Pointers

- **Amazon IoT**
  - [http://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html](http://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html)

- **Raspberry Pi**

- **Resource list** for course projects

- **Apply for $40 credits for Amazon AWS**
  - [https://aws.amazon.com/education/awssEducate/apply/](https://aws.amazon.com/education/awssEducate/apply/)
XaaS: Basics in 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 …
  - Amazon EC2

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

- **SaaS (Software as a Service):** email, virtual desktop, games …
  - Office 365
  - Stadia
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 you’re smart A/C controller

---

You Manage:
- APP
- Data
- Runtime
- Middleware
- OS
- Virtualization
- Server
- Storage
- Network

Service Provider Manages:
PaaS Example: Amazon IoT

Tools

Framework

AWS IoT

AWS IoT DEVICE SDK
Set of client libraries to connect, authenticate and exchange messages

AUTHENTICATION & AUTHORIZATION
Secure with mutual authentication and encryption

DEVICE GATEWAY
Communicate with devices via MQTT, WebSockets, and HTTP 1.1

RULES ENGINE
Transform device messages based on rules and route to AWS Services

DEVICE SHADOWS
Persistent device state during intermittent connections

REGISTRY
Assign a unique identity to each device

APPLICATIONS
Applications can connect to shadows at any time using an API

AWS SERVICES
With these endpoints you can deliver messages to every AWS service.
Software as a Service (SaaS)

- **SaaS**
  - You get a whole solution

- **Example**
  - Gmail
  - Dropbox
  - Office365
  - Stadia

Service Provider Manages:

- APP
- Data
- Runtime
- Middleware
- OS
- Virtualization
- Server
- Storage
- Network
XaaS: A Recap

On-Premise
- APP
- Data
- Runtime
- Middleware
- OS
- Virtualization
- Server
- Storage
- Network

IaaS
- APP
- Data
- Runtime
- Middleware
- OS
- Virtualization
- Server
- Storage
- Network

PaaS
- APP
- Data
- Runtime
- Middleware
- OS
- Virtualization
- Server
- Storage
- Network

SaaS
- APP
- Data
- Runtime
- Middleware
- OS
- Virtualization
- Server
- Storage
- Network

You Manage
Service Provider Manages

8/28/19
Tutorial: Hello! AWS IoT!!
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

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!

- **Temperature sensor**
  - Period: 5s (20Hz)

Diagram:
- **Publish** to **AWS IoT**
- **Forward** to **Amazon SNS**
- **Subscribe**
- **Message middleware**

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

- 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
Create a thing

1. AWS IoT Menu
   • Things ➔ Create

2. Give a name
Add your device to the thing registry

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

Name

CSE_521_Temp

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

Add another

Clear

Show thing shadow ▼
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 with CSR**
Upload your own certificate signing request (CSR) based on a private key you own.

**Use my certificate**
Register your CA certificate and use your own certificates for one or many devices.

**Skip certificate and create thing**
You will need to add a certificate to your thing later before your device can connect to AWS IoT.
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:

<table>
<thead>
<tr>
<th>Item Description</th>
<th>File Name</th>
<th>Download</th>
</tr>
</thead>
<tbody>
<tr>
<td>A certificate for this thing</td>
<td>208f60eb4f.cert.pem</td>
<td></td>
</tr>
<tr>
<td>A public key</td>
<td>208f60eb4f.public.key</td>
<td></td>
</tr>
<tr>
<td>A private key</td>
<td>208f60eb4f.private.key</td>
<td></td>
</tr>
</tbody>
</table>

You also need to download a root CA for AWS IoT:

- A root CA for AWS IoT: [Download]

Activate keys

- The keys and cert will be used later

Done

Attach a policy
A policy is attached to a key/cert

- It tells what this key/cert can do
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**

CSE_521_Temp_policy

**Add statements**

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

**Action**

`iot:*`

**Resource ARN**

`
`

**Effect**

- Allow
- Deny

[Add statement]

[Create]
This is the policy you created
Attach Policy

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

Click the cert

This is the key/cert you just created
CERTIFICATE

208f60eb4fab1b02f5d656963382b05cd5c690b481e710c6a3e899a...

