AWS based solar panel monitoring

by

Shaik mahaboob basha
Electronics & Communication Engineer

Index

Introduction

This is a own project to develop own server data base to perform task like ThinkSpeak and Blynk, where by this we can add more feature to our project and control the data as we required and the main important point is secure where third party website will share the data but in our project we can make secure everything in our AWS server

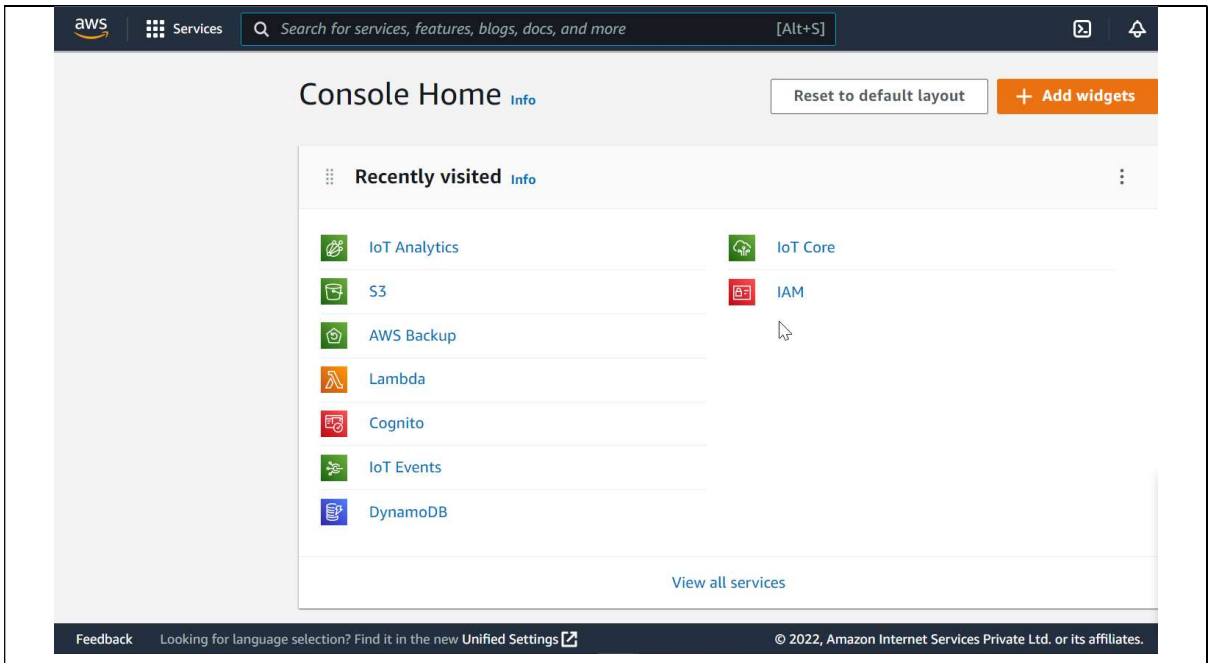
Plan

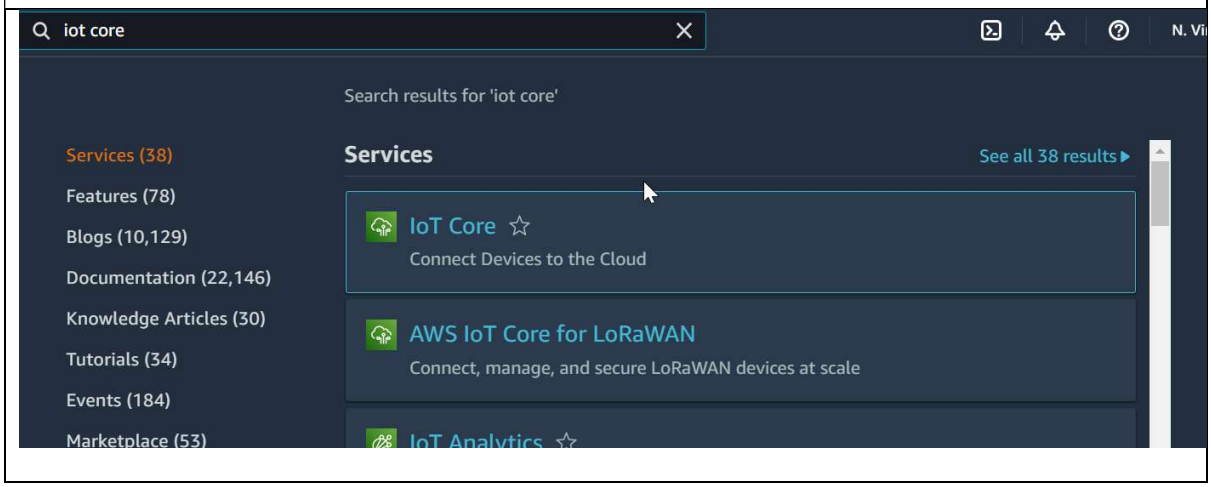
Here we setup process will be shown so that we can build our project faster

