GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL of ELECTRICAL and COMPUTER ENGINEERING

ECE 4150-A Spring 2021
Lab: Air Quality Monitoring Application using AWS IoT, DynamoDB, Lambda, CloudWatch and EC2

References:

[1] A. Bahga, V. Madisetti, "Cloud Computing Solutions Architect: A Hands-On Approach", ISBN: 978-0996025591

[2] https://aws.amazon.com/documentation/

[3] Guidelines for the Reporting of Daily Air Quality - the Air Quality Index (AQI), https://archive.epa.gov/ttn/ozone/web/pdf/rg701.pdf

Due Date:

The lab report will be **due on March 17, 2021 at 11:59 PM.**

In this lab, we will create an Air Quality Monitoring application.

The application uses the following:

1. AWS IoT thing to receive data from IoT devices and send to DynamoDB
2. DynamoDB tables to store raw pollutants data and computed AQI data
3. Lambda function to compute AQI from raw pollutants data.
4. Cloudwatch time-based event to trigger Lambda function to compute AQI every minute or hour
5. A web interface implemented in Django to visualize raw pollutants data and computed AQI

The **Air Quality Monitoring System** uses air quality monitoring nodes (IoT devices) that measure concentrations of various pollutants such as PM2.5, PM10, CO, SO2, O3, and NO2. Each node is known by an identifier called **stationID**. The IoT devices publish the raw data on pollutant concentrations to AWS IoT. Within AWS IoT, a rule to send the data to a DynamoDB table is created. The raw information is stored in a DynamoDB table. A Lambda function is used to compute average concentrations of the pollutants over 8 or 24 hours and then compute the Air Quality Index from average pollutant concentrations using the formula as shown below:

$$I_p = \frac{I_{Hi} - I_{Lo}}{BP_{HI} - BP_{Lo}}(C_p - BP_{Lo}) + I_{Lo}.$$

Where I_p = the index for pollutant p

C_p = the rounded concentration of pollutant p
BP_{Hi} = the breakpoint that is greater than or equal to C_p

BP_{Lo} = the breakpoint that is less than or equal to C_p

BP_{Hi} = the breakpoint that is greater than or equal to C_p

I_{Hi} = the AQI value corresponding to BP_{Hi}

I_{Lo} = the AQI value corresponding to BP_{Lo}

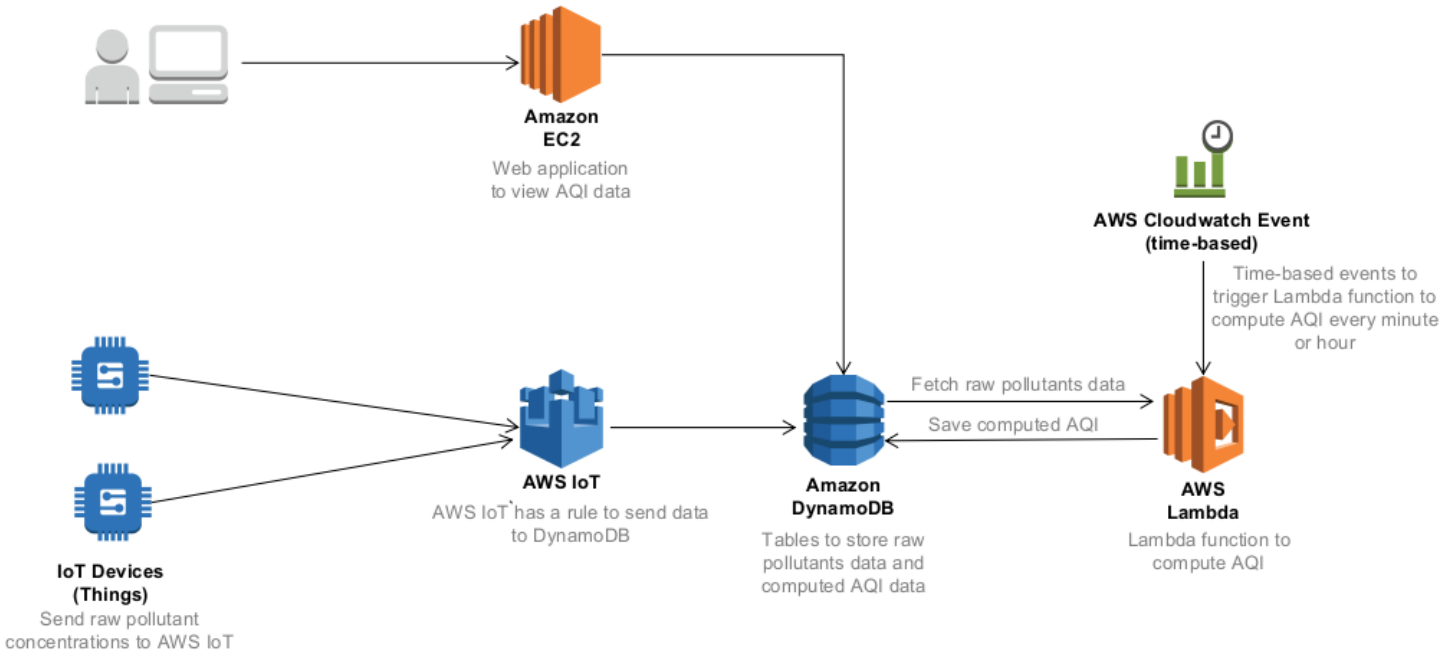


Fig.1 Architecture diagram of the Air Quality Monitoring application showing AWS services used

The breakpoint values used for AQI computation are shown in the table below.

This Breakpoint...						...equal this AQI		…and this category
o_3 (ppm) 8-hour	O_3 (ppm) 1-hour1	PM_{10} ($\mu g/m^3$)	$PM_{2.5}$ ($\mu g/m^3$)	CO (ppm)	SO_2 (ppm)	NO_2 (ppm)	AQI	
0.000 - 0.064	-	0 - 54	0.0 - 15.4	0.0 - 4.4	0.000 - 0.034	([2])	0 - 50	Good
0.065 - 0.084	-	55 - 154	15.5 -40.4	4.5 - 9.4	0.035 - 0.144	([2])	51 - 100	Moderate
0.085 - 0.104	0.125 - 0.164	155 - 254	40.5 - 65.4	9.5 - 12.4	0.145 - 0.224	([2])	101 - 150	Unhealthy for Sensitive Groups
0.105 - 0.124	0.165 - 0.204	255 - 354	65.5 - 150.4	12.5 - 15.4	0.225 - 0.304	([2])	151 - 200	Unhealthy
0.125 - 0.374 ([4])	0.205 - 0.404	355 - 424	150.5 - 250.4	15.5 - 30.4	0.305 - 0.604	0.65 - 1.24	201 - 300	Very unhealthy
([3])	0.405 - 0.504	425 - 504	250.5 - 350.4	30.5 - 40.4	0.605 - 0.804	1.25 - 1.64	301 - 400	Hazardous
([3])	0.505 - 0.604	505 - 604	350.5 - 500.4	40.5 - 50.4	0.805 - 1.004	1.65 - 2.04	401 - 500	Hazardous

Table.1 Breakpoint values used for AQI computation

The AQI is determined by the pollutant with the highest index. For example, if the PM2.5 AQI is 135, the PM10 AQI is 40, SO2 is 30, CO is 50, and all other pollutants are less than 135, then the AQI is 135, determined only by the concentration of PM2.5.

Follow the steps below to setup the air quality monitoring application in your AWS account.

1. Create DynamoDB Tables (5 points)

- Create two DynamoDB tables named **AirQualityData** and **AirQualityDataOutput** to store raw pollutant concentration data and computed AQI values.
- For these tables, use **stationID**, type *String*, as partition key, and **timestamp**, type *Number*, as sort key.

2. Create IAM Role (5 points)

- Create a new IAM role for a **Lambda Function** named **lambda_dynamodb_access**. Attach the policy that allows you full access to your DynamoDB.

3. Create IAM User (5 points)

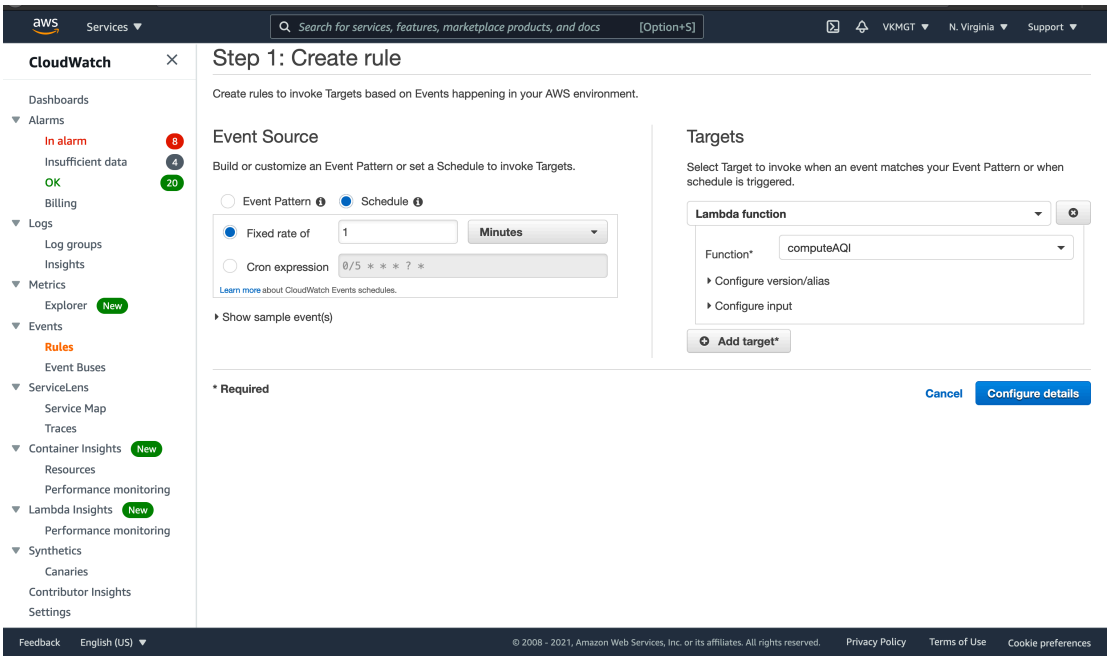
- Create a new IAM User for a for the Django dashboard in the EC2 instance named **Django_user**. Attach the policy that allows you full access to your DynamoDB. Save the **Access key ID** and **Secret access key**.

