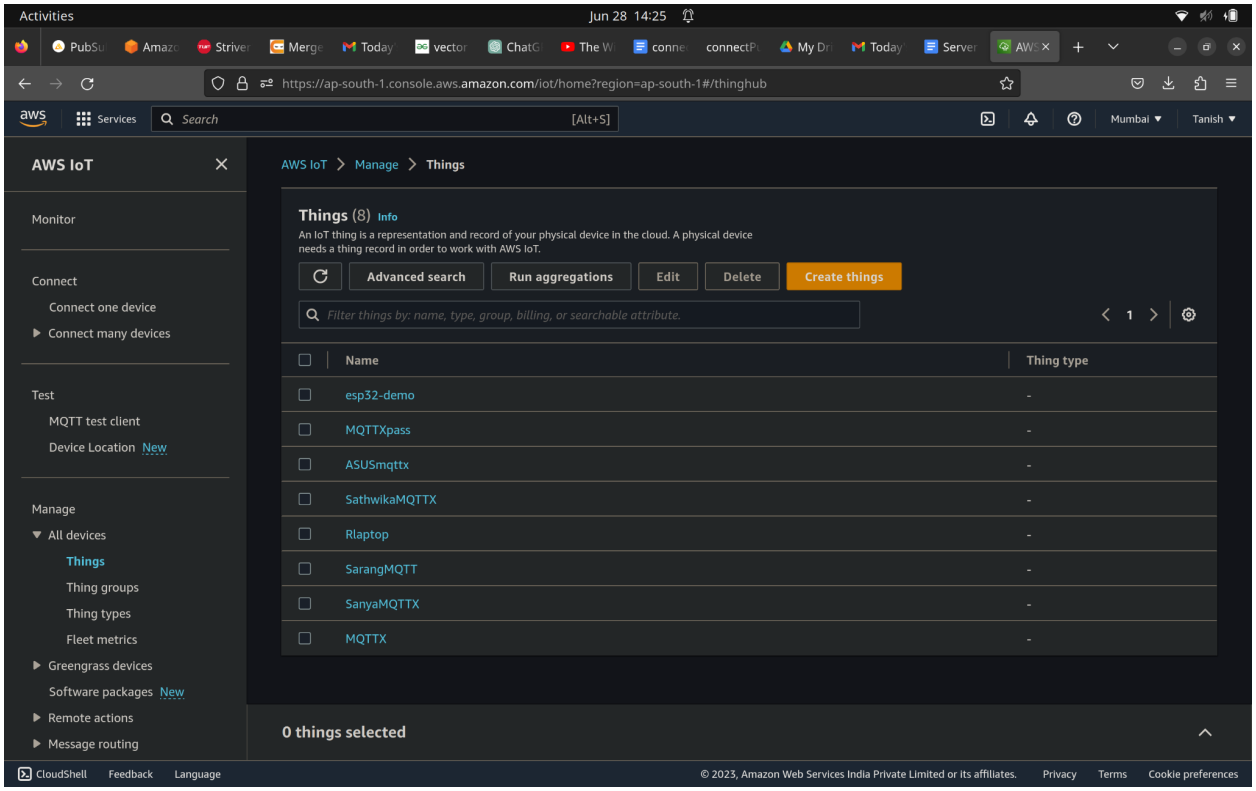
AWS IoT core is required to send and receive MQTT messages.
All other services are optional from a pure communication POV.

A comprehensive video describing the MQTT connection between esp32 and AWS IoT Core:
[Connecting ESP32 to AWS IoT Core - YouTube]

