IoT with AWS

Aws hands on

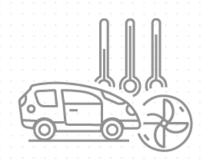


What is aws

- Amazon Web Services
- First largest pioneer in cloud services market
- Cloud Computing Solutions

- AWS IoT Core work?
- Connect and manage your devices









AWS IoT Core

Messages are transmitted and received using the MQTT protocol which minimizes the code footprint on the device and reduces network bandwidth requirements

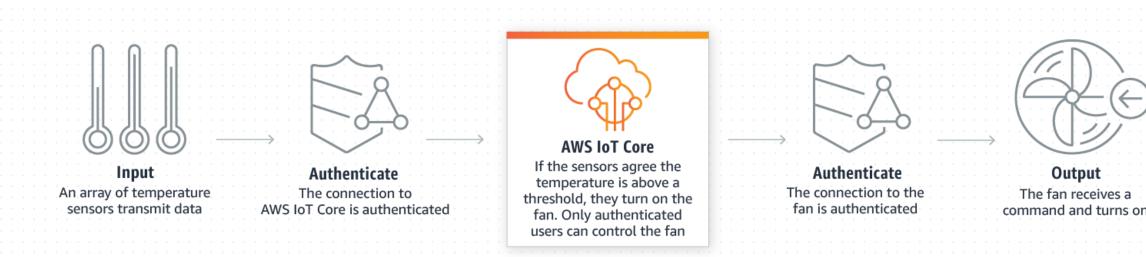


Devices communicate

AWS IoT Core enables devices to communicate with AWS services and each other

Secure device connections and data





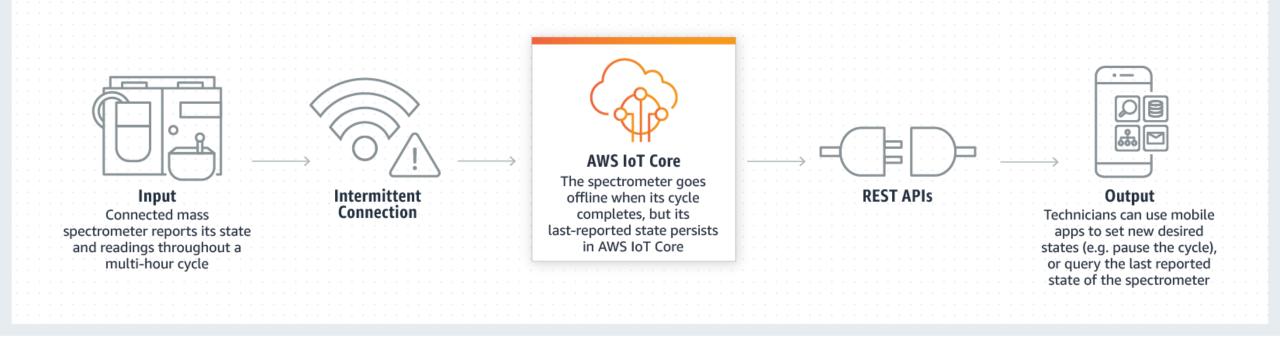
Process and Act on Data





Read and set device state at any time

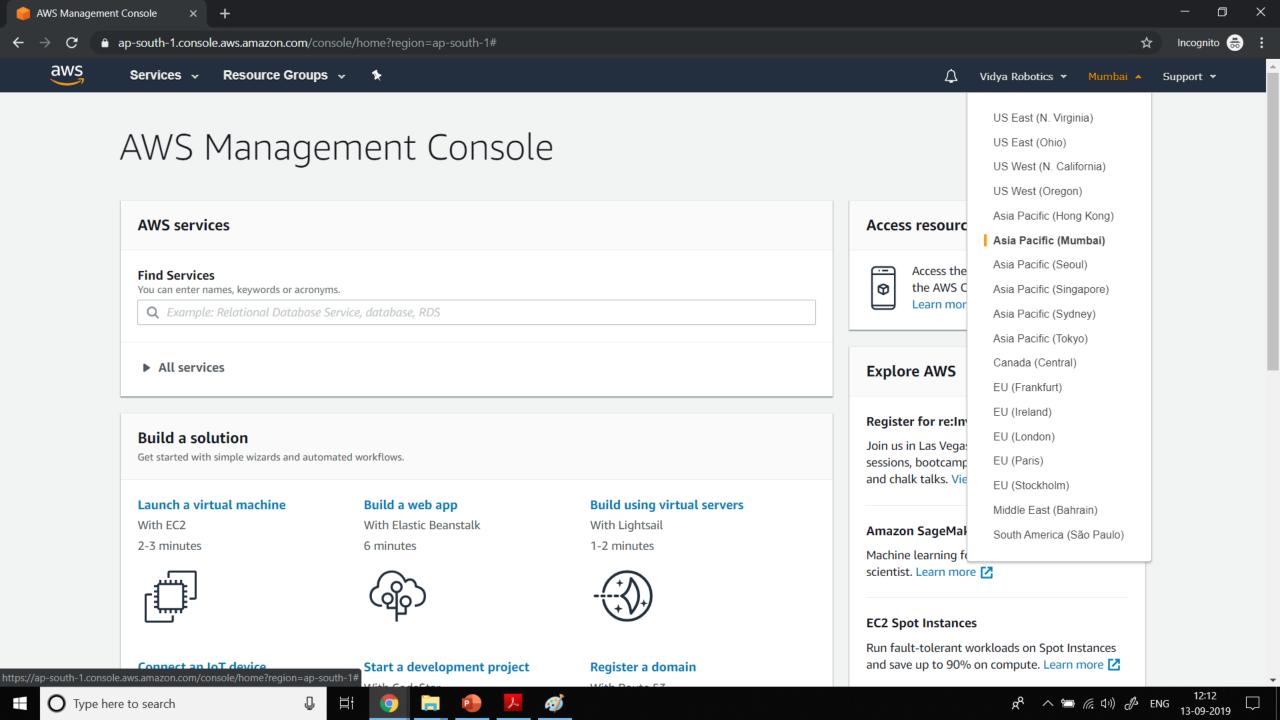


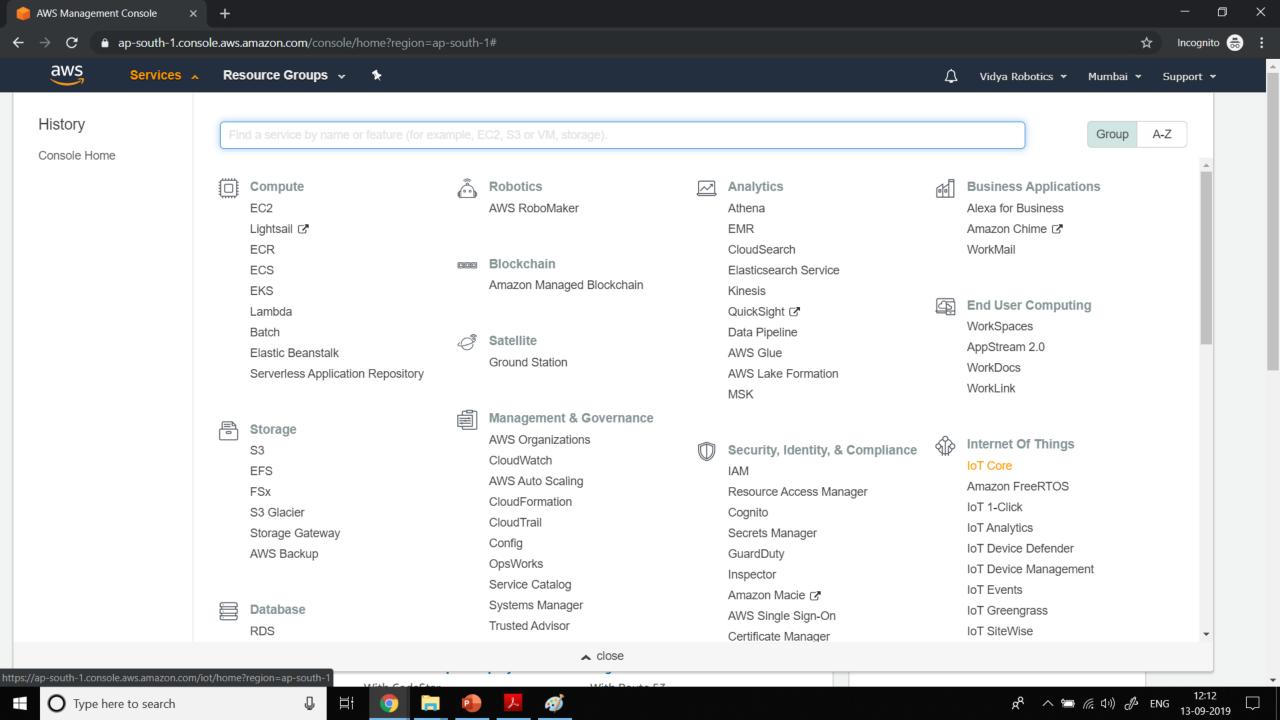


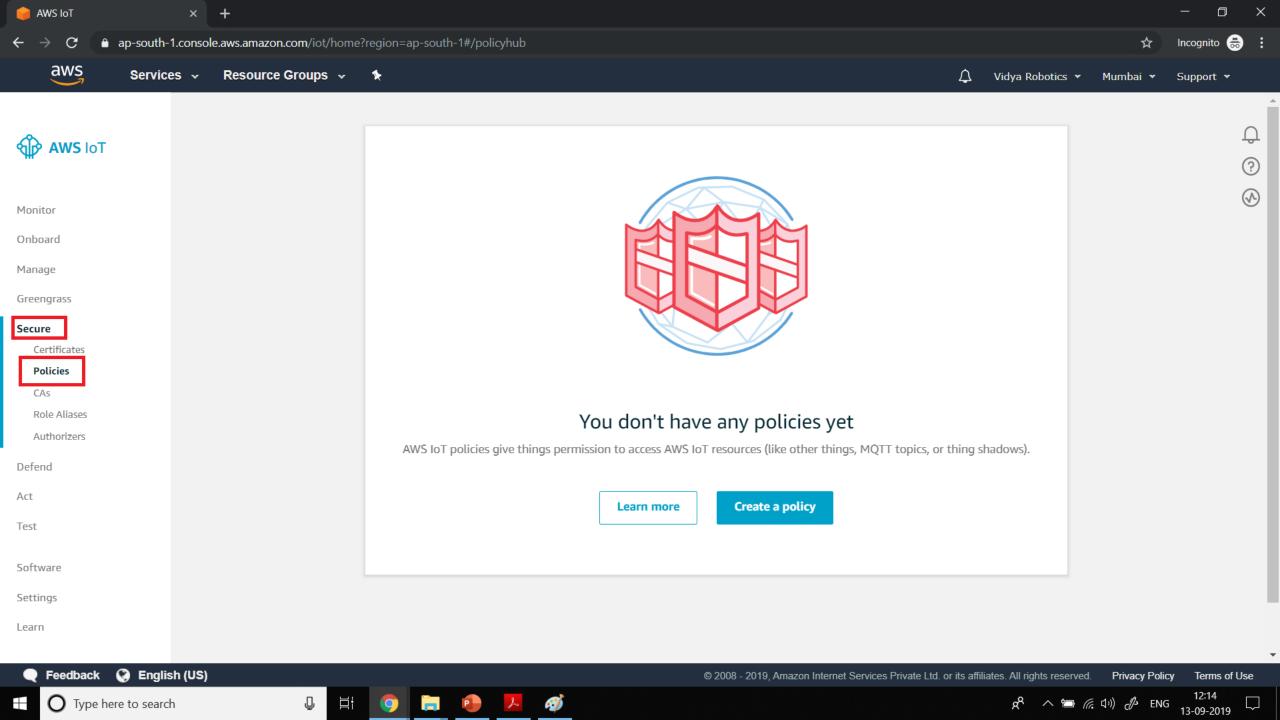
Get Started with AWS

- Signup for aws account
- Sign up for free 1 year trial
- Enter credit card information
- Log in









Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the AWS IoT Policies documentation page.

Name



Add statements

