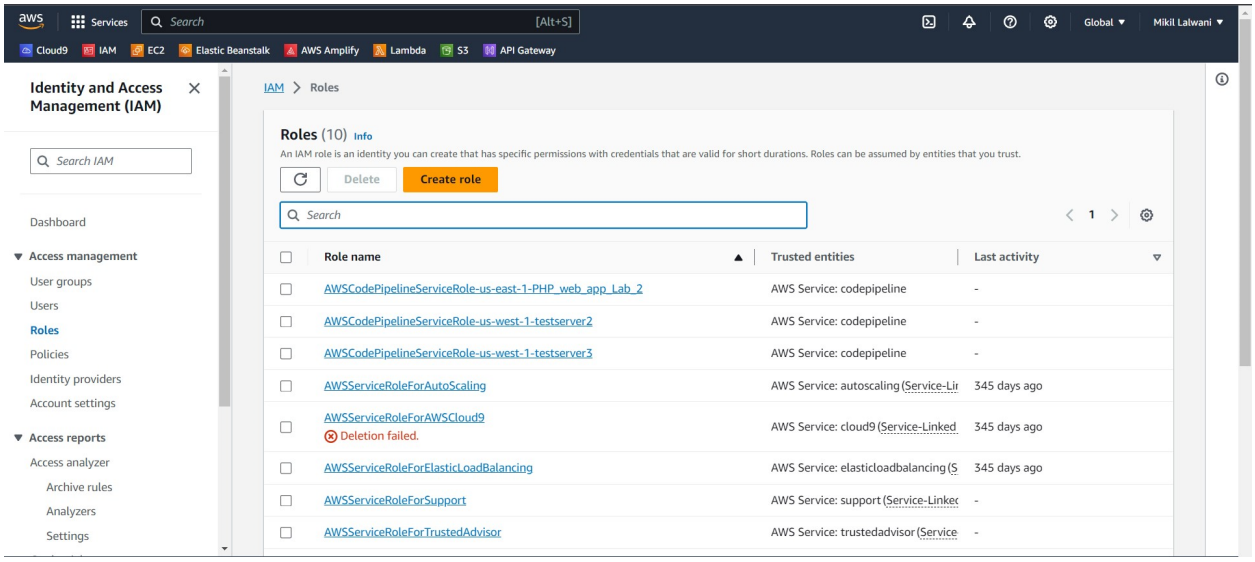
Lab Assignment 2: AWS Analytics

Group no -13 D20B

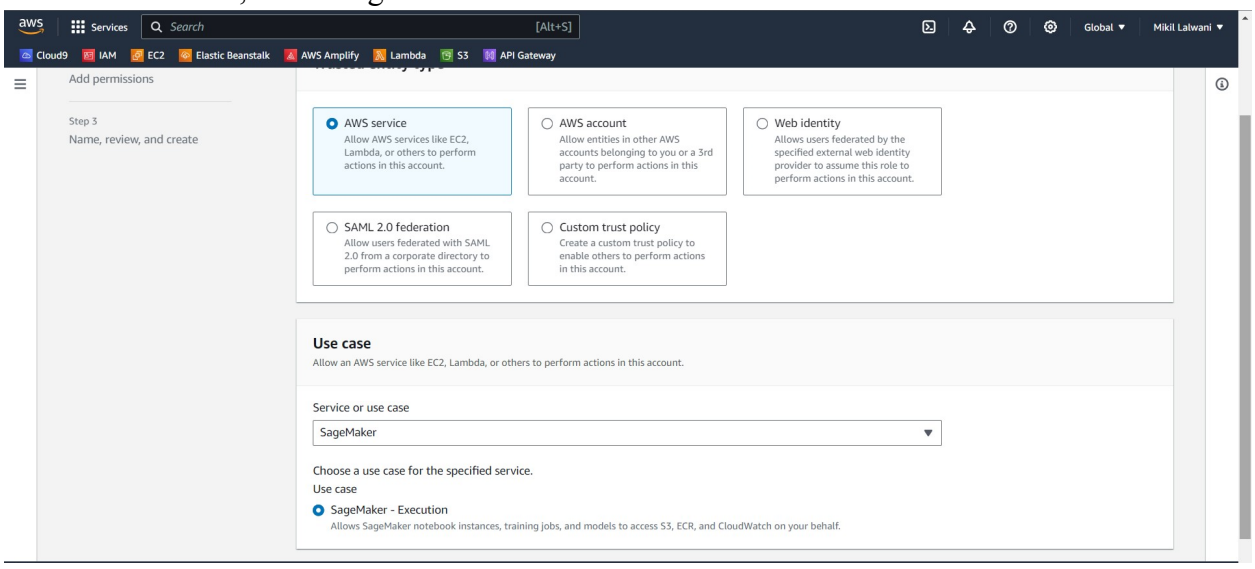
AWS IoT Analytics:

Step 1: Create SageMaker Role

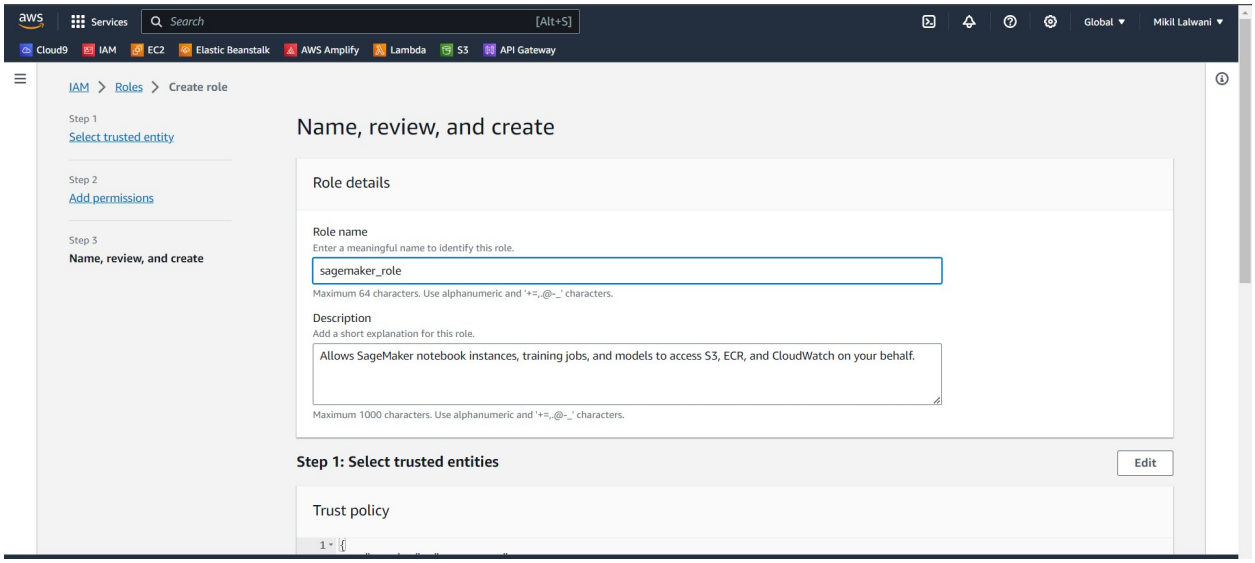
1. Login to AWS Management Console and change the region to Ireland.
2. Goto the IAM Management console, click on the Roles menu on the left and then click on the Create role button.



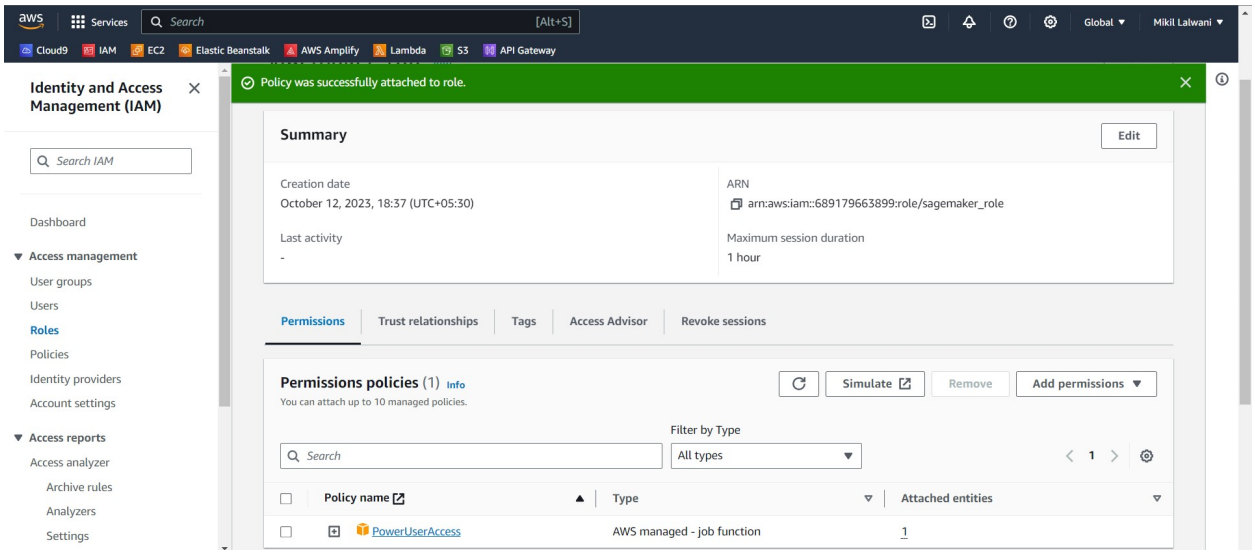
3. On the next screen, select SageMaker as the service



4. On the next screen, type in **sagemaker_role** as the role name and click on the Create role button.



5. The role is created in no time. Open the **sagemaker_role** role details, remove AmazonSageMakerFullAccess policy and attach **PowerUserAccess** policy to the role.

