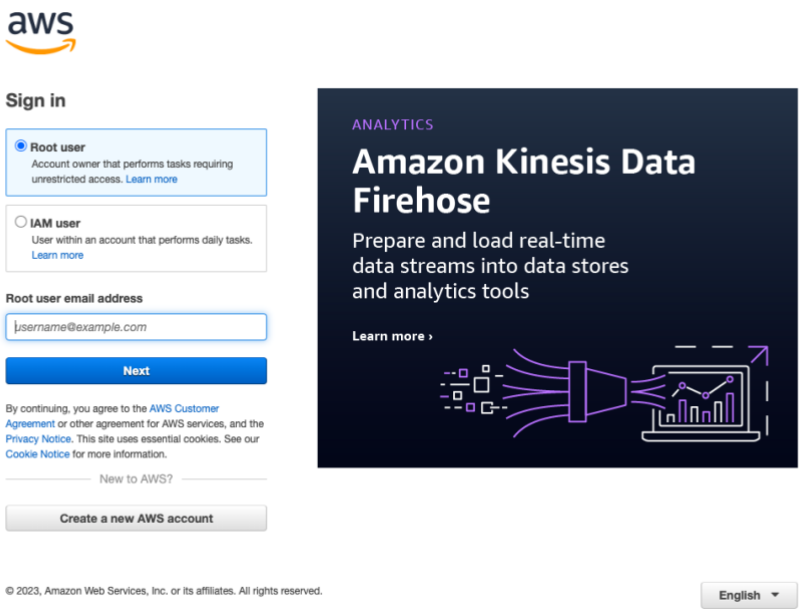
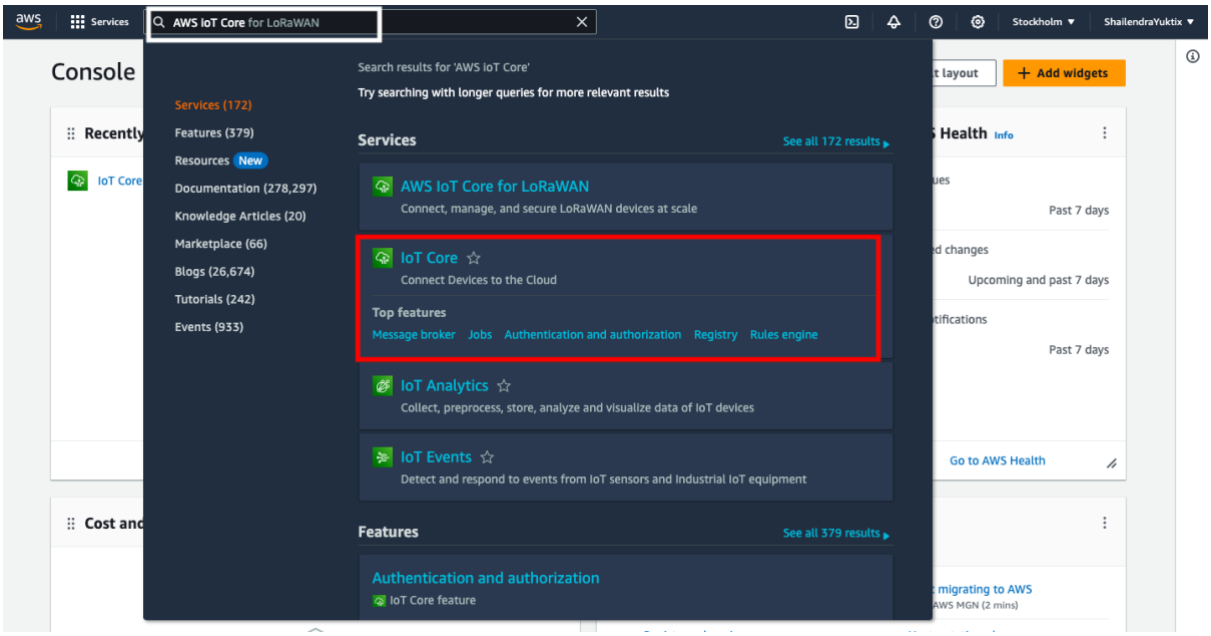
Account Creation in AWS

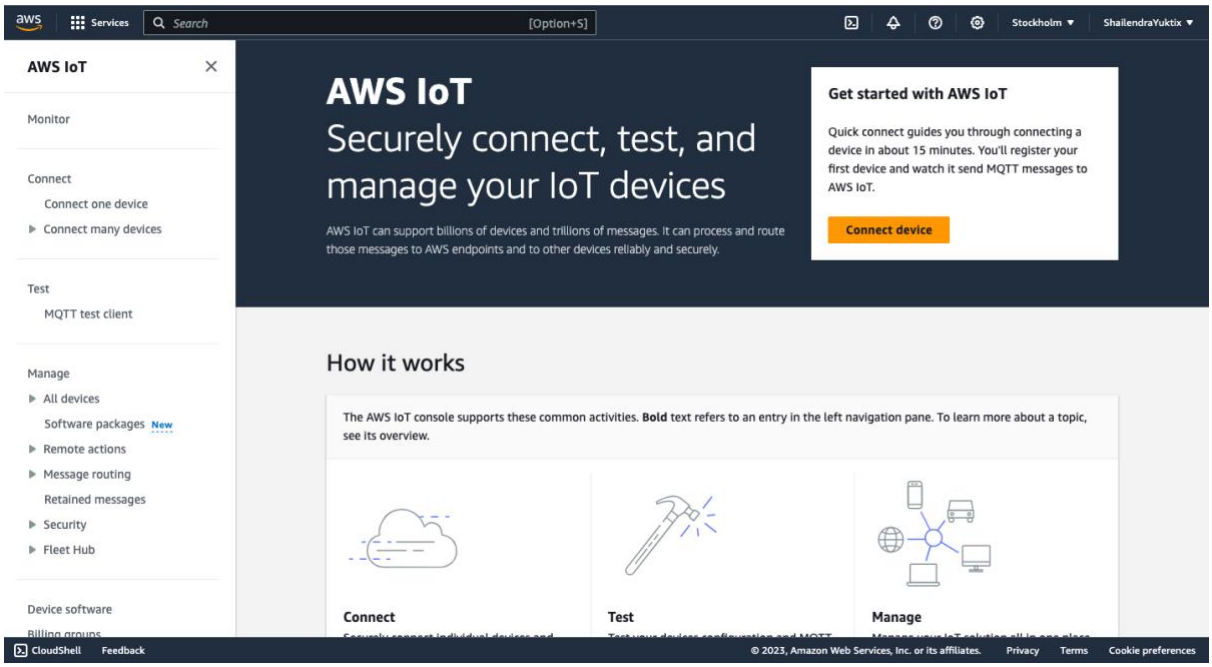
1. First of all you have to create an account in AWS. The account is free for the first year and it will need your Credit card for the purpose of verification.
2. As we are interested in AWS IoT Core, after verification and account creation, sign in to the AWS console as a root user.



