**IoT Security and Privacy**

**Lab 5 – Network Security on ESP32**

**(100 points)**

**Introduction**

In this assignment, students will connect their IoT kit to AWS IoT and use our MQTT test client to check if the IoT kit can send messages to AWS IoT.

Please visit [AWS Free Tier](https://aws.amazon.com/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc) to check free services from Amazon. For example, for 12 months following your initial sign-up date to AWS, you can send 250 K messages (published or delivered) per month for free with AWS IoT. Please ensure you do not send messages too fast and too many so as to avoid charge.

When you get into your AWS IoT portal as shown in Figure 1, click the *Domain configurations* option under *Connect* at the bottom left and you shall find your *Custom endpoint*, which works as a message broker/server. You can publish and subscribe to topics through your custom endpoint/broker. AWS IoT uses mutual authentication. In this assignment, Amazon will be a certificate authority (CA), create a public and private key pair and issue a certificate for our IoT kit, which also need Amazon’s CA certificate to authenticate the custom endpoint/broker. AWS IoT will keep the IoT kit’s certificate for the purpose of authenticating the IoT kit (called client too). This kind of client authentication is similar to the client authentication used by SSH.

Another important concept is the policy, which specifies who can use the custom endpoint for what. Please visit [AWS IoT Core policies](https://docs.aws.amazon.com/iot/latest/developerguide/iot-policies.html). below are *MQTT Policy Actions*.

* *iot:Connect* represents the permission to connect to the AWS IoT Core message broker.
* *iot:Publish* represents the permission to publish on an MQTT topic.
* *iot:Receive* represents the permission to receive a message from AWS IoT Core.
* *iot:Subscribe* represents the permission to subscribe to a topic filter.