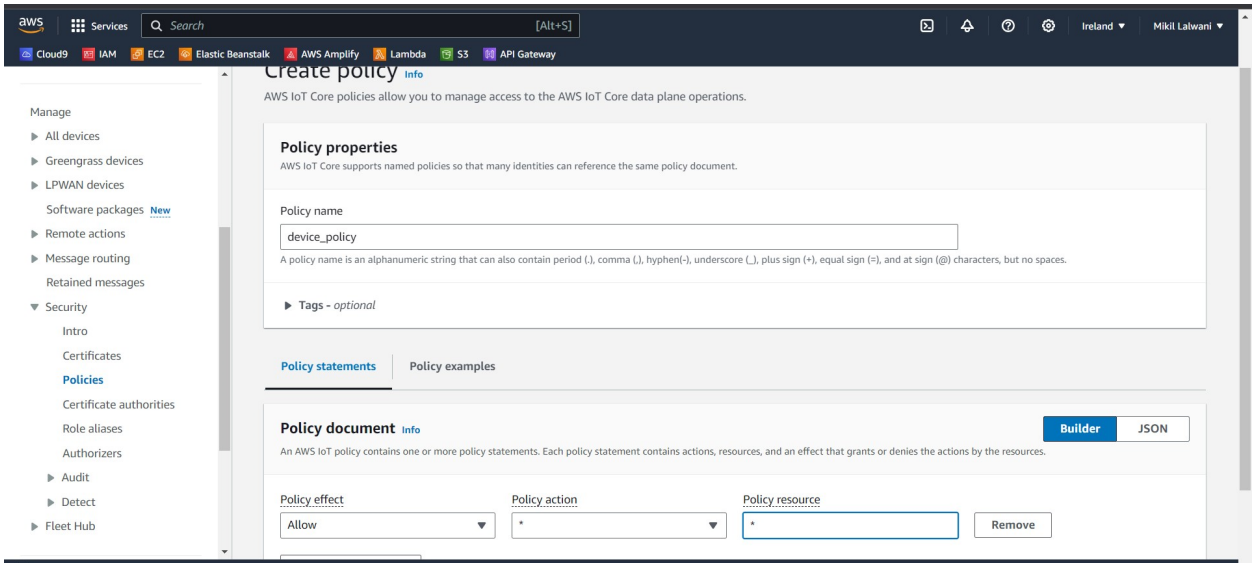


Step 2: Register the Device

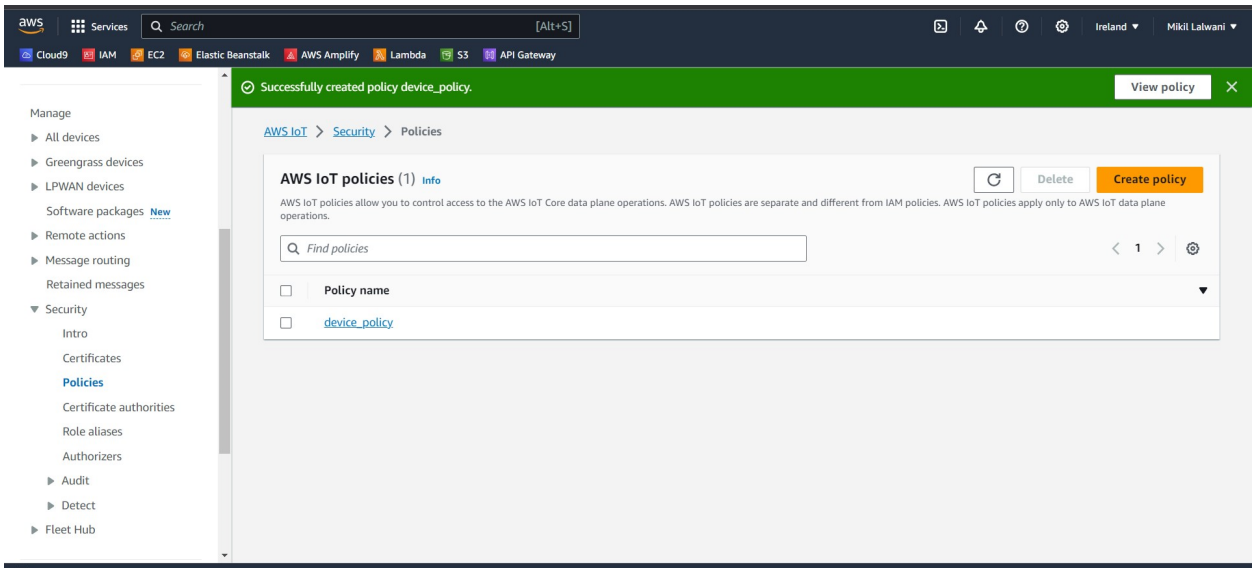
You register the device as Thing in the AWS IoT Core.

1. 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 Security on the left and then click on the Create a policy button.

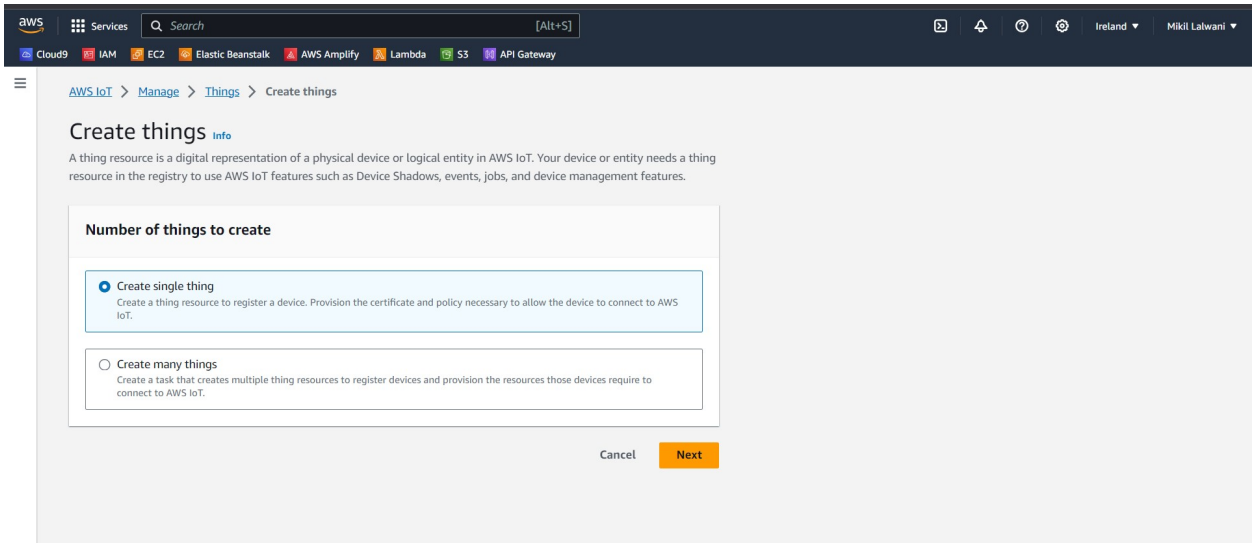
2. Enter policy name as **device_policy**, select **Allow** for the Policy Effect, enter "*" for the Policy Action and enter "*" for the Policy Resource, and click on the Create button.



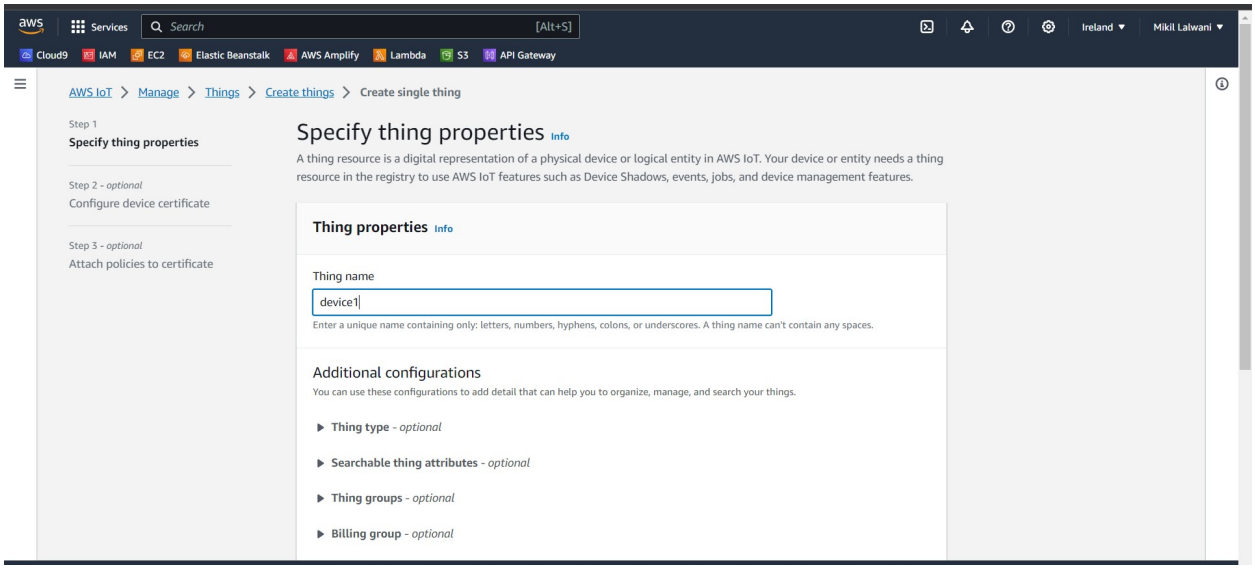
3. The policy is ready. After creating the policy, you will now create a device as a thing and attach the policy to it.



