



Vivekanand Education Society's

Institute of Technology

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

Department of Information Technology

IOE Lab

Lab Assignment - 2

Aim: Explore AWS Analytics tools.

Roll No.	70
Name	MAYURI SHRIDATTA YERANDE
Class	D20B
Subject	Internet of Everything
Grade:	

AIM: Explore AWS Analytics tools.

THEORY:

IoT (Internet of Things) and AWS (Amazon Web Services) Analytics are two important and interconnected concepts in the world of technology and data processing. Let's explore the theory behind IoT and AWS Analytics.

Internet of Things (IoT)

IoT refers to the network of physical objects or "things" embedded with sensors, software, and other technologies that enable them to collect and exchange data with other devices and systems over the internet. The key components of IoT include:

AWS Analytics for IoT

AWS provides a comprehensive suite of services for handling IoT data and performing analytics. Here's an overview of key AWS services and components relevant to IoT analytics:

AWS IoT Core: This service enables secure and scalable communication between IoT devices and the AWS cloud. It manages device connections, handles authentication, and allows you to route data to various AWS services.

IoT Analytics: AWS IoT Analytics is a service that allows you to process and analyze IoT data at scale. It supports SQL-based querying, data transformation, and integration with other AWS services like Amazon S3, AWS Lambda, and Amazon QuickSight.

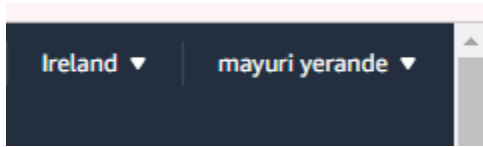
IoT Analytics Workflow on AWS

1. IoT devices collect data and transmit it to AWS IoT Core.
2. AWS IoT Core routes the data to relevant services like AWS IoT Analytics, Amazon Kinesis, or others.
3. AWS IoT Analytics processes and transforms the data.
4. Processed data can be stored in Amazon S3, analyzed in Redshift, or visualized in QuickSight.
5. AWS Lambda functions can be used for real-time actions or alerts based on the data.
6. Users can access insights and reports through QuickSight dashboards.

In summary, AWS offers a robust ecosystem of services to handle IoT data from end to end, making it possible to collect, store, process, analyze, and visualize data generated by IoT devices, ultimately enabling organizations to derive valuable insights and make data-driven decisions.

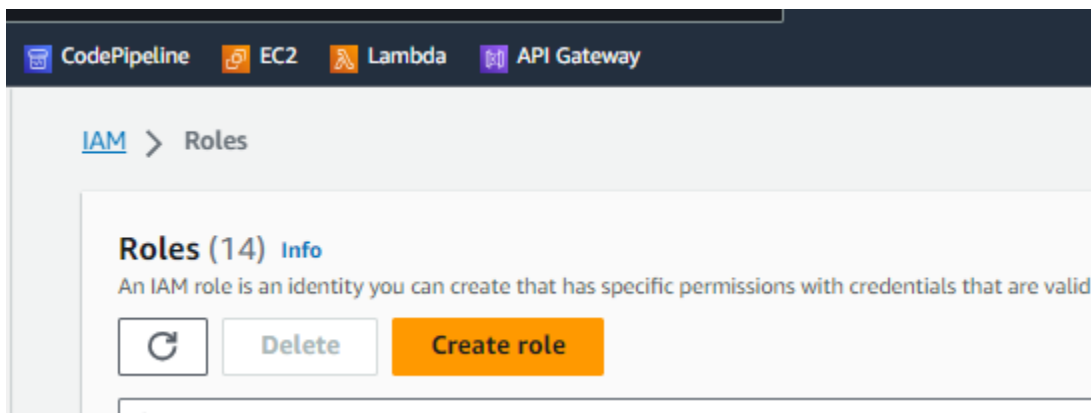
IMPLEMENTATION:

- Login to AWS Management Console and change the region to Ireland

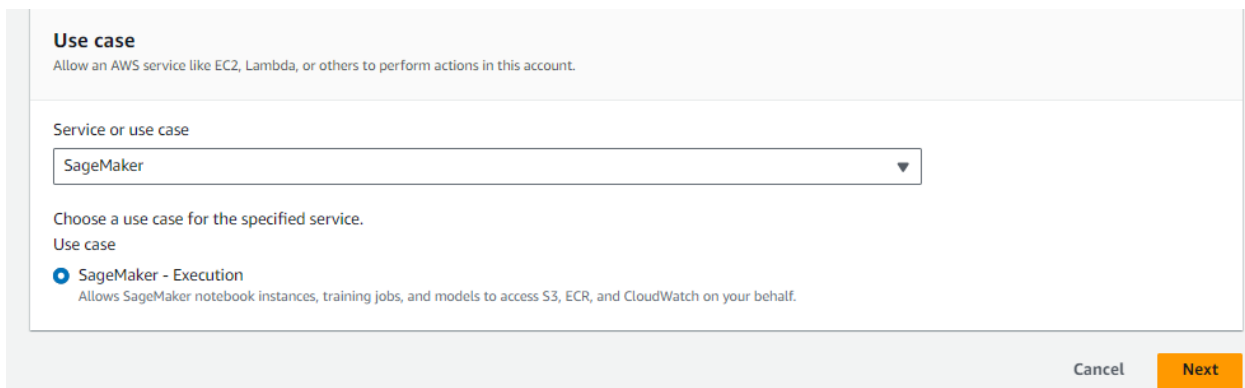


Creation of sagemaker role

- Goto the IAM Management console, click on the Roles menu on the left and then click on the Create role button.



- On the next screen, select SageMaker as the service and click on the Next: Permissions button.



- The role is created in no time. Open the sagemakerrole role details, remove AmazonSageMakerFullAccess policy and attach PowerUserAccess policy to the role.

Permissions policies (1) [Info](#)

You can attach up to 10 managed policies.

Filter by Type: All types

<input type="checkbox"/>	Policy name	Type	Attached entities
<input type="checkbox"/>	PowerUserAccess	AWS managed - job function	1

- The role is ready. In the next step, you register an IoT Device.
- You will first create an IoT policy which authorizes the device to perform actions within AWS IoT core. Goto the IoT Core Console, click on Policies menu under Secure in the left and then click on the Create policy button.

AWS IoT policies (0) [Info](#)

AWS IoT policies allow you to control access to the AWS IoT Core data plane operations. AWS IoT policies are separate and different from IAM policies. AWS IoT policies apply only to AWS IoT data plane operations.

Find policies

No policies

You don't have any AWS IoT policies in eu-west-1.

[Create](#)

- On the next screen, enter policy name, enter "iot:*" for the Action, enter "*" for the Resource ARN, select Allow for the Effect and click on the Create button.

Create policy [Info](#)

AWS IoT Core policies allow you to manage access to the AWS IoT Core data plane operations.

Policy properties

AWS IoT Core supports named policies so that many identities can reference the same policy document.

Policy name

sage_policy

A policy name is an alphanumeric string that can also contain period (.), comma (,), hyphen(-), underscore (_), plus sign (+), equal sign (=), and at sign (@) characters, but no spaces.

Tags - optional

The image shows two screenshots from the AWS IoT console. The top screenshot is the 'Policy document' builder, which includes a 'Builder' tab, a description of policy statements, and fields for 'Policy effect' (set to 'Allow'), 'Policy action' (set to '*'), and 'Policy resource' (set to '*'). There is a 'Remove' button and an 'Add new statement' button. The bottom screenshot shows the 'AWS IoT policies (1)' page in the console. It features a green success banner stating 'Successfully created policy sage_policy.' and a 'View policy' button. Below this, there's a search bar and a table listing the policy 'sage_policy'.

- The policy is ready. After creating the policy, you will now create a device as thing and attach the policy to it.
- On the AWS IoT Core console, click on Things menu under Manage in the left and then click on the Register a thing button.
- On the Creating AWS IoT things screen, click on the Create a single thing button.

