 UTC+0530	AuthenticatedUserRole	CREATE_IN_PROGRESS	Resource creation Initiated

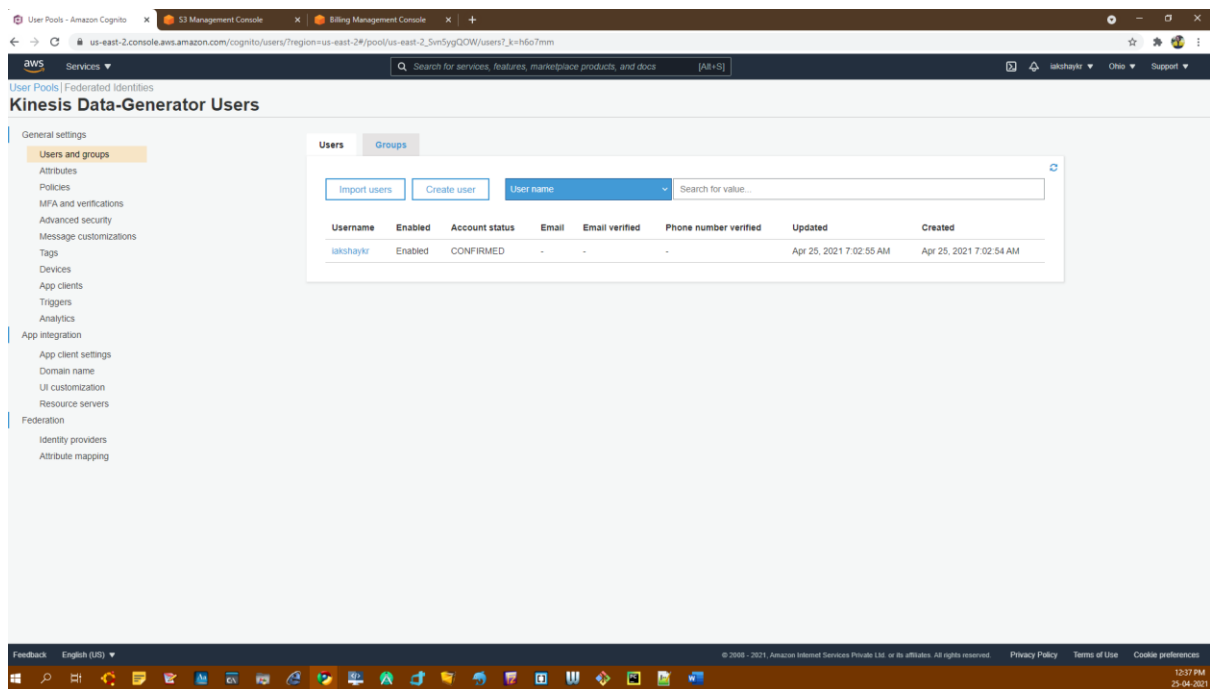
Stacks (1)

datastream
2021-04-25 12:32:11 UTC+0530
CREATE_COMPLETE

Stacks

Stack name	Status	Created time	Description
datastream	CREATE_COMPLETE	2021-04-25 12:32:11 UTC+0530	This template creates an Amazon Cognito User Pool and Identity Pool, with a single user. It assigns a role to authenticated users in the identity pool to enable the users to use the Kinesis Data Generator tool.

