

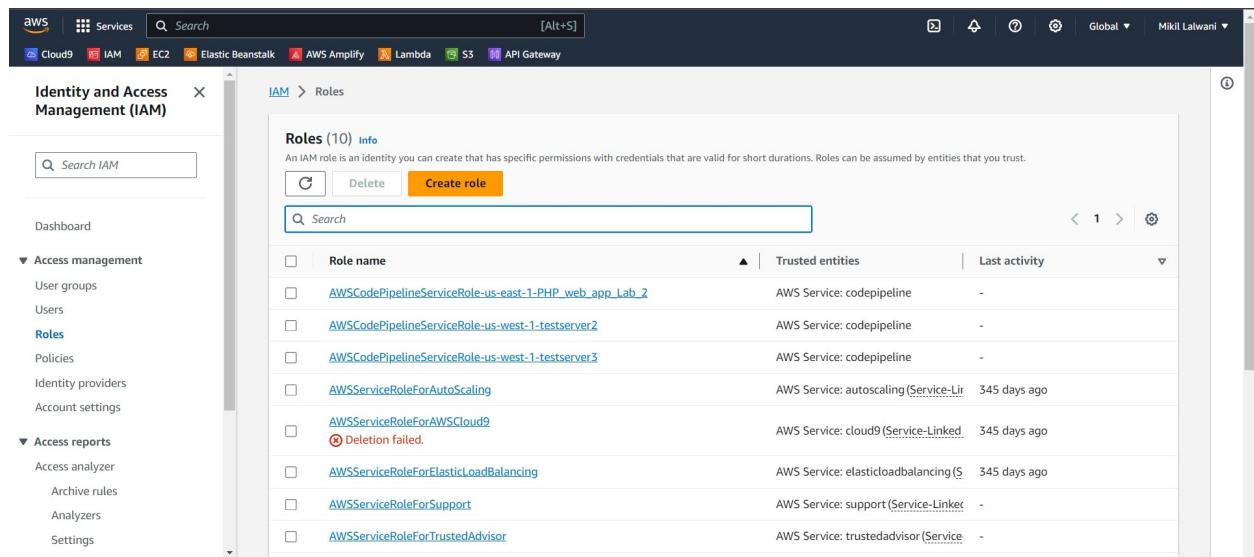
Lab Assignment 2: AWS Analytics

Group no -13 D20B

AWS IoT Analytics:

Step 1: Create SageMaker Role

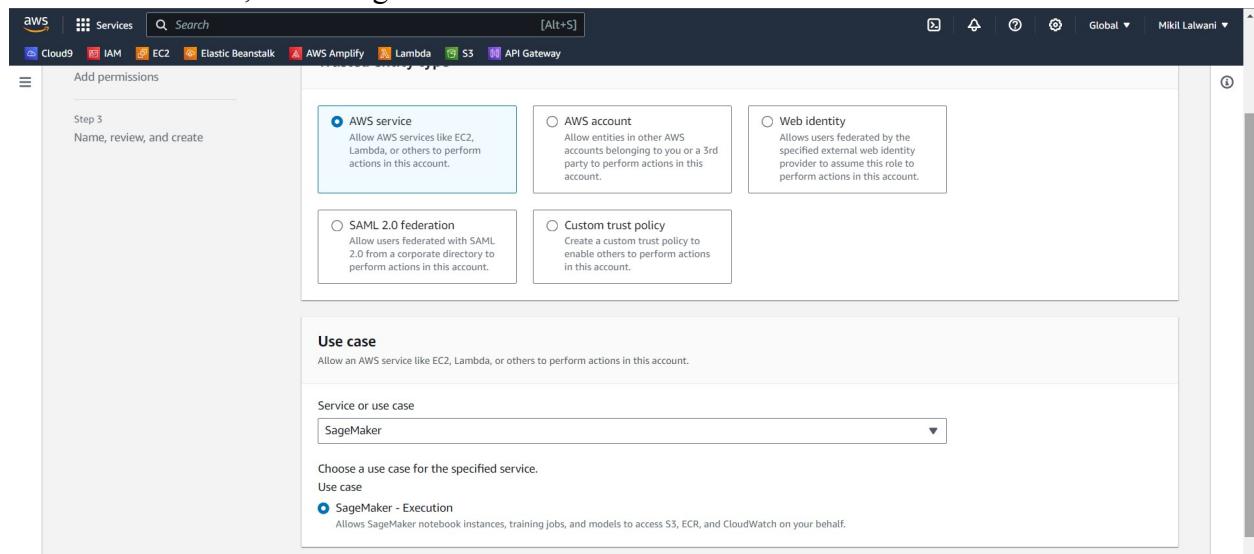
1. Login to AWS Management Console and change the region to Ireland.
2. Goto the IAM Management console, click on the Roles menu on the left and then click on the Create role button.



The screenshot shows the AWS IAM Roles page. The left sidebar includes sections for Access management (User groups, Users, Roles, Policies, Identity providers, Account settings) and Access reports (Access analyzer, Archive rules, Analyzers, Settings). The main content area displays a table of existing roles, each with a checkbox, role name, trusted entities, and last activity. One role, "AWSCodePipelineServiceRoleForAWSCloud9", has a red error icon next to it, indicating a deletion failed.

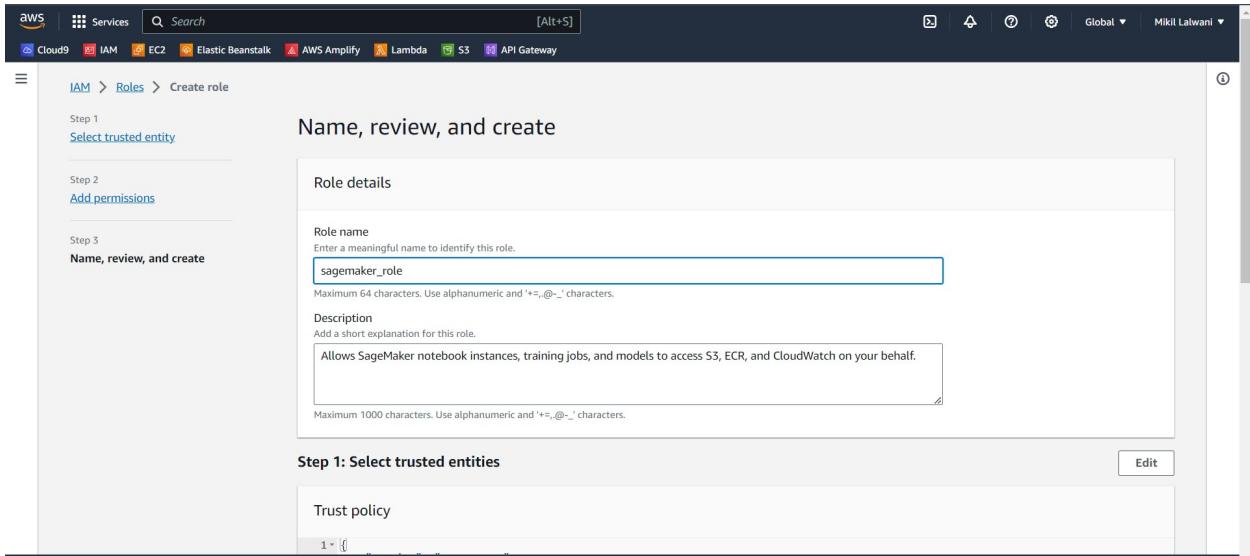
Role name	Trusted entities	Last activity
AWSCodePipelineServiceRole-us-east-1-PHP_web_app_Lab_2	AWS Service: codepipeline	-
AWSCodePipelineServiceRole-us-west-1-testserver2	AWS Service: codepipeline	-
AWSCodePipelineServiceRole-us-west-1-testserver3	AWS Service: codepipeline	-
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Liner)	345 days ago
AWSServiceRoleForAWSCloud9	AWS Service: cloud9 (Service-Linked)	345 days ago
AWSServiceRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (S)	345 days ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service)	-