Policy statements define the types of actions that can be performed by a resource.

Advanced mode



Add statement



Creating AWS IoT things

An IoT thing is a representation and record of your physical device in the cloud. Any physical device needs a thing record in order to work with AWS IoT. Learn more.

Register a single AWS IoT thing

Create a thing in your registry

Create a single thing

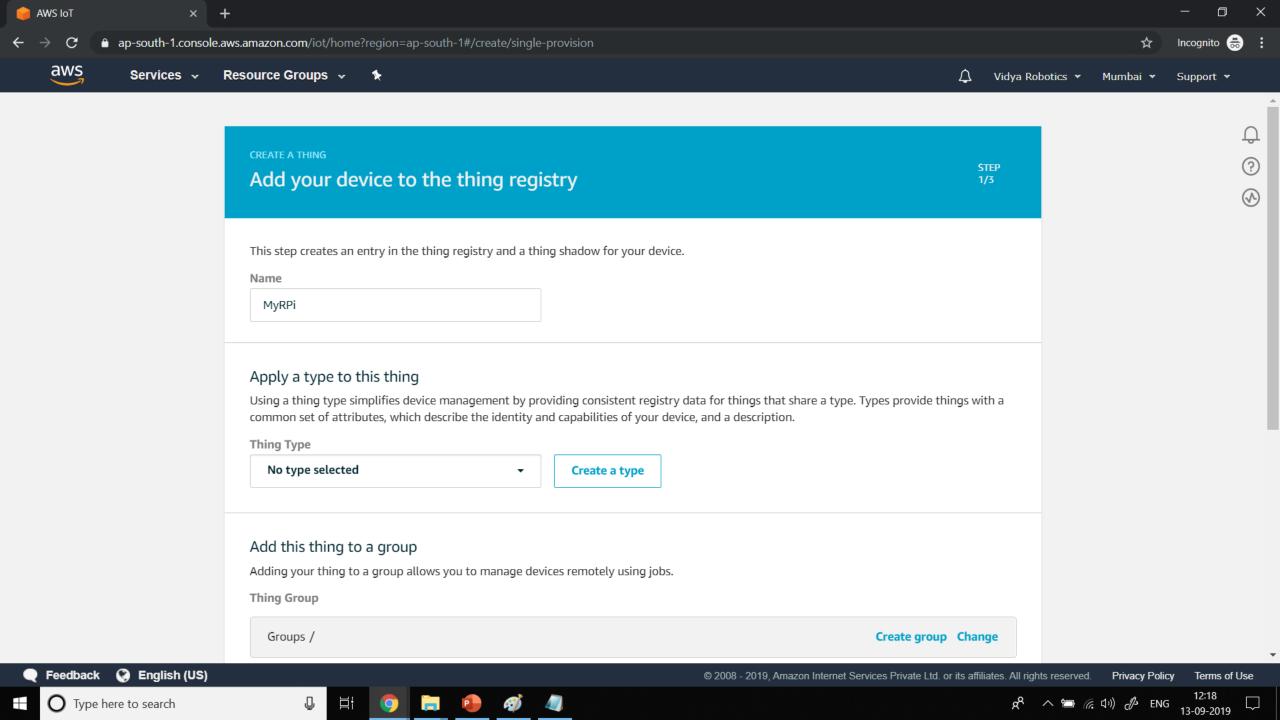
Bulk register many AWS IoT things

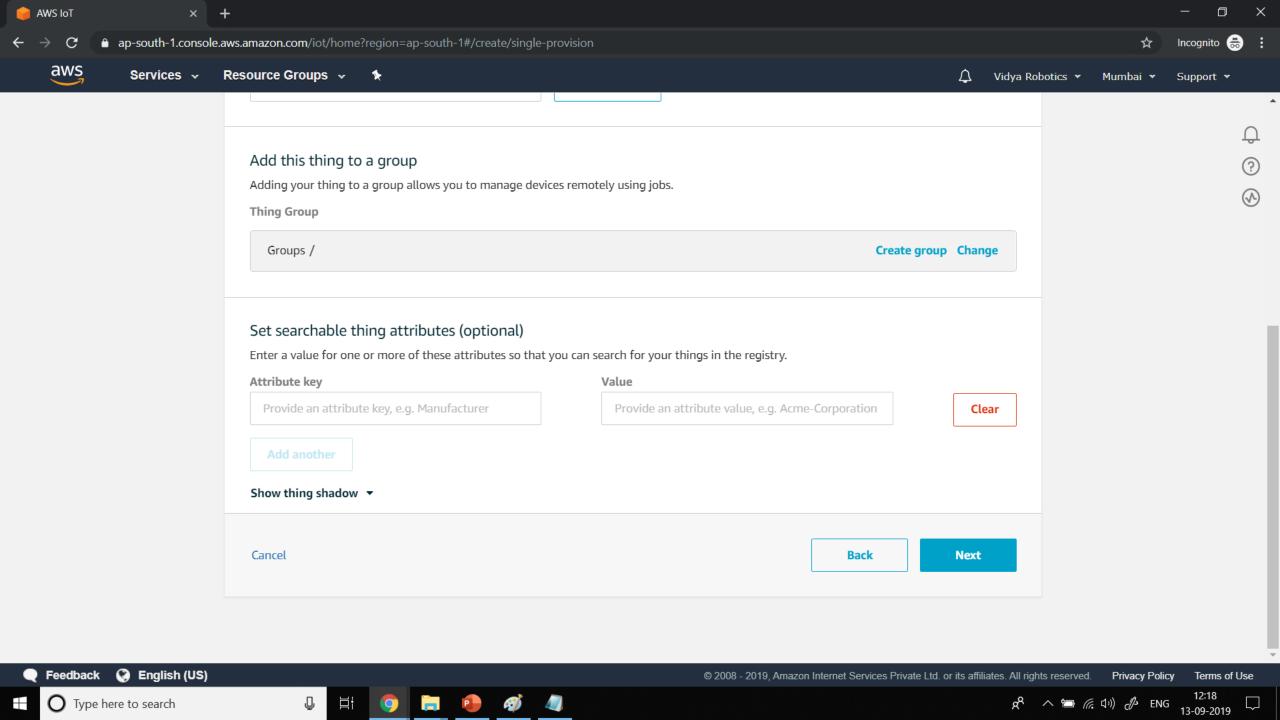
Create things in your registry for a large number of devices already using AWS IoT, or register devices so they are ready to connect to AWS IoT.

Create many things

Cancel

Create a single thing





Add a certificate for your thing

A certificate is used to authenticate your device's connection to AWS IoT.

One-click certificate creation (recommended)

This will generate a certificate, public key, and private key using AWS IoT's certificate authority.

Create certificate

Create with CSR

Upload your own certificate signing request (CSR) based on a private key you own.