The image shows the 'Manage' page in the AWS IoT console. The left sidebar has a 'Manage' section with options like 'All devices', 'Greengrass devices', 'LPWAN devices', 'Software packages', 'Remote actions', 'Message routing', 'Retained messages', and 'Security'. The main content area has a large heading 'Manage' and a subheading 'Manage your devices, IoT data, remote actions, security, and applications'. Below this is a 'Create a thing resource' box with a 'Create thing' button. At the bottom, there are sections for 'How it works' and 'Pricing'.

- On the Add your device to the thing registry screen, enter the thing name and click on the Next button.

The screenshot shows the AWS IoT console interface. The breadcrumb trail is: AWS IoT > Manage > Things > Create things > Create single thing. The left sidebar shows the steps: Step 1: Specify thing properties (active), Step 2 - optional: Configure device certificate, and Step 3 - optional: Attach policies to certificate. The main content area is titled 'Specify thing properties' with an 'Info' link. Below the title is a description: 'A thing resource is a digital representation of a physical device or logical entity in AWS IoT. Your device or entity needs a thing resource in the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features.' The 'Thing properties' section has a 'Thing name' input field containing 'sage_thing'. Below the input field is a note: 'Enter a unique name containing only: letters, numbers, hyphens, colons, or underscores. A thing name can't contain any spaces.'

- Attach your policy

The screenshot shows the AWS IoT console interface at the 'Attach policies to certificate' step. The breadcrumb trail is: AWS IoT > Manage > Things > Create things > Create single thing. The left sidebar shows the steps: Step 1: Specify thing properties, Step 2 - optional: Configure device certificate, and Step 3 - optional: Attach policies to certificate (active). The main content area is titled 'Attach policies to certificate - optional' with an 'Info' link. Below the title is a description: 'AWS IoT policies grant or deny access to AWS IoT resources. Attaching policies to the device certificate applies this access to the device.' The 'Policies (1/1)' section shows a search bar with 'Filter policies', a list of policies with checkboxes, and a 'Create policy' button. The policy 'sage_policy' is selected. At the bottom, there are 'Cancel', 'Previous', and 'Create thing' buttons.

- Download your certificates

Download certificates and keys ✕

Download certificate and key files to install on your device so that it can connect to AWS.

Device certificate

You can activate the certificate now, or later. The certificate must be active for a device to connect to AWS IoT.

