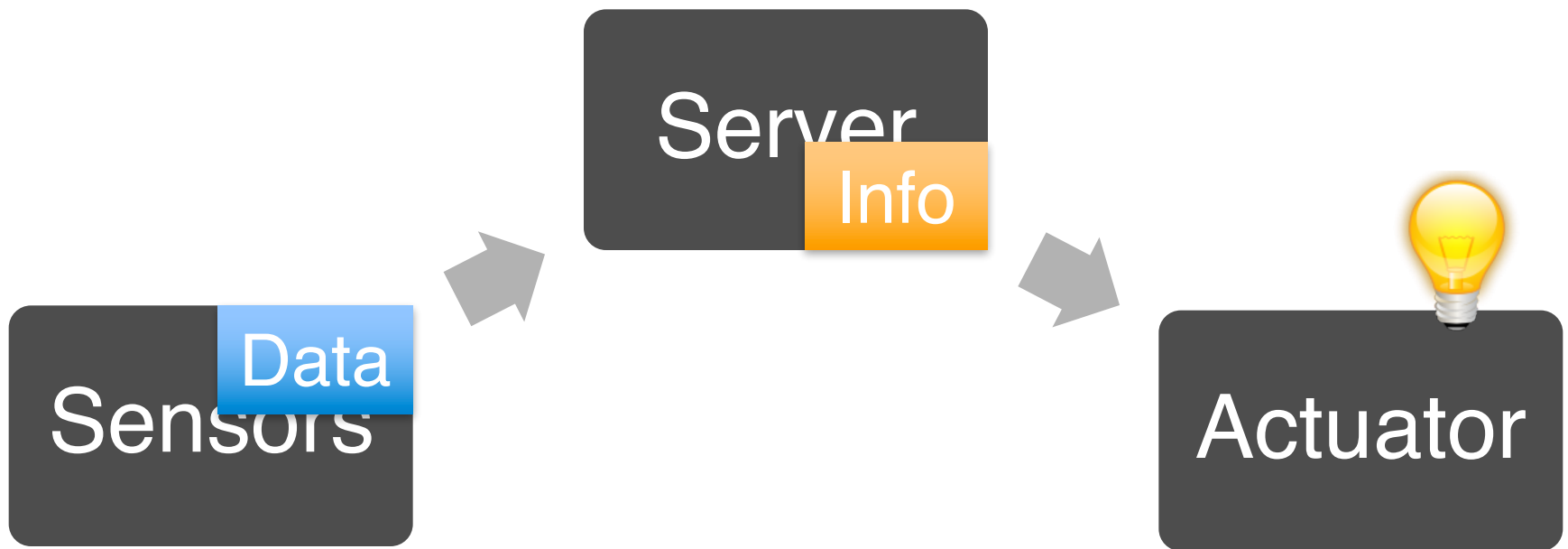




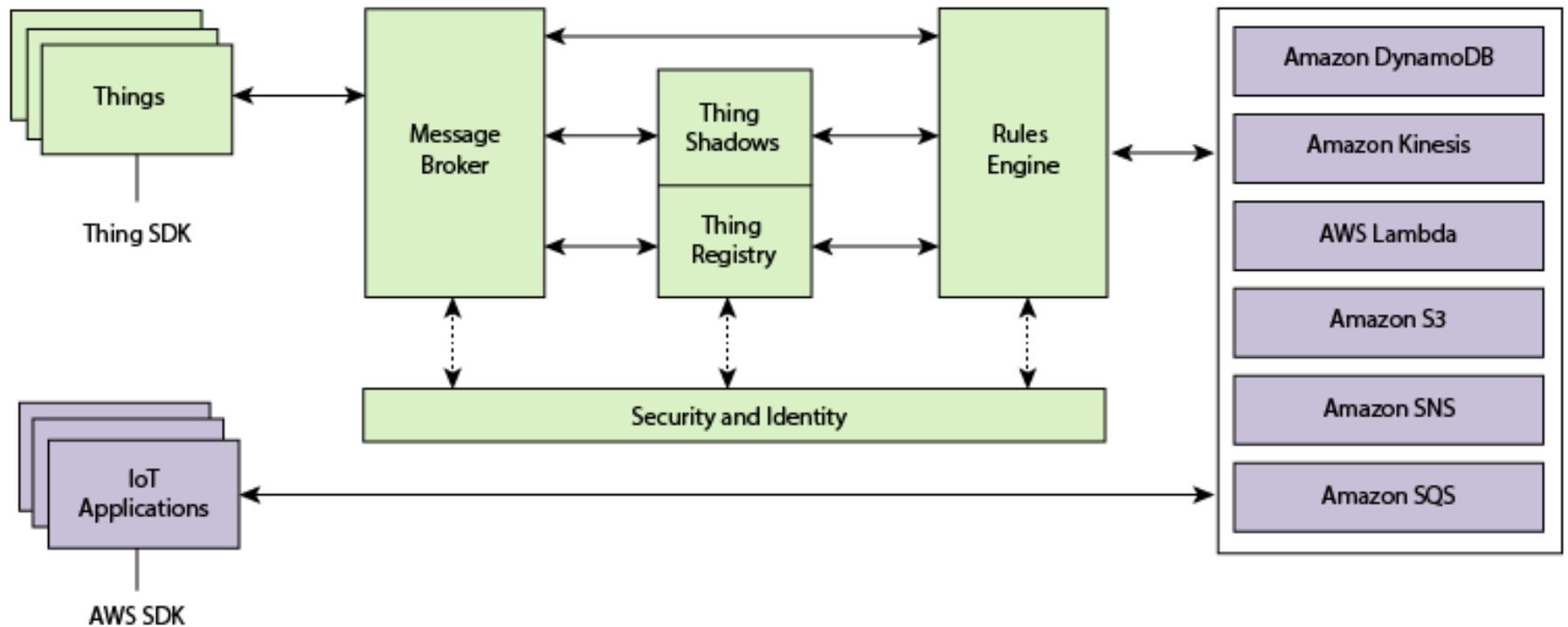
AWS IOT Services

► 周君哲

Centralized IoT Application







AWS IoT





Usage of AWS IoT

Amazon Web Services




Compute

-  **EC2**
Virtual Servers in the Cloud
-  **EC2 Container Service**
Run and Manage Docker Containers
-  **Elastic Beanstalk**
Run and Manage Web Apps
-  **Lambda**
Run Code in Response to Events







Storage & Content Delivery

-  **S3**
Scalable Storage in the Cloud
-  **CloudFront**
Global Content Delivery Network
-  **Elastic File System**
Fully Managed File System for EC2
-  **Glacier**
Archive Storage in the Cloud
-  **Snowball**


Developer Tools

-  **CodeCommit**
Store Code in Private Git Repositories
-  **CodeDeploy**
Automate Code Deployments
-  **CodePipeline**
Release Software using Continuous Delivery


Management Tools

-  **CloudWatch**
Monitor Resources and Applications
-  **CloudFormation**
Create and Manage Resources with Templates
-  **CloudTrail**
Track User Activity and API Usage
-  **Config**
Track Resource Inventory and Changes
-  **OpsWorks**
Automate Operations with Chef
-  **Service Catalog**






Internet of Things

-  **AWS IoT**
Connect Devices to the Cloud

Game Development

-  **GameLift**
Deploy and Scale Session-based Multiplayer Games

Mobile Services

-  **Mobile Hub**
Build, Test, and Monitor Mobile Apps