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

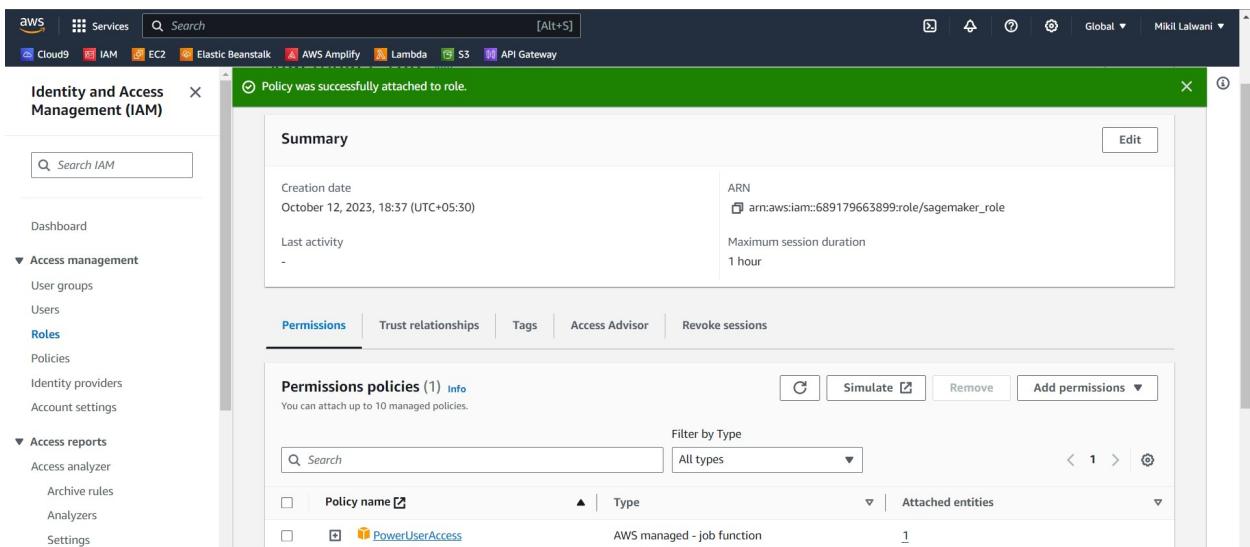


The screenshot shows the "Add permissions" step of the IAM Role creation wizard. It includes sections for "Step 3 Name, review, and create", "Use case" (allowing actions from EC2, Lambda, or others), and "Service or use case" (set to SageMaker). Under "Service or use case", a dropdown menu shows "SageMaker". At the bottom, a section for "Choose a use case for the specified service" lists "SageMaker - Execution" as selected, with a note that it allows SageMaker notebook instances, training jobs, and models to access S3, ECR, and CloudWatch on behalf of the user.

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.

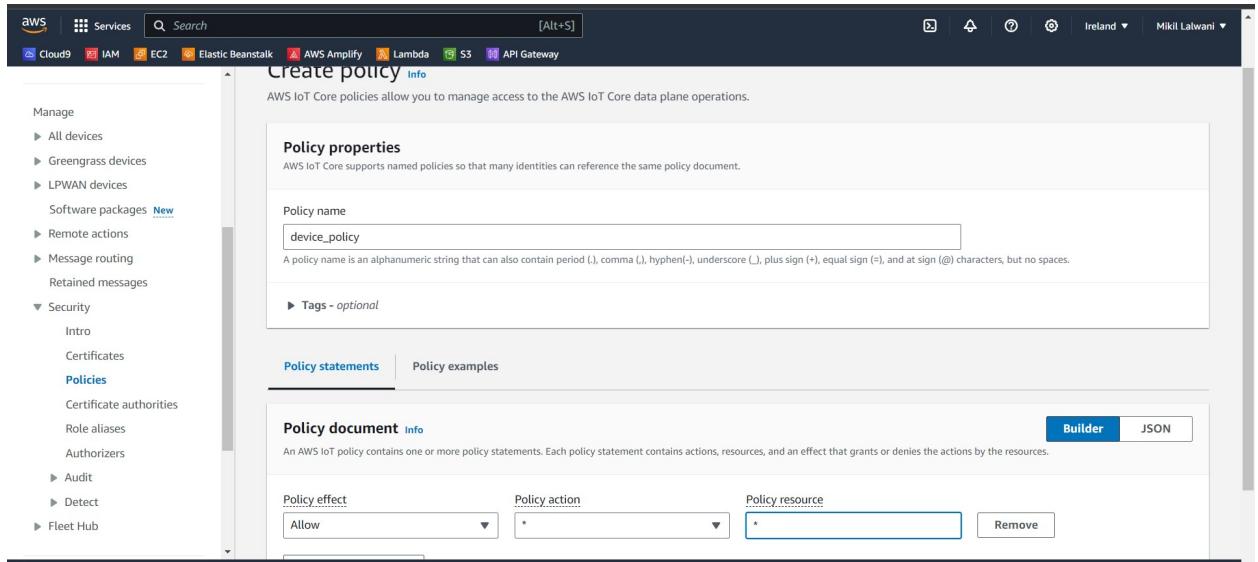


Step 2: Register the Device

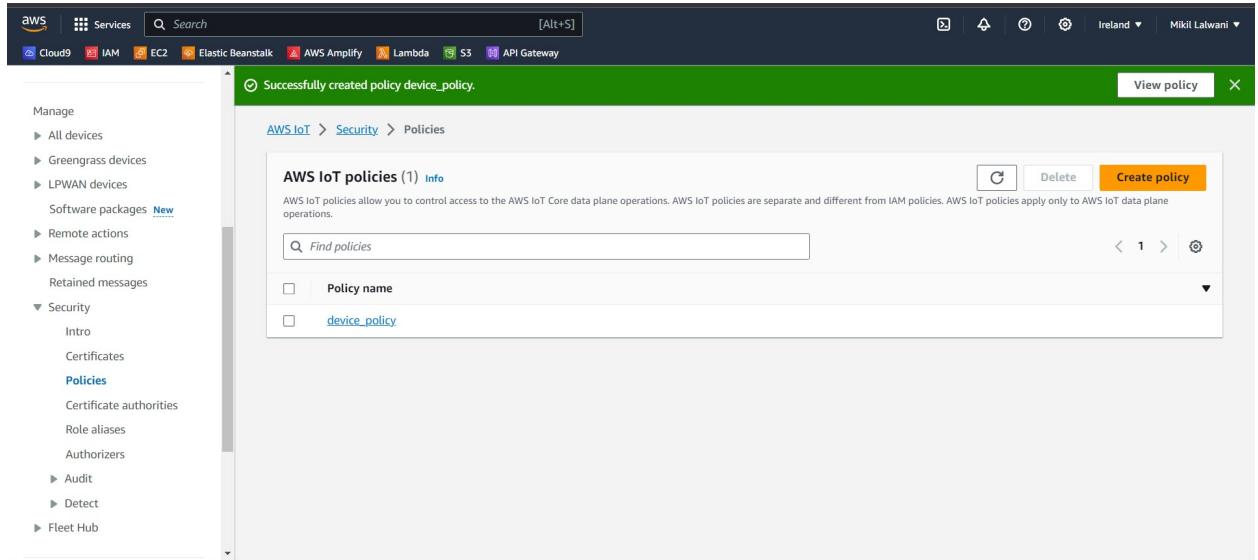
You register the device as Thing in the AWS IoT Core.

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.

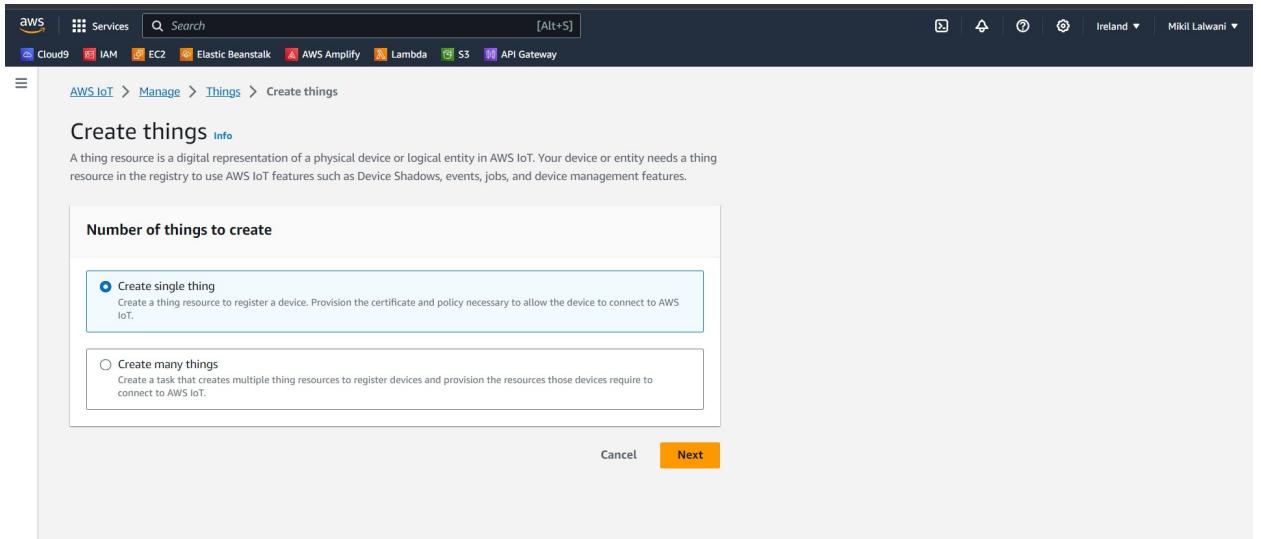
2. Enter policy name as **device_policy**, select **Allow** for the Policy Effect, enter "*" for the Policy Action and enter "*" for the Policy Resource, and click on the Create button.