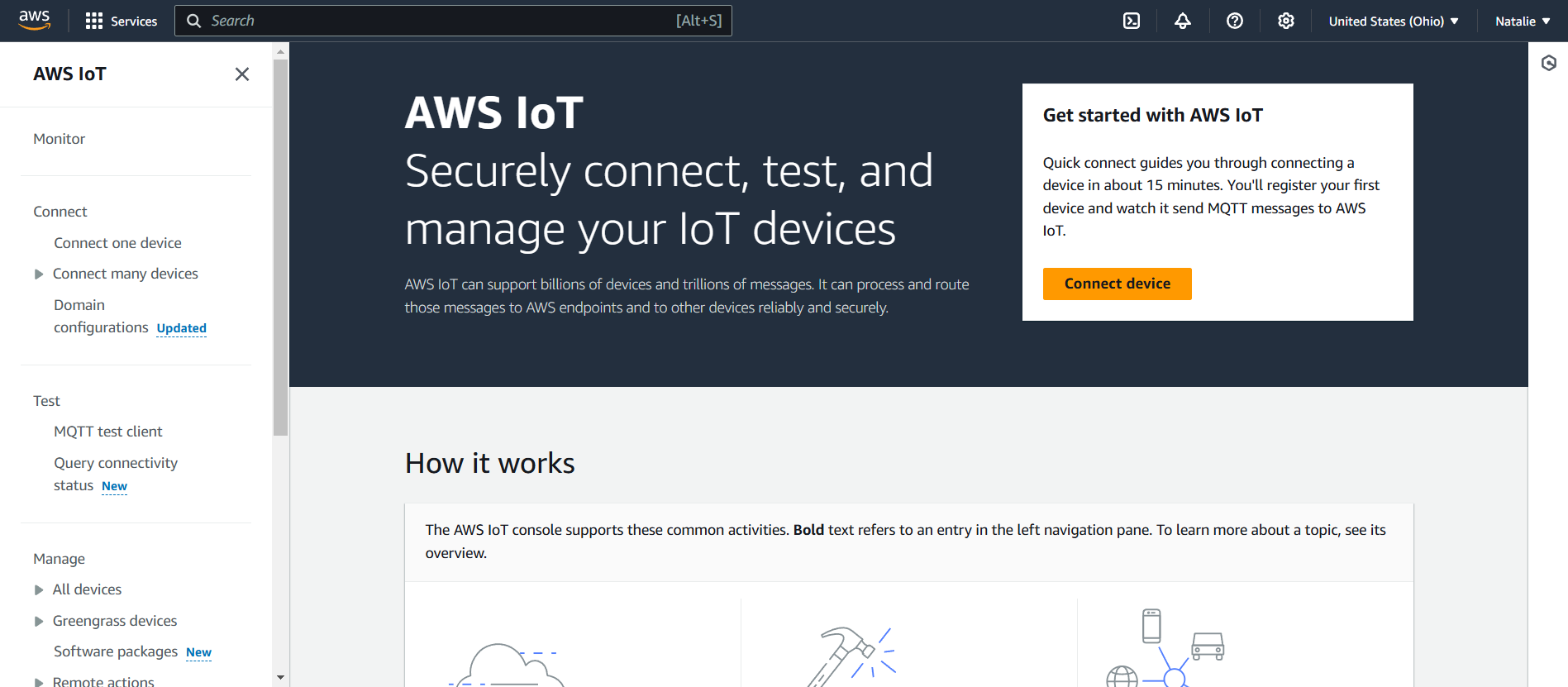
A screenshot of a computer

Description automatically generated

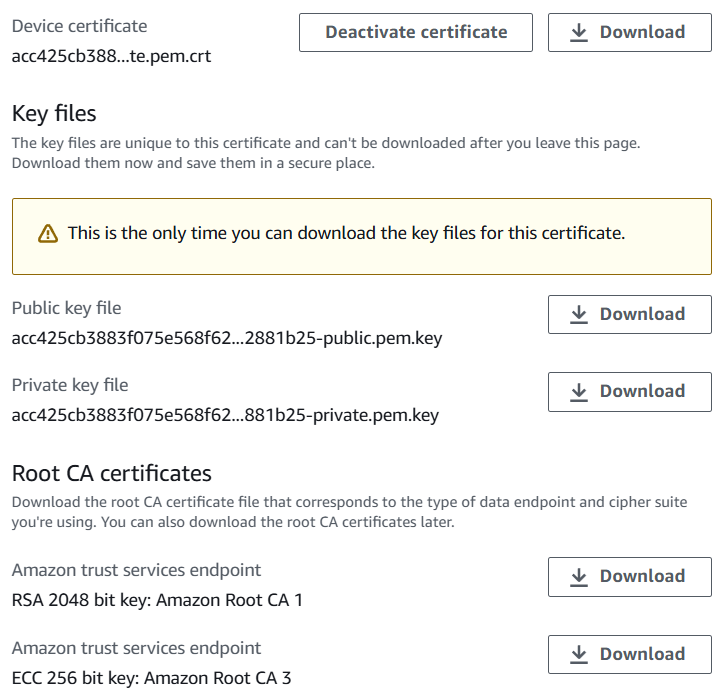
Figure 1 AWS IoT Portal

**Questions**

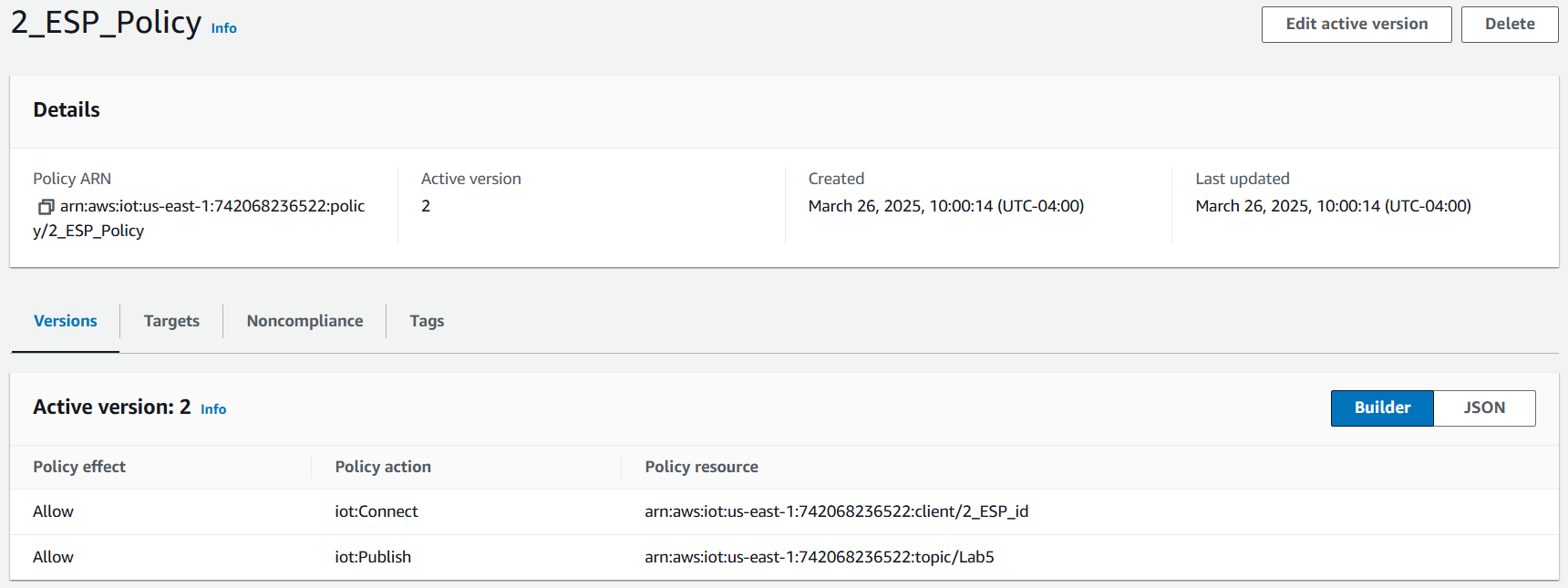
1. Please visit [Set up your AWS account](https://docs.aws.amazon.com/iot/latest/developerguide/setting-up.html) to create your AWS account. Please visit [AWS account root user credentials and IAM user credentials](https://docs.aws.amazon.com/general/latest/gr/root-vs-iam.html) to see the difference between root user and IAM user. Please provide a screenshot of your AWS IoT console. (10 point)