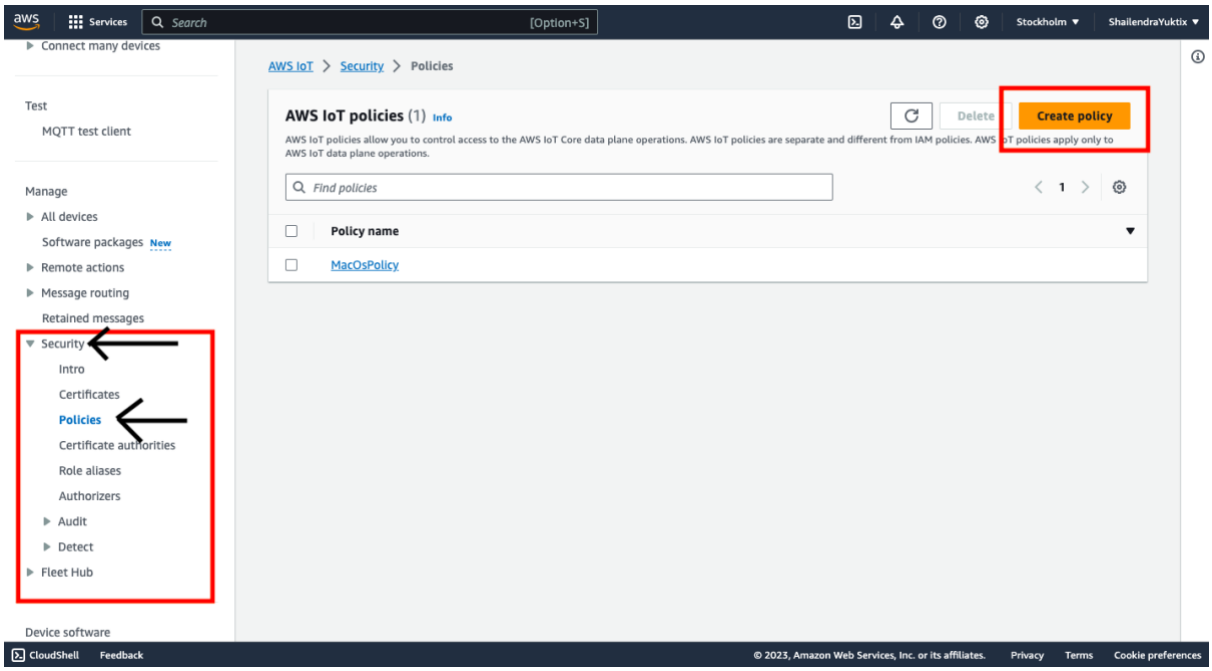
3. Put in your email id and password.
4. After the login, you will see AWS console. In the search bar, search for AWS IOT Core and select IOT Core from the drop down.



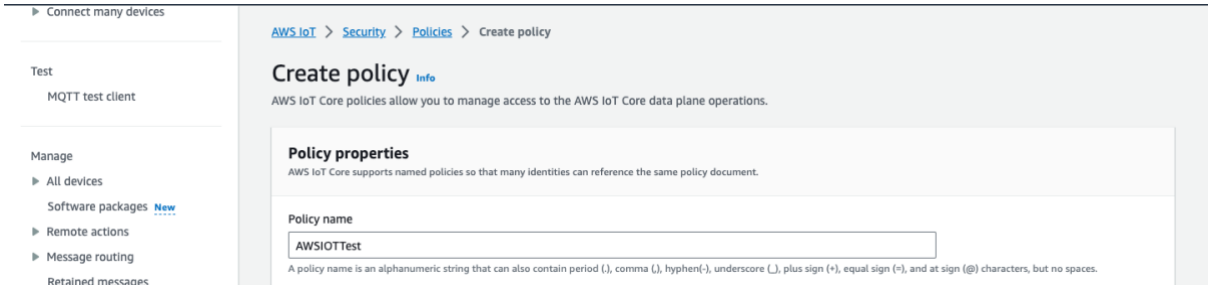
5. Once you click on IoT Core, you would be redirected to IoT core dashboard.



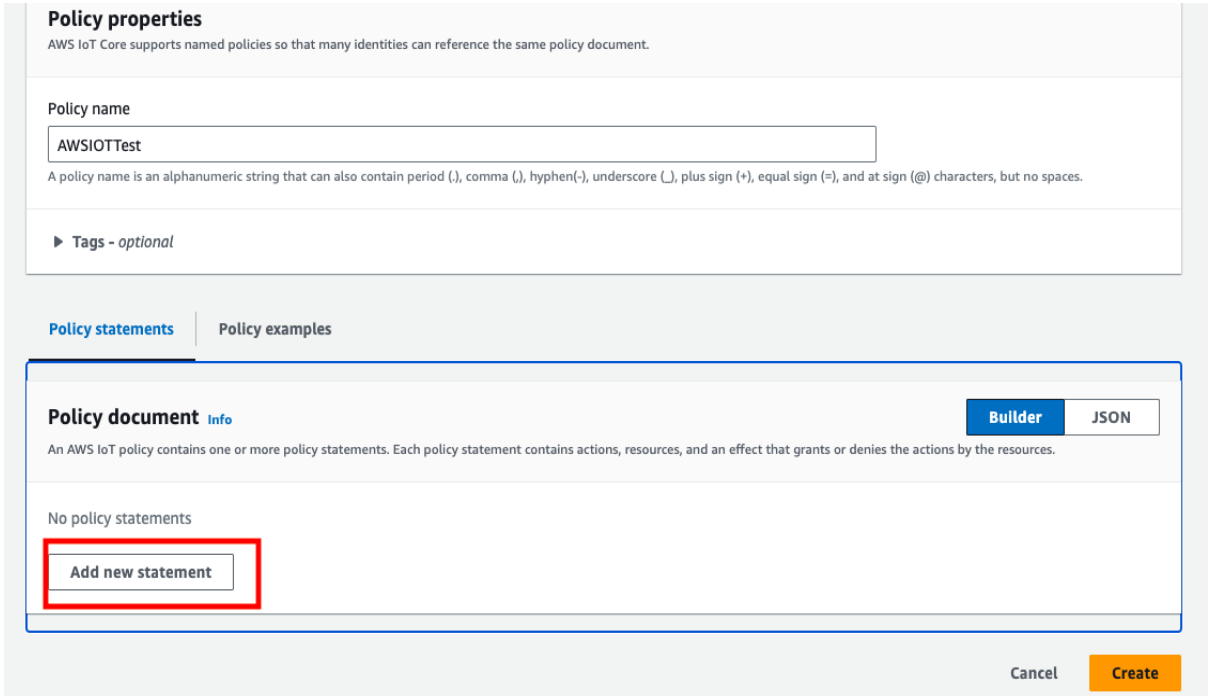
6. The first thing, we are going to do here is create a security policy that we are going to attach to the device that we will create. The policy is created to make sure that we give sufficient right to the device to send the data to the IoT core.
7. On the left hand side menu, click on **security** and then click on **Policies.**



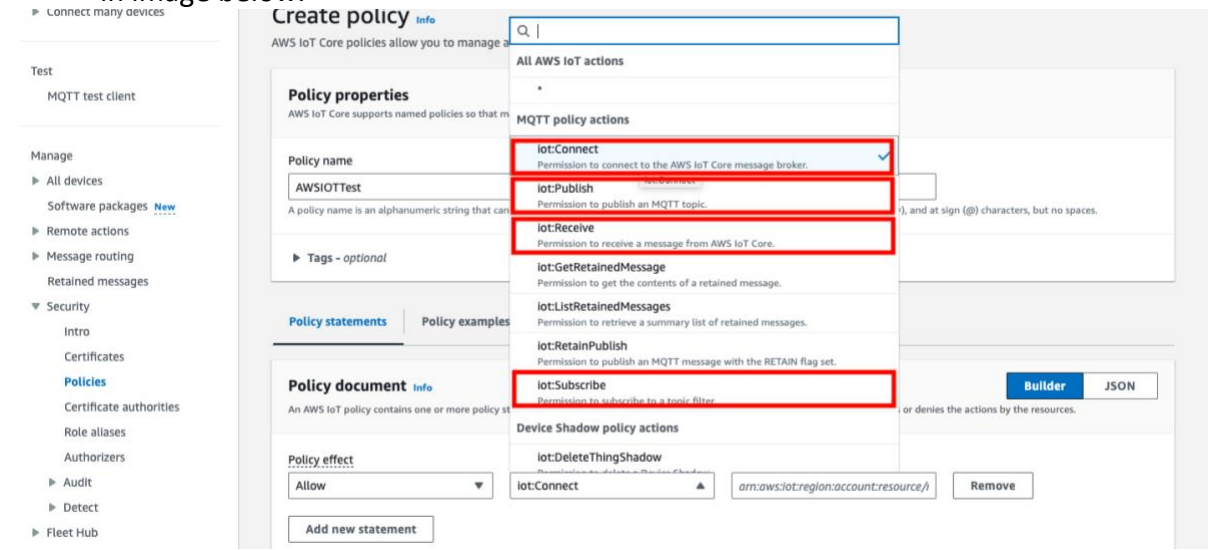
8. The click on Create Policy button on the top, right corner.
9. Once you click on create policy, it will ask to provide a policy name. Use any human readable name. Remember, this policy, we would have to attach to the device that we will create down the line.