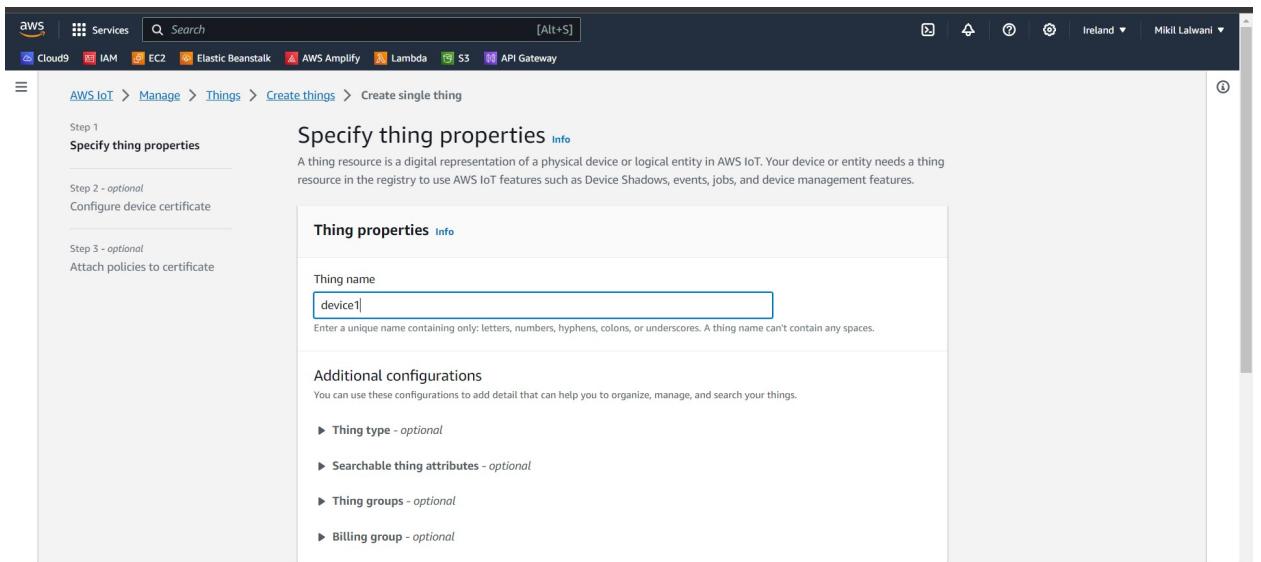
3. The policy is ready. After creating the policy, you will now create a device as a thing and attach the policy to it.



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.

AWS IoT > Manage > Things > Create things > Create single thing

Step 1
Specify thing properties

Step 2 - optional
Configure device certificate

Step 3 - optional
Attach policies to certificate

Configure device certificate - optional Info

A device requires a certificate to connect to AWS IoT. You can choose how to register a certificate for your device now, or you can create and register a certificate for your device later. Your device won't be able to connect to AWS IoT until it has an active certificate with an appropriate policy.

Device certificate

Auto-generate a new certificate (recommended)
Generate a certificate, public key, and private key using AWS IoT's certificate authority.

Use my certificate
Use a certificate signed by your own certificate authority.

Upload CSR
Register your CA and use your own certificates on one or many devices.

Skip creating a certificate at this time
You can create a certificate for this thing and attach a policy to the certificate at a later time.

Cancel Previous Next

CloudShell Feedback © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

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

AWS IoT > Manage > Things > Create things > Create single thing

Step 1
Specify thing properties

Step 2 - optional
Configure device certificate

Step 3 - optional
Attach policies to certificate

Attach policies to certificate - optional Info

AWS IoT policies grant or deny access to AWS IoT resources. Attaching policies to the device certificate applies this access to the device.

Policies (1/1)

Select up to 10 policies to attach to this certificate.

Name

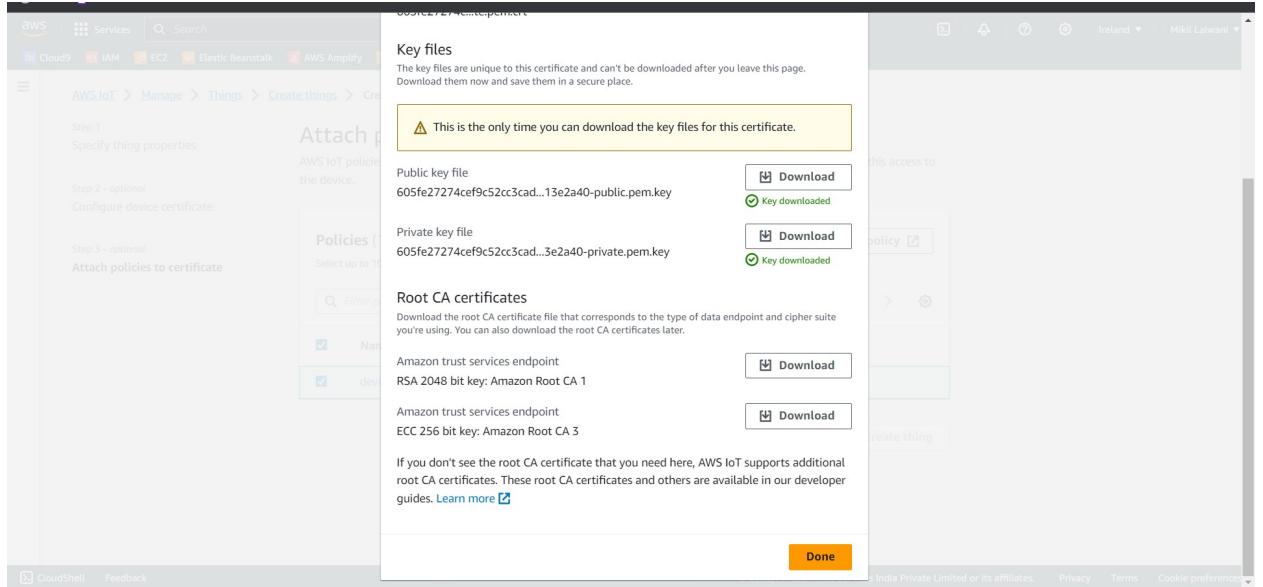
device_policy

Filter policies < 1 > @

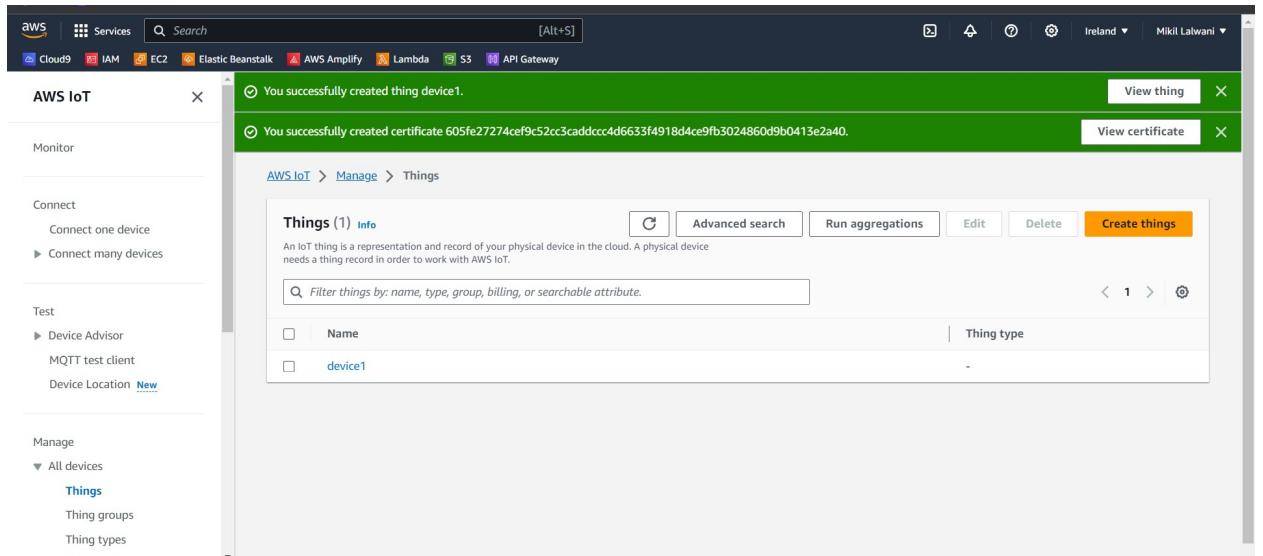
Cancel Previous Create thing

CloudShell Feedback © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

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.

