IOT Solution Alternatives

Build or Leverage AWS Iot (IAAS)
February 5, 2020

Agenda

- IOT Problem to solve
- Solution Requirements
- Scale issues
- Build your own alternative
- AWS IOT (“infrastructure as a service”) approach
  - What is AWS IOT
  - How does it work
  - Interfaces to ASW IOT
  - Solution example
  - Solution walk thru
- Comparison of Solutions
- Walk through live demo
- Discussion

IOT Project Requirements

IOT Project Needs

- Automation to care for plants remotely
  - Travel constantly, leave skiing on weekend
  - No one is around to care for the plants
- System Needs
  - System to access anywhere in world
  - See, sense environment, and care for plants
  - Scalable, reliable, easy to use, easy to repair
  - Dozens of sensors and controllers required
  - Capabilities to enable a commercial offering

Initial Product Thoughts

- Simple Phone App Management
- Light control
- Moisture sensors
- Temperature, CO_2, humidity, lumens
- Pump Controls watering/fertilizing
- Cameras
- Soil PH
- Image analysis for
  - Growth Rate
  - Health analysis
- Support dozens of sensors and controllers (1000's commercially)
- Multiples of each
- Low touch easy install
- Easy to diagnose and repair
- Send alerts
  - Email/SMS
- Complete life cycle monitoring. Seed to consumption

Decisions to make

- Component Selection
- Processor selection
- SW Selection
- Technology to connect all devices
- Device topology
- Network topology and connections
- App environment

Component Selection

- Very low cost devices
- Commodity devices that draw limited power
- UL listed safe devices to avoid issues
- Ethernet enabled
  - Device scale, easy integration, securable
  - Readily available on low cost device

Processor/SW Selection

- Two classes of processor needed
  - Simple sensor controller integration (sensors/controllers)
    - Minimize cost of entire solution
    - GPIO control is pretty simple operation
  - Complex service oriented needs
    - PI Broker, camera services, other complex devices
    - Want to leverage Open Source SW available
    -

Device Topology and Connection

- Needs to scale to 100’s of devices
- Repair and replace components
- Built with commodity SW and HW
- Needed to span 1000’s of square feet
- Secure and reliable
- Cost effective

Answer? - Ethernet

Target Markets

Markets to address

- Home Growers
  - Small scale per setup(dozens of controllers/sensors)
  - Single growing setup
  - Low costs required
  - Easy setup and operation
  - Packaged with a fixed set of equipment
- Commercial Growers
  - Large Scale (1000s of controllers)
  - Multiple grow rooms separately controlled
  - Low costs required
  - Easy manageability and repair at large scale
  - Costs and billing must be well understood
  - Seed to consumption audit trail legally required

Current Market Coverage

- Home Growers
  - Very niche
  - Limited offering
  - Zero compatibility between vendors
- Commercial Growers
  - Custom implementations
  - Limited standards
  - Seed to consumption data analysis limited

