

Cloud Tutorial: AWS IoT

CSE 520S Spring,
Jan. 16, 2020
Ruixuan Dai



Washington University in St. Louis

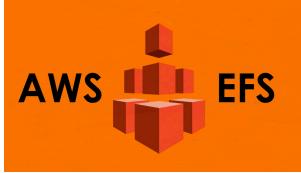
XaaS: Basics in Cloud Computing



Washington University in St. Louis

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

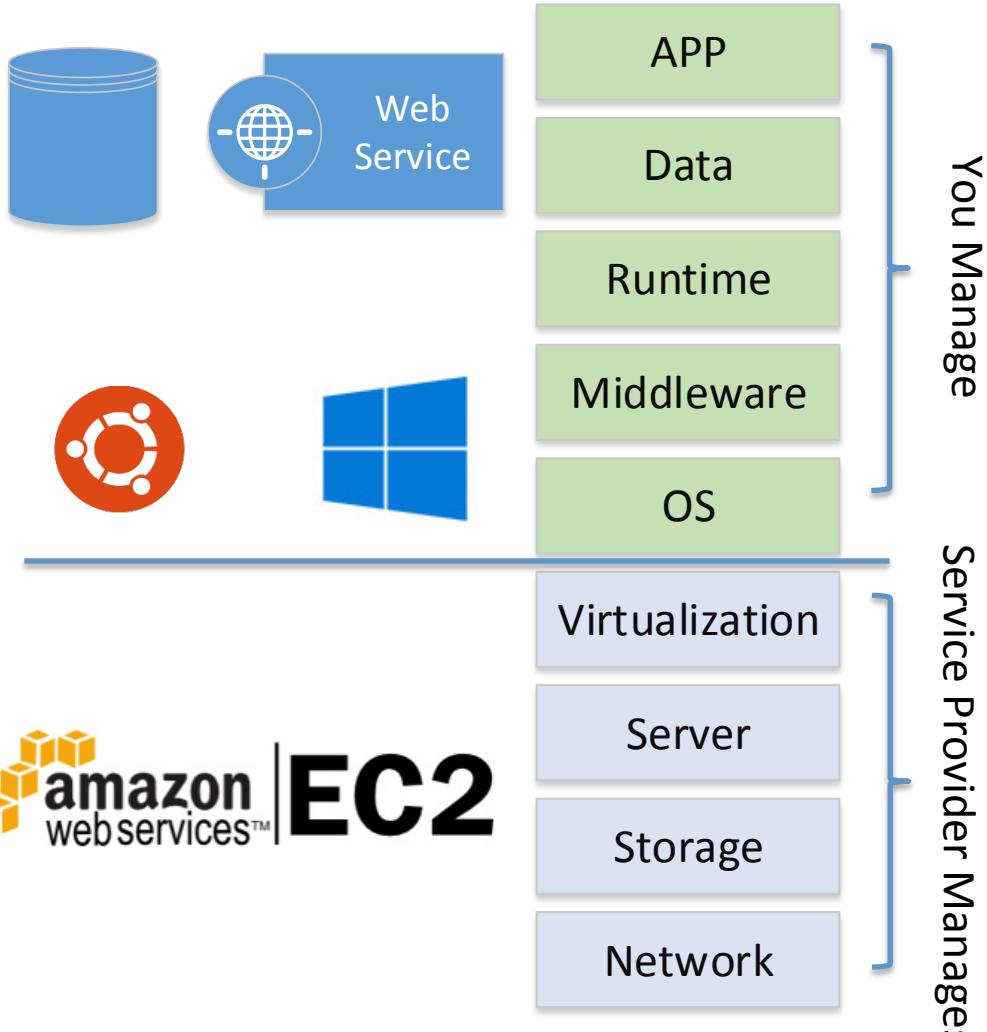
- Traditional
- installed and runs on personal computer
- You Manage and Deploy
 - ❑ Hardware
 - ❑ OS
 - ❑ Software
- Example
 - ❑ This presentation



Infrastructure as a Service (IaaS)

➤ IaaS

- ❑ "physical server box"
- ❑ Virtual Machine
 - Memory
 - Storage
 - CPU
 - Network



➤ Example

- ❑ AWS EC2
- ❑ AWS EFS

➤ Use case

- ❑ Build up your VM cluster

Platform as a Service (PaaS)

➤ PaaS

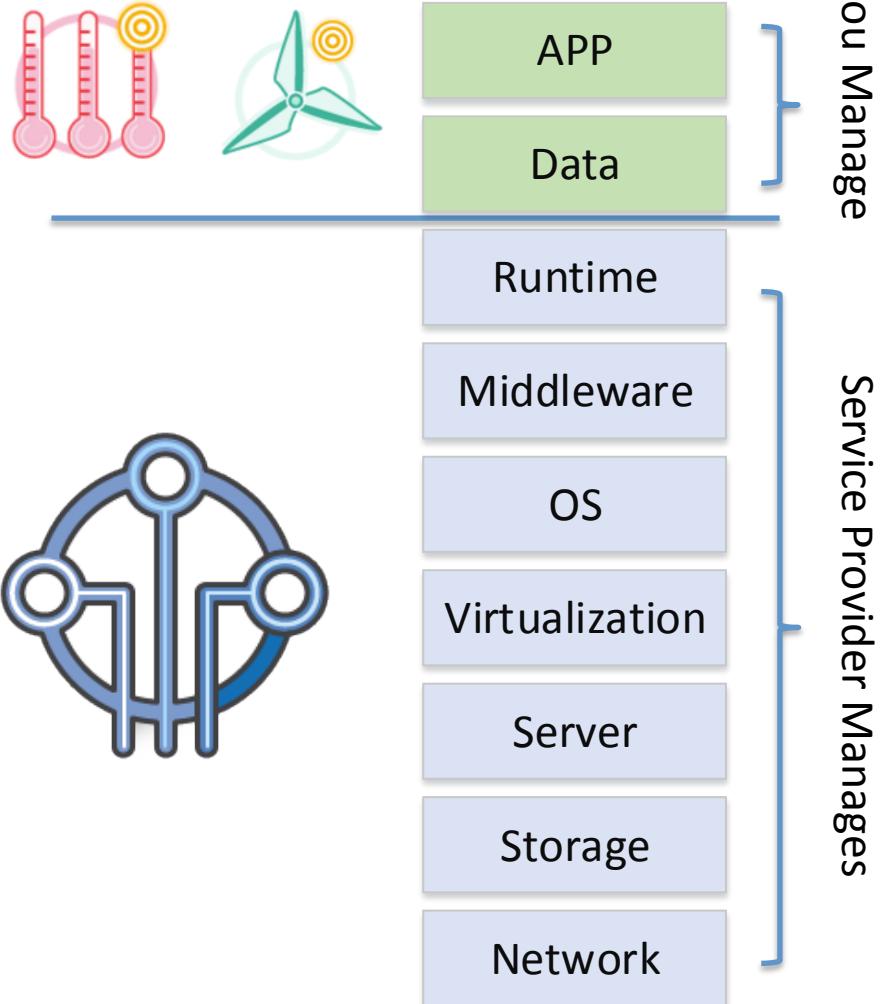
- ❑ You get a framework
- ❑ Host Application
- ❑ Tools