AWS IoT Analytics

Introducing the new IoT Analytics console experience
We're updating the console experience for you. Try the new experiences and let us know what you think. You can turn off the new experience from the navigation menu.

AWS IoT Analytics

Get started with AWS IoT Analytics

Use this one-click quick start to create your channel, pipeline, data store, and dataset. These resources process and archive your raw IoT device data.

Resources prefix
Enter a prefix for your resources. A prefix can describe your project.

Valid characters: a-z, 0-9, and _ (underscore).

MQTT topic - optional

Wildcards (* or #) cannot be part of a topic level. Multi-level wildcards (#) must be the last character if used.

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.

AWS IoT Analytics

AWS IoT Analytics

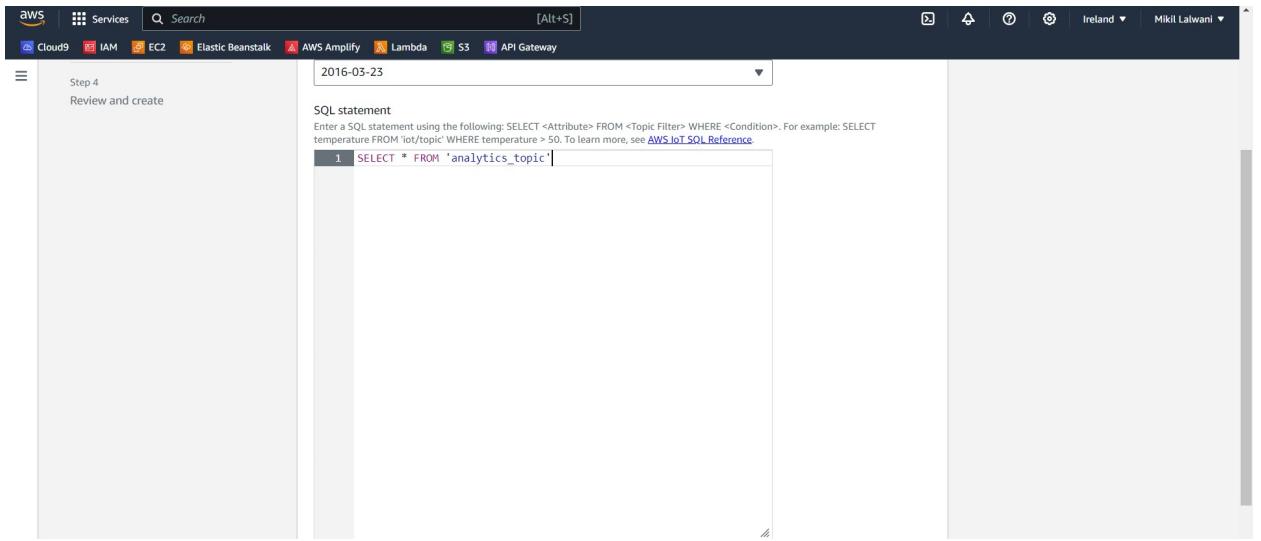
Get started with AWS IoT Analytics

Use this one-click quick start to create your channel, pipeline, data store, and dataset. These resources process and archive your raw IoT device data.

analytics_channel Succeeded
analytics_datastore Succeeded
analytics_pipeline Succeeded
analytics_dataset Succeeded
analytics_role Succeeded
analytics_rule

Prepare your data with AWS IoT Analytics

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



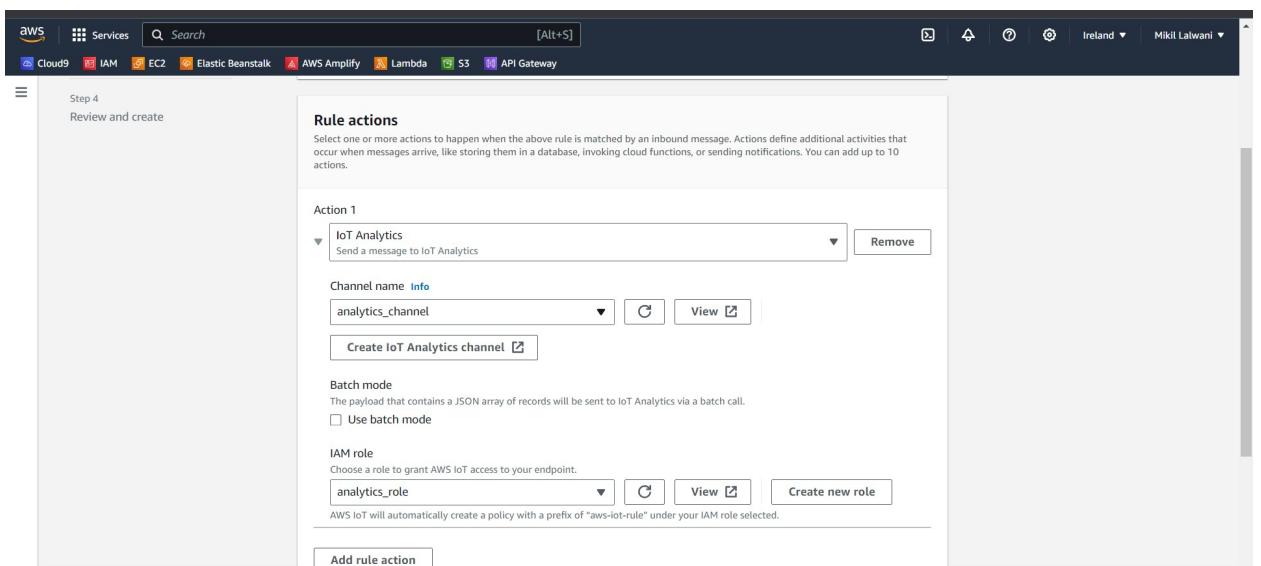
Step 4
Review and create

SQL statement

Enter a SQL statement using the following: SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. To learn more, see [AWS IoT SQL Reference](#)

```
1 | SELECT * FROM 'analytics_topic'
```

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



Step 4
Review and create

Rule actions

Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. You can add up to 10 actions.

Action 1

IoT Analytics
Send a message to IoT Analytics

Remove

Channel name **Info**

analytics_channel

Create IoT Analytics channel

Batch mode

The payload that contains a JSON array of records will be sent to IoT Analytics via a batch call.

Use batch mode

IAM role

Choose a role to grant AWS IoT access to your endpoint.

