CPSL
Cyber-Physical Systems Laboratory

Cloud Tutorial: AWS IoT

CSE 521S Fall

Sep. 17, 2020

Ruixuan (Corey) Dai

Washington University in St. Louis

CPSL
Cyber-Physical Systems Laboratory

XaaS: Basics in Cloud Computing

Cloud Computing

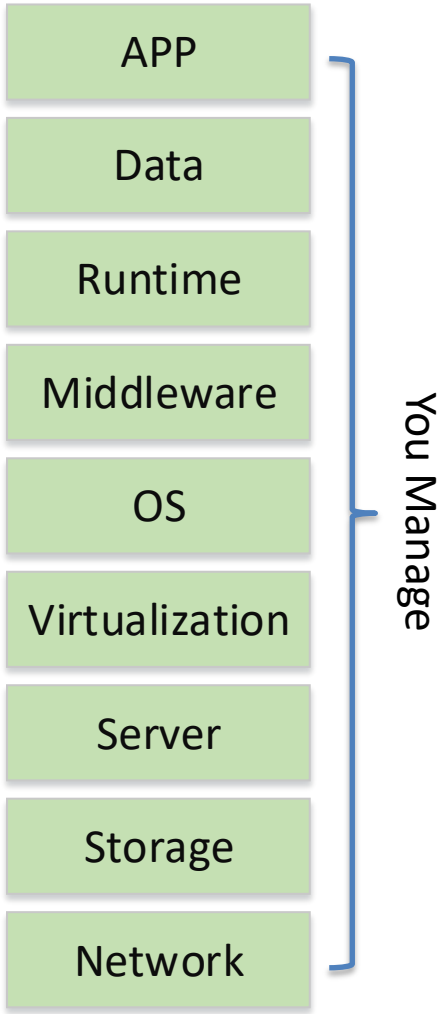
- Cloud computing provides **shared pool of configurable computing resource** to end users **on demand**
- Three service models
  - **IaaS (Infrastructure as a Service):** virtual machines, storage, network …

  - **PaaS (Platform as a Service):** execution runtime, middleware, web server, database, development tool …

  - **SaaS (Software as a Service):** email, virtual desktop, games …

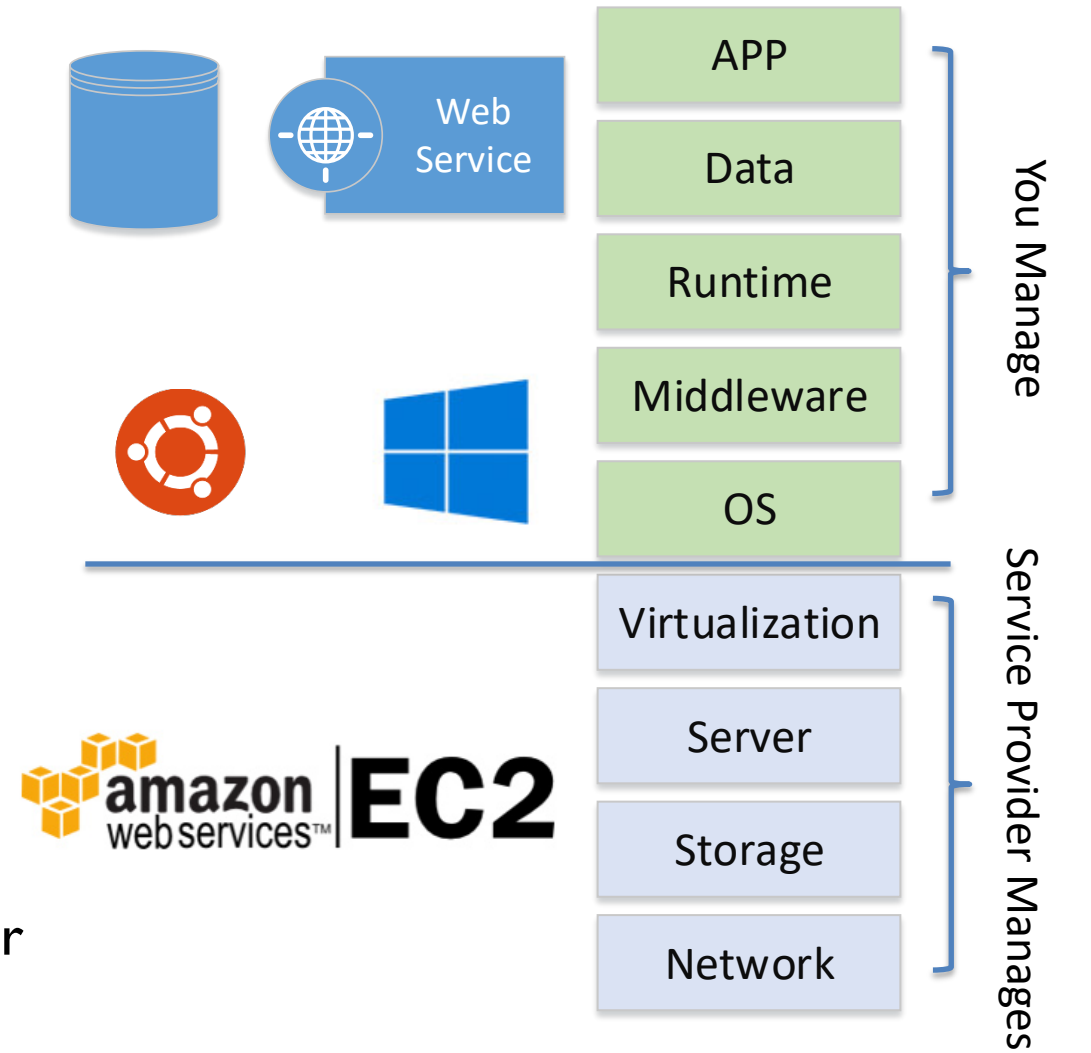
~~Cloud Services~~: On-premise Software

- Traditional
- installed and runs on personal computer
- You Manage and Deploy
  - Hardware
  - OS
  - Software
- Example
  - This presentation



Infrastructure as a Service (IaaS)

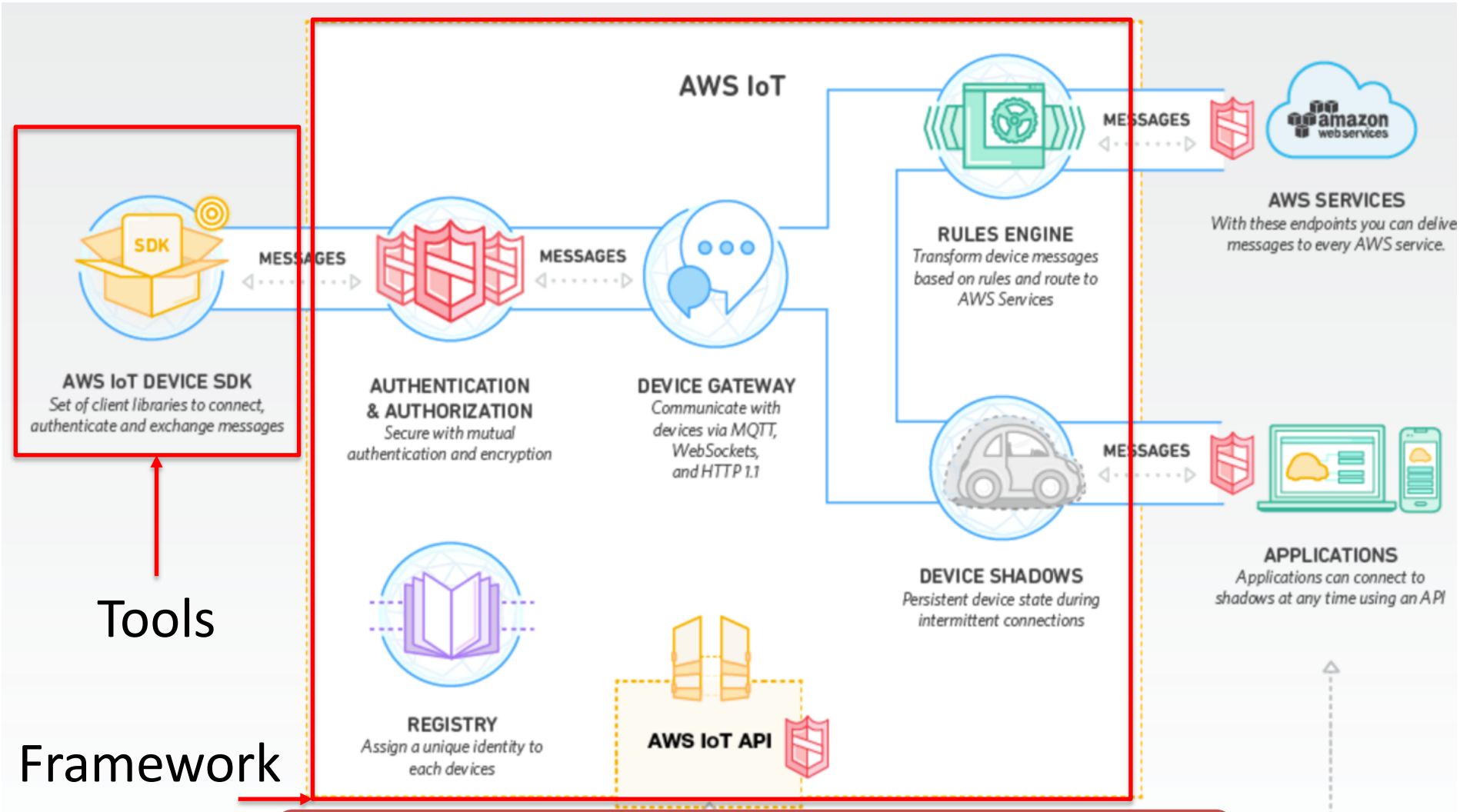
- IaaS
  - "physical server box"
  - Virtual Machine
    - Memory
    - Storage
    - CPU
    - Network
- Example
  - AWS EC2
  - AWS EFS
- Use case
  - Build up you VM cluster



Platform as a Service (PaaS)

- PaaS
  - You get a framework
  - Host Application
  - Tools
- Example
  - AWS IoT
- Use case
  - Build up you're smart A/C controller

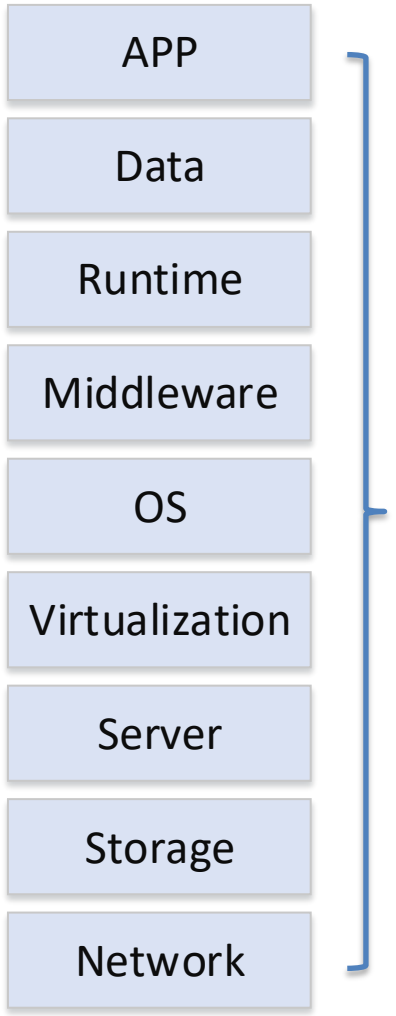
PaaS Example: AWS IoT



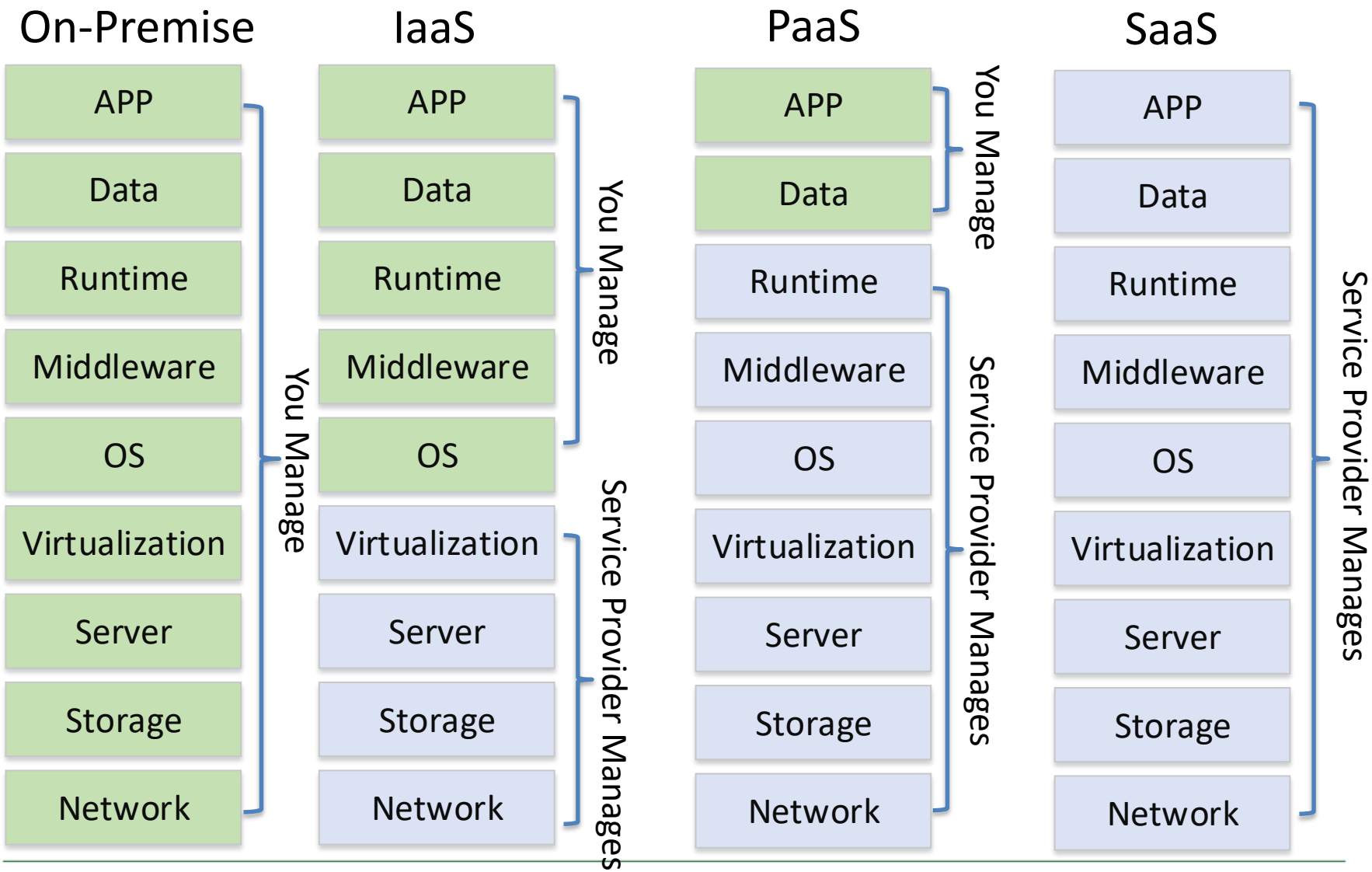
The essence is **MESSAGING MIDDLEWARE**
Send messages between sensors and servers…

