IoT Security and Privacy

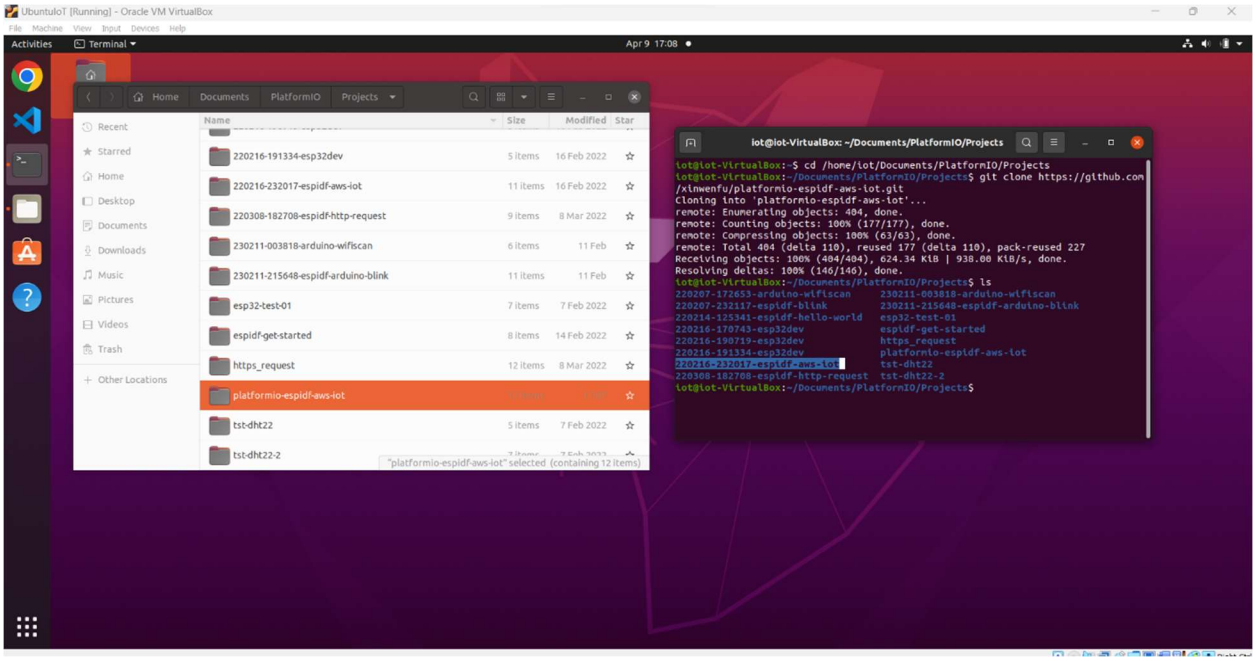
Assignment 5 – AWS IoT Core
(10 points)