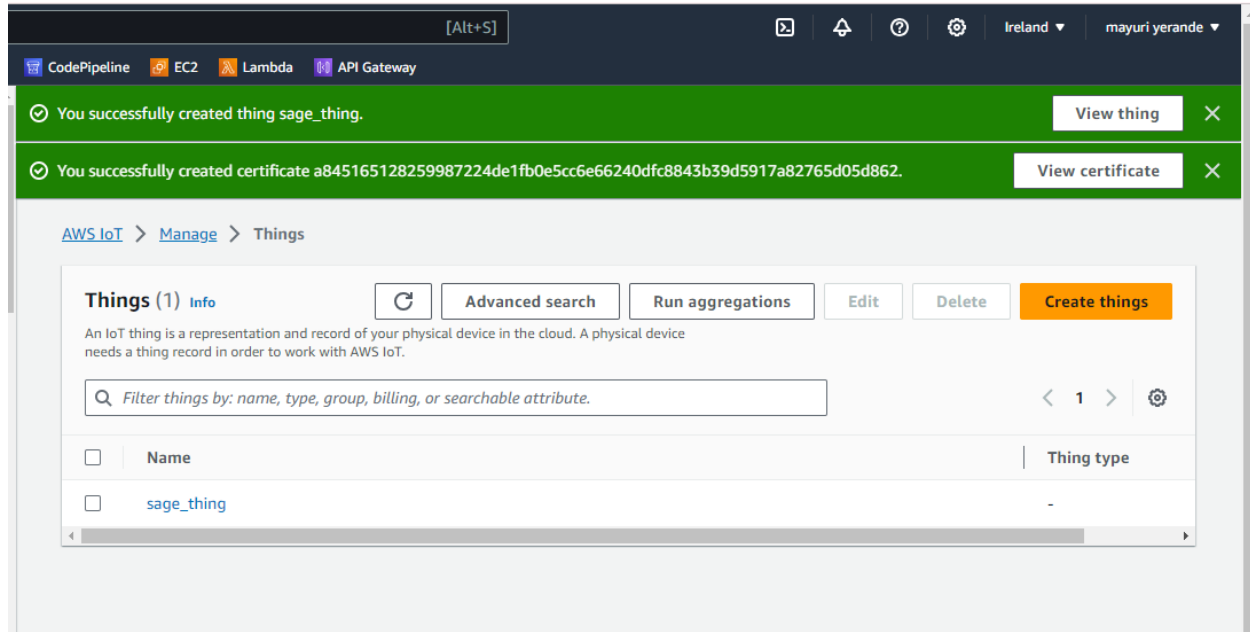
Device certificate

a8451651282...te.pem.crt

[Deactivate certificate](#)

[Download](#)

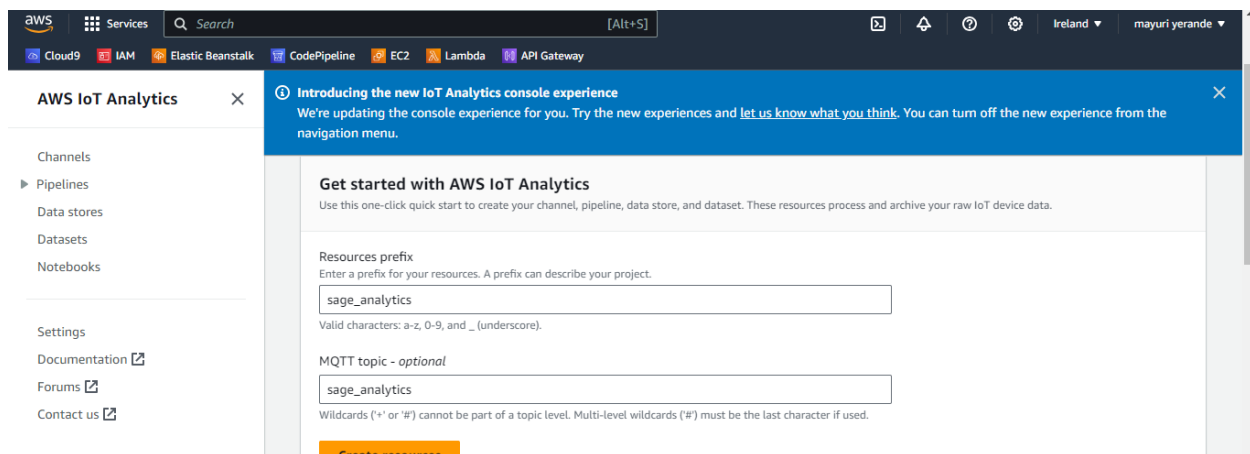
Key files



- Registration is complete now. The next step is to configure AWS IoT Analytics.

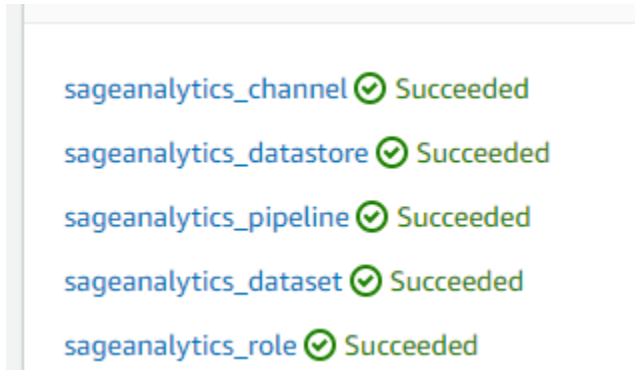
Configuring AWS Analytics

- Goto AWS IoT Analytics management console. Give resource and topic name and click on create resources