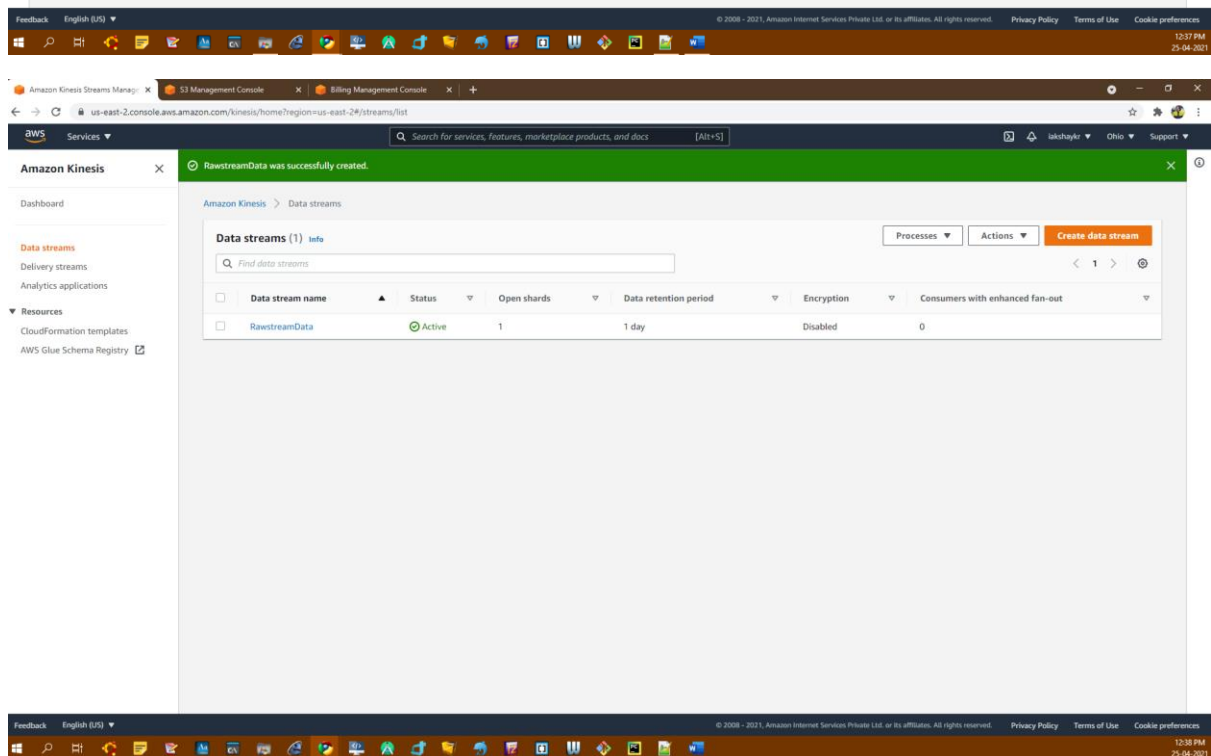
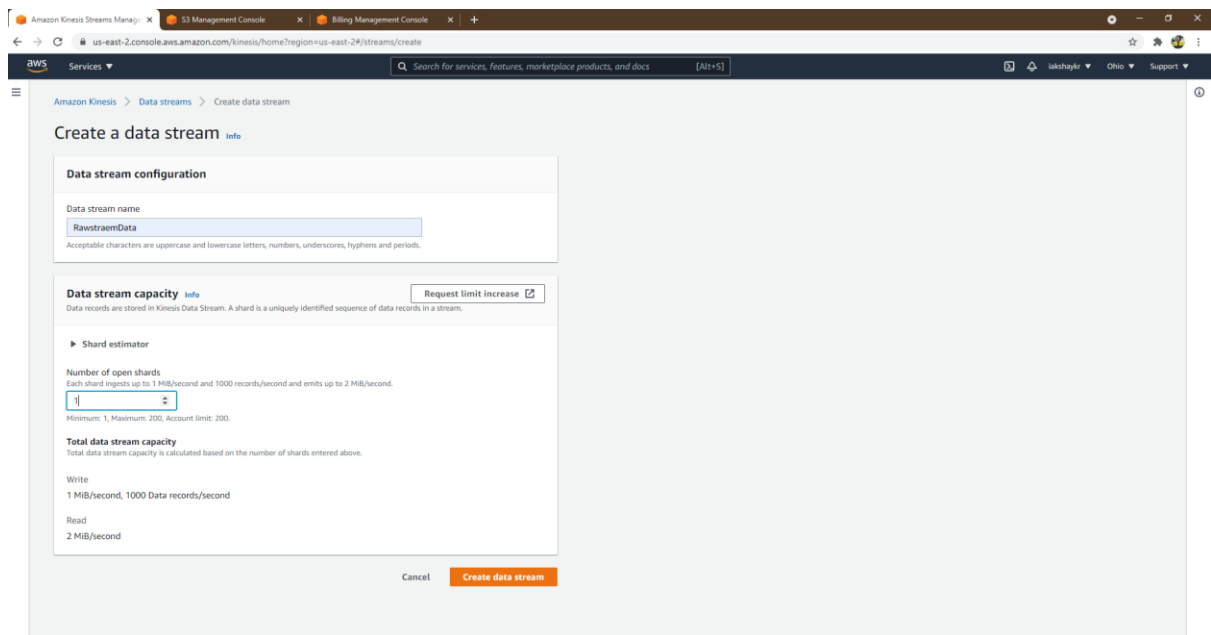
4. Successfully created the KDG-Users and Identities Pool and added user “iakshaykr” which will help later to login into the Kibana (ES)