♣ Create with CSR

Use my certificate

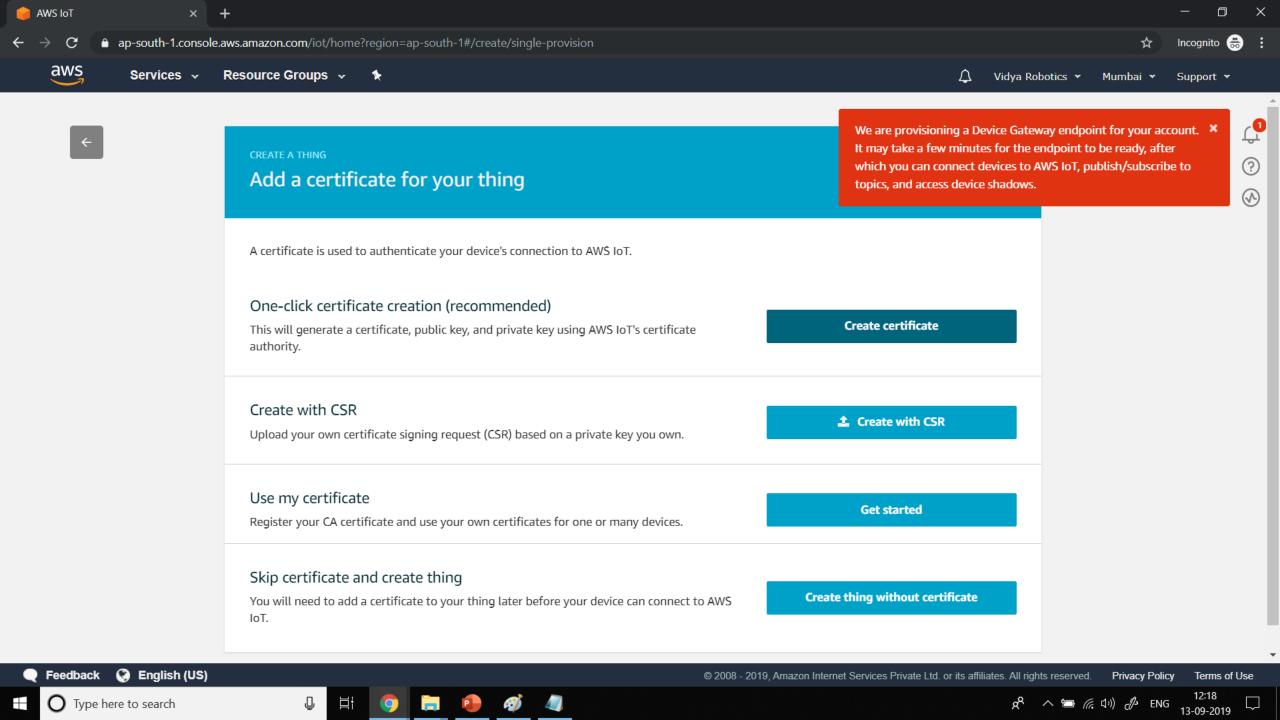
Register your CA certificate and use your own certificates for one or many devices.

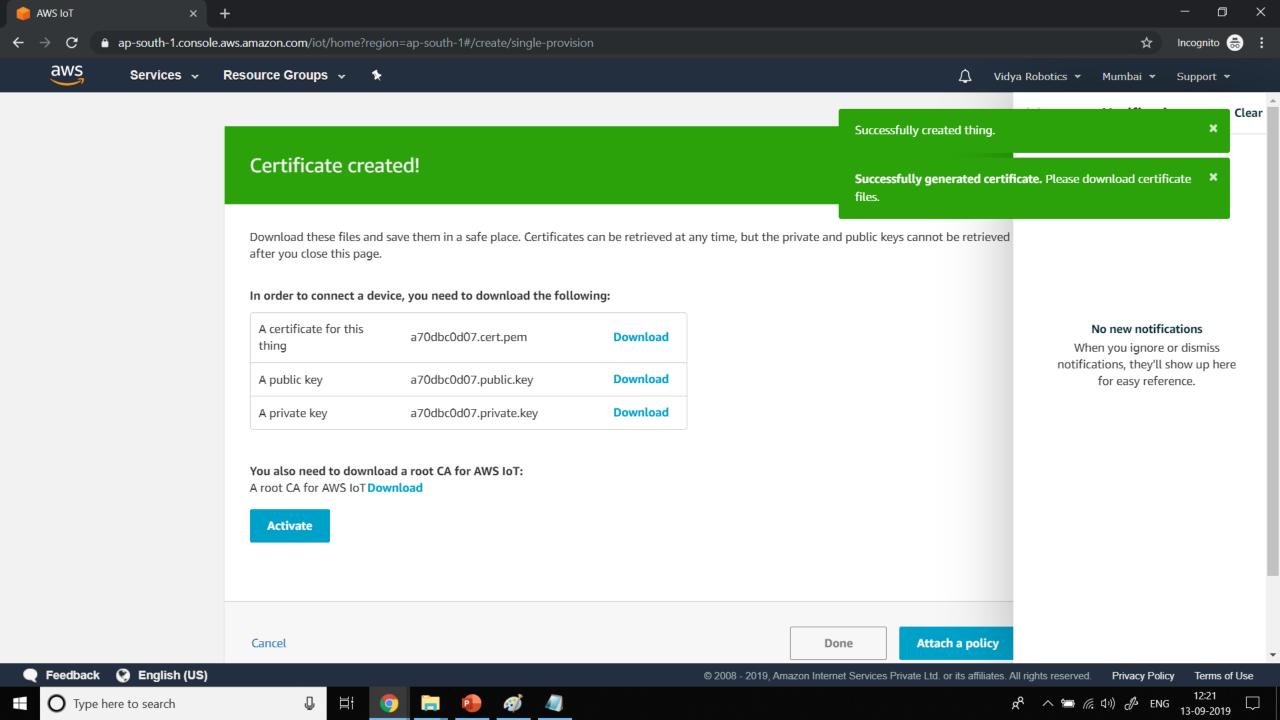
Get started

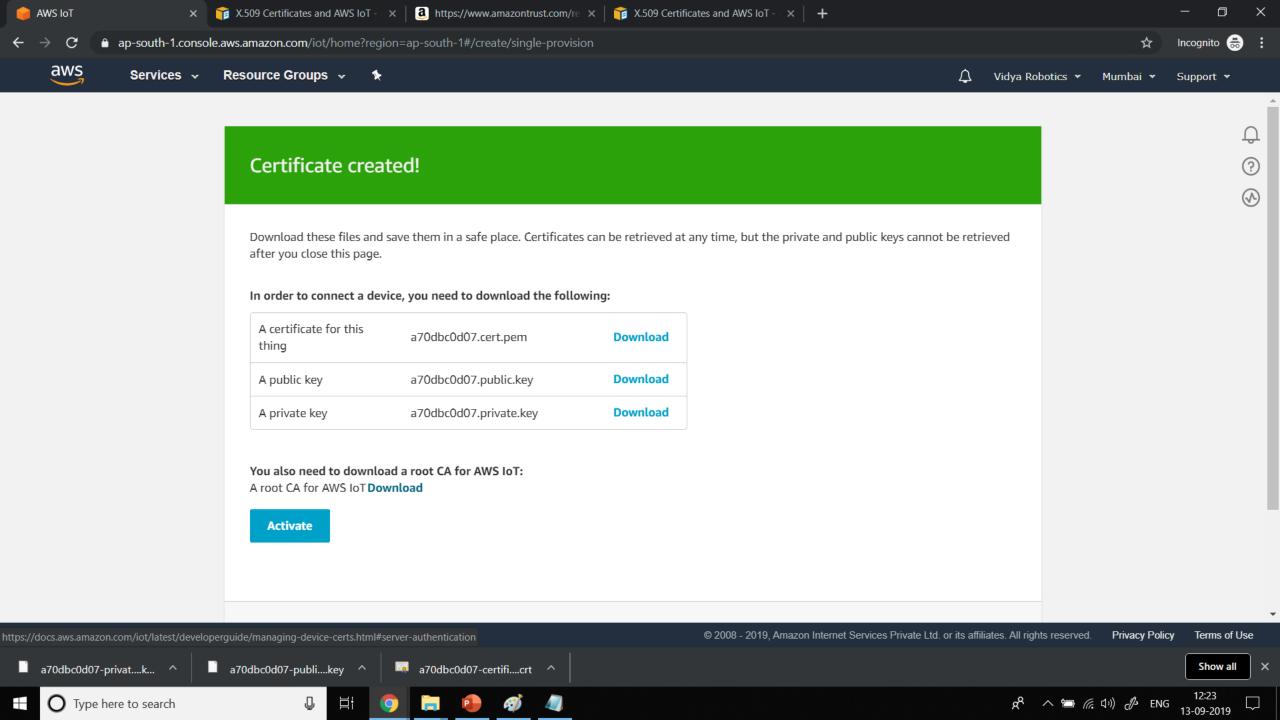
Skip certificate and create thing

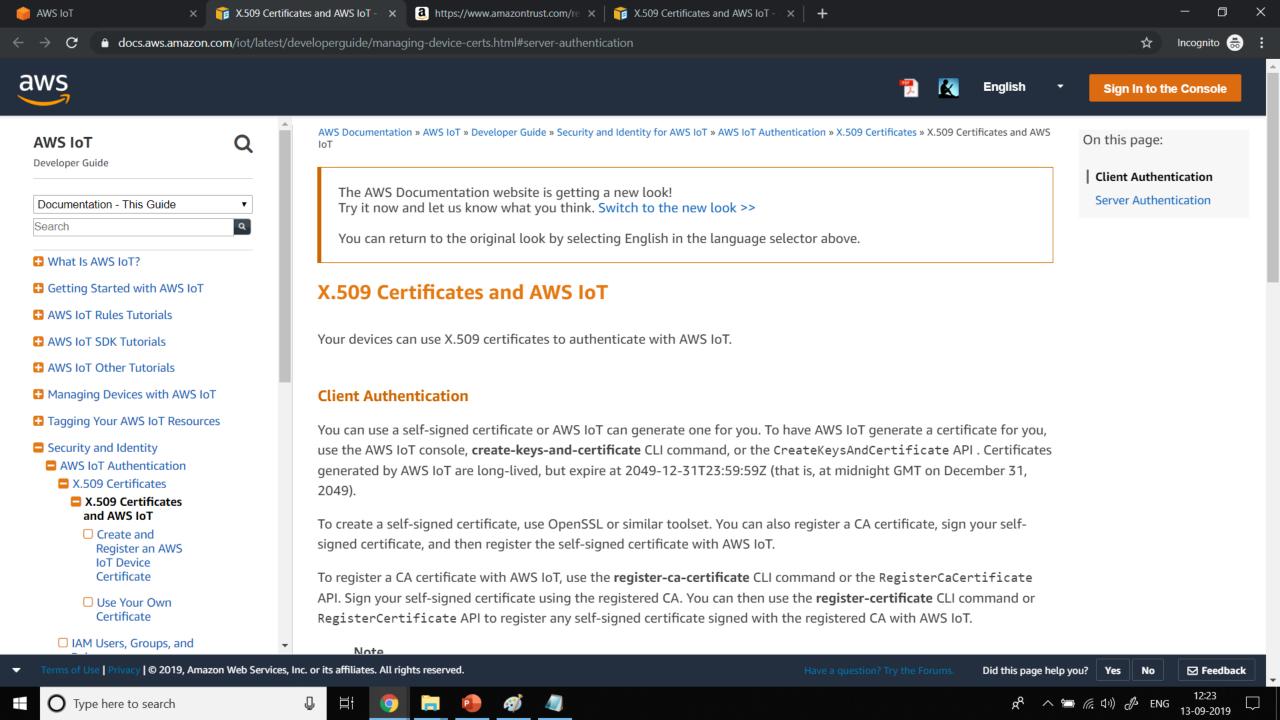
You will need to add a certificate to your thing later before your device can connect to AWS IoT.