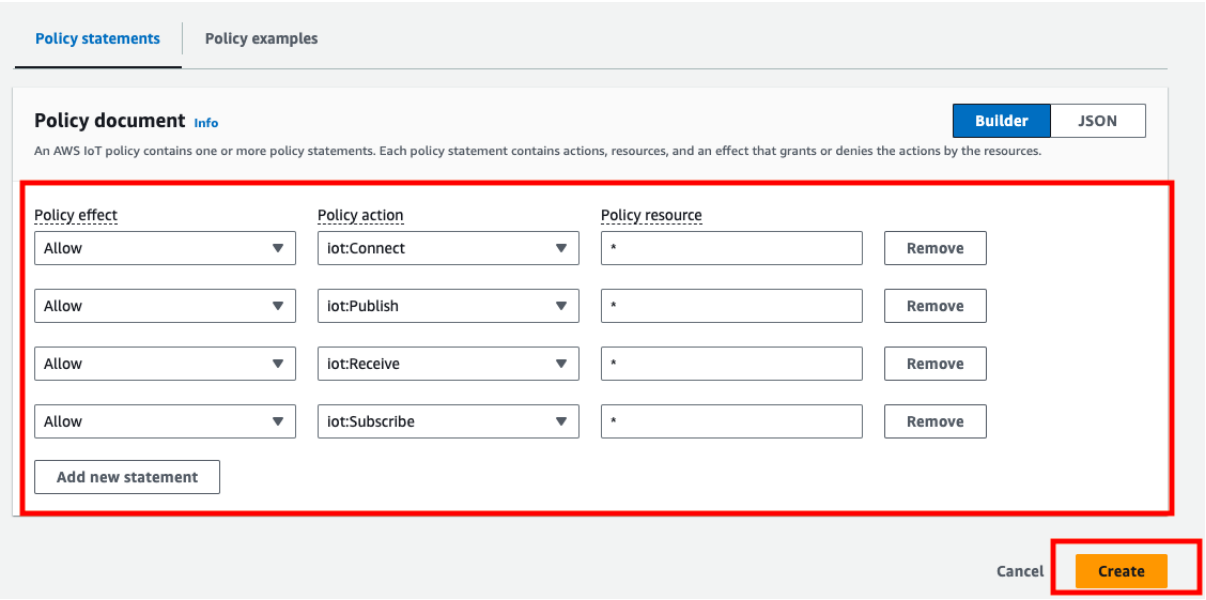
10. After the Policy name, scroll down. In the policy document, click on Add new statement.



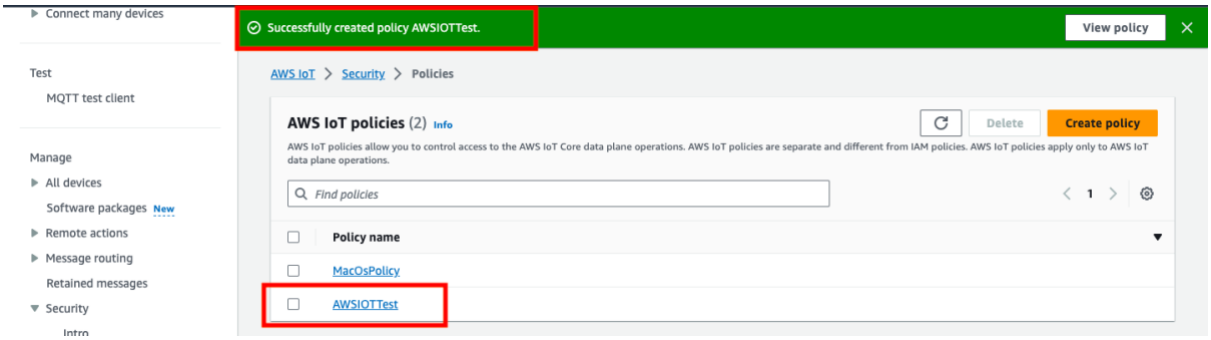
11. Now after clicking on create new statement, select Allow under Policy Effect. Then click on arrow under the Policy action and select IoT connect. Under the Policy resource, put *.
12. Repeat step 11, 4 time, till you have chosen all the highlighted Policy actions as show in image below.



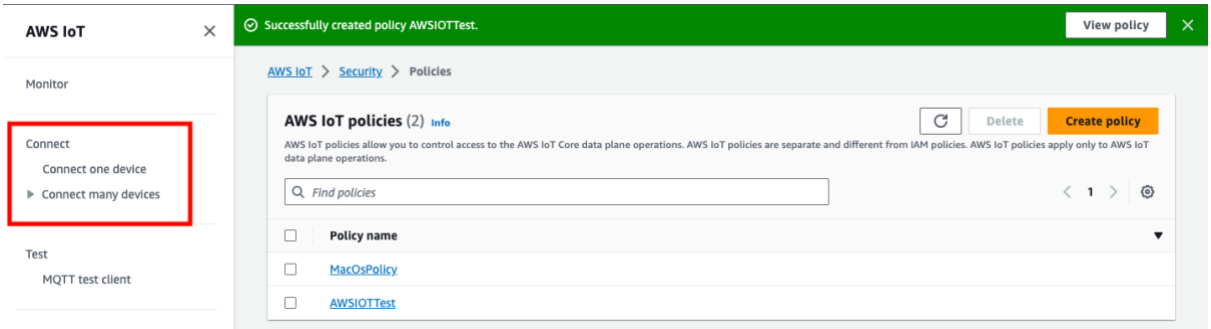
13. Please make sure that once you choose all the 4 Policy effect, Policy action and put in a * in Policy resource, your current screen look like below image. Once verified, click on **create**.



14. You will see your newly created policy in the next page.

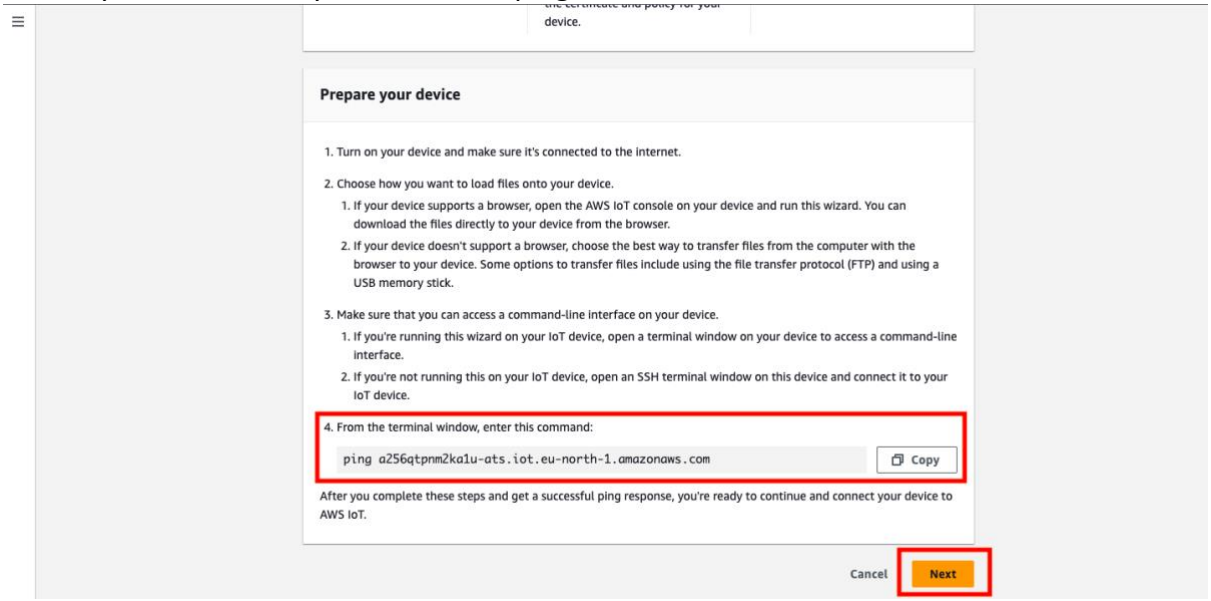


15. Now next step would be to create, one device (there is an option to create many devices and attach the same policy to all of them, but we will start with just 1 device. On the left hand side, click on **Connect one device.**