Aws setup

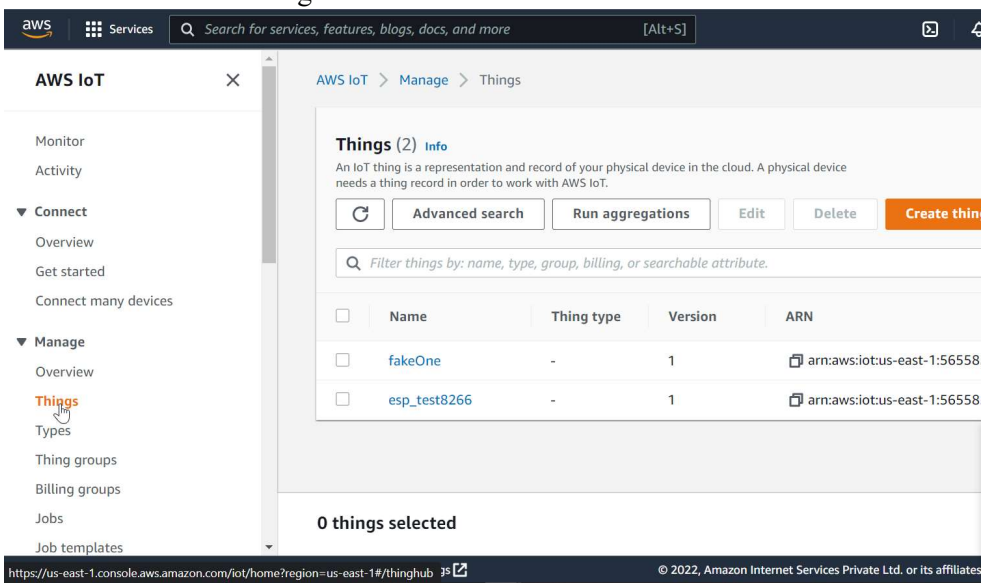
How to configure the AWS IOT things with the Arduino board :

1. First create a account of AWS which requires atm card , don't panic it wont take much money if your new then It will provide 1 year free trail at 2 rupees cost
2. Now open aws.amazon.com/console
3. Login with your user's name and password
4. Now on the search bar search about IOT core on search box