Partner: Ishan Patel

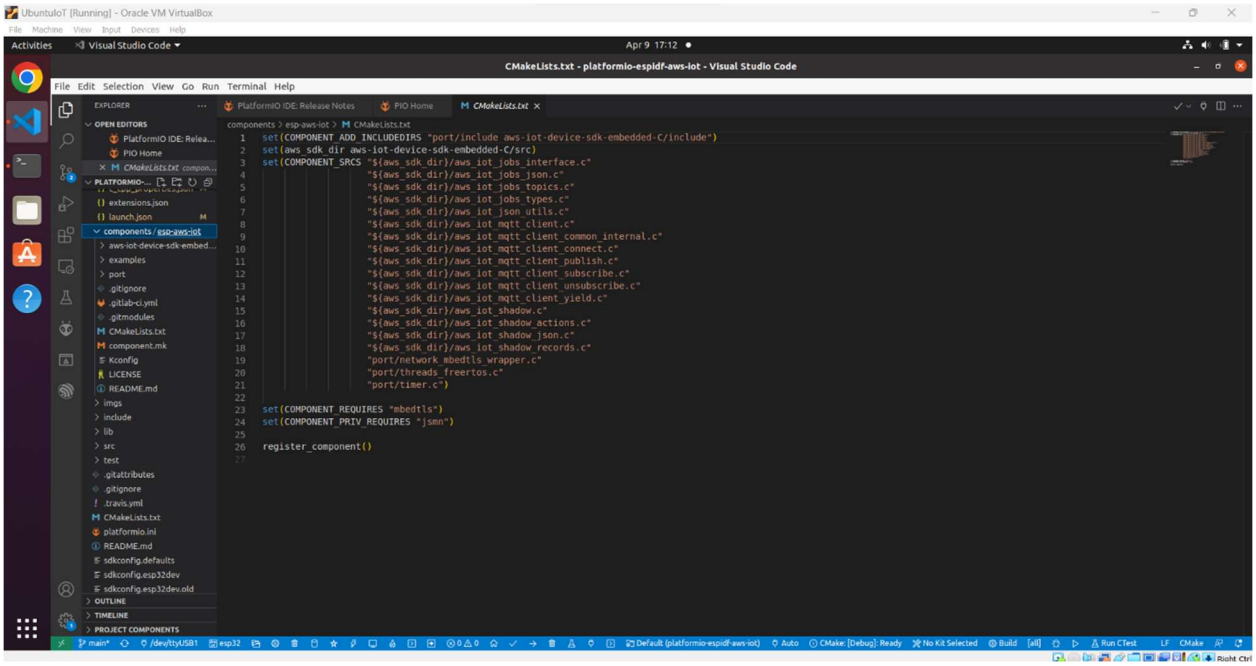
Questions

Students are asked to repeat the work in the tutorial [AWS IoT MQTT Subscribe/Publish Project].

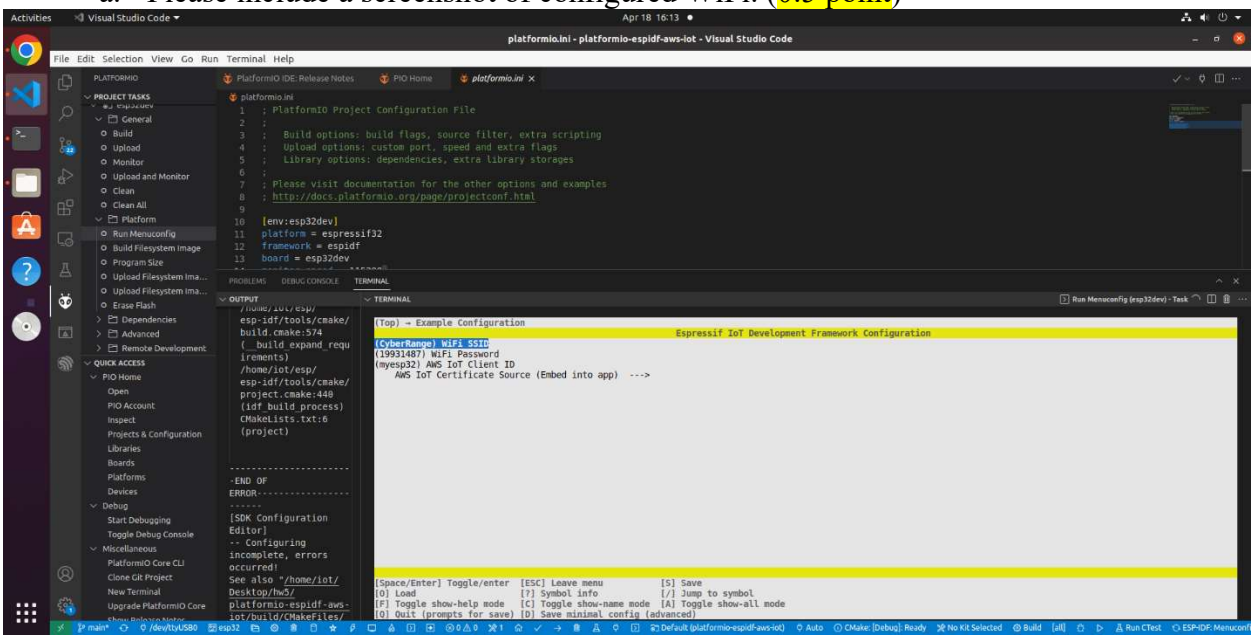
1. Read the section of *Download the project* and perform the actions. Please include a screenshot of the downloaded project in its folder. (0.5 point)



