Hands-on with AWS IoT

Julien Simon
Principal Technical Evangelist
Amazon Web Services

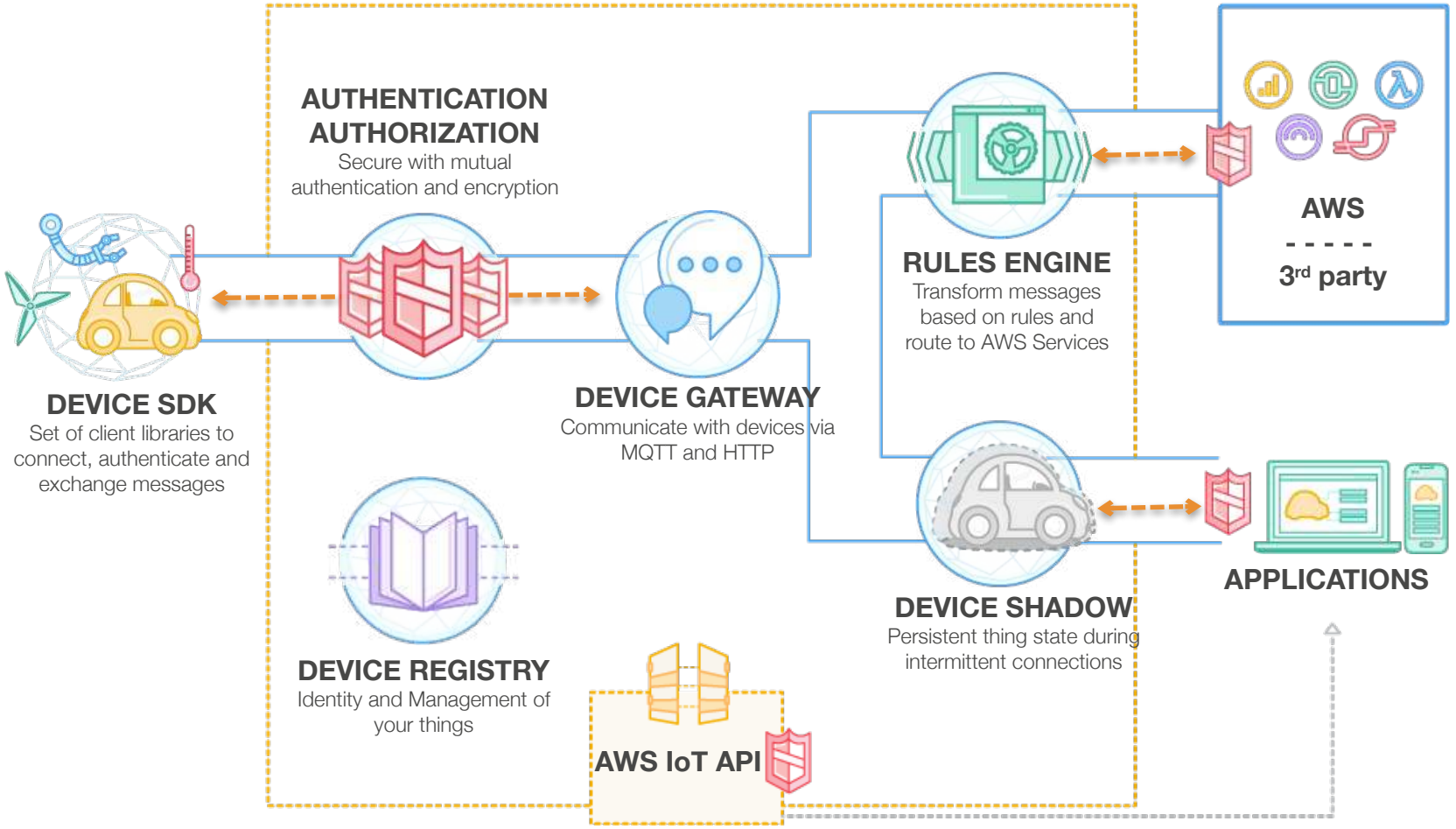
julsimon@amazon.com
@julsimon

07/11/2016

amazon web services

Agenda

- Overview of AWS IoT
- Devices & SDKs, with a focus on the Arduino Yún
- The MQTT protocol
- Managing “things”
- Routing AWS IoT messages to other AWS services
- Debugging AWS IoT applications
- And lots of AWS CLI, yeah!



Availability & Pricing

Region	Price
US East (N. Virginia)	$5 per million messages
US West (Oregon)	$5 per million messages
EU (Ireland)	$5 per million messages
EU (Frankfurt)	$5 per million messages
Asia Pacific (Sydney)	$6 per million messages
Asia Pacific (Seoul)	$6 per million messages
Asia Pacific (Tokyo)	$8 per million messages
Asia Pacific (Singapore)	$8 per million messages

- No minimum fee
- You are only charged on the number of incoming and outgoing messages
- 1 message = 512 bytes maximum
- Free tier: 250K messages / month for 12 months
- No charge when delivering to Amazon S3, Amazon DynamoDB, AWS Lambda, Amazon Kinesis, Amazon SNS, and Amazon SQS.