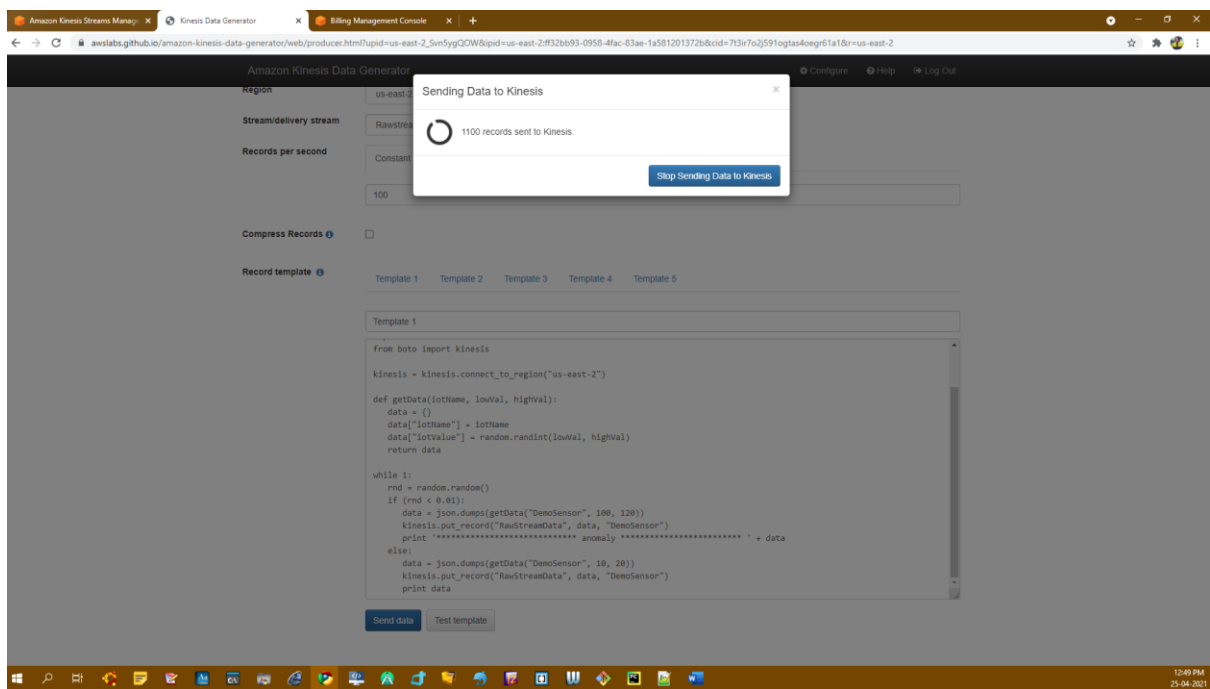
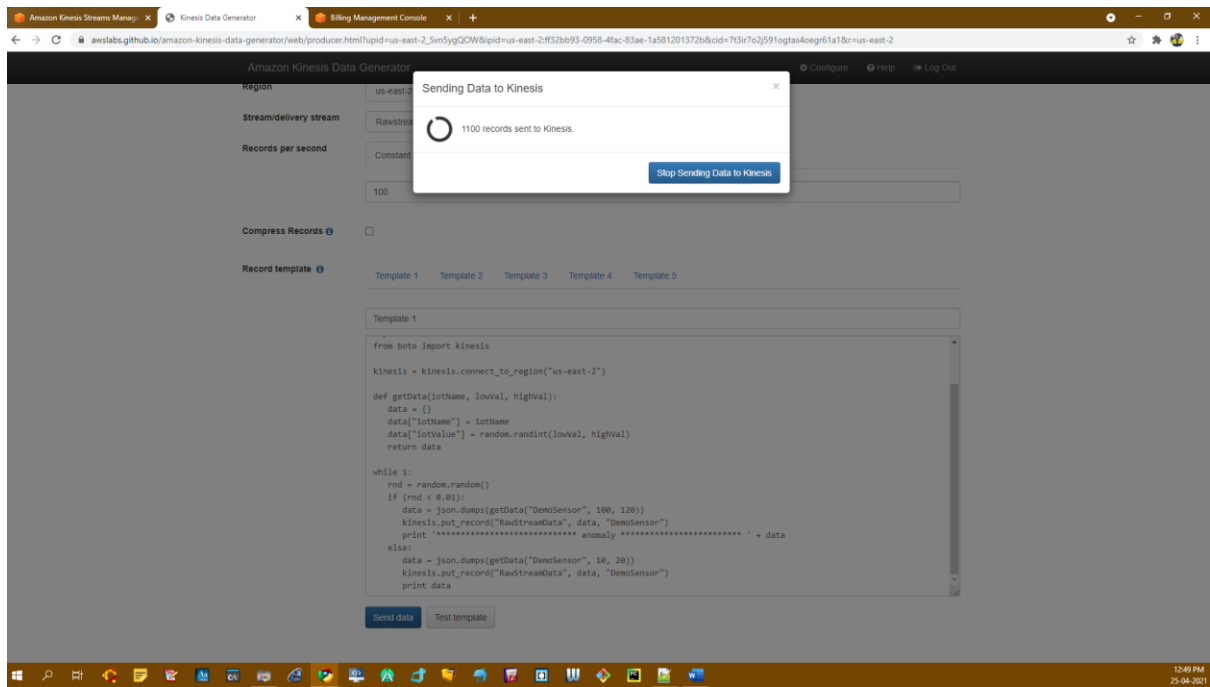
Kinesis Data Stream :