Software as a Service (SaaS)

- SaaS
  - You get a whole solution
- Example
  - Gmail
  - Dropbox
  - Office365



XaaS: A Recap


On-Premise
IaaS
PaaS
SaaS
APP
Data
Runtime
Middleware
OS
Virtualization
Server
Storage
Network
You Manage
Service Provider Manages
Service Provider Manages

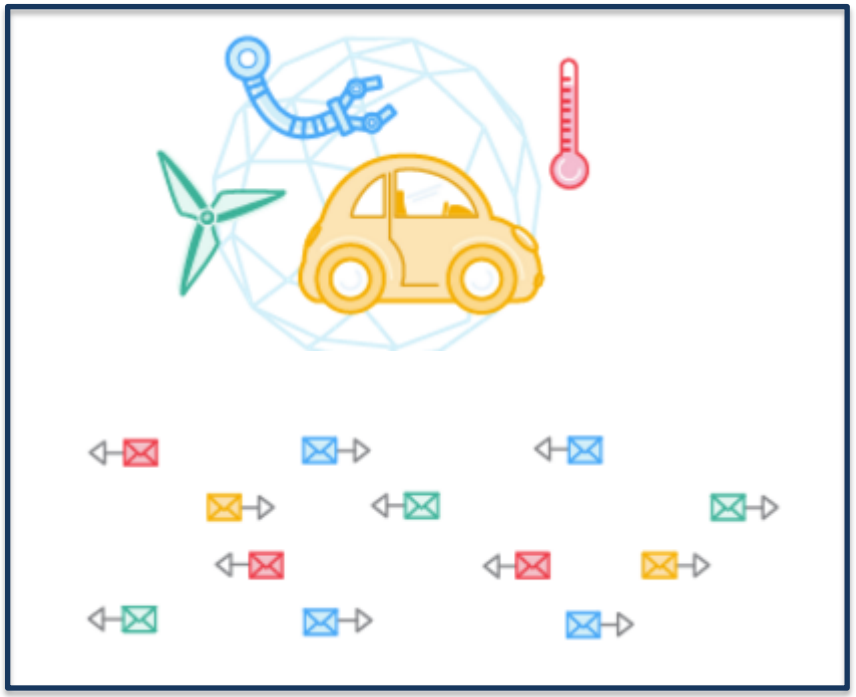
CPSL
Cyber-Physical Systems Laboratory

Tutorial: Hello! AWS IoT!!

Washington University in St. Louis

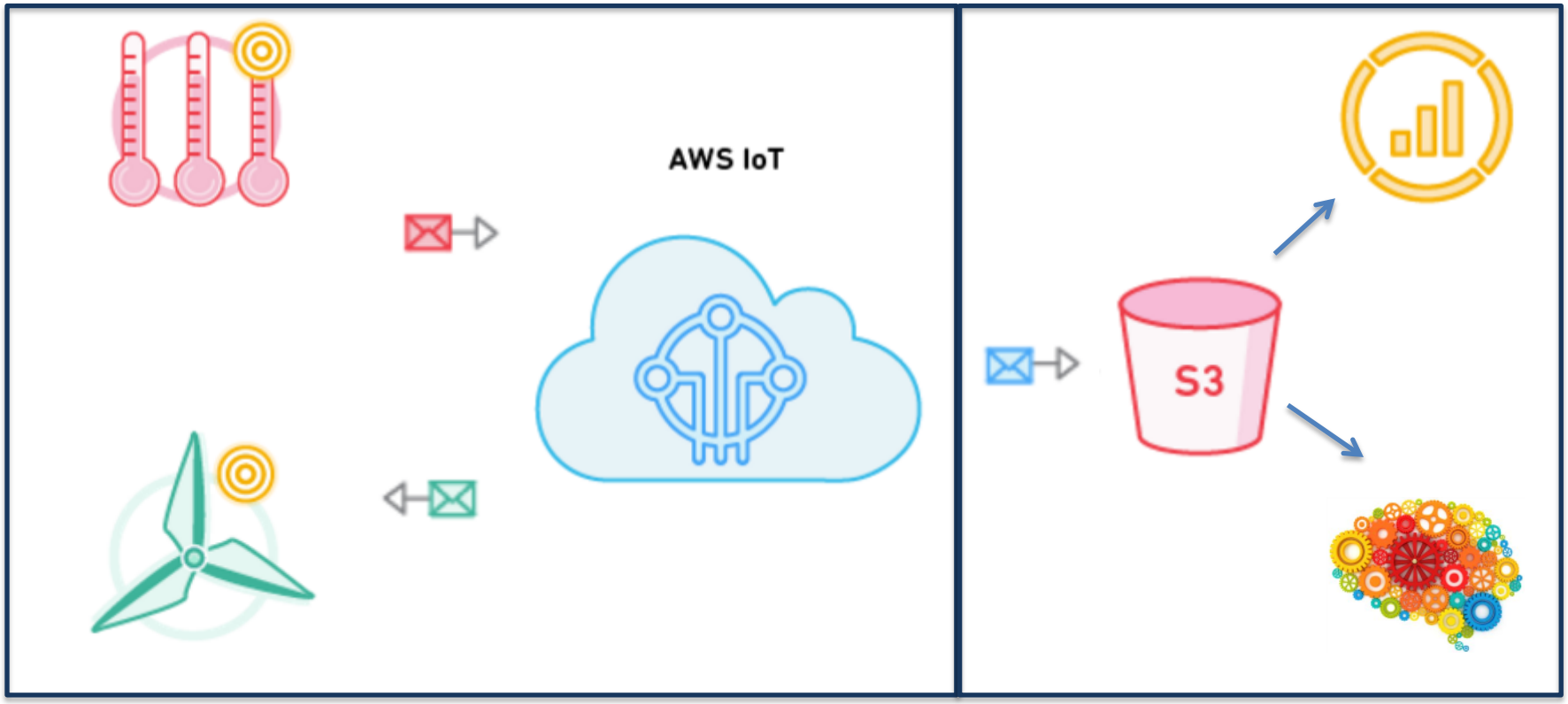
Internet-of-Things

- Internet-of-Things
  - Devices
    - Different Types
      - Sensors, actuators
  - Data and Command
    - Sensing the world
    - Give Response
  - Challenge
    - United: Connected + Communication
    - Smart: Data Analytics + Strategy



Source: https://aws.amazon.com/iot-platform/
http://www.brain-smart.net/smart-brain-health-blog/page/2/#axzz4W4oSp8a6

Solution: AWS IoT



United: Connect + Communication

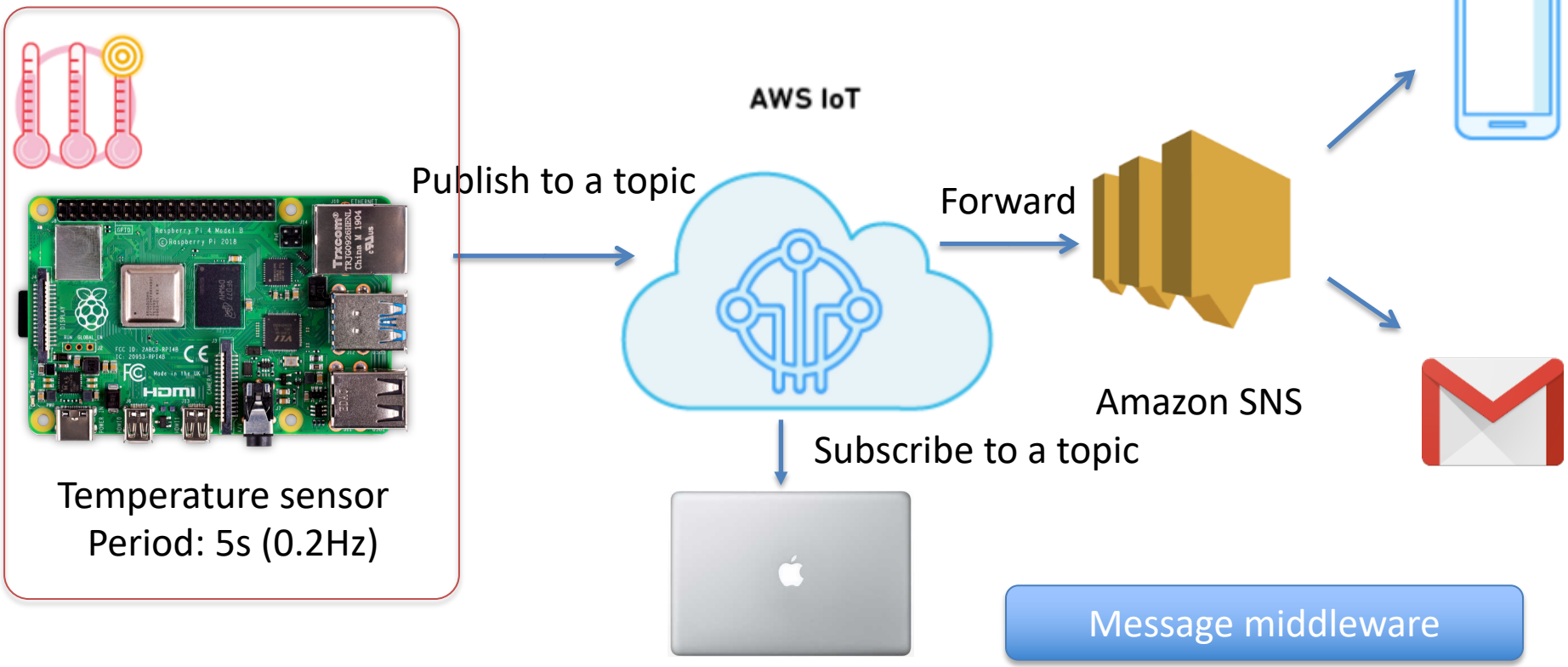
Stated: “Thing Shadow”

Smart: Other Cloud Service
Data Storage
Machine Learning

Source: https://aws.amazon.com/iot-platform/

Tutorial: Hello AWS IoT!

- Key concepts:
  - Publisher (e.g. Sensor), Subscriber (e.g. Server), **Topic**
    - **Topic is used to identify the message.**
- Not a traditional "peer-to-peer" communication.



Source: https://aws.amazon.com/iot-platform/

Basic Interact: Subscribe/Publish

- You can define your own Topic (Twitter Account)
- Subscriber can receive the message you published to your topic

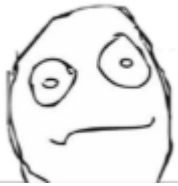
Twitter account
a "Topic"

Follow
"subscribe"

Tweet
"Publish"

I am going to publish a tweet.
"Today is beautiful "

Publisher



Subscriber

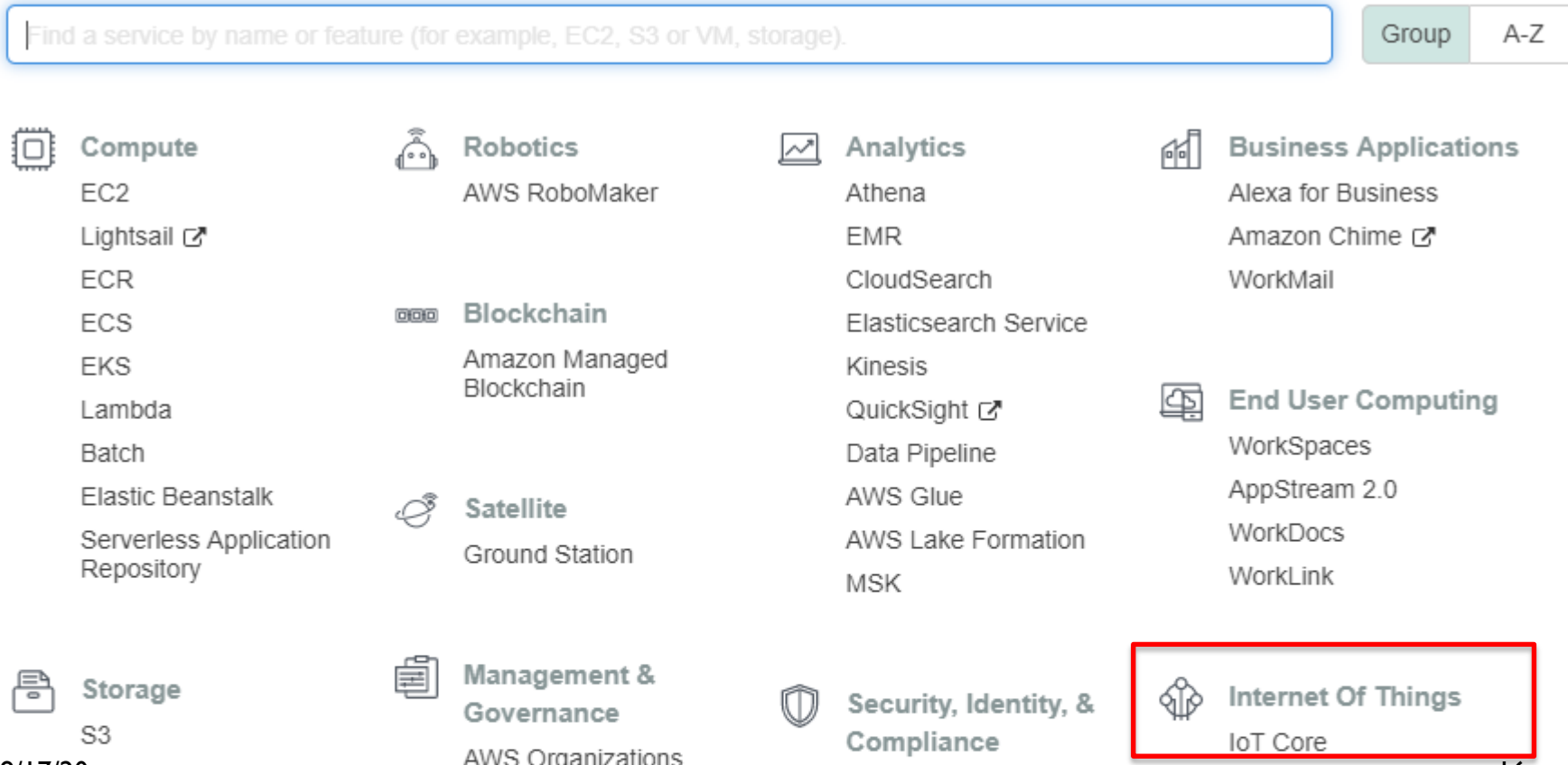
Received message from Trump:
"Today is beautiful"