4. Create Lambda function (5 points)

- Create a new **Lambda Function** for computing AQI from raw pollutant concentrations and saving the AQI values to DynamoDB.
- Create a function for **Python 2.7** and call it **computeAQI** using the code provided as lambda_function.py**,** attach the **role** created in the previous step.

5. Create CloudWatch Event (5 points)

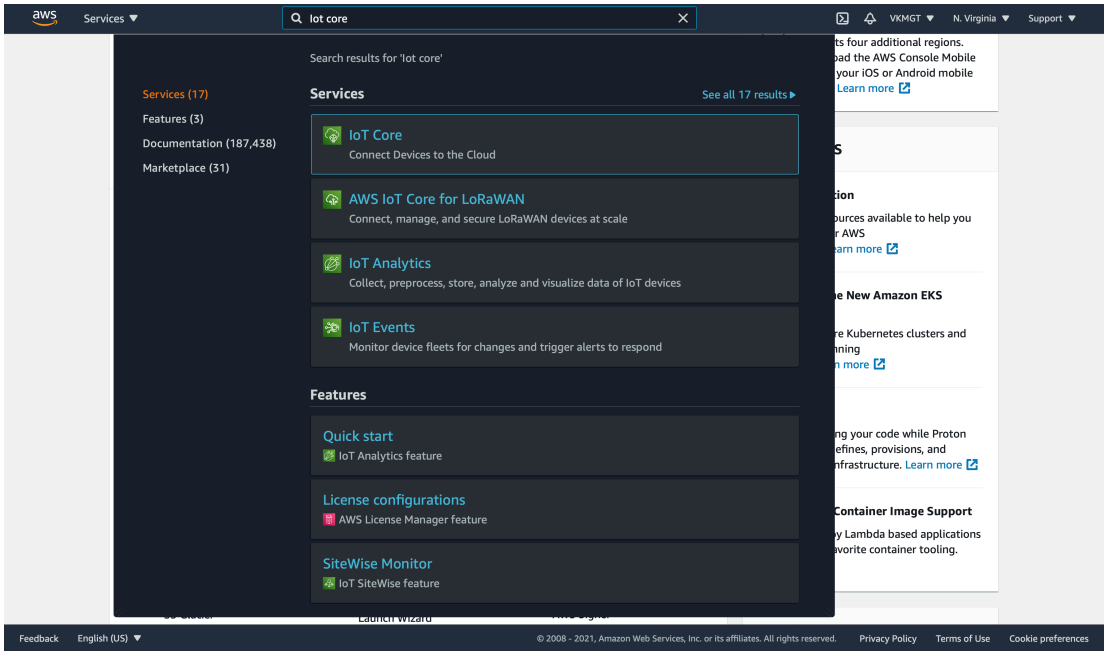
- From **CloudWatch** console, go to **Events**, and create a new **event rule** to trigger the Lambda function created in the previous step in a **schedule** pattern at a rate of 1 minute. Assign the lambda function to the event and name it **computeAQI_rule.** An example is shown below.

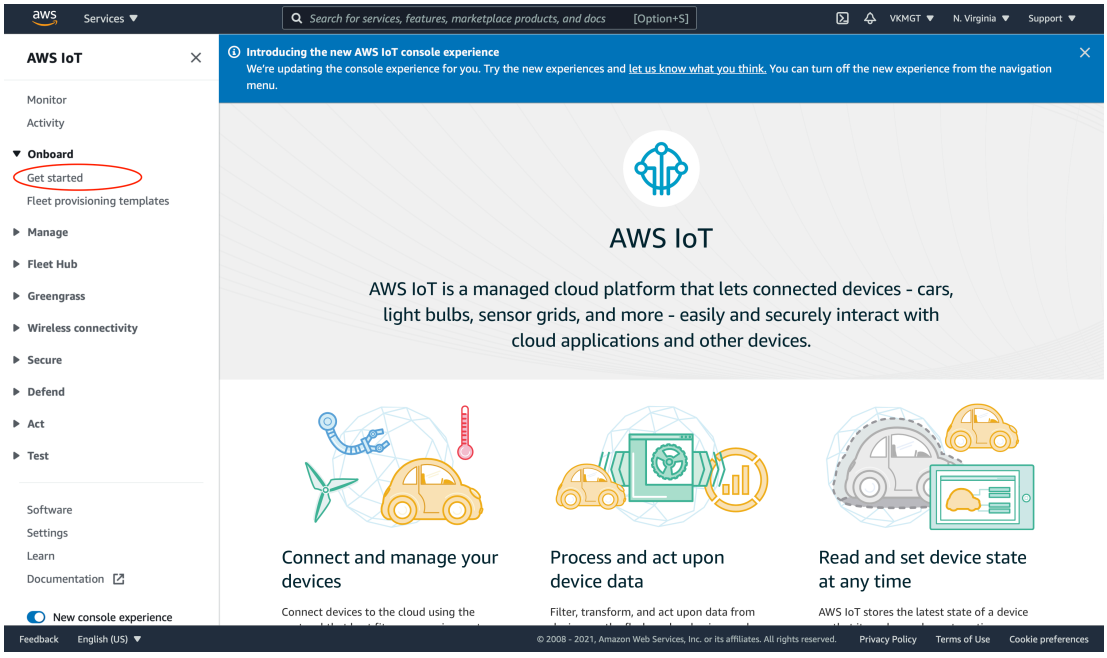


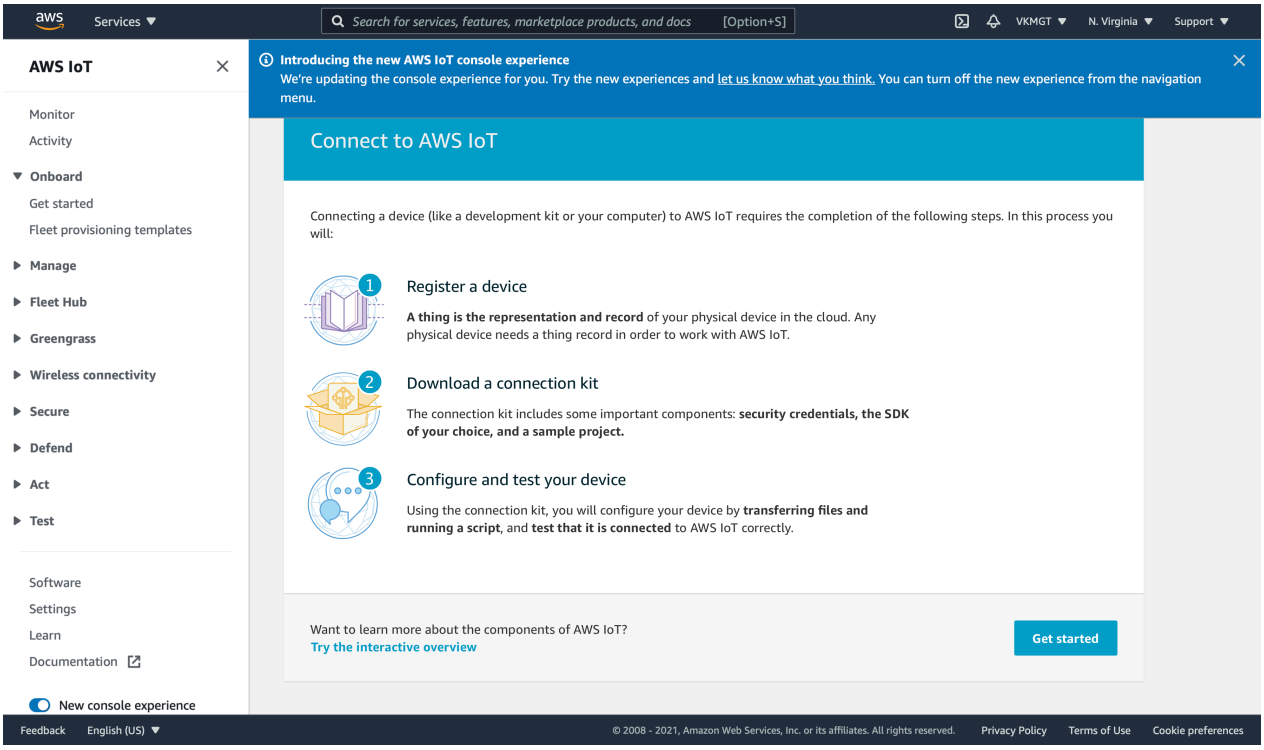
- Navigate back your lambda function, you should be able to see a CloudWatch event been added.