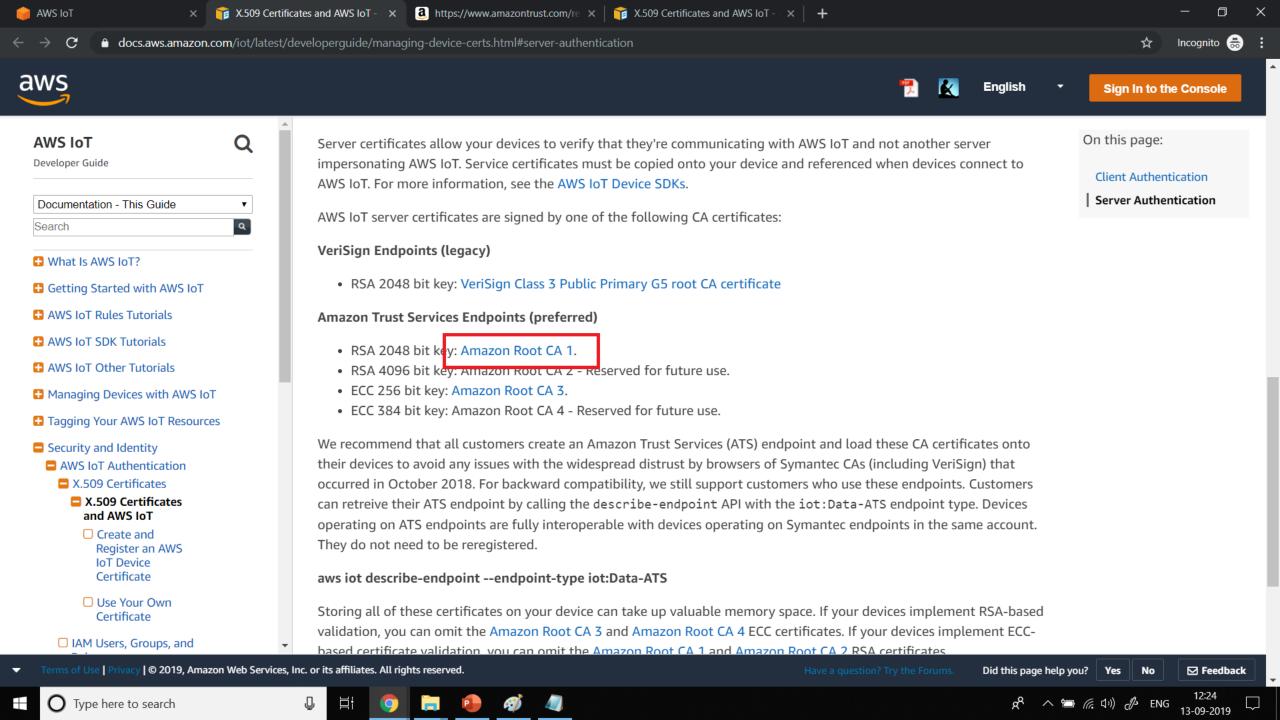
Create thing without certificate

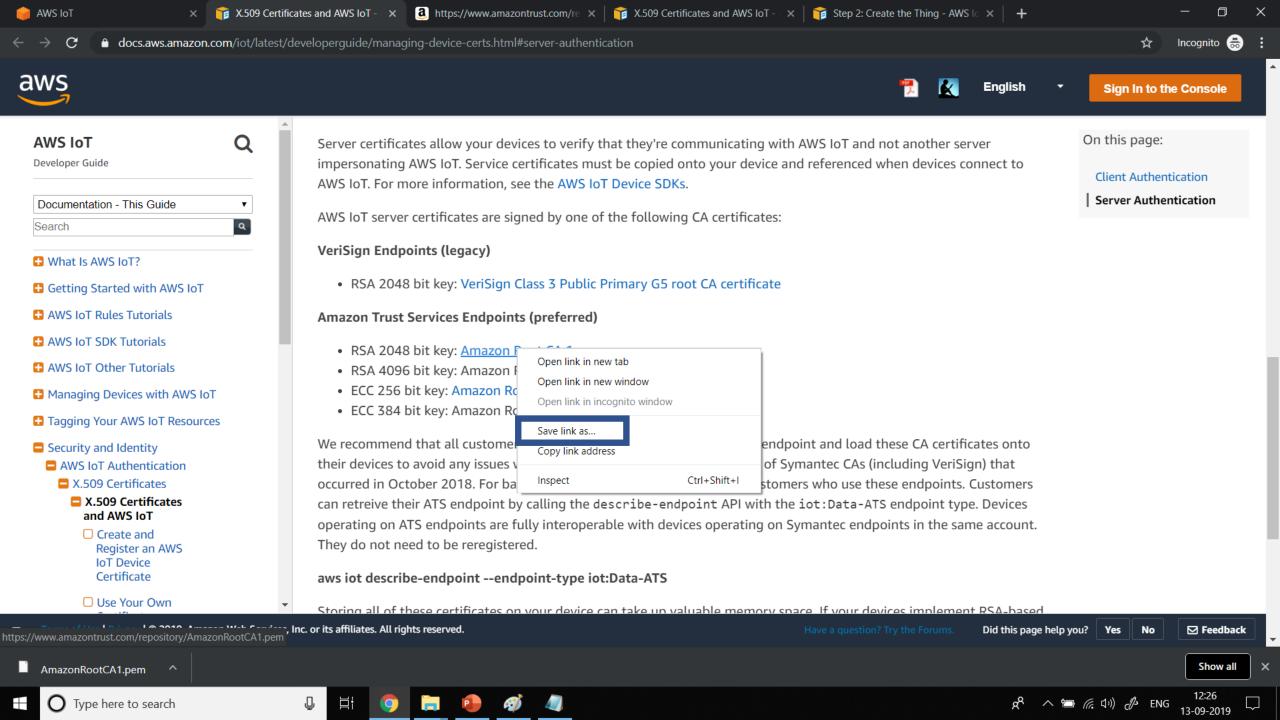


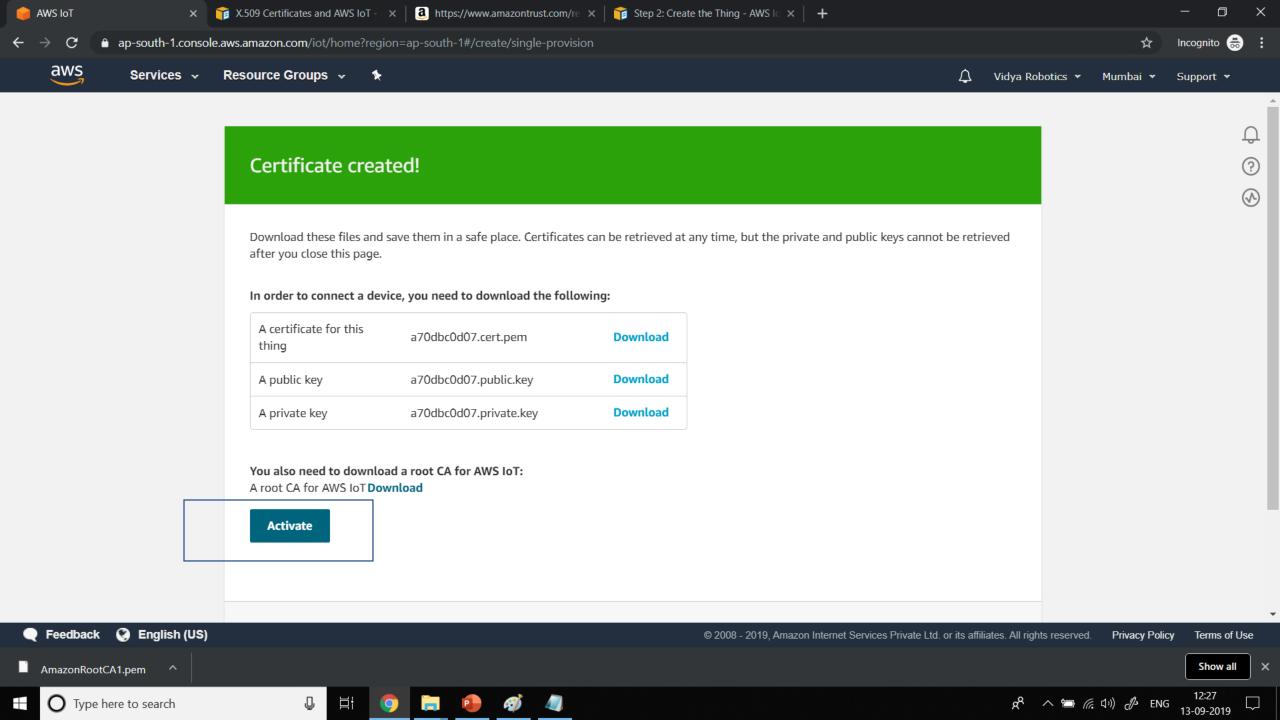


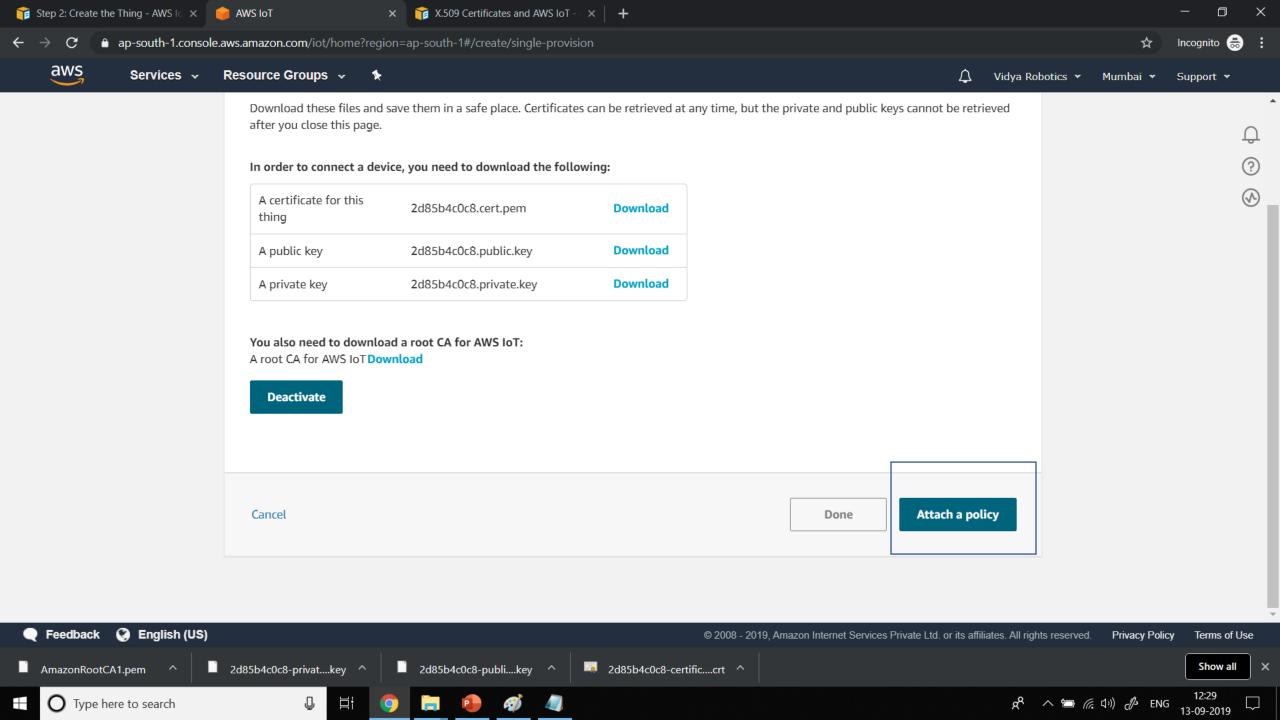


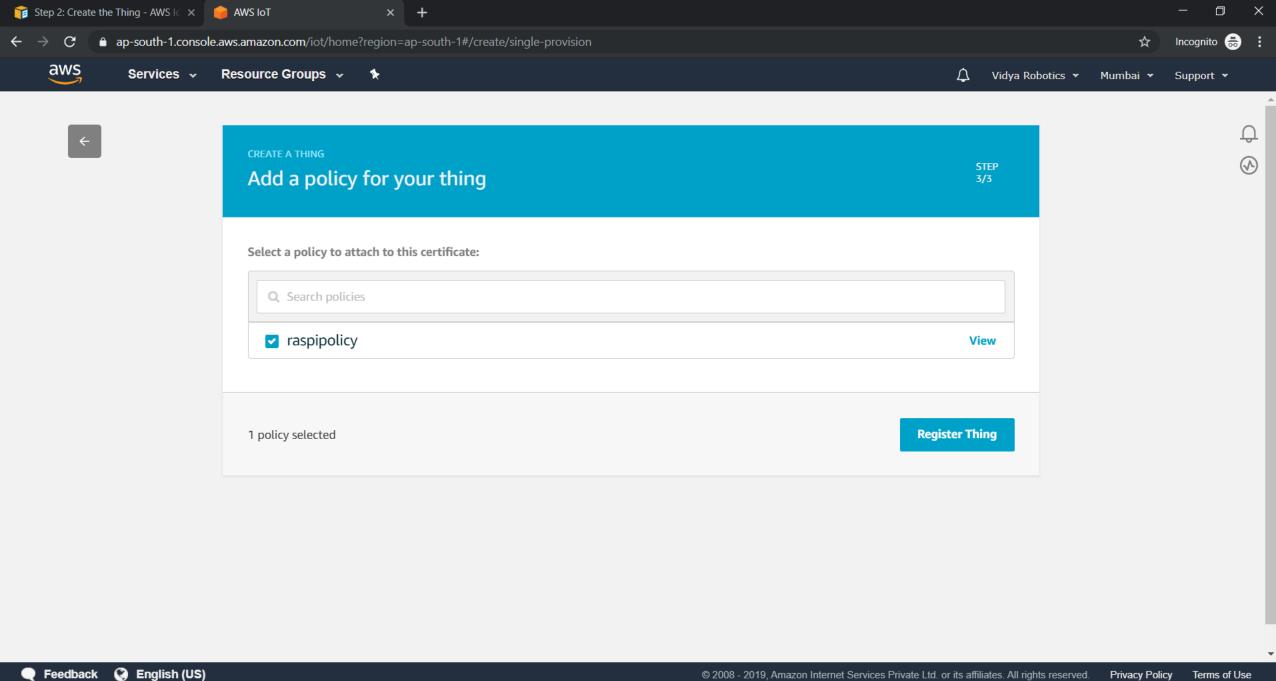










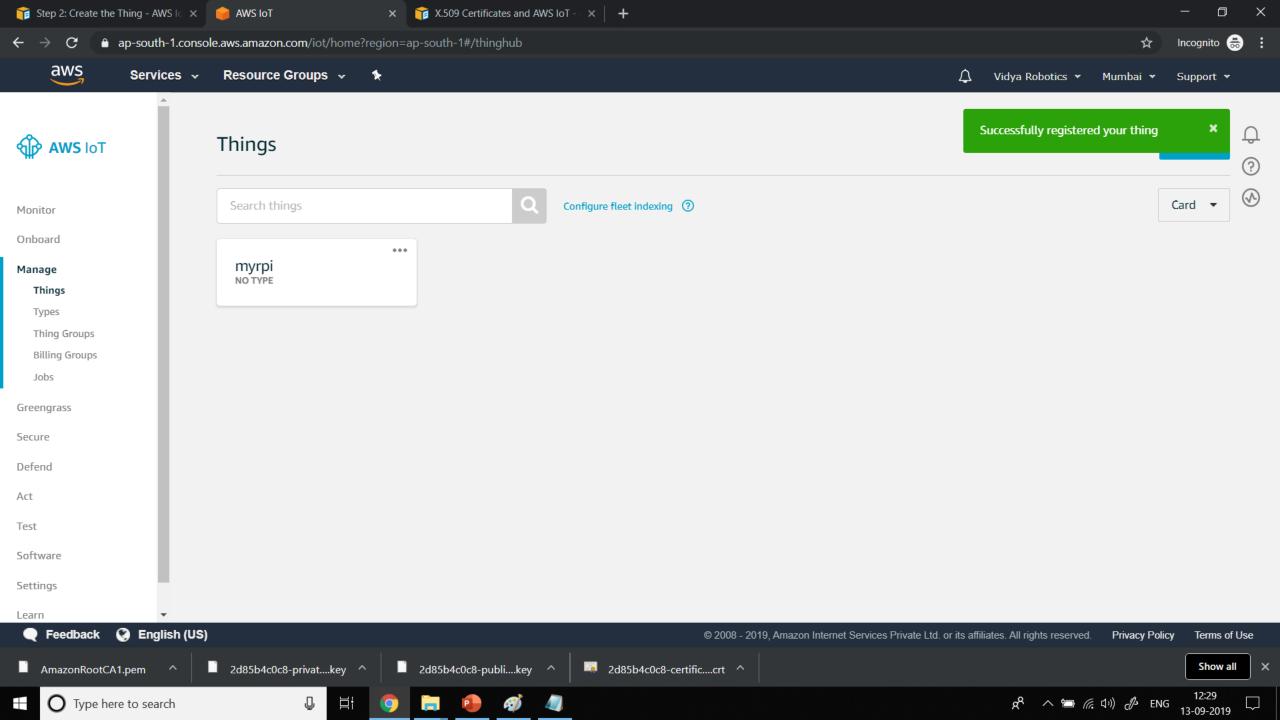


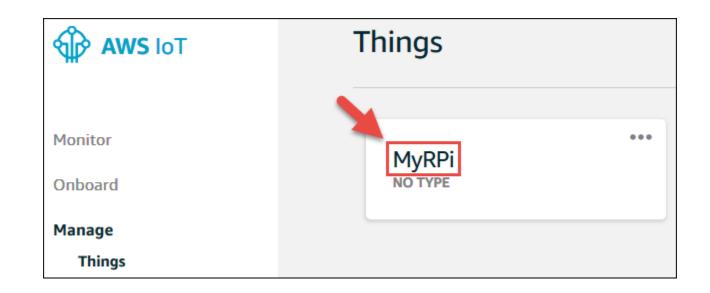
O Type here to search

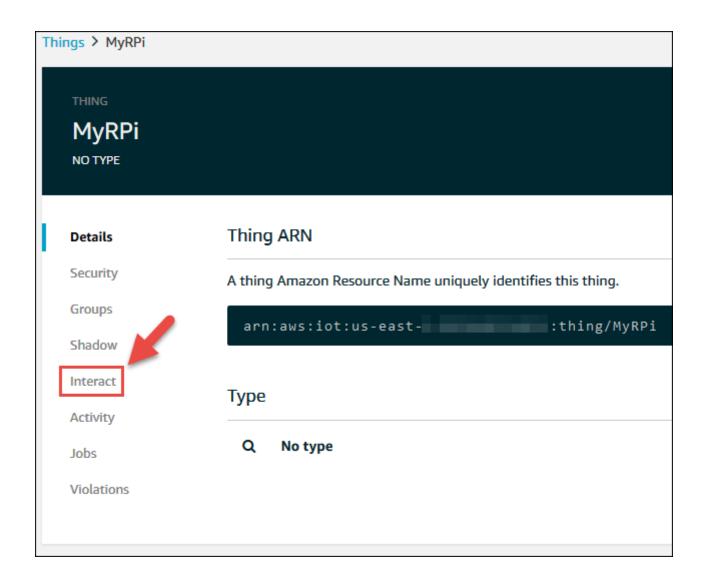












 Update to this thing shadow (for example, \$aws/things/MyRPi/shadow/update)

• Get this thing shadow (for example, \$aws/things/MyRPi/shadow/get)

 Accepted to get this thing shadow (for example, \$aws/things/MyRPi/shadow/get/accepted)

- Select the "Back" button.
- If the AWS IoT Device Management Introduction dialog is displayed, select Show Later or press Esc .
- In the service navigation pane, select **Test** .