Resources

- Amazon IoT
  - http://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html
- Raspberry Pi
  - https://www.raspberrypi.org/
- Resource list for course projects
  - http://cps.cse.wustl.edu/index.php/List_of_Projects
- Apply for $40 credits for Amazon AWS
  - https://aws.amazon.com/education/awseducate/apply/

Get into AWS Manage Console

- Create your own AWS account
- Sign In IoT Manage Console
  - https://aws.amazon.com/iot/



Step 1: Create a Virtual "Thing"

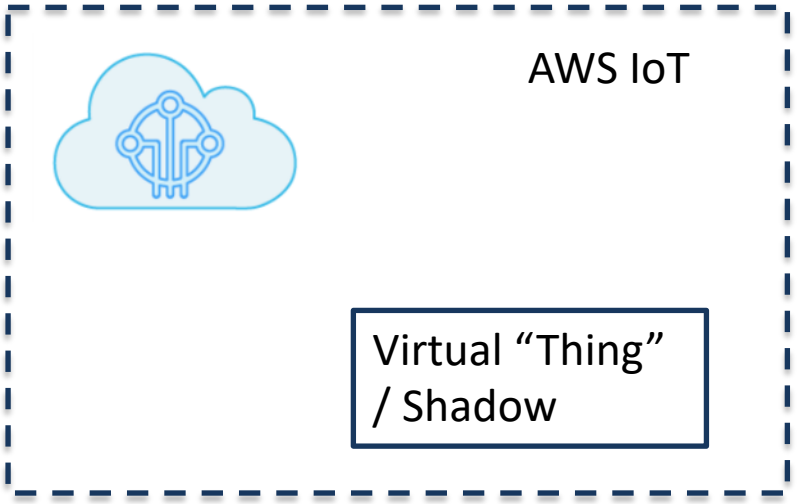
- Virtual "Thing": a **virtual copy** of your thing
  - A Thing in AWS IoT has a **"shadow"**
    - a JSON document that is used to **store and retrieve current state information for a device.**
      - **E.g.** Battery level, Connectivity, data

A "Dashboard" to show some info

  - **Shadow is a special topic in AWS IoT**

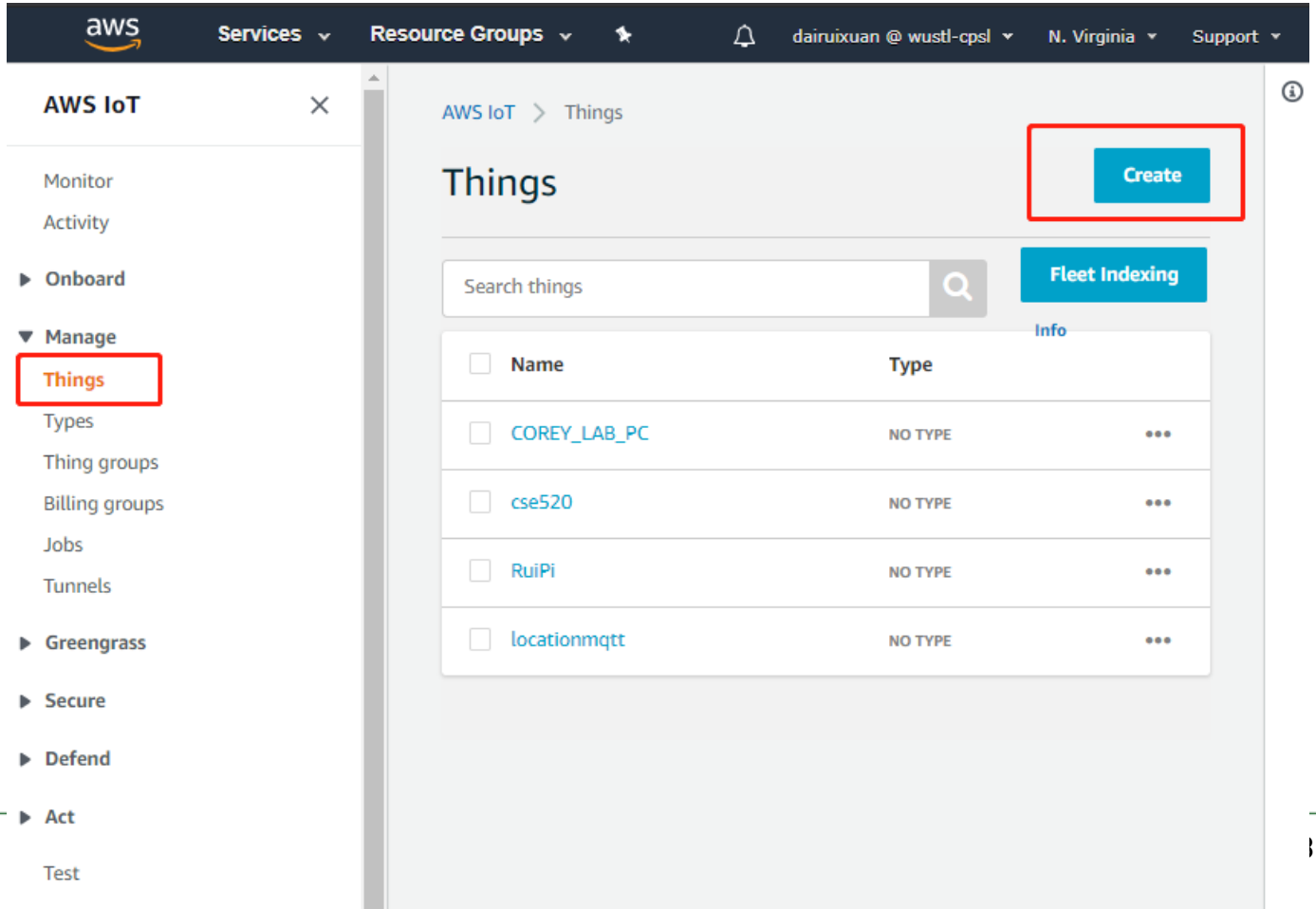
- **Certificates and policy**
  - **Authentication, Security**
  - **Permission and roles**

Certificates -> your ID
Policy -> your permission book



Create a thing

- 1. AWS IoT Menu
  - Things ➔ Create
- 2. Give a name



This step creates an entry in the thing registry and a thing shadow for your device.

Name

cse521

Apply a type to this thing

Using a thing type simplifies device management by providing consistent registry data for things that share a type. Types provide things with a common set of attributes, which describe the identity and capabilities of your device, and a description.

Thing Type

No type selected

Create a type

Add this thing to a group

Adding your thing to a group allows you to manage devices remotely using jobs.

Thing Group

Groups /

Create group Change

Set searchable thing attributes (optional)

Enter a value for one or more of these attributes so that you can search for your things in the registry.