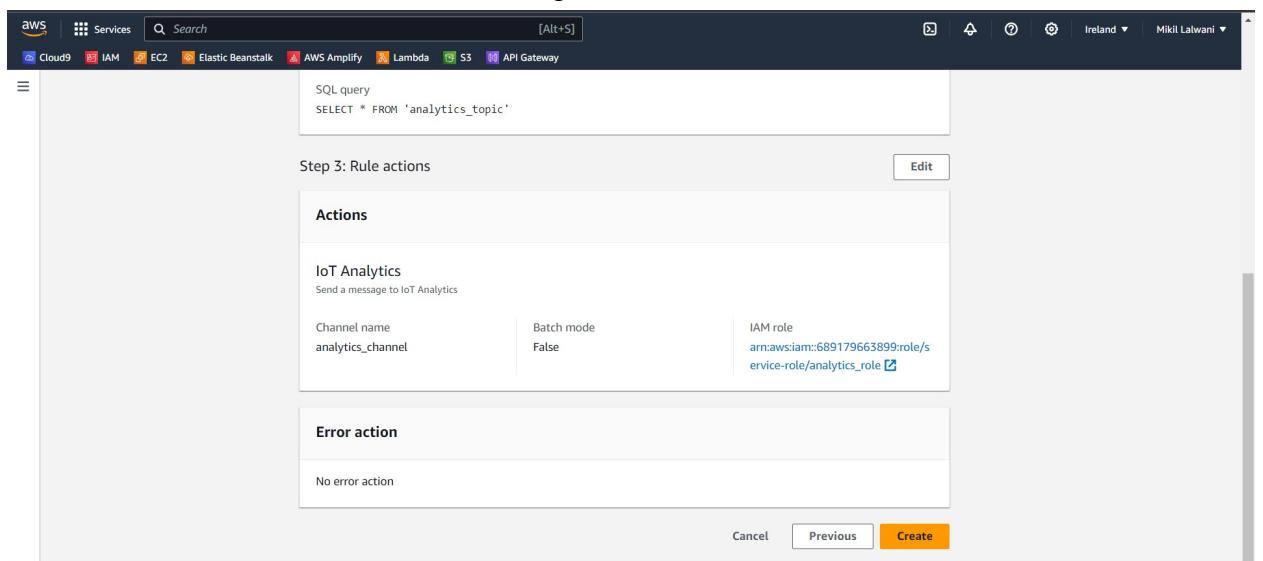
analytics_role

Create new role

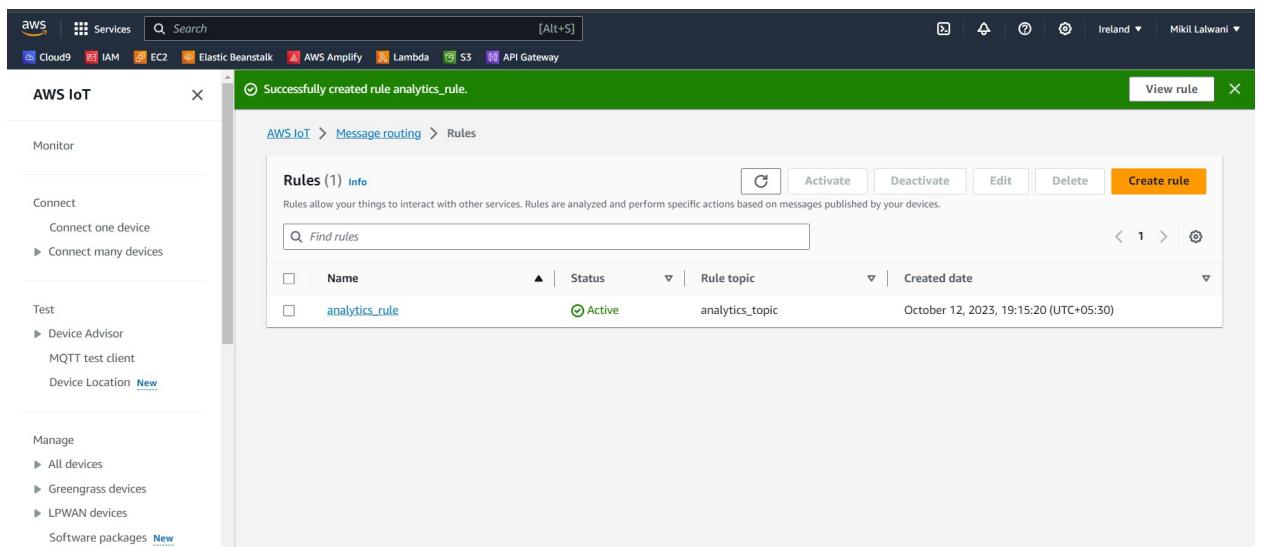
AWS IoT will automatically create a policy with a prefix of "aws-iot-rule" under your IAM role selected.

Add rule action

5. In the next step click on Create.



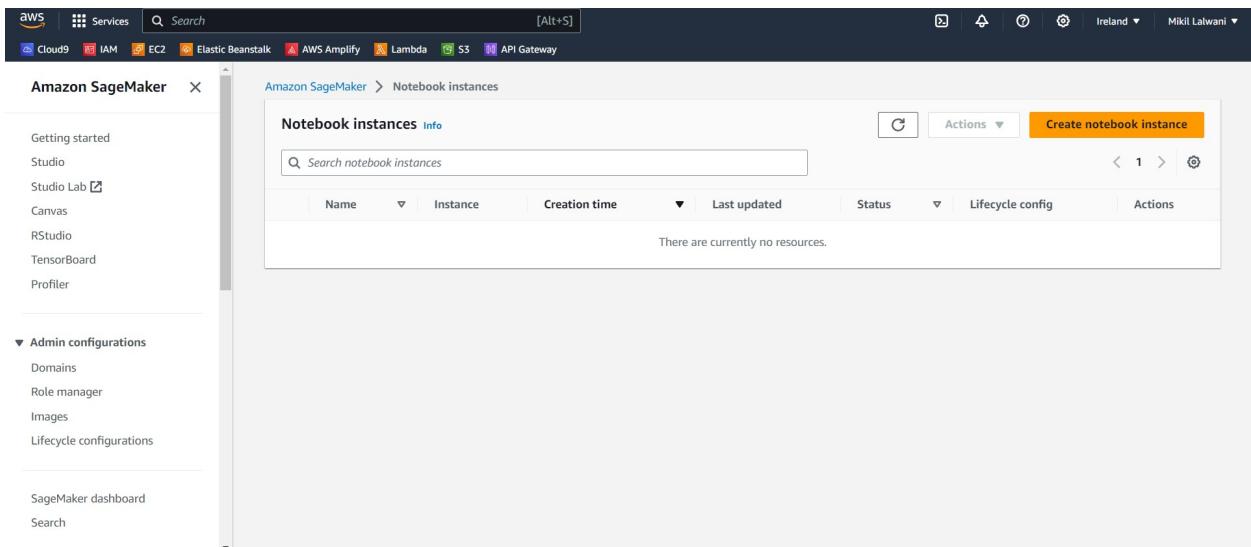
6. Our rule is finally created. The IoT Analytics resources are ready.



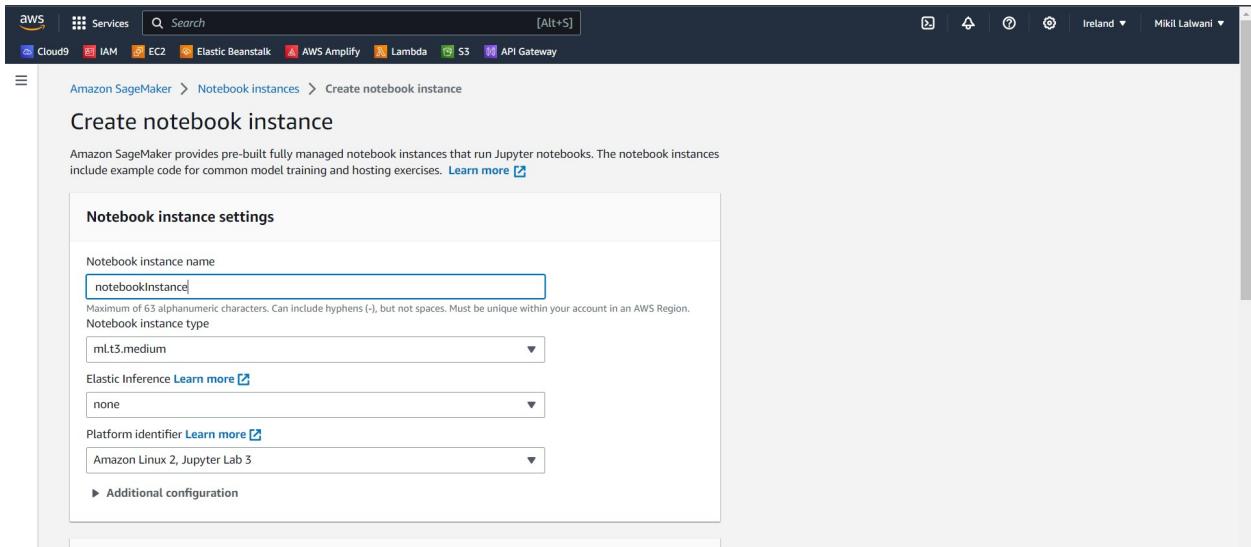
Step 4: Configure SageMaker Notebook

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.