- It will start creation of the resources, primarily - channel, pipeline, data stores, data set along with an IAM Role and IoT Rule. Wait till all the resources are created.
- The sage_Analytcs_rule IoT Rule is responsible to send all the messages published to sage_Analytcs_topic topic to sage_Analytcs__channel channel. You can go to IoT Core

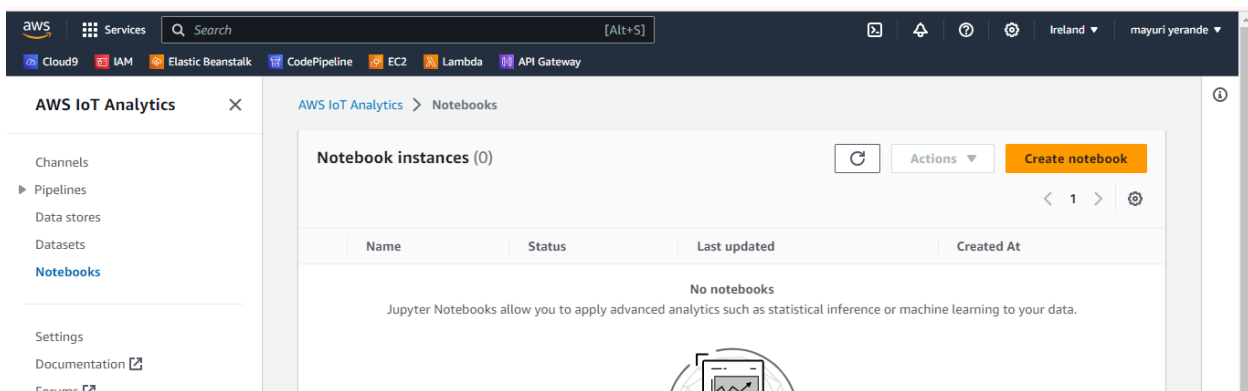
Management console and check `dojoanalytics_topicrule` configuration details under the Act menu.



