Task08- Step B

- AWS Account를 만들고 개발환경을 구축하라
- Thing을 생성하고 인증서와 정책을 생성하여 연결 (Seoul Region)
  이름: ESP32_testButton
- Simple Notification Services에 다음의 Topic을 생성하라
  Topic name: emailToMe
  동작: 자신의 email로 Payload를 전송
- 다음과 같은 규칙을 생성하라
  ** Topic 'esp32/button' 이 게시되면
  자신의 email로 Payload를 전송 (메시지형식은 RAW)
- AWS-IoT Core의 테스트(Client emulator)에서
  'esp32/button' topic을 게시함으로 자신의 메일로 Payload가 전송되는지 확인하라
  ** Payload: { "message": "ESP32 Button Pressed!" }

Task08- Step A

- 다음과 같은 Thing을 생성하라
  ** name: ESP32_BME280
  ** topic: esp32/bme280
  ** payload example:
  { "temp": 25, "humid": 40, "press": 1010 }
- 다음과 같은 규칙을 생성하라
  ** esp32/bme280 topic를 받고 "temp"가 30 이상이면
  자신의 email로 payload를 전송

- AWS-IoT Core 테스트를 사용하여 동작상태를 검증하라

Task08- Step C

- BME280과 LED를 가진 ESP32를 Thing으로 생성하라
  1. 온도가 40 이상이면 자신의 email로 통보하고
  2. 온도가 30 이상이고 습도가 40 이상일때 LED를 OFF하고
  3. 온도가 20 이하이고 습도가 10 이하일때 LED를 ON하는

IoT 시스템을 설계하라

- AWS-IoT Core 테스트를 사용하여 동작상태를 검증하라
  (esp32/led 구독)

** 작업구성을 "IoT 주제 재게시"로 할것

** Topic과 Payload 자유롭게 결정

** Hint: esp32_bmeled_control2.. SELECT {"led": "OFF"} FROM 'esp32/bme280' WHERE temp >= 30 AND humid >= 40, republish('esp32/led')