6. Create IoT Core (10 points)

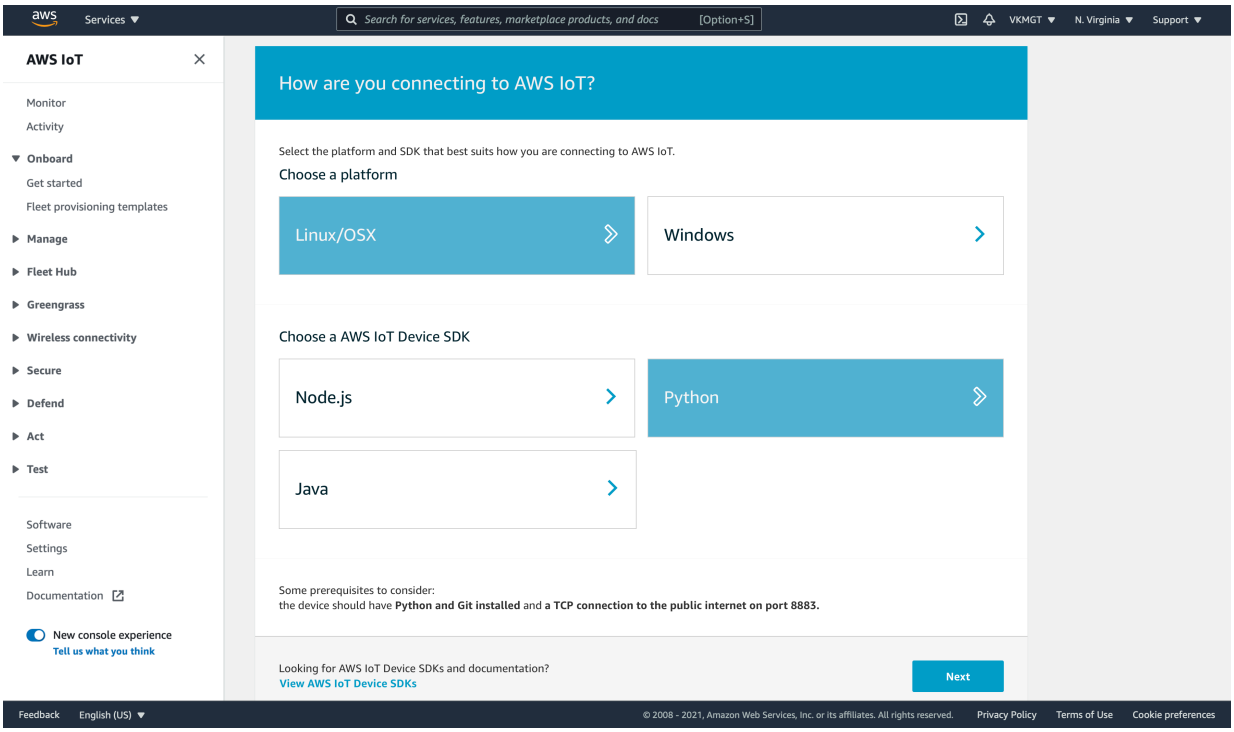
- Navigate to AWS IoT core console and provision a thing using the Connect to AWS IoT wizard



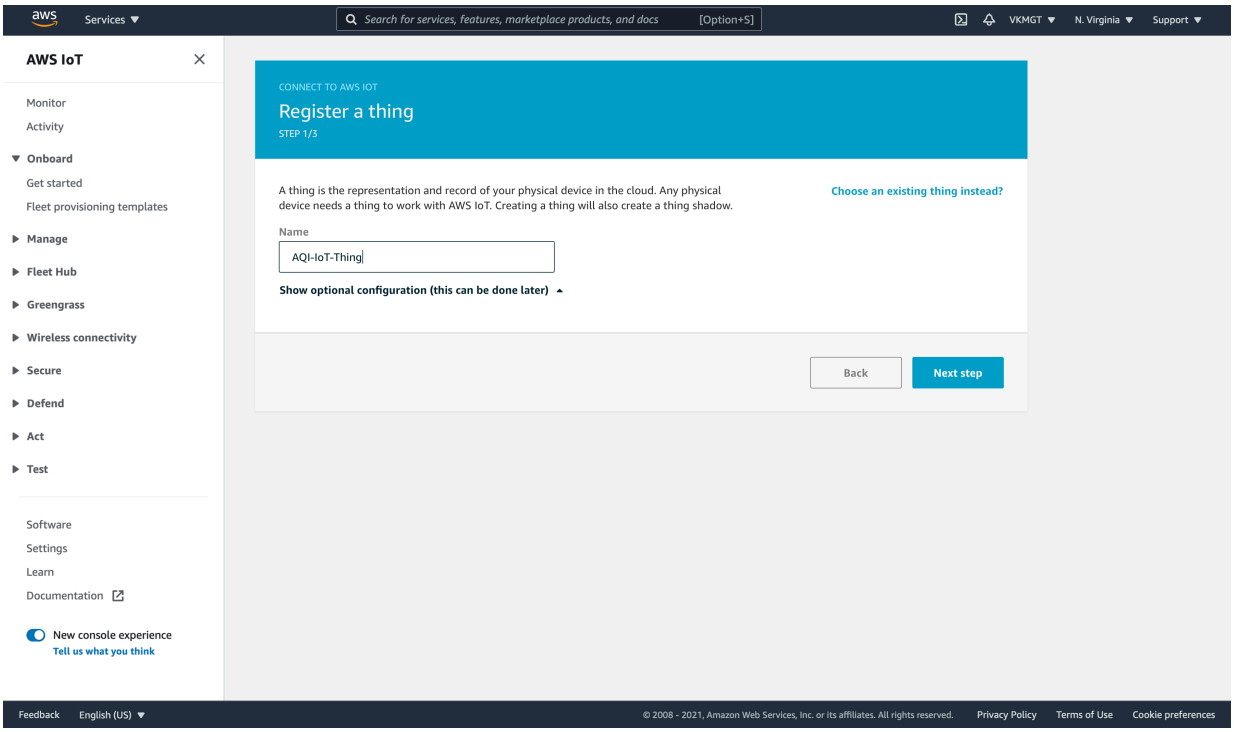




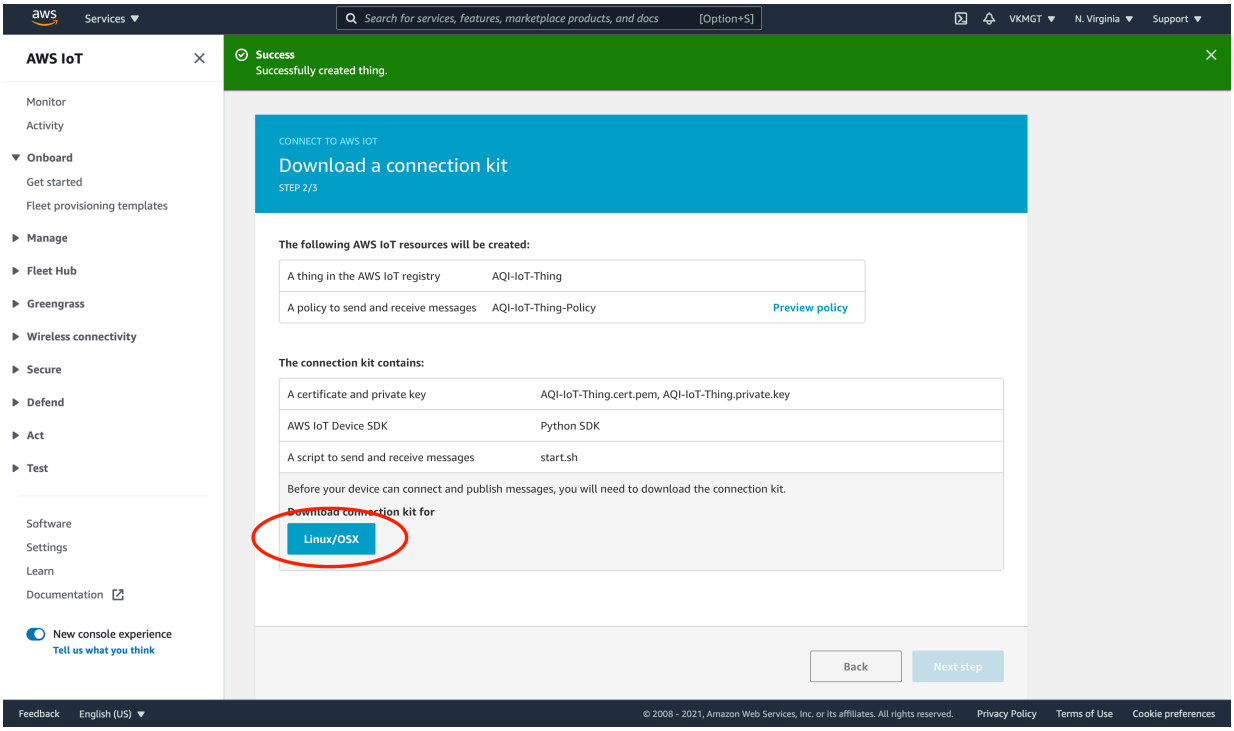
- Select Linux/OSX if you are using a MacOS or Linux based computer, otherwise select Windows.