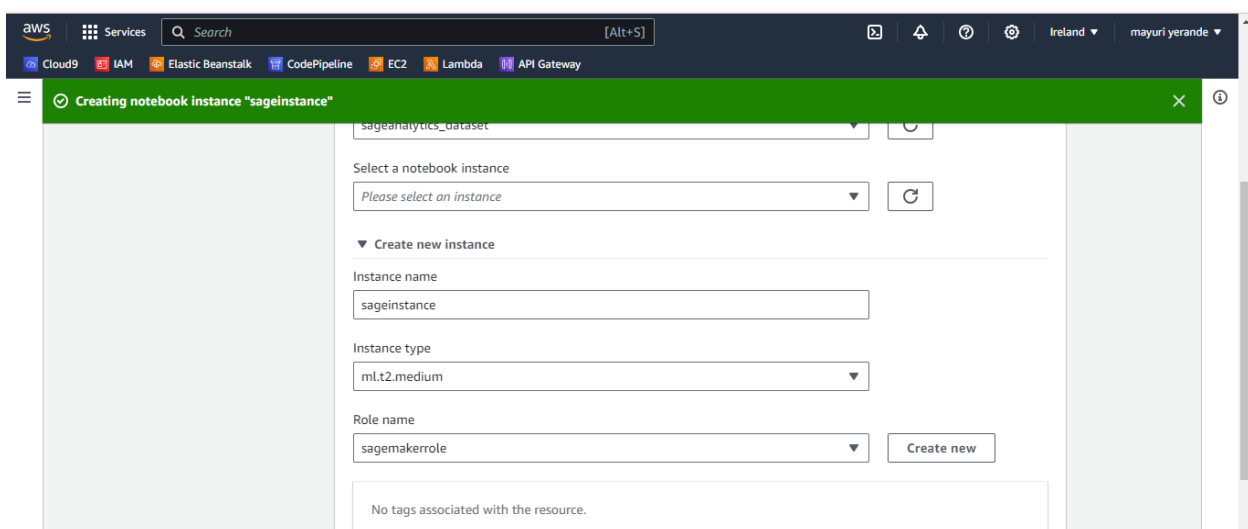
- The IoT Analytics resources are ready

Creation of Notebook

- Goto Amazon SageMaker console. Go to notebooks and create notebook



- Create a new notebook instance



Select a template

Step 2
Setup notebook

Step 3
Review and create

Setup notebook

Notebook name
sage_notebook

Select dataset source
sageanalytics_dataset

Select a notebook instance
sageinstance1

► Create new instance

Cancel Previous **Next**

- Notebook configuration is done

aws Services Search [Alt+S] Ireland mayuri.yerande

Cloud9 IAM Elastic Beanstalk CodePipeline EC2 Lambda API Gateway

AWS IoT Analytics

Channels

Pipelines

Data stores

Datasets

Notebooks

Settings

Documentation

AWS IoT Analytics > Notebooks

Notebook instances (2)

Name	Status	Last updated	Created At
sageinstance1	In service	Oct 7, 2023 10:56:28 PM +0530	Oct 7, 2023 10:52:17 PM +0530
sageinstance	In service	Oct 7, 2023 10:54:47 PM +0530	Oct 7, 2023 10:50:52 PM +0530

Publishing the data and setting up the dataset

- On the AWS IoT Core console, click on Test menu in the left to open MQTT client. Click on the Publish to a topic link.
- Copy paste the given codes and publish it one by one

Subscribe to a topic **Publish to a topic**

Topic name
The topic name identifies the message. The message payload will be published to this topic with a Quality of Service (QoS) of 0.

Q sage_topic

Message payload

```
{
  "temperature": 40,
  "vibration": 30,
  "pressure": 25
}
```

► Additional configuration

Publish

<div><div>▼ topic</div><div><pre>{ "temperature": 39, "vibration": 40, "pressure": 44 }</pre></div><div>► Properties</div></div>	October 07, 2023, 23:18:59 (UTC+0530)
<div><div>▼ topic</div><div><pre>{ "temperature": 21, "vibration": 21, "pressure": 19 }</pre></div><div>► Properties</div></div>	October 07, 2023, 23:18:46 (UTC+0530)
<div><div></div><div><pre>{ "temperature": 28, "vibration": 23, "pressure": 25 }</pre></div><div>► Properties</div></div>	
<div><div>▼ topic</div><div><pre>{ "temperature": 22, "vibration": 22, "pressure": 29 }</pre></div><div>► Properties</div></div>	October 07, 2023, 23:18:16 (UTC+0530)

- Go to Iot analytics
- Select your dataset
- Click on “run now”

The screenshot shows the AWS IoT Analytics console. A green notification banner at the top states: "You've successfully started the query for your dataset. Dataset content version: 73b2fb3f-e30a-4e57-82ff-1ae98642593d." The left sidebar contains navigation links: Channels, Pipelines, Data stores, **Datasets**, Notebooks, Settings, Documentation, Forums, and Contact us. The main content area displays the "sageanalytics_dataset" page. It includes a "Run now" button and a "Delete" button. Below this is an "Overview" section with the following details:

Property	Value
Dataset ARN	am:aws:iotanalytics:eu-west-1:378963872694:dataset/sageanalytics_dataset
Created	Oct 7, 2023 10:44:04 PM +0530
Last updated	Oct 7, 2023 10:44:04 PM +0530
Type	Query
Status	Active

- All good. The data is there in the Analytics Data Set. Let's use Jupyter Notebook to analyze the data.

The screenshot shows the Amazon SageMaker console and the Jupyter Notebook interface. The top part of the console displays the "sageinstance1" page with buttons for "Delete", "Stop", "Open Jupyter", and "Open JupyterLab". Below this is the "Notebook instance settings" section with the following details:

Property	Value
Name	sageinstance1
Notebook instance type	ml.t2.medium
ARN	arn:aws:sagemaker:eu-west-1:378963872694:notebook-instance/sageinstance1
Elastic Inference	-

The bottom part of the screenshot shows the Jupyter Notebook interface. The top bar includes the "jupyter" logo and buttons for "Open JupyterLab", "Quit", and "Logout". The left sidebar contains tabs for "Files", "Running", "Clusters", "Conda", and "SageMaker Examples". The main content area shows a file explorer with a folder named "IoTAnalytics" and a file named "IoTAnalytics". The file explorer also includes a table with columns for "Name", "Last Modified", and "File size".