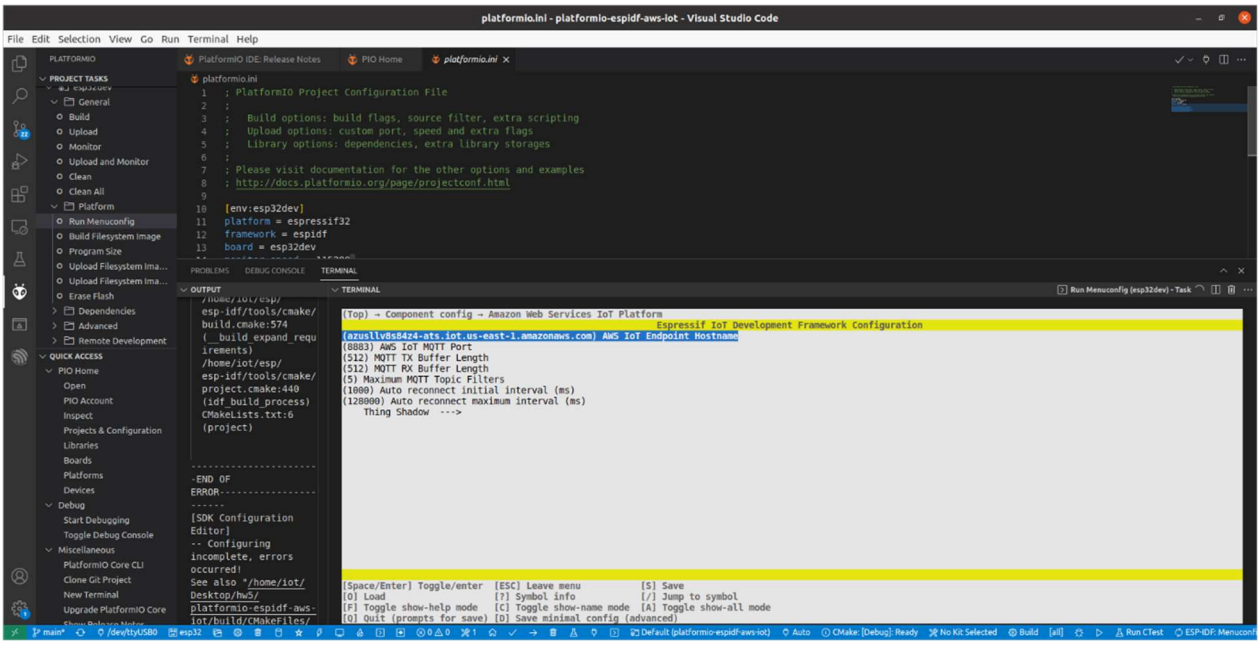
2. Please read the section of *Load the project into VS Code* and perform the actions. Please include a screenshot of the VS code with the loaded project. (0.5 point)



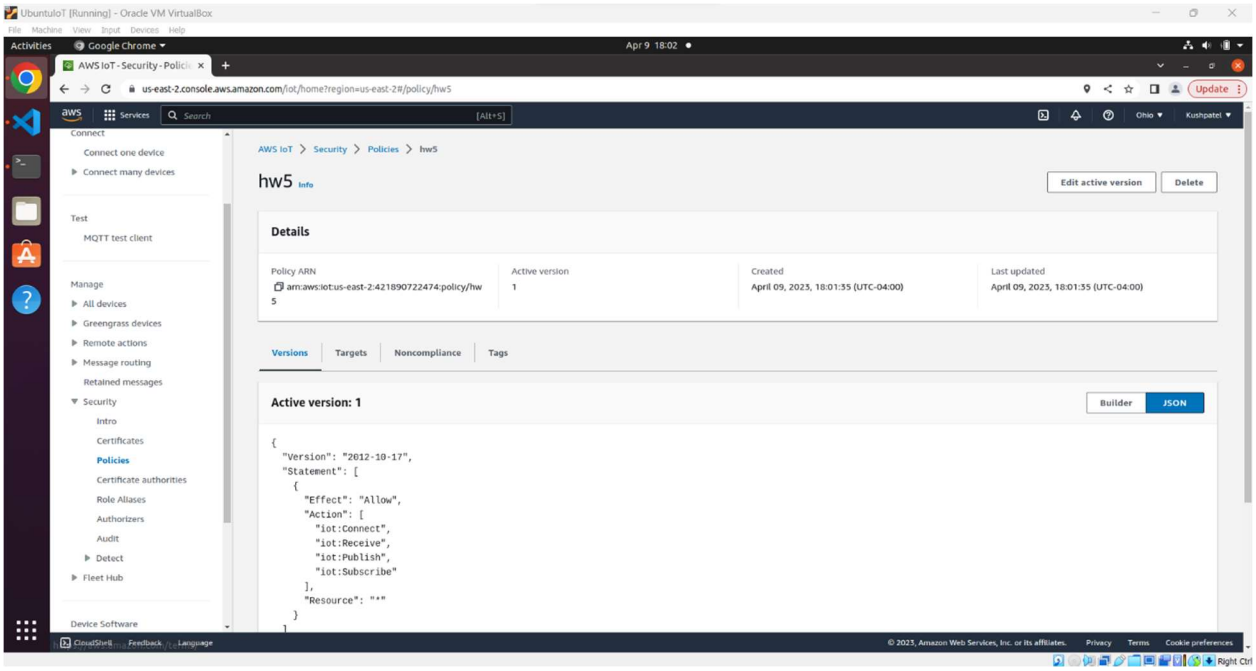
3. Please read the section of *memuconfig* and perform all actions.
    a. Please include a screenshot of configured WiFi. (0.5 point)



