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.

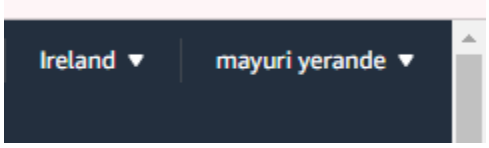
IoT Analytics: AWS IoT Analytics is a service that allows you to process and analyze IoT data at scale. It supports SQL-based querying, data transformation, and integration with other AWS services like Amazon S3, AWS Lambda, and Amazon QuickSight.

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.

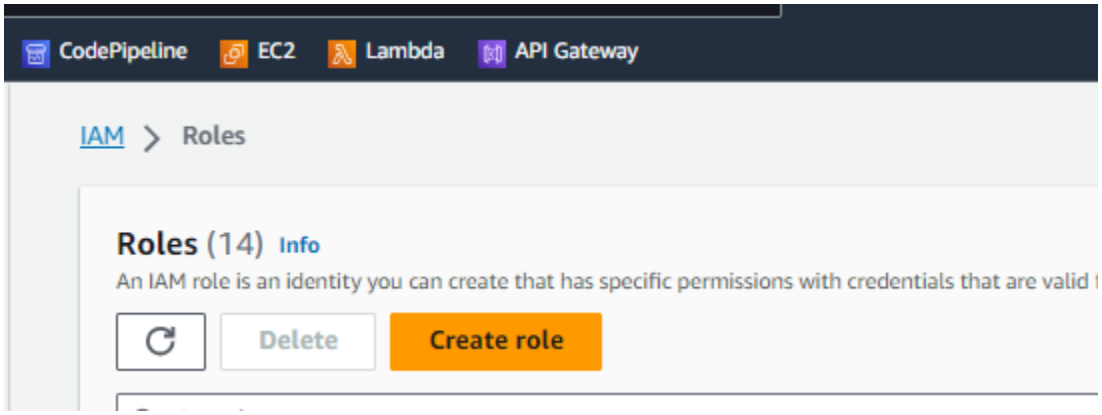
IMPLEMENTATION:

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

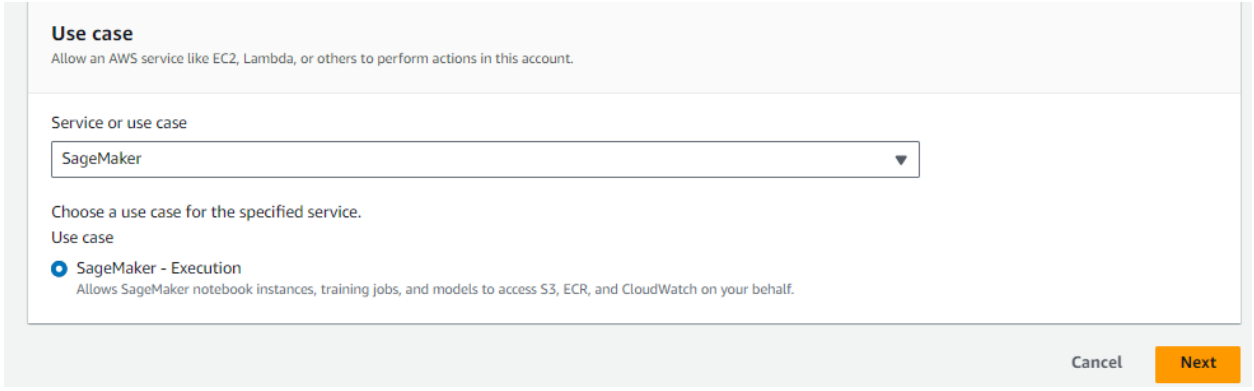


Creation of sagemaker role

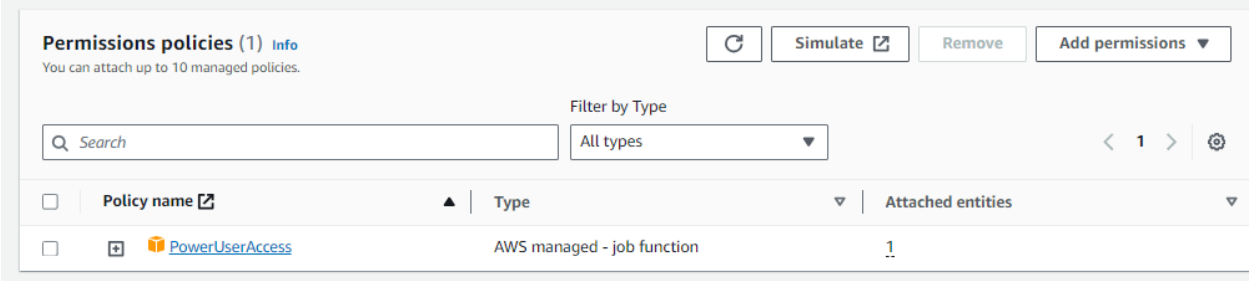
- Goto the IAM Management console, click on the Roles menu on the left and then click on the Create role button.



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

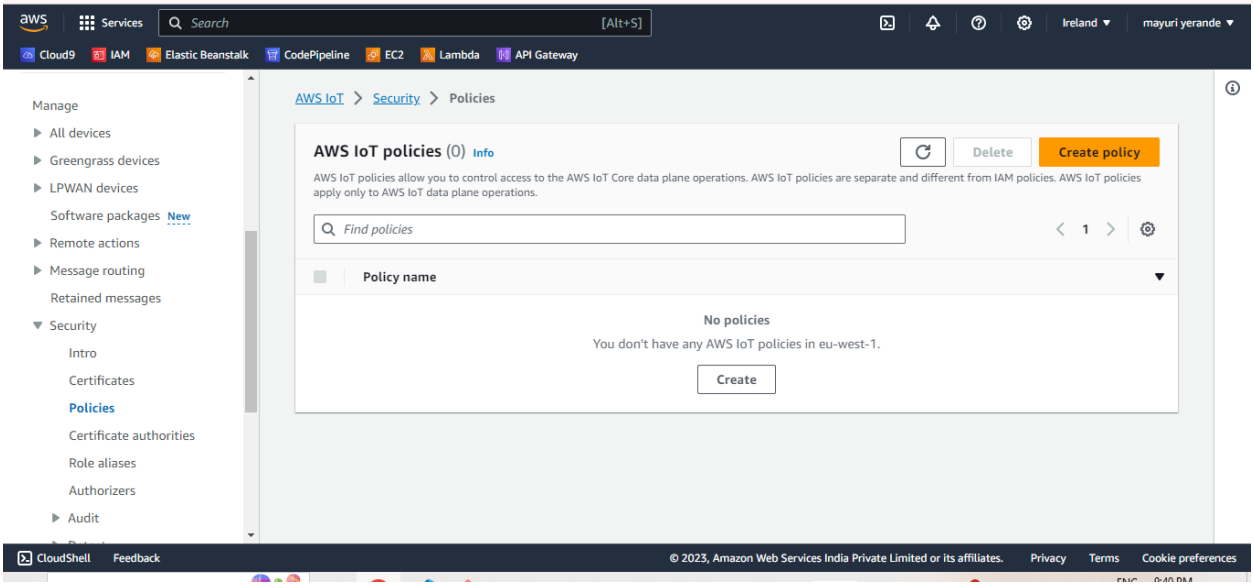


- The role is created in no time. Open the sagemakerrole role details, remove AmazonSageMakerFullAccess policy and attach PowerUserAccess policy to the role.