Devices & SDKs

amazon web services

Official AWS IoT Starter Kits

AVNET BROADCOM

MARVELL

RENESAS

TEXAS INSTRUMENTS

MICROCHIP

intel

MEDIATEK

ARROW

QUALCOMM

seeed
Grow the Difference

seeed
Grow the Difference

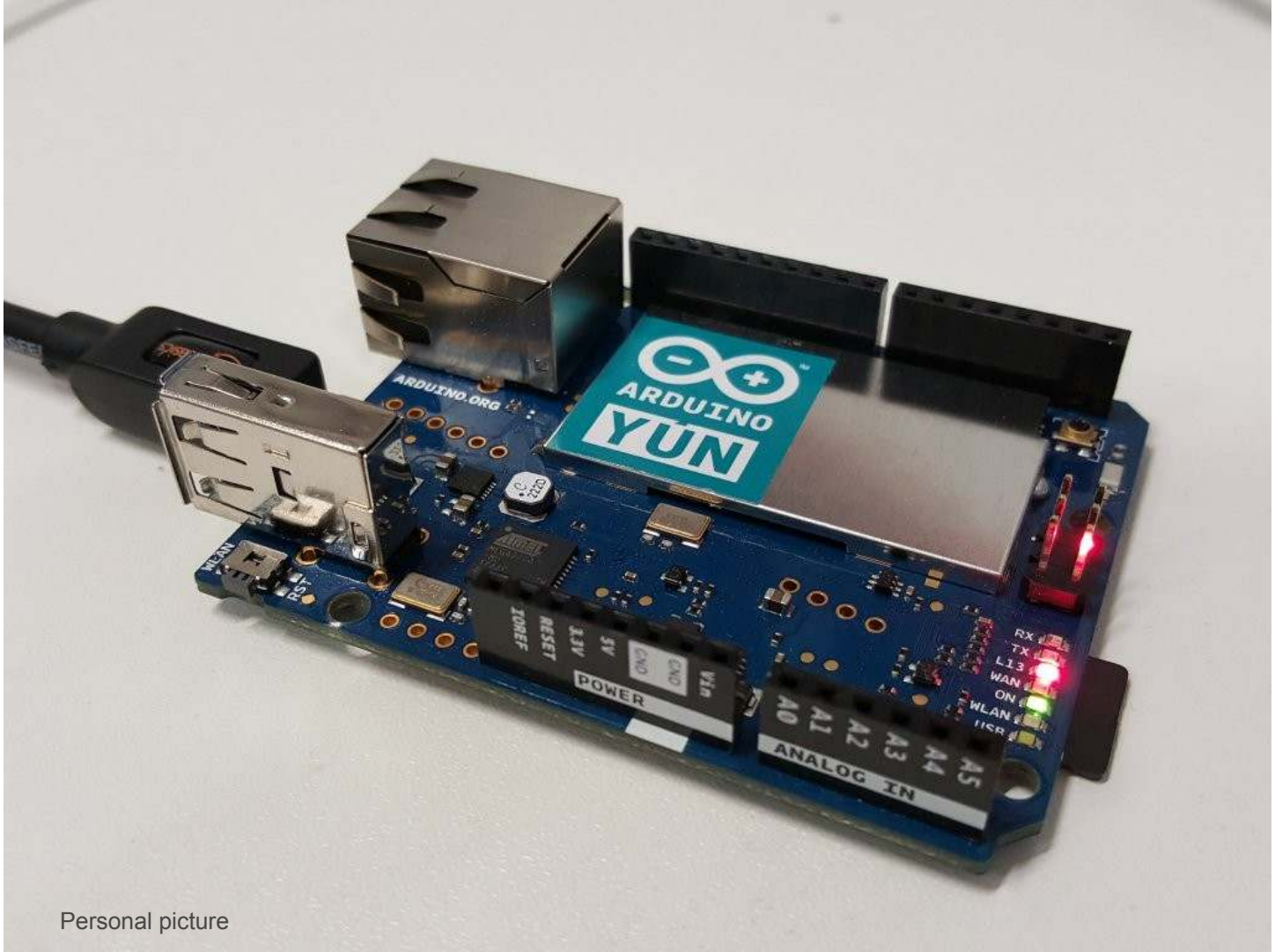
amazon
web services

Software platforms supported by AWS IoT

- Arduino Yún: https://github.com/aws/aws-iot-device-sdk-arduino-yun
- Javascript: https://github.com/aws/aws-iot-device-sdk-js
- Embedded C: https://github.com/aws/aws-iot-device-sdk-embedded-C

- Android: https://github.com/aws/aws-sdk-android/
- iOS: https://github.com/awslabs/aws-sdk-ios-samples

- Java (07/16): https://github.com/aws/aws-iot-device-sdk-java
- Python (07/16): https://github.com/aws/aws-iot-device-sdk-python



Personal picture

Hardware Shopping List

Arduino Yun ATmega32u4 Microcontroller Board A000008

by Arduino Org

$65.66 ~~$74.95~~ Prime

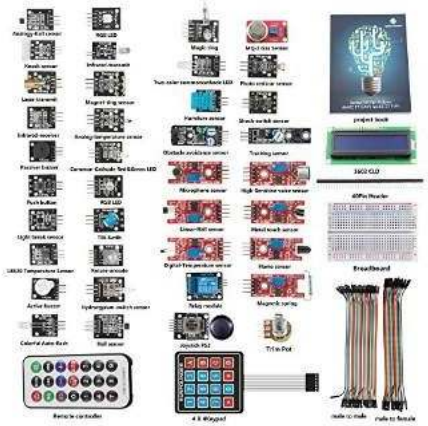
Get it by **Monday, Mar 21**

More Buying Choices
$65.00 new (17 offers)
$59.99 used (1 offer)

4.4 ▾ 68

FREE Shipping on eligible orders

Electronics: See all 153 items



SunFounder 37 modules Arduino Sensor Kit for Arduino UNO R3 Mega2560 Mega328 Nano (without controller)

by SunFounder

$68.99 Prime

Get it by **Monday, Mar 21**

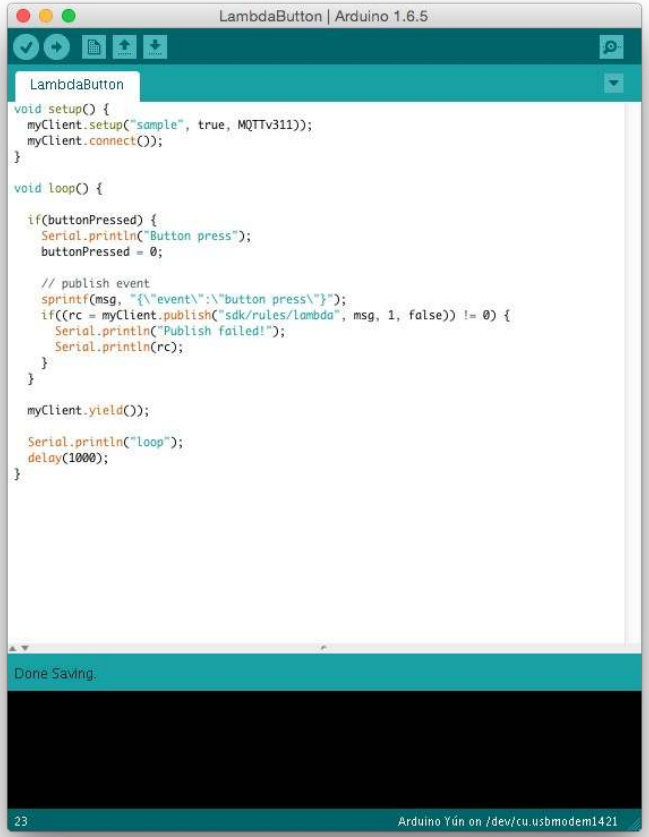
More Buying Choices
$68.99 new (64 offers)