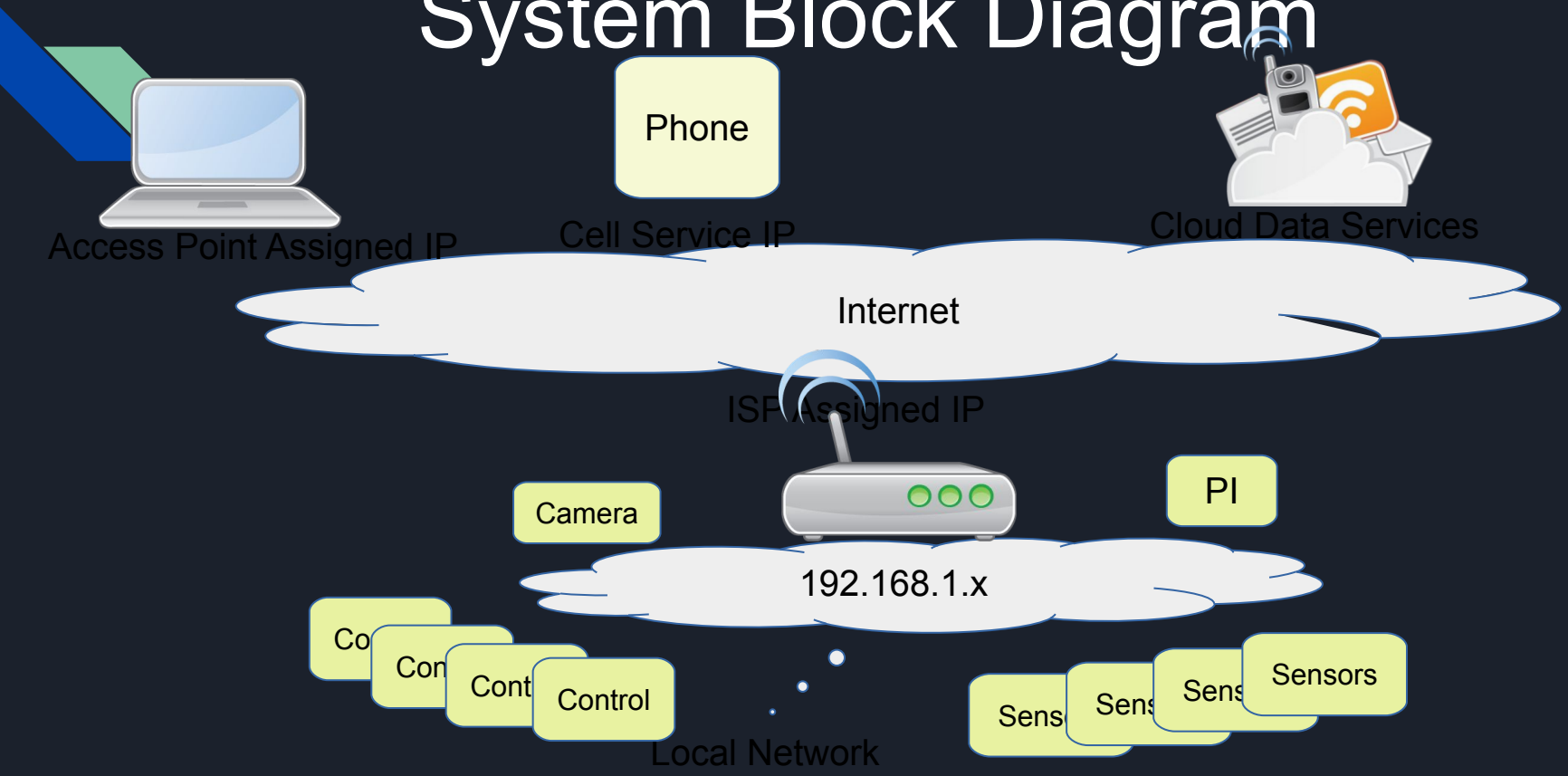
Custom Build

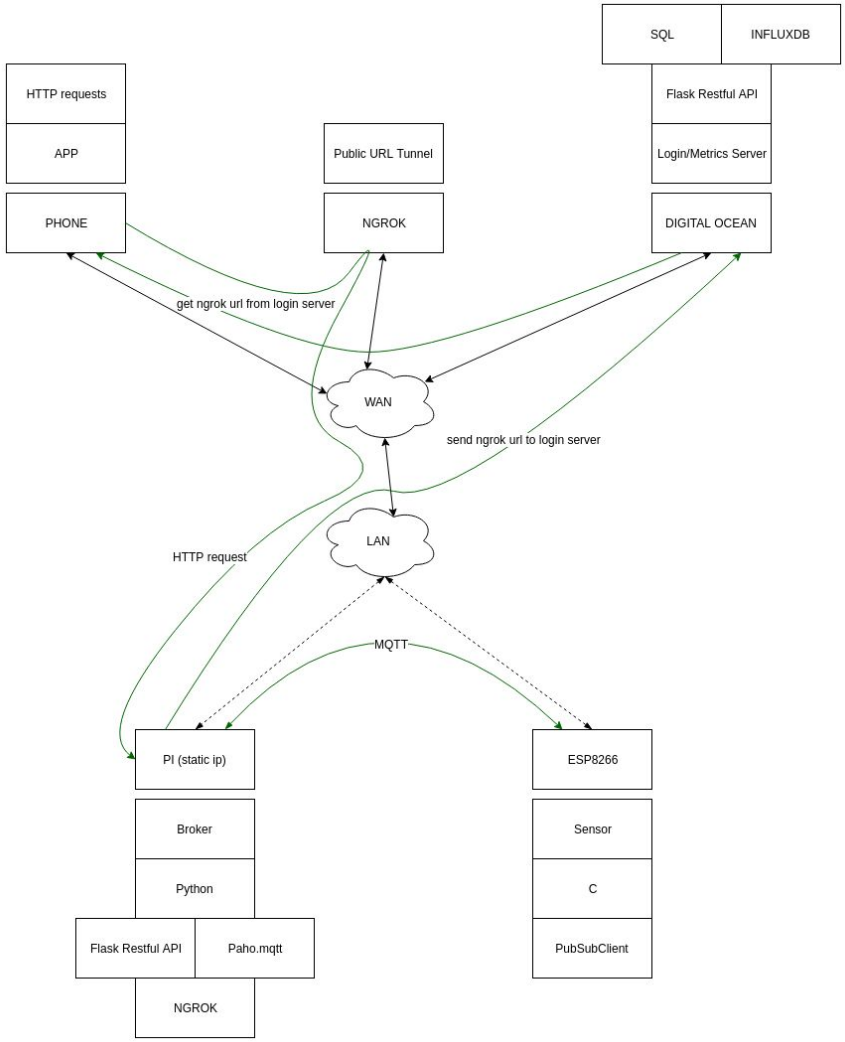
Implementation decisions

- Raspberry Pi home gateway to Internet
- Simple Controllers/Sensors connected to local Lan with Arduinos
- Cameras were connected to Pi Zeros for function and cost purposes
- Pi, controllers, and sensors connect with MQTT messaging protocol
- PI implemented a Restful API to be managed from Internet
- Well known IP address achieved with ngrok service
- Digital Ocean was a low cost cloud compute provider
- Py code was implemented in Python

Prototypes were up and running fairly quickly. Security was password protected login and data transferred was unencrypted.

System Block Diagram




SQL
INFLUXDB
HTTP requests
APP
PHONE
Public URL Tunnel
NGROK
Flask Restful API
Login/Metrics Server
DIGITAL OCEAN
get ngrok url from login server
WAN
send ngrok url to login server
LAN
HTTP request
MQTT
PI (static ip)
Broker
Python
Flask Restful API
Paho.mqtt
NGROK
ESP8266
Sensor
C
PubSubClient

Device Connection Strategy

Simple Configuration

- DHCP assigns all local devices
- PI static 192.168.1.2
- Devices register w/PI
- Mac address is UID
- Broker uses UID for pub/sub
- Ques constant thru restart/power cycles
- Ngrok as well non-IP broker

Issues Encountered

- 192.168.1.2 set in router
- Persistent messages wreaked havoc
- Have a more complex PI and device config, but not worth it now
- Only have to port forward PI
- Non static IP’s

PI Broker Operation

- Subscribe devices to PI device namespace
- Set up device type node map on msg receipt
- Persist that node map
- Enable Restful server
- Start system services

- System Services
  - Light management
  - Pump Management
  - Sensor alert
  - Camera services
  - API services
  - Metric collection

All internet access in through the restful server

AWS IOT

AWS IoT provides secure, bi-directional communication between Internet-connected devices and the AWS Cloud. This enables you to collect telemetry data from multiple devices; then store and analyze the data and manage the devices at extremely large scale.

AWS IOT Components

- Alexa Voice Service (AVS) Integration for AWS IoT
- Custom Authentication service
- Device gateway
- Device Provisioning service
- Device shadow
- Device Shadow service
- Group registry
- Jobs service
- Message broker
- Registry
- Rules engine
- Security and Identity service