➤ Example

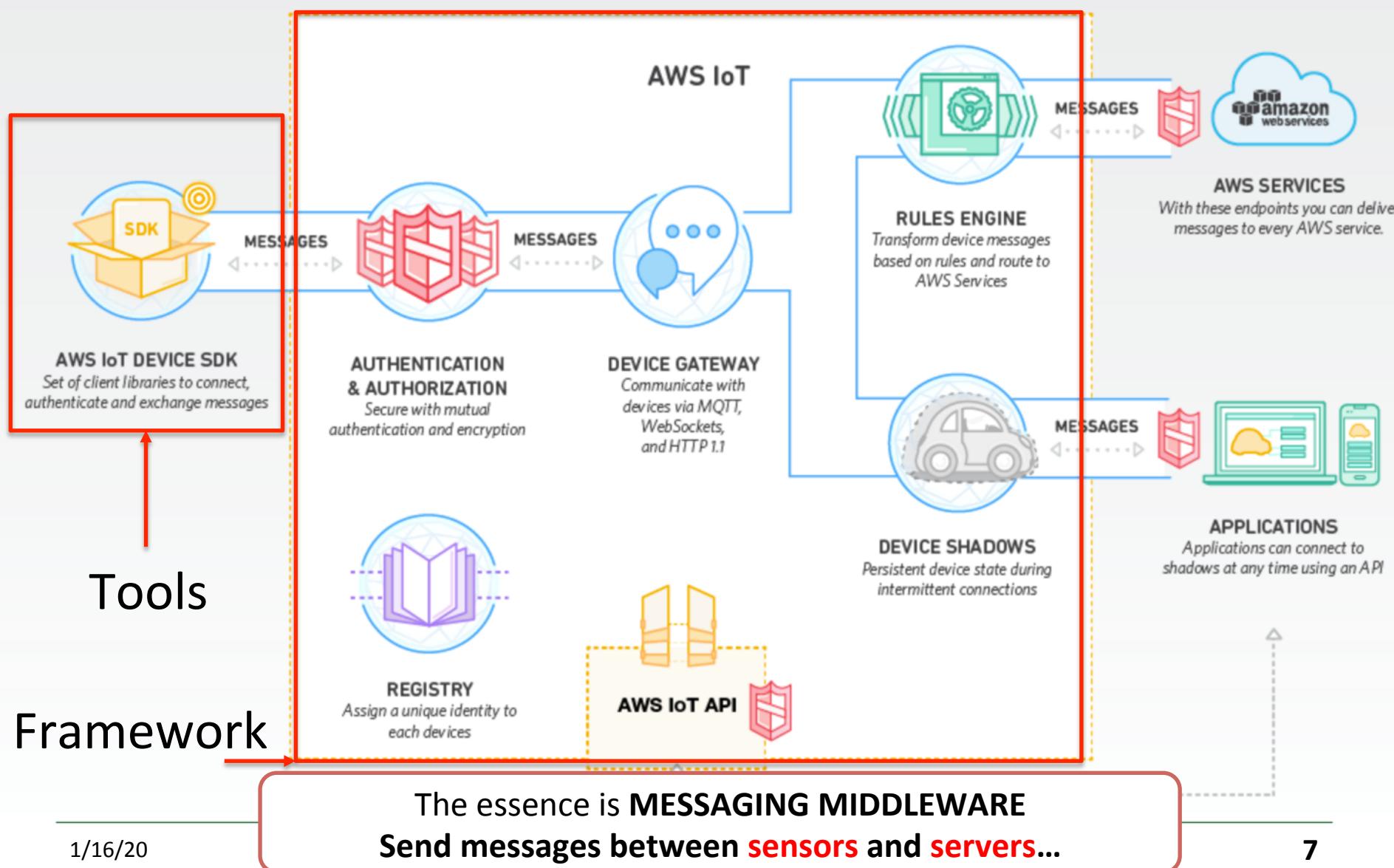
- ❑ AWS IoT

➤ Use case

- ❑ Build up your smart A/C controller



PaaS Example: AWS IoT



Software as a Service (SaaS)

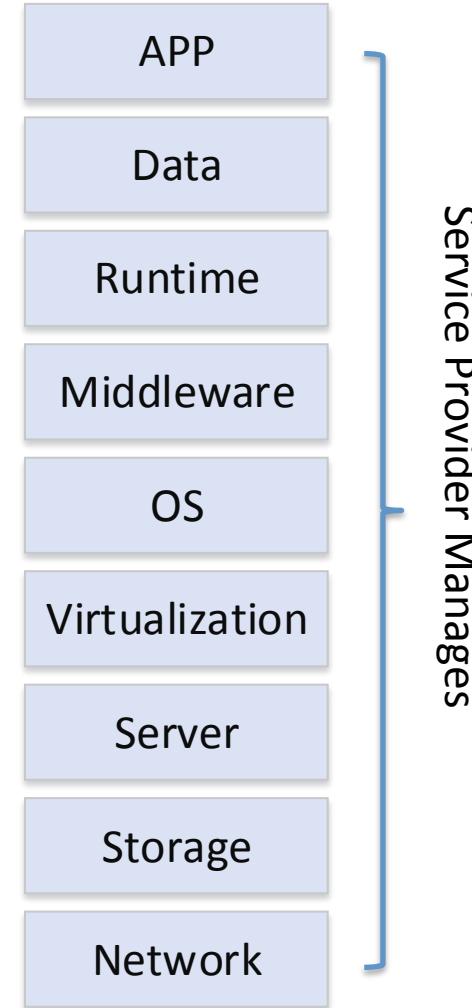
➤ SaaS

- ❑ You get a whole solution

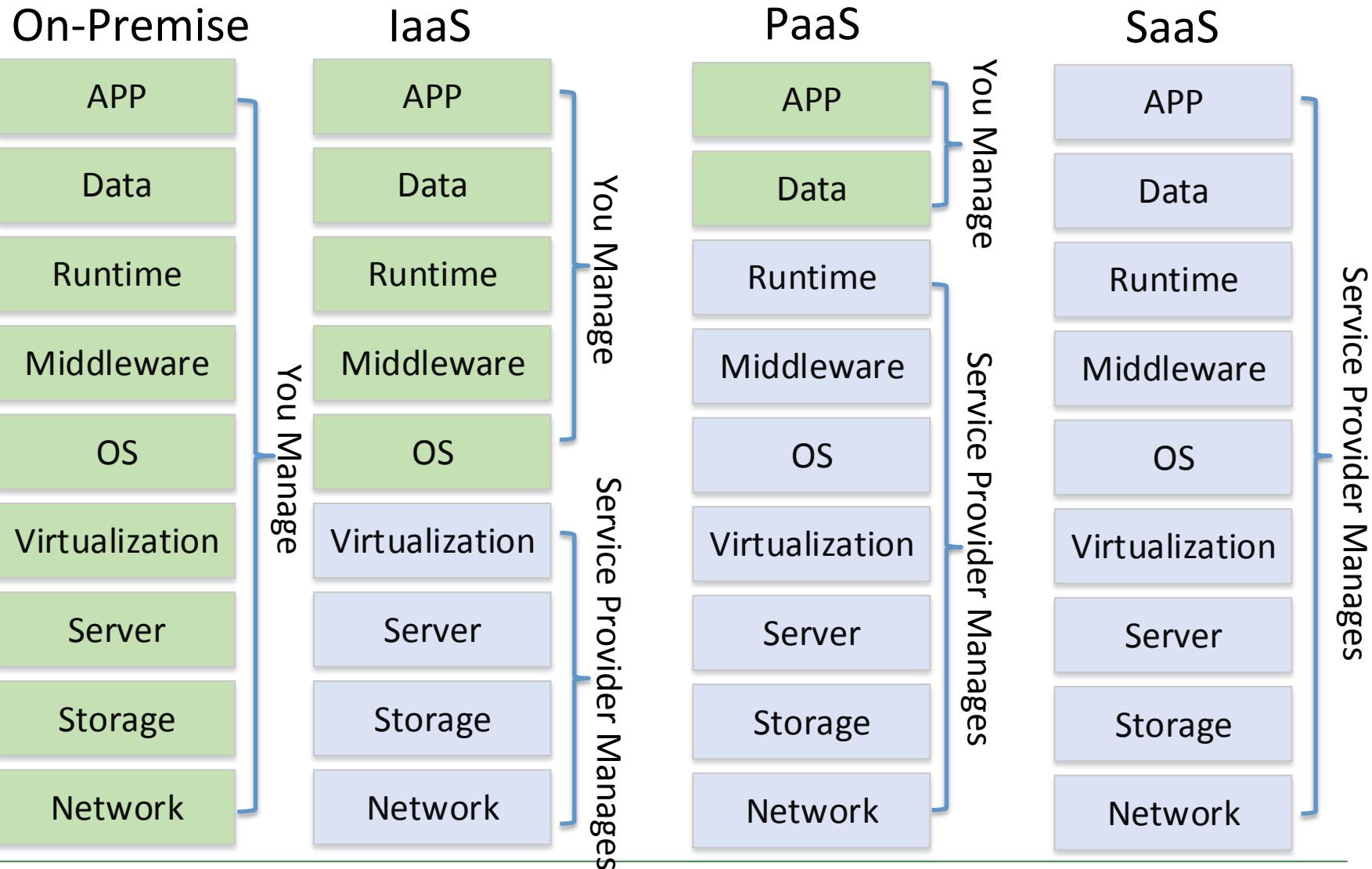


➤ Example

- ❑ Gmail
- ❑ Dropbox
- ❑ Office365



XaaS: A Recap



Tutorial: Hello! AWS IoT!!