4.5 ▾ 92

FREE Shipping on eligible orders

Electronics: See all 76 items

Software Shopping List

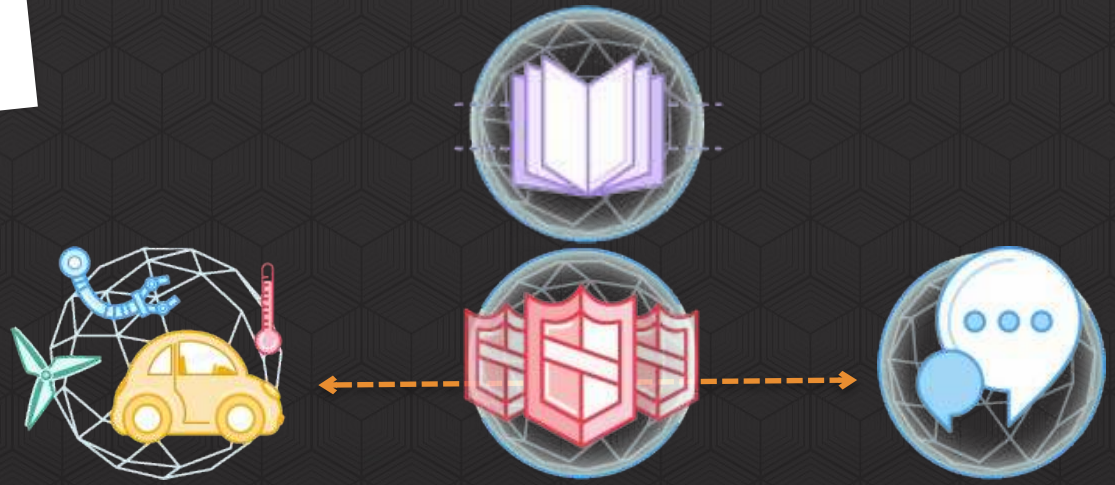


Arduino IDE and librairies
http://arduino.org/software

Arduino Web Editor & Cloud Platform
https://aws.amazon.com/blogs/aws/arduino-web-editor-and-cloud-platform-powered-by-aws/

Tip: ArduinoJson, a JSON library for embedded systems
https://github.com/bblanchon/ArduinoJson

amazon web services



Things

Managing things

- Thing Registry
- Secure Identity for Things
- Secure Communications with Things
- Fine-grained Authorization for:
  - Thing Management
  - Access to messages
  - Access to AWS services

AWS IoT is supported by AWS CloudFormation (07/16)

Creating a thing

```
% aws iot create-thing --thing-name myThing

% aws iot describe-thing --thing-name myThing

% aws iot list-things
```

You can use thing types and attributes to organize and tag your things (07/16)
http://docs.aws.amazon.com/iot/latest/developerguide/thing-types.html

Creating a certificate and keys

```
% aws iot create-keys-and-certificate
--set-as-active
--certificate-pem-outfile cert.pem
--public-key-outfile publicKey.pem
--private-key-outfile privateKey.pem
```

The AWS IoT root certificate, the thing certificate and the thing private key must be installed on your device, e.g. https://github.com/aws/aws-iot-device-sdk-arduino-yun

You can also use your own certificates (04/16), ECC cryptography (05/16),
as well as just-in-time registration (08/16)
https://aws.amazon.com/blogs/mobile/use-your-own-certificate-with-aws-iot/
https://aws.amazon.com/blogs/iot/elliptic-curve-cryptography-and-forward-secrecy-support-in-aws-iot-3/
https://aws.amazon.com/blogs/aws/new-just-in-time-certificate-registration-for-aws-iot/

Creating a policy

```
% cat myPolicy.json
{
   "Version": "2012-10-17",
   "Statement": [{ "Effect": "Allow", "Action":
["iot:*"],
   "Resource": ["*"] }]
}
```

```
% aws iot create-policy
--policy-name PubSubToAnyTopic
--policy-document file://myPolicy.json
```

Assigning an identity to a Policy and a Thing

```
% aws iot attach-principal-policy
--policy-name PubSubToAnyTopic
--principal CERTIFICATE_ARN

% aws iot attach-thing-principal
--thing-name myThing
--principal CERTIFICATE_ARN
```

Arduino : connecting to AWS IoT

```
aws_iot_mqtt_client myClient;

if((rc = myClient.setup(AWS_IOT_CLIENT_ID)) == 0) {
  // Load user configuration
  if((rc = myClient.config(AWS_IOT_MQTT_HOST,
AWS_IOT_MQTT_PORT, AWS_IOT_ROOT_CA_PATH,
   AWS_IOT_PRIVATE_KEY_PATH, AWS_IOT_CERTIFICATE_PATH)) == 0) {
      if((rc = myClient.connect()) == 0) {
         // We are connected
         doSomethingUseful();
      }
   }
}
```

The MQTT protocol

amazon web services

Protocols supported by AWS IoT

- MQTT over HTTPS to publish and subscribe (IPv4 and IPv6)
- MQTT over WebSocket to publish and subscribe. Security is managed with AWS Signatures v4.
- HTTPS protocol to publish.

MQTT Protocol

- OASIS standard protocol (v3.1.1)
- Lightweight transport protocol that is useful for connected devices
- Publish-subscribe with topics

- MQTT is used on oil rigs, connected trucks, and many more critical applications
- Until now, customers had to build, maintain and scale a broker to use MQTT with cloud applications

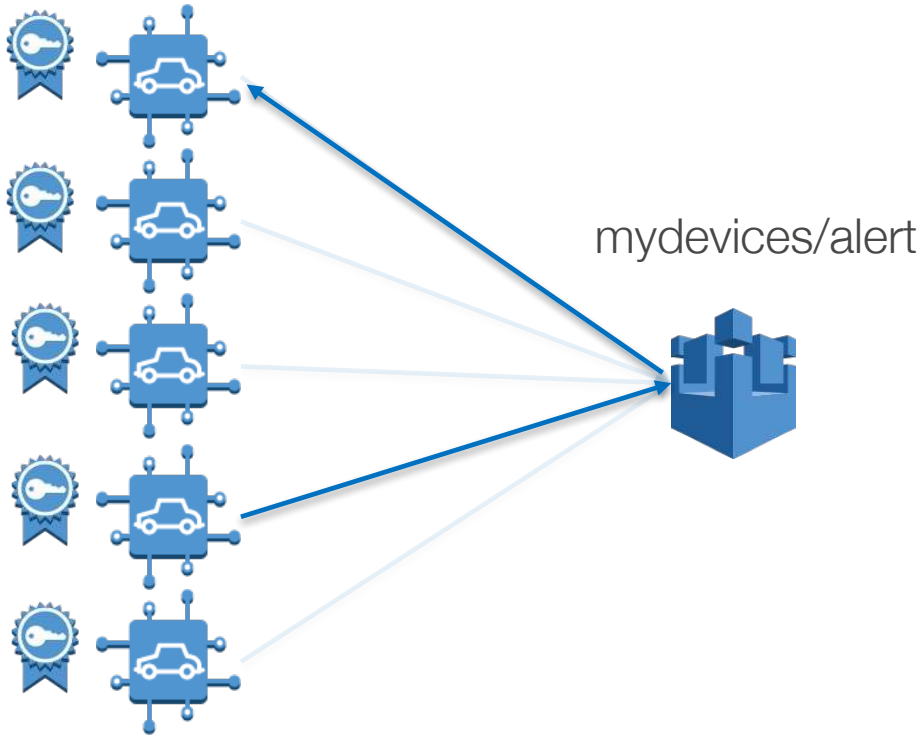
MQTTS vs HTTPS:

93x faster throughput
11.89x less battery to send
170.9x less battery to receive
50% less power to stay connected
8x less network overhead

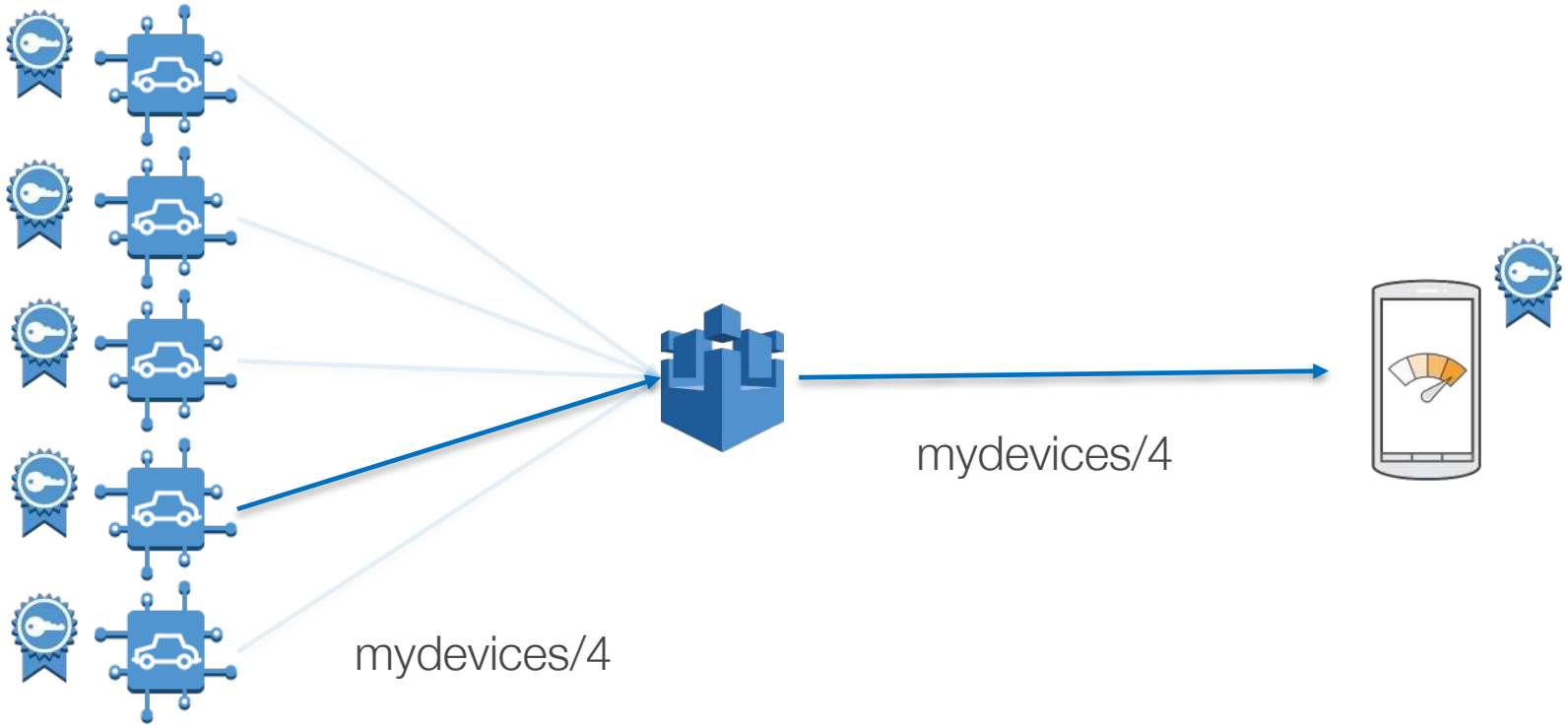
Source: http://stephendnicholas.com/archives/1217