5. Created the Data Stream to get the Generated data from “Kinesis Data Generator”

Taken 1 Shard only.



5. Sending the Data from Kinesis Data Generator to “RawStreamData” which is a Kinesis Data Stream



6. Created the Elasticsearch Domain which is used to setup the Kibana and as well Kinesis Firehouse .

The image consists of two screenshots from the AWS Management Console. The top screenshot shows the 'Create Elasticsearch domain' wizard, specifically the 'Choose deployment type' step. The 'Development and testing' option is selected under 'Deployment type', and '7.10 (latest)' is selected for the 'Elasticsearch version'. The bottom screenshot shows the 'Amazon Elasticsearch Service dashboard'. A table lists the domains, with one domain named 'rasterdata' shown. The dashboard also includes sections for 'Learning content' and 'Feature Spotlight'.

Create Elasticsearch domain

Step 1: Choose deployment type
Step 2: Configure domain
Step 3: Configure access and security
Step 4: Add tags - optional
Step 5: Review

Choose deployment type

Deployment types specify common settings for your use case. After creating the domain, you can change these settings at any time.

Deployment type

- ☐ Production
Multiple Availability Zones and dedicated master nodes for higher availability.
- ☒ Development and testing
One Availability Zone for when you just need an Elasticsearch endpoint.
- ☐ Custom
Choose settings from all available options.

Version

Select the version of Elasticsearch for your domain.

Elasticsearch version: 7.10 (latest)

Cancel Next

Amazon Elasticsearch Service dashboard

Create a new domain

My Elasticsearch domains

Domain	Elasticsearch version	Endpoint	Searchable documents	Elasticsearch cluster health	Free storage space	Minimum free storage space	UltraWarm storage usage	Domain status
rasterdata	7.10	Internet					Disabled	Loading

Learning content

- Set Alerts in Amazon Elasticsearch Service**
Learn how to monitor your log data and set thresholds and alerts. [Learn more](#)
- Upload Data**
Step-by-step tutorial to upload data to an Amazon Elasticsearch Service domain. [Learn more](#)
- UltraWarm for Amazon Elasticsearch Service**
Step-by-step tutorial to help you enable UltraWarm storage tier. [Learn more](#)

Feature Spotlight

- Fine-Grained Data Access**
Define granular permissions for indices, documents, or fields and extend Kibana with read-only views. [Learn more](#)
- UltraWarm for Amazon Elasticsearch Service**
Store and interactively analyze petabytes of logs at 1/10th the cost of your existing storage tiers. [Learn more](#)
- k-Nearest Neighbor (k-NN) Search**
Easily enable high scale, low latency nearest neighbor search on billions of documents across thousands of dimensions. [Learn more](#)