Washington University in St. Louis

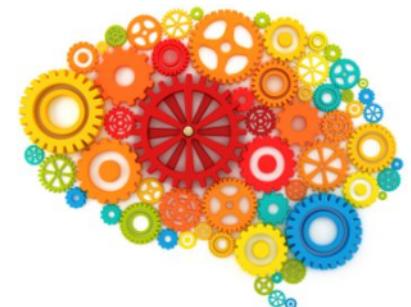
Internet-of-Things

➤ Things (Devices)

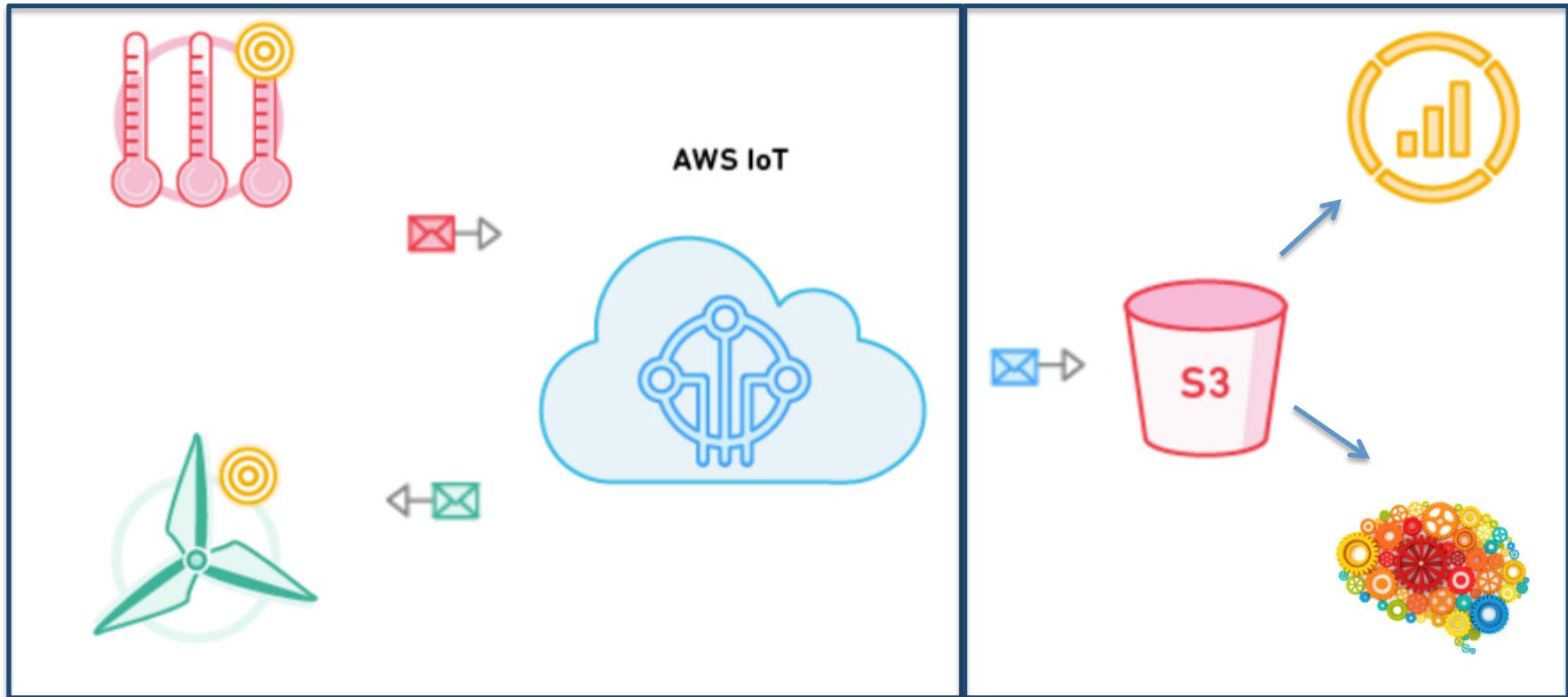
- Many of them
 - Different Types
 - Isolated Systems

- Data and Command
 - Sensing the world
 - Give Response

- Challenge
 - United: Connected + Communication
 - Smart: Data Analytics + Strategy



Solution: AWS IoT

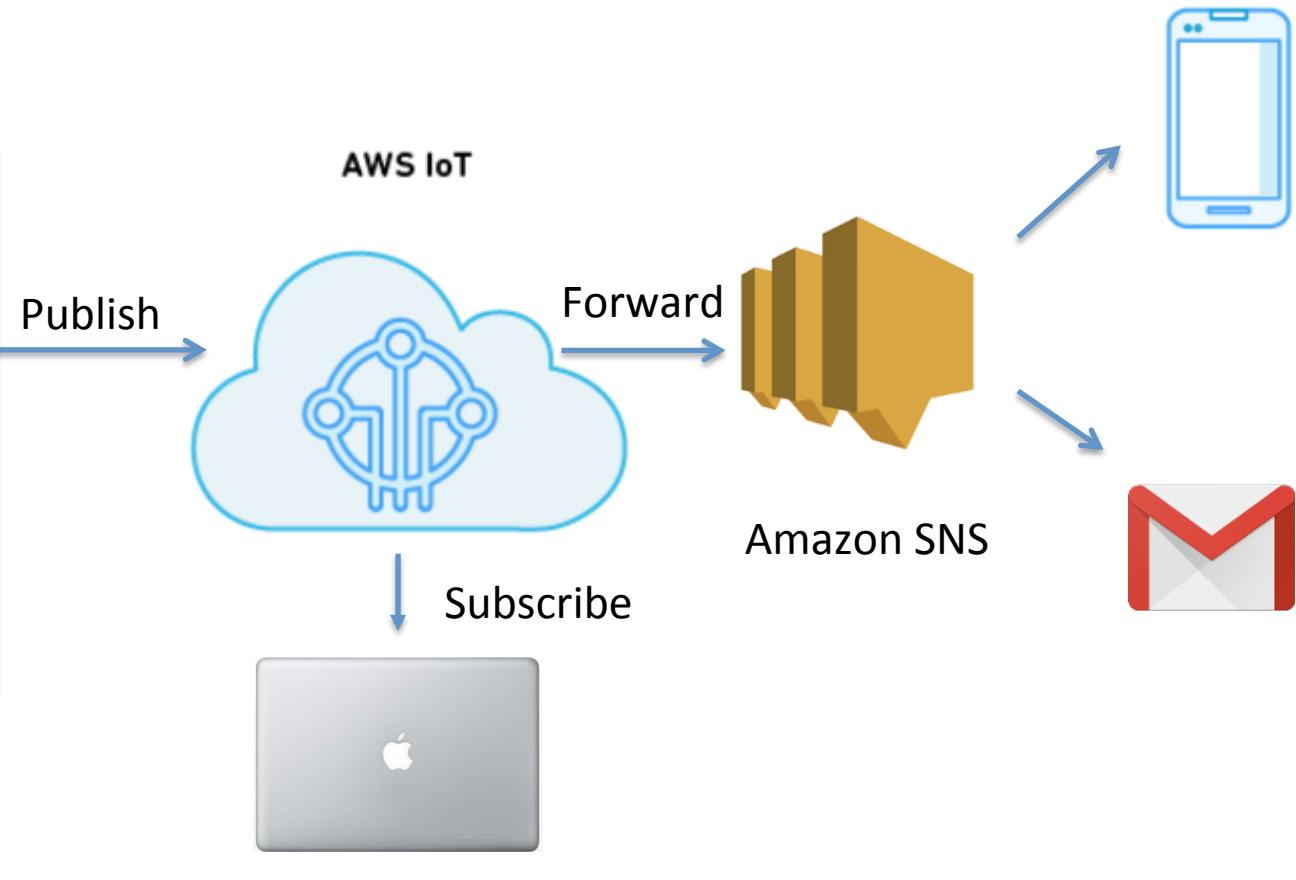
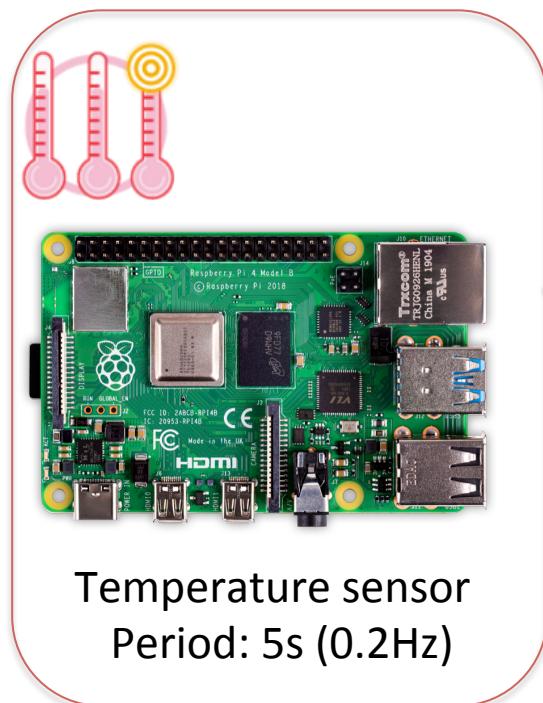


United: Connect + Communication

Stated: “Thing Shadow”

Smart: Other Cloud Service
Data Storage
Machine Learning

Tutorial: Hello AWS IoT!