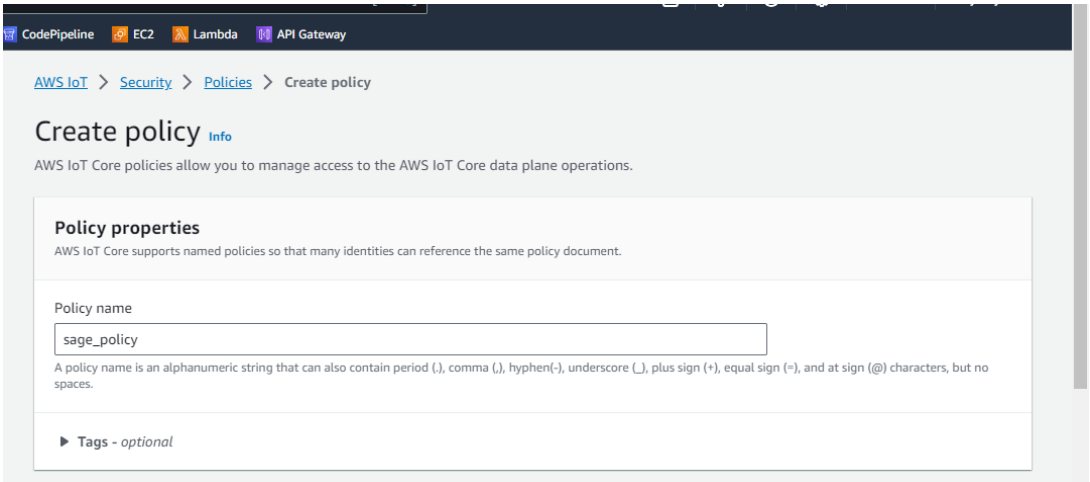


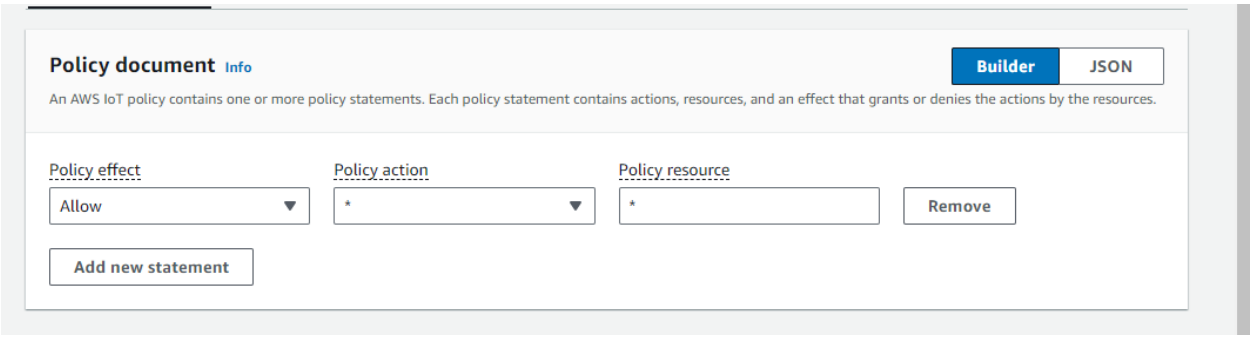
- The role is ready. In the next step, you register an IoT Device.

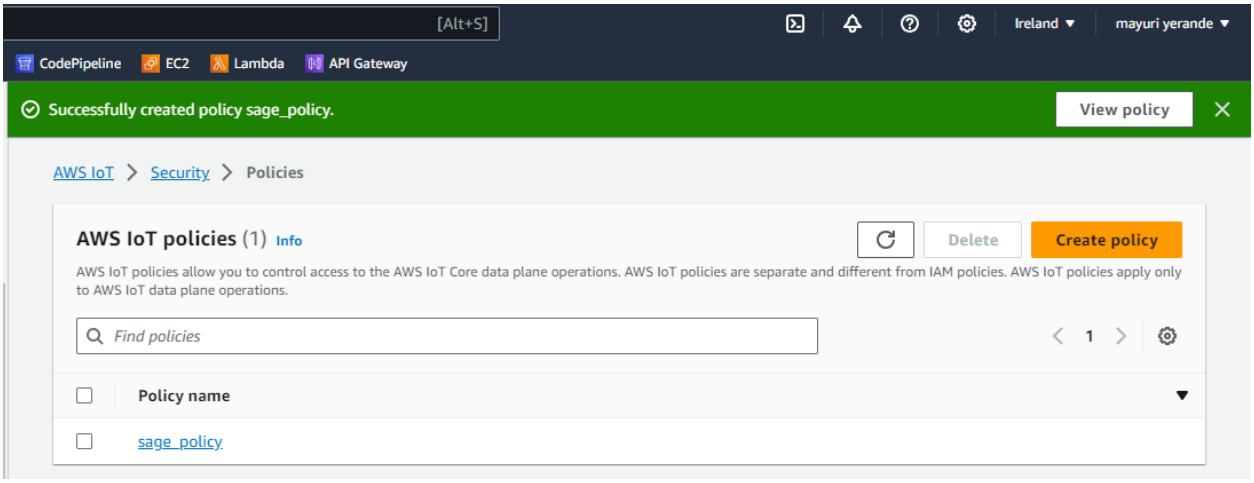
- You will first create an IoT policy which authorizes the device to perform actions within AWS IoT core. Goto the IoT Core Console, click on Policies menu under Secure in the left and then click on the Create a policy button.