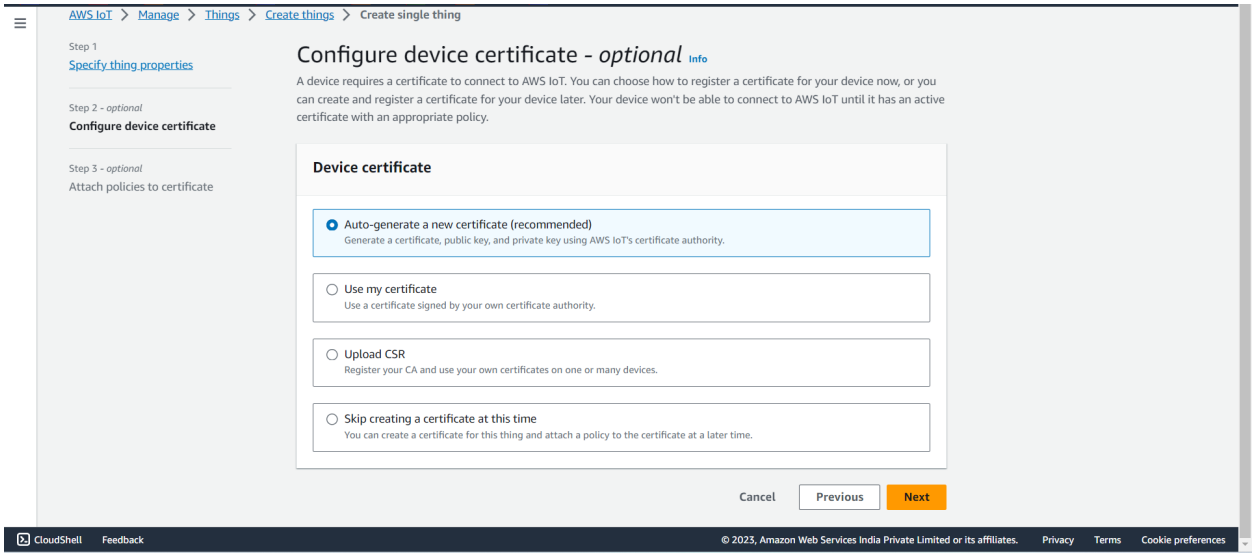
4. On the AWS IoT Core console, click on the Things menu under All devices on the left and then click on the Create a single thing.



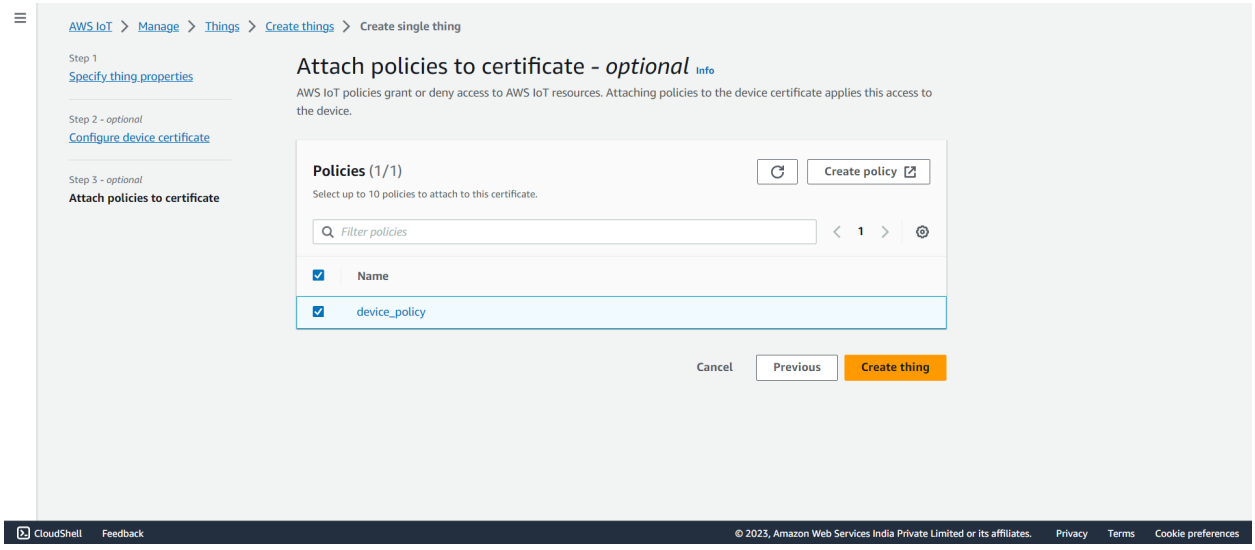
5. Enter the device name as **device1** and click on the Next button.



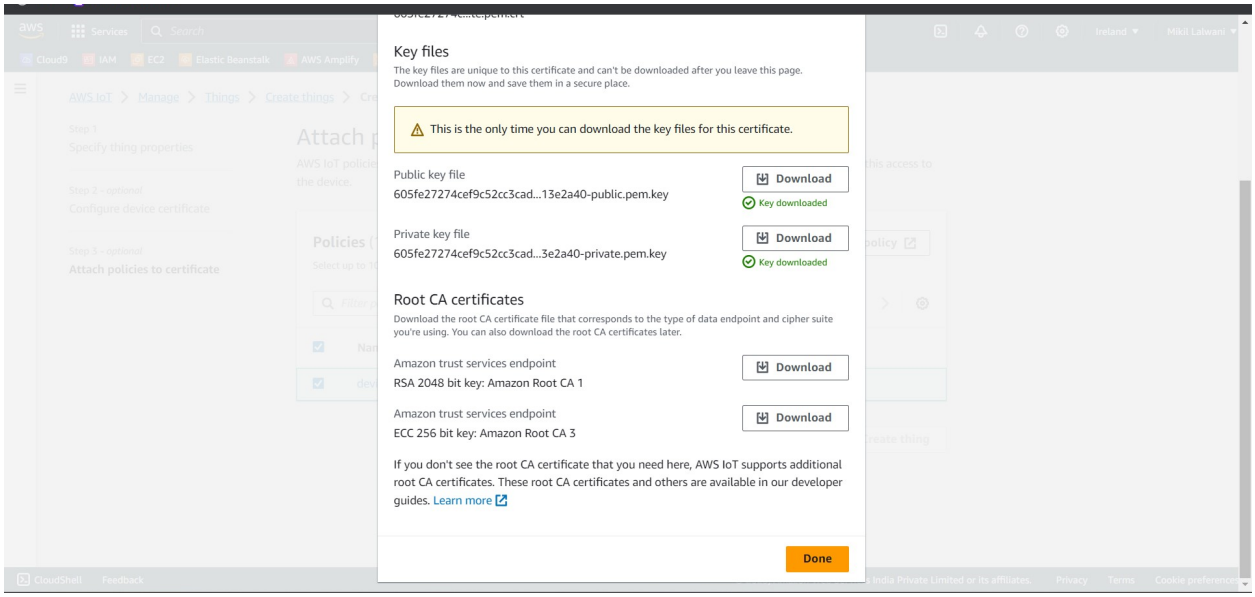
6. On the next screen, select Auto -generate a new certificate and click on Next.



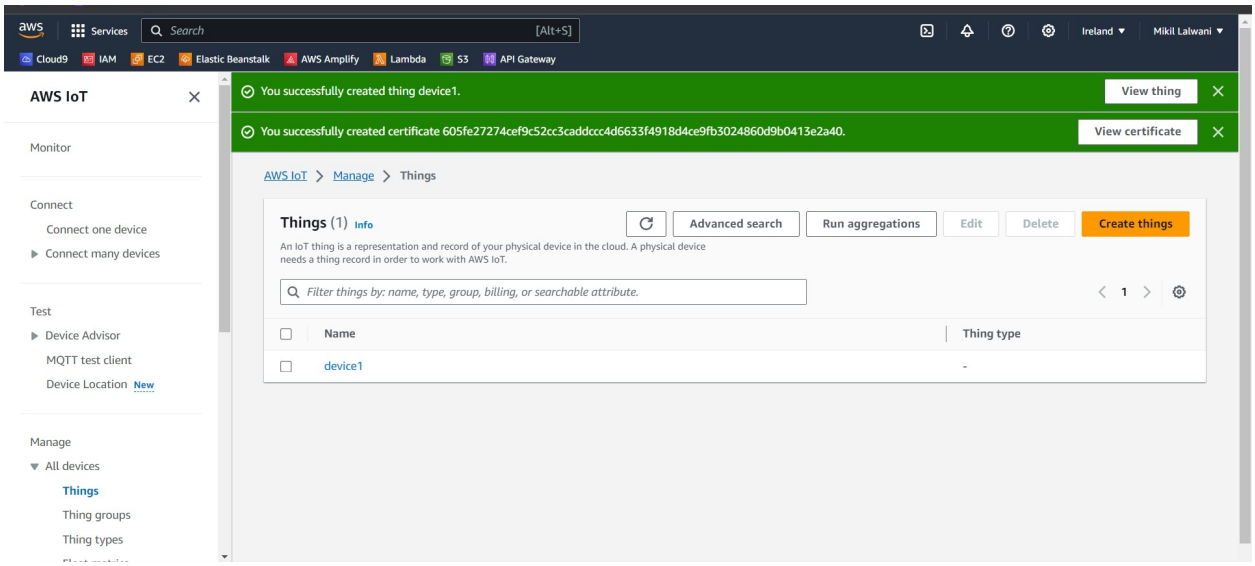
7. Select the created policy in the earlier step and click on create a thing.