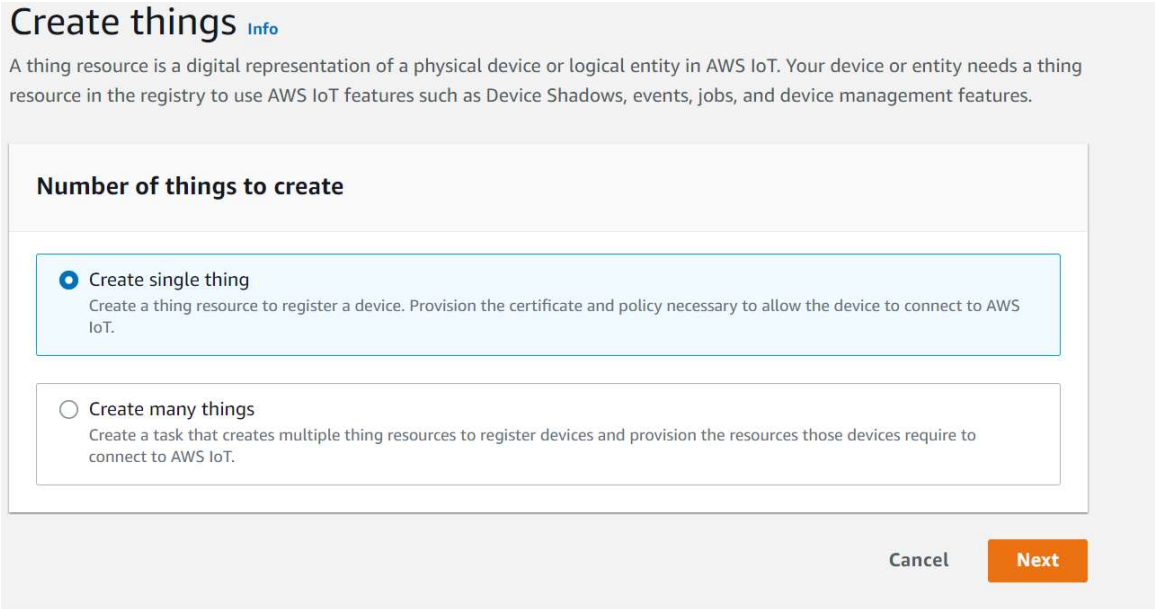




Go there and search about things



5. Now click on create things
    - Enter the single thing



- And chose the configuration
- Now we need certificates to get the connection between AWS so we use mqtt protocol , so we need certificates to get connection to communicate via internet (https) =➔ select autogenerated certificate

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

Specify thing properties Info

A thing resource is a digital representation of a physical device or logical entity in AWS IoT. Your device or entity needs a thing resource in the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features.

Thing properties Info

Thing name

documentThing

Enter a unique name containing only: letters, numbers, hyphens, colons, or underscores. A thing name can't contain any spaces.

Additional configurations

You can use these configurations to add detail that can help you to organize, manage, and search your things.

- **Thing type** - *optional*
- **Searchable thing attributes** - *optional*
- **Thing groups** - *optional*
- **Billing group** - *optional*

Device Shadow Info

Device Shadows allow connected devices to sync states with AWS. You can also get, update, or delete the state information of this thing's shadow using either HTTPs or MQTT topics.

- (●) No shadow
- ( ) Named shadow
  Create multiple shadows with different names to manage access to properties, and logically group your devices properties.
- ( ) Unnamed shadow (classic)
  A thing can have only one unnamed shadow.

Cancel | 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 you 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.

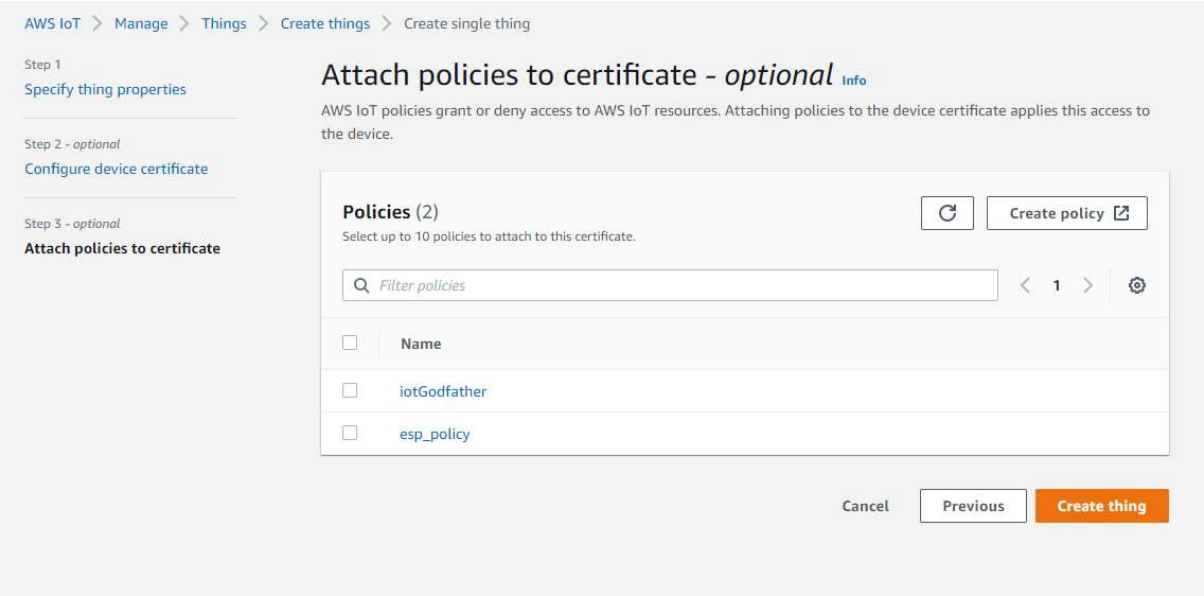
Cancel | Previous | Next

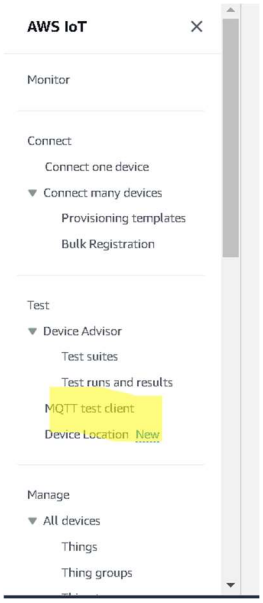
- now you need policy (which is like rules in which path want to receive data or not)
- where the topics line will be
    i. topicname/pub : publish data
    ii. topicname/sub : subscriber which receive the data as assign that rules to certain path
    iii. etc like that you can refer the other polices