- Now connect to the conda kernel here
- Run the first two cells
- Now write the following code and run the cells

- Now we will read the dataset

```
In [13]: import pandas as pd
import matplotlib as plt

df= pd.read_csv(dataset_url,header=0)
df
```

Out[13]:

	temperature	vibration	pressure	_dt
0	28	23	25	2023-10-07 00:00:00:000
1	39	40	44	2023-10-07 00:00:00:000
2	19	21	22	2023-10-07 00:00:00:000
3	19	18	22	2023-10-07 00:00:00:000
4	24	30	30	2023-10-07 00:00:00:000
5	35	21	19	2023-10-07 00:00:00:000
6	40	30	25	2023-10-07 00:00:00:000
7	32	22	29	2023-10-07 00:00:00:000
8	29	33	33	2023-10-07 00:00:00:000
9	34	20	20	2023-10-07 00:00:00:000
10	35	20	20	2023-10-07 00:00:00:000

- Now that we have got our dataset, we have successfully performed connection here
- **Now that our connection is successfully, we will perform analytics**

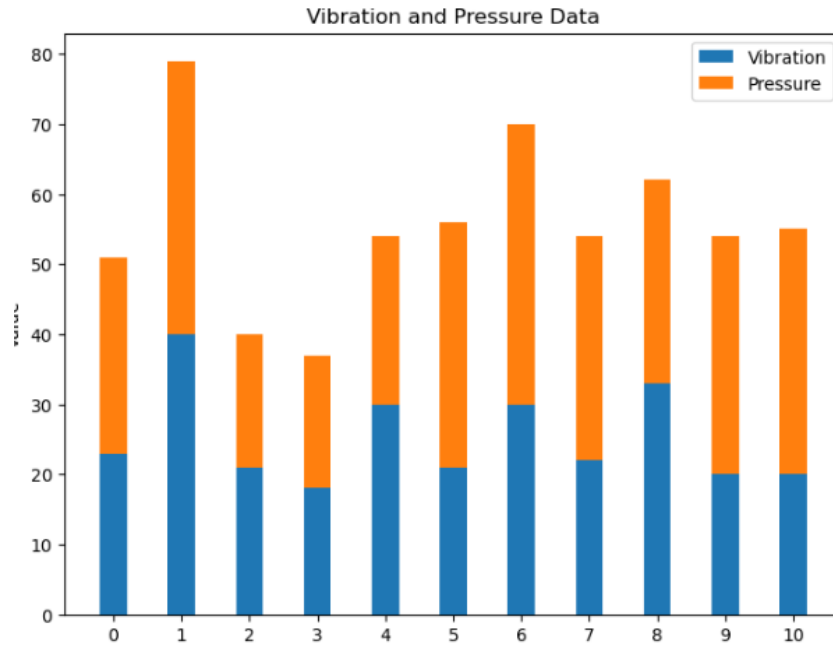
Analysis of data

- Bar graph for vibration vs pressure

```
In [19]: import matplotlib.pyplot as plt
import pandas as pd

vibration = df['vibration']
pressure = df['temperature']

# Create a bar graph
plt.figure(figsize=(8, 6))
plt.bar(df.index, vibration, width=0.4, label='Vibration')
plt.bar(df.index, pressure, width=0.4, label='Pressure', bottom=vibration)
plt.xlabel('Sample')
plt.ylabel('Value')
plt.title('Vibration and Pressure Data')
plt.xticks(df.index)
plt.legend()
plt.show()
```