Attribute key

Provide an attribute key, e.g. Manufacturer

Value

Provide an attribute value, e.g. Acme-Corporation

Clear

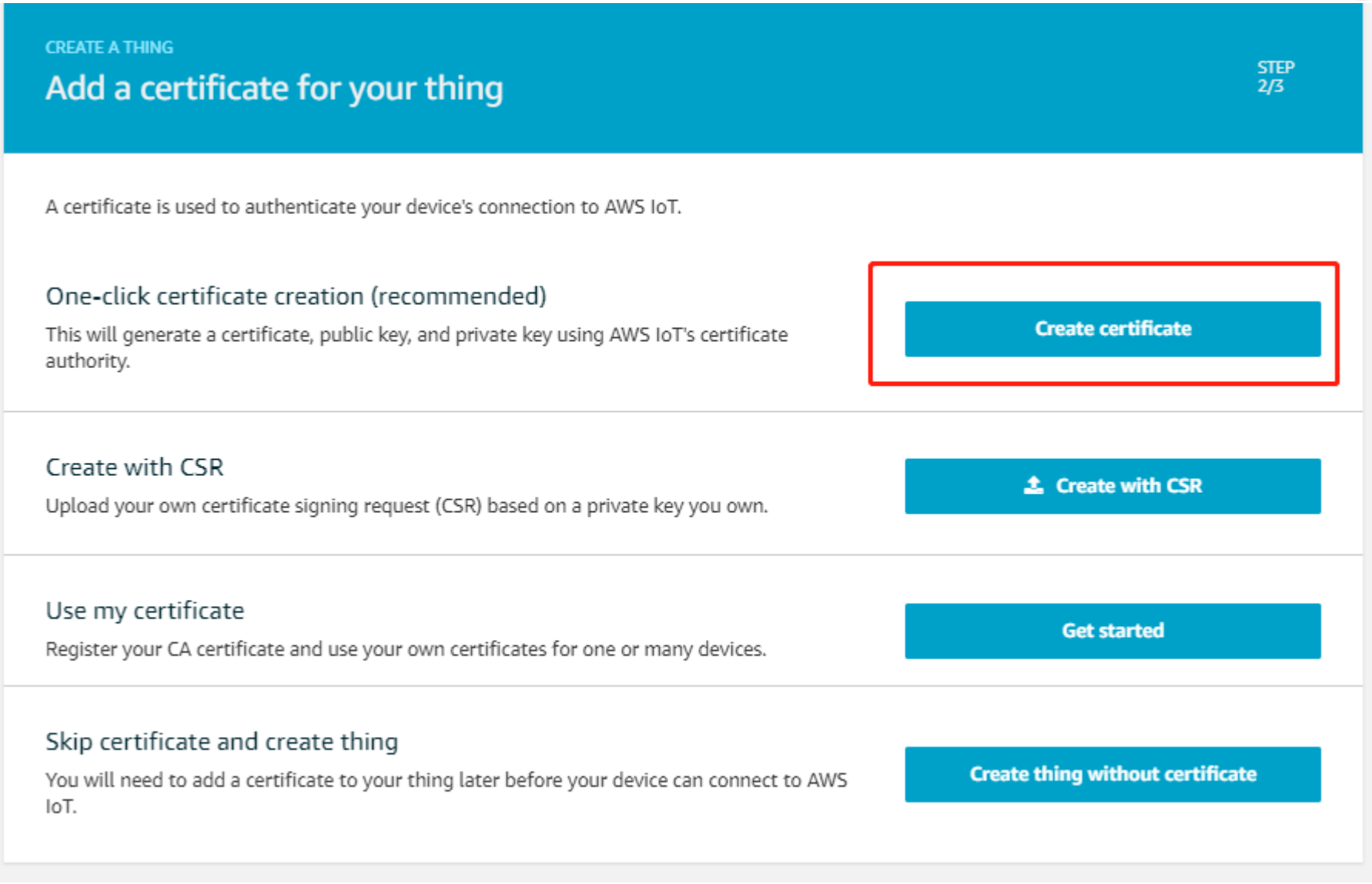
Add another

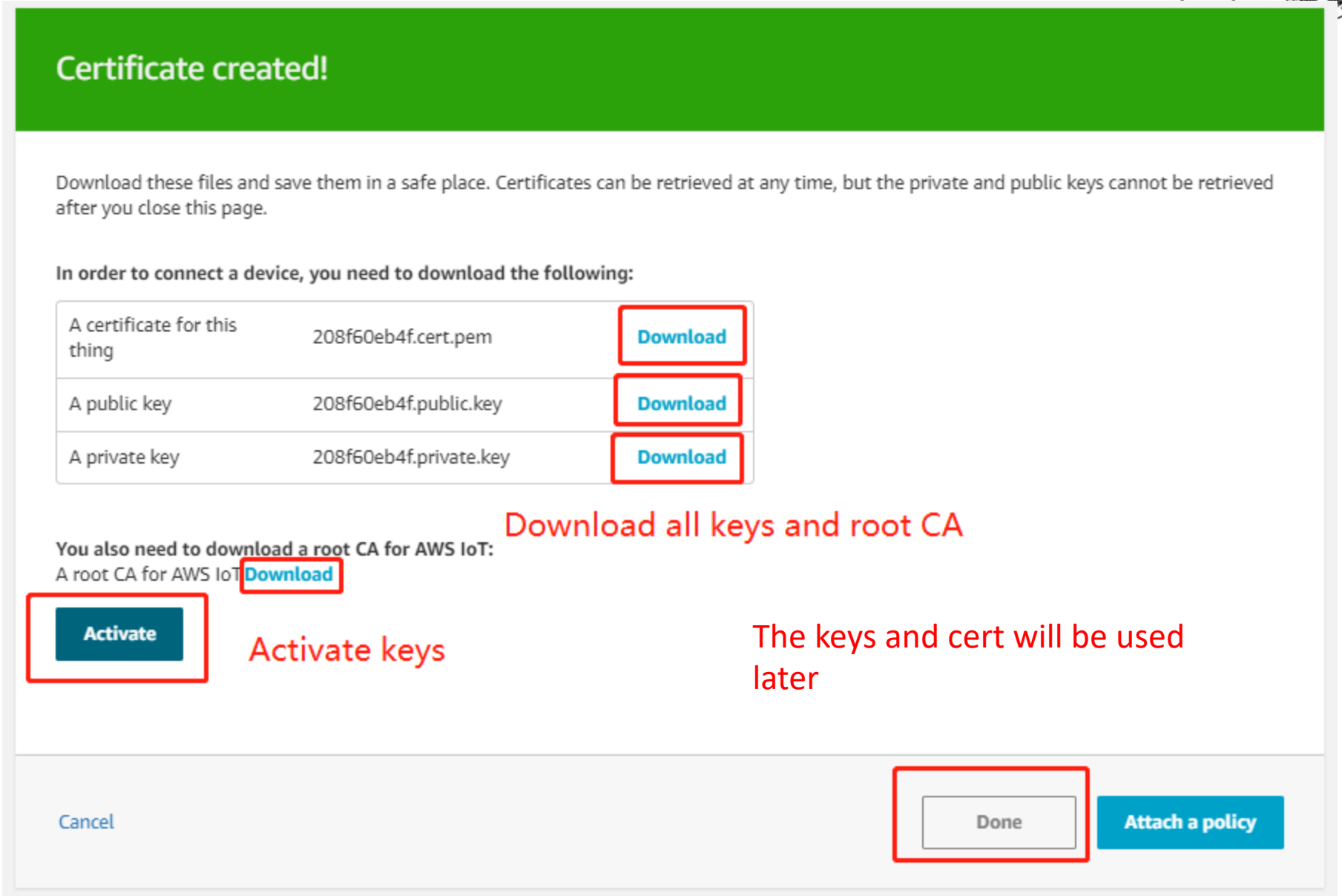
Show thing shadow

Cancel

Back

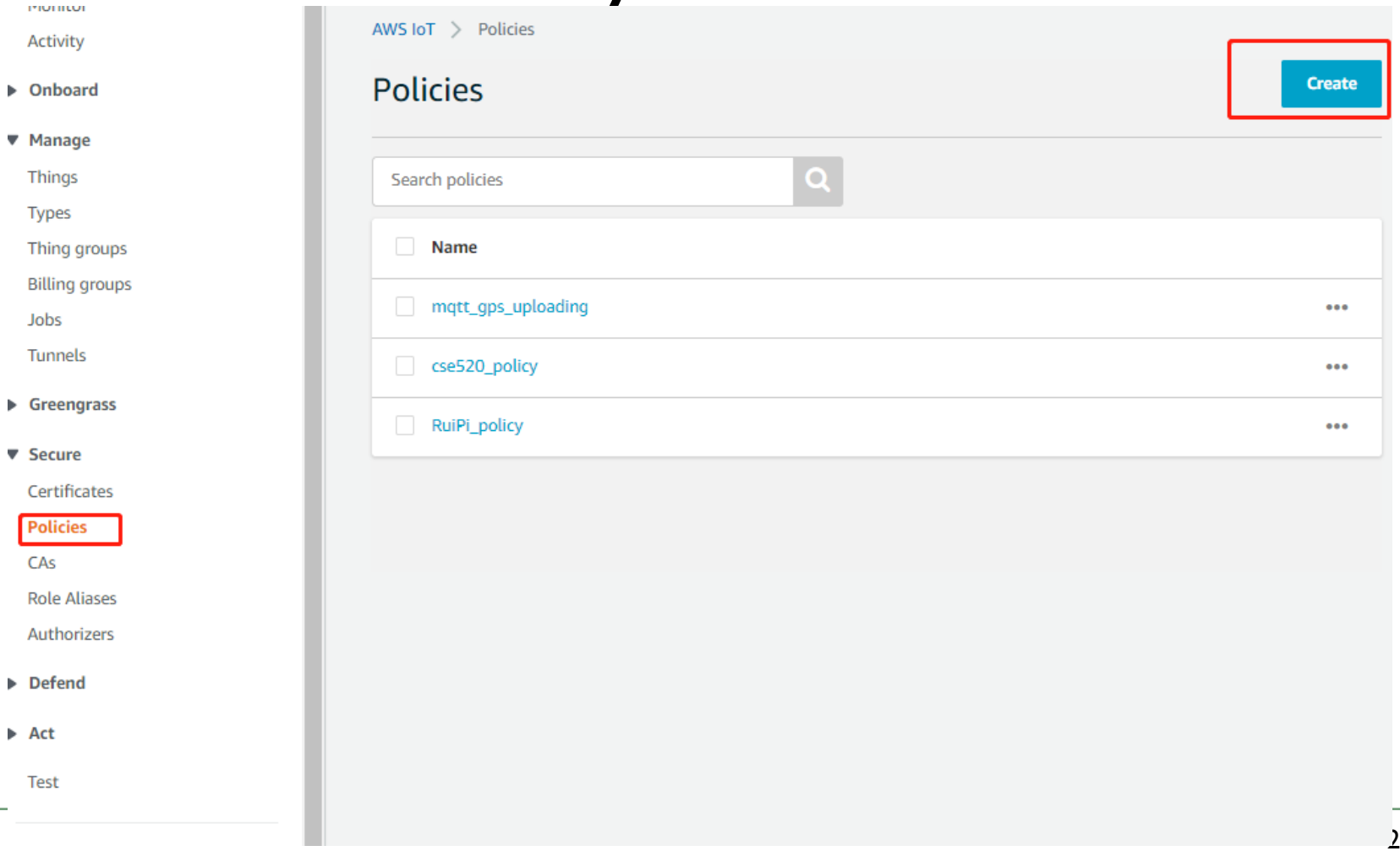
Next

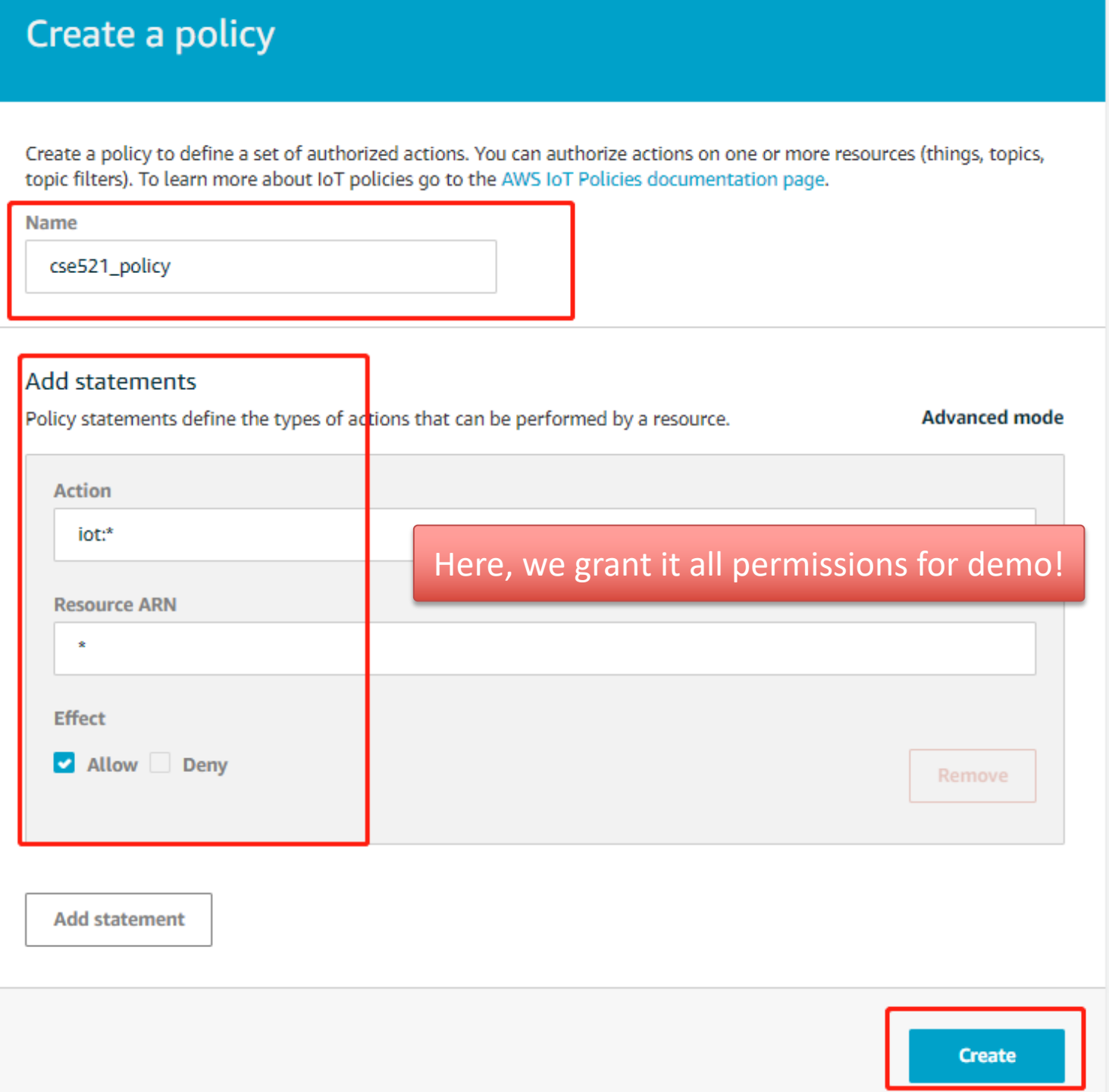

CREATE A THING
Add a certificate for your thing
STEP
2/3
A certificate is used to authenticate your device's connection to AWS IoT.
One-click certificate creation (recommended)
This will generate a certificate, public key, and private key using AWS IoT's certificate authority.
Create certificate
Create with CSR
Upload your own certificate signing request (CSR) based on a private key you own.
Create with CSR
Use my certificate
Register your CA certificate and use your own certificates for one or many devices.
Get started
Skip certificate and create thing
You will need to add a certificate to your thing later before your device can connect to AWS IoT.
Create thing without certificate

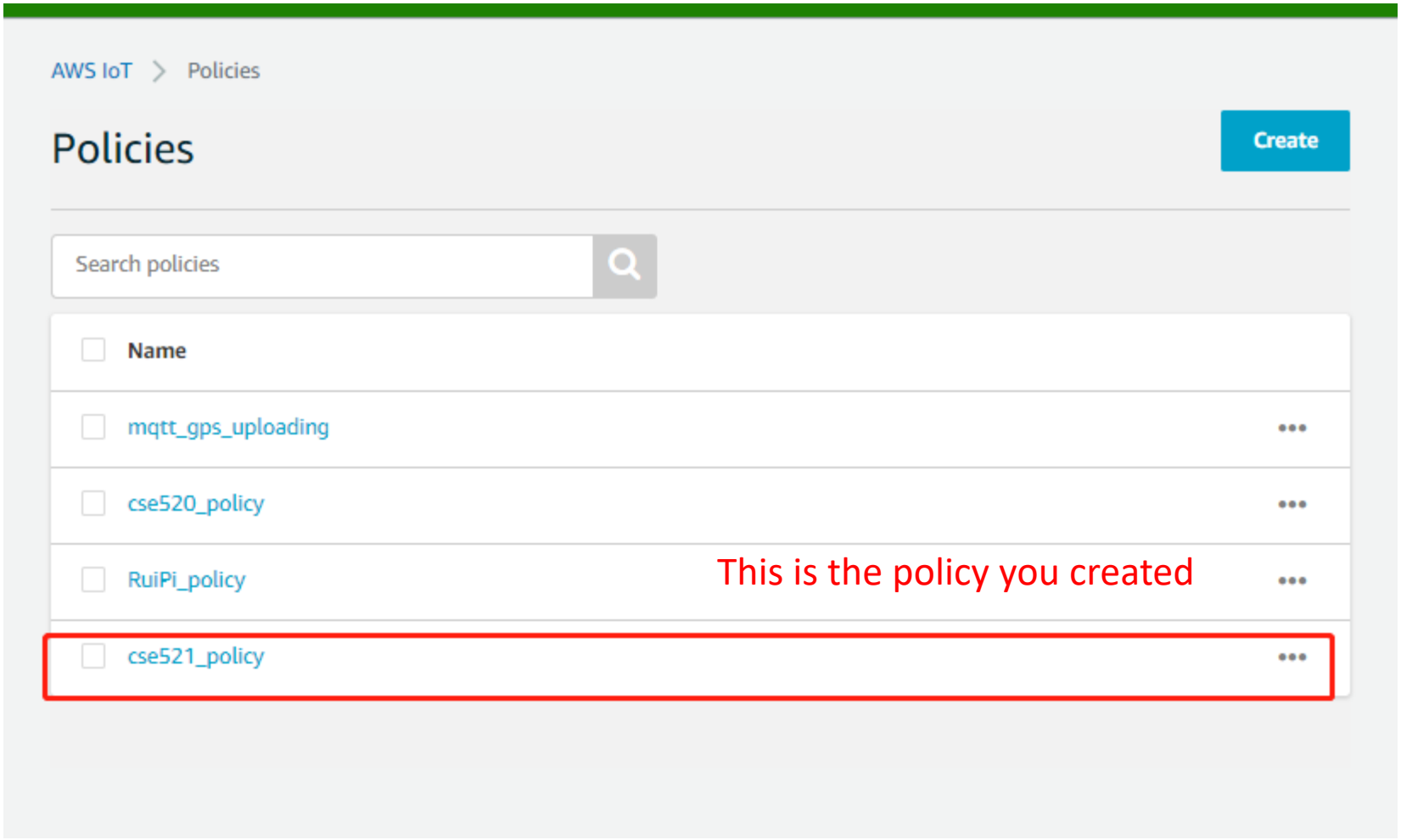

Certificate created!
Download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys cannot be retrieved after you close this page.
In order to connect a device, you need to download the following:
A certificate for this thing
208f60eb4f.cert.pem
Download
A public key
208f60eb4f.public.key
Download
A private key
208f60eb4f.private.key
Download
Download all keys and root CA
You also need to download a root CA for AWS IoT:
A root CA for AWS IoT Download
Activate
Activate keys
The keys and cert will be used later
Cancel
Done
Attach a policy