- For the subscription topic, enter the MQTT topic value
- (for example, \$aws/things/MyRPi/shadow/update)
- that you updated to this thing shadow record in previous procedure, and then select the subscription topic.





Install Library and Test Code

- sudo pip3 install AWSIoTPythonSDK
- aws_mqtt_test.py
- Change the MQTT URL with the ARN we've saved
- Change the names of certificate files

```
mqtt_url = "a1f7vy2goilgqr-ats.iot.ap-south-1.amazonaws.com"
root_ca = 'AmazonRootCA1.pem'
public_crt = 'b0d993d0a9-certificate.pem.crt'
private key = 'b0d993d0a9-private.pem.key'
```

AWS MQTT TEST Code

aws_mqtt_test.py



Email alerts

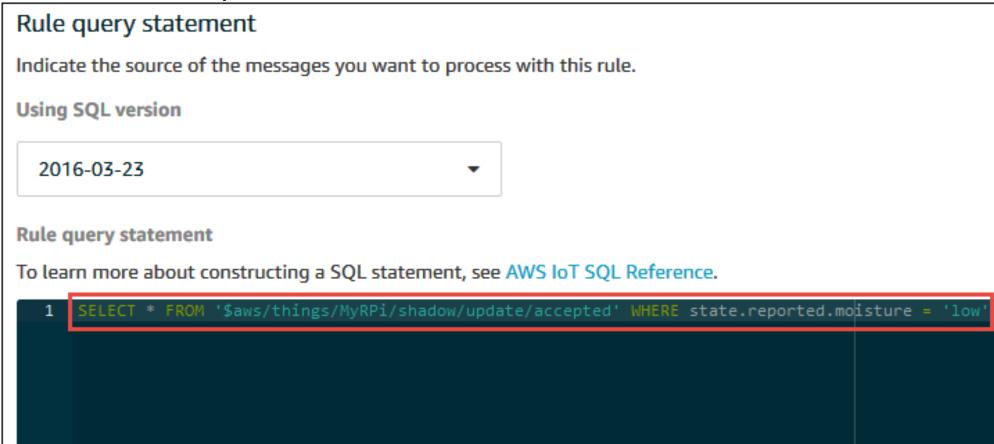


Email alert steps

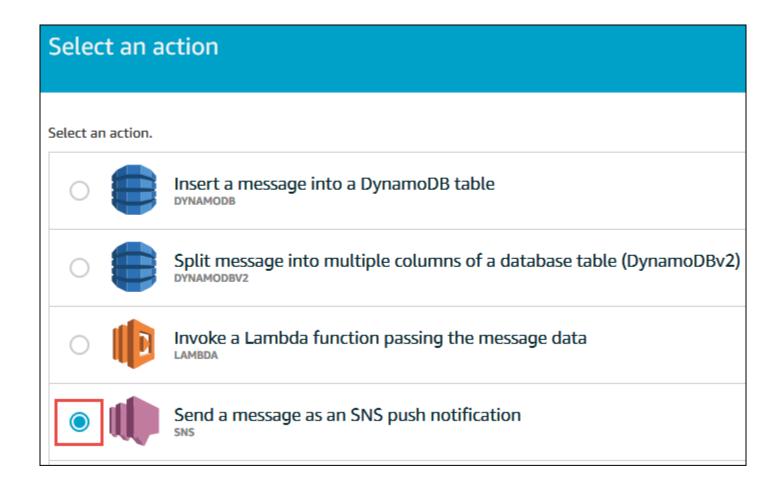
- you see that you don't have any rules dialogs, select Create a rule. Otherwise, choose Create.
- On the Create Rule page, enter a name for this rule (for example, MyRPiLowMoistureAlertRule). If you use a different name, be sure to replace it with this name in this example.
- For the description, please provide a meaningful description for this rule, for example, Sends an alert whenever soil moisture level readings are too low.
- For rule query statements, use the SQL version set to 2016-03-23. In the Rule Query Statement box, enter the following AWS IoT SQL statement in a single line, without any line breaks:

Rule Query

SELECT * FROM '\$aws/things/mypi/shadow/update/accepted'
 WHERE state.reported.moisture = 'low'

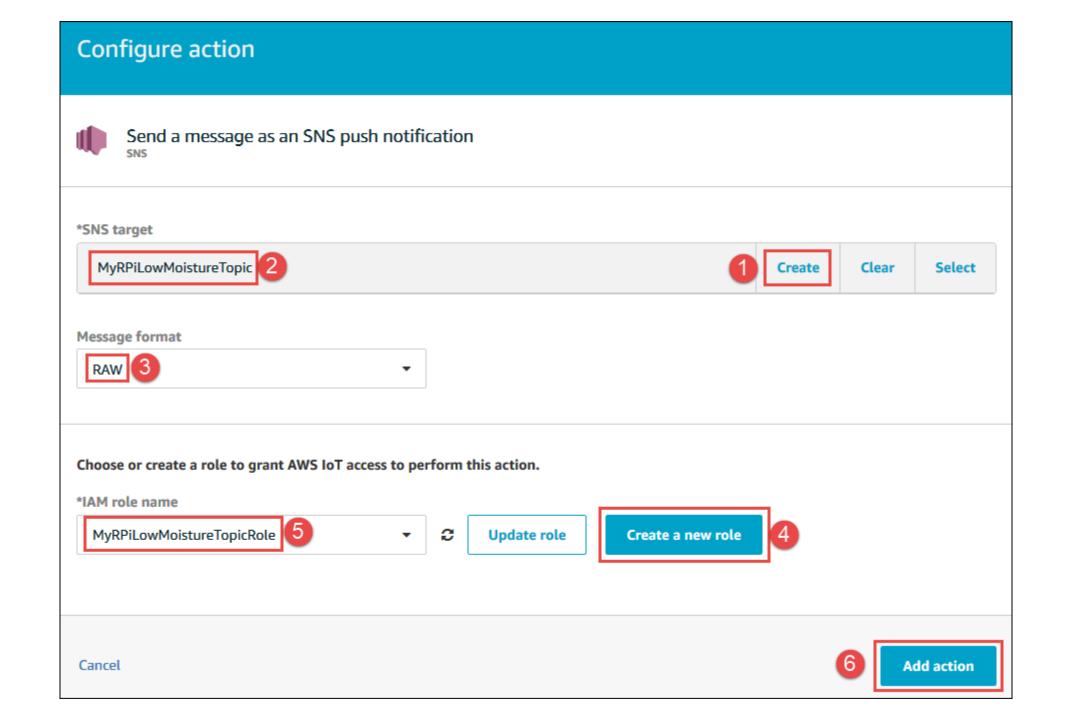


Select an action



Next Steps

- select Configure action .
- On the Configure Actions page, select Create for the SNS target. Enter the name of the SNS topic, such as MyRPiLowMoistureTopic, and select Create. If you choose to use a different name, be sure to replace it with this name in this example.
- For Message format , choose RAW .
- For the IAM role name, select Create a new role and enter a name for the new role, such as MyRPiLowMoistureTopicRole. If you choose to use a different name, be sure to replace it with this name in this example.
- Choose Create a new role.
- For the IAM role name, select MyRPiLowMoistureTopicRole.
- Select Add Action .



Subscription

Choose Create rule .

• Set up Amazon SNS to send messages to your email inbox via an Amazon SNS topic. In the AWS navigation bar, select a service. In the Find service by name or feature box, type SNS and press Enter.

In the service navigation pane, choose Subscribe.

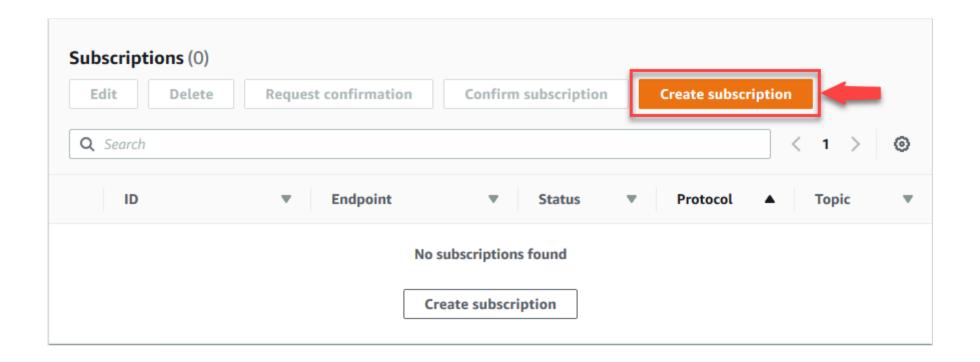
Dashboard



▼ Mobile

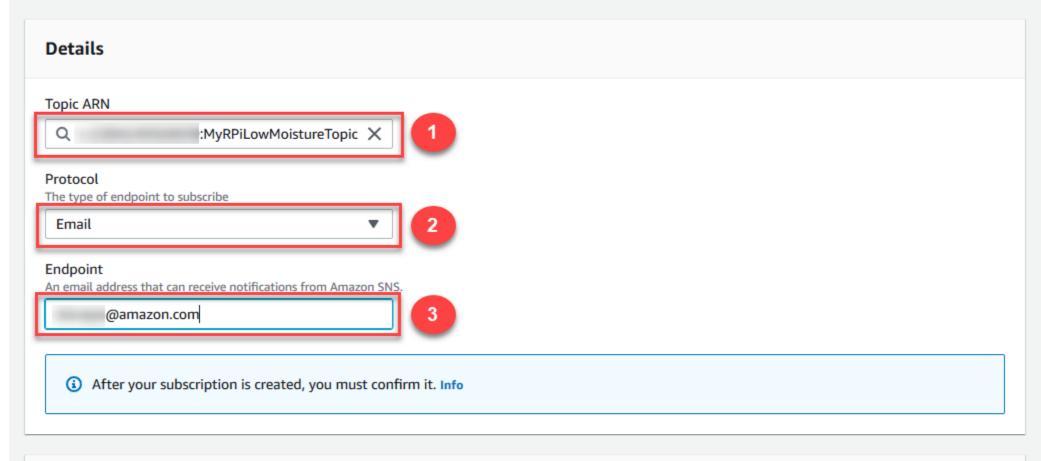
Push notifications

Text messaging (SMS)