- Pie chart for temperature, vibration and pressure

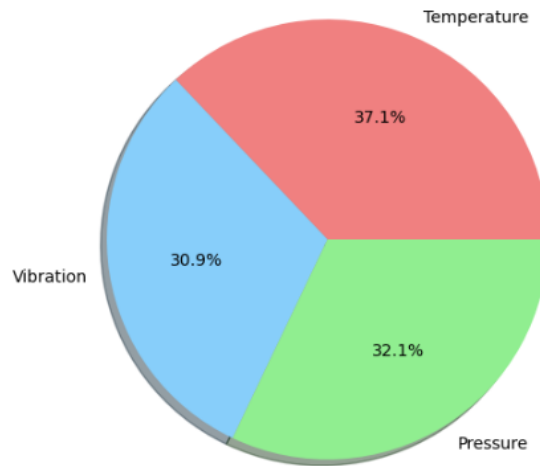
```
: import matplotlib.pyplot as plt
import pandas as pd

sum_temperature = df["temperature"].sum()
sum_vibration = df["vibration"].sum()
sum_pressure = df["pressure"].sum()

labels = ["Temperature", "Vibration", "Pressure"]
values = [sum_temperature, sum_vibration, sum_pressure]
colors = ["lightcoral", "lightskyblue", "lightgreen"]

plt.figure(figsize=(6, 6))
plt.pie(values, labels=labels, colors=colors, autopct='%1.1f%%', shadow=True)
plt.title("Pie Chart of Temperature, Vibration, and Pressure")
plt.show()
```

Pie Chart of Temperature, Vibration, and Pressure



- Box plot for each entity

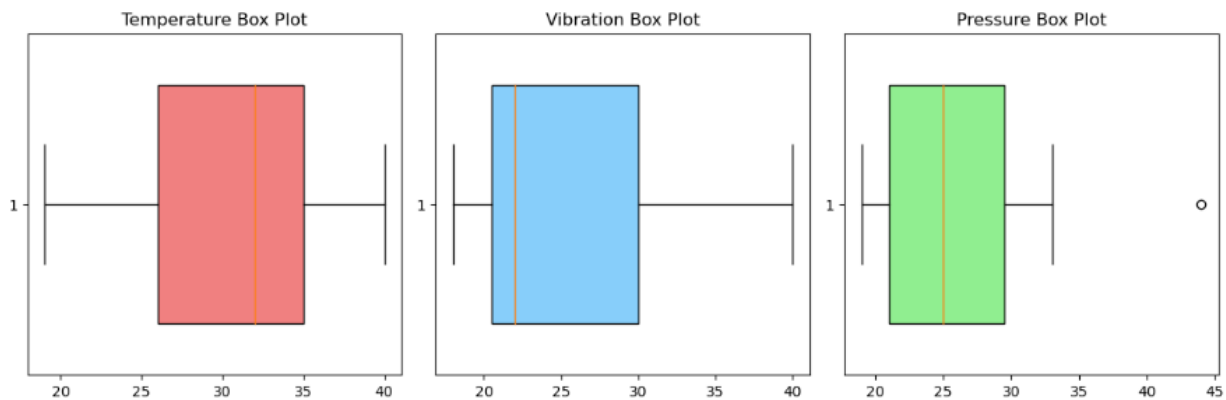
```
In [38]: plt.figure(figsize=(12, 4))

plt.subplot(131)
plt.boxplot(df["temperature"], vert=False, widths=0.7, patch_artist=True, boxprops=dict(facecolor="lightcoral"))
plt.title("Temperature Box Plot")

plt.subplot(132)
plt.boxplot(df["vibration"], vert=False, widths=0.7, patch_artist=True, boxprops=dict(facecolor="lightskyblue"))
plt.title("Vibration Box Plot")

plt.subplot(133)
plt.boxplot(df["pressure"], vert=False, widths=0.7, patch_artist=True, boxprops=dict(facecolor="lightgreen"))
plt.title("Pressure Box Plot")

plt.tight_layout()
plt.show()
```



CONCLUSION:

Thus we successfully performed AWS IOT Analytics. We published the data on IOT Core, set up our data according to it and then performed analytics on our created jupyter notebook.