Create Policy

- A policy is attached to a key/cert
  - **It tells what this key/cert can do**

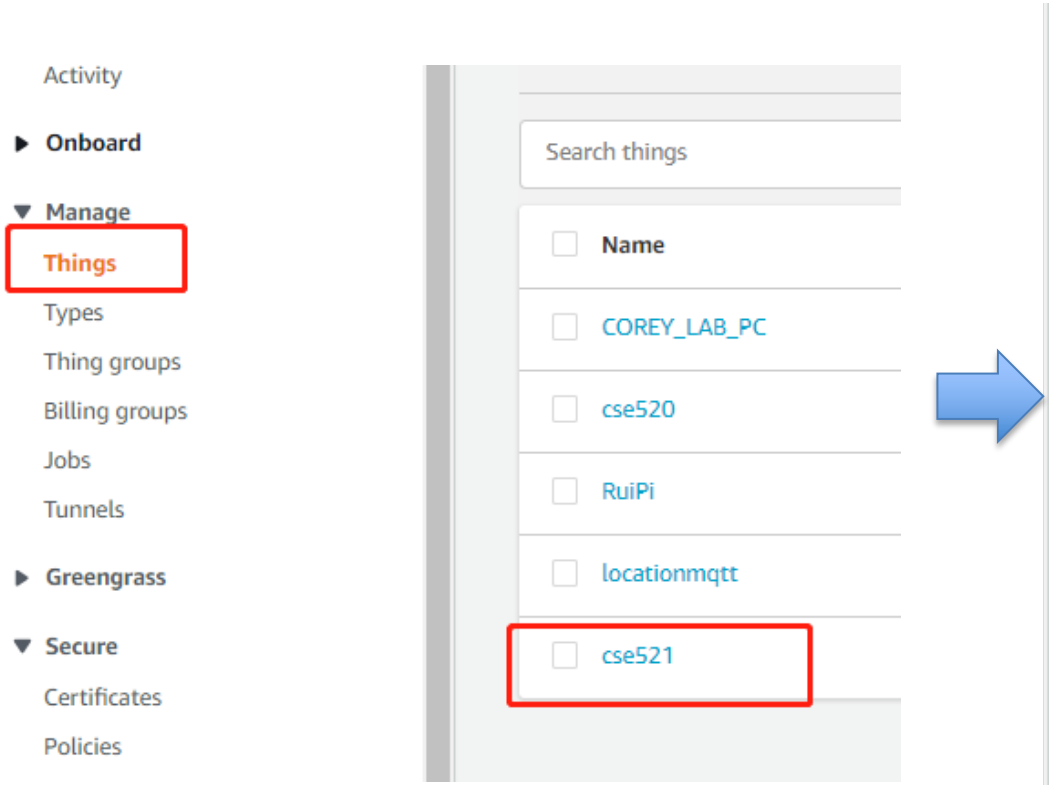


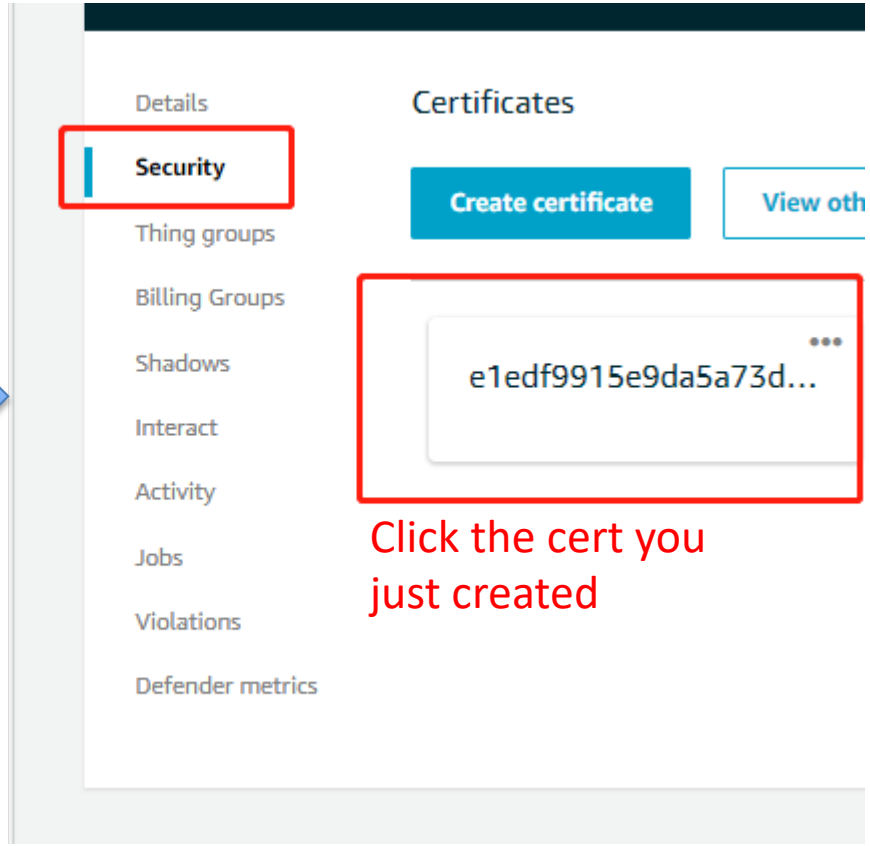

Create a policy
Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the AWS IoT Policies documentation page.
Name
cse521_policy
Add statements
Policy statements define the types of actions that can be performed by a resource.
Advanced mode
Action
iot:*
Here, we grant it all permissions for demo!
Resource ARN
*
Effect
Allow
Deny
Remove
Add statement
Create

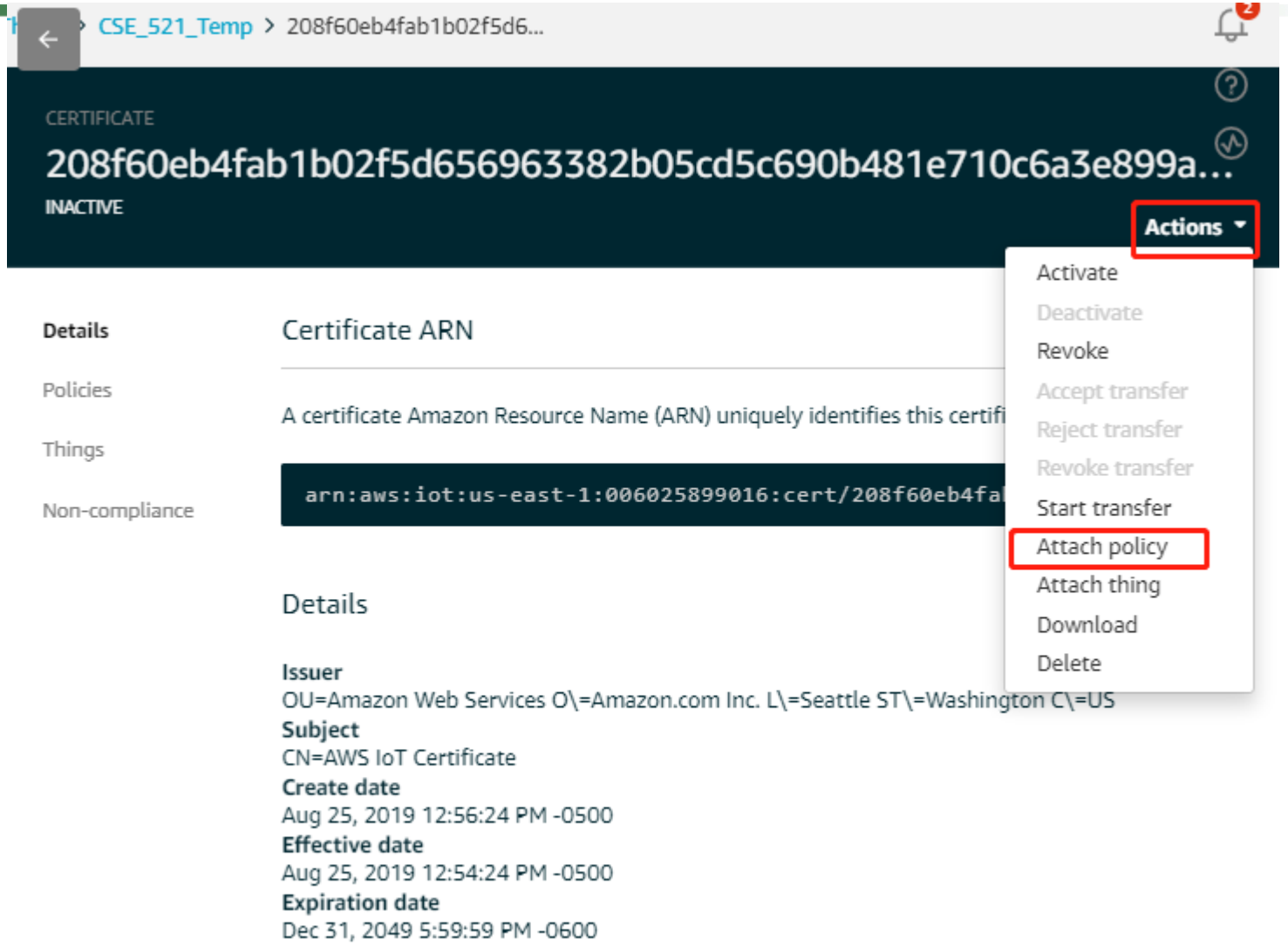

AWS IoT > Policies
Policies
Create
Search policies
Name
mqtt_gps_uploading
cse520_policy
RuiPi_policy
This is the policy you created
cse521_policy

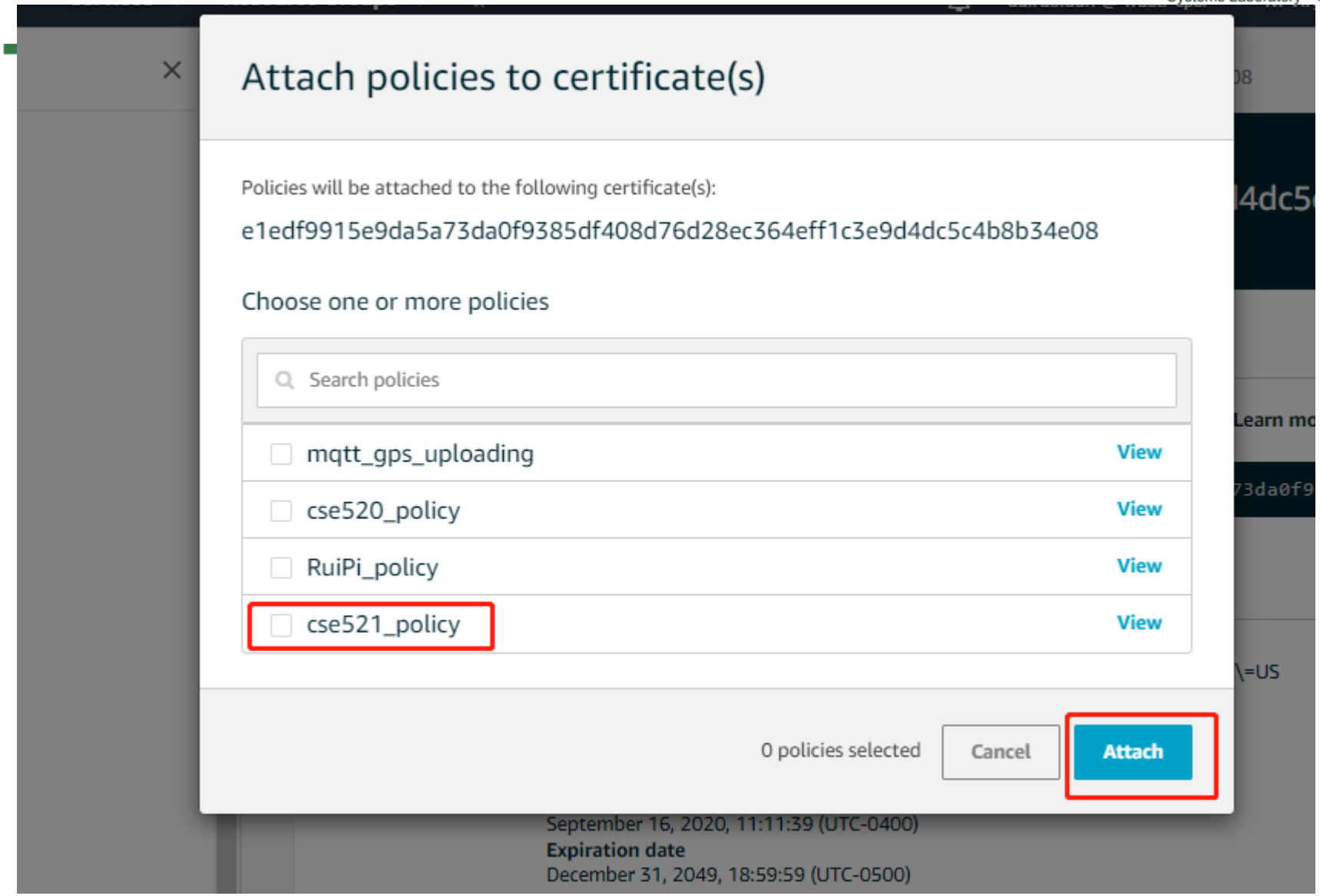
Attach Policy

- Attach Policy to the key/cert
  - A policy tells what this key/cert can do





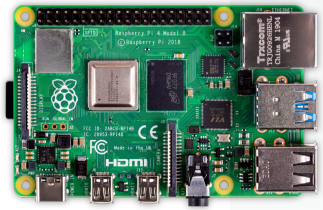

CSE_521_Temp > 208f60eb4fab1b02f5d6...
CERTIFICATE
208f60eb4fab1b02f5d656963382b05cd5c690b481e710c6a3e899a...
INACTIVE
Actions
Activate
Deactivate
Revoke
Accept transfer
Reject transfer
Revoke transfer
Start transfer
Attach policy
Attach thing
Download
Delete
Details
Policies
Things
Non-compliance
Certificate ARN
A certificate Amazon Resource Name (ARN) uniquely identifies this certifi
arn:aws:iot:us-east-1:006025899016:cert/208f60eb4fa
Details
Issuer
OU=Amazon Web Services O\=Amazon.com Inc. L\=Seattle ST\=Washington C\=US
Subject
CN=AWS IoT Certificate
Create date
Aug 25, 2019 12:56:24 PM -0500
Effective date
Aug 25, 2019 12:54:24 PM -0500
Expiration date
Dec 31, 2049 5:59:59 PM -0600