- For the **topic ARN**, select the ARN for the topic that was created when the action was previously configured during this process.
- For the **agreement** , select **Email** .
- For Endpoint, enter your email address.
- Choose Create subscription.
- Check the subscription confirmation email in your inbox titled AWS Notification Subscription Confirmation from no-reply@sns.amazonaws.com (AWS Notifications Subscription Confirmation from no-reply@sns.amazonaws.com) . Once you receive the email, open the email and click the Confirm subscription link. When you click the link, a confirmation page will appear in your web browser. You can close this confirmation page.

Create subscription

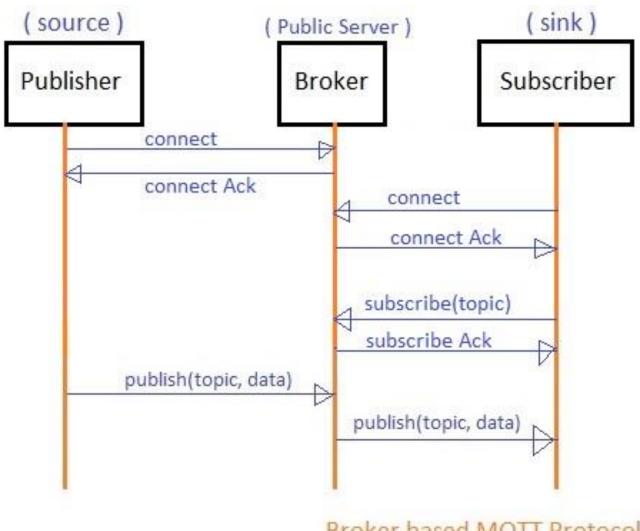


► Subscription filter policy - optional

This policy filters the messages that a subscriber receives. Info

Test Code

MQTT



Broker based MQTT Protocol

Test mqtt with sample Topics

MQTT Library Install

pip install paho-mqtt

• pip3 install paho-mqtt

AWS Data Storage

Using DynamoDB

DynamoDB rules allow you to take information from an incoming MQTT message and write it to a DynamoDB table.



Monitor

Onboard

Manage

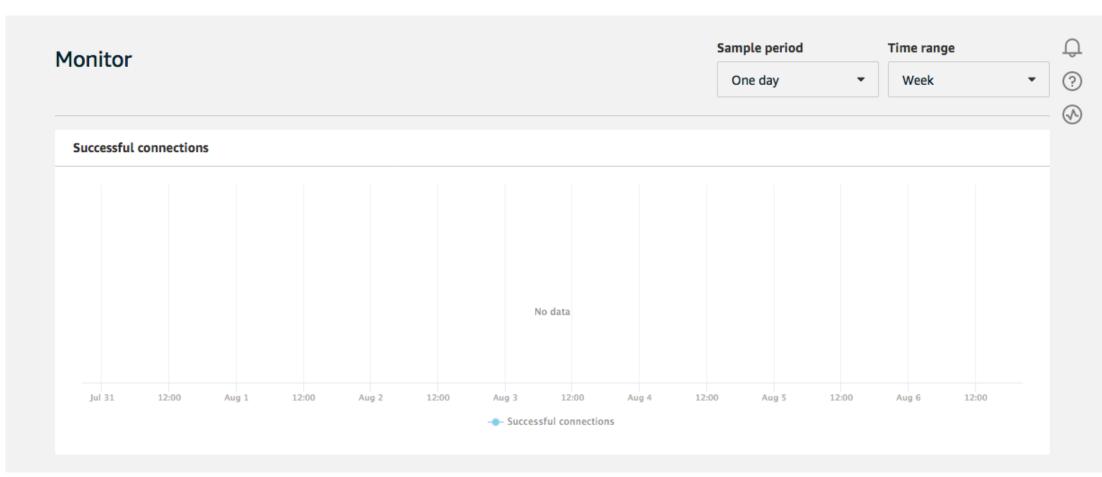
Greengrass

Secure

Defend

Act

Test





Monitor

Onboard

Manage

Greengrass

Secure

Defend

Act

Test

Software

Settings

Learn

Rules

Search rules Q







Card 🔻



Create a rule

Create a rule to evaluate messages sent by your things and specify what to do when a message is received (for example, write data to a DynamoDB table or invoke a Lambda function).

Name

GreenhouseRule

Description

A DynamoDB rule for a greenhouse

dynamoDB

• Under Rule query statement, choose the latest version from the Using SQL version list. For Rule query statement, enter:

SELECT * FROM 'my/greenhouse'

Rule query statement

Indicate the source of the messages you want to process with this rule.

Using SQL version



Rule query statement

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.

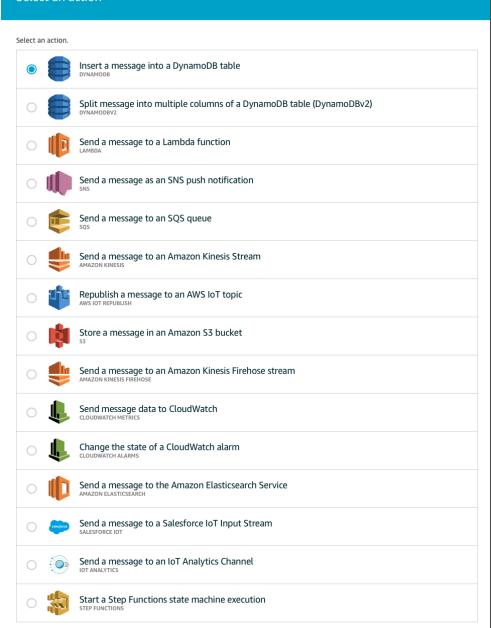


Set one or more actions

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. (*.required)



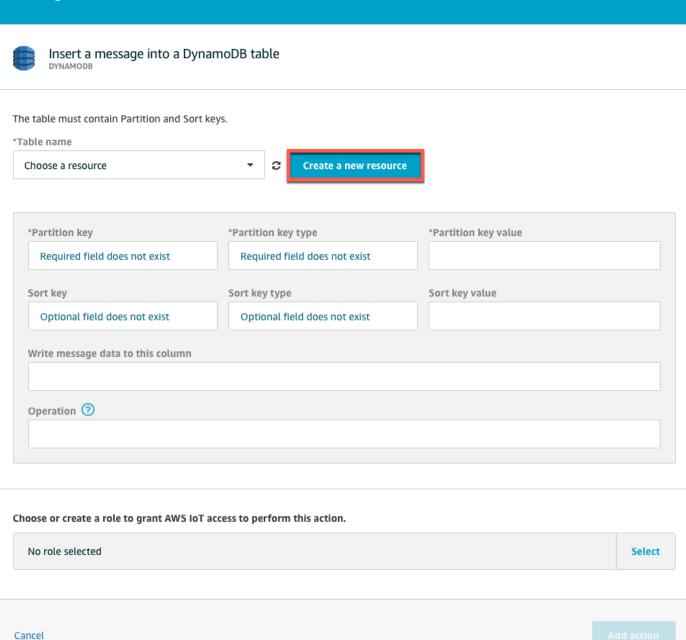
Select an action



dynamoDB

• On the Configure action page, choose Create a new resource.

Configure action



dynamoDB

• On the Amazon DynamoDB page, choose Create table.



Amazon DynamoDB

Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale. Its flexible data model and reliable performance make it a great fit for mobile, web, gaming, adtech, IoT, and many other applications.



Getting started guide



Create tables

Create DynamoDB tables with a few clicks. Just specify the desired read and write throughput for your table, and DynamoDB handles the rest

More about DynamoDB throughput



Add and query items

Once you have created a DynamoDB table, use the AWS SDKs to write, read, modify, and query items in DynamoDB.

DynamoDB API reference



Monitor and manage tables

Using the AWS Management Console, you can monitor performance and adjust the throughput of your tables, enabling you to scale seamlessly.

Monitoring tables

DynamoDB documentation & support

Getting started guide | FAQ | Developer guide | Forums | Report an issue

dynamoDB

- On the Create DynamoDB table page, enter a name in Table name.
- In Partition key, enter Row.
- Select Add sort key, and then enter PositionInRow in the Sort key field.
- Row represents a row of plants in a greenhouse.
- PositionInRow represents the position of a plant in the row.
- Choose String for both the partition and sort keys,
- Choose Create.
- It takes a few seconds to create your DynamoDB table.
- Close the browser tab where the Amazon DynamoDB console is open.
- If you don't close the tab, your DynamoDB table is not displayed in the Table name list on the Configure action page of the AWS IoT console



DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

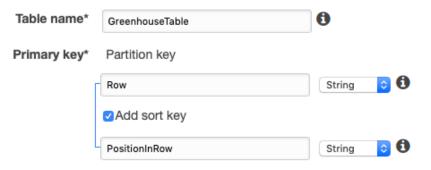


Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

✓ Use default settings

- · No secondary indexes.
- · Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".
- Encryption at Rest with DEFAULT encryption type NEW!