Pointers

- 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/>

Group
A-Z

| | | | |
|---|--|--|---|
|  Compute <ul style="list-style-type: none"> EC2 Lightsail ↗ ECR ECS EKS Lambda Batch Elastic Beanstalk Serverless Application Repository |  Robotics <ul style="list-style-type: none"> AWS RoboMaker |  Analytics <ul style="list-style-type: none"> Athena EMR CloudSearch Elasticsearch Service Kinesis QuickSight ↗ Data Pipeline AWS Glue AWS Lake Formation MSK |  Business Applications <ul style="list-style-type: none"> Alexa for Business Amazon Chime ↗ WorkMail |
|  Blockchain <ul style="list-style-type: none"> Amazon Managed Blockchain | |  End User Computing <ul style="list-style-type: none"> WorkSpaces AppStream 2.0 WorkDocs WorkLink | |
|  Storage <ul style="list-style-type: none"> S3 EFS FSx |  Management & Governance <ul style="list-style-type: none"> AWS Organizations CloudWatch |  Security, Identity, & Compliance <ul style="list-style-type: none"> IAM Resource Access Manager |  Internet Of Things <ul style="list-style-type: none"> IoT Core Amazon FreeRTOS IoT 1 Click |

Step 1: Create a Virtual "Thing"

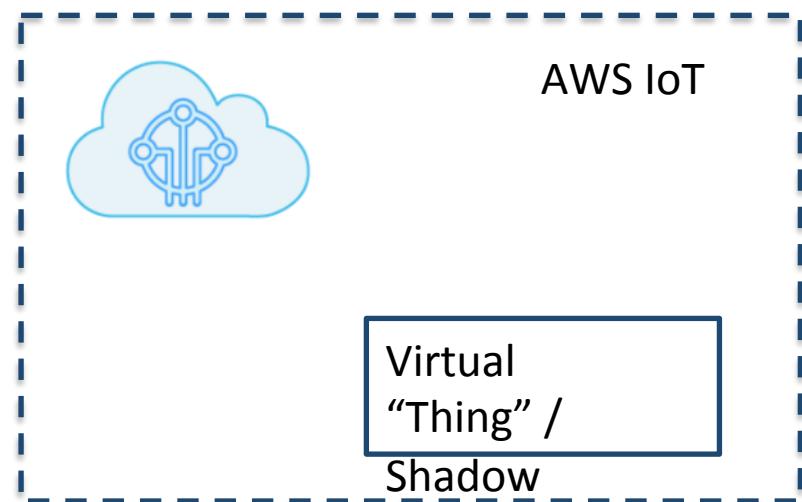
- A Thing in AWS IoT has a “**shadow**”
 - ❑ a JSON document that is used to **store and retrieve current state information for a device.**
 - **E.g.** Battery level, Connectivity, data

A “Dashboard” to show some info
 - ❑ **Shadow is a special topic in AWS IoT**

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

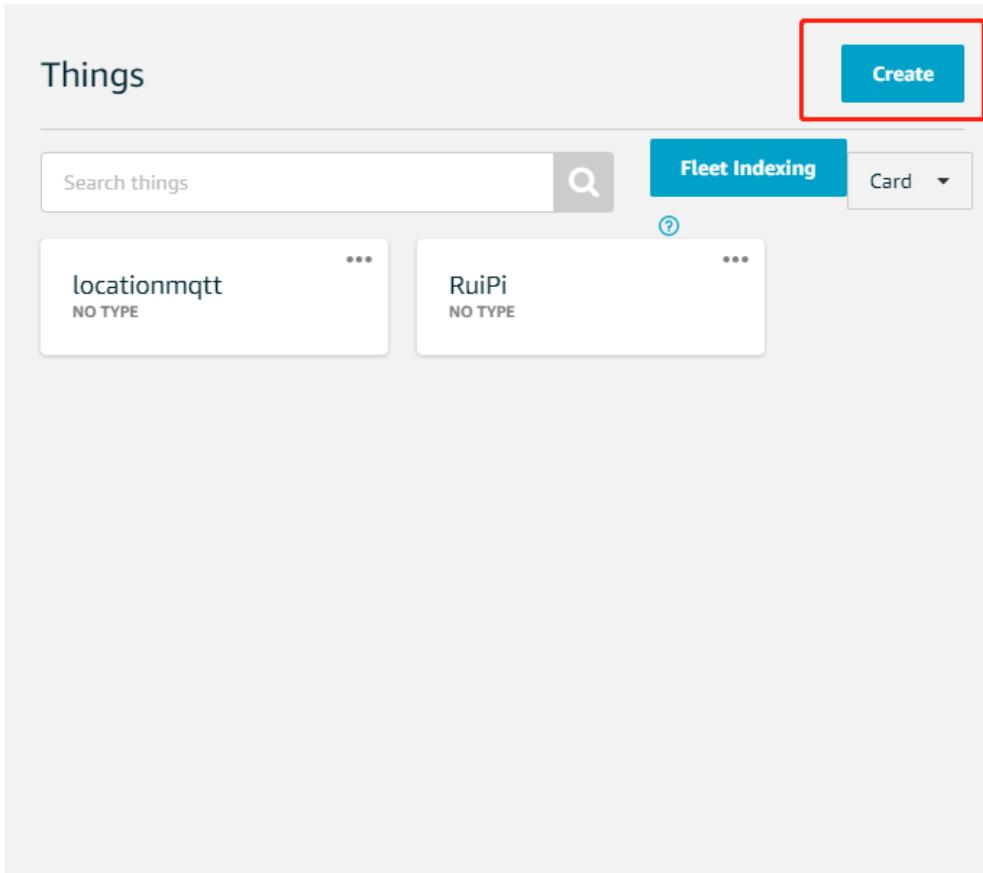
Certificates → your ID

Policy → your permission book



Create a thing

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



The screenshot shows the AWS IoT Things interface. On the left, a sidebar menu lists various AWS IoT services: Monitor, Onboard, Manage Things (which is highlighted with a red box), Types, Thing Groups, Billing Groups, Jobs, Tunnels, Greengrass, Secure, Defend, Act, and Test. The main content area is titled "Things" and contains a search bar, a "Fleet Indexing" button, and a "Card" dropdown. Below the search bar are two card-like boxes: one for "locationmqtt" (NO TYPE) and another for "RuiPi" (NO TYPE). A large blue "Create" button is located in the top right corner of the main content area, also highlighted with a red box.

1/16/20

17

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

Name

cse520

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.

[!\[\]\(eca9dca13a2688a2d5e75f35d4cc16aa_img.jpg\) 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

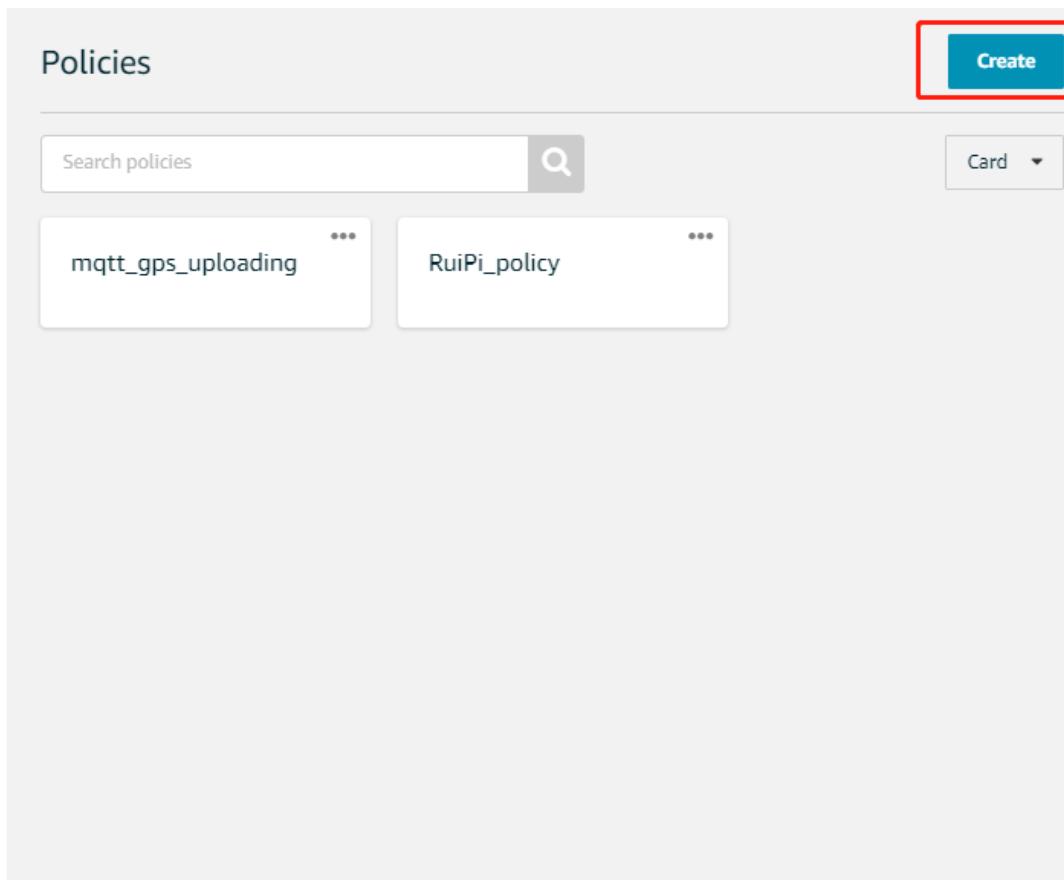
[Cancel](#)

[Done](#)

[Attach a policy](#)

Create Policy

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



The screenshot shows the AWS IoT Policies interface. On the left, there's a sidebar with navigation links: Monitor, Onboard, Manage, Greengrass, Secure (highlighted with a red box), Certificates, Policies (highlighted with a red box), CAs, Role Aliases, Authorizers, Defend, Act, and Test. The main area is titled "Policies" and contains a search bar labeled "Search policies" with a magnifying glass icon. Below the search bar are two policy cards: "mqtt_gps_uploading" and "RuiPi_policy", each with three dots at the top right. In the top right corner of the main area, there's a blue "Create" button with a white outline, which is also highlighted with a red box.