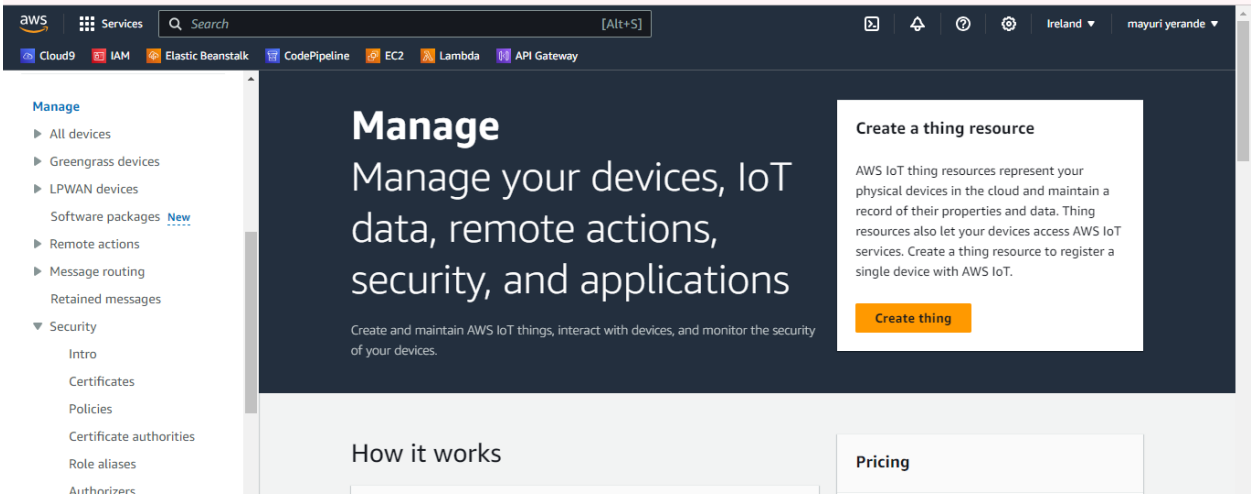
- On the next screen, enter policy name, enter “iot:*" for the Action, enter "*" for the Resource ARN, select Allow for the Effect and click on the Create button.



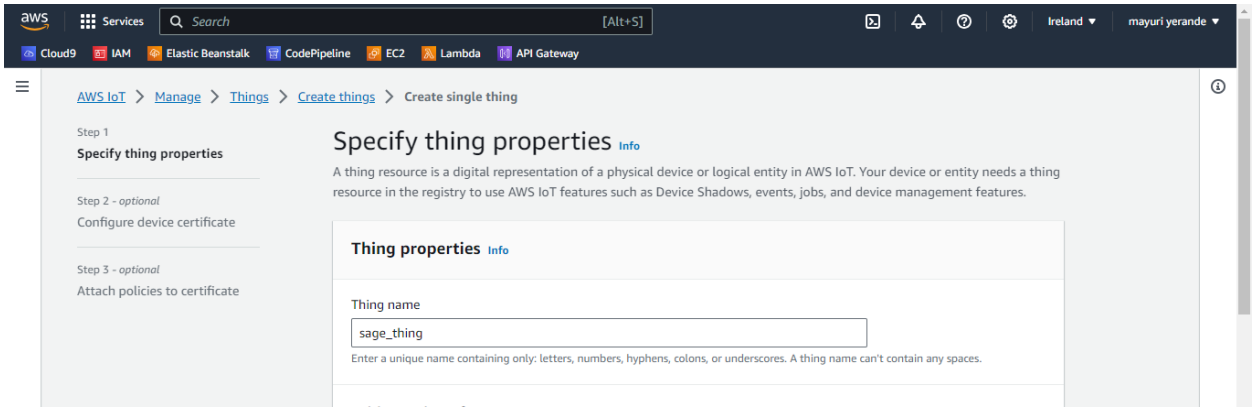




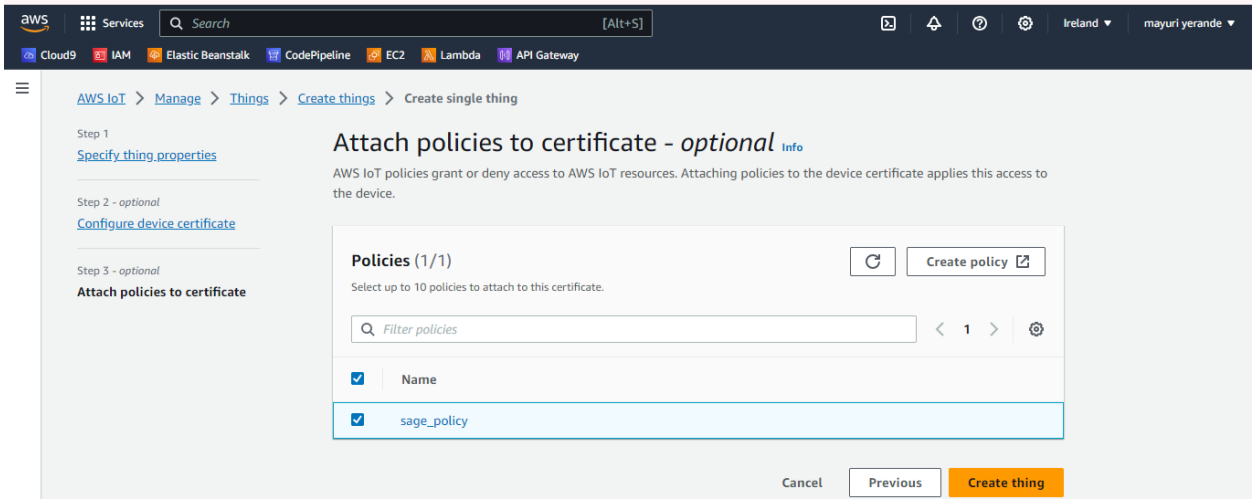
- The policy is ready. After creating the policy, you will now create a device as thing and attach the policy to it.
- On the AWS IoT Core console, click on Things menu under Manage in the left and the click on the Register a thing button.
- On the Creating AWS IoT things screen, click on the Create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



