Cloud Tutorial: AWS IoT

CSE 520S
Haoran Li
Jan/26/2021
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
- Azure Cloud

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
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

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

You Manage

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

You Manage

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

You Manage

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

Service Provider Manages
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
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!
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

- 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.

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
Next
Back
Step 2: Certificate and its Policy

AWS IoT

Virtual “Thing” / Shadow

Certificate

Attach

Policy
Create a Thing

Add a certificate for your thing

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.
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

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

- A root CA for AWS IoT: [Download]

**Download all keys and root CA**

**Activate keys**

**The keys and cert will be used later**

Done
Create 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

cse520_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
Activate
Deactivate
Revoke
Accept transfer
Reject transfer
Revoke transfer
Start transfer
Attach policy
Attach thing
Download
Delete
Attach policies to certificate(s)

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

Choose one or more policies

- mqtt_gps_uploading
- RuiPi_policy
- cse520_policy

1 policy selected

[Attach]
AWS Things Summary

- **Shadow**: Store/retrieve some information
- **Certificate**: authenticate the device
- **Policy**: define the roles/permissions of the certificate
Before Connecting our Device:

Test with the web MQTT client
Step 2: Certificate and its Policy

AWS IoT

Certificate

Attach

Virtual “Thing” / Shadow

Policy
Basic Interact: Publish to the “Shadow”

- Get your “Shadow”
  - In your Thing Page

Default (Classic) Shadow Topic:
$aws/things/YOUR_THING_NAME/shadow/update$
Find your MQTT Shadow Topic

➢ Be sure to “learn more”

Details
Security
Thing groups
Billing Groups

Shadows
Interact
Activity
Jobs
Violations
Defender metrics

Shadow ARN

A shadow ARN uniquely identifies the shadow for this thing.

arn:aws:iot:us-west-2:401317363811:thing/cse520

For more info on using shadow

Learn more

Shadow Document

Last update: January 26, 2021, 12:01:36 (UTC-0600)

Shadow state:

```js
{
  "desired": {
    "welcome": "aws-iot"
  },
  "reported": {
    "welcome": "aws-iot"
  }
}
```
Using Embedded **Test Client** to Test

- In AWS IoT Page

Subscribe to a topic

- Publish to a topic

$aws/things/cse520/shadow...

Subscription topic

$aws/things/cse520/shadow/update

Max message capture

100

Quality of Service

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

Your Shadow Topic

Topic Message
A Shadow Message is a JSON object.

**Shadow message has strict formats.**

Please see

Update Shadow

In your “Thing” Page

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:

```
{
  "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
Connect the Device:

Update shadow states from your device
Step 3: Connection

AWS IoT SDK

AWS IoT

Certificate

Virtual “Thing” / Shadow

Copy

Attach

Policy

Report “Value”
Connect your Device

- Copy certificates to your **physical things**
  - Note: through `scp` utility
  - Downloaded before!

- Choose your **AWS SDK** (support MQTT)
  - Node JS
  - Python
    - `python3 -m pip install awsiotsdk`
    - *(python2 was deprecated)*
  - Java
  - ...

- Set up your client with SDK and the certificates

---

1/28/21
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.py
   -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)

   ```
   host = "a10qe38moifilm-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
   ```
SDK and Demo Codes

- AWS IoT SDK Documentation Page:
  - [https://docs.aws.amazon.com/iot/latest/developerguide/iot-sdk.html](https://docs.aws.amazon.com/iot/latest/developerguide/iot-sdk.html)

- Python 3 SDK:
  - [https://github.com/aws/aws-iot-device-sdk-python-v2](https://github.com/aws/aws-iot-device-sdk-python-v2)
More: Rule Engine, Link with SNS services

Simple Notification Service

- Publish
- Subscribe

AWS

Virtual “Thing” / Shadow

Rules

Forward

Topic: CSE520_Tutorial

AWS IoT

Amazon SNS
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 to learn more, see AWS IoT SQL Reference.

- Add an Action:
  - Forward this message to SNS
  - Specify Dest ARN
  - Enable Rule
Notification on SMS & Email

AWS Notification Message

520Tutor> {"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:00c543527d1a-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

1/28/21
One More Thing: Account Security

➢ DON’T UPLOAD YOUR KEY PUBLICLY!!!
What if... $50,000 AWS Bill!

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?

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
Exercise

- 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 clients. The two clients can share the same certs. You can use the shadow as the topic.
    - (You should see one terminal sends message, and another terminal receives that message)
Thanks!