-  **Cognito**
User Identity and App Data Synchronization
-  **Device Farm**
Test Android, iOS, and Web Apps on Real Devices in the Cloud
-  **Mobile Analytics**
Collect, View and Export App Analytics
-  **SNS**
Push Notification Service

Usage of AWS IoT



AWS ▾

Services ▾

Edit ▾



Oregon ▾

Support ▾



AWS IoT

[Resources](#) | [MQTT Client](#) | [Tutorial](#) | [Settings](#) | [7 notifications](#)

Resources

[+ Create a resource](#)

[↗ Connect AWS IoT Button](#)



Filter by resource names or by resource type (below)

All 0/2 things **0/0 thing types** 0/0 rules 0/0 CAs
0/2 certificates 0/2 policies

Select all

Actions ▾

First Previous **1** Next Last

Usage of AWS IoT

AWS Services Edit

AWS IoT

Resources | MQTT Client | Tutorial

Resources

Close create panel

Connect AWS IoT Button

Create a thing

Create a thing type

Create a rule

Use my certificate

Create a certificate

Create a policy

Create a thing

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

Name

WKworkshop

Choose a thing type

You can associate a thing type to your thing. If you do not want to associate your thing with a type, choose **No type**

No type

Attributes used in a thing search

Next (optional), you can use thing attributes to describe the identity and capabilities of your device. Each attribute is a key-value pair.