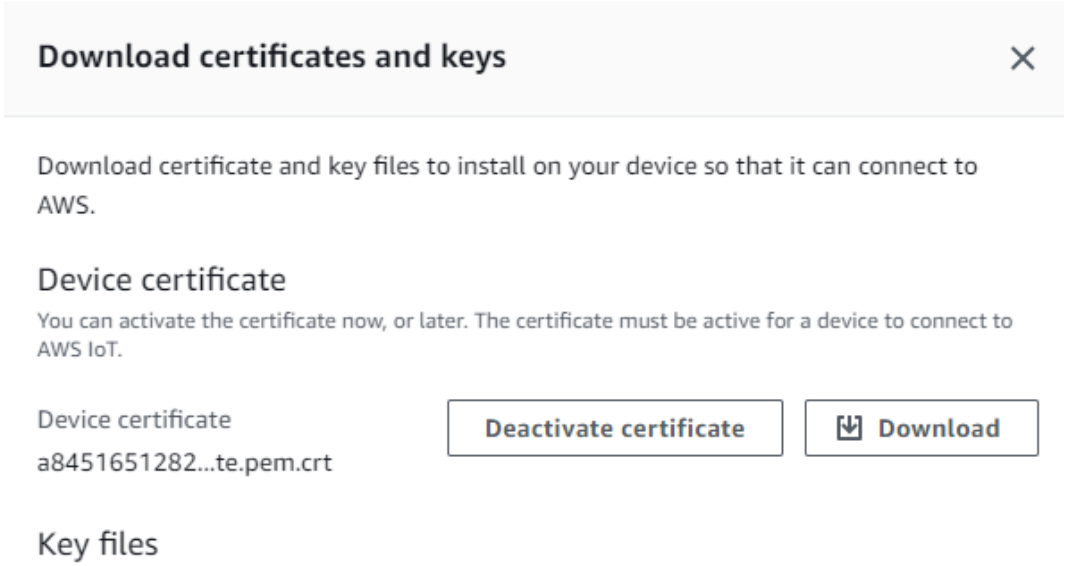
- On the Add your device to the thing registry screen, enter the thing name and click on the Next button.

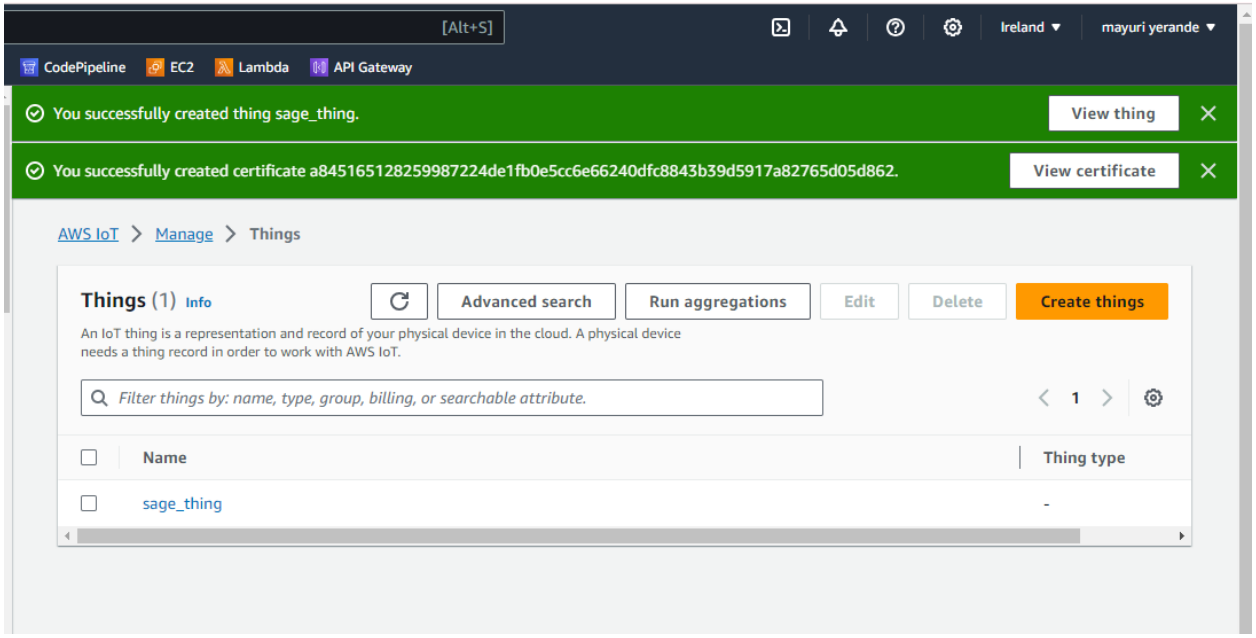


- Attach your policy



- Download your certificates

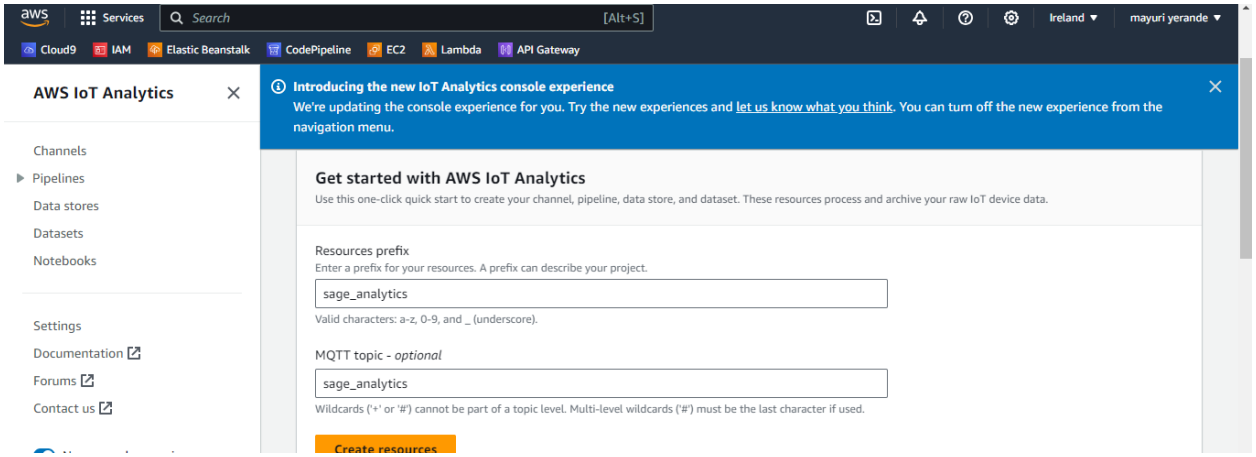




- Registration is complete now. The next step is to configure AWS IoT Analytics.