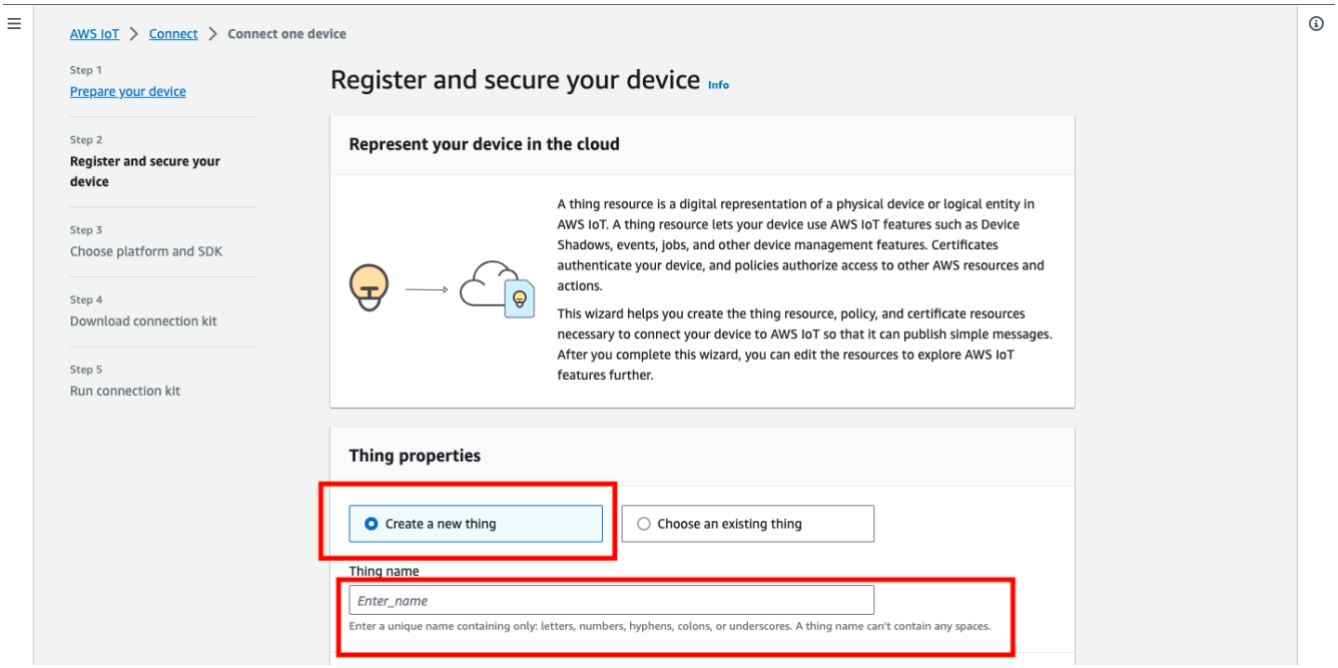


16. Once, you click on Connect one device, a new page will open, follow the steps mentioned there. Just to test, if your device is able to communicate with IoT core end point, in the bottom of the page, use the command to use the **PING** Test. You

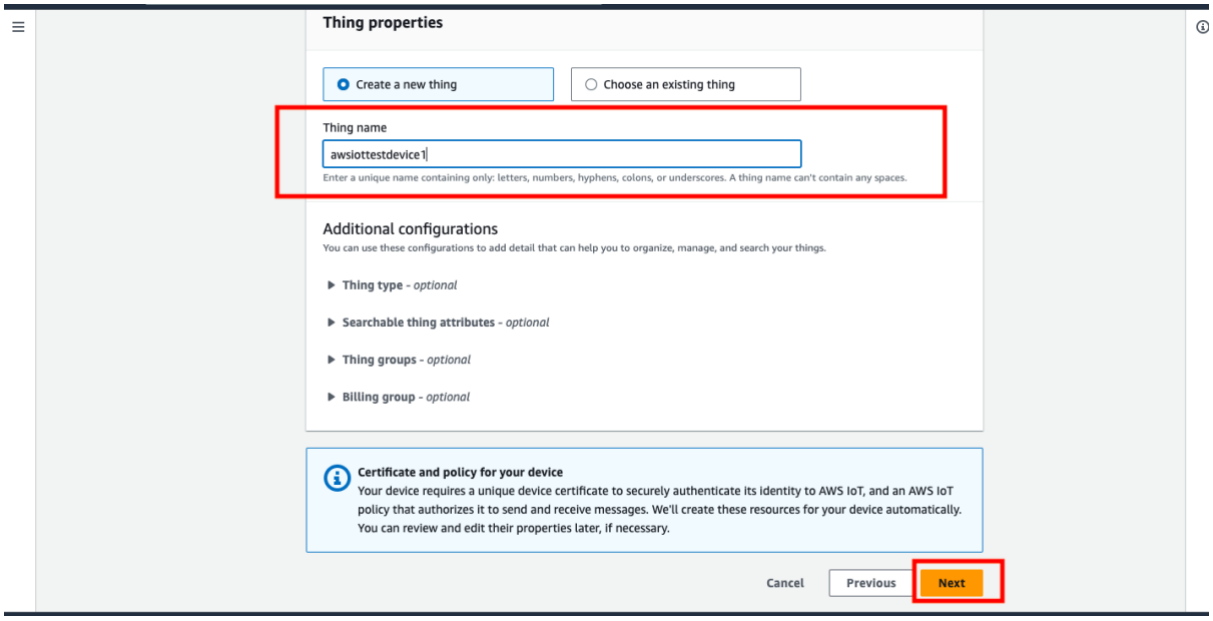
can run the command on the terminal (if you are using RPI or any other SBC). Once you do that and you are able to ping, click on **Next**



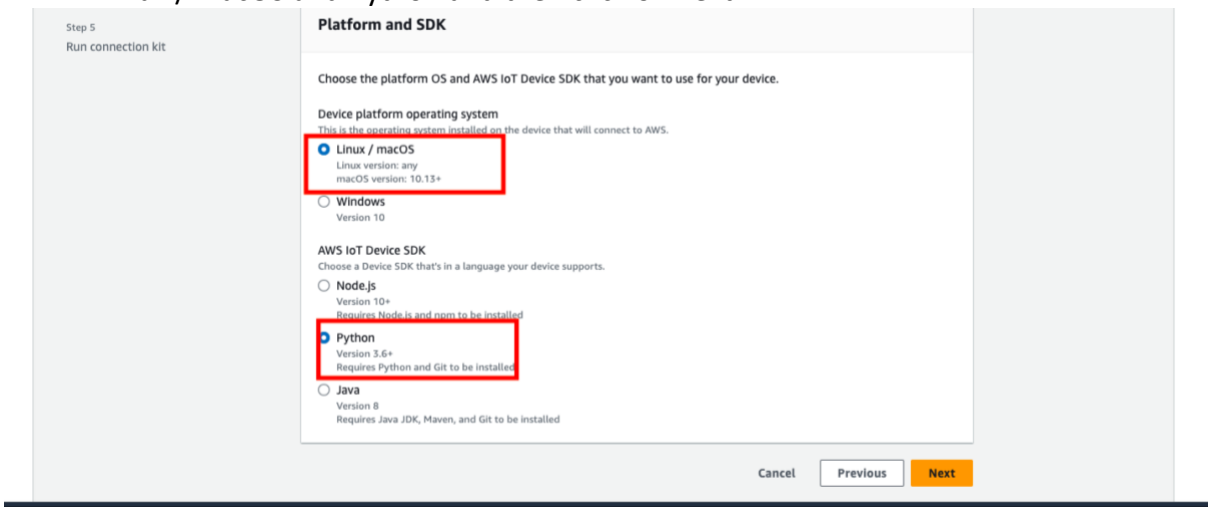
17. On the next screen, select **Create a New thing** Or you can choose the existing thing. Thing here is the device. Once you choose, Create a New thing, provide the thing name. Make sure, you follow all the rules of naming convention.