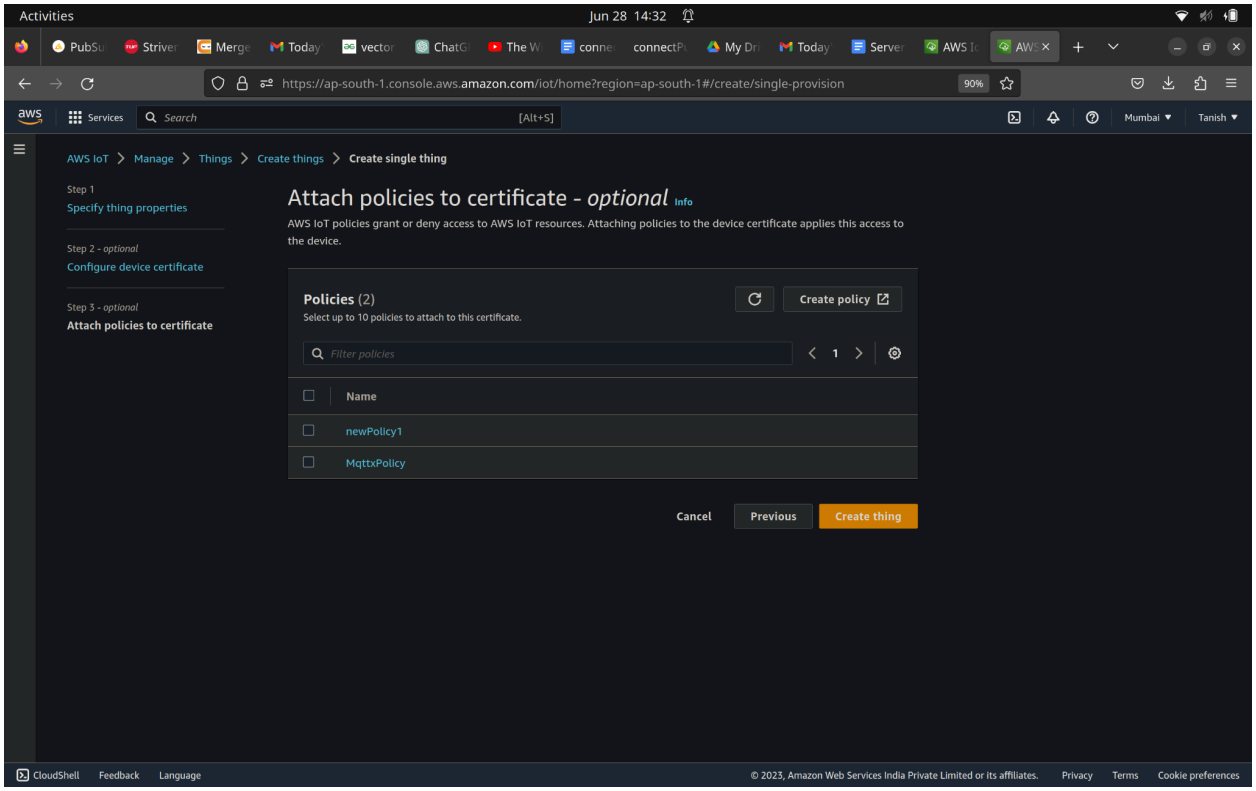
AWS server side:

Following is documentation on creating a thing in AWS to get the required device authentication certificates. These certificates are essential to connect AWS IoT Core and your device.