Configuring AWS Analytics

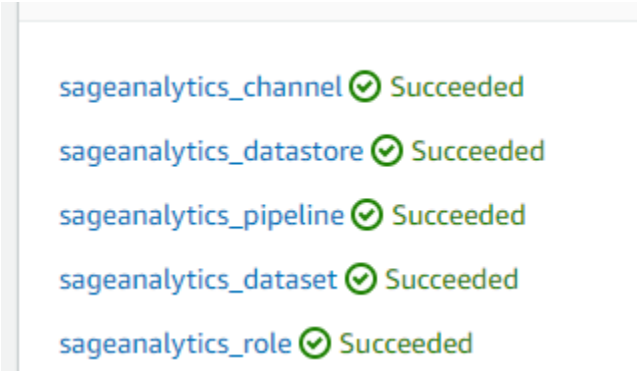
- Goto AWS IoT Analytics management console. Give resource and topic name and click on create resources



- 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

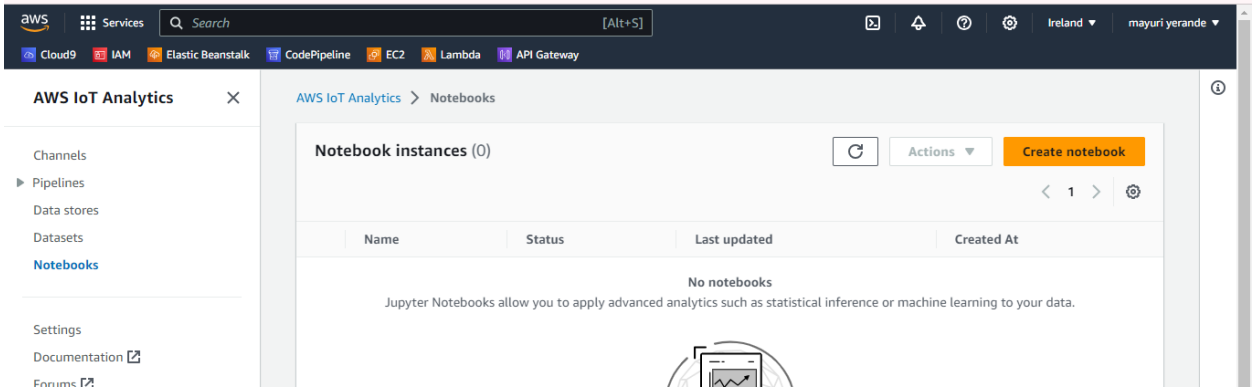
Management console and check dojoanalytics_topicrule configuration details under the Act menu.



