Lab5: Introduction to AWS and RESTful API

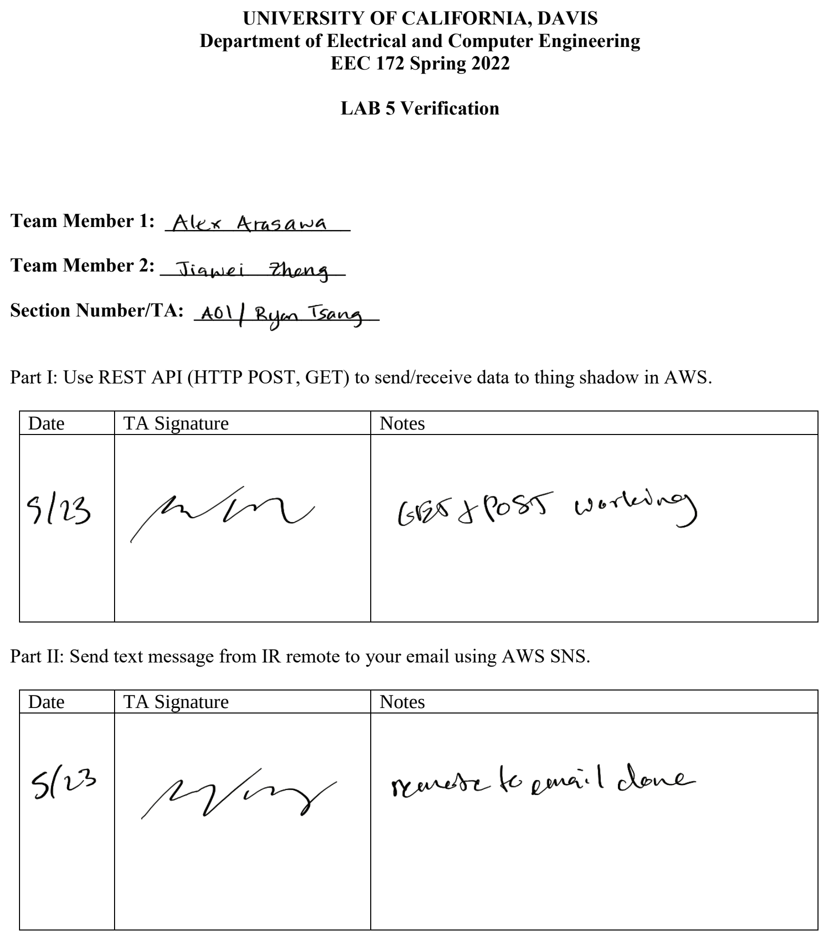
Alexander Arasawa, Jiawei Zheng

June 11, 2022

EEC 172

Professor Soheil Ghiasihafezi

1 Check off from TA:



UNIVERSITY OF CALIFORNIA, DAVIS
Department of Electrical and Computer Engineering
EEC 172 Spring 2022

LAB 5 Verification

Team Member 1: Alex Arasawa

Team Member 2: Jiawei Zheng

Section Number/TA: A01 / Ryan Tsang

Part I: Use REST API (HTTP POST, GET) to send/receive data to thing shadow in AWS.

Date	TA Signature	Notes
5/23		GET & POST working

Part II: Send text message from IR remote to your email using AWS SNS.

Date	TA Signature	Notes
5/23		remote to email done

2 Introduction

The main goal of the lab was to learn how to connect the CC3200 LaunchPad with Amazon Web Services (AWS). The first part of the lab was to connect the CC3200 LaunchPad to AWS. The second part was to setup Simple Notification Service (SNS) on AWS so that we can compose a text using an IR remote and send it to our email. Lastly, integrate multi-tap texting to send text messages to our email over AWS.

3 Background

CCS software is an IDE for configuring TI's microcontrollers and embedded processors. The software contains many tools to help with creating and debugging applications: CSS includes an optimized C/C++ compiler, a code editor, a project build environment, a debugger, and many other features.

CCS Uniflash is a standalone tool used to program flash memory of TI's microcontrollers and program flash memory of Sitara processors.

TI Pin Mux Tool is a software used for configuring pins, peripheral signals, and other components of a system.

AWS IoT Core is an Amazon Web Service (AWS) that allows you to connect IoT devices and route messages to other AWS services.

References:

https://www.ti.com/tool/CCSTUDIO

https://www.ti.com/tool/UNIFLASH

https://www.ti.com/tool/SYSCONFIG

https://aws.amazon.com/iot-core/?nc=sn&loc=2&dn=3

4 Goals/Objectives

4.1 Part I: Connecting to AWS (securely) and updating your Shadow

4.1.1 Part I.A: Setting up an Amazon AWS account

Create an Amazon Web Services (AWS) account and watch a tutorial on how AWS and IoT works.

What you should learn from this task:

- How to setup an AWS account
- Get familiar with the controls under the AWS IoT Core service

4.1.2 Part I.B: Setting Up Your First Device Thing/Shadow with the AWS IoT Console

Follow the instructions on adding a thing/shadow to AWS IoT and which certificates/keys are needed to connect to AWS.