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

cse520_policy

Add statements

Policy statements define the types of actions that can be performed by a resource.

[Advanced mode](#)

Action

iot:*

Resource ARN

*

Effect

Allow Deny

[Remove](#)

[Add statement](#)

[Create](#)



Policies

Monitor

Onboard

Manage

Greengrass

Secure

Certificates

Policies

CAs

Role Aliases

Authorizers

Defend

Act

Test

Search policies



mqtt_gps_uploading

...

RuiPi_policy

...

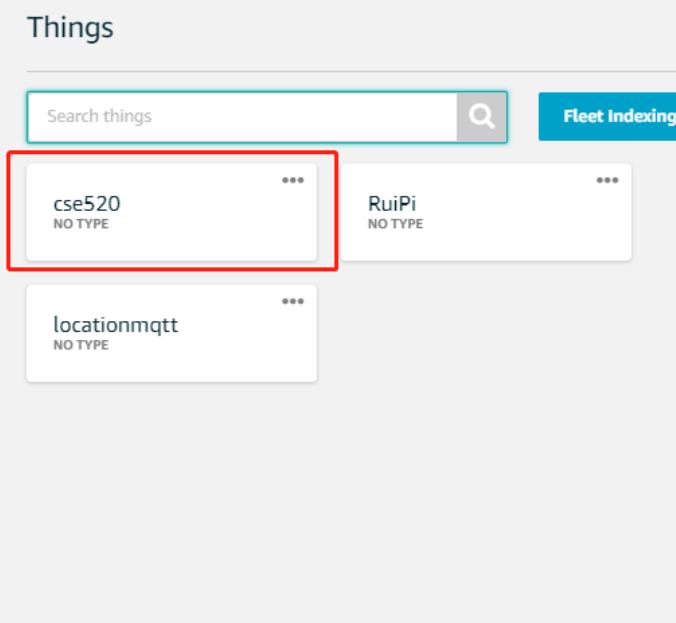
cse520_policy

...

This is the policy you created

Attach Policy

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



The screenshot shows the AWS IoT Things interface. On the left, there's a sidebar with various options like Monitor, Onboard, Manage (which is selected and highlighted with a red box), Types, Thing Groups, Billing Groups, Jobs, Tunnels, Greengrass, Secure, Defend, Act, and Test. The main area is titled "Things" and contains a search bar and a "Fleet Indexing" button. Three items are listed: "cse520 NO TYPE", "RuiPi NO TYPE", and "locationmqtt NO TYPE". The "cse520" item is highlighted with a red box. A large blue arrow points from the "cse520" item to its detailed view on the right.

THING
CSE_521_Temp
NO TYPE

Details Certificates

Security (highlighted with a red box)

Create certificate View other options

Thing Groups
Billing Groups
Shadow
Interact
Activity
Jobs
Violations
Defender metrics

208f60eb4fab1b02f5...
Click the cert (highlighted with a red box)

This is the key/cert you just created

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

Certificate ARN

A certificate Amazon Resource Name (ARN) uniquely identifies this certificate.

`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

Amazon AWS CloudWatch Metrics Services Resource Groups N. Virginia

Attach policies to certificate(s)

Policies will be attached to the following certificate(s):
fa474a4756fb5e5cefd1aefcd9411b90344d96a8e38e28c2b2f50e9cb18d3ed2

Choose one or more policies

Search policies

| Policy Name | Action |
|---|----------------------|
| <input type="checkbox"/> mqtt_gps_uploading | View |
| <input type="checkbox"/> RuiPi_policy | View |
| <input checked="" type="checkbox"/> cse520_policy | View |

1 policy selected

Cancel **Attach**

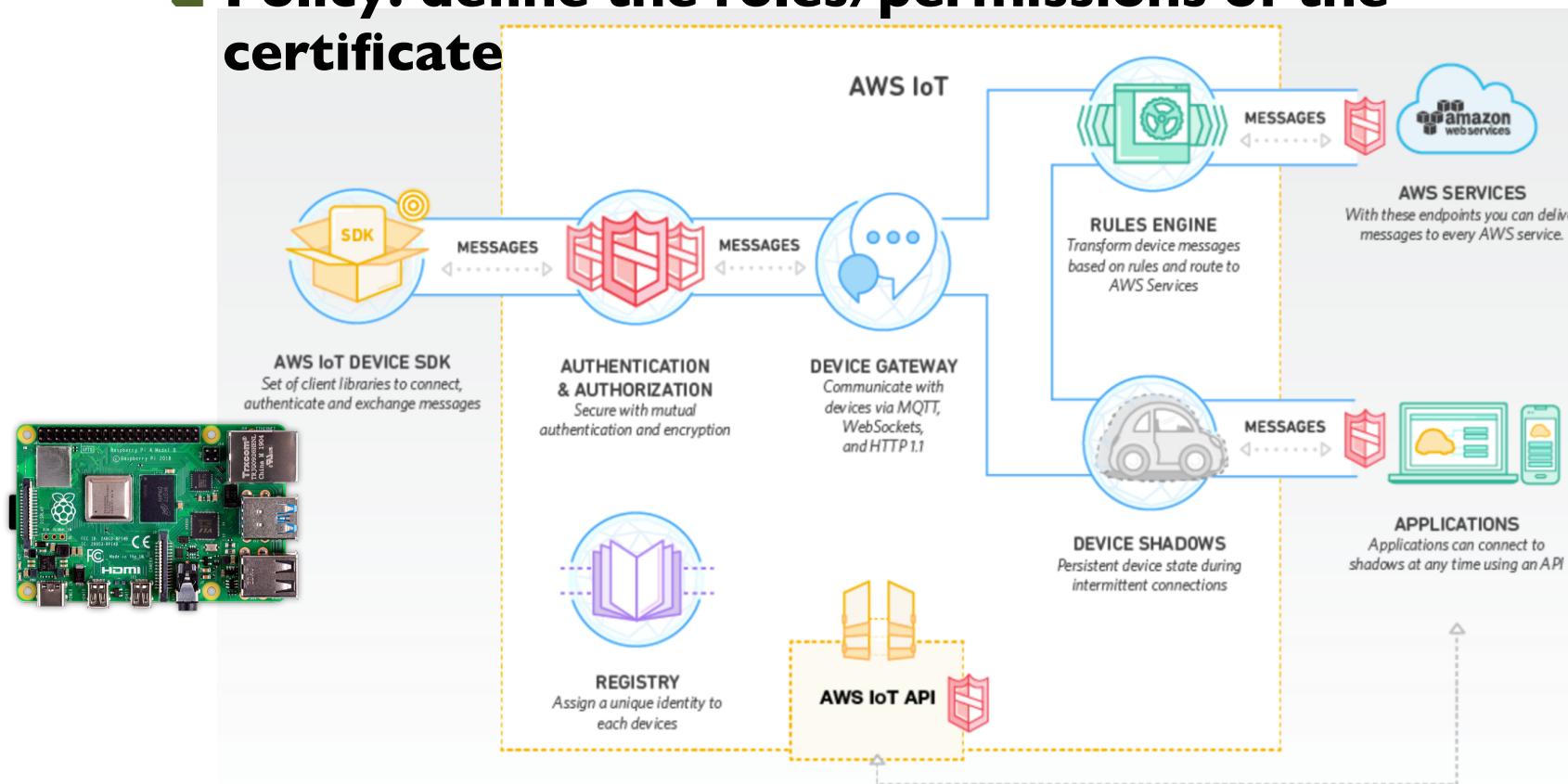
Details Policies Things Non-compliance

cse520

CERTIFICATE fa474a4756fb5e5cefd1aefcd9411b90344d96a8e38e28c2b2f50e9cb18d3ed2 ACTIVE

AWS Things Summary

- ❑ **Shadow:** Store/retrieve some information
- ❑ **Certificate:** authenticate the device
- ❑ **Policy:** define the roles/permissions of the certificate



Let's test it online!



Washington University in St. Louis

Basic Interact: Publish to the “Shadow”

- Get your “Shadow”
 - In your Thing Page

THING
cse520
 NO TYPE

- Details
- Security
- Thing Groups
- Billing Groups
- Shadow**
- Interact
- Activity
- Jobs
- Violations
- Defender metrics

Shadow ARN

A shadow ARN uniquely identifies the shadow for this thing. [Learn more](#)

arn:aws:iot:us-east-1:006025899016:thing/cse520

Shadow Document

Last update: Dec 31, 1969 6:00:00 PM -0600

Shadow state:
 {}

Metadata:

```
{
  "metadata": {},
  "timestamp": 1579110725,
  "version": 3
}
```

Find your “Shadow” Topic

- Topic: can be seen as the “address”

THING
cse520
NO TYPE

Details This thing already appears to be connected.