https://docs.aws.amazon.com/iot/latest/developerguide/iot-policies.html

https://docs.aws.amazon.com/iot/latest/developerguide/pub-sub-policy.html

you can the traffic on mqtt test client by adding topics



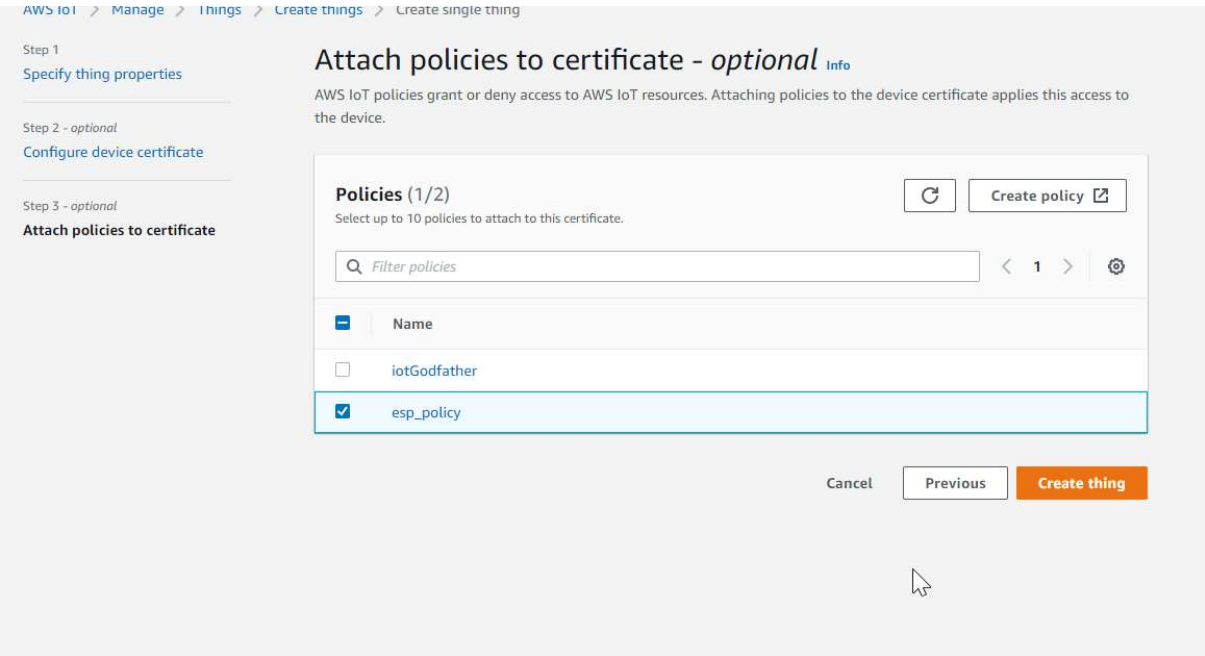


- Assign name and add policy actions as shown below

Active version: 11 Info

Policy effect	Policy action	Policy resource
Allow	iot:Connect	*
Allow	iot:Connect	arn:aws:iot:us-east-1:565585161358:client/esp_test8266
Allow	iot:Publish	*
Allow	iot:Publish	arn:aws:iot:us-east-1:565585161358:topic/esp8266/pub
Allow	iot:Publish	arn:aws:iot:us-east-1:565585161358:topic/$aws/things/esp_test8266/shadow/update
Allow	iot:Publish	arn:aws:iot:us-east-1:565585161358:topic/$aws/things/esp_test8266/shadow/get
Allow	iot:Receive	*
Allow	iot:Subscribe	*
Allow	iot:Subscribe	arn:aws:iot:us-east-1:565585161358:topicfilter/esp8266/sub
Allow	iot:Subscribe	arn:aws:iot:us-east-1:565585161358:topicfilter/$aws/things/esp_test8266/shadow/update/delta
Allow	iot:Subscribe	arn:aws:iot:us-east-1:565585161358:topicfilter/$aws/things/esp_test8266/shadow/get/accepted