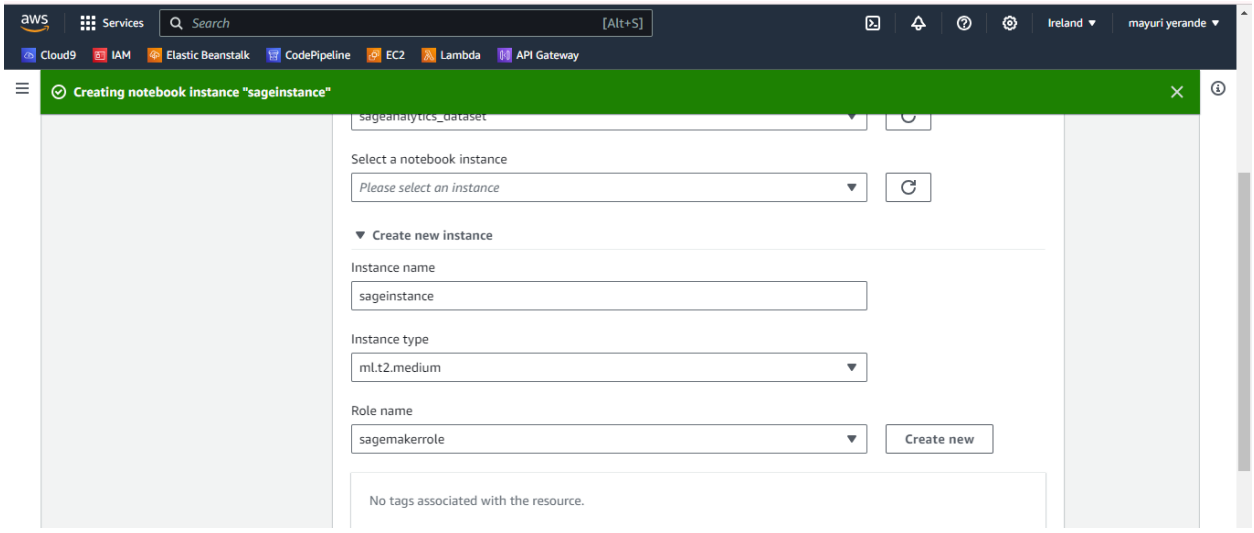
- The IoT Analytics resources are ready

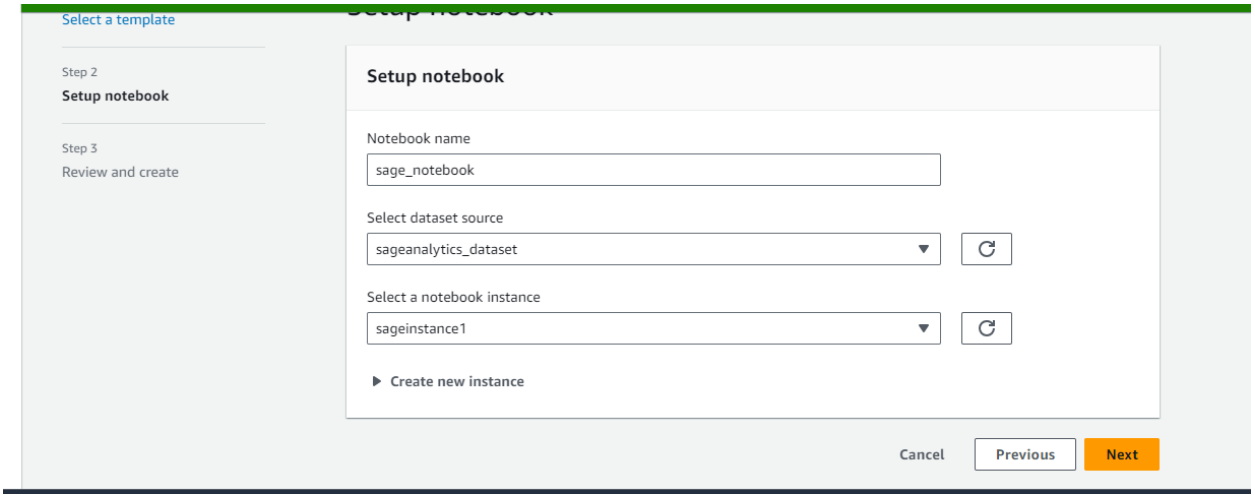
Creation of Notebook

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

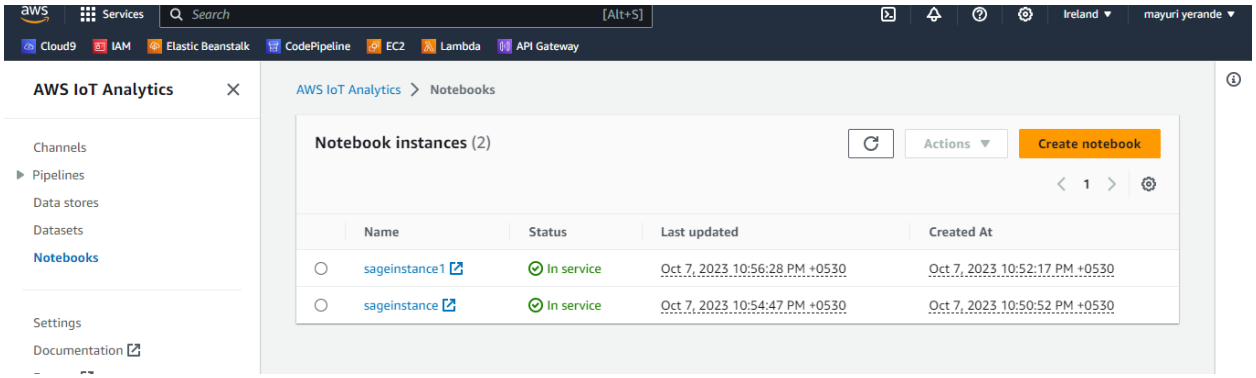


- Create a new notebook instance



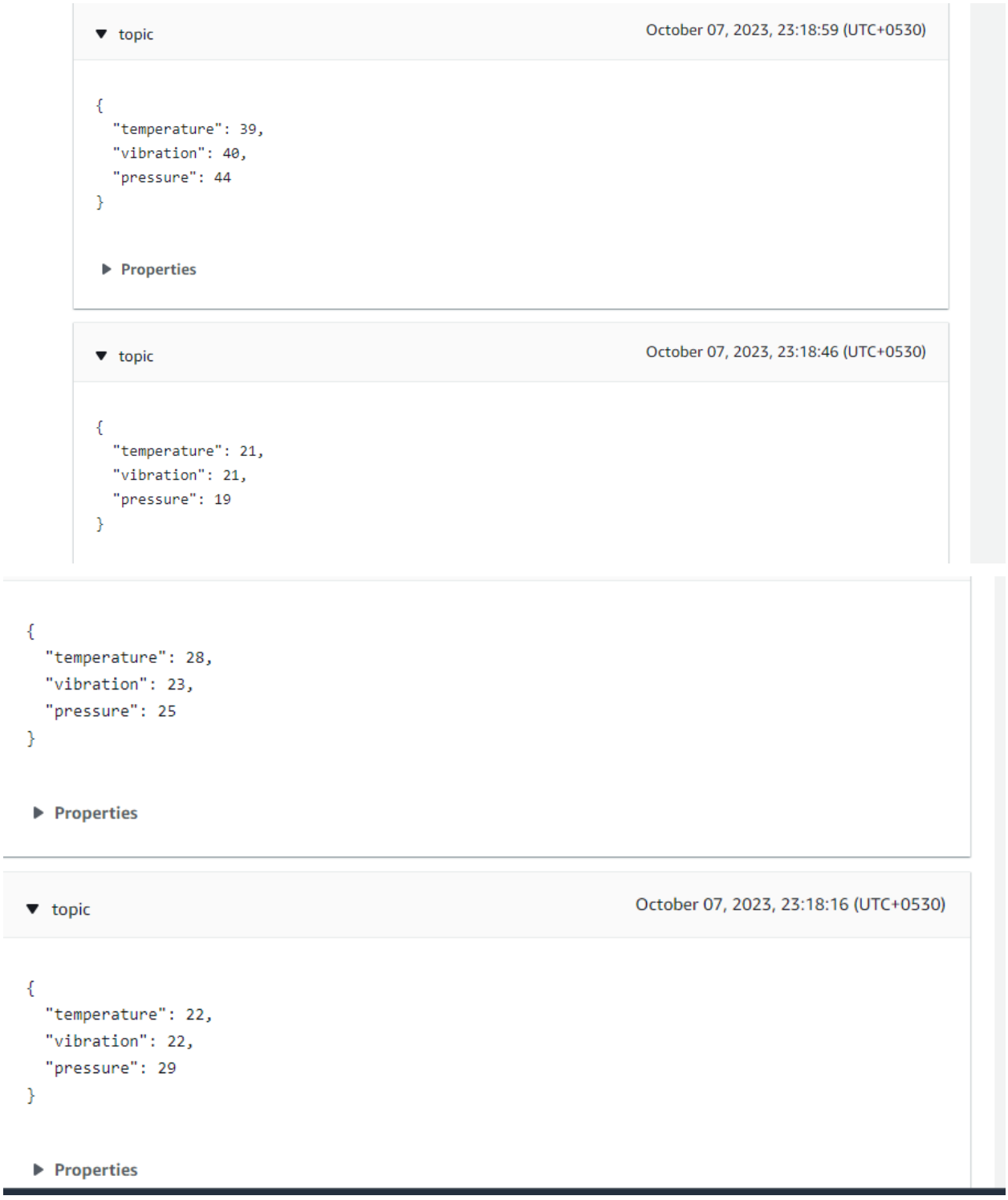


- Notebook configuration is done

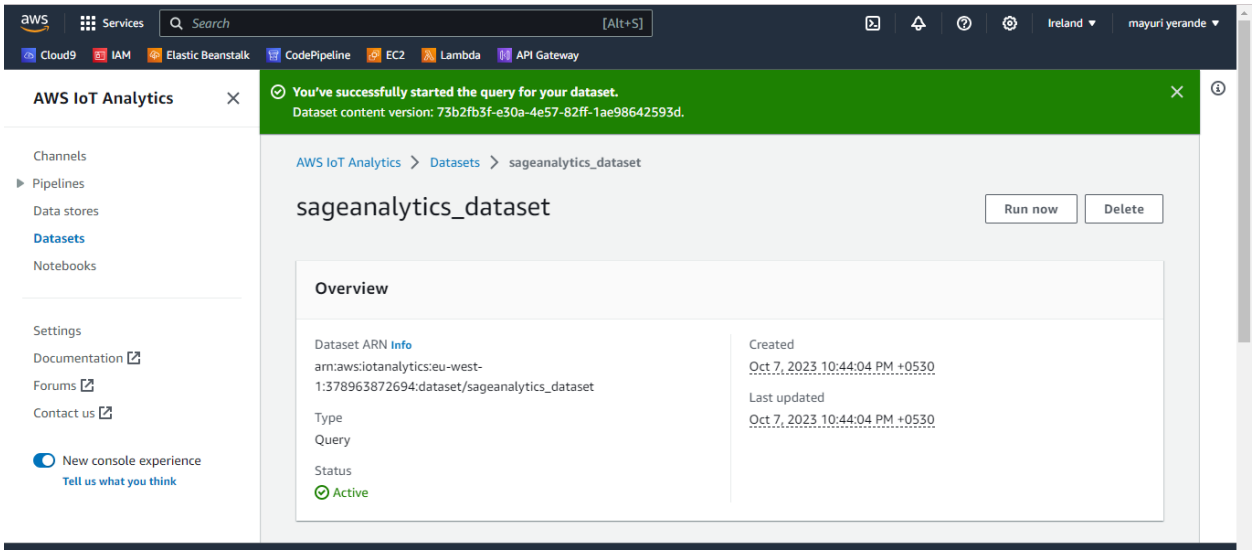