Security

Thing Groups

Billing Groups

Shadow

Interact

Activity

Jobs

Violations

Defender metrics

HTTPS

Update your Thing Shadow using this Rest API Endpoint. [Learn more](#)

a10qe38noifilm-ats.iot.us-east-1.amazonaws.com

MQTT

Use topics to enable applications and things to get, update, or delete the state information

[Learn more](#)

Shadow Topic

[Update to this thing shadow](#)

\$aws/things/cse520/shadow/update

Update to this thing shadow was accepted

\$aws/things/cse520/shadow/update/accepted



AWS IoT

Monitor

Onboard

Manage

Greengrass

Secure

Defend

Act

Test

Subscribe to a topic

Publish to a topic

• \$aws/things/cse520/shadow... ×

Subscribe

Devices publish MQTT messages on topics. You can use this page to subscribe to a topic and receive these messages.

Subscription topic

\$aws/things/cse520/shadow/update

Max message capture ?

100

Quality of Service ?

0 - This client will not acknowledge to the Device Gateway when messages are received

1 - This client will acknowledge to the Device Gateway when messages are received

MQTT payload display

Auto-format JSON payloads (improves readability)

Display payloads as strings (more accurate)

Display raw payloads (in hexadecimal)

Publish

Specify a topic and a message to publish with a QoS of

\$aws/things/cse520/shadow/update

```
1 [{  
2   "message": "Hello from AWS IoT console"  
3 }]
```



Shadow Message

```
{ "state": {  
    "reported": {  
        "time": "13:45",  
        "temperature": "25"  
    },  
    "message": "Hello from AWS IoT 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

THING
cse520
NO TYPE

- [Details](#)
- [Security](#)
- [Thing Groups](#)
- [Billing Groups](#)
- [Shadow](#)
- [Interact](#)
- [Activity](#)
- [Jobs](#)
- [Violations](#)
- [Defender metrics](#)

Shadow ARN

A shadow ARN uniquely identifies the shadow for this thing. [Learn more](#)

arn:aws:iot:us-east-1:006025899016:thing/cse520

Shadow Document

Last update: Jan 15, 2020 11:56:23 AM -0600

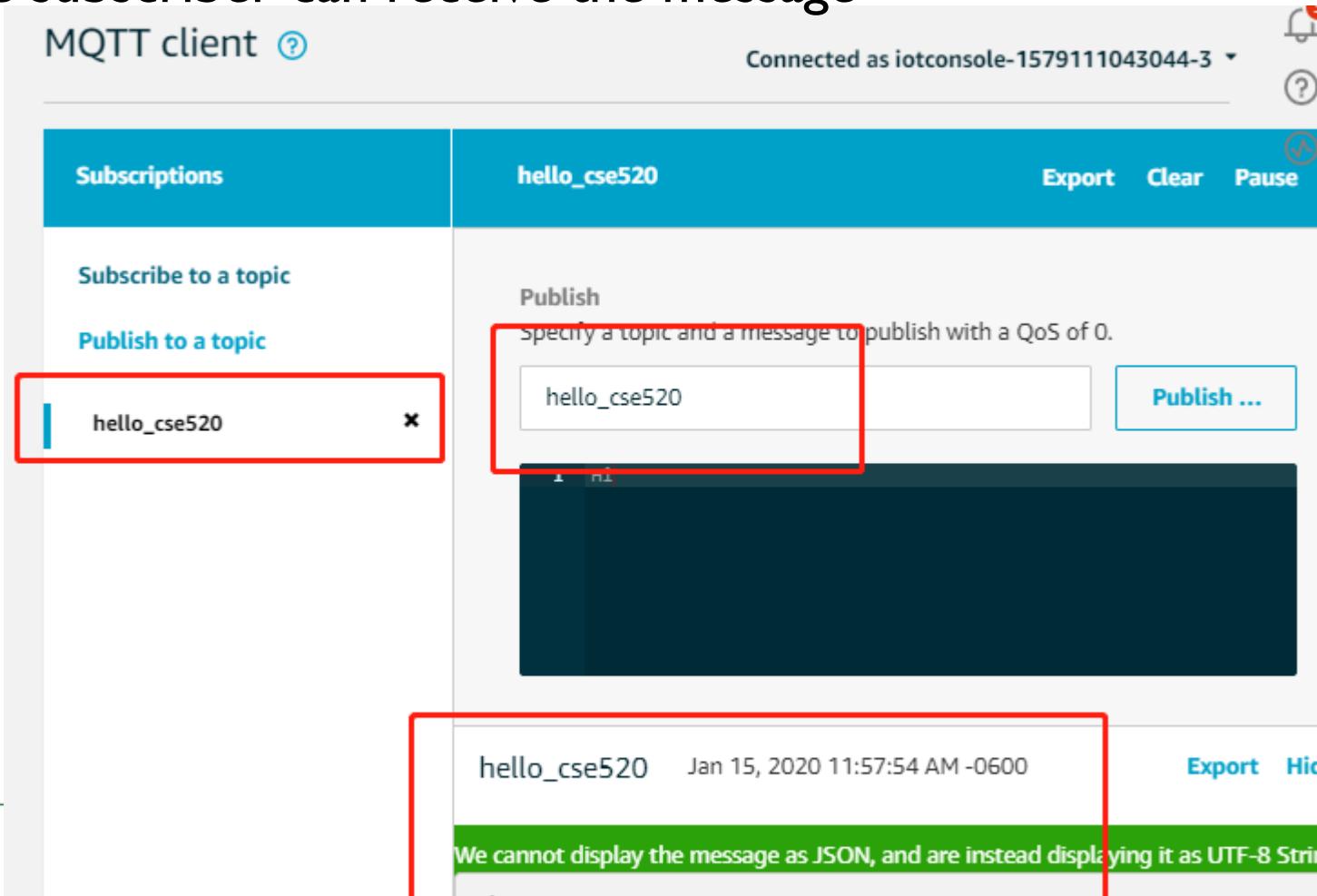
Shadow state:

```
{
  "reported": {
    "time": "13:45",
    "temperature": "25"
  }
}
```

Metadata:

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



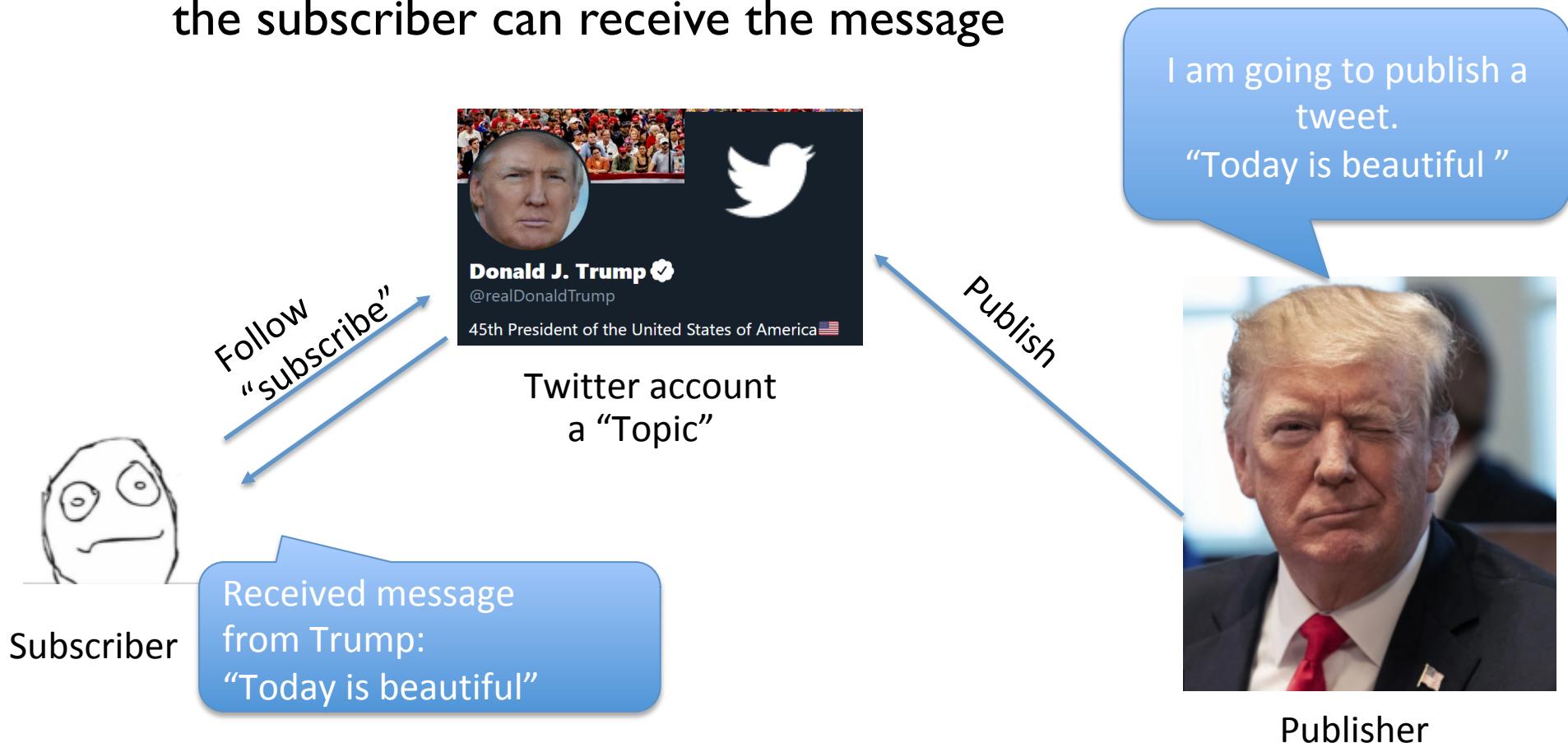
The screenshot shows a MQTT client interface with the following elements:

- MQTT client** (with a question mark icon) and **Connected as iotconsole-1579111043044-3**.
- Subscriptions** section: A list containing **hello_cse520**, which is highlighted with a red box.
- hello_cse520** section: A **Publish** area with a text input field containing **hello_cse520**, also highlighted with a red box. To the right are **Export**, **Clear**, and **Pause** buttons.
- Publish** instructions: **Specify a topic and a message to publish with a QoS of 0.**
- Publish ...** button.
- Message History**: A list showing a single message entry:

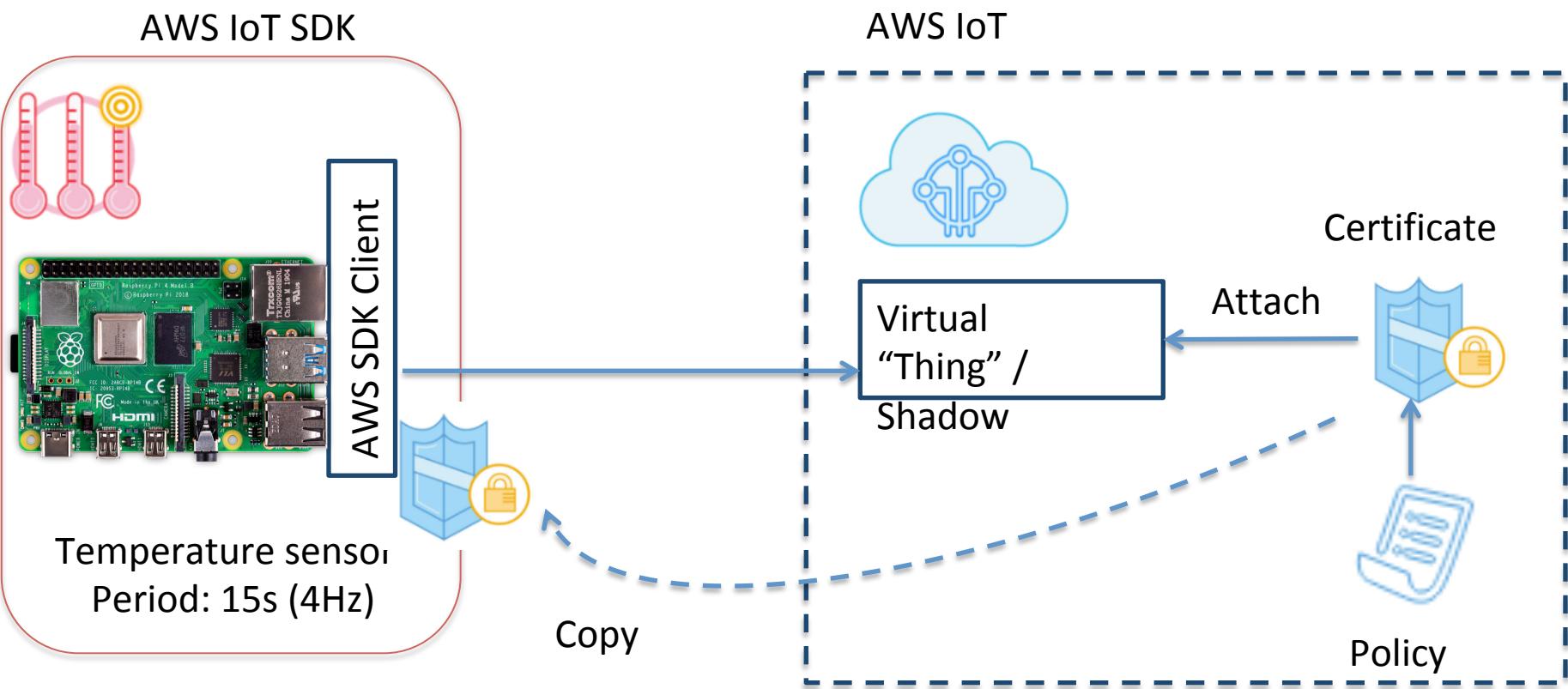
| | | | |
|---|---------------------------------------|---------------|-------------|
| hello_cse520 | Jan 15, 2020 11:57:54 AM -0600 | Export | Hide |
| We cannot display the message as JSON, and are instead displaying it as UTF-8 String | | | |

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**

- ❑ Note: through **scp** utility
- ❑ Downloaded before!

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](#)

➤ Choose your **AWS SDK** (support MQTT)

- ❑ Node JS
- ❑ Python (pip install AWSIoTPythonSDK)
- ❑ Java
- ❑ ...

➤ Set up your client with SDK and the certificates

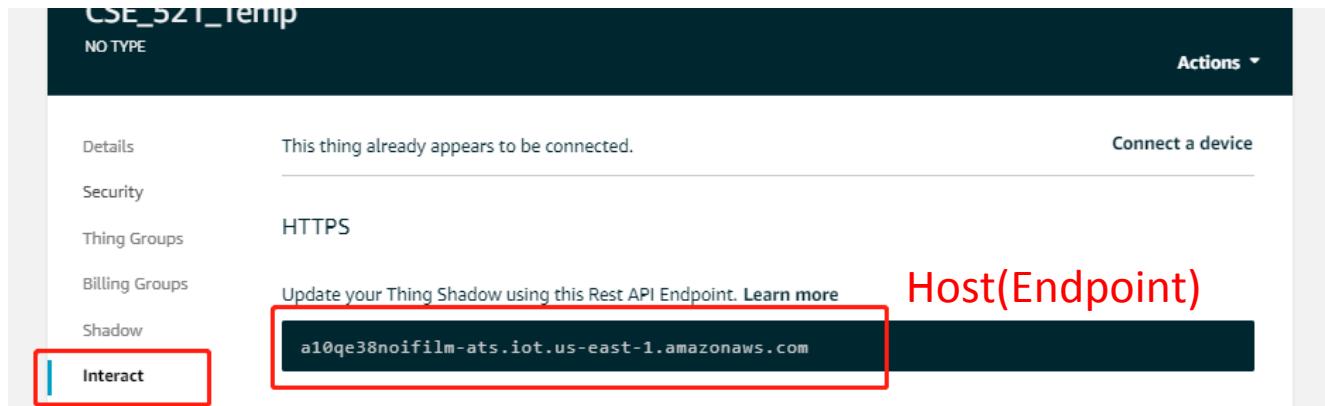


Some Notes

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

```
pi@RuiPi:~/cse520_demo $ ls -l
total 20
-rw-r--r-- 1 pi pi 3414 Jan 15 12:17 demo.pv
-rw-r--r-- 1 pi pi 1224 Jan 15 12:17 fa474a4756-certificate.pem.crt
-rw-r--r-- 1 pi pi 1675 Jan 15 12:17 fa474a4756-private.pem.key
-rw-r--r-- 1 pi pi 451 Jan 15 12:17 fa474a4756-public.pem.key
-rw-r--r-- 1 pi pi 1188 Jan 15 12:17 root-CA.crt
pi@RuiPi:~/cse520_demo $
```

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



The screenshot shows the AWS IoT Device Management console for a device named "CSE_520_temp". The "Interact" button is highlighted with a red box. The "Host(Endpoint)" field contains the value "a10qe38noifilm-ats.iot.us-east-1.amazonaws.com", which is also highlighted with a red box.