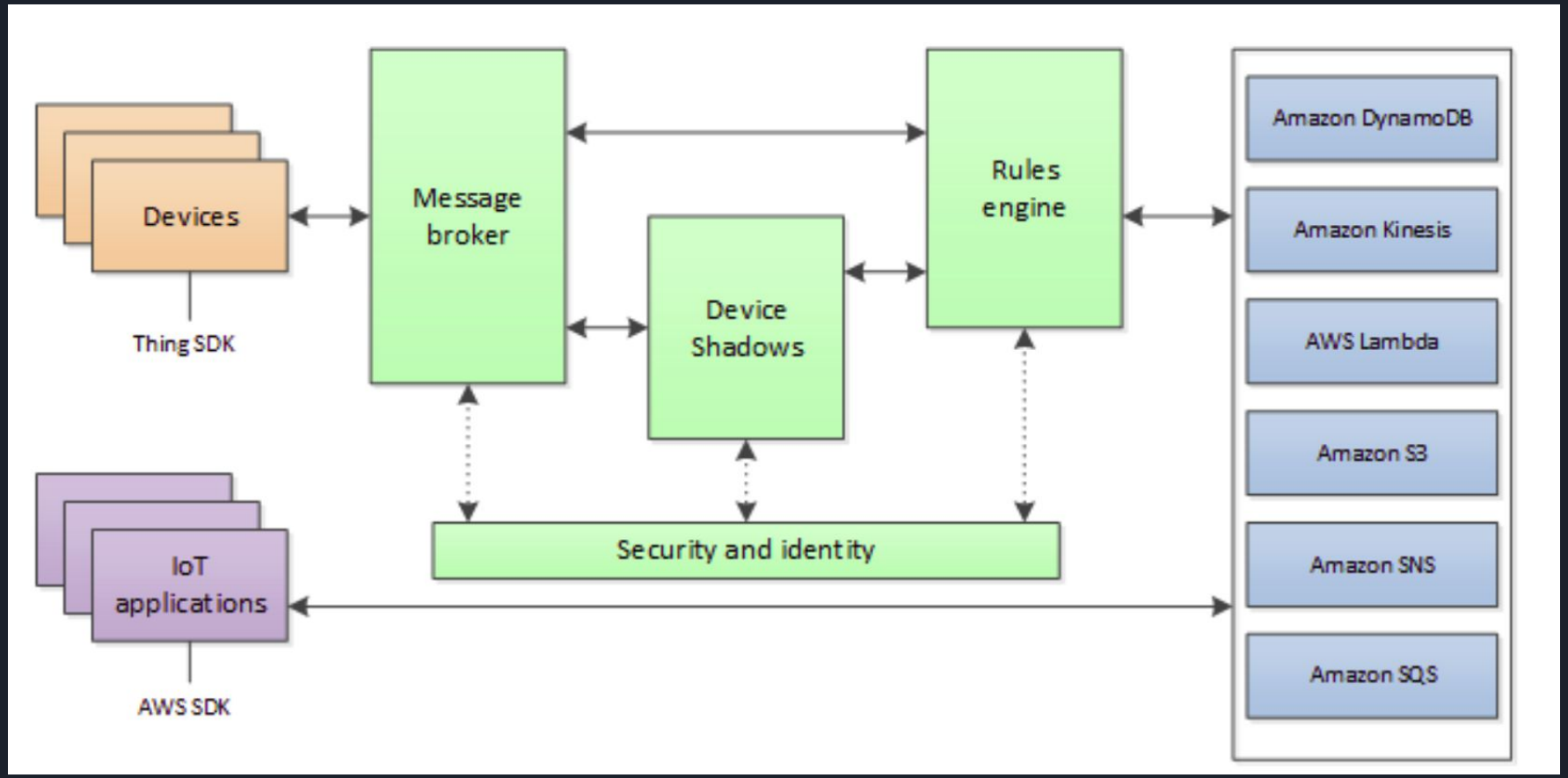
Accessing AWS IoT

- **AWS Command Line Interface (AWS CLI)**—Run commands for AWS IoT on Windows, macOS, and Linux. These commands allow you to create and manage things, certificates, rules, and policies. To get started, see the AWS Command Line Interface User Guide. For more information about the commands for AWS IoT, see iot in the *AWS CLI Command Reference*.
- **AWS IoT API**—Build your IoT applications using HTTP or HTTPS requests. These API actions allow you to programmatically create and manage things, certificates, rules, and policies. For more information about the API actions for AWS IoT, see Actions in the *AWS IoT API Reference*.
- **AWS SDKs**—Build your IoT applications using language-specific APIs. These SDKs wrap the HTTP/HTTPS API and allow you to program in any of the supported languages. For more information, see AWS SDKs and Tools.
- **AWS IoT Device SDKs**—Build applications that run on devices that send messages to and receive messages from AWS IoT. For more information see, AWS IoT SDKs.

Integrated Available Services

- **Amazon Simple Storage Service**—Provides scalable storage in the AWS Cloud. For more information, see Amazon S3.
- **Amazon DynamoDB**—Provides managed NoSQL databases. For more information, see Amazon DynamoDB.
- **Amazon Kinesis**—Enables real-time processing of streaming data at a massive scale. For more information, see Amazon Kinesis.
- **AWS Lambda**—Runs your code on virtual servers from Amazon EC2 in response to events. For more information, see AWS Lambda.
- **Amazon Simple Notification Service**—Sends or receives notifications. For more information, see Amazon SNS.
- **Amazon Simple Queue Service**—Stores data in a queue to be retrieved by applications. For more information, see Amazon SQS.