amazon web services

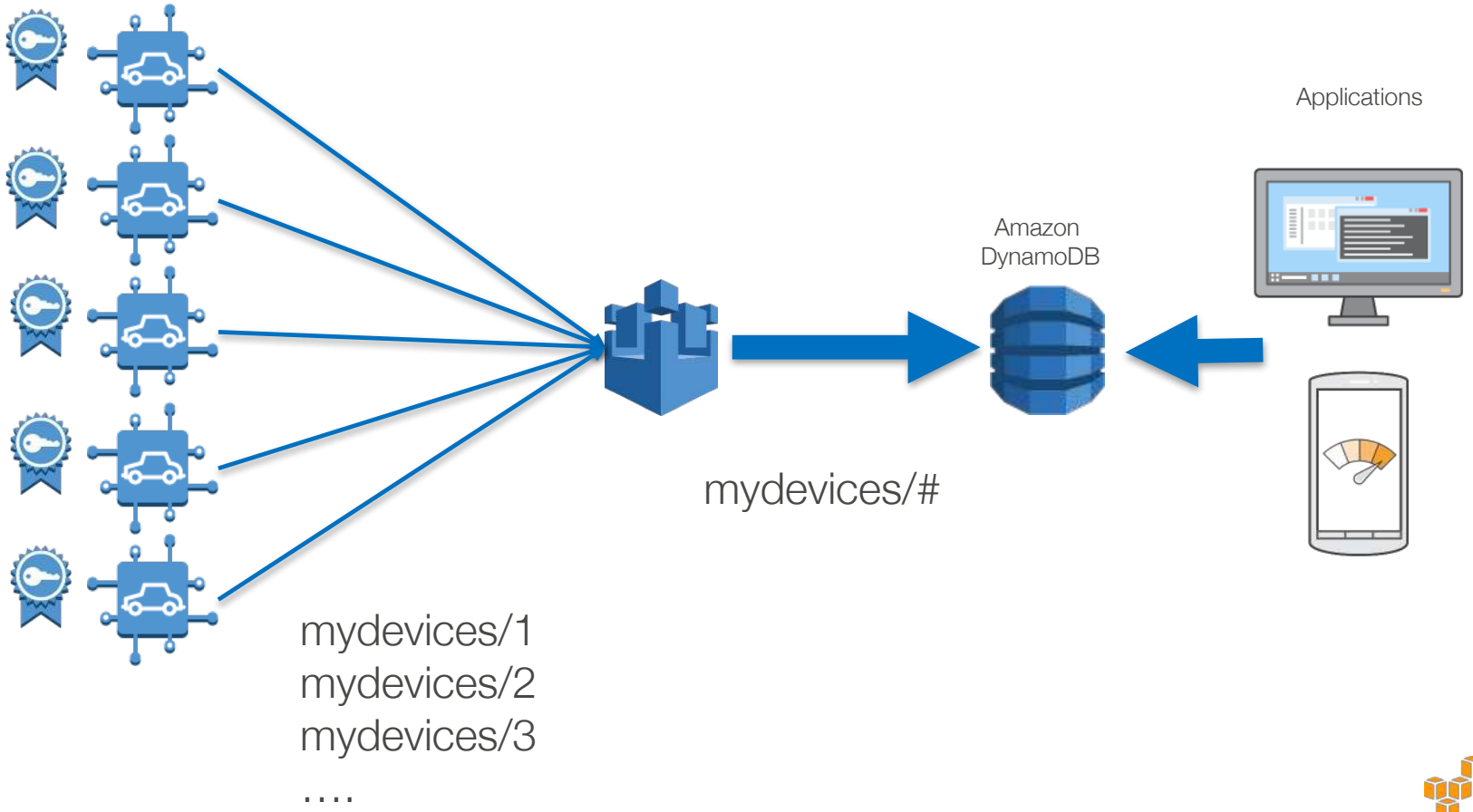
MQTT: device-to-device communication



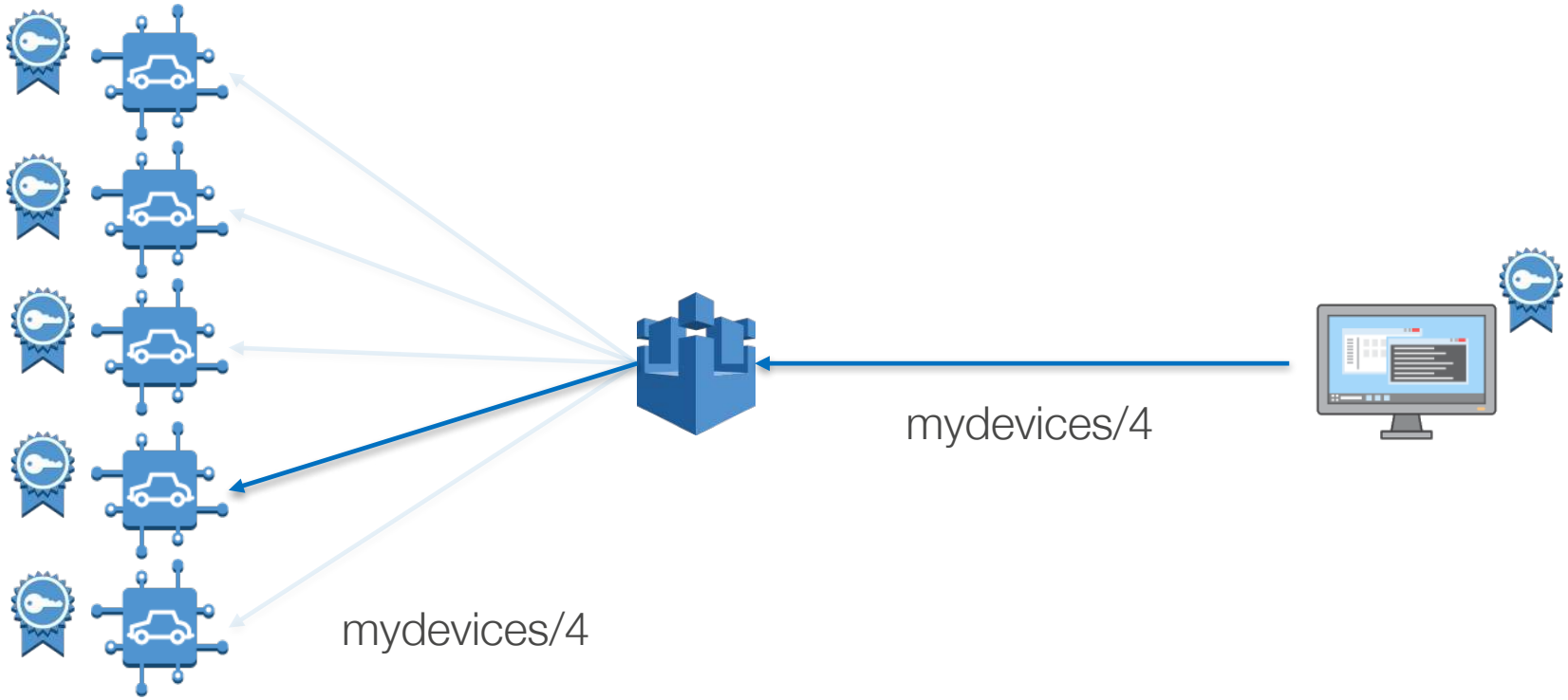
MQTT: collect data from a device



MQTT: aggregate data from many devices



MQTT: update a device



MQTT: QoS 0 (at most once)

MQTT: QoS 1 (at least once)