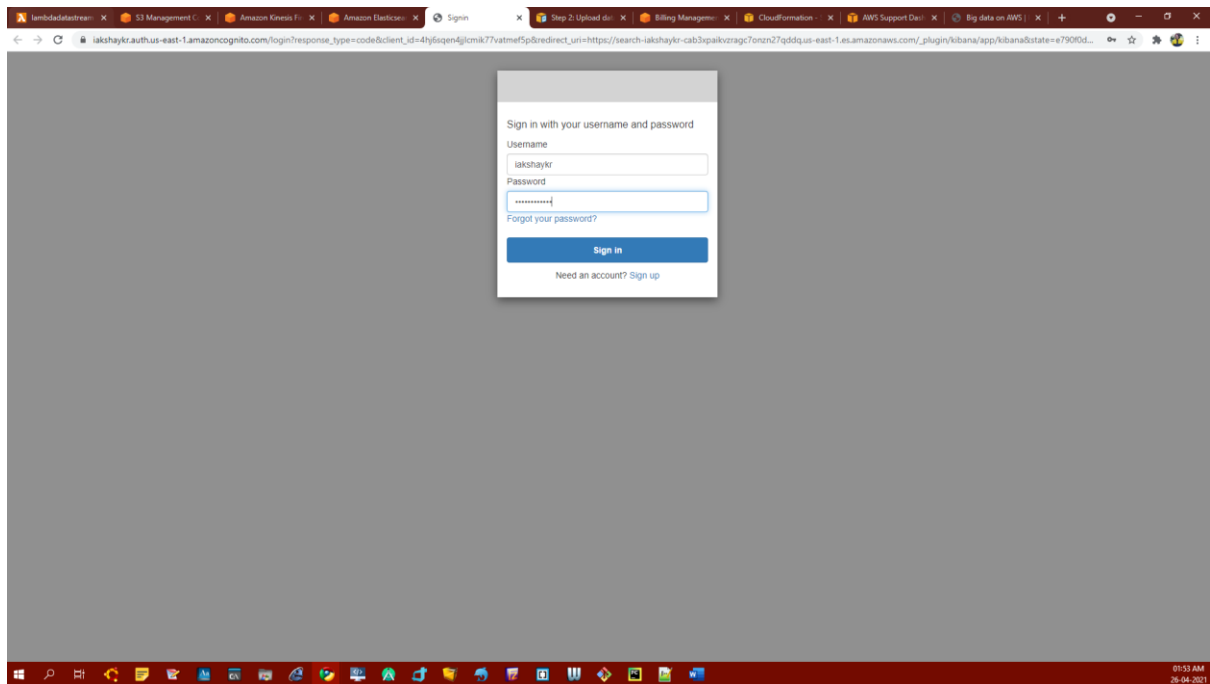
7. Created the Data Analytics to visualize the incoming data/throughput



8. Setting up the Kinesis Firehose to deliver the stream data to Elasticsearch domain to visualize into Kibana

The first screenshot shows the 'Kinesis Data Firehose - Create delivery stream' wizard. In Step 1: Name and source, the 'New delivery stream' section has 'RawstreamData' entered as the stream name. The 'Choose a source' section has 'Direct PUT or other sources' selected. A diagram titled 'Firehose data flow overview' shows data flowing from a 'Source' to a 'Firehose delivery stream' (containing 'Source records' and 'Processed records') and then to a 'Destination'. The second screenshot shows the 'bigdataaaws' Elasticsearch domain in the 'Overview' tab. Key details include: Domain status: Active; Elasticsearch version: 7.10; Endpoint: https://search-bigdataaaws-i3wema4u2npo5bn3y2bgwft4.us-east-1.es.amazonaws.com; Domain ARN: arn:aws:es:us-east-1:229161520992:domain/bigdataaaws; Availability zones: 1; Instance type (data): r5.large.elasticsearch; Number of nodes: 1; Data nodes storage type: EBS; EBS volume type: General Purpose (SSD); EBS volume size: 10 GiB; Auto-Tune: Enabled; Maintenance window: Disabled; Upgrade status: -; Start hour for the daily automated snapshot: 00:00 UTC (default); Fine-grained access control: Enabled.

9. Login into Kibana using the cognito user which we have created in setup 2nd and 3rd using Kinesis Data Generators-User and Identities



10. Logged in successfully into Kibana to visualize the data coming from data-firehouse which is worked as a delivery system .