Publishing the data and setting up the dataset

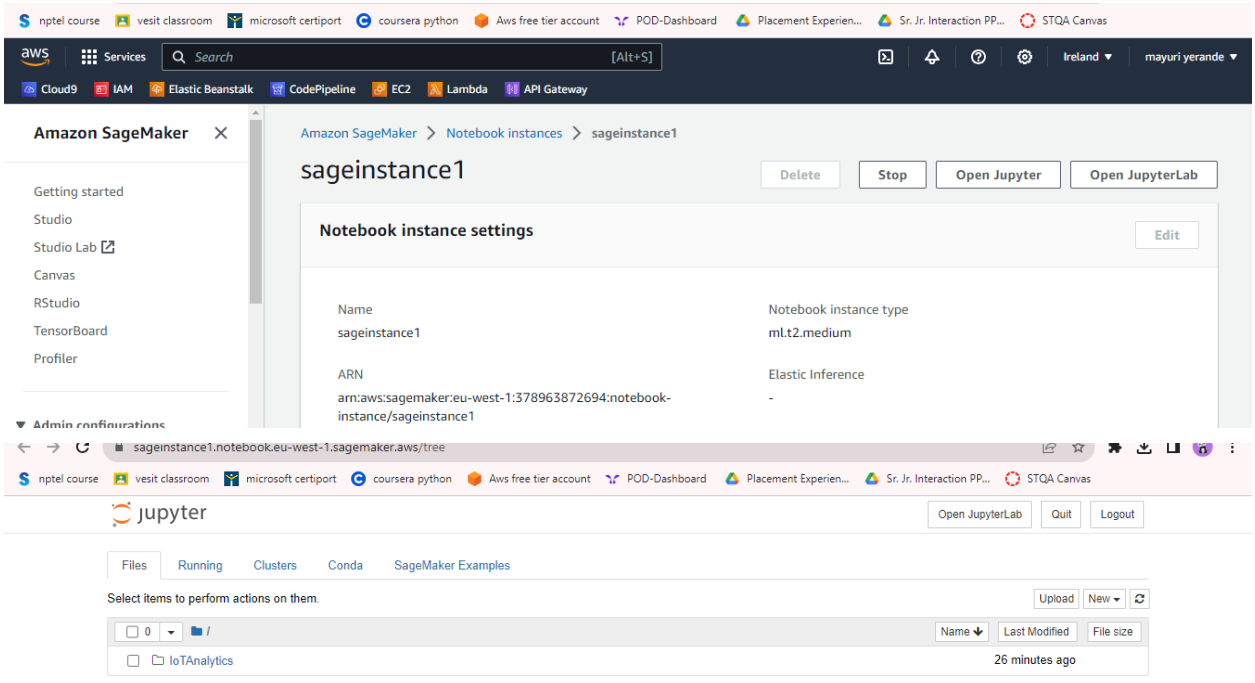
- On the AWS IoT Core console, click on Test menu in the left to open MQTT client. Click on the Publish to a topic link.
- Copy paste the given codes and publish it one by one

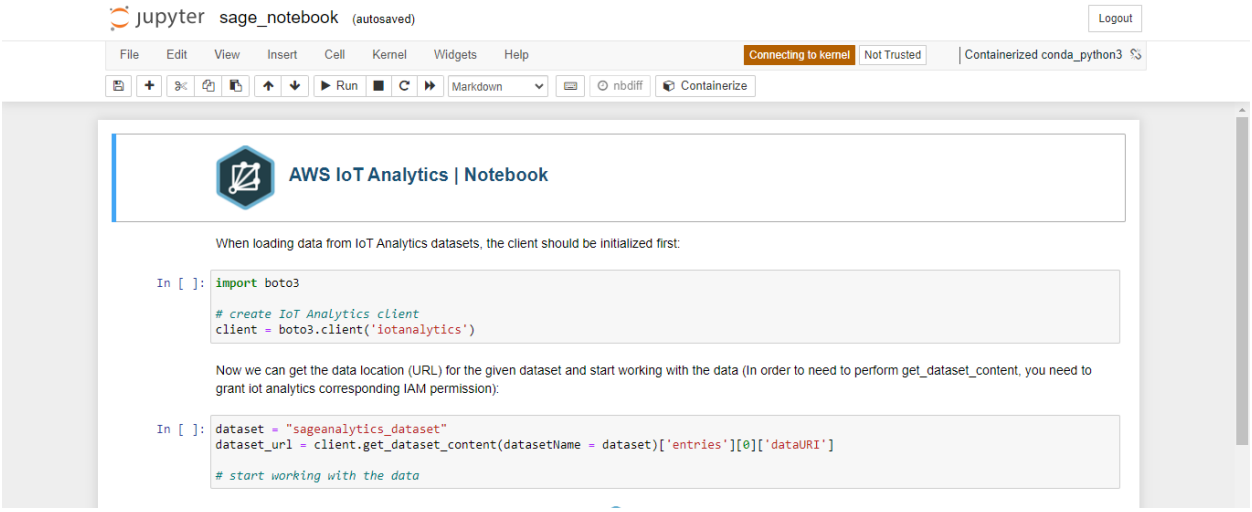


- Go to Iot analytics
- Select your dataset
- Click on “run now”