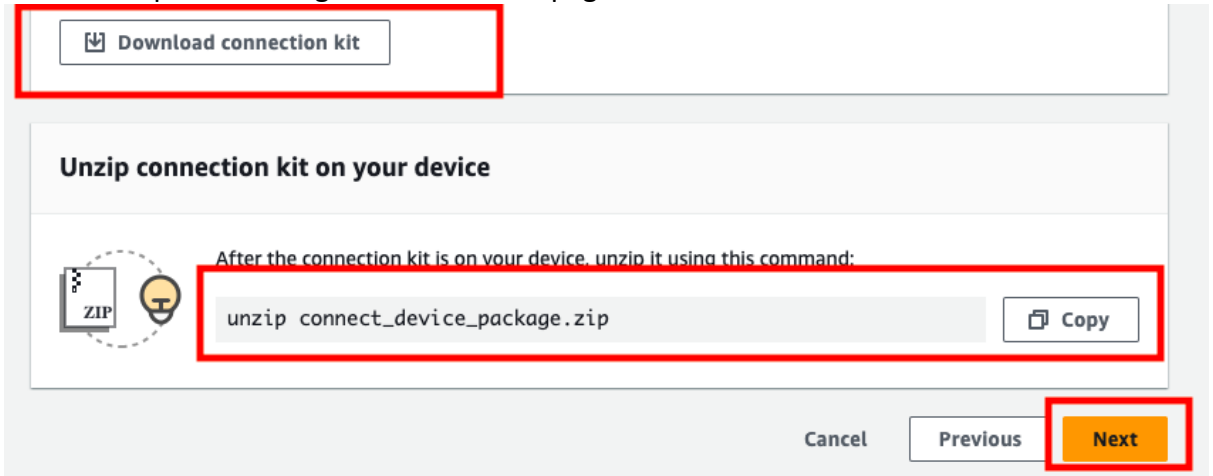
18. Leave all the other options and click on Next.



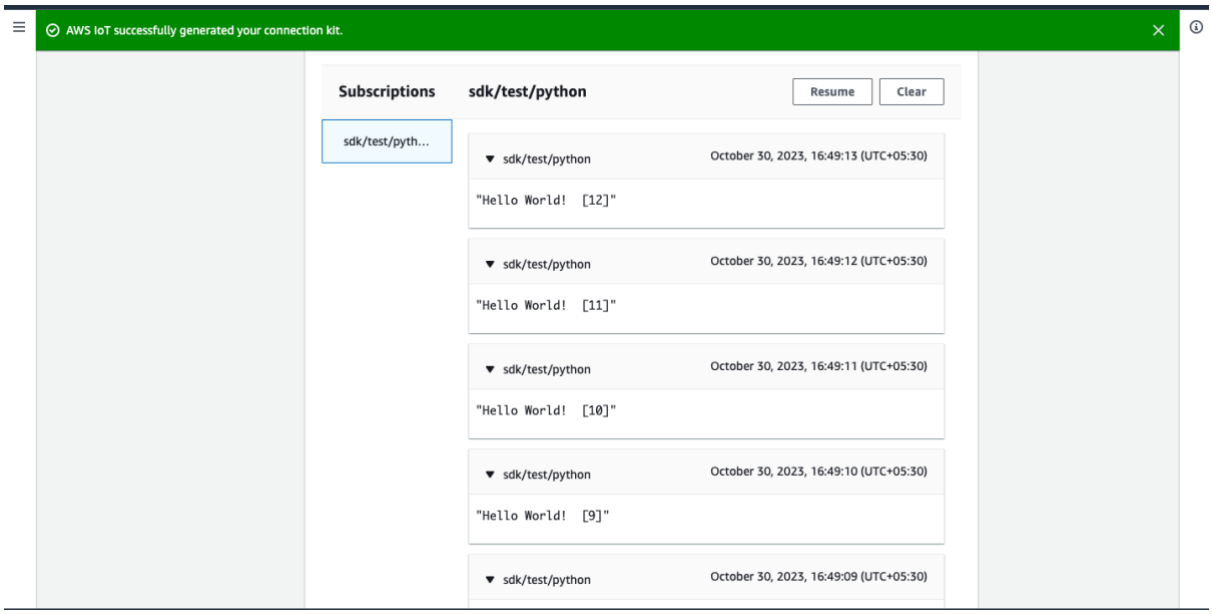
19. In the next screen, you would be asked to choose the OS and Language. Choose Linux / macOS and Python and then click on next.



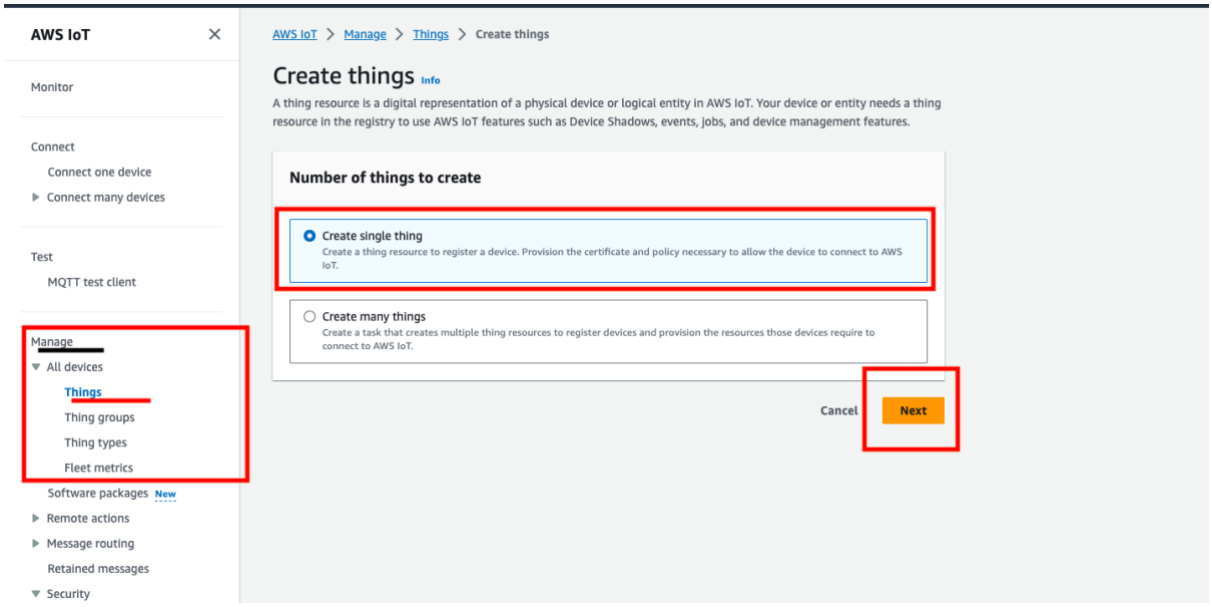
20. In the next screen, download the connection kit. Once downloaded, unzip using the unzip command given in the same page. Then click on next.