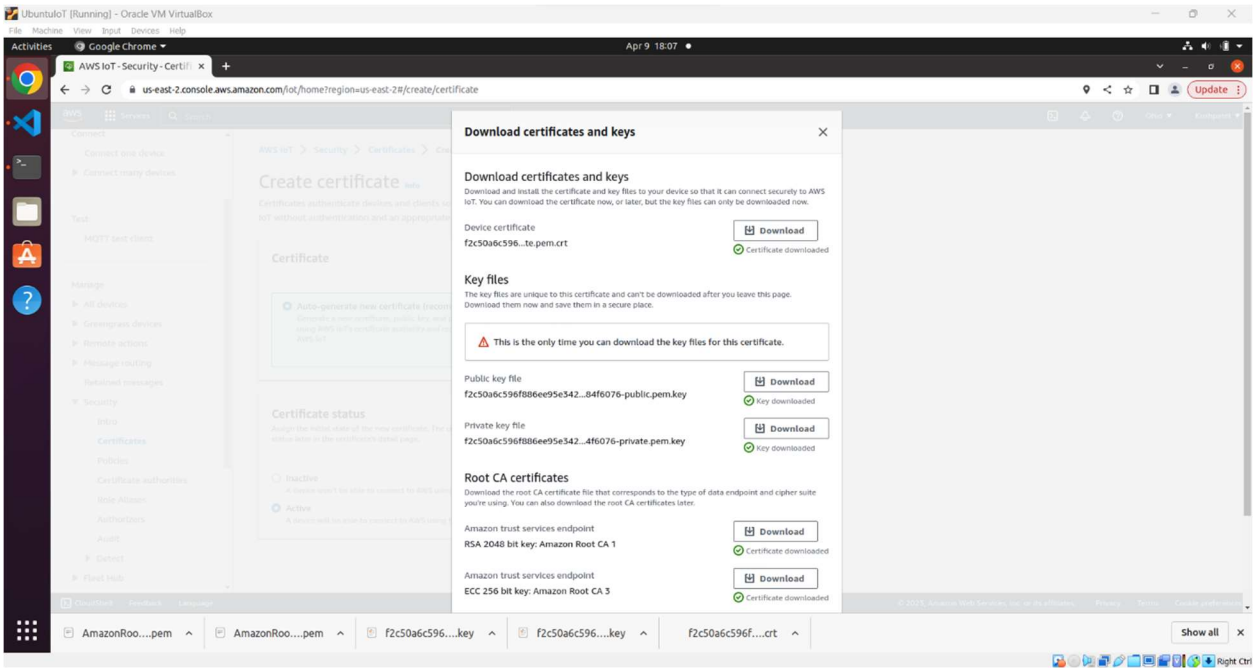
b. Include a screenshot of configured AWS IoT endpoint hostname. (0.5 point)