How does it work?



Details

Security

Thing groups

Billing Groups

Shadow

Interact

Activity

Jobs

Violations

Defender metrics

new

Shadow ARN

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

```
arn:aws:iot:us-west-2:376625362739:thing/TempHumdLgt
```

Shadow Document

Delete **Edit**

Last update: Feb 5, 2020 11:27:19 AM -0500

Shadow state:


```
{
  "reported": {
    "temp": 29,
    "Humdity": 35,
    "Lumens": 890
  }
}
```


Metadata:


```
{
  "metadata": {
    "reported": {
      "temp": {
        "timestamp": 1580920039
      },
      "Humdity": {
        "timestamp": 1580920039
      },
      "Lumens": {
        "timestamp": 1580920039
      }
    }
  },
  "version": 15
}
```

Device Shadow Service Topic Namespace

$aws/things/myLightBulb/shadow/update/accepted

$aws/things/myLightBulb/shadow/update/rejected

$aws/things/myLightBulb/shadow/update/delta

$aws/things/myLightBulb/shadow/get/accepted

$aws/things/myLightBulb/shadow/get/rejected

$aws/things/myLightBulb/shadow/delete/accepted

$aws/things/myLightBulb/shadow/delete/rejected

$aws/things/myLightBulb/shadow/update/documents

Device Shadow service is implemented and orchestrated through this topic namespace. Coordination between device and applications is achieved through these message buffers.

AWS IoT

AWS IoT is a managed cloud platform that lets connected devices - cars, light bulbs, sensor grids, and more - easily and securely interact with cloud applications and other devices.

Get started

AWS IoT

Monitor

Onboard

Manage

Greengrass

Secure

Defend

Act

Test

Software

Settings

Learn

Welcome to the AWS IoT Console

To get started, you can jump into the recommended starting points below, or explore other learning resources as needed.

See how AWS IoT works

Explore an interactive tutorial through the components of AWS IoT.

Start the tutorial

It takes 5 minutes

Connect to AWS IoT

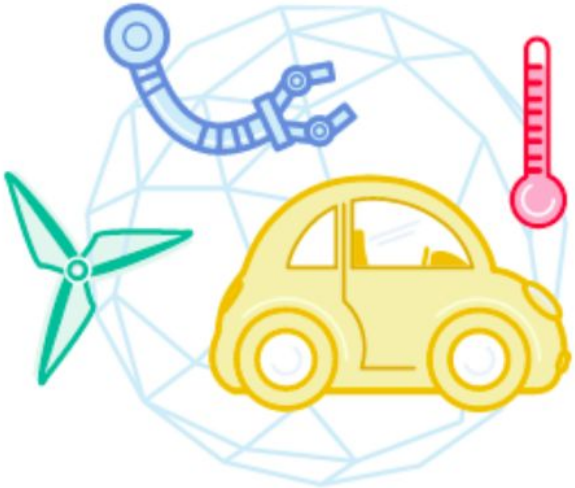
Connect a device, a mobile or web app to AWS IoT in a few easy steps!

View connection options

Explore documentation

The AWS IoT documentation is a great resource for more details.

Go to documentation



You don't have any things yet

A thing is the representation of a device in the cloud.

Learn more

Register a thing

Creating AWS IoT things

An IoT thing is a representation and record of your phyisical device in the cloud. Any physical device needs a thing record in order to work with AWS IoT. Learn more.

Register a single AWS IoT thing

Create a thing in your registry

Create a single thing

Bulk register many AWS IoT things

Create things in your registry for a large number of devices already using AWS IoT, or register devices so they are ready to connect to AWS IoT.

Create many things

Cancel

Create a single thing

CREATE A THING

Add your device to the thing registry

STEP 1/3

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

Name

MyIotThing

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.

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	c3c4ff2375.cert.pem	Download
A public key	c3c4ff2375.public.key	Download
A private key	c3c4ff2375.private.key	Download

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

Activate

Cancel

Done

Attach a policy

You don't have any policies yet

AWS IoT policies give things permission to access AWS IoT resources (like other things, MQTT topics, or thing shadows).