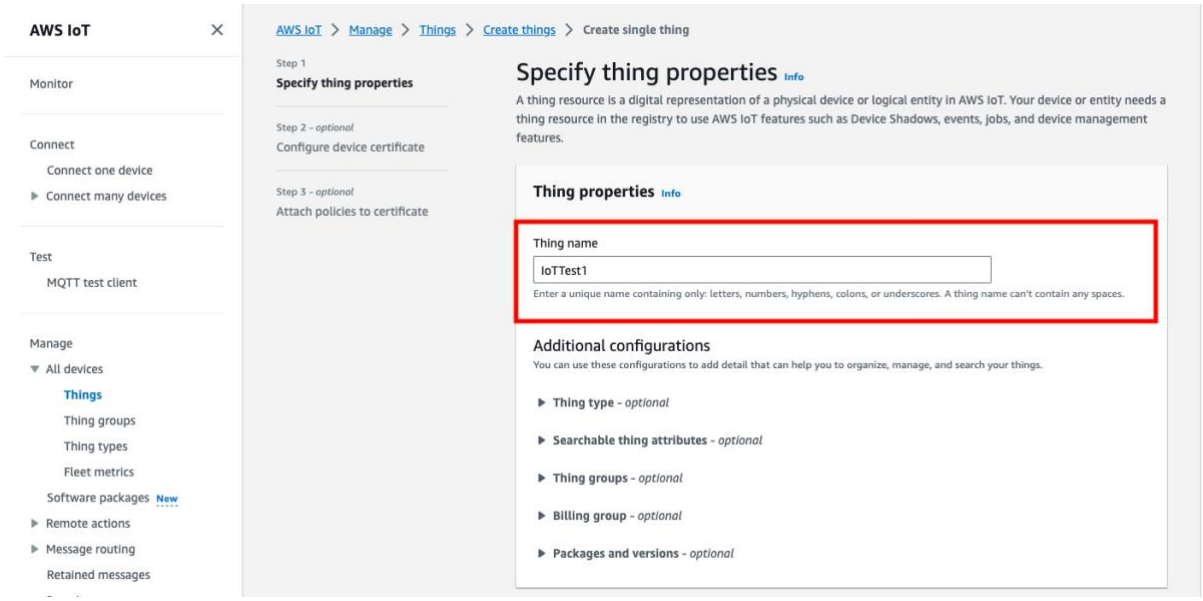
21. Once you click on next, follow the steps given in page to run the start.sh file.
22. Once you run the file, if you see the messages posted from the device on this page, that means, you are able to successfully connect a device to IoT core.

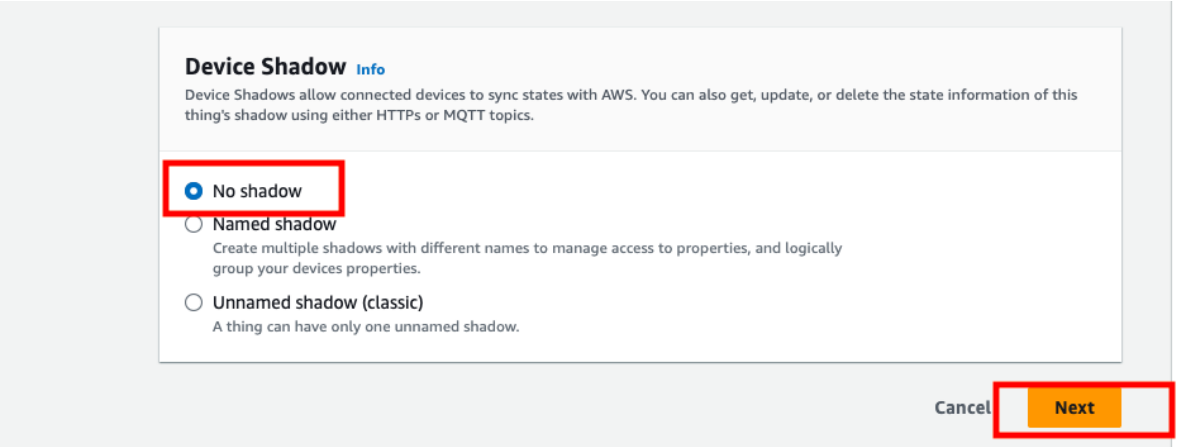


23. Now as you are able to connect one thing (for test purpose), now lets **Create one thing.** Click on the left hand side menu option on the top (three horizontal lines). And then Under Manage, click on Things and a new screen will come. Now select create Single thing and then click on next

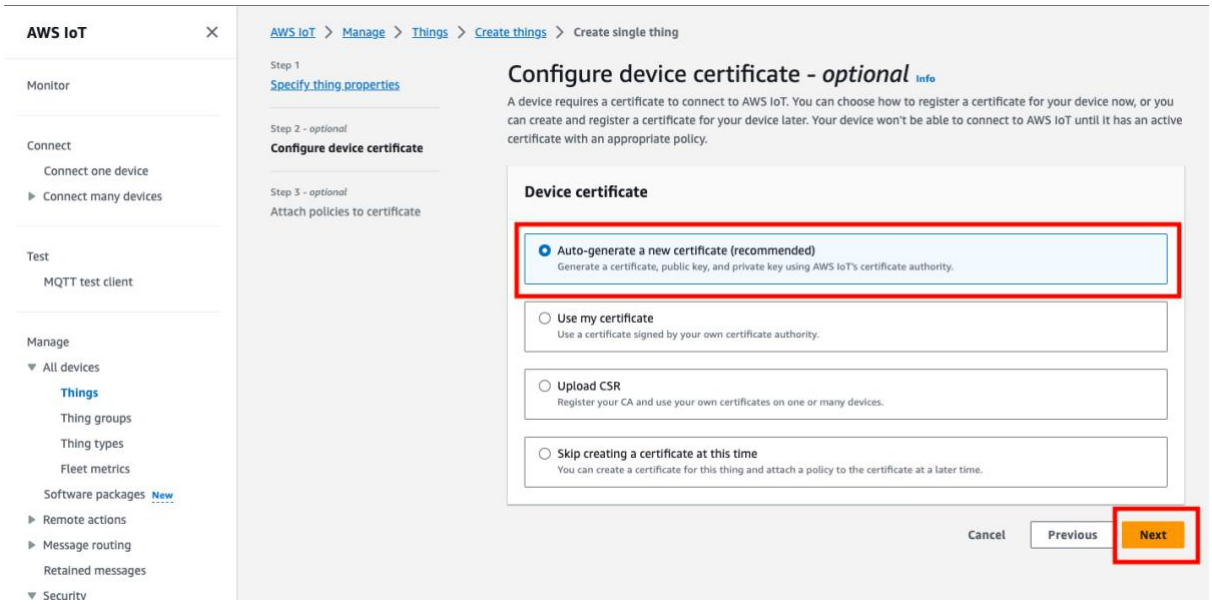


24. In next screen, under thing properties provide a name to your thing, below that, select no shadow (default) and then click on next.