1

2

3

PUBLISH QoS1

4

5

PUBLISH QoS1

4

PUBACK

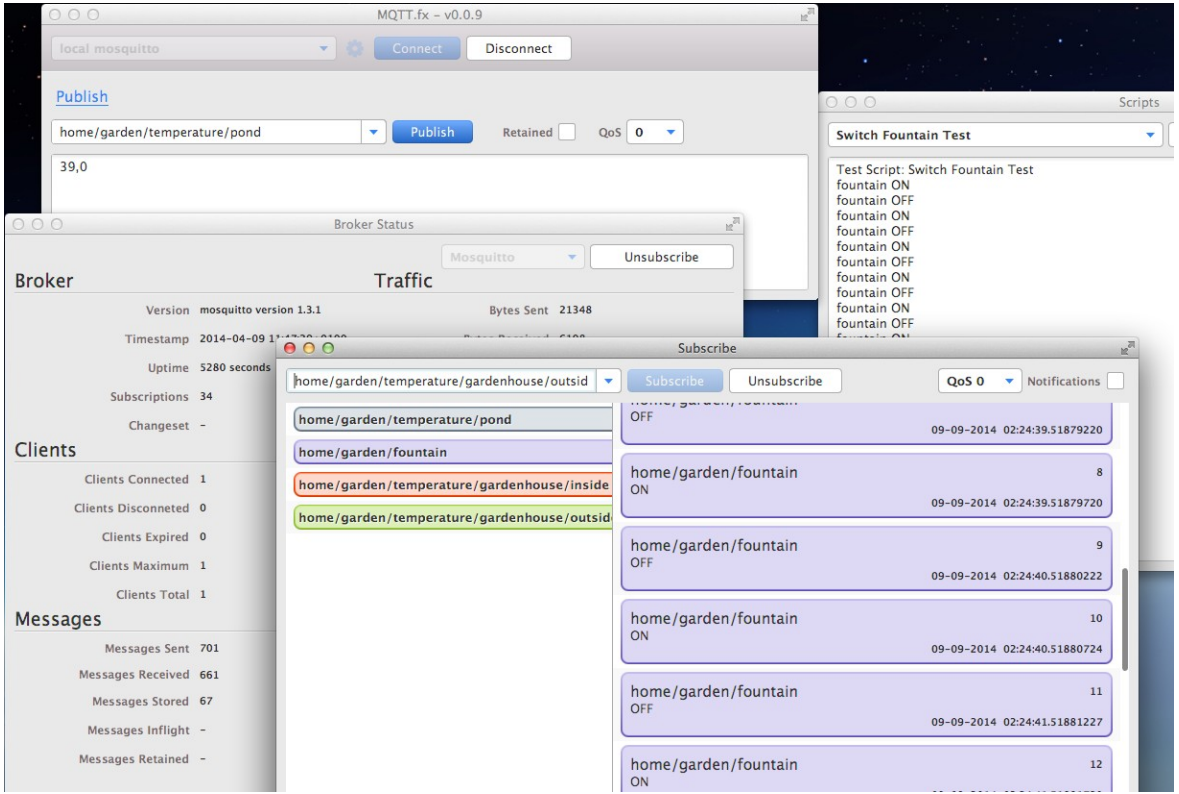
6

1,2,3,4,5,6

amazon web services

MQTT.fx

http://mqttfx.jfx4ee.org/



amazon web services

Arduino : subscribing and publishing to a topic

```
if ((rc=myClient.subscribe(”myTopic", 1, msg_callback)) != 0)
{
    Serial.println("Subscribe failed!");
   Serial.println(rc);
}
```

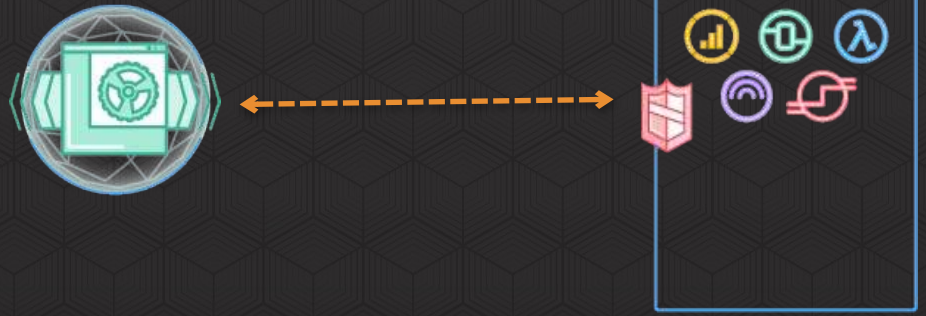
```
if((rc = myClient.publish(”myTopic", msg, strlen(msg),
    1, false)) != 0)
{
    Serial.println("Publish failed!");
    Serial.println(rc);
}
```

amazon web services

Arduino : callback for incoming messages

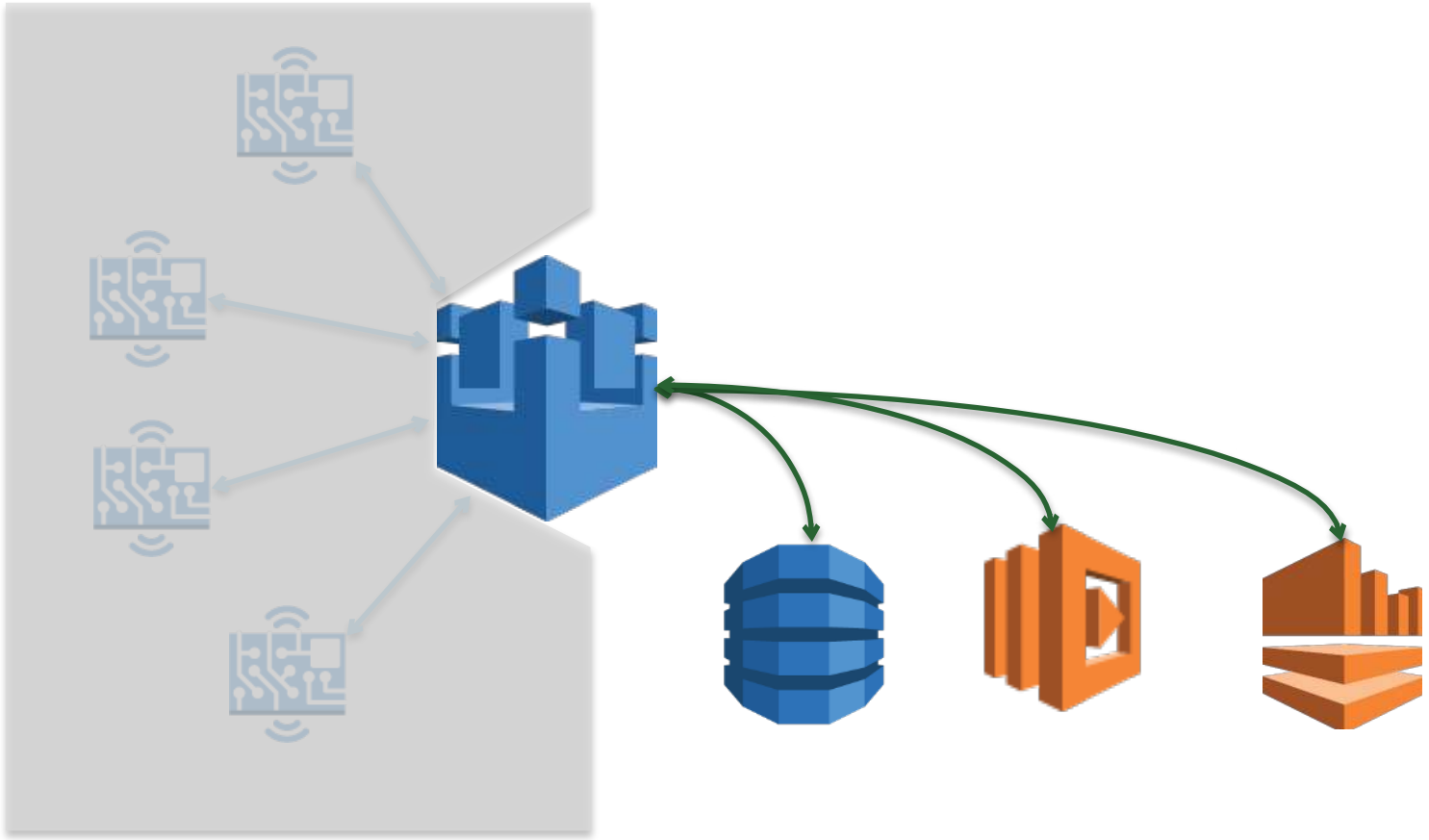
```
// Basic callback function that prints out the message

void msg_callback(char* src, int len) {
   Serial.println("CALLBACK:");
   for(int i = 0; i < len; i++) {
         Serial.print(src[i]);
   }
   Serial.println("");
}
```