8. Download all the keys and certificates in one folder.



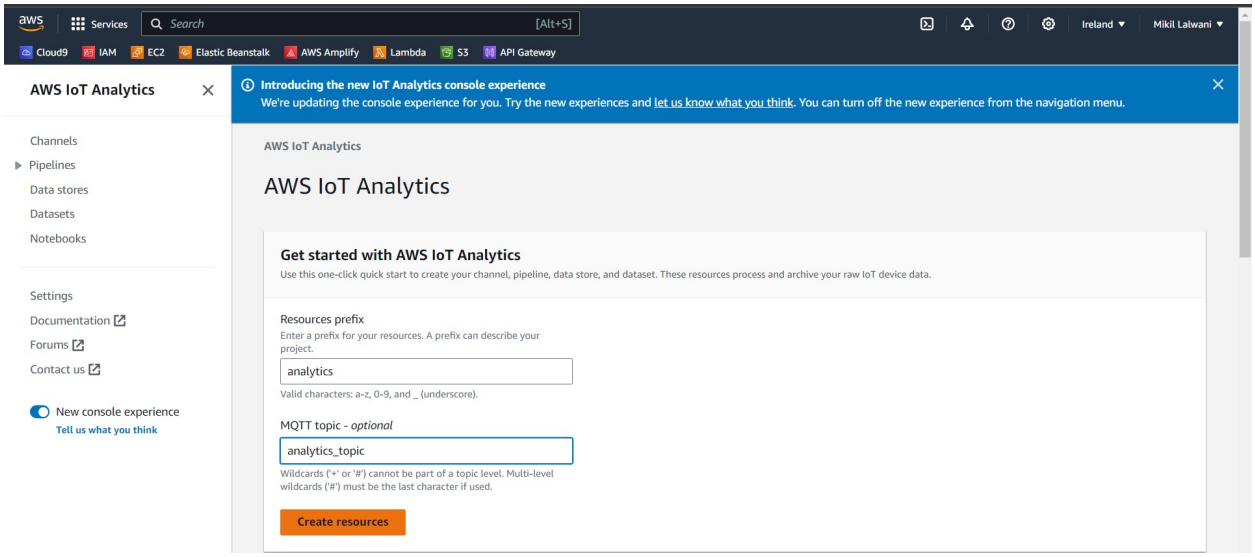
9. After clicking on the Done button your thing will be created.



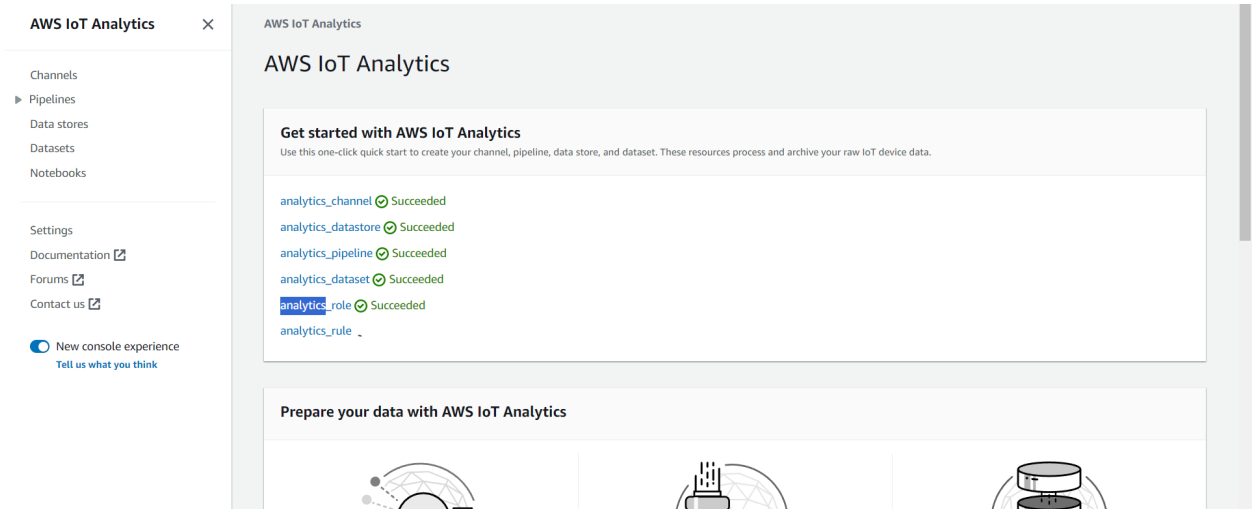
Step 3: Configure IoT Analytics

You configure AWS IoT Analytics which will basically create a channel, pipeline, data store and data set.

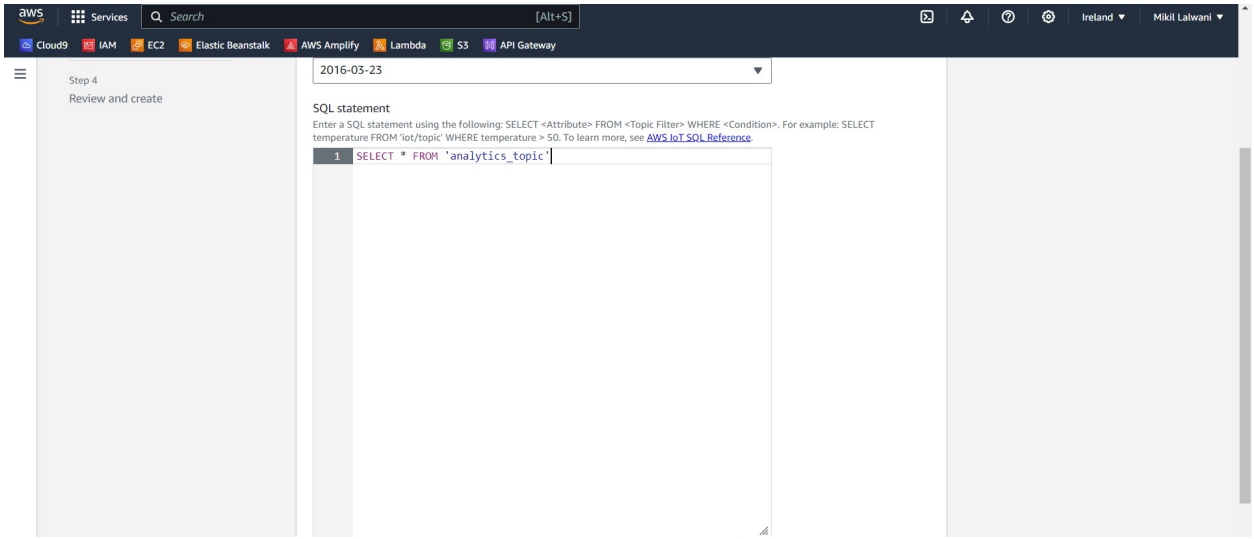
1. Goto AWS IoT Analytics management console. Type in analytics as the resource prefix and type in **analytics_topic** as the topic and then click on Create resources.



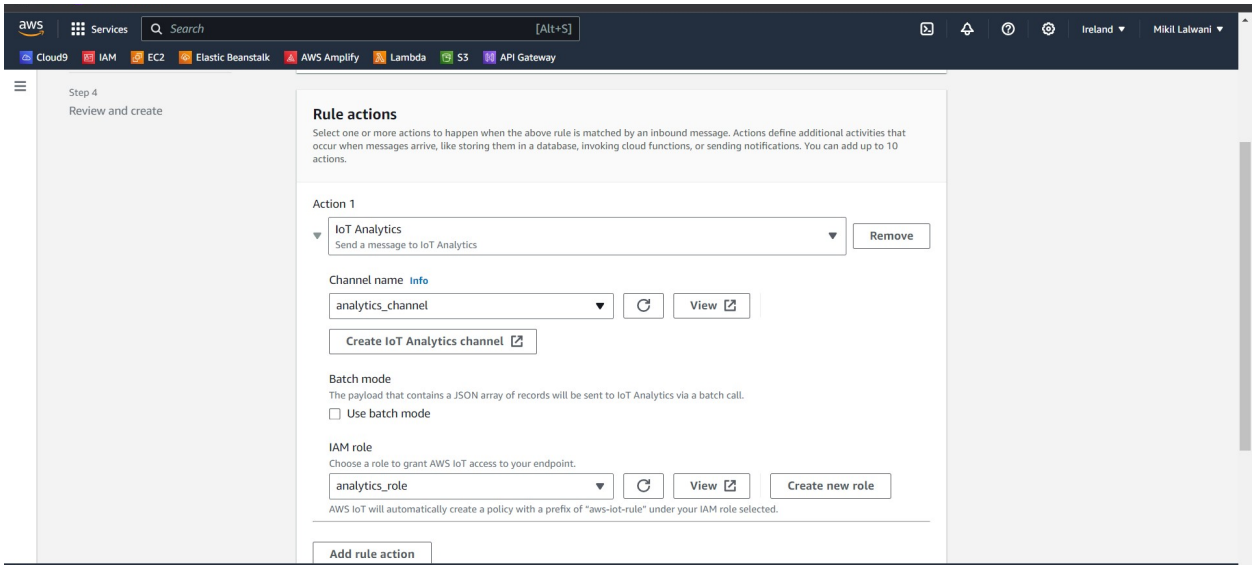
2. In this step all the resources will be created. But for me **analytics_rule** was not created so i have to manually create it.



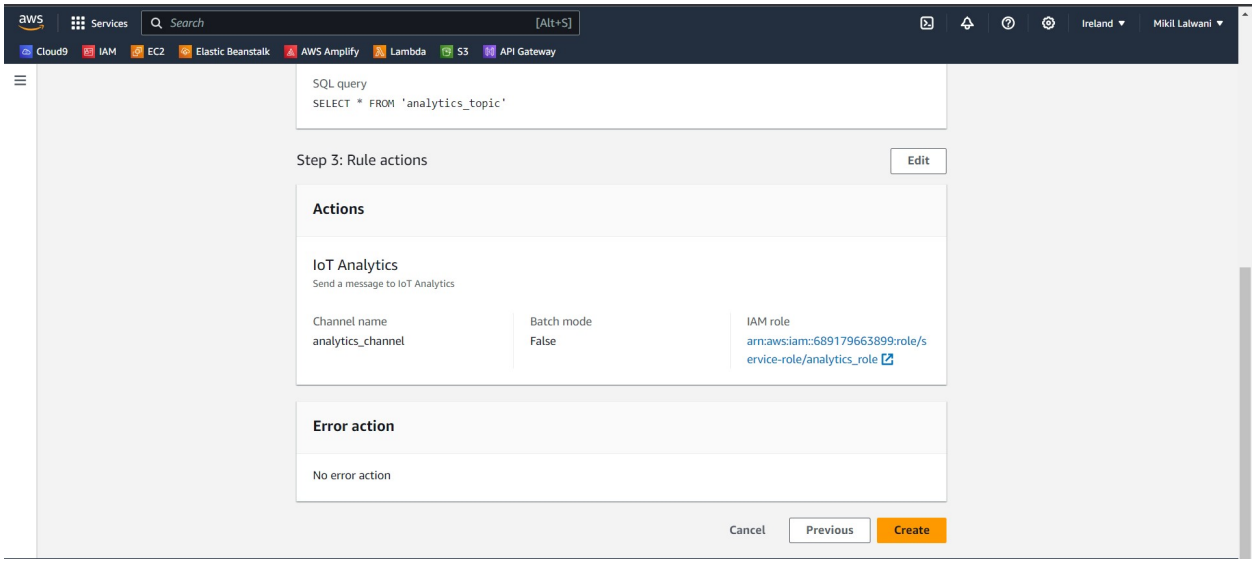
3. To create **analytics_rule** manually. Go to IOT Core console and click on rules engine. And enter the following query in SQL statement.