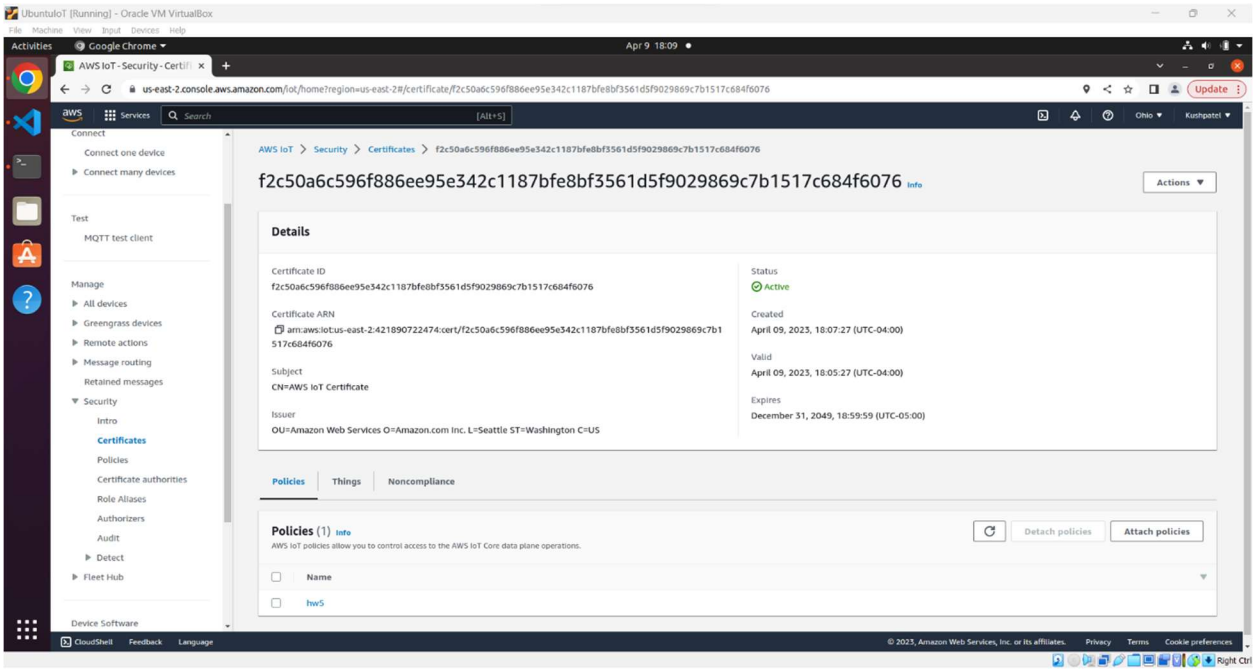
4. Please read the section of *Create policy at AWS IoT console* and perform all actions. Please include a screenshot of created policy. (1 point)



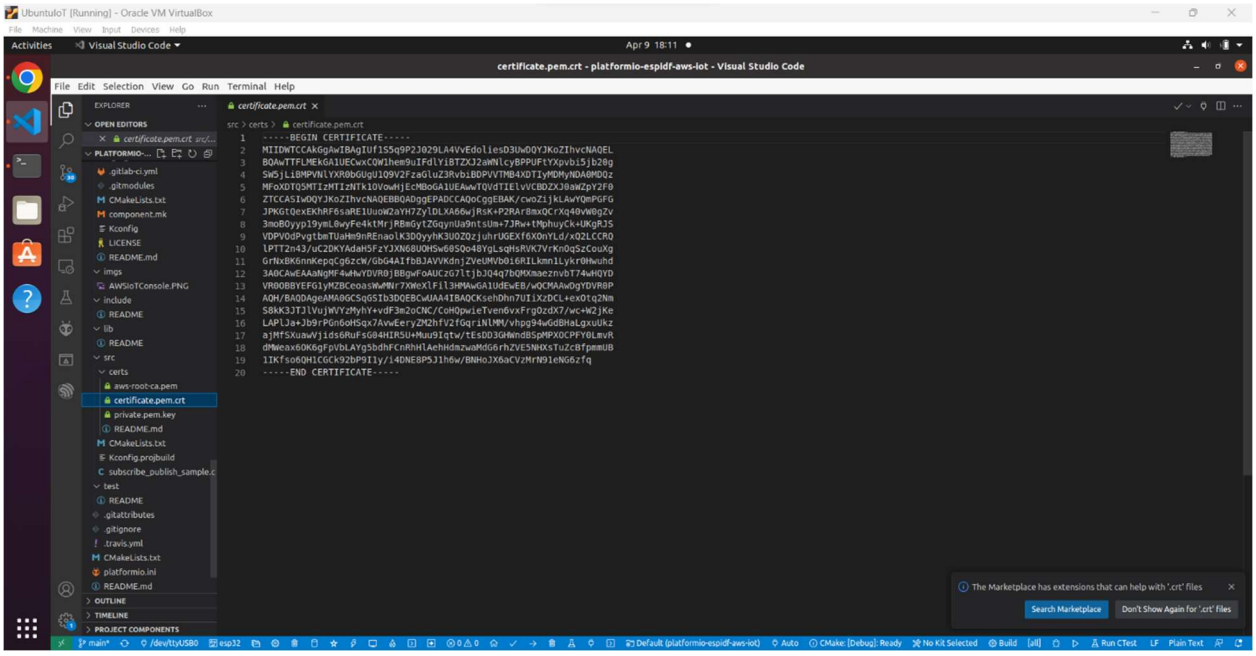
5. Please read the section of *Create certificate at AWS IoT console* and perform all actions.
    a. Please include a screenshot of the AWS IoT Core webpage where keys and certificates can be downloaded. (1 point)