Rules

Granting AWS IoT access to AWS services



Defining a trust policy for AWS IoT

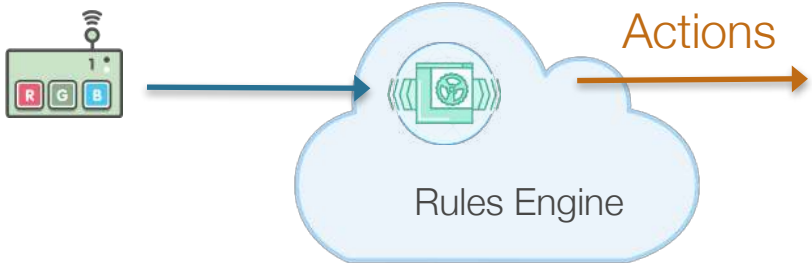
```
% cat iot-role-trust.json
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"",
         "Effect":"Allow",
         "Principal":{
            "Service":"iot.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

Applying the trust policy to AWS IoT

```
% aws iam create-role --role-name my-iot-role
  --assume-role-policy-document file://iot-role-trust.json
{
    "Role": {
        "AssumeRolePolicyDocument": {…},
        "RoleId": "AROAJY7VZX5GEZ3Q7ILU4",
        "CreateDate": "2016-03-19T12:07:03.904Z",
        "RoleName": "my-iot-role",
        "Path": "/",
        "Arn": "arn:aws:iam::613904931467:role/my-iot-role"
    }
}
```

AWS IoT Rules

Rules connect AWS IoT to External Endpoints and AWS Services



Actions

1. AWS Services
(Direct Integration)

Amazon S3
Amazon DynamoDB
Amazon Kinesis
Amazon SNS
AWS Lambda
Amazon SQS
Amazon CloudWatch
Amazon Elasticsearch
Amazon Machine Learning

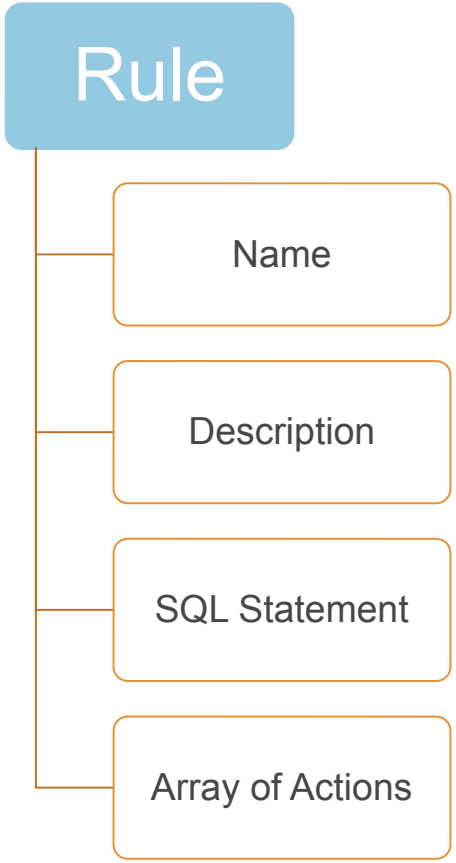
2. Rest of AWS
(via Amazon Kinesis, AWS Lambda, Amazon S3, and more)

Amazon RDS
Amazon Glacier
Amazon Redshift
Amazon EC2

3. External Endpoints
(via Lambda and SNS)

amazon web services

AWS IoT Rules Engine



Simple & Familiar Syntax

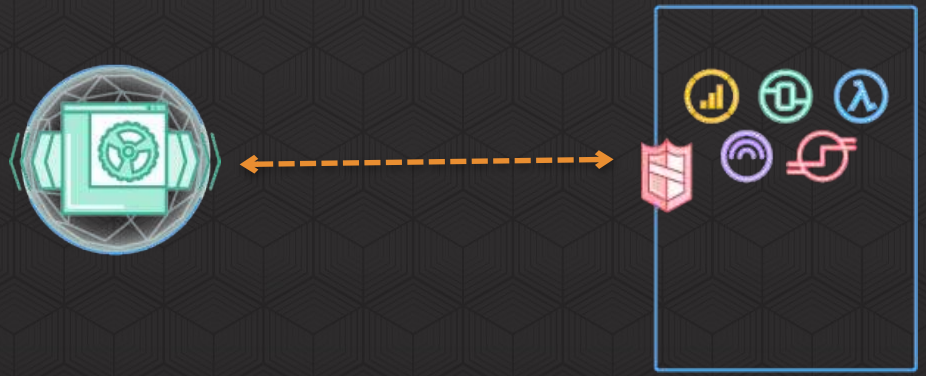
- SQL Statement to define topic filter
- Optional WHERE clause
- Advanced JSON support

Many functions available

- String manipulation (regex support)
- Mathematical operations
- Crypto support
- UUID, Timestamp, rand, etc.

Creating a rule to write to DynamoDB