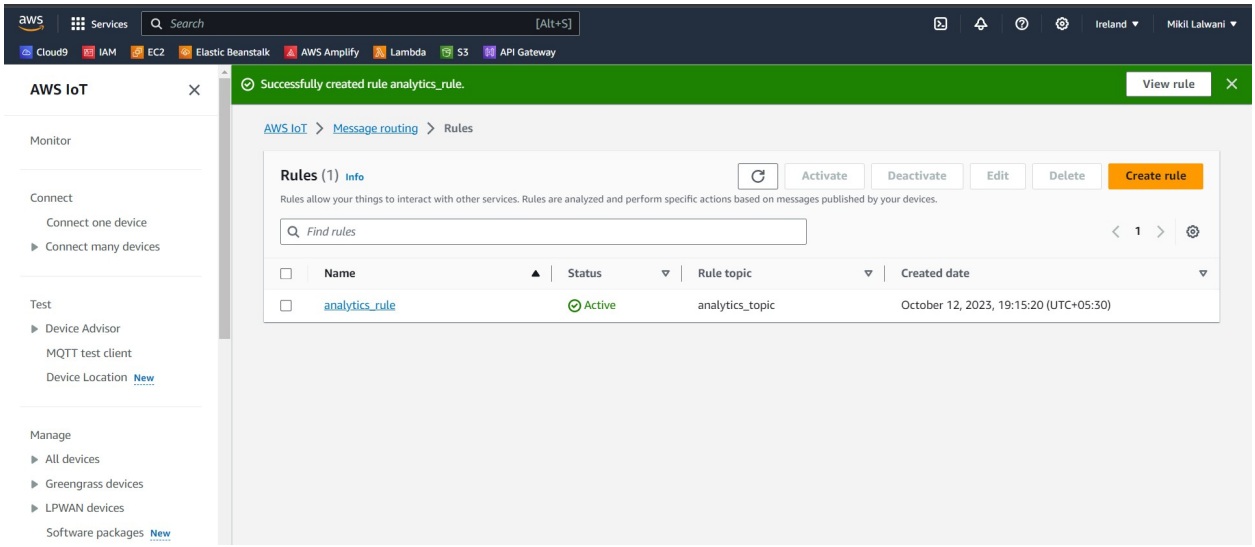
4. First select IOT Analytics and then select the channel name and IAM role and click on Create a new role.



5. In the next step click on Create.



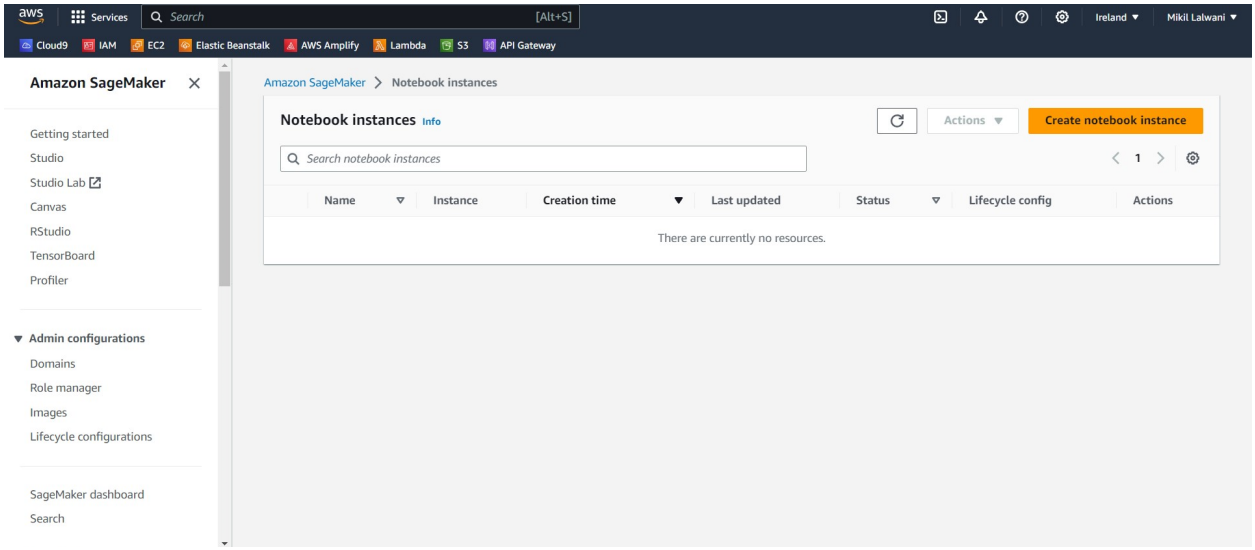
6. Our rule is finally created. The IoT Analytics resources are ready.



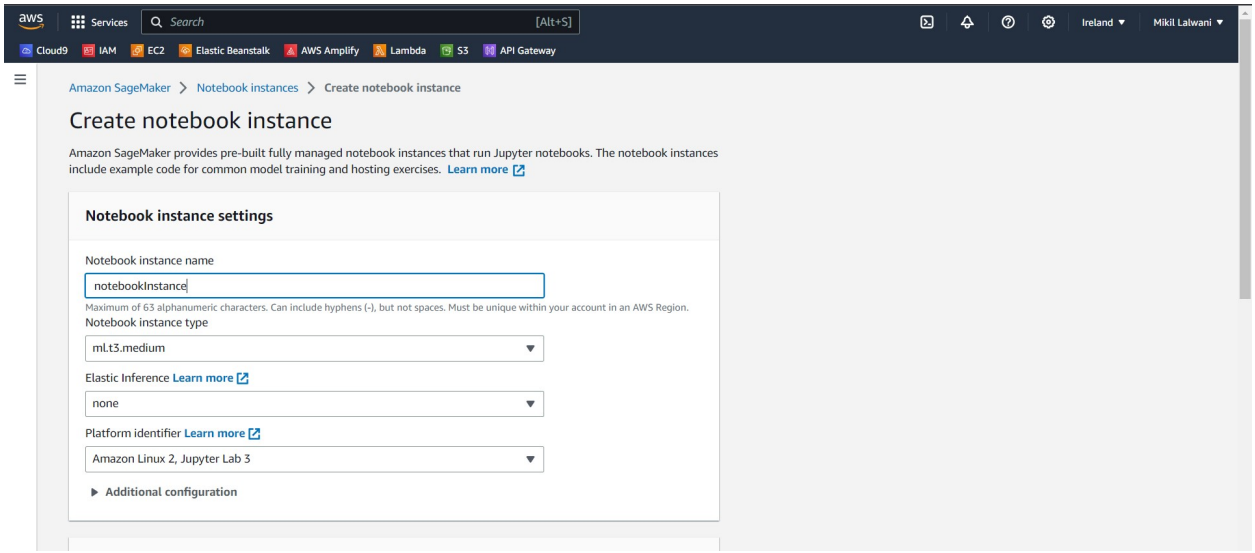
Step 4: Configure SageMaker Notebook

You launch the SageMaker Notebook and associate it with AWS IoT Analytics data set so that you can use Jupyter Notebook to analyze data in the dataset.

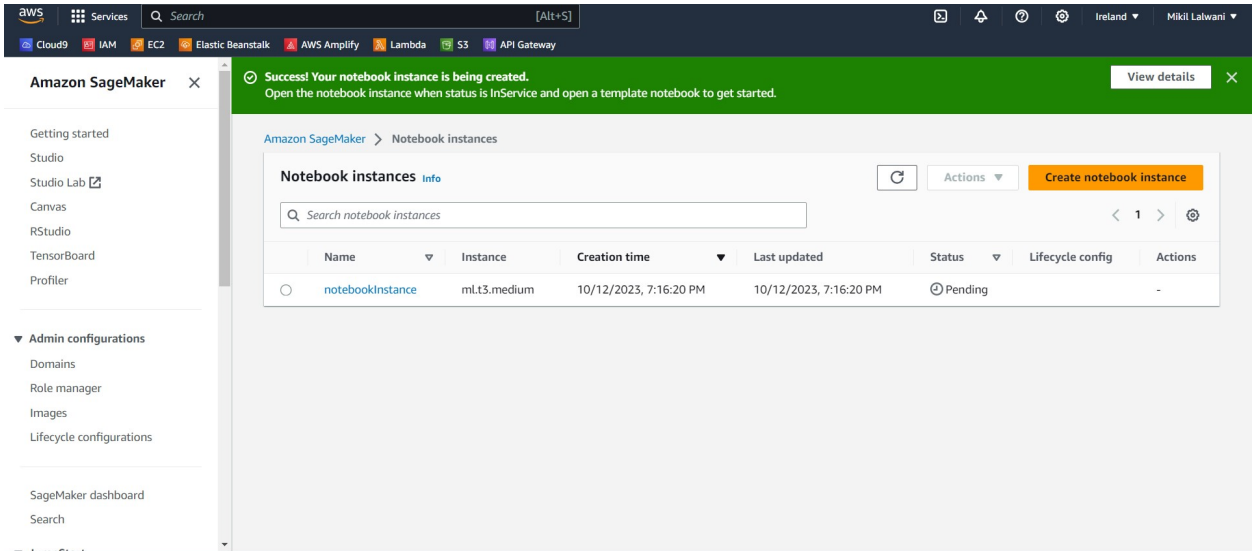
1. Goto Amazon SageMaker console. Select Notebook instances in the left and then click on the Create notebook instance button.