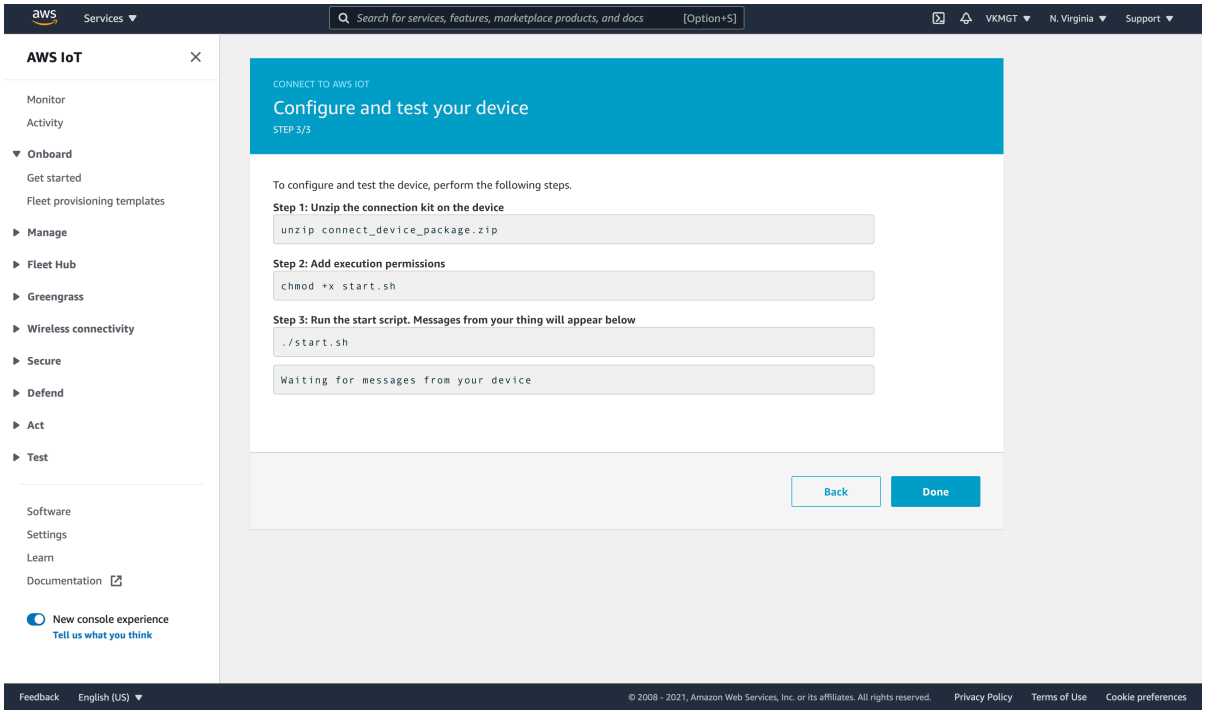
- Name the IoT Thing **AQI-IoT-Thing**



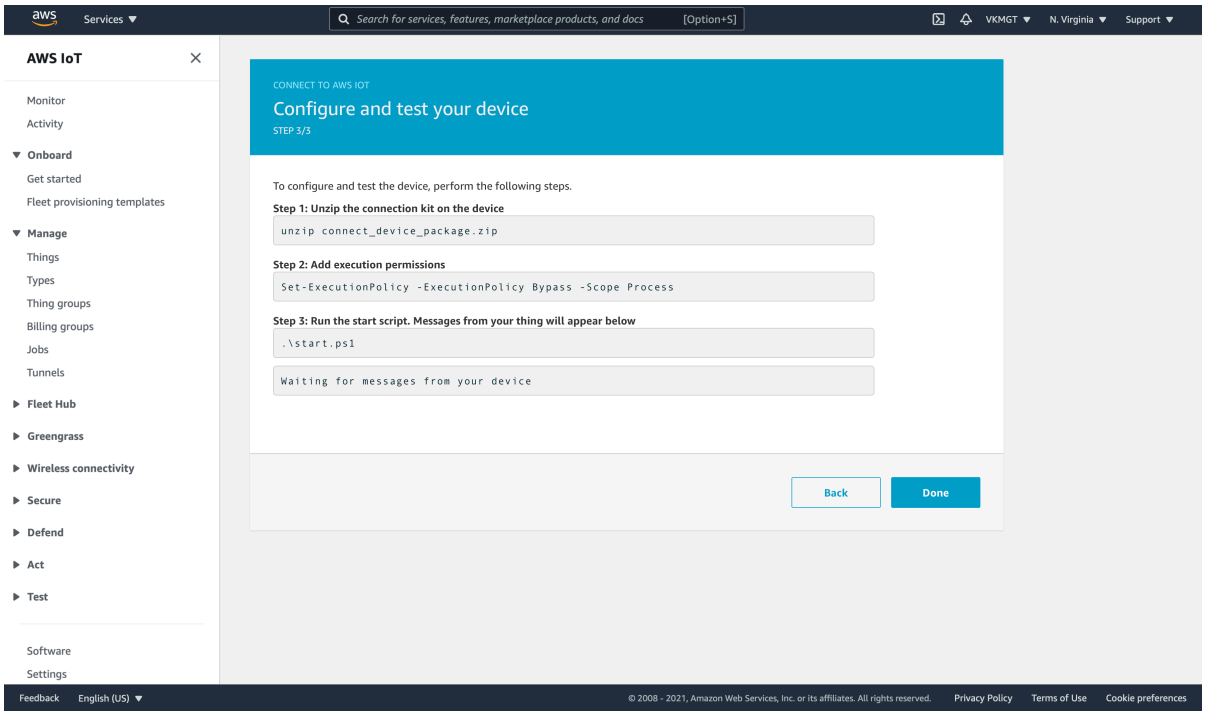
- Download the connection kit generated and run the start script to test the device.