- Note here the name esp_text8266 must be replaced with thing name which is created on previously which the name is documentThing



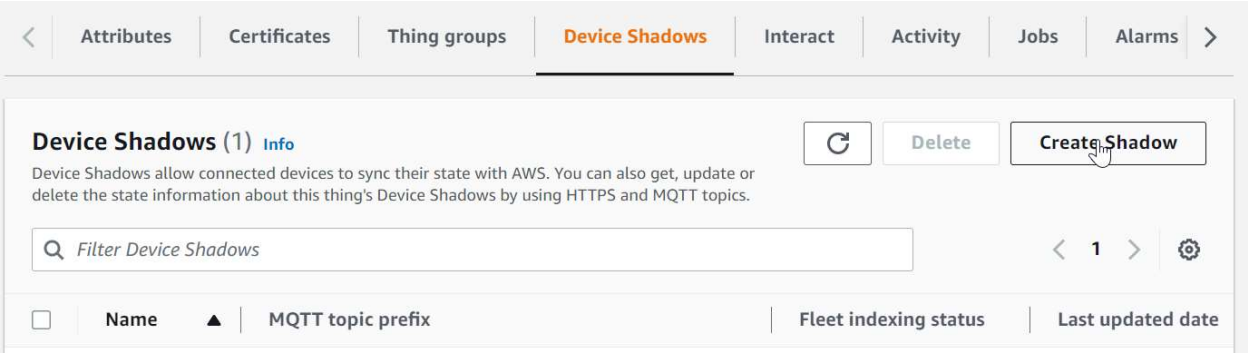
After creating the policy now select the policy and just press create thing

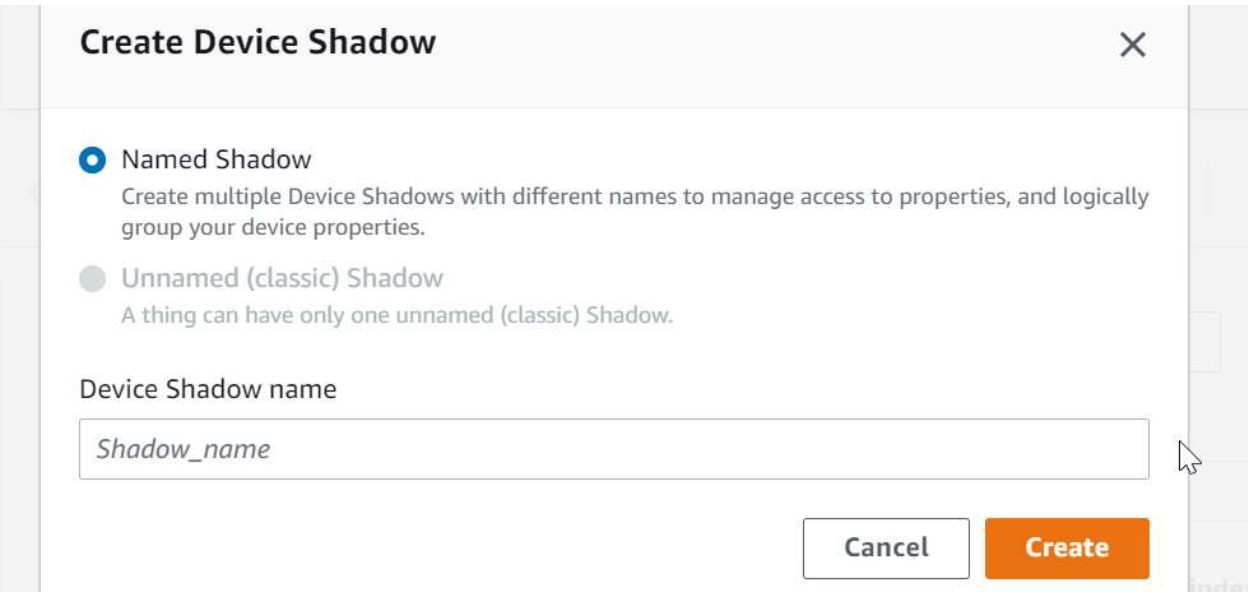
- Now download the certificates which are 1. Certificate 2. Public & private certificate 3. Rootca certificate

Save it all on the separate folder

For the detail information about the aws iot shoadow refer
https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-document.html