INACTIVE

Details
Certificate ARN

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

arn:aws:iot:us-east-1:006025899016:cert/208f60eb4fab1b02f5d656963382b05cd5c690b481e710c6a3e899a...

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

Actions
Attach policy
Activate
Deactivate
Revoke
Accept transfer
Reject transfer
Revoke transfer
Start transfer
Attach thing
Download
Delete
Attach policies to certificate(s)

Policies will be attached to the following certificate(s):
208f60eb4fab1b02f5d656963382b05cd5c690b481e710c6a3e899a73b5f9080

Choose one or more policies

- [ ] mqtt_gps_uploading
- [x] CSE_521_Temp_policy

1 policy selected
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!
Basic Interact: Publish to the “Shadow”

➢ Get your “Shadow”
   - In your Thing Page

THING
CSE_521.Temp
NO TYPE

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

This thing already appears to be connected.

HTTPS

Update your Thing Shadow using this REST API Endpoint. Learn more

a10qe38no1fim-ats.iot.us-east-1.amazonaws.com

MQTT

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

Learn more

Update to this thing shadow

$aws/things/CSE_521.Temp/shadow/update

Update to this thing shadow was accepted

$aws/things/CSE_521.Temp/shadow/update/accepted
Using Embedded MQTT Client to Test
- In AWS IoT Page

Your Shadow Topic
Topic Message

8/28/19
A Shadow Message is a JSON object. **Shadow message has strict formats.**

Update Shadow

In your “Thing” Page

Thing ARN

A shadow ARN uniquely identifies the shadow for this thing. Learn more


Shadow Document

Last update: Aug 25, 2019 1:33:35 PM -0500

Shadow state:

```json
{
  "reported": {
    "time": "13:45",
    "temperature": "25"
  }
}
```
Basic Interact: Subscribe/Publish

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

Subscription topic
- CSE521

Max message capture
- 100

Quality of Service
- 0 - This client will not acknowledge to the Device Gateway that messages are received
- 1 - This client will acknowledge to the Device Gateway that 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 0.

- CSE521

Message:
- Hello
Step 2: Connect a “Physical” Device

Temperature sensor
Period: 15s (4Hz)

AWS IoT SDK

MQTT Client

AWS IoT

Virtual “Thing” / Shadow

Certificate

Attach

Policy

Copy

8/28/19
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: 208f50eb4f.cert.pem  
  Download
- A public key: 208f50eb4f.public.key  
  Download
- A private key: 208f50eb4f.private.key  
  Download

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

```
[ec2-user@$ip-172-31-92-213 temp_demo]$ ls
aws-iot-device-sdk-python
demo.py
CSE_521_temp.cert.pem
CSE_521_temp.private.key
CSE_521_temp.public.key
[ec2-user@$ip-172-31-92-213 temp_demo]$
```

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

```
host = "a10qe38noifilm-ats.iot.us-east-1.amazonaws.com"  # Your thing's endpoint. See tutorial slides
rootCAPath = "root-CA.crt"
certificatePath = "e556be14fc-certificate.pem.crt"
privateKeyPath = "e556be14fc-private.pem.key"
port = 8883
```
SDK and Demo Codes

- https://github.com/aws/aws-iot-device-sdk-python
More: Rule Engine, Link with SNS services

Simple Notification Service

- Publish from EC2 t2.micro to AWS IoT
- Rules: Virtual "Thing" / Shadow
- Forward to Topic: CSE521_Tutorial
- Subscribe to Amazon SNS
Create a Rule in Amazon IoT

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

**Rule query statement**

```sql
SELECT * FROM '$aws/things/RaspberryPi/shadow/update/accepted'
```

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

8/28/19
Notification on SMS & Email

3:11 PM (28 minutes ago)

520Tutor no-reply@sns.amazon\n
"state":{"reported":{"Value":45}},"metadata":{"reported":{"Value":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!

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?
Thanks!

Ruixuan Dai

Aug. 29, 2019