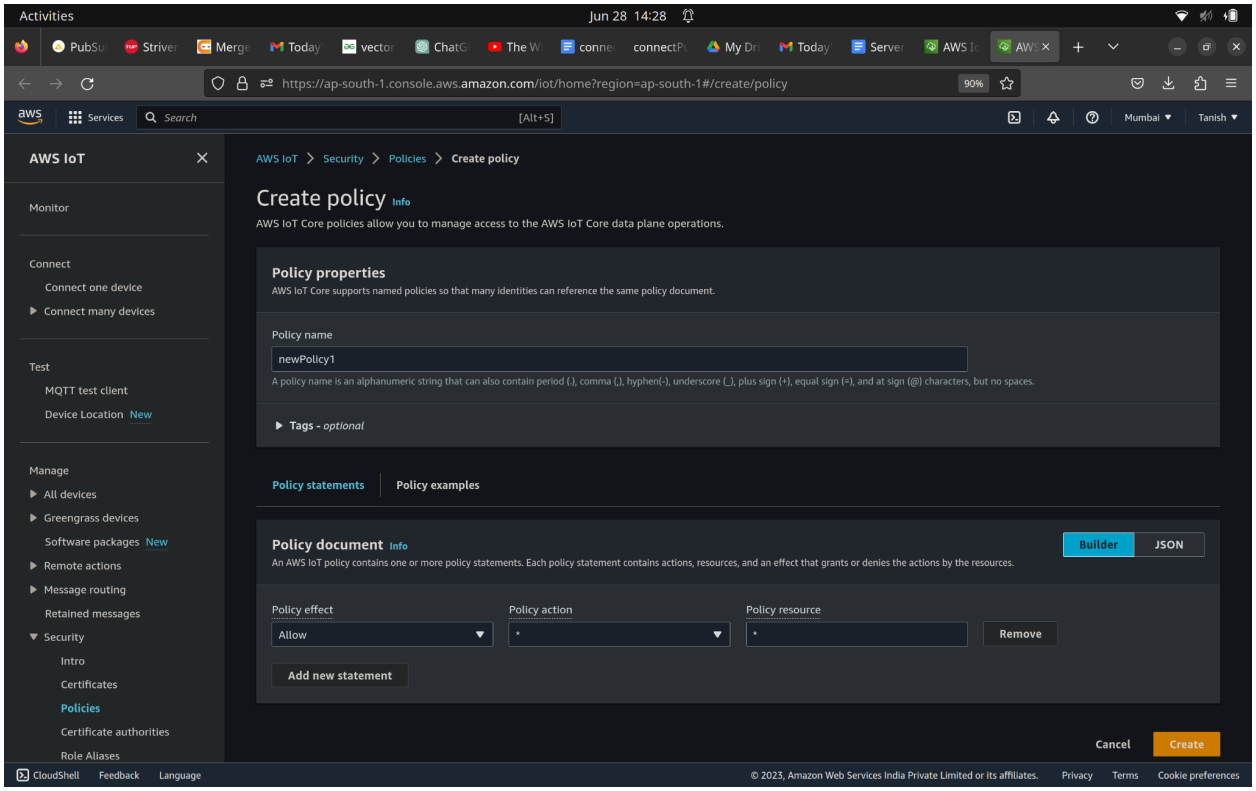


This is the things section of the AWS IoT core service. We will be required to create a new thing to generate the certificates. Click on the “Create things” button and follow along with the

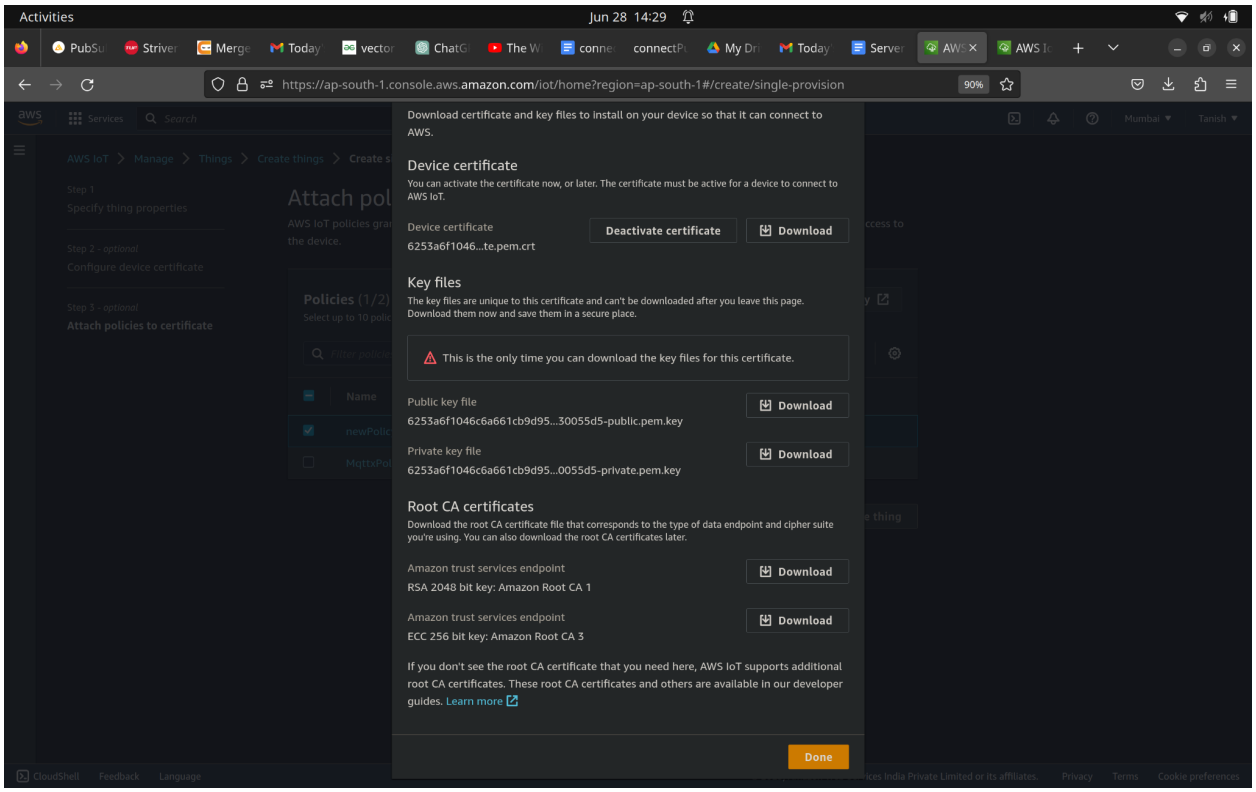
instructions by clicking the “Next” button.