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





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

```
In [3]: dataset_url
Out[3]: 'https://aws-iot-analytics-dataset-1685b8a4-338a-4431-bdde-c61337048ebf.s3.eu-west-1.amazonaws.com/results/1ec6ac3f-18ed-47cc-
ff6-4d0f41c137eb.csv?X-Amz-Security-Token=IQoJb3JpZ2luX2VjEIr%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaCWV1LXdlc3QtMSJHMEUCIAIWAXh2HZrr
GkJVhEPqcjeJ3m%2Bd7q75%2FHPx0h4DtivAiEAt5NwEDgOZhYx9tGj8SOiTkEMny%2Bh2wctJ3KfCCppiYoq3wIIk%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FARA
Ggw1NDUzMDIzNDM4MDUiDEDivQfmmgNq2W6AOyqzAqWF63zjXTQ5QJzxomfUZG6Mrmfjmg4Z%2BHPv9mr9NAr8MNfP5%2Bd5H90Kdc02iIiga6IHNBkK7b4MNUudzQ
XA4%2BhJU8Uv4n6mhOFDwYQR34OvYXRHgsJbVunyruzqzco5WN6ttHjJZhAKgljUwACIQZ%2FuLmMMUtTOU%2FLRdFFGJoP4YbBCactGY65Y9Nh8cqym4BVtKi%2Bg
8WRgBtMTOz%2BzHS%2F%2BBFF0IUtVEJ1KYtLO9J6f%2BtEpim0QdzAo7Z3hvfKVQVvWEa3L%2BnkLDg95aK5cM3BBwIYrT9mnOQ826CAmPxIQ1sRK99mSwthg4Ezk
fcrEJ8xWJEIufCxt8OpcnuerC%2FyX3SB9RG%2FzjnRBwt5q4SXaxo3VwVipZXwdo%2BOaX%2Fvf9y0RKd7GjdYrcXn8twVUIZAEwzbiGqQY6vwFf2KSFAhgrI1qdk
SjrMjPUNMDvnkIRlkxLzqW%2B%2FyHw8YxSYitRFuiIeWS2zCKBgDOSykAbit9W1nIc0yWHRyKDCSv7VL1cFwLYRwCRTuoerLQhJc5OXhukaKGUKyM24JZPN4TsK94
vLyxpr1HpMmVxs%2B3yMhugr%2BjW96FoVUb3%2BlJAolossAXtBPG3fZt6D3XP9tgmE%2BToFLY%2FxOrAfQnvL%2FILFeEZl8AggziPYXqRqpcVjB26eVYFYL9hv
Dg%3D%3D&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20231007T175837Z&X-Amz-SignedHeaders=host&X-Amz-Expires=7200&X-Amz-Creden
ial=ASIAX55UNOB65G2X6XHJ%2F20231007%2Feu-west-1%2Fs3%2Faws4_request&X-Amz-Signature=8730ea93855b07673993edc0b57ea04d9bd6ef94ad
d7f9baf90bee5f1fcf8ab'
```

- 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**

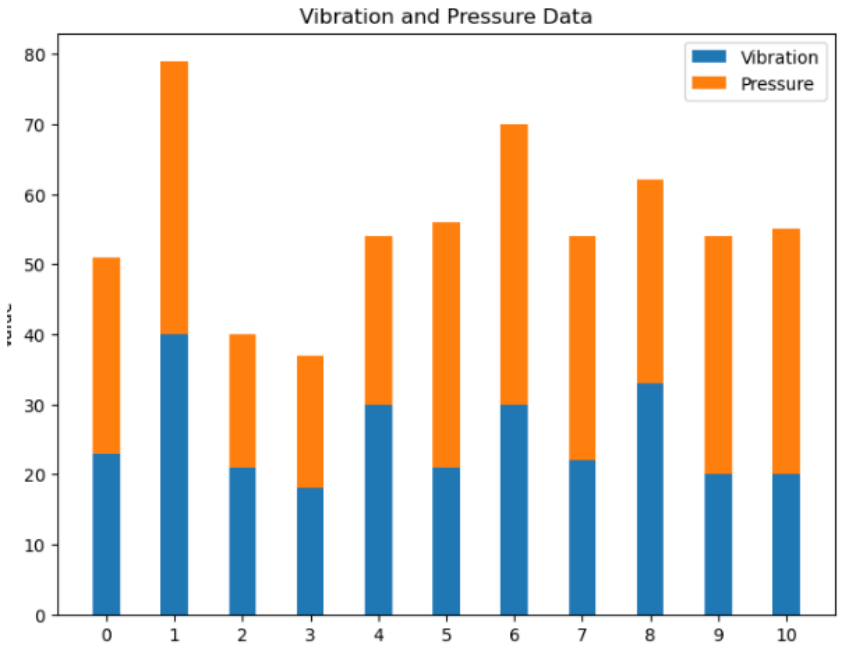
Analysis of data

- 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()
```



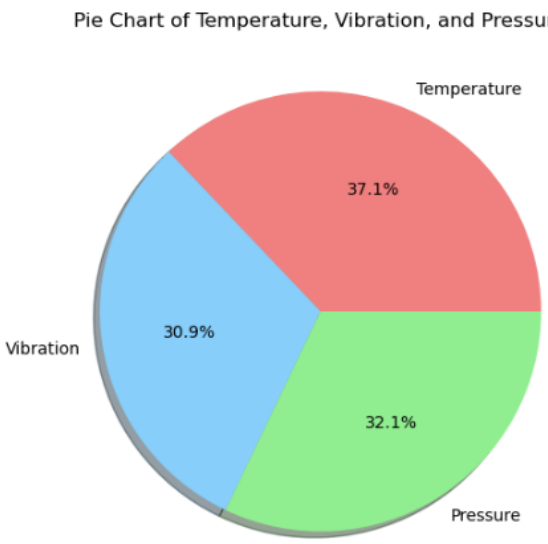
- Pie chart for temperature, vibration and pressure

```
: 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()
```



- Box plot for each entity

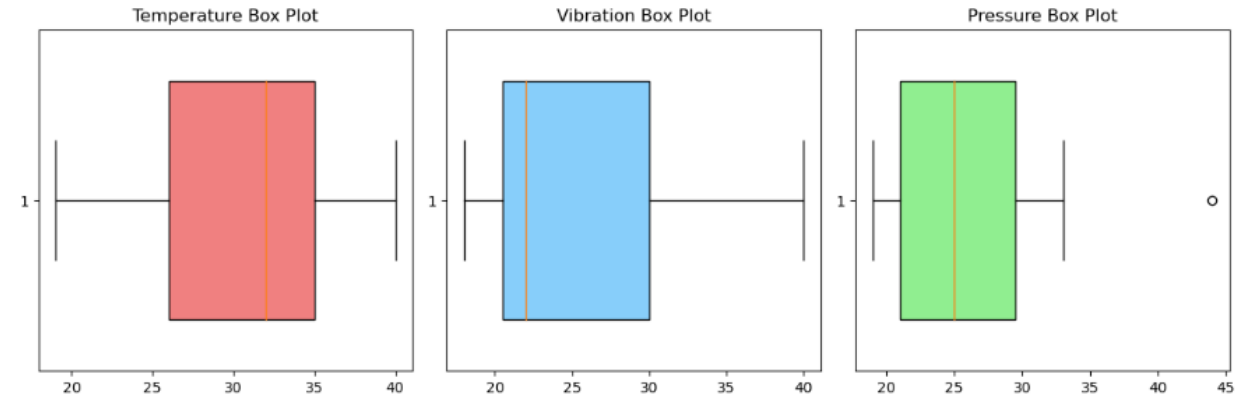
```
In [38]:
plt.figure(figsize=(12, 4))

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.