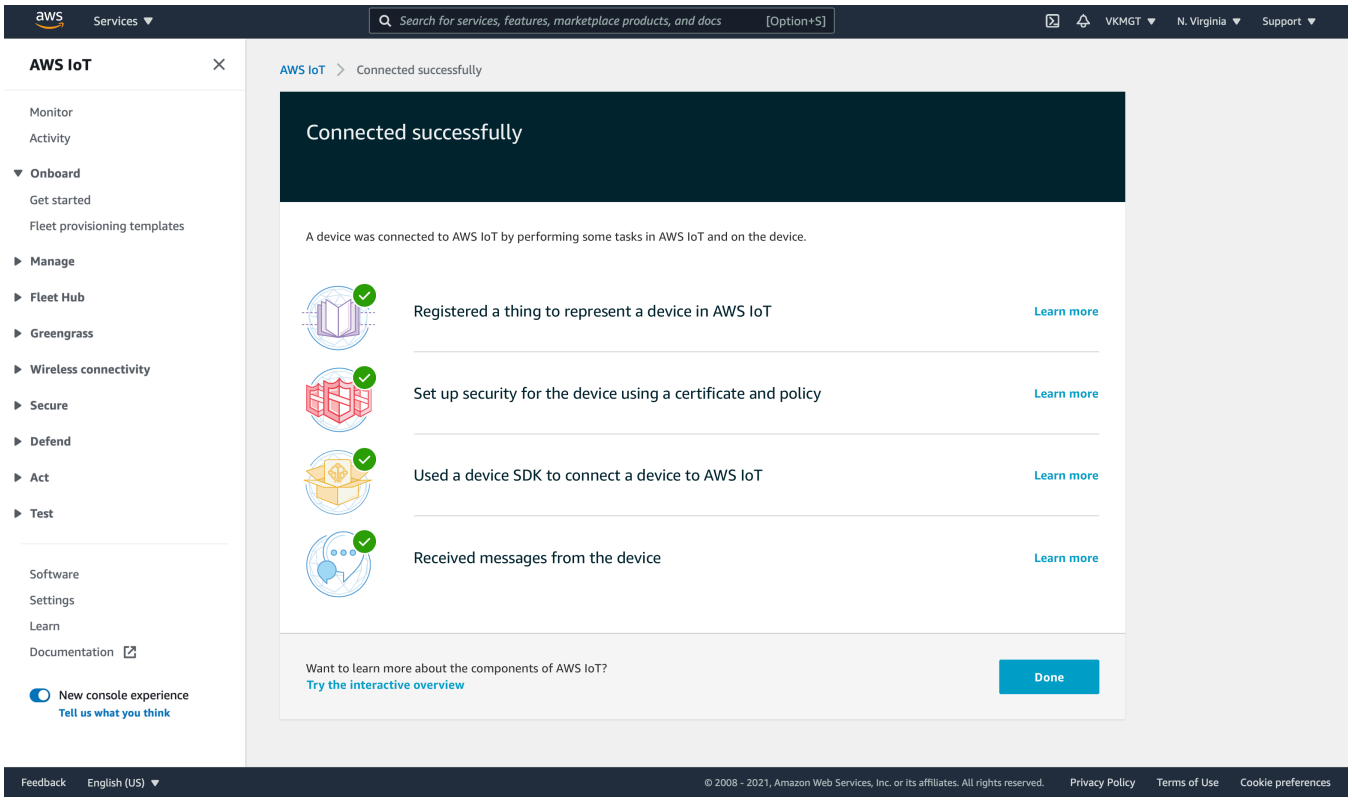


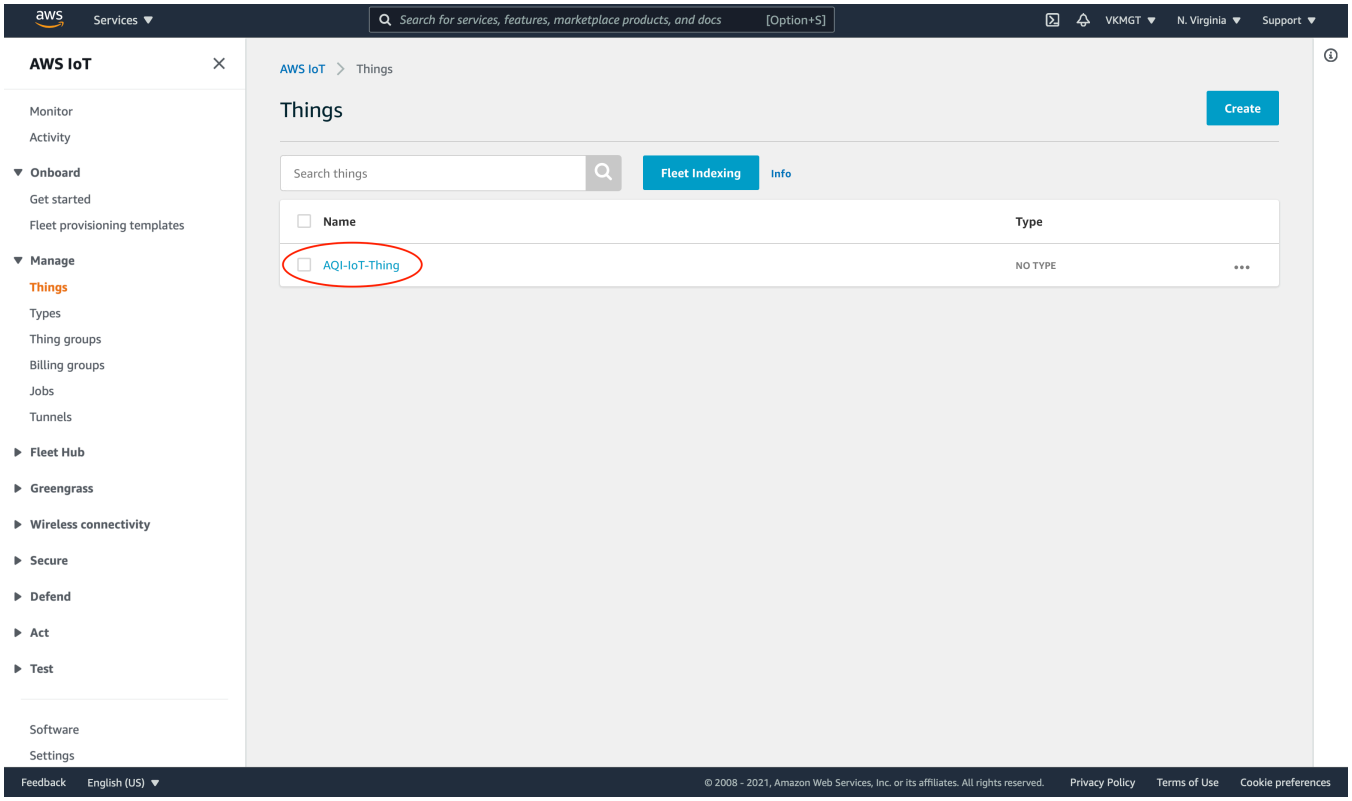
- For MacOS or Linux users

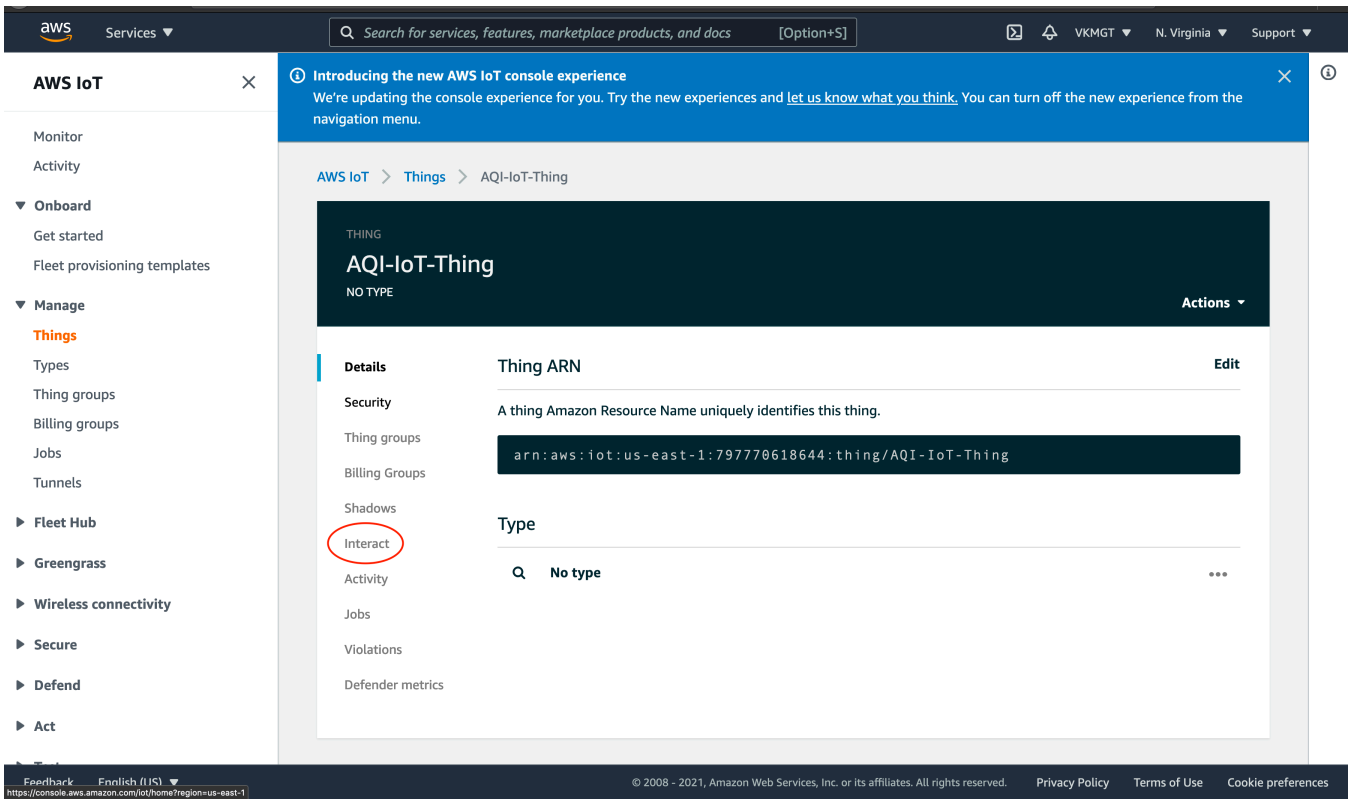


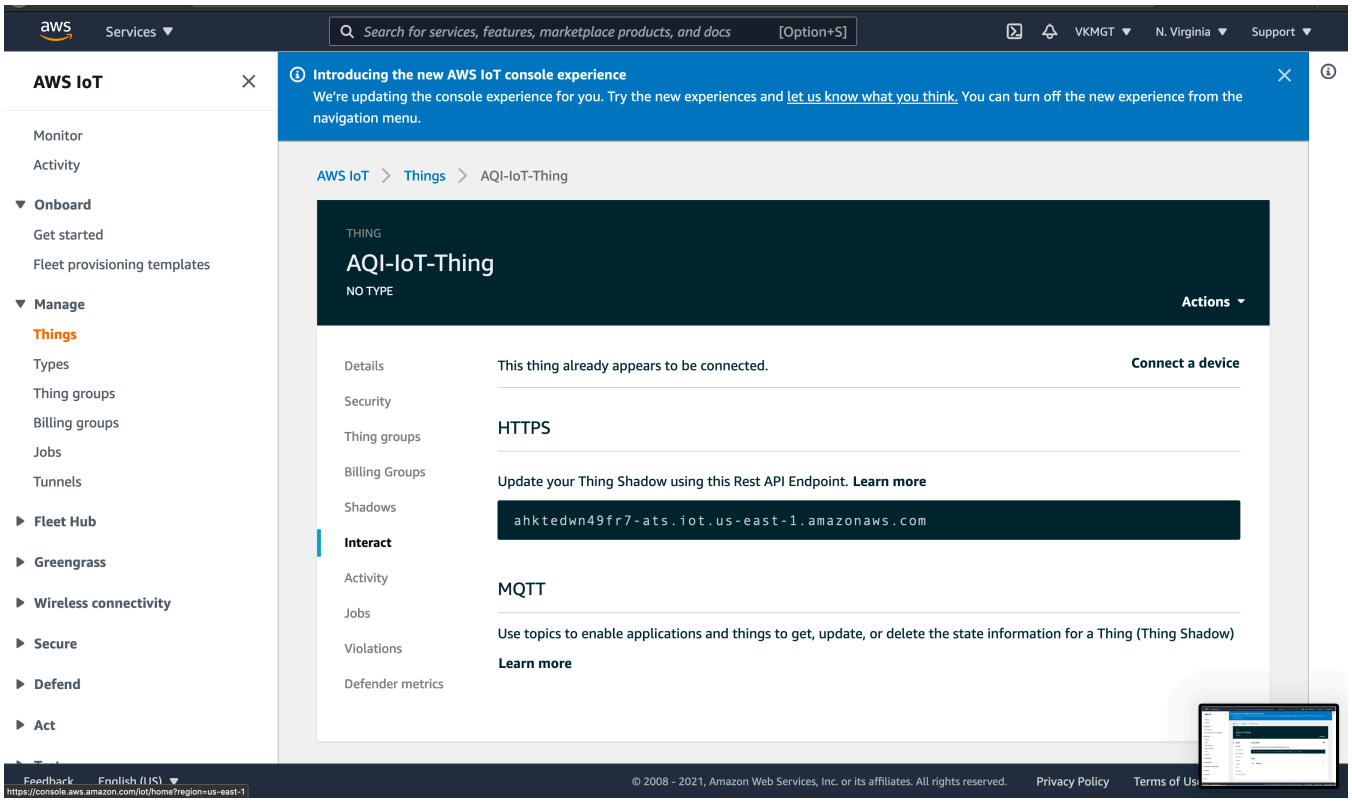
- For Windows users




Services
Search for services, features, marketplace products, and docs [Option+S]
VKMGT
N. Virginia
Support
AWS IoT
Monitor
Activity
Onboard
Get started
Fleet provisioning templates
Manage
Fleet Hub
Greengrass
Wireless connectivity
Secure
Defend
Act
Test
Software
Settings
Learn
Documentation
New console experience
Tell us what you think
AWS IoT > Connected successfully
Connected successfully
A device was connected to AWS IoT by performing some tasks in AWS IoT and on the device.
Registered a thing to represent a device in AWS IoT
Learn more
Set up security for the device using a certificate and policy
Learn more
Used a device SDK to connect a device to AWS IoT
Learn more
Received messages from the device
Learn more
Want to learn more about the components of AWS IoT?
Try the interactive overview
Done
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

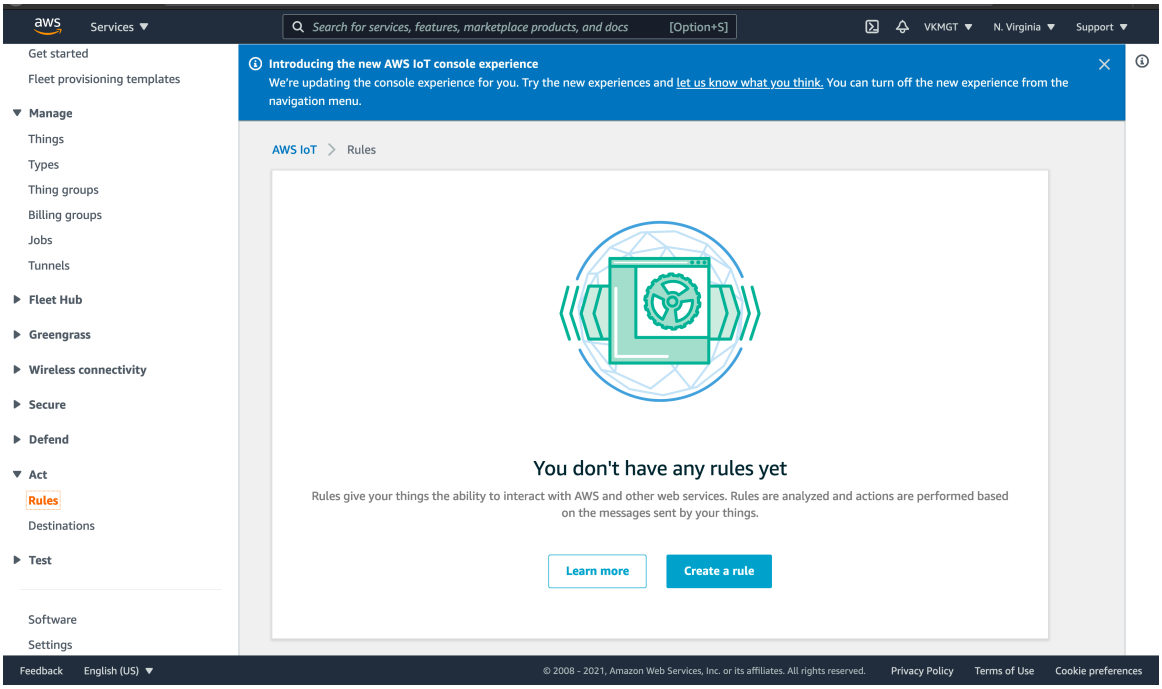

Services
Search for services, features, marketplace products, and docs [Option+S]
VKMGT
N. Virginia
Support
AWS IoT
Monitor
Activity
Onboard
Get started
Fleet provisioning templates
Manage
Things
Types
Thing groups
Billing groups
Jobs
Tunnels
Fleet Hub
Greengrass
Wireless connectivity
Secure
Defend
Act
Test
Software
Settings
AWS IoT > Things
Things
Create
Search things
Fleet Indexing
Info
Name
Type
AQI-IoT-Thing
NO TYPE
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences


aws
Services
Search for services, features, marketplace products, and docs [Option+S]
VKMGT
N. Virginia
Support
AWS IoT
Introducing the new AWS IoT 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.