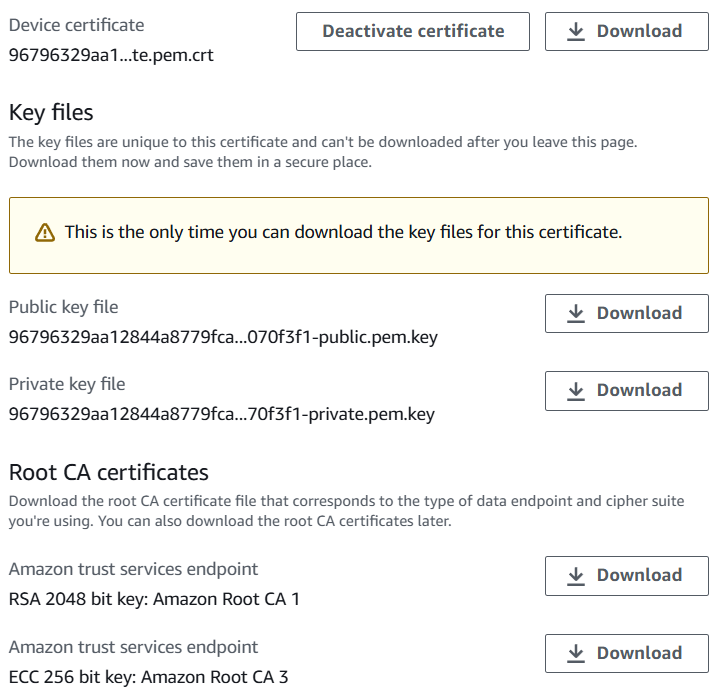
1. Please visit [Create AWS IoT resources](https://docs.aws.amazon.com/iot/latest/developerguide/create-iot-resources.html) and follow steps in this article.
   1. *Create a Thing object*. This Thing should be called ($YOUR\_GROUPNUMBER)\_ESP. (10 point)
      1. Be sure to download the keys and certificate



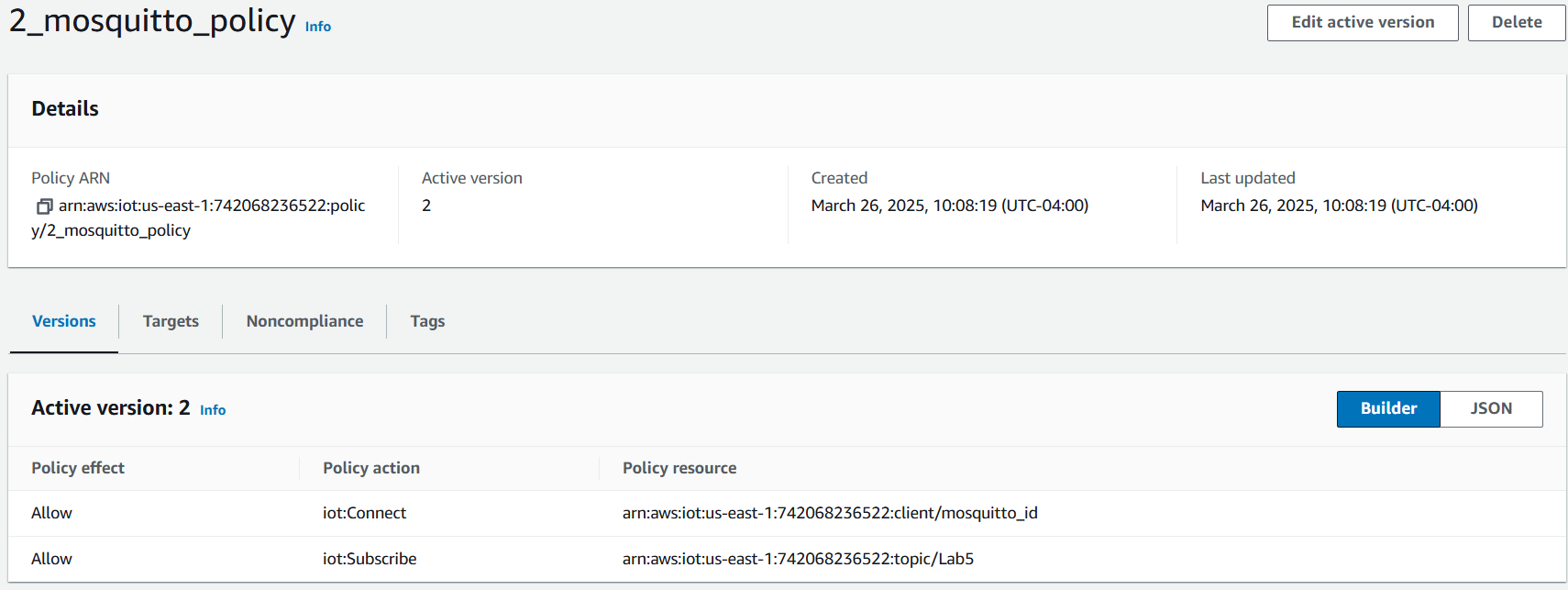
* 1. *Create an AWS IoT policy*. This should be a secure policy used for this assignment. This means that the allowed actions and resources should be specific to what is being asked here. Your ESP Thing should be allowed to connect with the client ID “($YOUR\_GROUPNUMBER)\_ESP\_id” and publish to the “Lab5” topic. Please provide a screenshot of the created policy. (10 point)
     1. REMEMBER: Lone wildcards (\*), while useful for debugging, are not secure. Any use of lone wildcards in your final submission will forfeit these 10 points