Learn more

Create a policy

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

My_IoT_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

aws Services Resource Groups stevenpo64 Oregon Support

AWS IoT

Monitor
Onboard
Manage
Things
Types
Thing Groups
Billing Groups
Jobs
Tunnels
Greengrass
Secure
Defend
Act
Test
Software
Settings

Things

Create

Search things

Fleet Indexing

List

Name	Type
Pi3Group_Core	NO TYPE
MyRaspberryPi	NO TYPE
ThirdBulb	LIGHTBULB
MyOtherLightBulb	LIGHTBULB
MyLightBulb	LIGHTBULB
SampleIoTThing	NO TYPE
MyRPi	NO TYPE
MyIoTThing	NO TYPE

CERTIFICATE

c3c4ff237568f6f99c92b729f00c83fa7ef43cb77fef8f7ea2aa470d990c8816

ACTIVE

Actions

- Activate
- Deactivate
- Revoke
- Accept transfer
- Reject transfer
- Revoke transfer
- Start transfer
- Attach policy
- Attach thing
- Download
- Delete

- **Details**
- Policies
- Things
- Non-compliance

Certificate ARN

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

```
arn:aws:iot:us-west-2:504350838278:cert/c3c4ff237568f6f99c92b729
```

Details

Issuer
OU=Amazon Web Services O\=Amazon.com Inc. L\=Seattle ST\=Washington C\=US
Subject
CN=AWS IoT Certificate
Create date
Aug 6, 2019 2:09:27 PM -0700
Effective date
Aug 6, 2019 2:07:27 PM -0700
Expiration date
Dec 31, 2049 3:59:59 PM -0800

Attach policies to certificate(s)

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

c3c4ff237568f6f99c92b729f00c83fa7ef43cb77fef8f7ea2aa470d990c8816

Choose one or more policies

Search policies

- [x] My_IoT_Policy View

1 policy selected Cancel Attach

Device is ready

Now,

Set up a rule!

AWS IoT

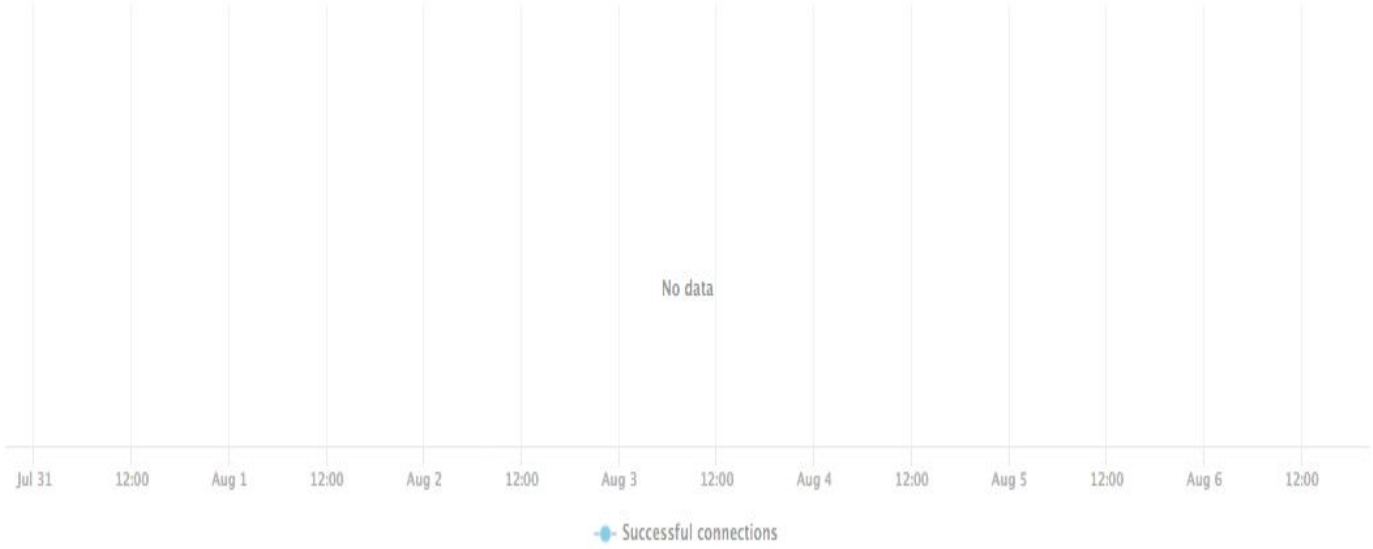
Monitor

Sample period: One day

Time range: Week

- Monitor
- Onboard
- Manage
- Greengrass
- Secure
- Defend
- Act
- Test

Successful connections



AWS IoT

Monitor

Onboard

Manage

Greengrass

Secure

Defend

Act

Test

Software

Settings

Learn

Rules

Create

Search rules

Card

Create a rule

Create a rule to evaluate messages sent by your things and specify what to do when a message is received (for example, write data to a DynamoDB table or invoke a Lambda function).