Monitor
Activity
Onboard
Get started
Fleet provisioning templates
Manage
Things
Types
Thing groups
Billing groups
Jobs
Tunnels
Fleet Hub
Greengrass
Wireless connectivity
Secure
Defend
Act
AWS IoT > Things > AQI-IoT-Thing
THING
AQI-IoT-Thing
NO TYPE
Actions
Details
Security
Thing groups
Billing Groups
Shadows
Interact
Activity
Jobs
Violations
Defender metrics
Thing ARN
Edit
A thing Amazon Resource Name uniquely identifies this thing.
arn:aws:iot:us-east-1:797770618644:thing/AQI-IoT-Thing
Type
No type
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences
https://console.aws.amazon.com/iot/home?region=us-east-1

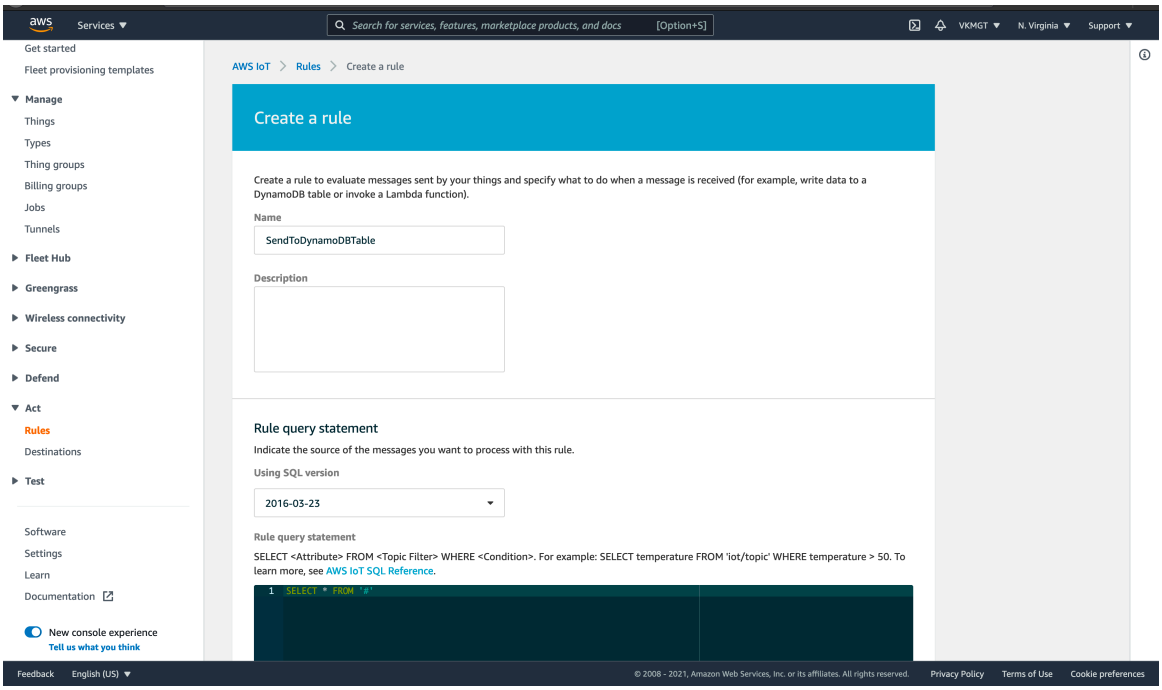

aws
Services
Search for services, features, marketplace products, and docs [Option+S]
VKMGT
N. Virginia
Support
AWS IoT
Introducing the new AWS IoT console experience
We're updating the console experience for you. Try the new experiences and let us know what you think. You can turn off the new experience from the navigation menu.
Monitor
Activity
Onboard
Get started
Fleet provisioning templates
Manage
Things
Types
Thing groups
Billing groups
Jobs
Tunnels
Fleet Hub
Greengrass
Wireless connectivity
Secure
Defend
Act
AWS IoT > Things > AQI-IoT-Thing
THING
AQI-IoT-Thing
NO TYPE
Actions
Details
Security
Thing groups
Billing Groups
Shadows
Interact
Activity
Jobs
Violations
Defender metrics
This thing already appears to be connected.
Connect a device
HTTPS
Update your Thing Shadow using this Rest API Endpoint. Learn more
ahktedwn49fr7-ats.iot.us-east-1.amazonaws.com
MQTT
Use topics to enable applications and things to get, update, or delete the state information for a Thing (Thing Shadow)
Learn more
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
https://console.aws.amazon.com/iot/home?region=us-east-1