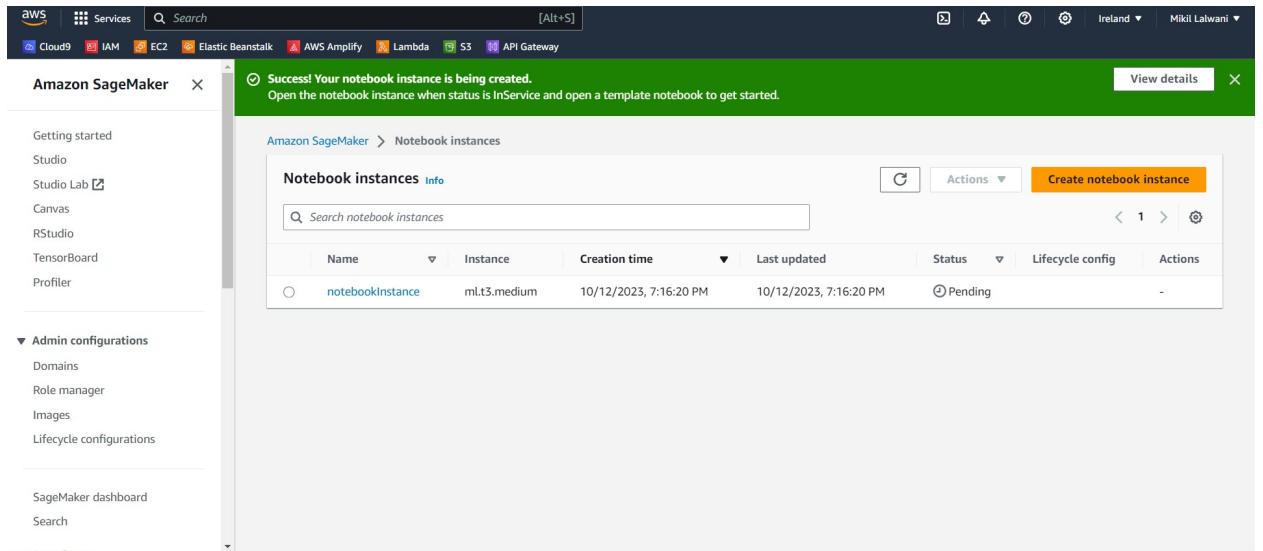
1. Goto Amazon SageMaker console. Select Notebook instances in the left and then click on the Create notebook instance button.



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.

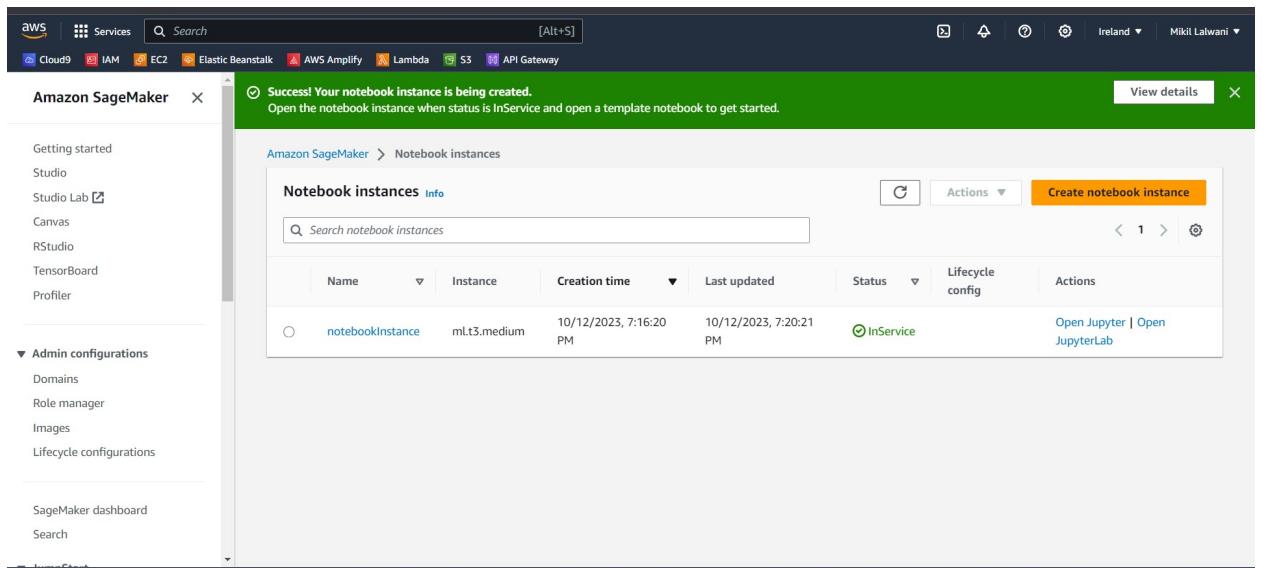


3. The notebook instance launch starts. Wait till the status changes to InService.



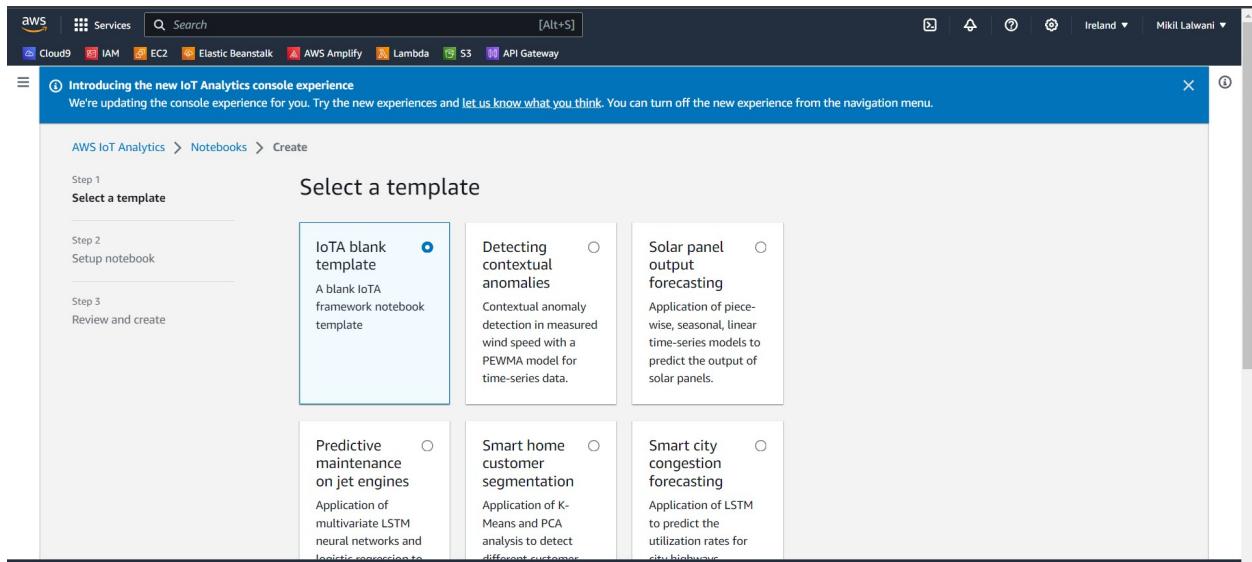
The screenshot shows the Amazon SageMaker console interface. At the top, there's a navigation bar with the AWS logo, services like Cloud9, IAM, EC2, Elastic Beanstalk, AWS Amplify, Lambda, S3, and API Gateway, and user information for 'Mikil Lalwani'. Below the navigation bar is a green success banner that reads: 'Success! Your notebook instance is being created. Open the notebook instance when status is InService and open a template notebook to get started.' To the right of the banner is a 'View details' button. The main content area is titled 'Amazon SageMaker > Notebook instances' and contains a table titled 'Notebook instances Info'. The table has columns for Name, Instance, Creation time, Last updated, Status, Lifecycle config, and Actions. There is one row in the table with the following data: Name is 'notebookinstance', Instance is 'ml.t3.medium', Creation time is '10/12/2023, 7:16:20 PM', Last updated is '10/12/2023, 7:16:20 PM', Status is 'Pending', Lifecycle config is 'Lifecycle config', and Actions is a dropdown menu. On the left side of the page, there's a sidebar with links for Getting started, Studio, Studio Lab, Canvas, RStudio, TensorBoard, Profiler, Admin configurations (Domains, Role manager, Images, Lifecycle configurations), SageMaker dashboard, and Search.