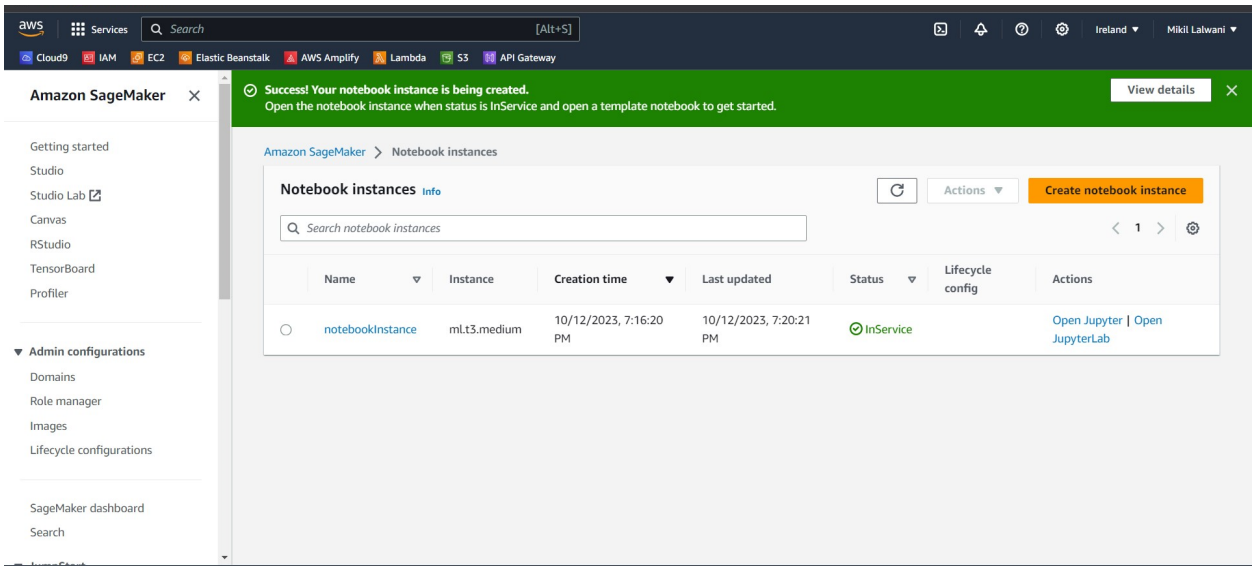
2. On the next screen, type in **notebook_instance** as the notebook instance name, select **sagemaker_role** as the IAM role. Leave rest of the configuration as the default and click on the Create notebook instance button.



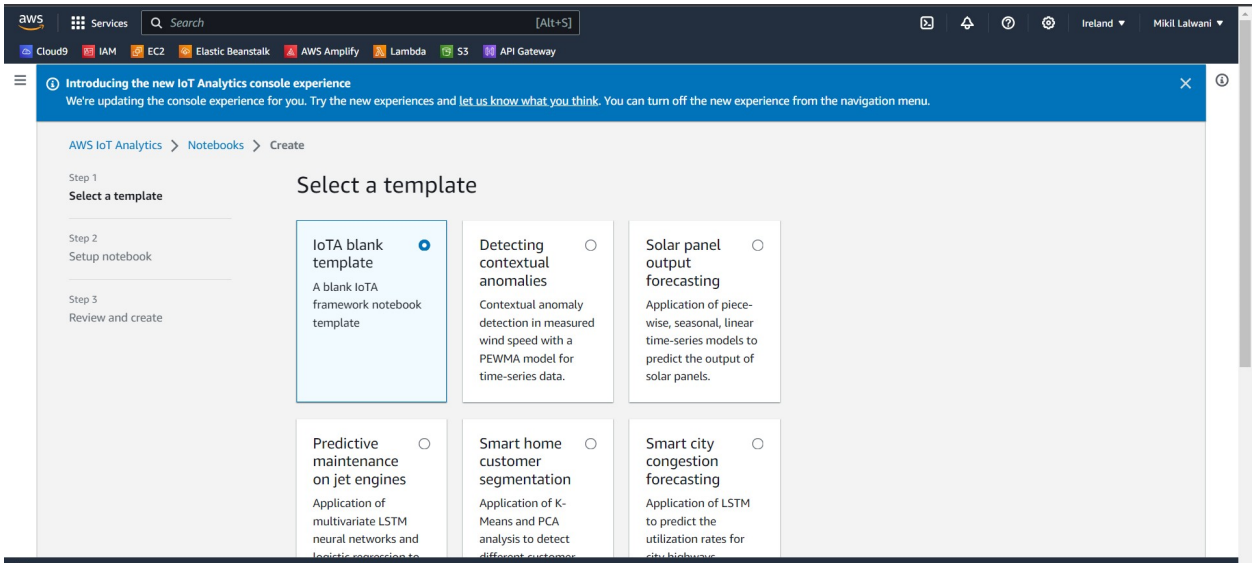
3. The notebook instance launch starts. Wait till the status changes to InService.



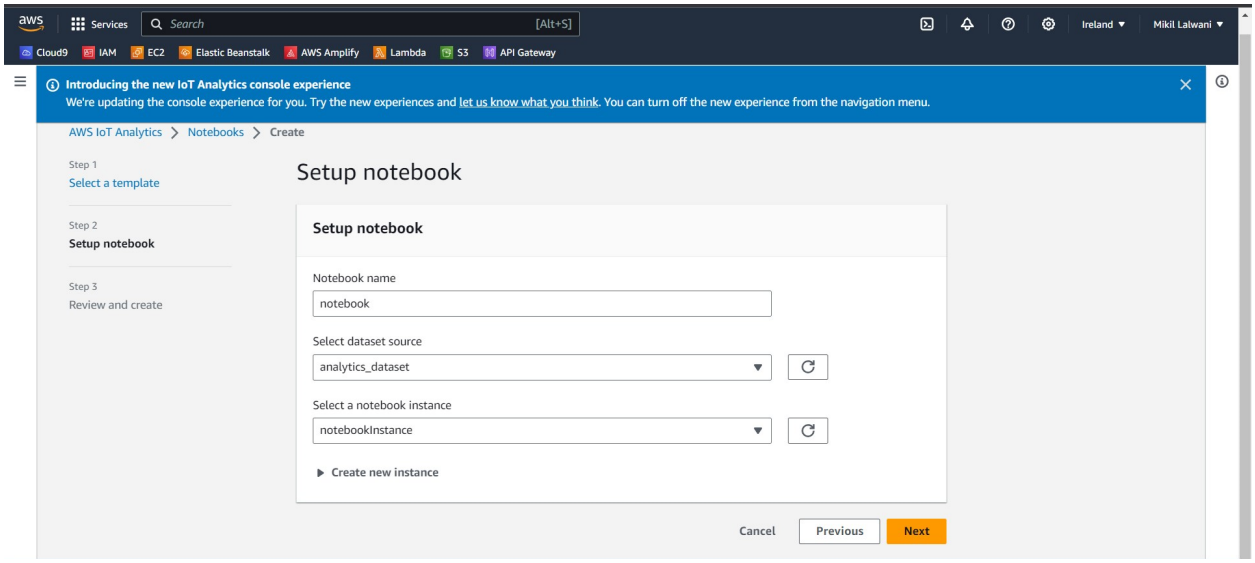
4. The notebook is ready. Let's associate it with the AWS IoT Analytics data set.

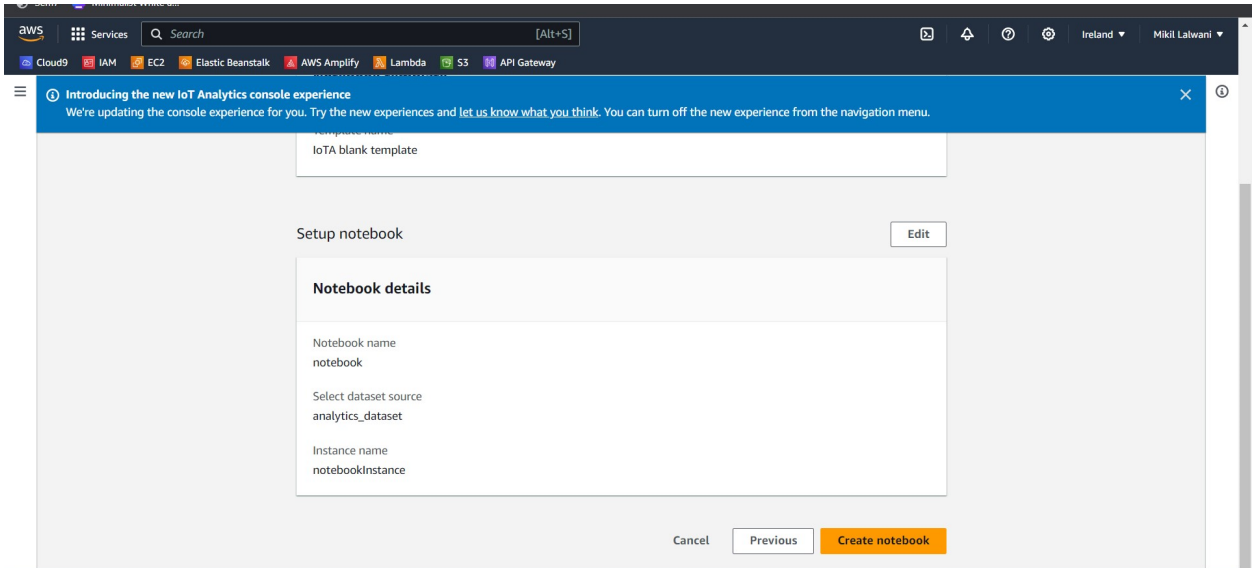


5. Goto AWS IoT Analytics management console. Click on the Notebooks menu in the left and then click on the Create button.