```
% cat topic1-dynamodb-rule.json
{
  "sql": "SELECT * FROM 'topic1'",
  "ruleDisabled": false,
  "actions": [{
      "dynamoDB": {
          "tableName": "iot-topic1-table",
          "roleArn": "arn:aws:iam::613904931467:role/my-iot-role",
          "hashKeyField": "deviceId",
          "hashKeyValue": "${deviceId}",
          "rangeKeyField": "timestamp",
          "rangeKeyValue": "${timestamp()}"
      }
  }]
}
% aws iot create-topic-rule --rule-name topic1-dynamodb-rule
--topic-rule-payload file://topic1-dynamodb-rule.json
```



Debugging

How can you debug AWS IoT applications?

- Testing with MQTT.fx (or a similar tool) is not enough
- CloudWatch Logs: the only way to see what is happening inside AWS IoT
  - Permission issue
  - Rule issue
  - Incorrect JSON message
  - Etc.
- These logs are not enabled by default
  - Define a policy allowing AWS IoT to access CloudWatch logs
  - Attach the policy to the AWS IoT role (same one as for external services)

Defining a policy for CloudWatch Logs

```
% cat iot-policy-logs.json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:PutMetricFilter",
                "logs:PutRetentionPolicy"
             ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Enabling CloudWatch Logs for AWS IoT

```
% aws iam create-policy
--policy-name my-iot-policy-logs --policy-document file://iot-policy-logs.json
{
    "Policy": {
        "PolicyName": "my-iot-policy-logs",
        "CreateDate": "2016-03-19T12:24:16.072Z",
        "AttachmentCount": 0,
        "IsAttachable": true,
        "PolicyId": "ANPAIK73XIV3QG5FF5TX6",
        "DefaultVersionId": "v1",
        "Path": "/",
        "Arn": "arn:aws:iam::613904931467:policy/my-iot-policy-logs",
        "UpdateDate": "2016-03-19T12:24:16.072Z"
    }
}

% aws iam attach-role-policy --role-name my-iot-role
--policy-arn "arn:aws:iam::613904931467:policy/my-iot-policy-logs"

% aws iot set-logging-options
--logging-options-payload roleArn="arn:aws:iam::613904931467:role/my-iot-role",logLevel="INFO"
```

Demo : logging events in CloudWatch Logs

2016-03-19 15:34:23.300 TRACEID:eb1a7666-28c3-4ab4-83a2-f87f66406025
PRINCIPALID:e016283e5191f574f1f76c0278bee9e4d2d4b355d5299b6d16ac4c527f8522b0 [INFO]
EVENT:PublishEvent TOPICNAME:topic1 MESSAGE:PublishIn Status: SUCCESS

2016-03-19 15:34:23.403 TRACEID:eb1a7666-28c3-4ab4-83a2-f87f66406025
PRINCIPALID:e016283e5191f574f1f76c0278bee9e4d2d4b355d5299b6d16ac4c527f8522b0 [INFO]
EVENT:MatchingRuleFound TOPICNAME:topic1 CLIENTID:6071974a42ea4594a96446a137b0520b MESSAGE:Matching rule found: topic1_dynamodb_rule

2016-03-19 15:34:23.887 TRACEID:eb1a7666-28c3-4ab4-83a2-f87f66406025
PRINCIPALID:e016283e5191f574f1f76c0278bee9e4d2d4b355d5299b6d16ac4c527f8522b0 [INFO]
EVENT:DynamoActionSuccess TOPICNAME:topic1 CLIENTID:6071974a42ea4594a96446a137b0520b MESSAGE:Successfully put Dynamo record. Message arrived on: topic1, Action: dynamo, Table: iot-topic1-table, HashKeyField: deviceId, HashKeyValue: 1234, RangeKeyField: timestamp, RangeKeyValue: 1458401663404

2016-03-19 17:02:46.691 TRACEID:f8ee7d3f-3c3c-4c23-8458-bf92c6c56c0b
PRINCIPALID:e016283e5191f574f1f76c0278bee9e4d2d4b355d5299b6d16ac4c527f8522b0 [INFO]
EVENT:PublishEvent TOPICNAME:topic1 MESSAGE:PublishIn Status: SUCCESS

2016-03-19 17:02:46.804 TRACEID:f8ee7d3f-3c3c-4c23-8458-bf92c6c56c0b
PRINCIPALID:e016283e5191f574f1f76c0278bee9e4d2d4b355d5299b6d16ac4c527f8522b0 [INFO]
EVENT:MatchingRuleFound TOPICNAME:topic1 CLIENTID:6071974a42ea4594a96446a137b0520b MESSAGE:Matching rule found: topic1_dynamodb_rule

2016-03-19 17:02:47.268 TRACEID:f8ee7d3f-3c3c-4c23-8458-bf92c6c56c0b
PRINCIPALID:e016283e5191f574f1f76c0278bee9e4d2d4b355d5299b6d16ac4c527f8522b0 [ERROR]
EVENT:DynamoActionFailure TOPICNAME:topic1 CLIENTID:6071974a42ea4594a96446a137b0520b MESSAGE:Failed to put Dynamo record. The error received was One or more parameter values were invalid: An AttributeValue may not contain an empty string (Service: AmazonDynamoDBv2; Status Code: 400; Error Code: ValidationException; Request ID: CTUP5HKKUONPR9718LQ9QC4J9VVV4KQNSO5AEMVJF66Q9ASUAAJG). Message arrived on: topic1, Action: dynamo, Table: iot-topic1-table, HashKeyField: deviceId, HashKeyValue: , RangeKeyField: timestamp, RangeKeyValue: 1458406966804

amazon web services

Now it’s your turn!

https://aws.amazon.com/iot/

https://aws.amazon.com/free/

https://aws.amazon.com/usergroups/europe/

More sessions

- 8/11, 10:00 A 60-minute tour of AWS Compute
- 9/11, 10:00 DevOps on AWS
- 9/11, 11:00 Running Docker clusters on AWS

- 21/11, 11:00 Move fast, build things with AWS
- 22/11, 11:00 Deep Dive on Amazon RDS

Thank You !

Julien Simon
Principal Technical Evangelist
Amazon Web Services

julsimon@amazon.com
@julsimon

amazon web services