```
host = "a10qe38noifilm-ats.iot.us-east-1.amazonaws.com" # Your thing's endpoint. See tutorial slides
rootCAPath = "root-CA.crt"
certificatePath = "fa474a4756-certificate.pem.crt"
privateKeyPath = "fa474a4756-private.pem.key"
port = 8883
clientId = "CSE520"
topic = "$aws/things/cse520/shadow/update" # Shadow topic of your Thing
```

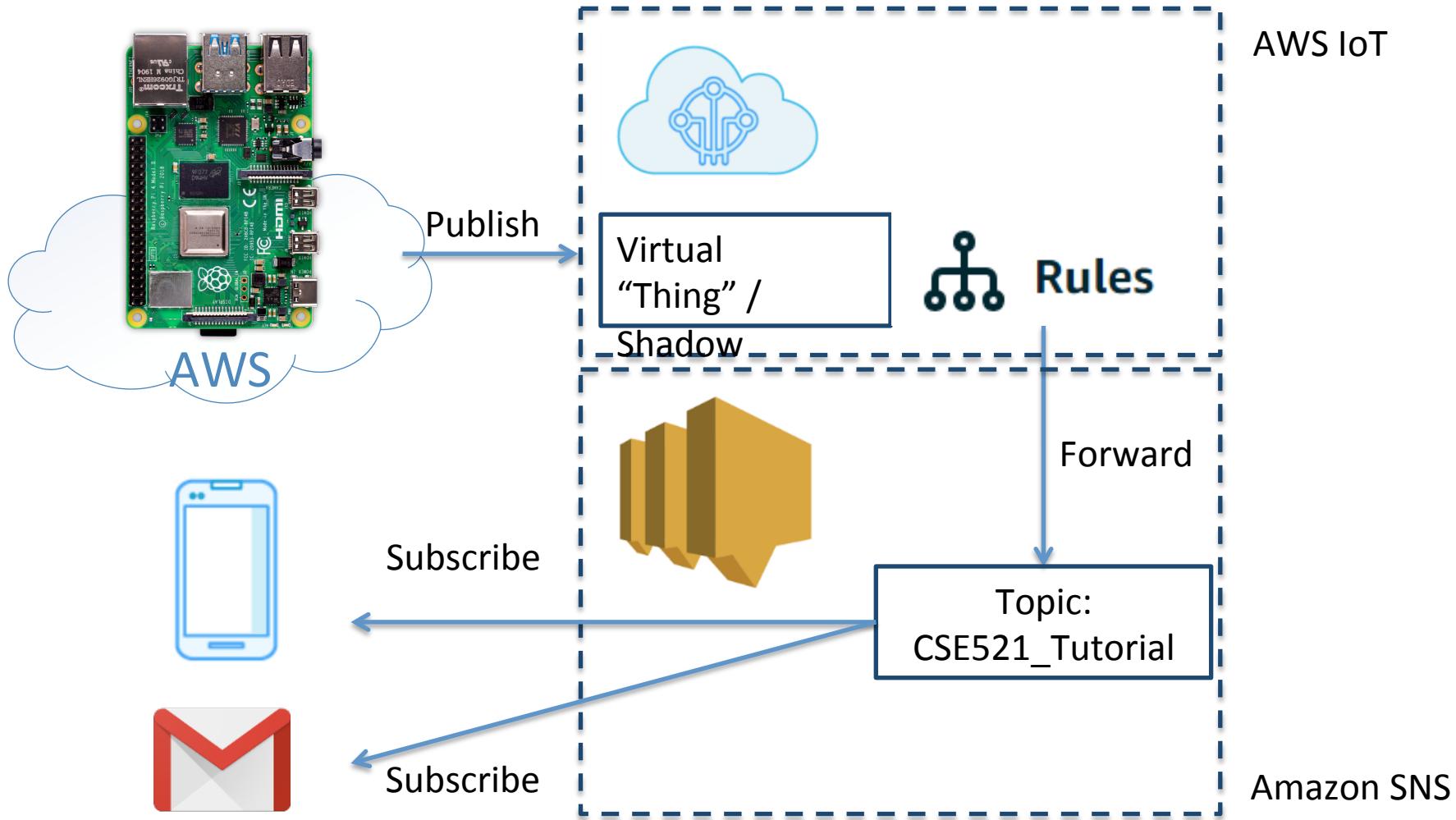
Modify the code

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)

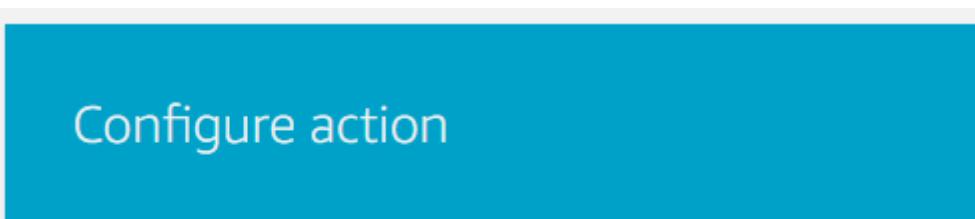
Rule query statement

SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT te
learn more, see [AWS IoT SQL Reference](#).

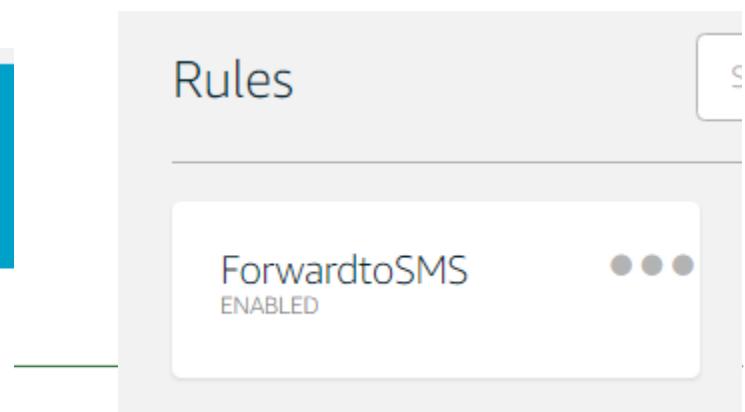
```
1 | SELECT * FROM 'iot/topic'
```

➤ Add an Action:

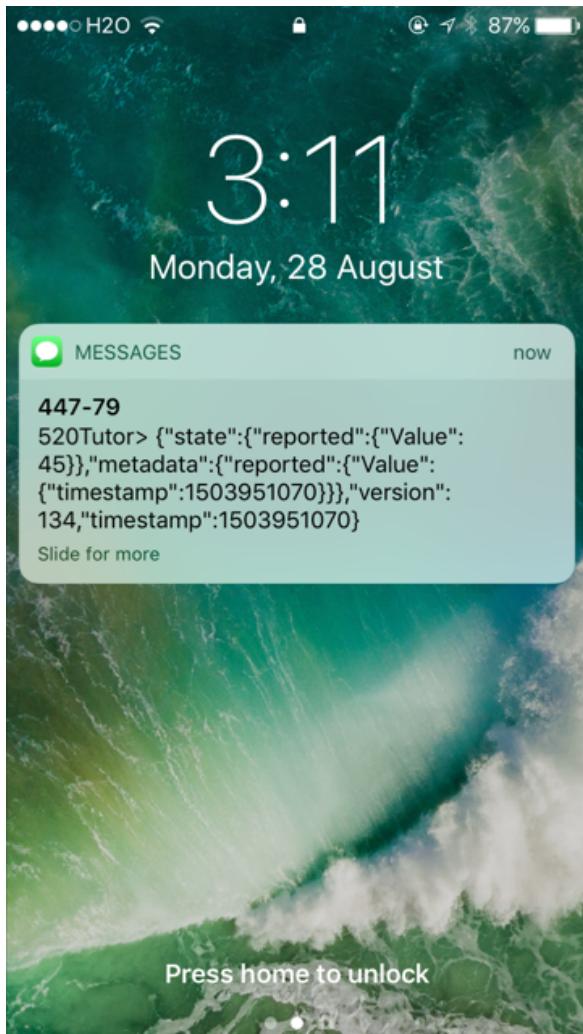
- Forward this message to SNS
- Specify Dest ARN
- Enable Rule



Send a message as an SNS push notification



Notification on SMS & Email



AWS Notification Message Inbox x

 520Tutor no-reply@sns.amazonaws.com 3:11 PM (28 minutes ago) Star Print Unsubscribe

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](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!



What if... \$50,000 AWS Bill!

Quora

Ask or Search Quora

Ask Question

Fraud

Amazon Web Services

Amazon.com (product)

Hackers

+3



My AWS account was hacked and I have a \$50,000 bill, how can I reduce the amount I need to pay?

For years, my bill was never above \$350/month on my single AWS instance. Then over the weekend someone got hold of my private key and launched hundreds of instances and racked up a \$50,000 bill before I found out about it on Tuesday. Amazon had sent a warning by email at \$15,000 saying they had found **our key posted publicly**, but I didn't see it. Naturally, this is a devastating amount of money to pay. I'm not saying I shouldn't pay anything, but this just a crazy amount in context. Amazon knew the account was compromised, that is why they sent an email, they knew the account history and I had only spent \$213 the previous month. I almost feel they deliberately let it ride to try to earn more money. Does anyone have any experience with this sort of problem?

Project 0: AWS IoT

- Create Your AWS account
- Setup your things with certificates and policy
- Publish and subscribe a topic
 - ❑ Update your thing's Shadow with AWS IoT Test Console
 - See tutorial
 - ❑ Use your computer as a “Thing” to update your thing’s Shadow
 - Install AWS IoT SDK, copy the certs, modify the code
 - Publish message to your Shadow Topic from your computer
 - ❑ Publish and Subscribe on your computer
 - Publish a message to a topic in one client from your PC
 - Subscribe to the **same topic** in another client from your PC
 - Hints: the “clientID” cannot be the same for the two client. The two clients can share the same certs. You can use the shadow as the topic.
 - (You should see one terminal sends massage, and another terminal receives that message)
- Email your results (inline with some screenshots, you don't need to write a report) to **dairuixuan@wustl.edu**

Some project examples

- 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

Thanks!

Ruixuan Dai



Washington University in St. Louis