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

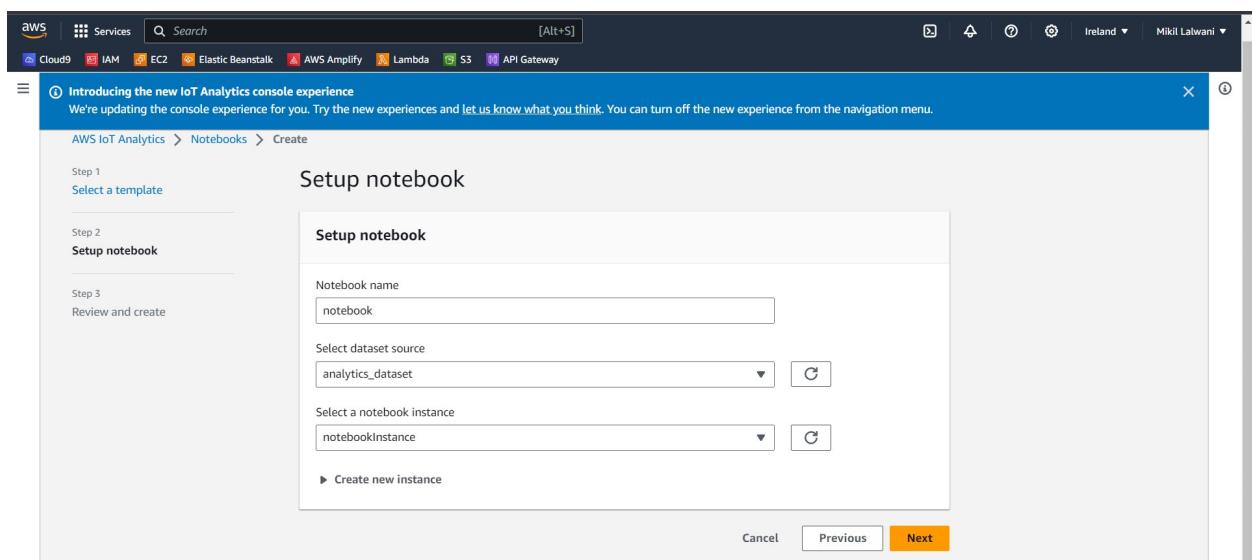


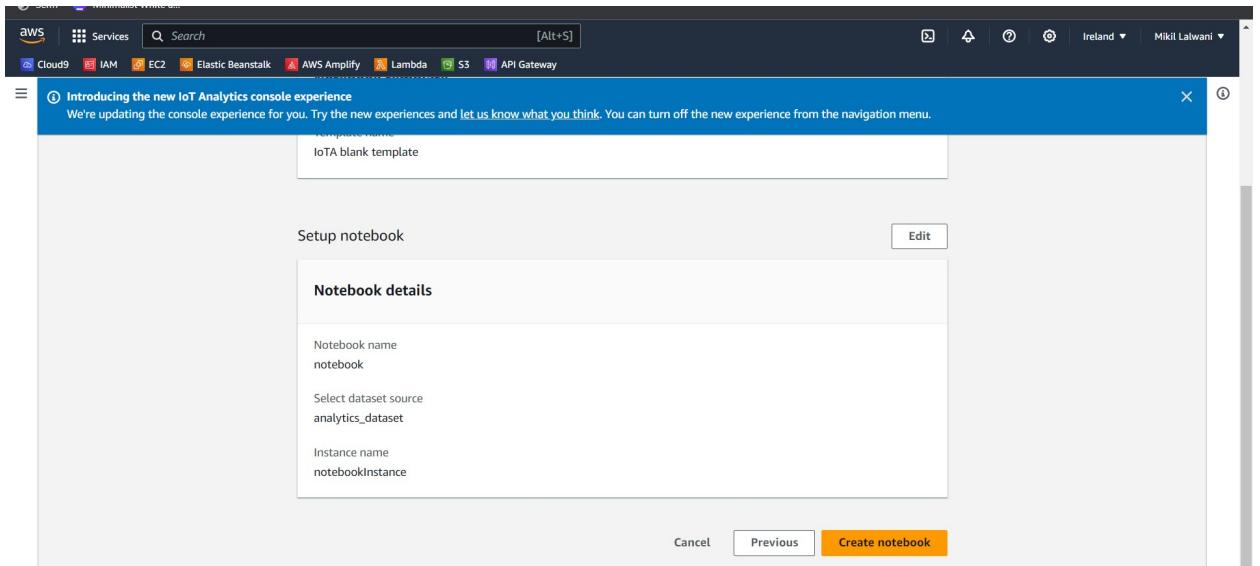
This screenshot is identical to the previous one, showing the same success banner and table for the 'notebookinstance'. However, the status column now shows 'InService' instead of 'Pending'. The 'Actions' column still contains the 'Open Jupyter | Open JupyterLab' link. The rest of the interface, including the sidebar and overall layout, remains the same.

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

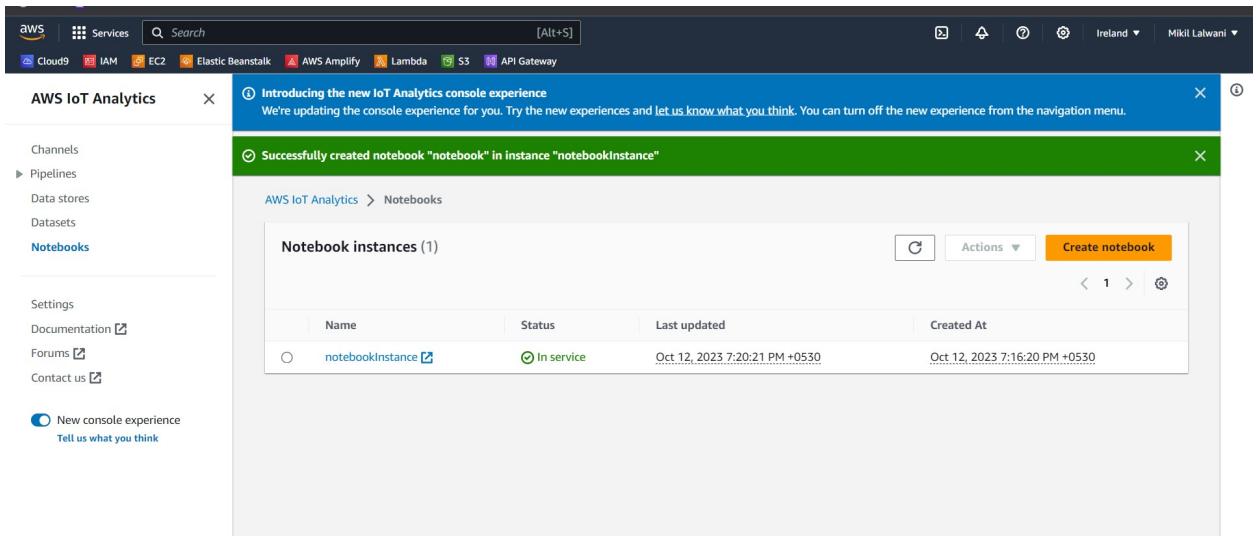


6. On the next screen, type in a notebook for the name. Select **analytics_dataset** for the data set. Select **notebook_instance** for the notebook instance. Finally, click on the Create Notebook button.



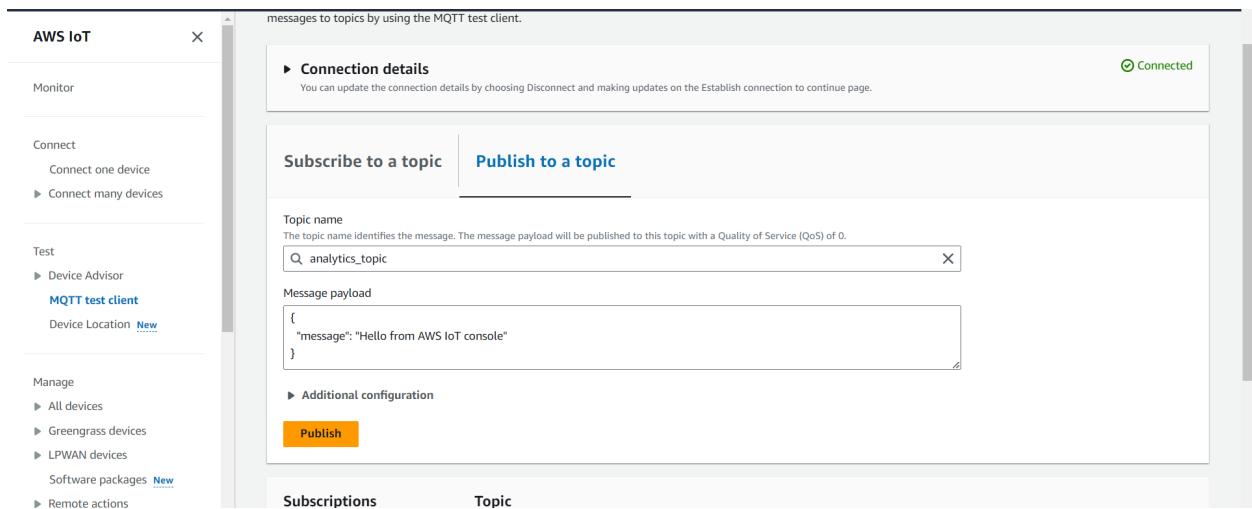


7. The notebook configuration is ready.



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.