Name

MySNSRule

Description

A more complex SNS rule.

Rule query statement

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

Using SQL version

2016-03-23

Rule query statement

SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. To learn more, see AWS IoT SQL Reference.

```
1  SELECT *, topic(3) as thing FROM '$aws/things/+/shadow/update/accepted'
```

Set one or more actions

Select one or more actions to happen when the above rule 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)

Add action

Select an action

Select an action.

- Insert a message into a DynamoDB table
  DYNAMODB
- Split message into multiple columns of a DynamoDB table (DynamoDBv2)
  DYNAMODBV2
- Send a message to a Lambda function
  LAMBDA
- Send a message as an SNS push notification
  SNS
- Send a message to an SQS queue
  SQS
- Send a message to an Amazon Kinesis Stream
  AMAZON KINESIS
- Republish a message to an AWS IoT topic
  AWS IOT REPUBLISH
- Store a message in an Amazon S3 bucket
  S3
- Send a message to an Amazon Kinesis Firehose stream
  AMAZON KINESIS FIREHOSE
- Send message data to CloudWatch
  CLOUDWATCH METRICS
- Change the state of a CloudWatch alarm
  CLOUDWATCH ALARMS
- Send a message to the Amazon Elasticsearch Service
  AMAZON ELASTICSEARCH
- Send a message to a Salesforce IoT Input Stream
  SALESFORCE IOT
- Send a message to IoT Analytics
  IOT ANALYTICS
- Send a message to an IoT Events Input
  IOT EVENTS
- Start a Step Functions state machine execution
  STEP FUNCTIONS

Cancel

Configure action

Configure action

Send a message as an SNS push notification

SNS

*SNS target

No topic selected | Create | Select

Message format

Select

Choose or create a role to grant AWS IoT access to perform this action.

No role selected | Select

Cancel

Add action

Create

Name

MySNSTopic

Cancel

Create

Configure action

Send a message as an SNS push notification

SNS

*SNS target

MySNSTopic | Create | Clear | Select

Message format

RAW

Choose or create a role to grant AWS IoT access to perform this action.

No role selected | Create Role | Select

Cancel

Add action

Create a rule

Create a rule to evaluate messages sent by your things and specify what to do when a message is received (for example, write data to a DynamoDB table or invoke a Lambda function).

Name

MySNSRule

Description

A more complex SNS rule.

Rule query statement

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

Using SQL version

2016-03-23

Rule query statement

SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. To learn more, see AWS IoT SQL Reference.

```
1 SELECT *, topic(2) as thing FROM '$aws/things/+/shadow/update/accepted'
```

Set one or more actions

Select one or more actions to happen when the above rule 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)

Send a message as an SNS push notification
MySNSTopic

Remove Edit

Add action

Error action

Optionally set an action that will be executed when something goes wrong with processing your rule.

Add action

Tags

Apply tags to your resources to help organize and identify them. A tag consists of a case-sensitive key-value pair. Learn more about tagging your AWS resources.

Tag name

Provide a tag name, e.g. Manufacturer

Value

Provide a tag value, e.g. Acme-Corporation

Clear

Add another

Cancel

Create rule

AWS Walk thru

AWS has a wealth of services, very rich feature function, and scales. Ever expanding feature function.

Implementation Comparison

Homegrown Solution

- Quickly Prototyped
- MQTT message broker
- No security
- Single User scale
- Scales to 100's of devices
- Custom built device infrastructure
- Heavily leverages Open Source SW
- Multiple tools infrastructure to integrate
- Free infrastructure costs

AWS IOT (IaaS)

- Quickly prototyped
- MQTT message broker
- Complete Commercial Infrastructure
  - Complete Security Solution
  - Extremely large scale
  - Multiple User capabilities
  - Separate resource billing
  - Compute, analytics, storage, Database,....
  - Security Audits, Update tools,....
- Relatively low infrastructure costs
- SDK's and libraries available for many devices