7. Create IoT Rule (10 points)

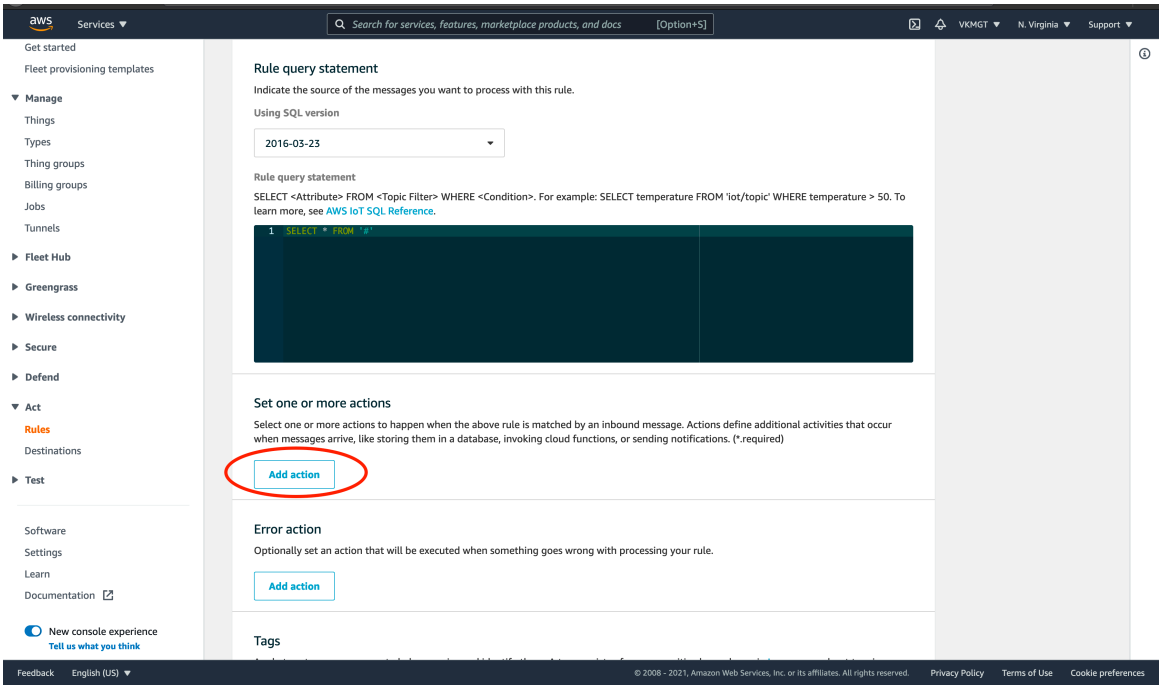
- Create an IoT rule to send data to DynamoDB. In the AWS IoT Core console, select the **Act** dropdown button on the left panel, and then click **Rules**.

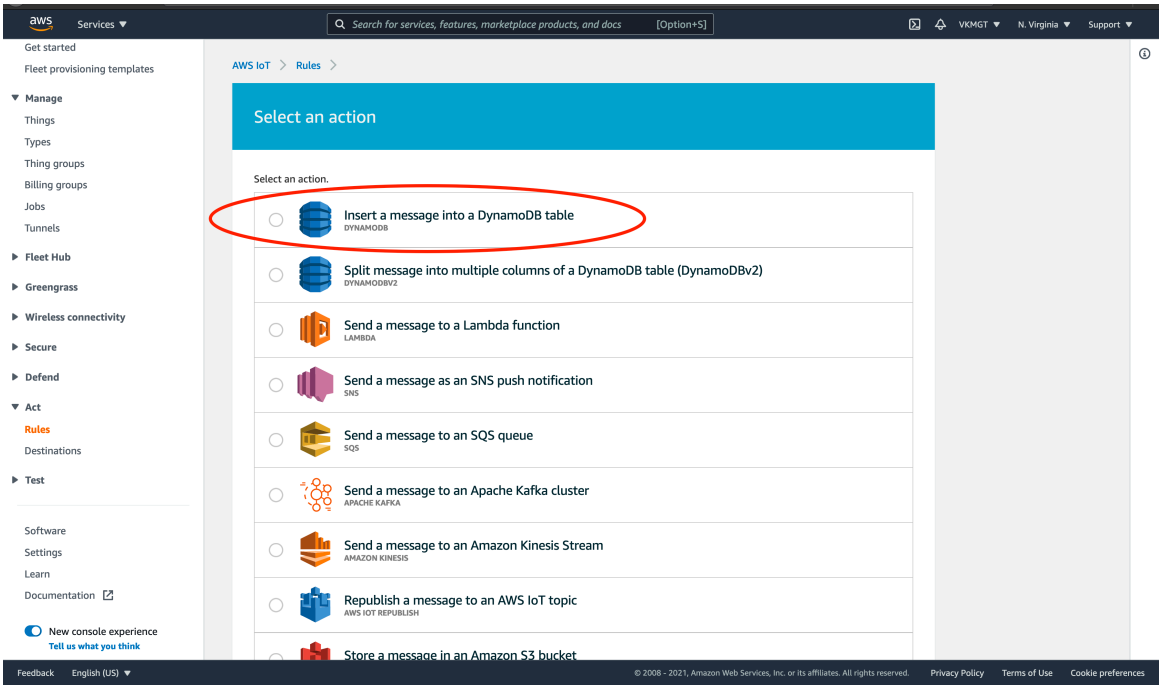


- In rule query statement enter: **SELECT * FROM '#'** as shown below.