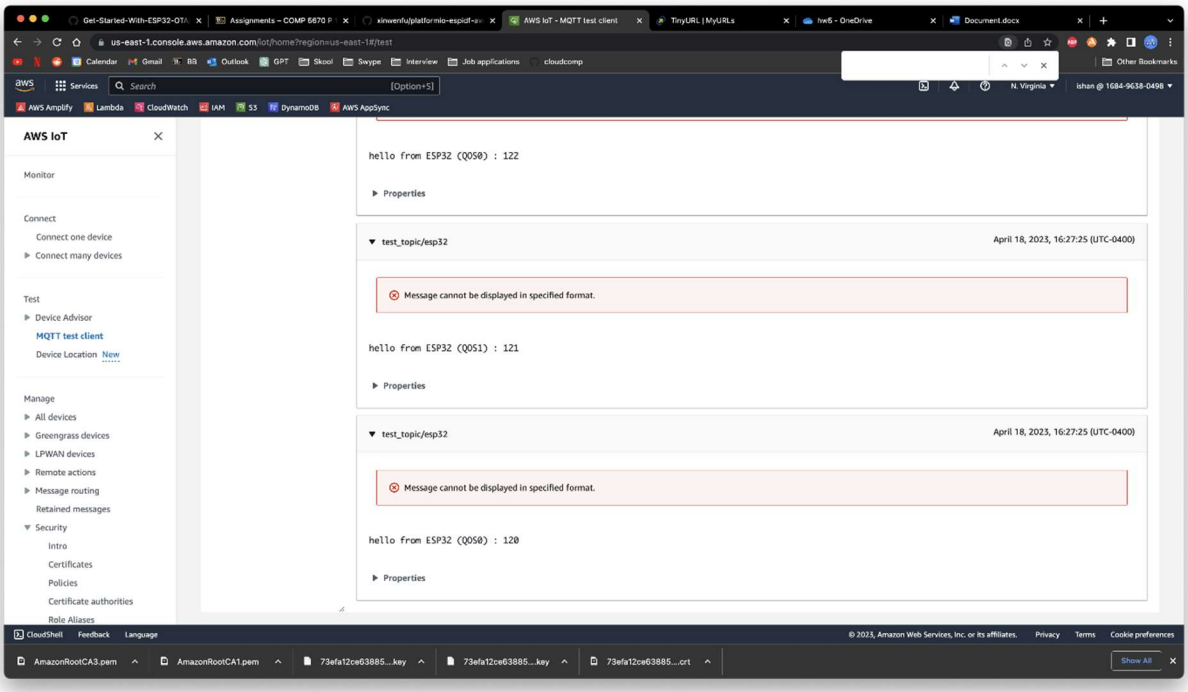
    b. Please include a screenshot of the AWS IoT Core webpage where the policy is attached to the certificate. (1 point)