Add attribute

Usage of AWS IoT

The screenshot displays the AWS IoT console interface. At the top, there's a navigation bar with 'AWS', 'Services', and 'Edit' dropdowns, along with user and region information. Below this, the 'AWS IoT' header is visible. The main content area is titled 'Resources' and includes a '+ Create a resource' button. A search bar shows 'WK' with a dropdown menu. Below the search bar, statistics are listed: 'All 1/3 things 0/0 thing types 0/0 rules 0/0 CAs 0/2 certificates 0/2 policies'. A card for 'WKworkshop' is shown, featuring a robot icon and a square icon. To the right, a 'Detail' panel for 'WKworkshop' is displayed, showing fields like 'Name', 'REST API endpoint', 'Thing type', 'MQTT topic', 'Last update', 'Attributes used in a thing search', and 'Linked certificates'. At the bottom of this panel, there are two buttons: 'Create a rule' and 'Connect a device', with the latter highlighted by a red rectangle. The footer contains 'Feedback', 'English', and copyright information.

Resources [+ Create a resource](#)

WK

All 1/3 things 0/0 thing types 0/0 rules 0/0 CAs 0/2 certificates 0/2 policies

WKworkshop

Detail

Name WKworkshop

REST API endpoint <https://a1trumz0n7awwt.iot.us-west-2.amazonaws.com/things/WKworkshop/shadow>

Thing type No type

MQTT topic Saws/things/WKworkshop/shadow/update

Last update No state [Refresh](#)



Attributes used in a thing search None


Linked certificates None

[Create a rule](#) [Connect a device](#)

[Feedback](#) [English](#) Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Usage of AWS IoT

 AWS IoT



Connect a device

Connect your device to one of our many supported SDKs.

☐ **Embedded C** ☒ **NodeJS**

☐ **Arduino Yún**

Please download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys will not be retrievable after closing this form.

- [Download public key](#)
- [Download private key](#)
- [Download certificate](#)

Confirm & start connecting

AWS IoT Python SDK

- ▶ `pip install AWSIoTPythonSDK`

AWSIoTMQTTClient

This is the client class used for plain MQTT communication with AWS IoT. You can initialize and configure the client like this:

```
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient

myMQTTClient = AWSIoTMQTTClient("myClientID")
myMQTTClient.configureEndpoint("YOUR.ENDPOINT", 8883)
myMQTTClient.configureCredentials("YOUR/ROOT/CA/PATH", "PRIVATE/KEY/PATH", "CERTIFICATE/PATH")
myMQTTClient.configureOfflinePublishQueueing(-1) # Infinite offline Publish queueing
myMQTTClient.configureDrainingFrequency(2) # Draining: 2 Hz
myMQTTClient.configureConnectDisconnectTimeout(10) # 10 sec
myMQTTClient.configureMQTTOperationTimeout(5) # 5 sec
```

AWS IoT Python SDK

For basic MQTT operations, your script will look like this:

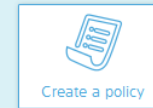
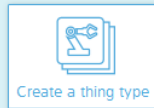
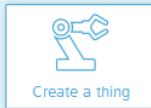
```
...  
myMQTTClient.connect()  
myMQTTClient.publish("myTopic", "myPayload", 0)  
myMQTTClient.subscribe("myTopic", 1, customCallback)  
myMQTTClient.unsubscribe("myTopic")  
myMQTTClient.disconnect()
```

AWS IoT Rules Engine

Resources

✕ Close create panel

🔗 Connect AWS IoT Button



Create a rule


Create a rule to evaluate inbound messages published into AWS IoT. Your rule can deliver a message to the topic of another device, or to a cloud endpoint such as a DynamoDB table.

Name your rule and add an optional description.

Name WKfireDetect

Description If too hot, alarm


AWS IoT Rules Engine

 AWS IoT

Indicate the source of the messages you want to process with this rule.

Rule query statement	<input type="text" value="SELECT * FROM 'Room1' WHERE temp > 200"/>
SQL version	<div>2016-03-23-beta</div>
Attribute	<div>*</div> <div></div>
Topic filter	<div>Room1</div> <div></div>
Condition	<div>temp > 200</div> <div></div>

AWS IoT Rules Engine

 AWS IoT Res

Condition ⓘ

Select one or more actions to happen when the above rule query is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. (* required)

Choose an action

AWS IoT Republish ▼

This will republish the message to another AWS IoT topic.

***Topic** ⓘ

Choose or create a new role to grant AWS IoT the access to the selected AWS IoT resource to perform this action.