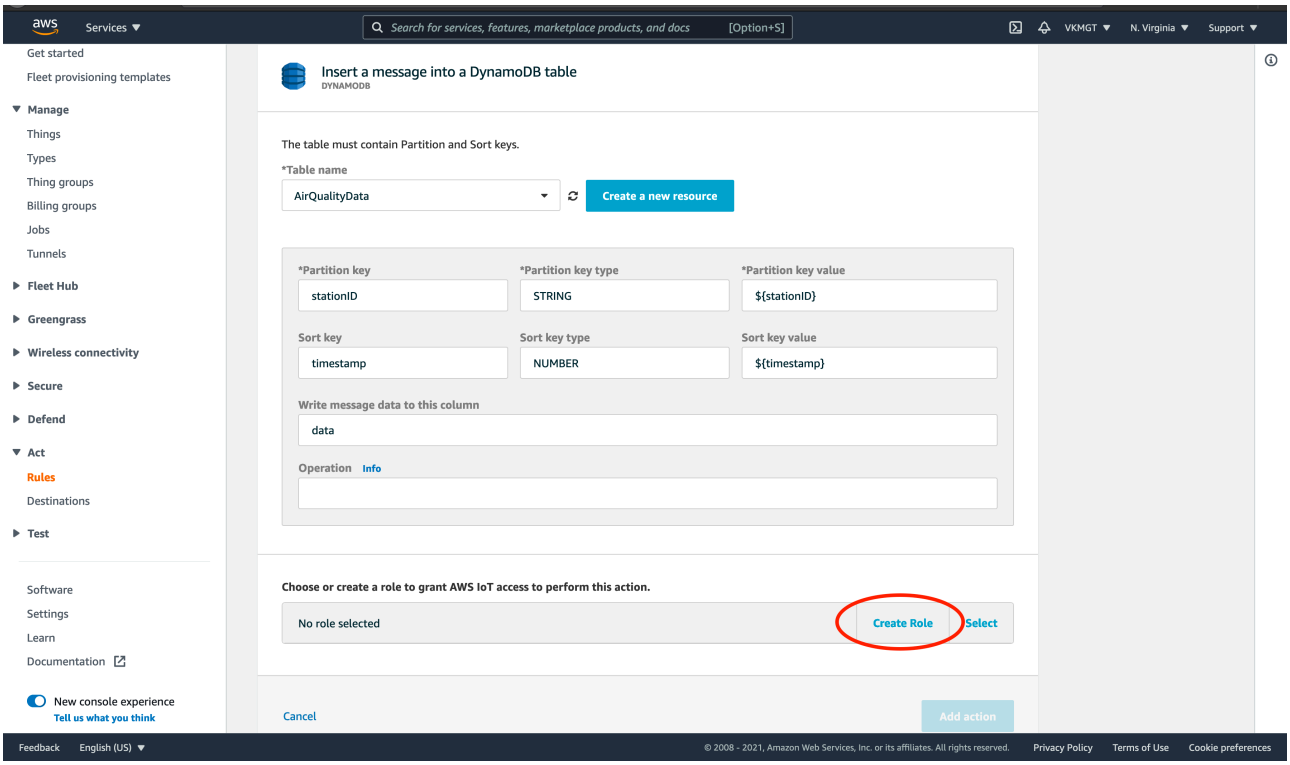


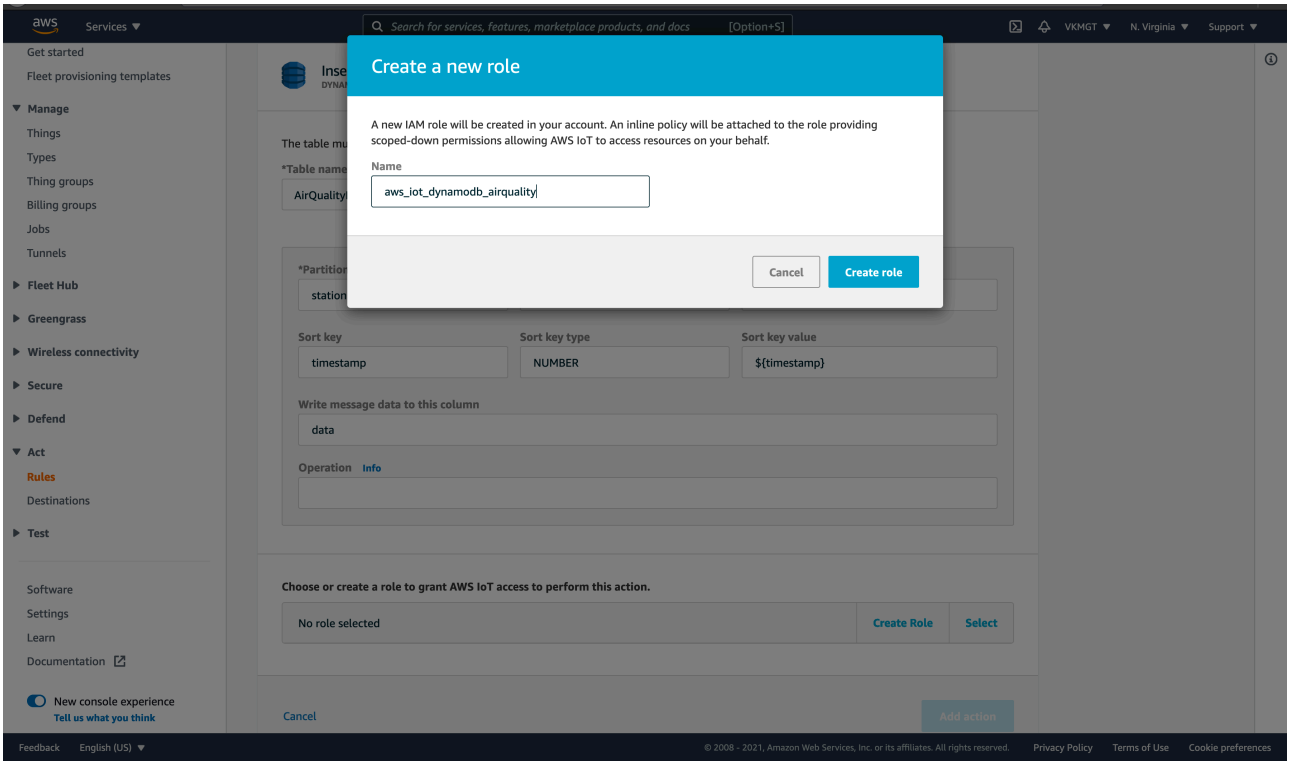
- Insert a new action to insert data into the **AirQualityData** table in DynamoDB. Configure the action as shown below





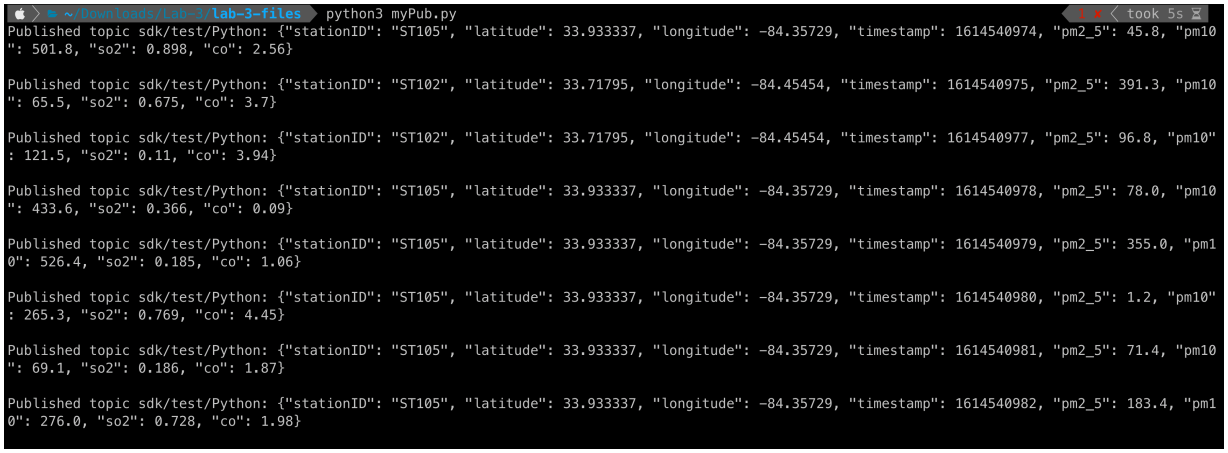
- Create a new role by clicking **Create Role** to allow access to insert the data into DynamoDB. Name it **aws_iot_dynamodb_airquality**, as shown below





8. Run the synthetic data generator to send data to AWS IoT Core (5 points)

- A synthetic data generator is provided with this lab (**myPub.py**). This data generator generates data on raw pollutant concentrations for two stations.
- Update the connection strings in this file to match the AWS IoT thing you created previously.
- Run the data generator to publish synthetic data on raw pollutant measurements to AWS IoT.



9. Verify Data in DynamoDB (5 points)

- After running the synthetic data generator for a minute, navigate back to the DynamoDB created previously, see if the raw air quality data sent by the publisher python program and the computed AQI data are saved in the corresponding databases.

10. Deploy the air quality monitoring Django application on EC2 (5 points)

- Deploy the air quality monitoring Django application provided with this lab on EC2.
- Update your EC2 instance by running this command once you are inside the instance:

```
sudo apt update
```

- Ensure you have the package installer pip3 and Python3 installed on the instance. You can use this command to instance these dependencies:

```
sudo apt install python3-pip
```

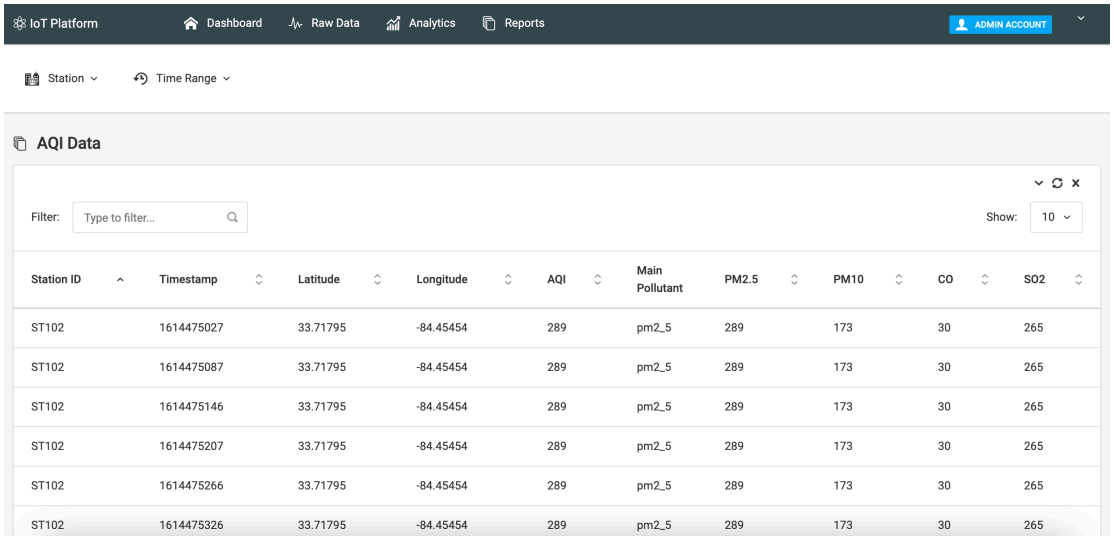
- Inside the **aqi-django-app** directory, run the following commands to install all the dependecies.

```
pip3 install -r requirements.txt
```

- After installing all required packages, run the Django application and view it within the browser.

Python3 manage.py runserver

- You should be able to view the IoT platform as:



11.Exercise (40)

- Create a new method in the Django application that allows a user to download the raw data in a CSV format.
- Using the programming language of preference, create a script to plot the data in a graphical line chart for each of the measurements and stations using the CSV with the raw data.
- Modify the script above to get the data from DynamoDB and display the graphical line chart changing in real-time. **** You can run it locally on your computer****
- Create a new Lambda Function with CloudWatch event rule that allows you to send notifications via email when the measurements are outside a particular threshold (See the table 1) .

Deliverables:

1. A video showing all the full functionalities of your Air Quality Monitoring Application, including the ones in the final exercise. Please show your AWS username clearly to indicate that you are using your account. You may briefly explain the functions that you created for the last activity, **but please make sure to limit the total length to 10 minutes.**
2. The complete code with the modifications needed to complete each exercise, including the new lambda function.