What you should learn from this task:

- Learn how to create a thing/shadow for the CC3200 LaunchPad
- Learn what certificates and keys are used for
- Which certificates are important to keep for setting up an encrypted connection to AWS
- How to attach certificates and keys to a device thing/shadow
- How to setup permissions for the device thing/shadow to use IoT functions of AWS

4.1.3 Part I.C: Making a Policy to Allow Update Update/Get Status from Thing Shadows

Create two policies to give the device thing privilege to access AWS through REST GET and POST commands from the to update a shadow.

What you should learn from this task:

- How device shadows represent the state of your device on the server
- Concept of device privilege to perform actions requested
- How to create policies for the device shadow to perform actions
- How to attach policies to a certificate to allow GET and POST commands via REST from devices associated with the certificate

4.1.4 Part I.D: Converting the Keys/Certificates for Use with the CC3200

Use OpenSSL to convert downloaded keys from .pem to .der format for the LaunchPad.

What you should learn from this task:

- What format the CC3200 LaunchPad uses for certificates/keys
- The components for connection security between the LaunchPad and AWS through TLS protocol (public key, private key, root certificate)
- How to use OpenSSL to convert certificate/key formats

4.1.5 Part I.E: Using UniFlash to flash key and certificates to CC3200

Follow instructions to flash keys and certificates onto the CC3200 device so that it can securely connect and push actions to AWS shadow.

What you should learn from this task:

- How to use UniFlash to flash certificates and keys to the CC3200 LaunchPad
- How to list files in UniFlash

4.1.6 Part I.F: Accessing AWS using the RESTful API

Configure correct keys and certificates, date and time, and format POST and GET JSON in the program to send REST commands to the device shadow.

What you should learn from this task:

- How to format JSONs so that AWS things can run GET and POST requests
- How to program GET and POST methods
- How to configure connections for AWS to connect using TLS protocol
- How to connect to the Internet through an Access Point by modifying common.h

4.2 Part II: Using SNS to send a text to your email

4.2.1 Part II.A: Creating an SNS Topic

Create a new topic in the SNS service and create a subscription for the email.

What you should learn from this task:

- How to use the SNS module for different modes of notification
- How to link an email to an SNS subscription

4.2.2 Part II.B: Creating an IoT Rule

Create a rule that will trigger when you push updates to the device shadow.

What you should learn from this task:

- How to create rules in AWS to trigger when an event occurs
- How to modify the SQL statement to clean-up the message

4.2.3 Part II.C: Integrate your IR Remote multi-tap texting to send a message to your email

Integrate the AWS function to Lab 3 in order to allow the IR remote to compose text messages and send them to your email.

What you should learn from this task:

- How to integrate the Lab 3 functionality with the connection protocol

5 Methods

5.1 Part I: Connecting to AWS (securely) and updating your Shadow

There were clear instructions on how to setup AWS and how to start a device thing for keeping a device state on AWS. First we started by setting up an Amazon Web Service account in order to access AWS IoT Core and learning how AWS works. Then we setup the device 'thing' on AWS and a shadow for the CC3200 LaunchPad. Where we were provided keys and certificates to associate with the shadow. Next we established a policy for GET and POST that allows the device shadow the privilege of fetching status information and posting status for the device shadow. After we had to convert the public and private keys from the .pem format to .der which was compatible for the CC3200 LaunchPad using OpenSSL. Following the conversion we flashed the CC3200 with the root certificate and public and private keys so that it can establish a secure connection to AWS over TLS protocol. Finally, we tested the RESTful commands by sending GET and POST commands to the device shadow to update the shadow state and fetch its current status. In this part we had to double check the TLS code by verifying the correct version, cipher, port number, etc. Also to make sure time matched reasonably well so when settime() was called it worked correctly.

5.2 Part II: Using SNS to send a text to your email

The first part of the this was clear on how to create a subscription and link SNS to an email. First we had to go to the SNS module and add a subscription for email. Then we create an IoT rule that allowed a method to occur when the shadow for the device was updated using MQTT (email in this case). The last part was integrating the prior code with Lab 3 code.

6 Discussion

The first challenge was dealing with the certificates. The smaller sub-issue being that OpenSSL was hard to locate on the lab computers. We ended up finding out that it was available under Cygwin. Then the main issue was knowing when the certificates were flashed correctly. Using list files should have helped since we could see for sure that they were correctly flashed in the file system. However for the most part, the challenge was that the wifi we used was not working properly for us. This was resolved by using a mobile hotspot as an Access Point.

7 Conclusion

In this lab, we learned how to use AWS to create a device shadow for our CC3200 LaunchPad. How to configure the device show to allow it to perform GET and POST commands through policies and what to do with certificates and keys provided. Also how to convert those certificates and keys to a format usable by the CC3200. In the second part of the lab, we learned how to create SNS subscriptions to trigger when the shadow was updated. And how to integrate this code with our Lab 3 code to send text when the IR remote sends messages.

8 Contribution

We did all parts of the lab together so 50/50 contribution.