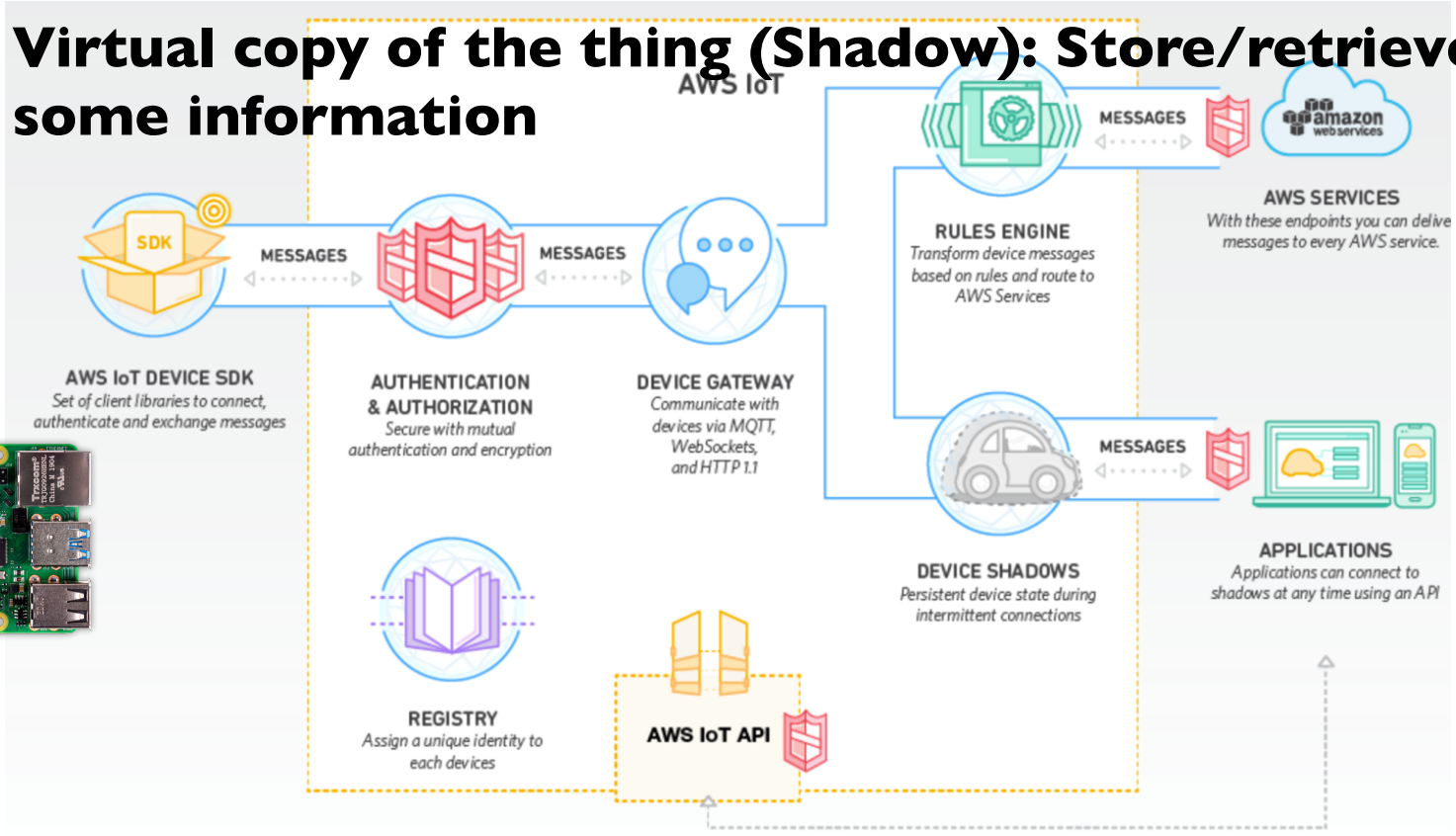

Attach policies to certificate(s)
Policies will be attached to the following certificate(s):
e1edf9915e9da5a73da0f9385df408d76d28ec364eff1c3e9d4dc5c4b8b34e08
Choose one or more policies
Search policies
mqtt_gps_uploading
View
cse520_policy
View
RuiPi_policy
View
cse521_policy
View
0 policies selected
Cancel
Attach
September 16, 2020, 11:11:39 (UTC-0400)
Expiration date
December 31, 2049, 18:59:59 (UTC-0500)

Now, you have a virtual thing on AWS!

AWS Things Summary

- **Certificate: authenticate the device**
- **Policy: define the roles/permissions of the certificate**
- **Virtual copy of the thing (Shadow): Store/retrieve some information**





CPSL
Cyber-Physical Systems Laboratory

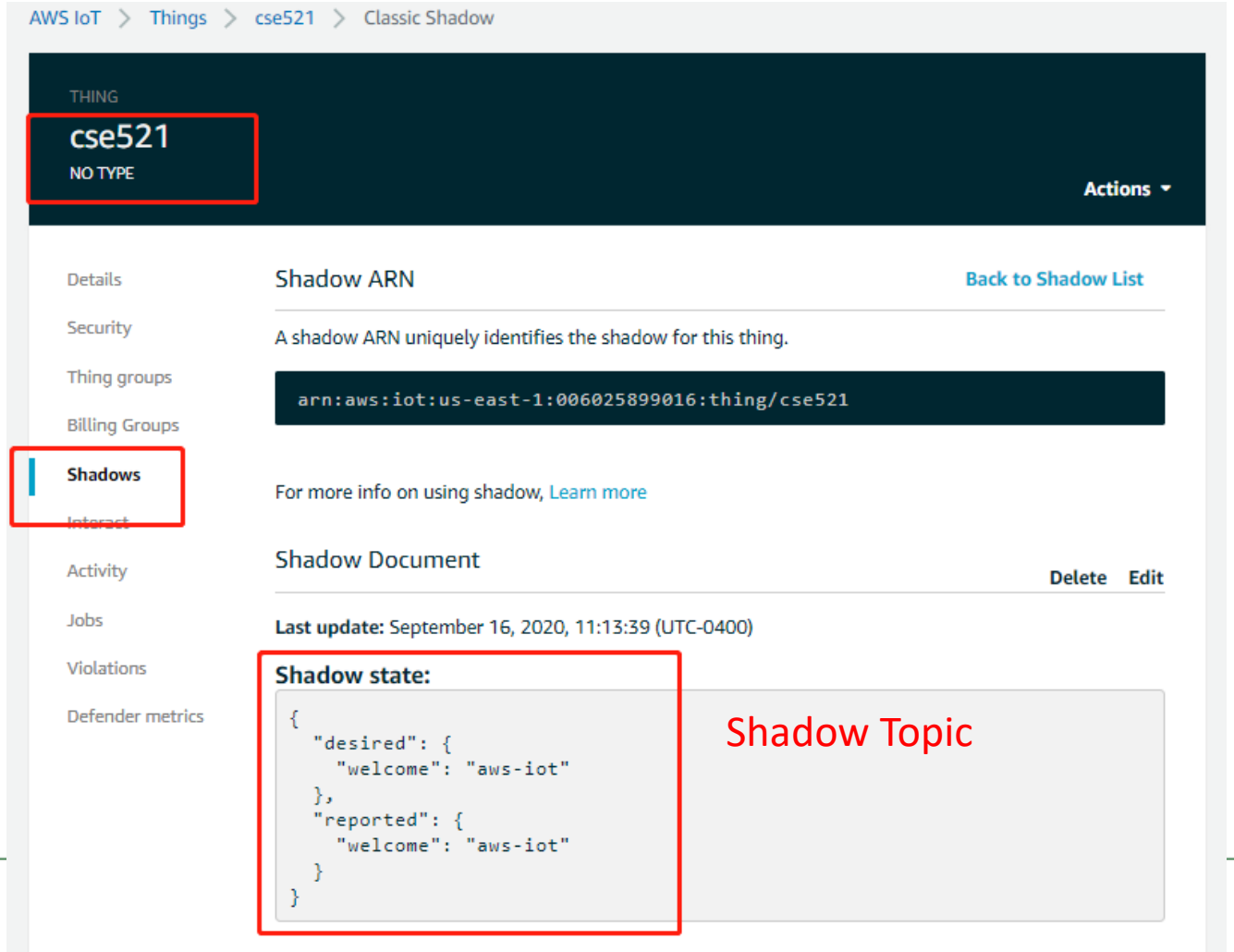
Let's test it online!

Washington University in St. Louis

Basic Interact: Publish to the "Shadow"

- Get your "Shadow"
  - In your Thing Page

Shadow is a special topic



Find your "Shadow" Topic

- Topic: can be seen as the "address"

Shadow Topic:
$aws/things/*cse521*/shadow/update

Device Shadow MQTT topics

PDF | Kindle

The Device Shadow service uses reserved MQTT topics to enable devices and apps to get, update, or delete the state information for a device (shadow).

Publishing and subscribing on shadow topics requires topic-based authorization. AWS IoT reserves the right to add new topics to the existing topic structure. For this reason, we recommend that you avoid wild card subscriptions to shadow topics. For example, avoid subscribing to topic filters like `$aws/things/thingName/shadow/#` because the number of topics that match this topic filter might increase as AWS IoT introduces new shadow topics. For examples of the messages published on these topics see Interacting with shadows.

Shadows can be named or unnamed (classic). The topics used by each differ only in the topic prefix. This table shows the topic prefix used by each shadow type.

ShadowTopicPrefix value	Shadow type
$aws/things/*thingName*/shadow	Unnamed (classic) shadow
$aws/things/*thingName*/shadow/name/*shadowName*	Named shadow

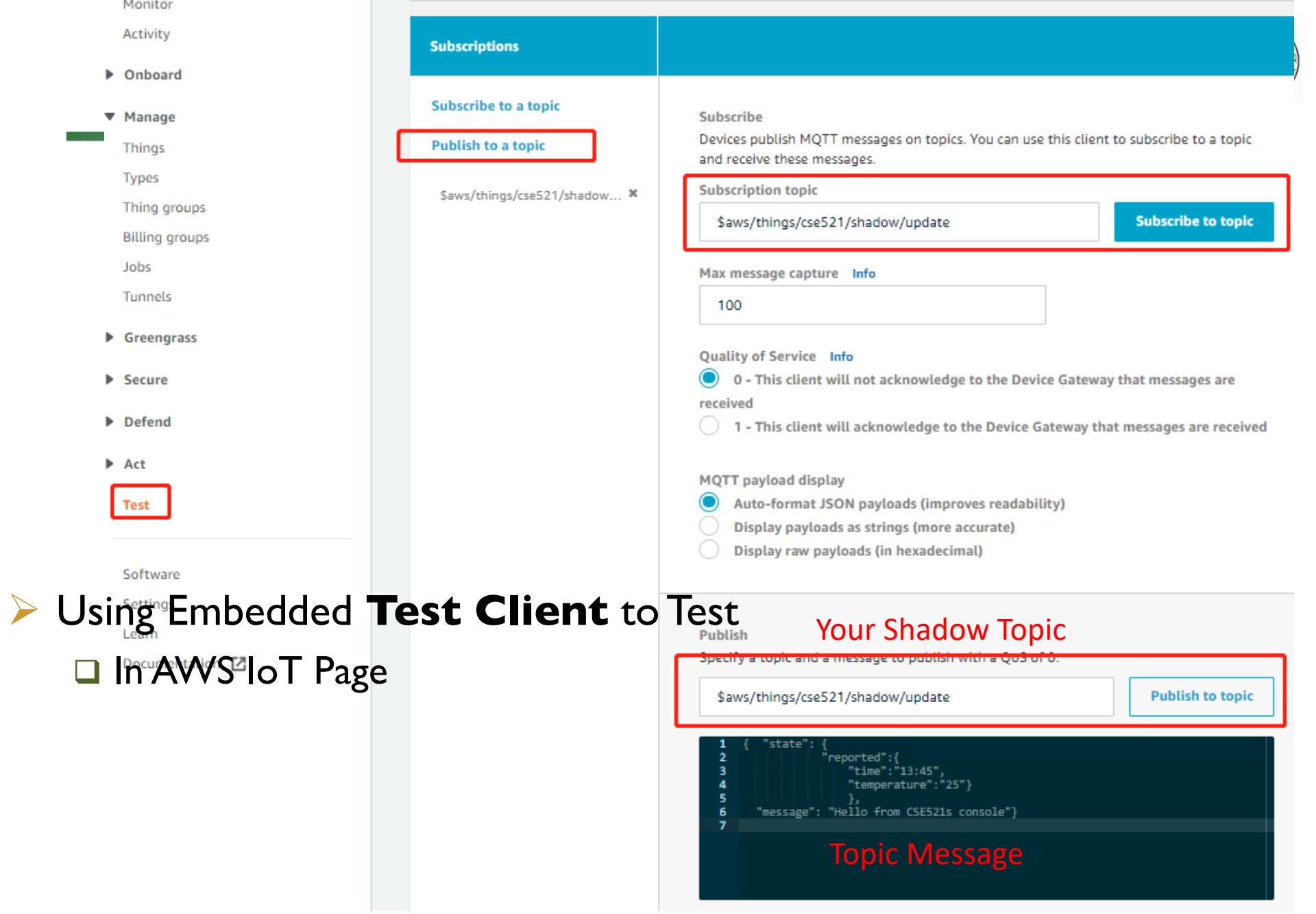
To create a complete topic, select the *ShadowTopicPrefix* for the type of shadow to which you want to refer, replace *thingName*, and *shadowName* if applicable, with their corresponding values, and then append that with the topic stub as shown in the following sections.

/update

Publish a request state document to this topic to update the device's shadow:

```
ShadowTopicPrefix/update
```


https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-mqtt.html?icmpid=docs_iot_console



- Using Embedded **Test Client** to Test
  - In AWS IoT Page

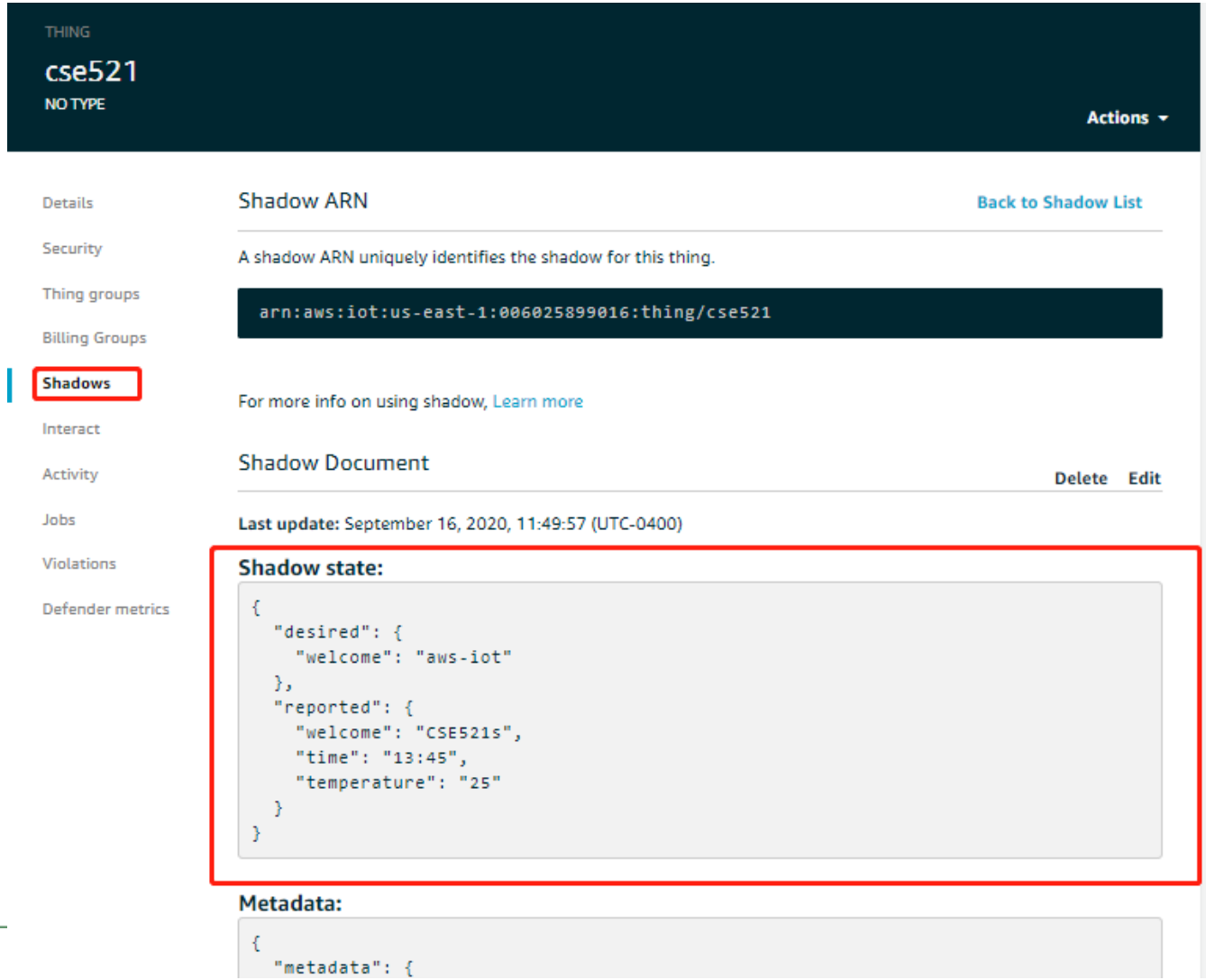
Shadow Message


```
{  "state": {
          "reported":{
           "welcome": "CSE521s",
             "time":"13:45",
             "temperature":"25"}
             },
  "message": "Hello from CSE521s console"}
```