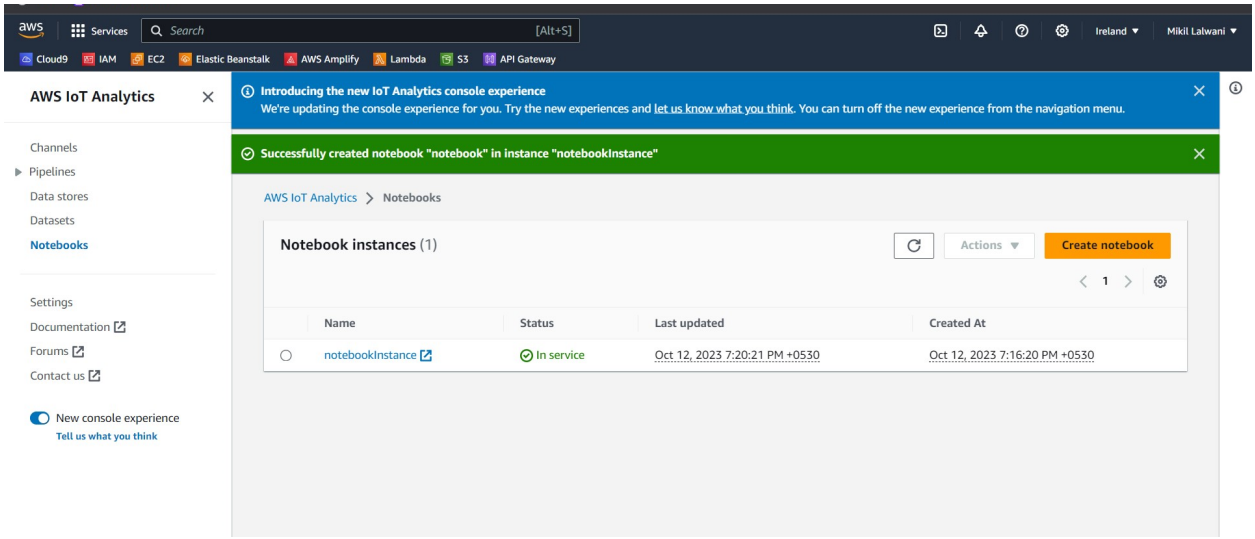


6. On the next screen, type in a notebook for the name. Select **analytics_dataset** for the data set. Select **notebook_instance** for the notebook instance. Finally, click on the Create Notebook button.



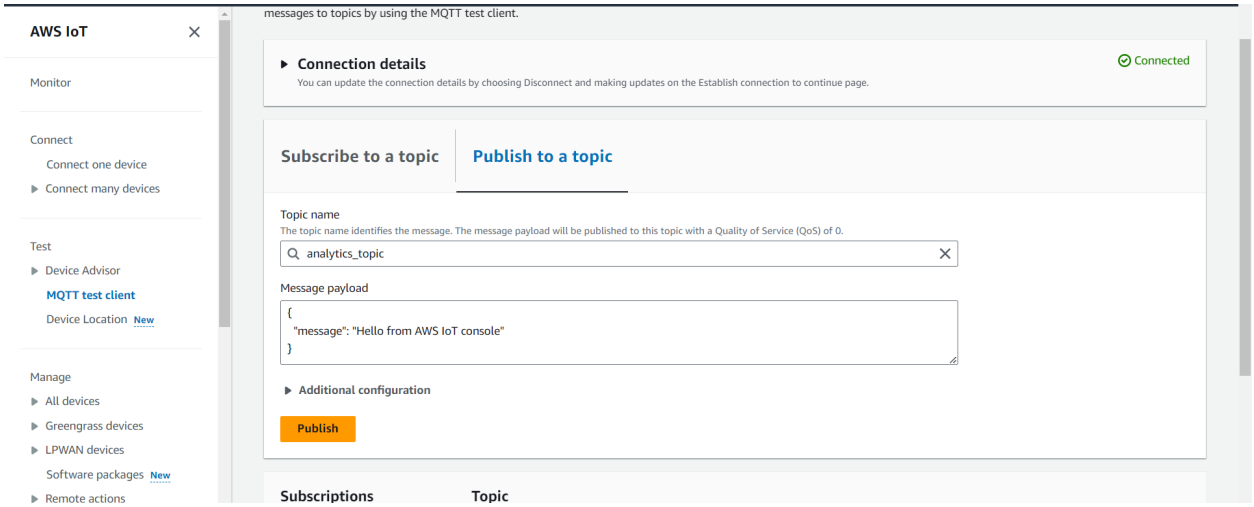


7. The notebook configuration is ready.



Step 5: Publish Messages to the Topic

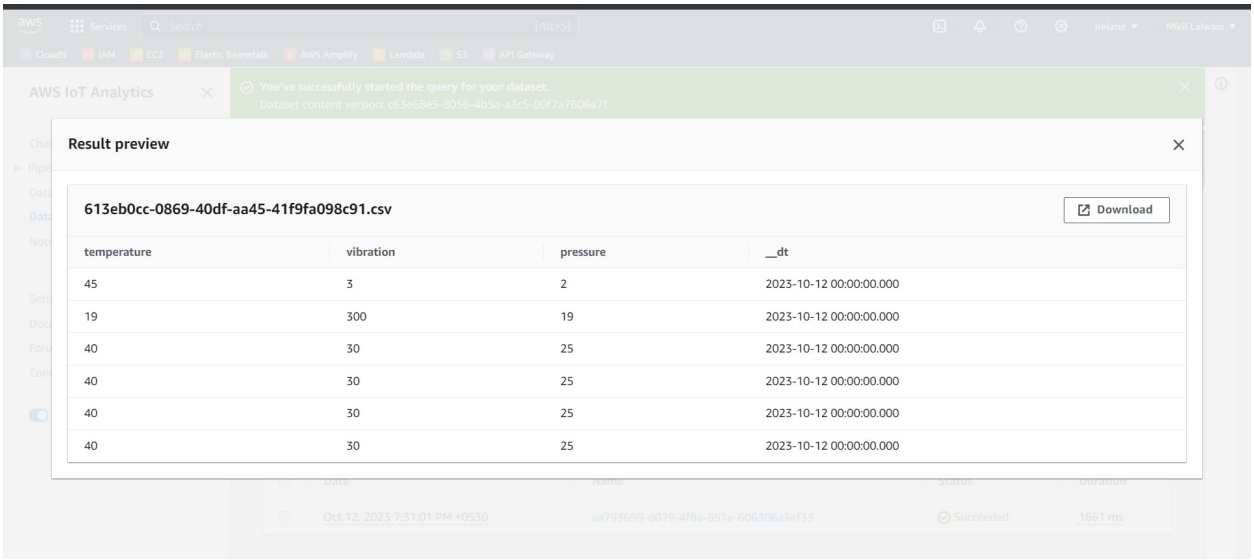
1. On the AWS IoT Core console, click on MQTT test client menu in the left to open MQTT client. Click on the Publish to a topic link.



2. It moves to the publish topic part of the MQTT client, type in **analytics_topic** as the topic name. Copy-Paste the message shown below and click on the Publish to topic button.

   {
   "temperature": 40,
   "vibration": 30,
   "pressure": 25
   }
3. The message is published to the AWS IoT Core. Repeat the previous step to publish by changing the values.
4. All the messages got published. Due to IoT rule which is looking for the messages published to the topic **analytics_topic**, the rule will pick the messages and route to the AWS IoT Analytics Data Set.
5. Let's check the data in the IoT Analytics Data Set. Goto AWS IoT Analytics management console, click on the Data sets menu on the left and click on **analytics_dataset** to see the details.
6. On the next screen, click on the Run now
7. Click on the content and click on the dataset.
8. On the same screen, it will show the data published to the data set. You can see the timestamp has been added by AWS IoT Analytics to the each published message to

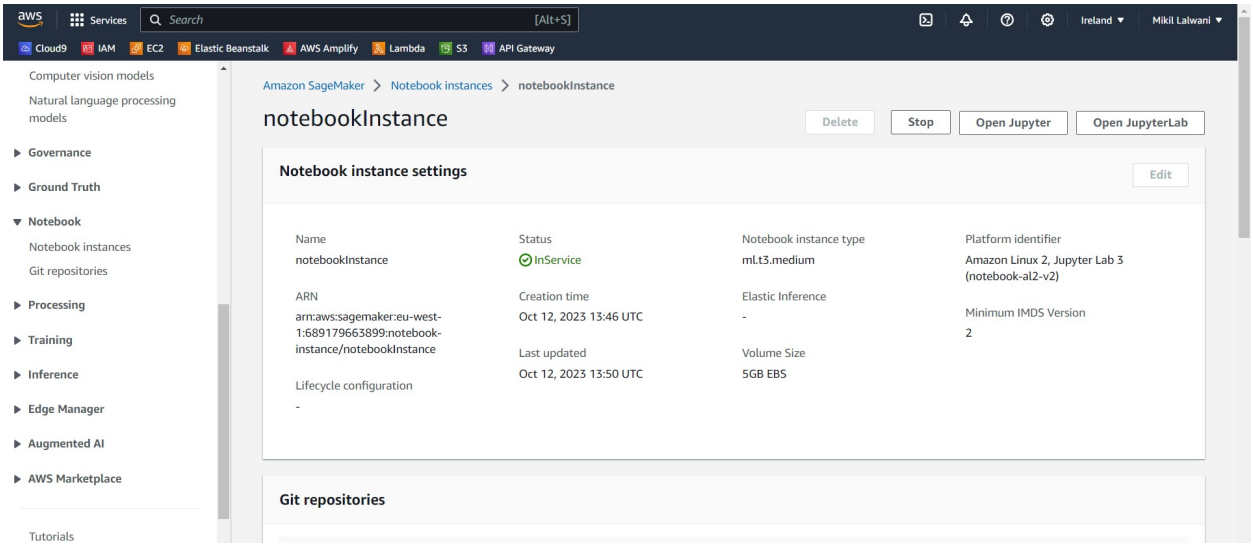
convert it into a time-series data.