The screenshot shows the AWS IoT Analytics management console. At the top, a green banner says "You've successfully started the query for your dataset." Below it, a "Result preview" section displays a CSV file named "613eb0cc-0869-40df-aa45-41f9fa098c91.csv". The CSV contains four columns: temperature, vibration, pressure, and __dt. The data rows are:

temperature	vibration	pressure	__dt
45	3	2	2023-10-12 00:00:00.000
19	300	19	2023-10-12 00:00:00.000
40	30	25	2023-10-12 00:00:00.000
40	30	25	2023-10-12 00:00:00.000
40	30	25	2023-10-12 00:00:00.000
40	30	25	2023-10-12 00:00:00.000

At the bottom, there is a summary table with the following data:

Date	Name	Status	Duration
Oct 12, 2023 7:31:01 PM +0530	aa793699-d079-4f8a-851e-606306a3ef33	Succeeded	1661 ms

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.

The screenshot shows the AWS SageMaker management console. On the left, a sidebar lists various services: Computer vision models, Natural language processing models, Governance, Ground Truth, Notebook (selected), Notebook instances, Git repositories, Processing, Training, Inference, Edge Manager, Augmented AI, and AWS Marketplace. Under the Notebook section, "Notebook instances" is selected. In the main area, the "notebookInstance" page is shown. It displays "Notebook instance settings" for a notebook instance named "notebookInstance". The settings include:

Name	Status	Notebook instance type	Platform identifier
notebookInstance	InService	m1t3.medium	Amazon Linux 2, Jupyter Lab 3 (notebook-al2-v2)

Other details shown include ARN, Creation time, Last updated, and Volume Size.

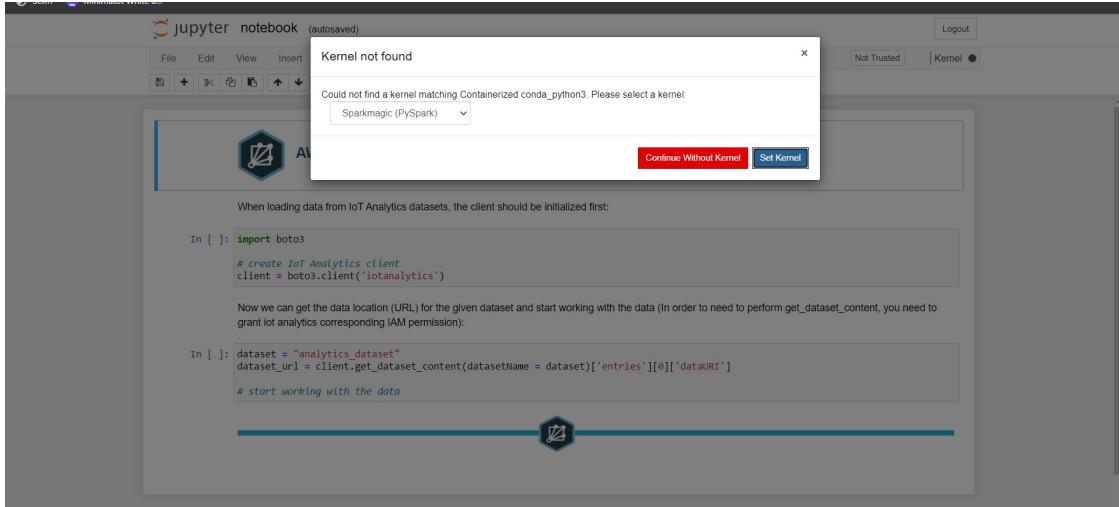
2. Select the IoTAnalytics.



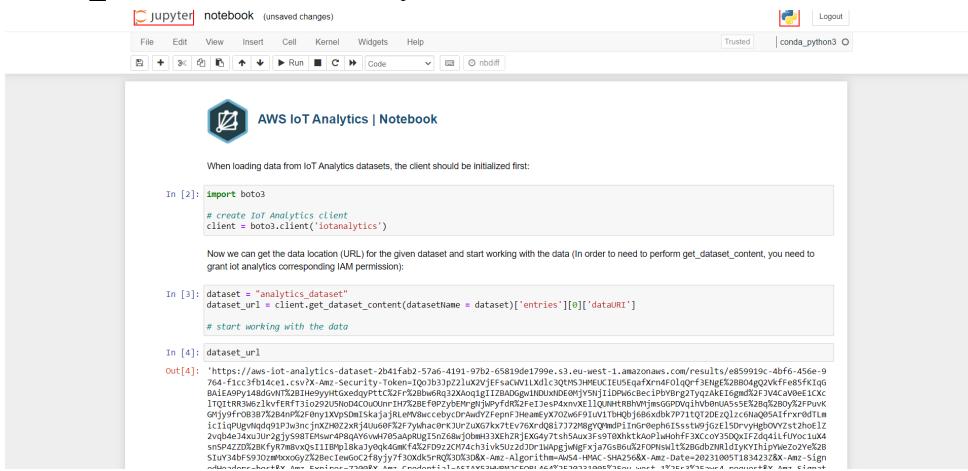
3. Now select the python notebook.



4. It will open Jupyter 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
```

jupyter notebook (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

In [3]: `import pandas as pd`

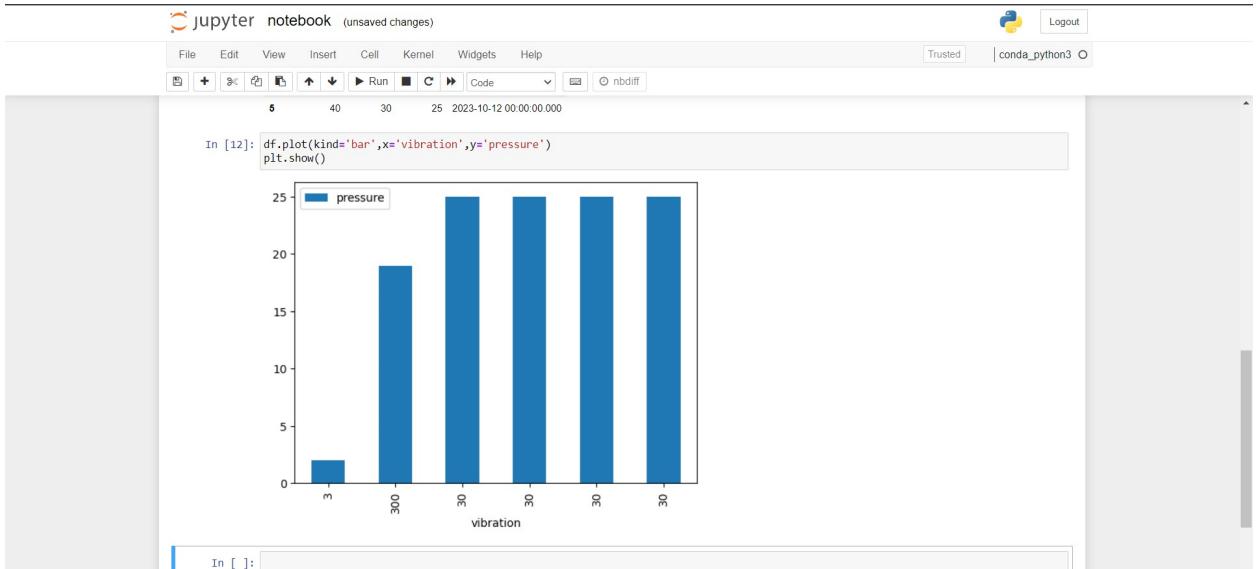
In [4]: `import matplotlib.pyplot as plt`

Out[4]:

	temperature	vibration	pressure	_dt
0	45	3	2	2023-10-12 00:00:00.000
1	19	300	19	2023-10-12 00:00:00.000
2	40	30	25	2023-10-12 00:00:00.000
3	40	30	25	2023-10-12 00:00:00.000
4	40	30	25	2023-10-12 00:00:00.000
5	40	30	25	2023-10-12 00:00:00.000

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.