A Shadow Message is a JSON object.
Shadow message has strict formats.
Please see
https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-document-syntax.html

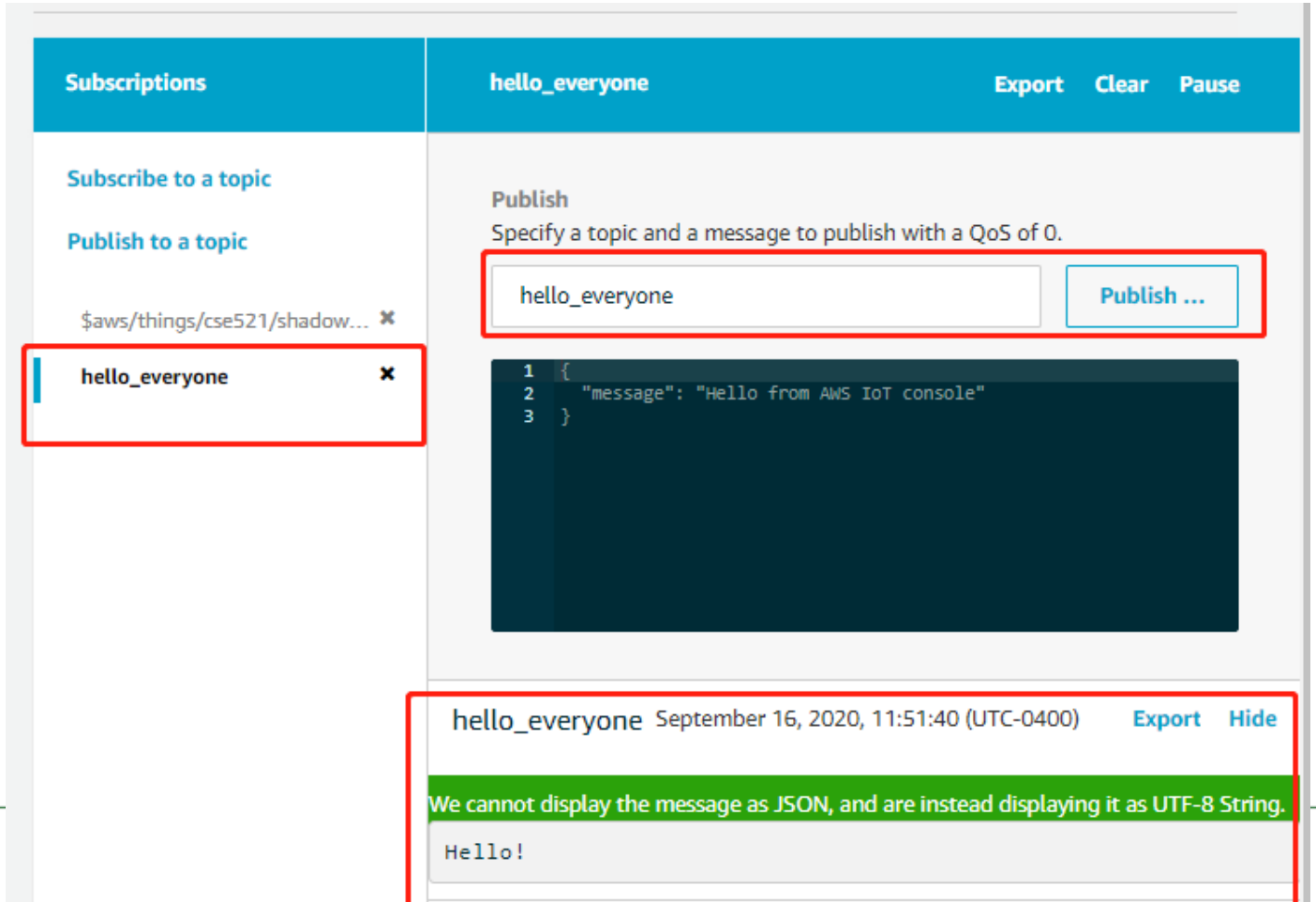
Update Shadow

- In your "Thing" Page

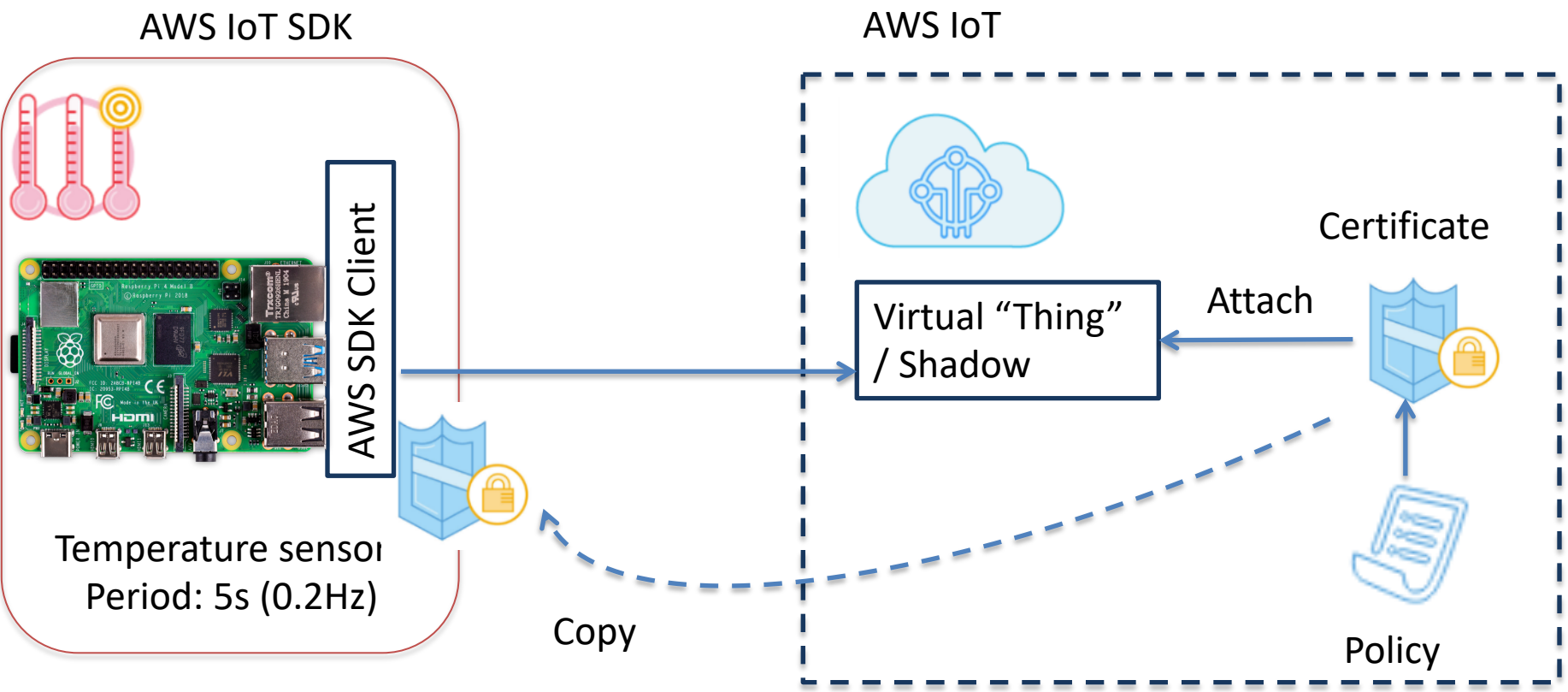


Basic Interact: Subscribe/Publish

- You can define your own Topic
- Once you have a subscriber that is subscribed to the topic, the subscriber can receive the message

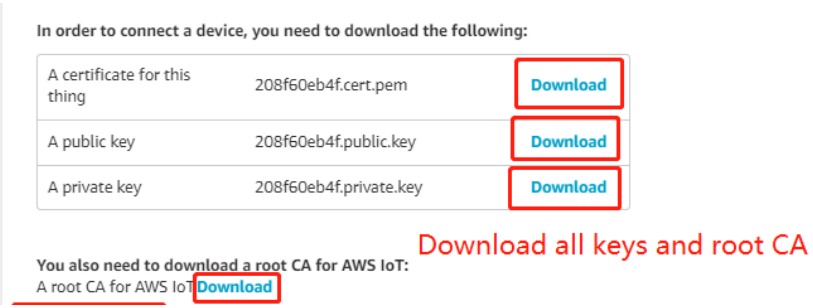


Step 2: Connect a “Physical” Device



Connect your Device

- Copy certificates to your **physical things**
  - Downloaded before!



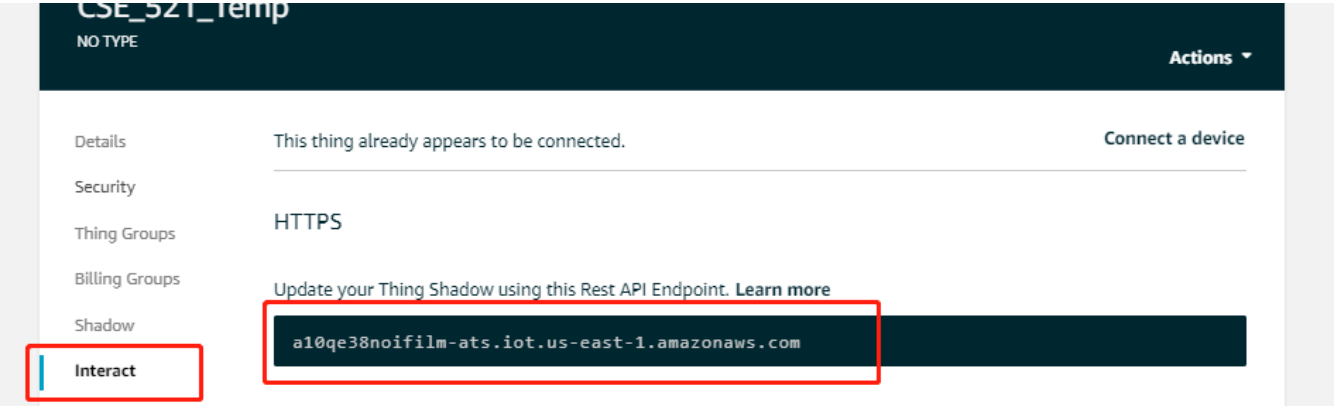
Download all keys and root CA

- Choose your **AWS SDK** (support MQTT)
  - Node JS
  - Python (pip install AWSIoTPythonSDK)
  - Java
  - …
- Set up your client with SDK and the certificates

Some Notes

- 1. Copy the certificates/keys to your real thing

- 2. You will need the endpoint and port (8883)



Host(Endpoint)

- 3. Set up your configuration of the code with SDK

```
host = "a10qe38noifilm-ats.iot.us-east-1.amazonaws.com" # Your thing's endpoint. See tutorial slides
rootCAPath = "root-CA.crt"
certificatePath = "e1edf9915e-certificate.pem.crt"
privateKeyPath = "e1edf9915e-private.pem.key"
port = 8883
clientId = "CSE521"
topic = "$aws/things/cse521/shadow/update" # Shadow topic of your Thing
```

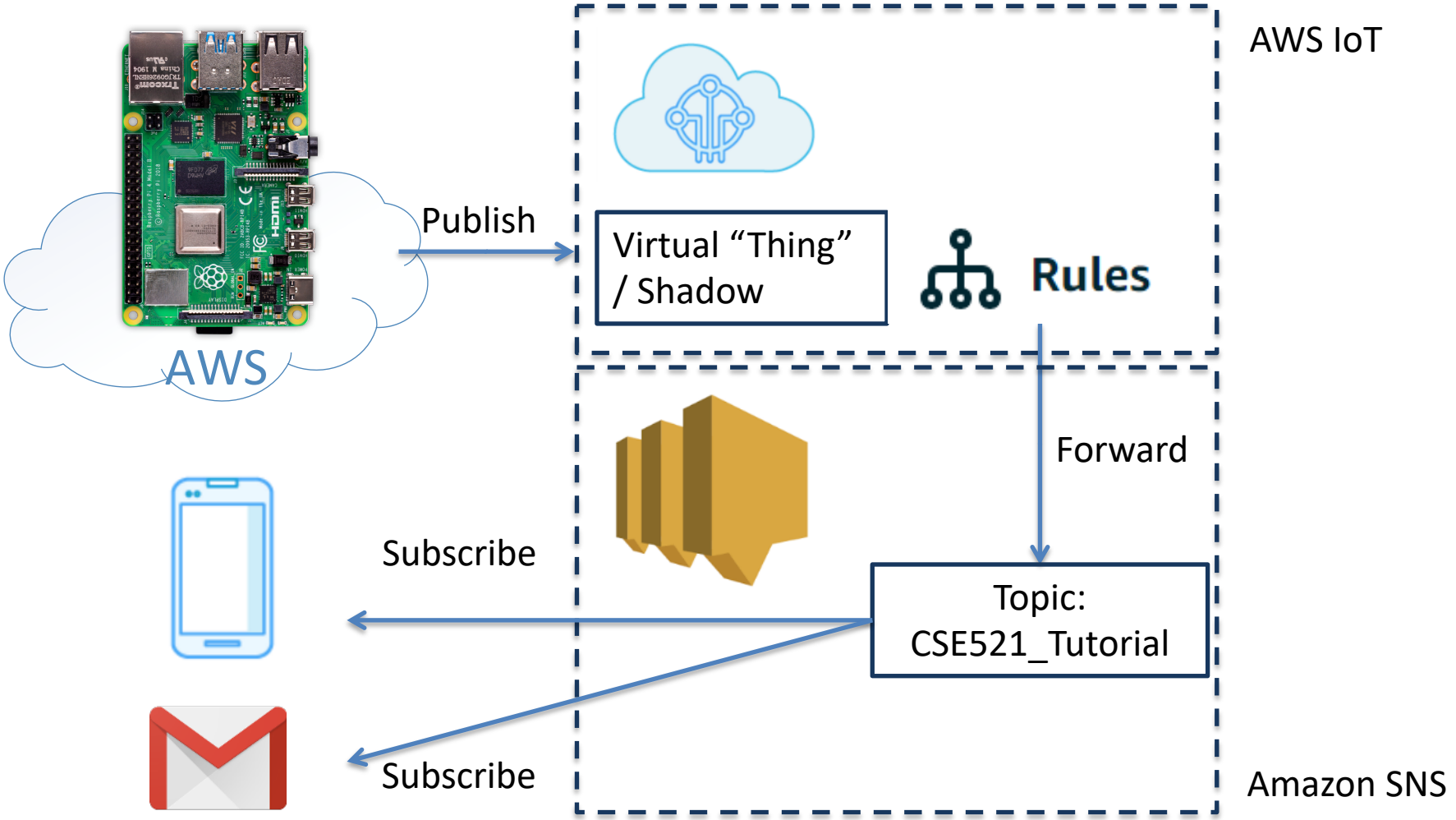
Change your code accordingly!