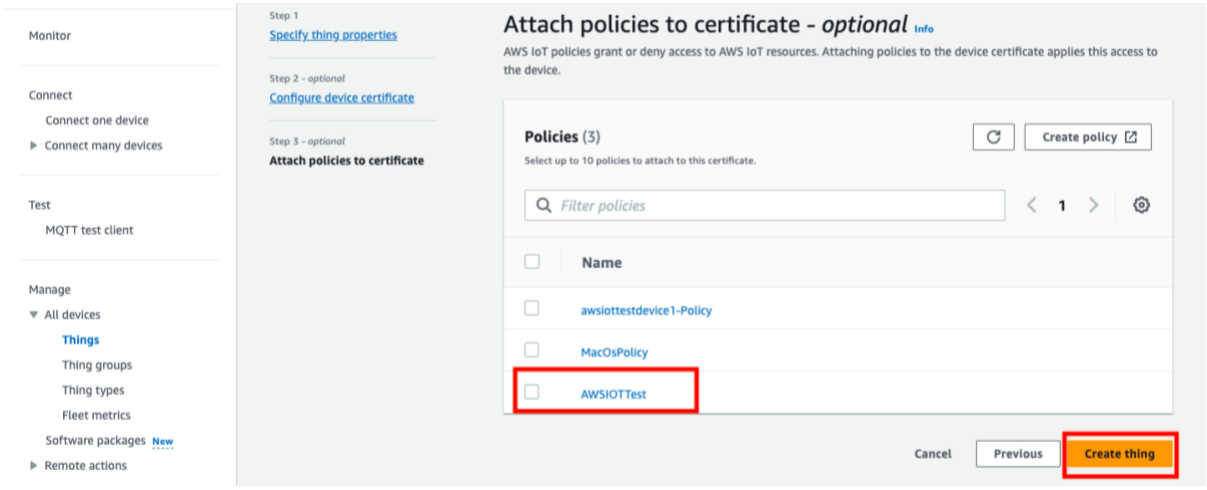




25. In the new screen, go with default option – Auto Generate a new certificate and click on next.

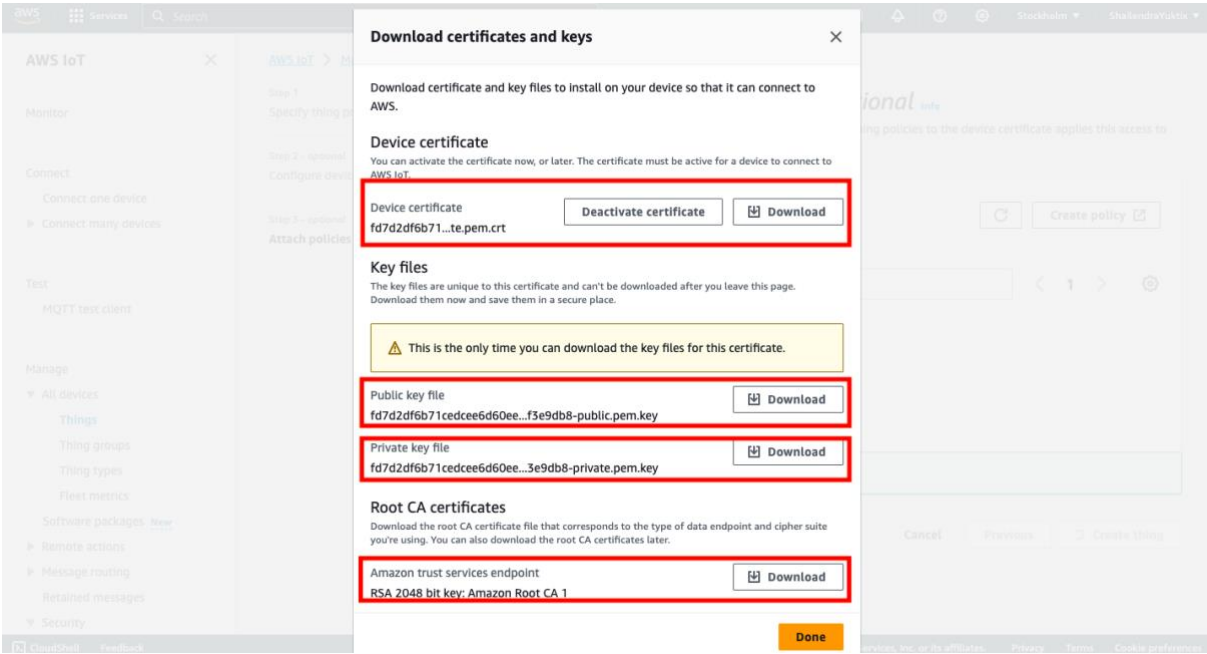


26. In the next screen, select the policy that you created in step 14 and click on **create thing.**



27. Now in the next screen, you would be show certain certificates. Please note, that you have to download all of them and rename them as show below (for ease of use in the code). These certificates will be in the same directory as that of our code.

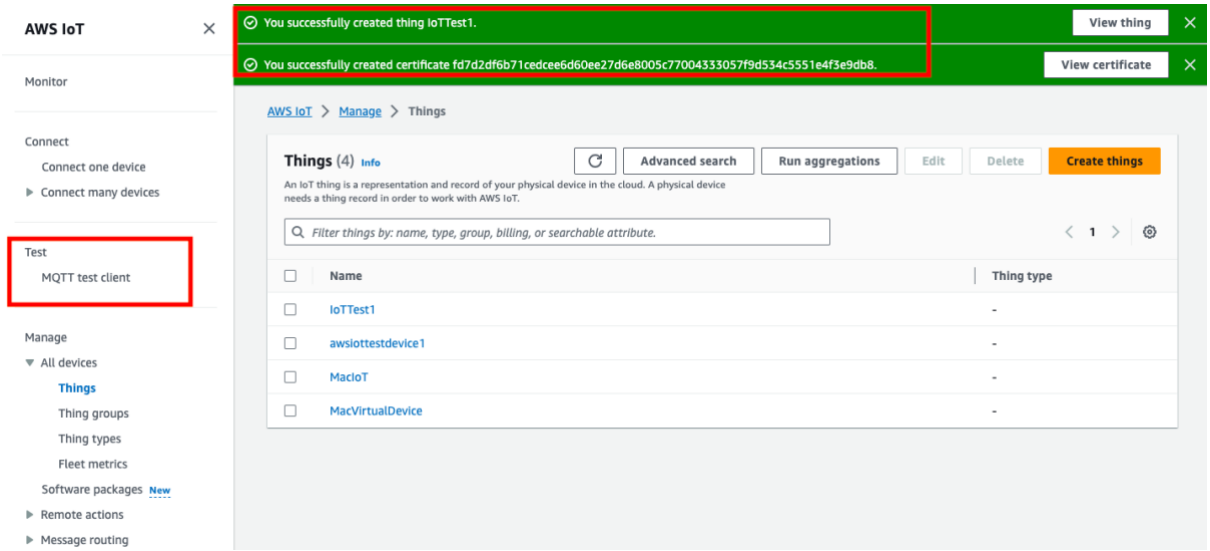
Certificate File Names	
File	File Path and name
Private Key	private.pem.key
Public Key	Not required for now (public.pem.key)
Device Certificate	Device.pem.cert
Root CA certificate	Amazon-root-CA-1.pem