* 1. *Create a second Thing object*. Please provide a screenshot of the listed keys and certificate which you can download. This Thing should be called ($YOUR\_GROUPNUMBER)\_mosquitto. (10 point)



* 1. *Create an AWS IoT policy*. This should be a secure policy used for this assignment. This means that the allowed actions and resources should be specific to what is being asked here. Your mosquitto Thing should be allowed to connect with the client ID “mosquitto\_id” and subscribe to the “Lab5” topicfilter. Please provide a screenshot of the created policy. (10 point)
     1. REMEMBER: Lone wildcards (\*), while useful for debugging, are not secure. Any use of lone wildcards in your final submission will forfeit these 10 points



1. Use the downloaded keys and certificates to have mosquitto subscribe to the “Lab5” topic on your AWS account. Send a message with your group number from the IoT MQTT console to that topic and show that mosquitto was able to receive it with a screen shot. (10 point)
2. Modify the provided code to have your ESP client id match the one allowed to connect. Add your certificate, private key, and root ca to the code. Modify the topic the ESP should publish to be “Lab5” Compile and upload to your ESP. (10 point)
3. Show that your mosquitto subscriber is receiving messages from the ESP with a screen shot. (10 point)
4. Please dump your IoT kit’s flash with *esptool.py* (Lab 2) and search the flash for the private key of the IoT kit.
   1. Please discuss the security implications of the fact that the private key of the IoT kit can be stolen from the dumped flash. (10 point)
   2. Our IoT kit contains a crypto co-processor ATECC608A, which has an internal secure storage and hardware crypto acceleration. ATECC608A can be used to hold the private key, which cannot be extracted from outside of ATECC608A and does not leave the secure storage. All necessary cryptographic operations are done inside of ATECC608A. Please explain why the private key is needed by our IoT kit so that we need to protect it. (10 point)

./mosquitto -c "C:\Users\Owner\CLionProjects\EEL-5739\_Internet-of-Things-Security-and-Privacy\Lab5\mosquitto.conf" -v

./mosquitto\_sub -h "atqmjh23xiie5-ats.iot.us-east-1.amazonaws.com" -p 8883 --cafile "C:\Users\Owner\CLionProjects\EEL-5739\_Internet-of-Things-Security-and-Privacy\Lab5\Certificates\_mosquitto\AmazonRootCA1.pem" --cert "C:\Users\Owner\CLionProjects\EEL-5739\_Internet-of-Things-Security-and-Privacy\Lab5\Certificates\_mosquitto\96796329aa12844a8779fca1930be958b4caf523e6aa6fcc562aed3a1070f3f1-certificate.pem.crt" --key "C:\Users\Owner\CLionProjects\EEL-5739\_Internet-of-Things-Security-and-Privacy\Lab5\Certificates\_mosquitto\96796329aa12844a8779fca1930be958b4caf523e6aa6fcc562aed3a1070f3f1-private.pem.key" -t "Lab5" -i "mosquitto\_id" -d -v