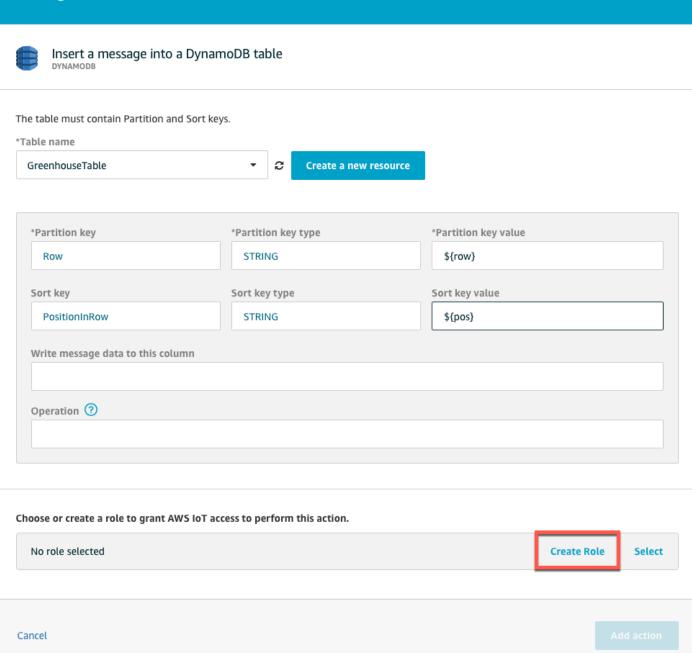
You do not have the required role to enable Auto Scaling by default. Please refer to documentation.

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

dynamoDB

- On the Configure action page, choose your new table from the Table name list.
- In Partition key value, enter \${row}. This instructs the rule to take the value of the row attribute from the MQTT message and write it into the Row column in the DynamoDB table. In Sort key value, enter \${pos}. This writes the value of the pos attribute into the PositionInRow column. Leave Write message data to this column blank. By default, the entire message is written to a column in the table named Payload. Choose Create a new role.

Configure action

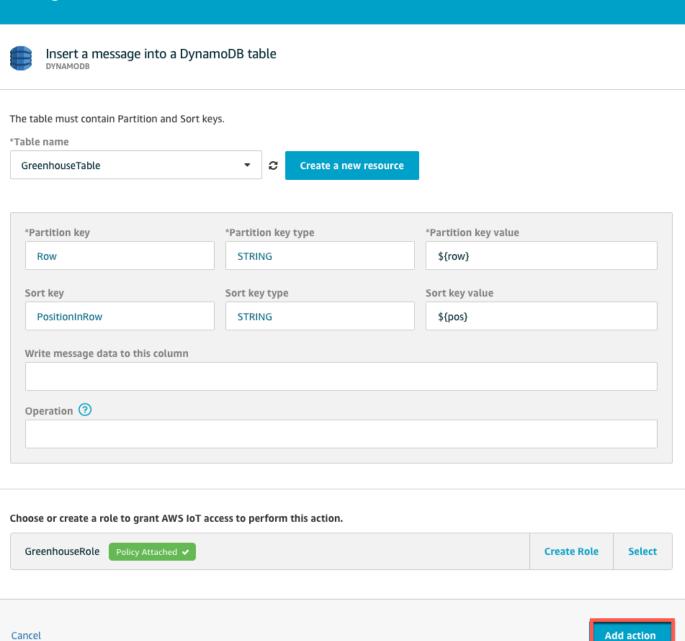


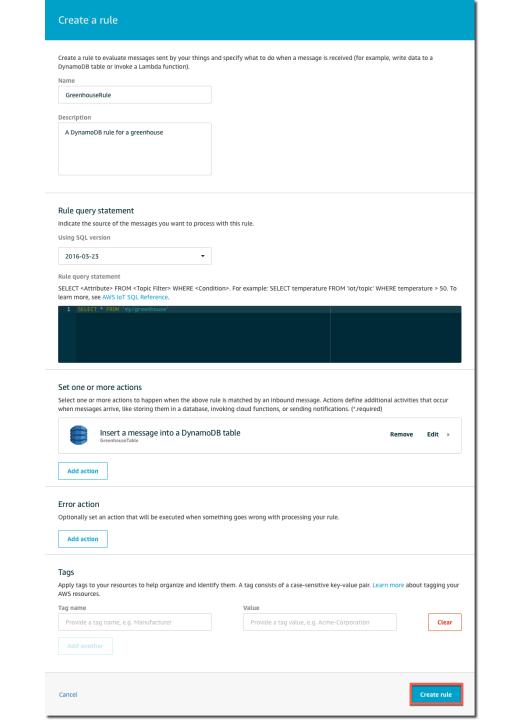
dynamoDB

• In Create a new role, enter a unique name in Name, and then choose Create role.

Create a new role A new IAM role will be created in your account. An inline policy will be attached to the role providing scoped-down permissions allowing AWS IoT to access resources on your behalf. Name GreenhouseRole Cancel Create role

Configure action

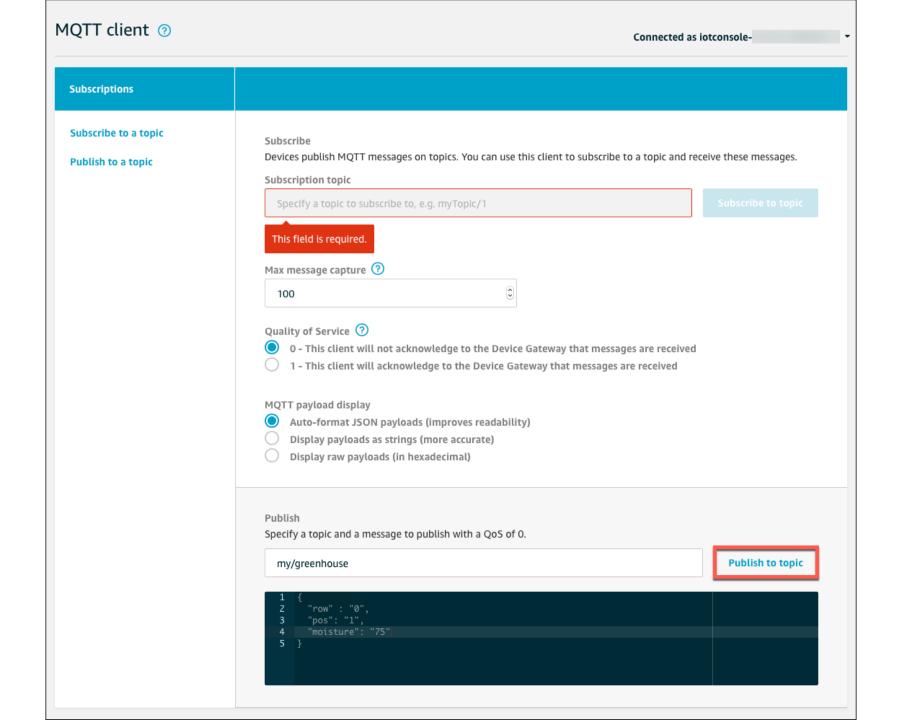




 To test the rule, open the AWS IoT console and from the navigation pane, choose Test.

Choose Publish to a topic. In the Publish section, enter
 my/greenhouse as the topic. In the message area, enter the following
 JSON:

```
{
    "row" : "0",
    "pos" : "0",
    "moisture" : "75"
    }
```



DynamoDB

Dashboard

Tables

Backups

Reserved capacity

Preferences

DAX

Dashboard

Clusters

Subnet groups

Parameter groups

Events

Create table

Amazon DynamoDB is a fully managed non-relational database service that provides fast and predictable performance with seamless scalability.

Create table

Recent alerts

No CloudWatch alarms have been triggered.

View all in CloudWatch

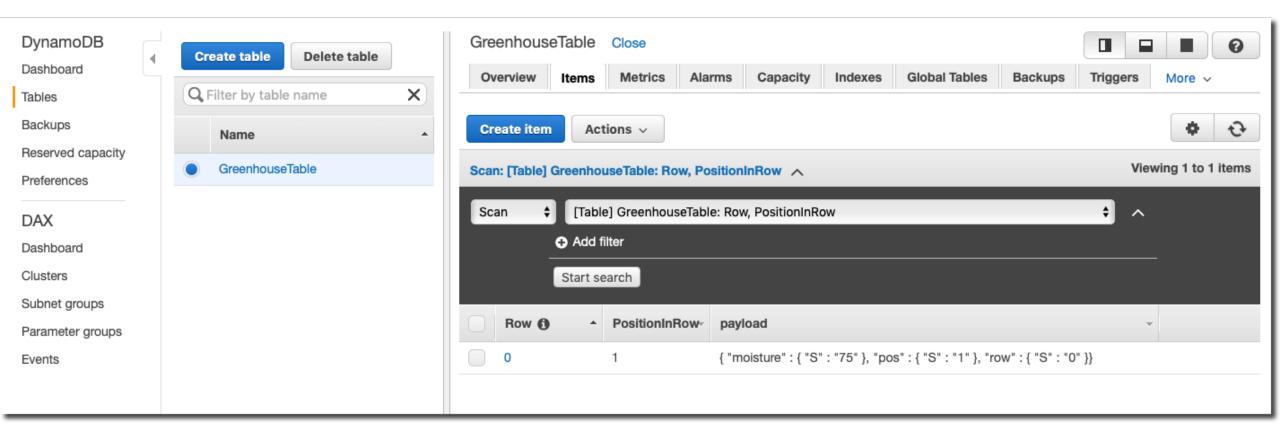
Total capacity for US West (Oregon)

| Provisioned read capacity | 5 | Reserved read capacity | 0 |
|----------------------------|---|-------------------------|---|
| Provisioned write capacity | 5 | Reserved write capacity | 0 |

Service health

| Current Status | | Details |
|----------------|--------------------------|-------------------------------|
| 0 | Amazon DynamoDB (Oregon) | Service is operating normally |

View complete service health details



Thank You