SDK and Demo Codes

- https://github.com/aws/aws-iot-device-sdk-python
- https://github.com/aws/aws-iot-device-sdk-python/blob/master/samples/basicPubSub/basicPubSub.py

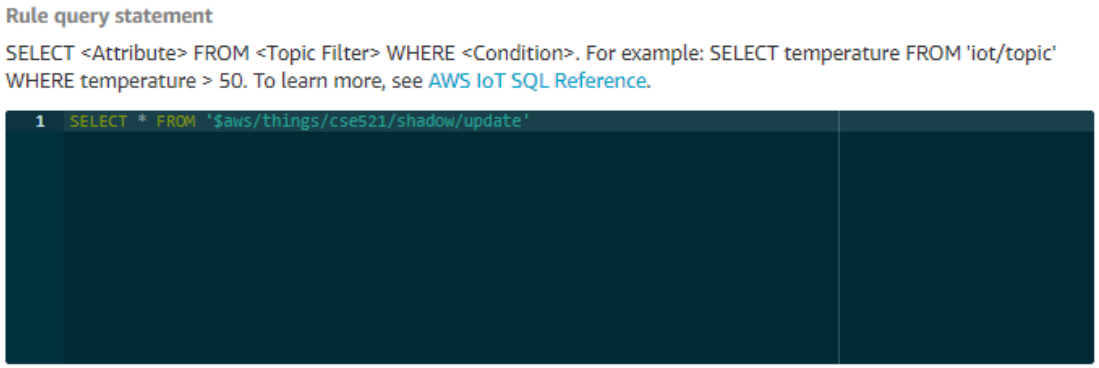
More: Rule Engine, Link with SNS services

- Simple Notification Service

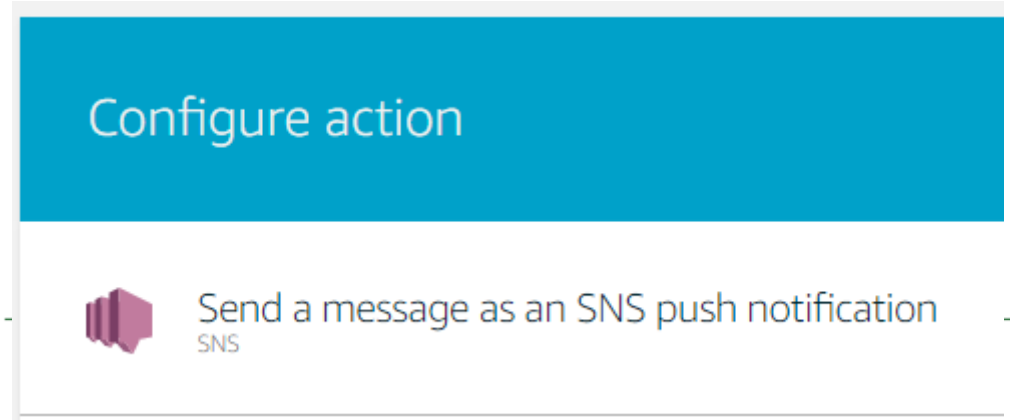


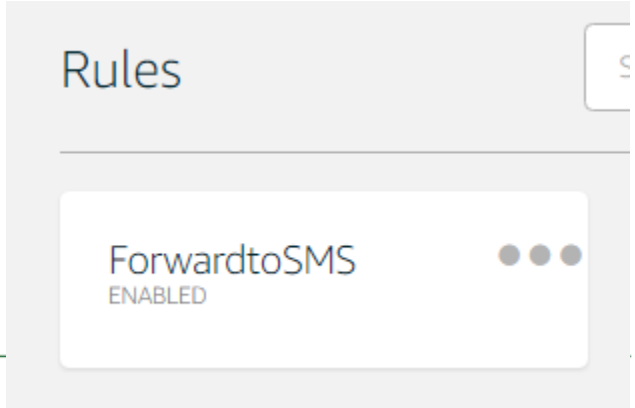
Create a Rule in Amazon IoT

- Add a query to filter your interesting topic (event)



- Add an Action:
  - Forward this message to SNS
  - Specify Dest ARN
  - Enable Rule





Notification on SMS & Email

AWS Notification Message

Inbox x

520Tutor no-reply@sns.amazona 3:11 PM (28 minutes ago)

to me

{"state":{"reported":{"Value":45}},"metadata":{"reported":{"Value":{"timestamp":1503951070}}},"version":134,"timestamp":1503951070}

--

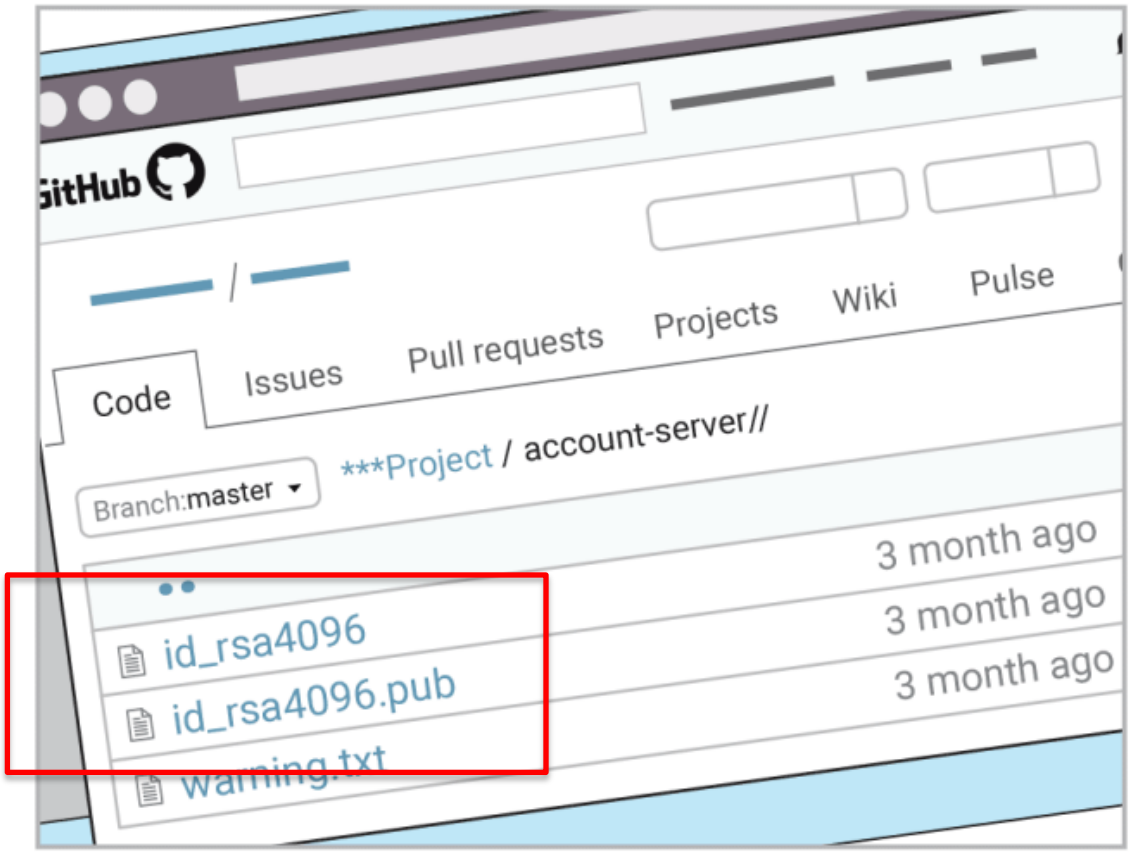
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-west-2.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-west-2:401317363811:CSE520S_Tutorial:00c54352-7d1a-4c09-9cc1-15aed3c395e3&Endpoint=naroahlee@gmail.com

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at
https://aws.amazon.com/support

One More Thing: Account Security

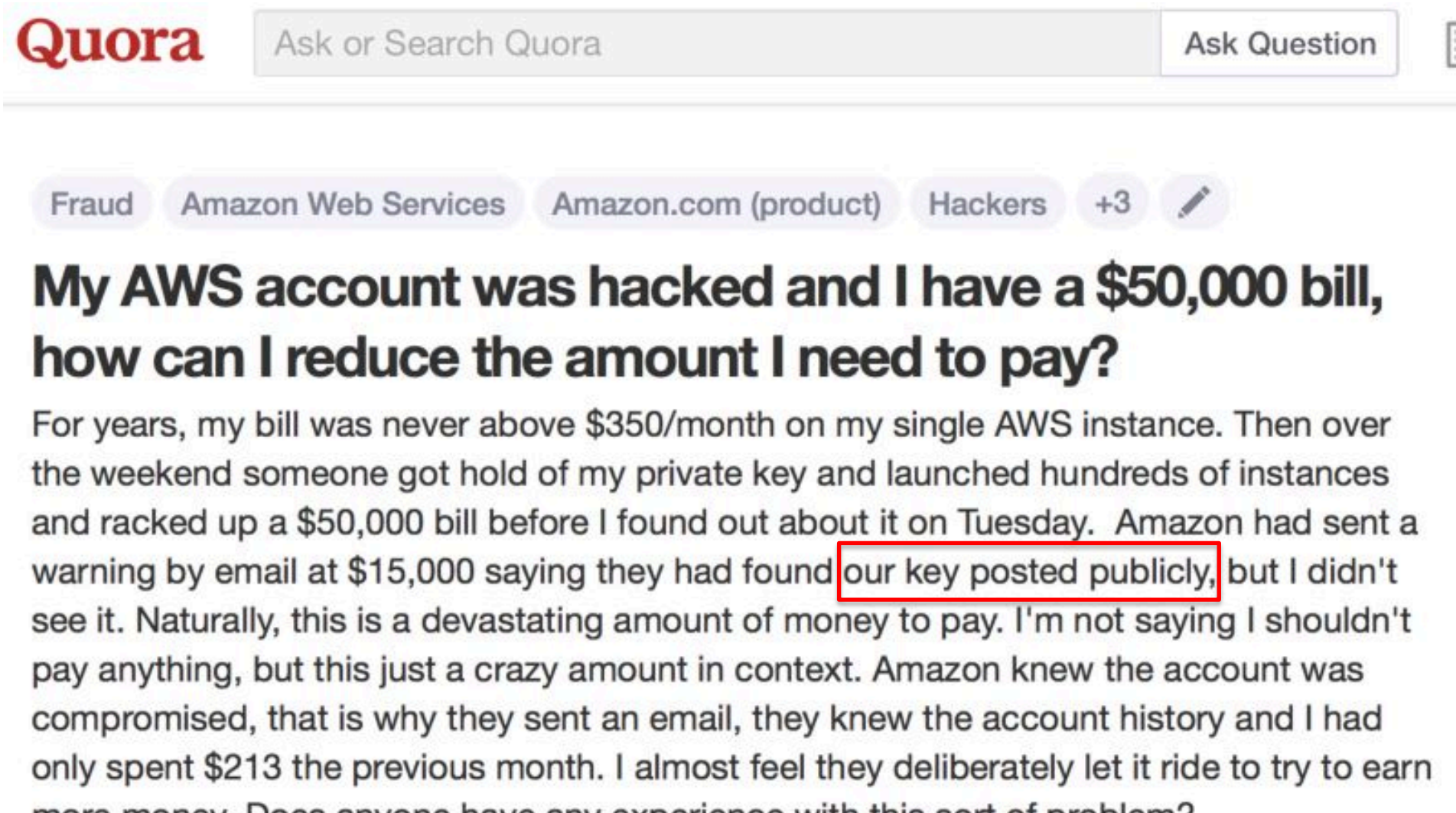
- DON'T UPLOAD YOUR KEY PUBLICLY!!!

Time to Open Source!



Source: WeChat Subscriptions: 西乔《神秘的程序员们 39》 Geek Life Chpt 39.

What if… $50,000 AWS Bill!



Source: https://www.quora.com/My-AWS-account-was-hacked-and-I-have-a-50-000-bill-how-can-I-reduce-the-amount-I-need-to-pay

Some project examples

CPSL Cyber-Physical Systems Laboratory

- Gesture recognition with smartwatch
  - Recognize the specific gesture to control the light
- Smart pet feeder
  - Food dispenser with schedules and smart control
- Smart mirror
  - Show personalized info in the mirror

Sensing, Connecting, Smart

If you have any question about the project, feel free to send me an Email

CPSL
Cyber-Physical Systems Laboratory

Thanks!

Ruixuan Dai

Washington University in St.Louis