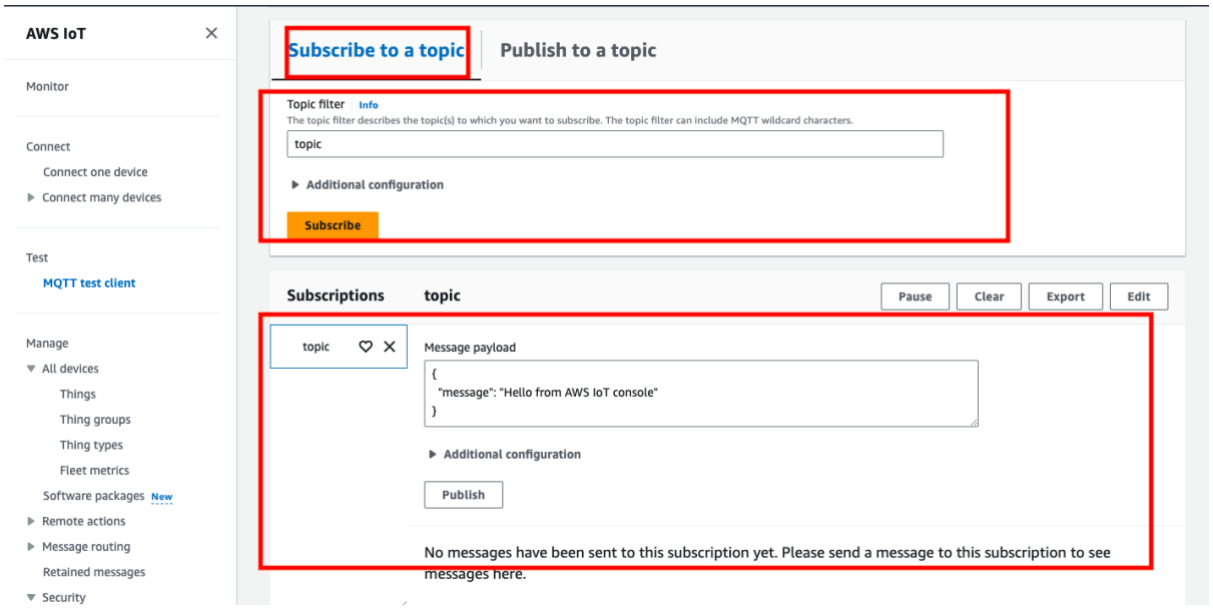
28. In case if you are using any Linux system or macOS for testing purpose, better is to create a virtual environment so that multiple version of Python mess with your code.
29. I am using Python3.7 for this project.
30. Once you run the code and you have the right certificates in place. You should see following in the your linux or RPI machine.

```
(awsiot) Shailendras-MacBook-Air:AWSIoTCore shailendra0408$ python3 TempSensor.py
26.375310839752338
{"message":"OK","traceId":"4061a0bc-7c16-9282-72b9-bb8c64c0b333"}27.39171710081452
{"message":"OK","traceId":"102c3f6c-79a6-f021-9bda-3d2367004a8c"}32.811024725857195
{"message":"OK","traceId":"805a57af-ccbc-d275-baf9-170a44fd7222"}20.969604349927092
{"message":"OK","traceId":"75bcce2a-7380-59a8-0e3f-05876fe2755b"}34.50321472313371
{"message":"OK","traceId":"29c90731-4c8c-c8a5-8d29-bda5bbe801b8"}32.43374653323447
{"message":"OK","traceId":"80d5922a-1dce-0e7e-45c3-d868fbc27393"}22.749645855866994
{"message":"OK","traceId":"3428af6c-7da5-ed8d-1834-d6a117cda26e"}30.855992526200556
{"message":"OK","traceId":"5eababdc-4859-8048-2515-b5b570b1d485"}25.243679186571857
{"message":"OK","traceId":"5cb7e288-202e-1bc2-ad24-930ae428bca6"}22.39276796251556
{"message":"OK","traceId":"265d6c89-932f-ae42-5b28-c3c94a6f29b8"}33.589881140721495
{"message":"OK","traceId":"8968c392-7b22-2bdc-178d-61fea51256a9"}33.309791099863595
{"message":"OK","traceId":"73ff8c45-3915-4798-193b-9529b4c9f005"}21.221091338618866
{"message":"OK","traceId":"db7be886-b8bb-d44c-01f3-de95d3a70829"}29.755330807414396
{"message":"OK","traceId":"f0f5993a-5fd4-de6d-2fdb-af6049deb6e8"}34.7000065801024
{"message":"OK","traceId":"3bea3502-6e3a-6272-589e-253690c0afaa"}28.59896154280363
{"message":"OK","traceId":"30364669-4ebc-a93c-d3c3-64d47668003f"}27.404390310426766
{"message":"OK","traceId":"93ae48b5-e65d-7060-eafd-b87206758ecb"}34.63134995607386
{"message":"OK","traceId":"5604a026-b4e5-9a94-3cbe-c2112215f900"}23.895844306461623
{"message":"OK","traceId":"23db2412-9465-39bb-a15f-c4602bace6a7"}31.913427585151048
{"message":"OK","traceId":"6eb140ed-e6ed-7aaa-f434-ce2d38fd9903"}23.583658896674855
{"message":"OK","traceId":"2a1e26a8-1e13-80d4-244b-d8001064beb7"}26.358638656012907
{"message":"OK","traceId":"92e8bc29-87b8-13c0-af85-828b4f4fde75"}22.958987945676206
{"message":"OK","traceId":"c6b3b4da-71a1-a0e2-0612-54ce828eeaa1"}25.276902262207425
{"message":"OK","traceId":"af940cdc-a799-312b-33db-0b8932f32844"}20.25925683147578
{"message":"OK","traceId":"53a328dd-c6bf-746f-ba82-a0c8396b727d"}31.010885717022305
{"message":"OK","traceId":"29216b75-383d-e462-3fc6-c50fe6ed0253"}
```

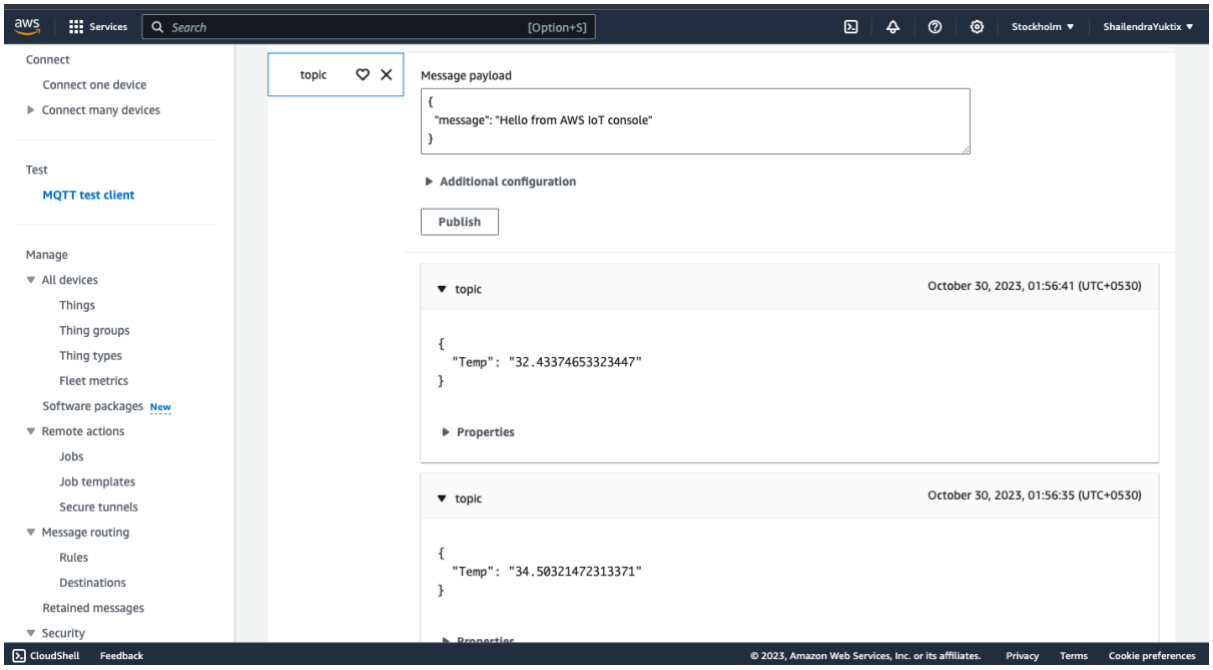
31. On the AWS IoT Core, on the left hand side menu, search for Test and then below that MQTT test client (in this, we are going to use HTTPS to send MQTT message to a kind of broker over a topic).



32. Once you click on the MQTT test client, in the new page, under Subscribe to a topic, enter the name of the topic you are subscribing to. For now, write **topic** and then click on subscribe.



33. Once you subscribe, you will start receiving data from the device or Linux machine or macOS.



34. Now next step is to create a rule for this device, which will invoke a Lambda function. That Lambda function will write this IoT device data into AWS Timestream.