Click on the “Create policy” button.



Name the policy as you like and put an asterisk(*) in the “Policy action” and “Policy resource” fields to give all permissions.

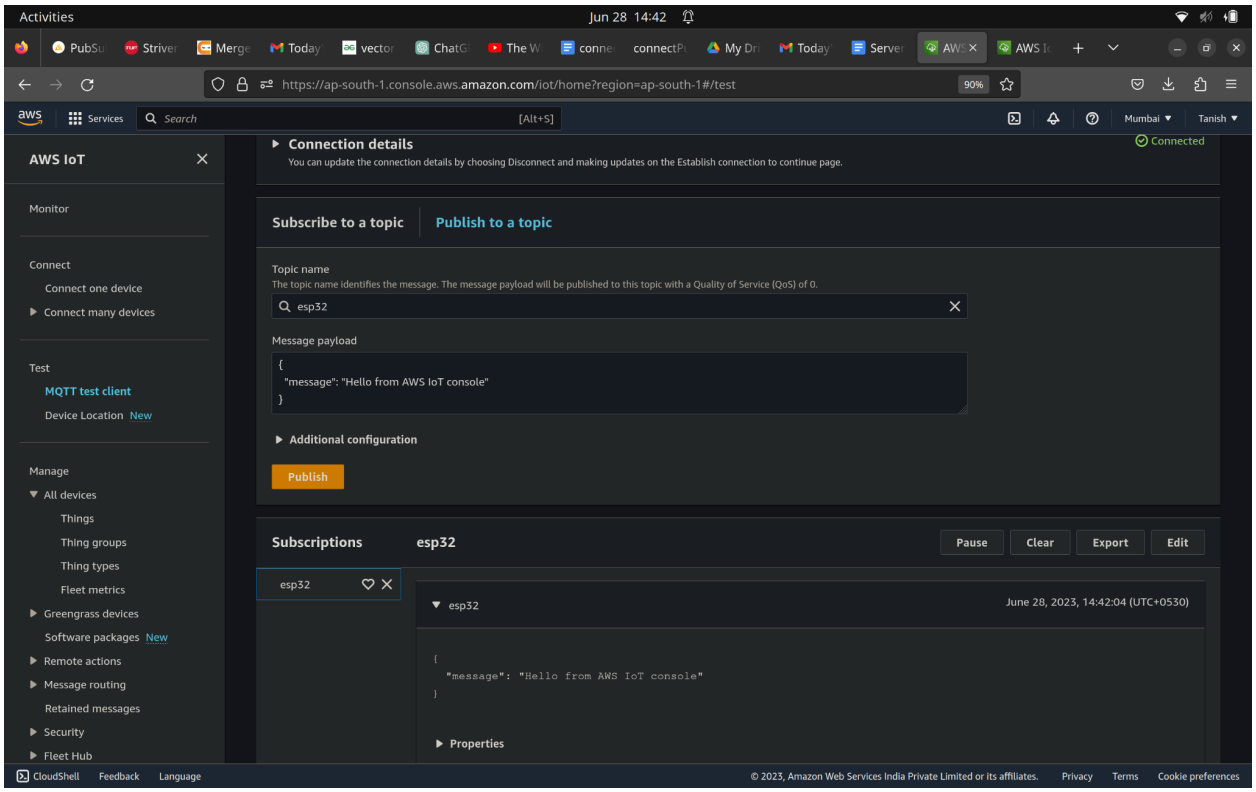


Download all the files except the ECC 256 bit key(This is the only time we can view/download these documents).

Note: These certificates are portable and can be transferred between the devices.

Now use these certificates in connectPubSub.py or connectToFromAWS.ino (i.e secrets.h) to publish and subscribe to topics.

IOT Core has a test Client that you can use to verify whether messages are being transmitted as intended.



Client-side:

You need an Arduino IDE to connect to the esp32 client device, as described in the youtube video linked at the top.

There are two codes:
1, connectToFromAWS.ino with secrets.h a together are code that runs on the esp32 device. The connectToFromAWS.ino file needs to be in the same directory as the secrets.h so that it can read the variable values from the secrets file. This enables modbus slave devices to communicate with aws iot core using modbus to MQTT gateway running on esp32. This communication is two-way.

2,connectPubSub.py to connect any device that can run Python to the AWS IoT core for two-way MQTT communication.