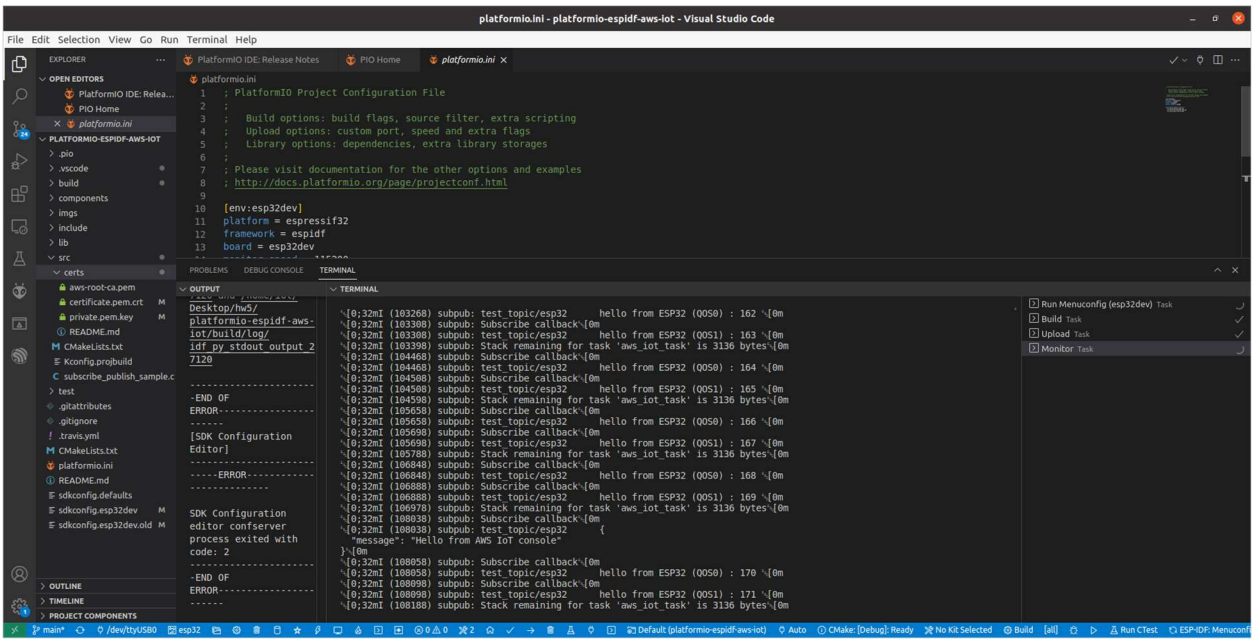
6. Please read the section of *Change device certificate and key file within VS code*. Please include a screenshot of the folder *src->certs* within VS Code. (1 point)



7. Please read the section of *Build, upload and serial monitor*.
    a. Please include a screenshot of AWS IoT Core's MQTT test client receiving messages from the ESP32. (1 point)



    b. Please include a screenshot of the ESP32 receiving messages from the AWS IoT Core's MQTT test client. (1 point)



8. In this assignment, students experienced generating a public key (contained in a certificate) and private key pair and use it to connect the ESP32 to the AWS IoT Core via secure MQTT (i.e., MQTT over SSL/TLS). Recall that the JTAG debugger can be used to extract the firmware out of the ESP32.
    a. What is the danger of hardcoding the private key and public key into the firmware? (1 point)

       The attacker can steal the private key and use it to pretend being ESP32. Than the attacker has access to all the data that the ESP32 contains or is sending/receiving.

    b. There are ways of protecting private keys and public keys. Secure storage is one way. For example, in a crypto-coprocessor like ATECC608A, the private key is stored inside of the crypto-coprocessor chip and never leaves the chip. The public key can also be stored in the chip. Although the public key can be accessed from outside of the chip, it cannot be changed. The crypto-coprocessor accepts messages and performs all the specified crypto processing. Please discuss how to use ATECC608A in the project above, perform the needed crypto operations communicating with the AWS IoT Core while protecting the public key and private key. (1 point)
       Note: students are not required (actually cannot since ATECC608A is not available) to do the actual project on ATECC608A.

       First generate a new key on the chip and store in the chip just like how ATECC608A says. Than use the public key from the chip to configure the AWS IOT Core to accepted incoming messages. When sending messages, ATECC608A would sign the messages with the private key.