Before that ,go to iot things ➔ click the name ➔ click on shadows ➔ click on create things





now assign the data which as per require data

Request state document

A request state document has the following format:

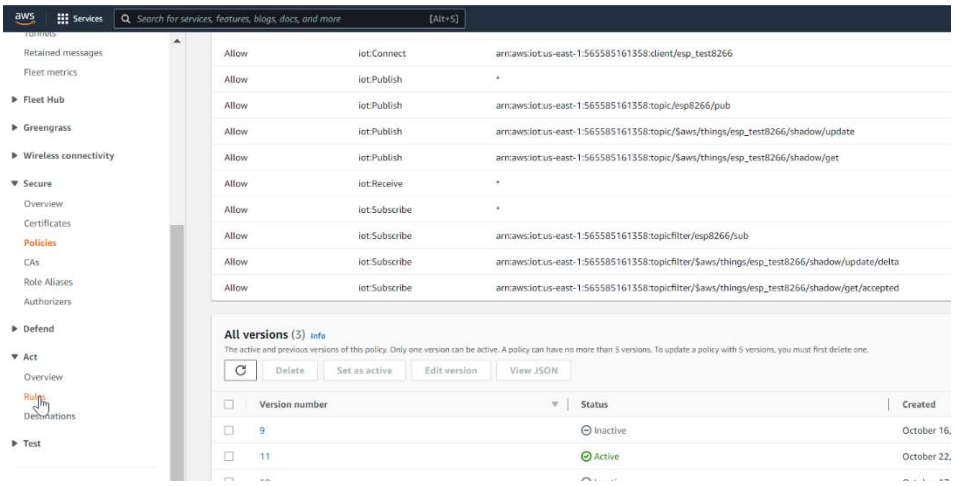
```
{
    "state": {
        "desired": {
            "attribute1": integer2,
            "attribute2": "string2",
            ...
            "attributeN": boolean2
        },
        "reported": {
            "attribute1": integer1,
            "attribute2": "string1",
            ...
            "attributeN": boolean1
        }
    },
    "clientToken": "token",
    "version": version
}
```

To configure the json file must refer the aws shadow document what is desired , reported states are

Where desired is the state which an iot device what to on/off to be in future , where reported state is the value already present in the present which would change or kept according to desired value

And then assign the name with unnamed . now

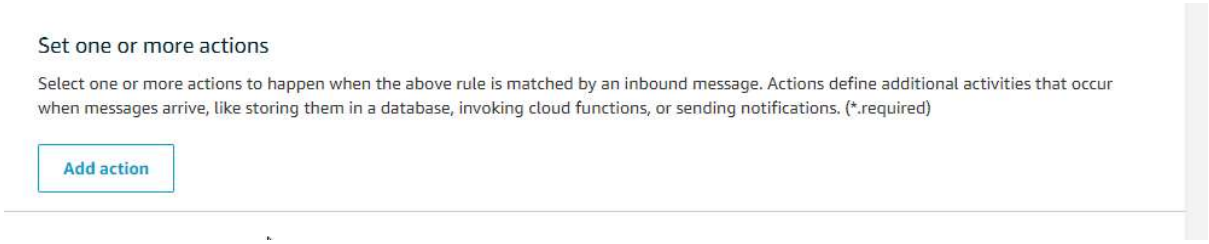
6. After setting the iot device to communicate with aws server in a particular topic now just create a rule to send the all data to store to dyanamodb
   Note: dyanomodb is another aws service which store the data, you can search on the aws search bar

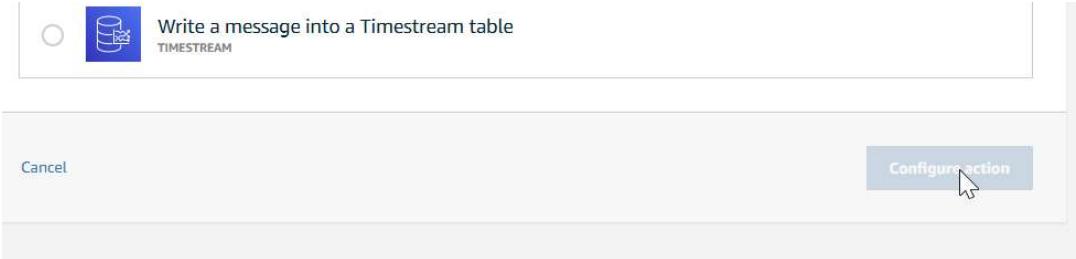


- Open rules, which would be any where as new interface in aws as present it is on message routes
- Now create rule with name which can be anything
- Now insert the sql command to select the all the data in the particular topic

```
SELECT * , timestamp() AS ts FROM 'esp8266/pub'
```