Step 6: Analyze Data in Notebook

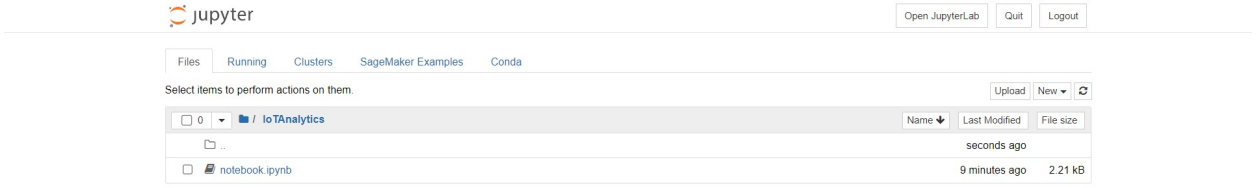
In this step, you use Jupyter Notebook to analyze the data published in the IoT Analytics Data Set.

1. Goto AWS IoT Analytics management console and click on the Notebooks menu on the left. Expand the details for the **notebookinstance** and click on the Open in Jupyter link.

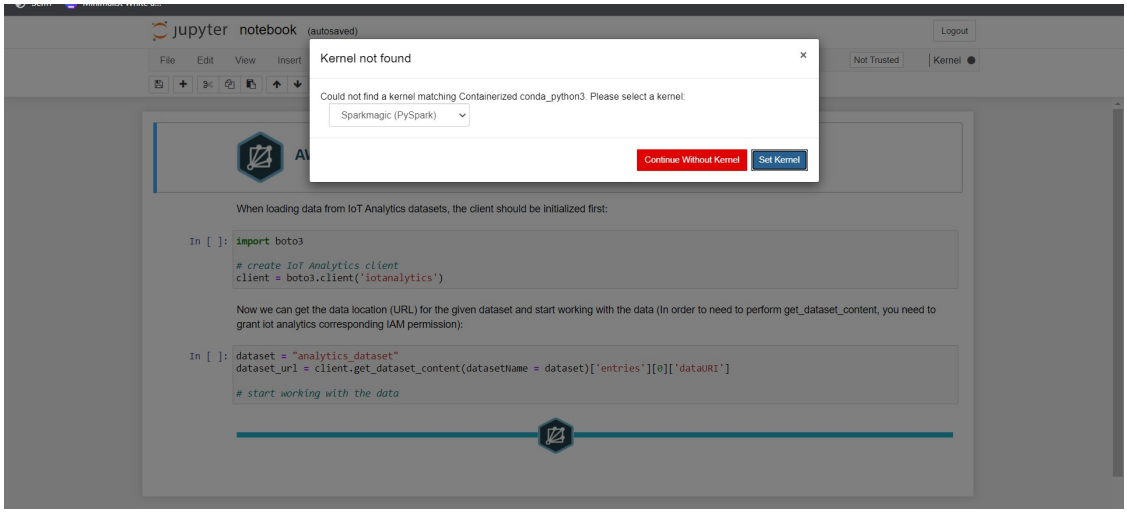


2. Select the IoTAnalytics.

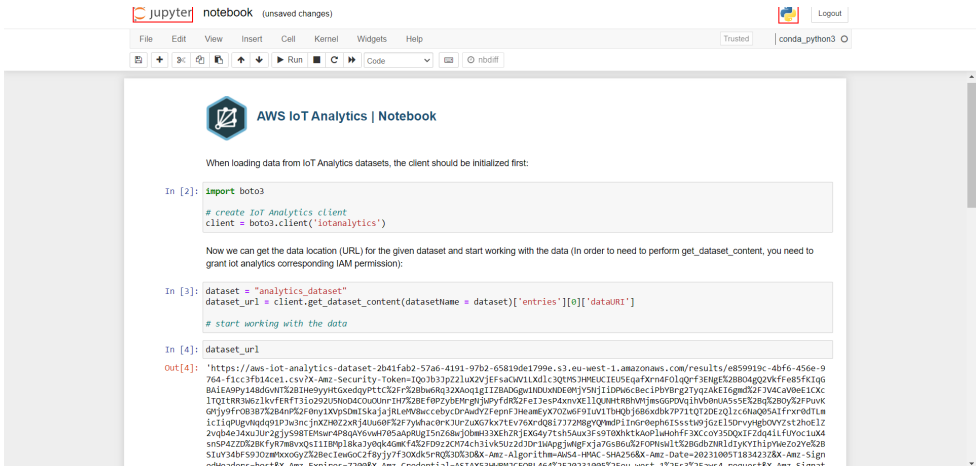
3. Now select the python notebook.



4. It will open Jypyter Notebook in a new browser window or tab. It shows popup for Kernel not found. Select python as the kernel and click on the Set Kernel button.

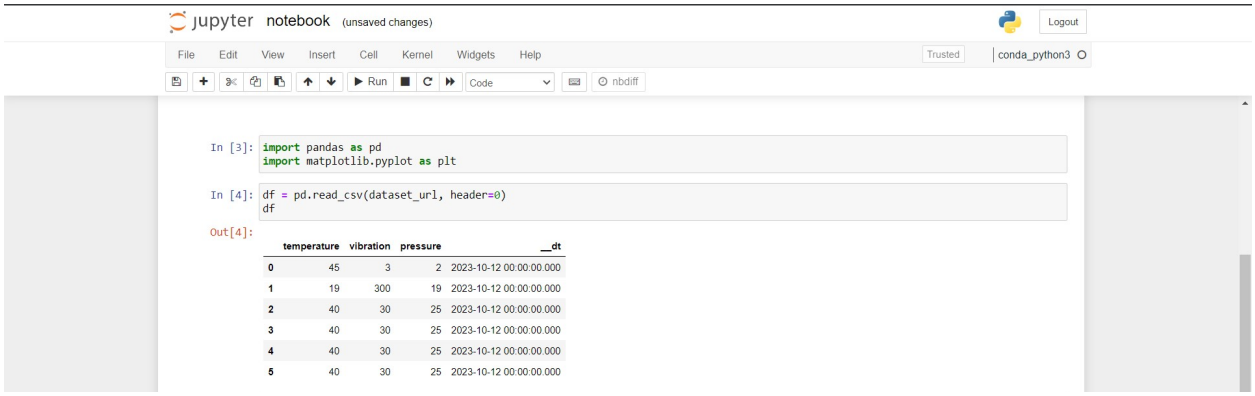


5. The notebook already has some code generated. Run the code in the first cell which creates boto3 client for the IoT Analytics. Run the next cell which gets signed S3 URL dataset_url for the IoT Analytics data set data which is stored in the S3 bucket.

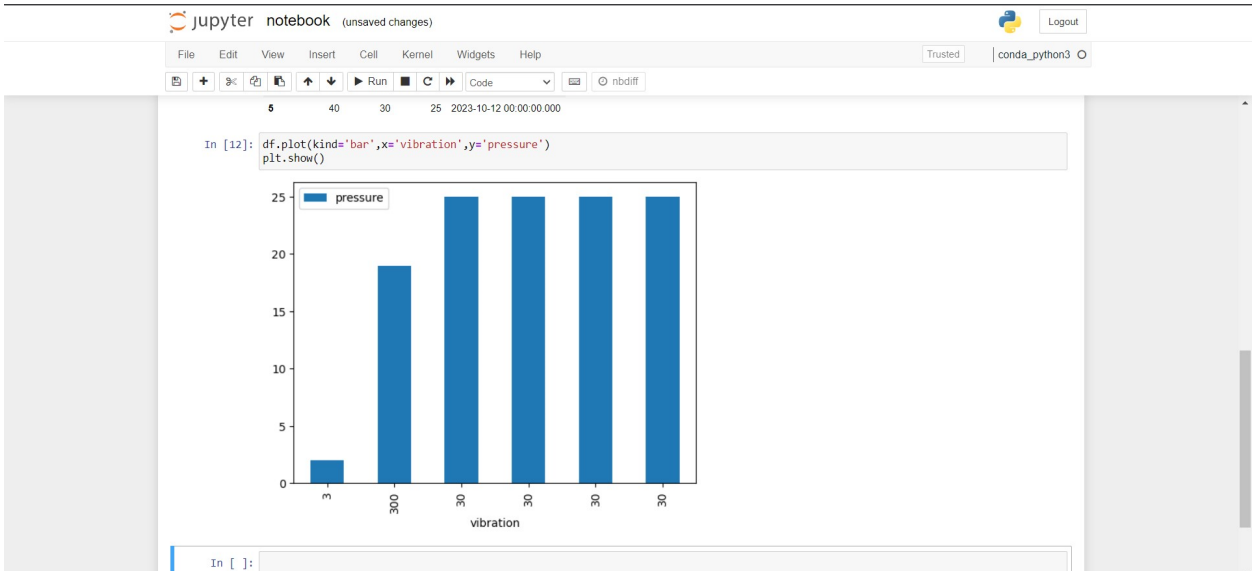


6. Add another cell and run the following code to import pandas and matplotlib libraries for the data analysis. Use read_csv method to read data from the S3 bucket and populate into df dataframe. Finally, print the data from the dataframe.

import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv(dataset_url, header=0)
df



7. Run the following code to plot vibration against pressure in the bar graph.
df.plot(kind='bar',x='vibration',y='pressure')
plt.show()



Conclusion:

The analysis reveals a positive correlation between temperature, vibration, and pressure parameters, indicating potential interdependencies in the system under study.