***Role name**

WKfire ▼

 ⓘ [Create a new role](#)

[Cancel](#) [Add action](#)

[Create](#)

AWS IoT MQTT Client

The screenshot displays the AWS IoT MQTT Client console interface. At the top, a blue navigation bar contains the AWS IoT logo and a menu with 'Resources', 'MQTT Client' (highlighted with a red box), 'Tutorial', 'Settings', and '0 notifications'. Below the navigation bar, the main content area is titled 'Resources' and includes a '+ Create a resource' button and a 'Connect AWS IoT Button' link. A filter bar allows filtering by resource names or types, showing counts for 'All', '0/3 things', '0/0 thing types', '1/1 rules', and '0/0 CAs'. Below the filter bar, a resource card for 'WKfireDetect' is shown, indicating it is 'ENABLED'. The card includes a share icon and a checkbox. On the right side of the resource list, there are 'Select all' and 'Actions' buttons. At the bottom right, pagination controls show 'First', 'Previous', '1' (selected), 'Next', and 'Last'.

AWS IoT

Resources | **MQTT Client** | Tutorial | Settings | 0 notifications

Resources

+ Create a resource

Connect AWS IoT Button

Filter by resource names or by resource type (below)

All 0/3 things 0/0 thing types **1/1 rules** 0/0 CAs
0/4 certificates 0/3 policies


WKfireDetect

ENABLED

Select all Actions

First Previous **1** Next Last

AWS IoT MQTT Client

 AWS IoT

Resources | MQTT Client | Tutorial | Settings | 0 notifications

You're not connected to the Device Gateway

- Go to the "Device Gateway connection" tab
- Type in your desired Client ID or generate one using the "Generate client ID" button
- Once the connection to the Device Gateway succeeds, Subscribe or Publish to topics using the "Subscribe to topic" and "Publish to topic" tabs
- If the connection to the Device Gateway fails, ensure that your Client ID is less than 128 bytes and encoded in UTF-8

MQTT Client Actions

Device Gateway connection

Subscribe to topic

Publish to topic

Publish log

Connection: Not connected


Client ID

189a1

Connect

Generate client ID

AWS IoT MQTT Client


 AWS IoT


Resources | MQTT Client | Tutorial | Settings | 0 notifications


You don't have any subscriptions.


- Go to the "Subscribe to topic" tab
- Type in your desired Topic Filter and QoS, then click "Subscribe"
- Once subscribed, you can unsubscribe from the topic by clicking the 'x' in the right corner of the tab that will be generated

MQTT Client Actions

Device Gateway connection 

Subscribe to topic 

Publish to topic 

Publish log 

Subscription topic

Max message capture

Quality of service (QoS) ☒ 0 ☐ 1

Subscribe

AWS IoT MQTT Client

 **AWS IoT**

Resources | MQTT Client | Tutorial | Settings | 0 notifications

Clear all messages

fire ✕



Clear messages

You don't have any messages on this subscription.

You might need to wait a bit for your messages to come in, or feel free to publish messages here.

- Go to the "Publish to topic" tab
- Choose your Topic, Quality of service, and Payload to send, then click "Publish"
- If you do not see your message after publishing, check to make sure the topic you are publishing to would be matched by the topic filter.

MQTT Client Actions

Device Gateway connection 

Subscribe to topic 

Publish to topic 

Publish log 

Payload

Room: 1

Quality of Service: 1

```
{ "temp": 30, "time": "11:00" }
```

Publish

AWS IoT MQTT Client

The screenshot displays the AWS IoT MQTT Client interface. At the top, the header includes the AWS IoT logo and navigation links: Resources, MQTT Client, Tutorial, Settings, and 0 notifications. A 'Clear all messages' button is located in the top left. The main area shows a message being received on the 'fire' topic, with a timestamp of 'Aug 23, 2016 4:42:01 PM'. The message content is a JSON object: `{ "temp": 300, "time": "11:30" }`. The MQTT Client Actions panel on the right includes a 'Device Gateway connection' status (green checkmark), a 'Subscribe to topic' button, and a 'Publish to topic' button. A 'Clear messages' button is also present in the message list area.

Clear all messages

fire x

Clear messages

Aug 23, 2016 4:42:01 PM

fire

```
1 {
2   "temp": 300,
3   "time": "11:30"
4 }
```

MQTT Client Actions

Device Gateway connection ✓

Subscribe to topic

Publish to topic