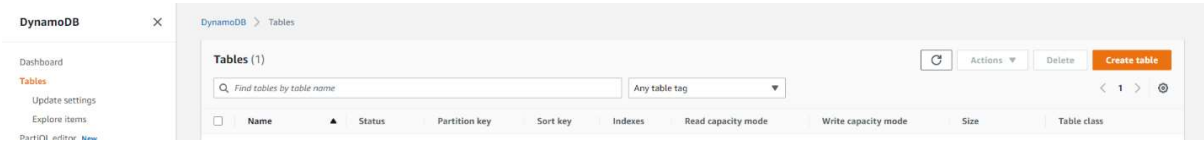
- Here note the topic must be the which we are sending the data
- Now add the action which is insert message to dynamodb as show on figure





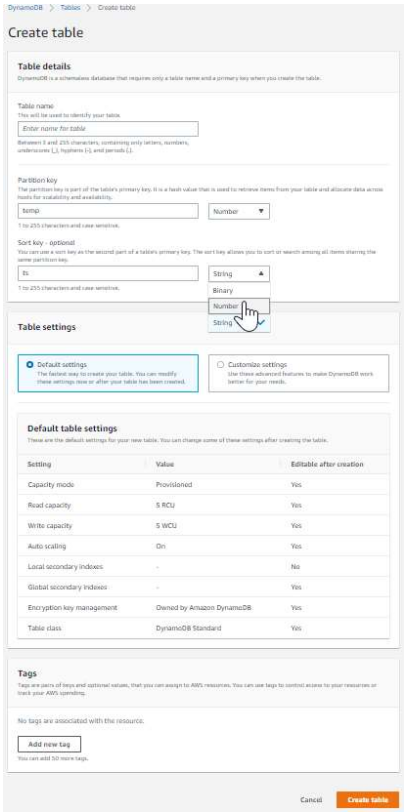
- Then click configure

7. Now configure the dynamodb to receive the data in a particular order
    - Now search for the dynamo db and go into it
    - Now on left side click tables



    - And next click on create tables give a name
    - Now important is that keep the **name as it is** which are actually sending from iot topic to dynamodb
        i. Partition key (which on table line)first data comes from iot topic as number
        ii. Sort key some other data iot topic second as number/string (which depends on data)
        iii. Select the default things that's it click the create thing

Now important last thing make sure every service is enable /on and then run it



8. Client code by using aws dynamodb CRUD code
    - https://github.com/charan-sai-v/aws-dynamodb-crud-operations-in-nodejs
    - Set the website code as the given data code via github

You can refer another github : https://github.com/debsahu/ESP-MQTT-AWS-IoT-Core/blob/master/doc/README.md

Board code with esp8266

Here we have to know about IoT devices and the way of coding to control ,

1. Get the code from the git hub and make sure the credentials download from aws IoT things certificates are present on the same folder or copy and past on secrets file
   Code: https://github.com/mahaboobtech/aws-Iot/tree/main/nodeMcuCode/aws_esp_pubsub

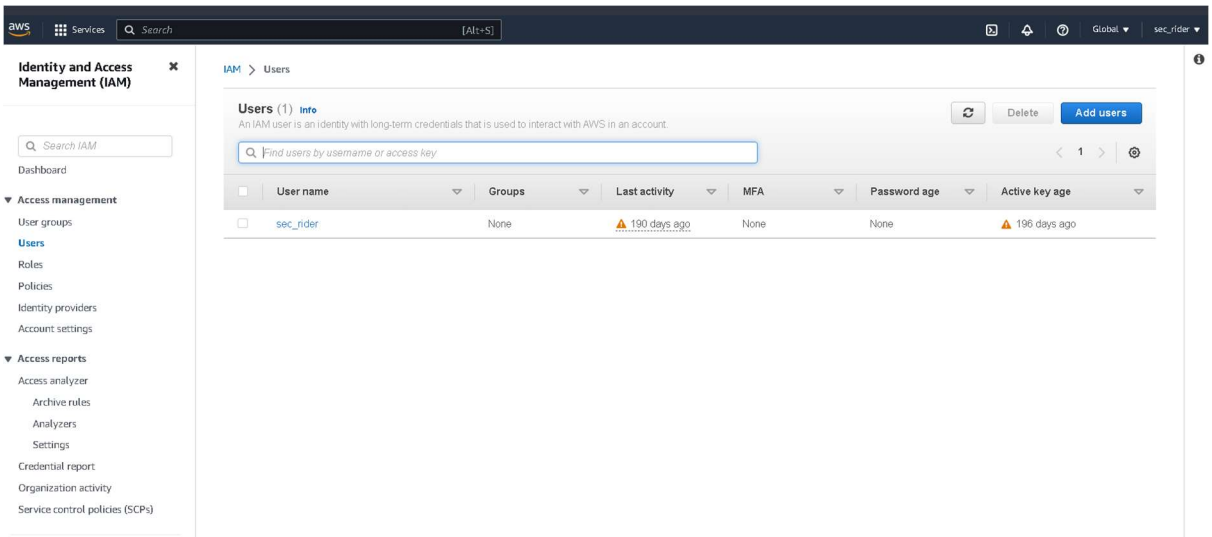
2. Now assign the part where the sensors or the attachments to be work or read the data
3. Now assign the data to a variable as mention as comment in the file and next set the topicname/line where to be publish and set the code to the read from the server as you know to control as choice your
4. Now after the i/o devices are perfectly configured and setted controlled by aws server now we just test the whether the board is perfectly connected or not you can use mqtt test client or aws dashboard of IoT core

Web based client code

Before that we need nodejs software which is free

First we need permission to access the dynamodb data from aws server for that

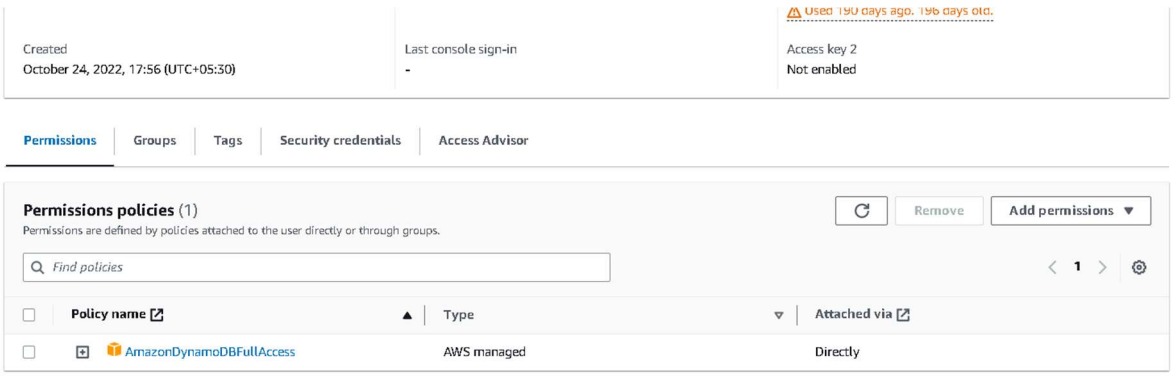
Search Iam and go:



Click on add users and assign name

Next select attach policies directly & next select amazonDynamoDBFullAcess click next

Now create …… that’s it



Now for access search credentials on **setting** and **security credentials** tab

Get that credentials on a folder

Now here the code part begins :

Go take the github : code now use

1. Client code by using aws dynamodb CRUD code
    - https://github.com/charan-sai-v/aws-dynamodb-crud-operations-in-nodejs
    - Set the website code as the given data code via github

You can refer another github : https://github.com/debsahu/ESP-MQTT-AWS-IoT-Core/blob/master/doc/README.md

Assign the credentials on given secrets file by copy pasting or assign the credentials file path

Html code part

And on html file set the graph and location where the data controlled or visualize by according code and javascript where every comment is present only thing you need to understand code and interface it perfectly with the credentials

Here the credentials taken from I am service which as newly created user by that user access we are able to see and control the data

Future scope

On future we can add forethere control of sending message to esp8266 board and we can perform the task what we need to control

And also we can use AI to control the IoT things which perform automatically as in the one of aws service

References

[1] https://github.com/charan-sai-v/aws-dynamodb-crud-operations-in-nodejs
[2] https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-document.html
[3] https://docs.aws.amazon.com/iot/latest/developerguide/iot-device-shadows.html
[4] https://docs.aws.amazon.com/iot/index.html
[5] https://docs.aws.amazon.com/iot/latest/developerguide/iot-policies.html
[6] https://docs.aws.amazon.com/iot/latest/developerguide/pub-sub-policy.html
[7] https://www.youtube.com/watch?v=28FS2qix2u4