

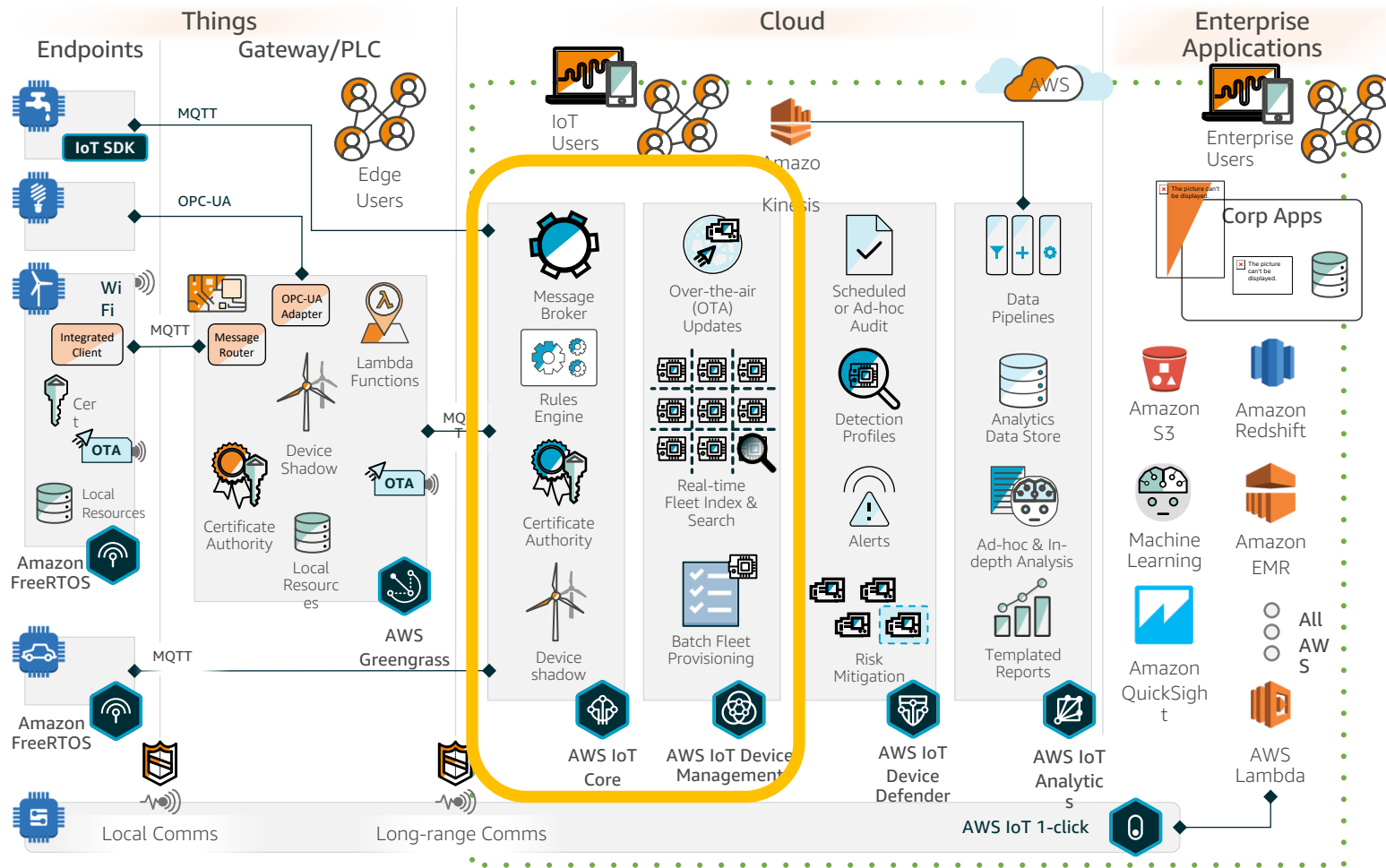


Device Provisioning Options with AWS IoT

Philipp Sacha

Specialist Solutions Architect IoT, Amazon Web Services

IoT with AWS



AWS IoT – Starting To Explore...



Monitor



Onboard



Manage



Secure



Act



Test



Software



Settings



Learn

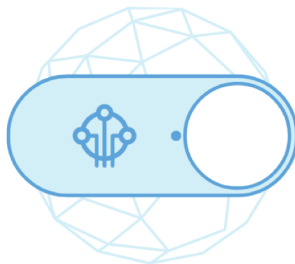
Connect to AWS IoT



Configuring a device

Connect a device or your computer to AWS IoT using the connection wizard for AWS IoT Device SDKs.

[Get started](#)



AWS IoT Button

The AWS IoT Button is a single-purpose device that sends a message to AWS IoT with a press of a button.

[Configure a button](#)

Don't have a button? [Buy one](#)



AWS IoT Starter Kit

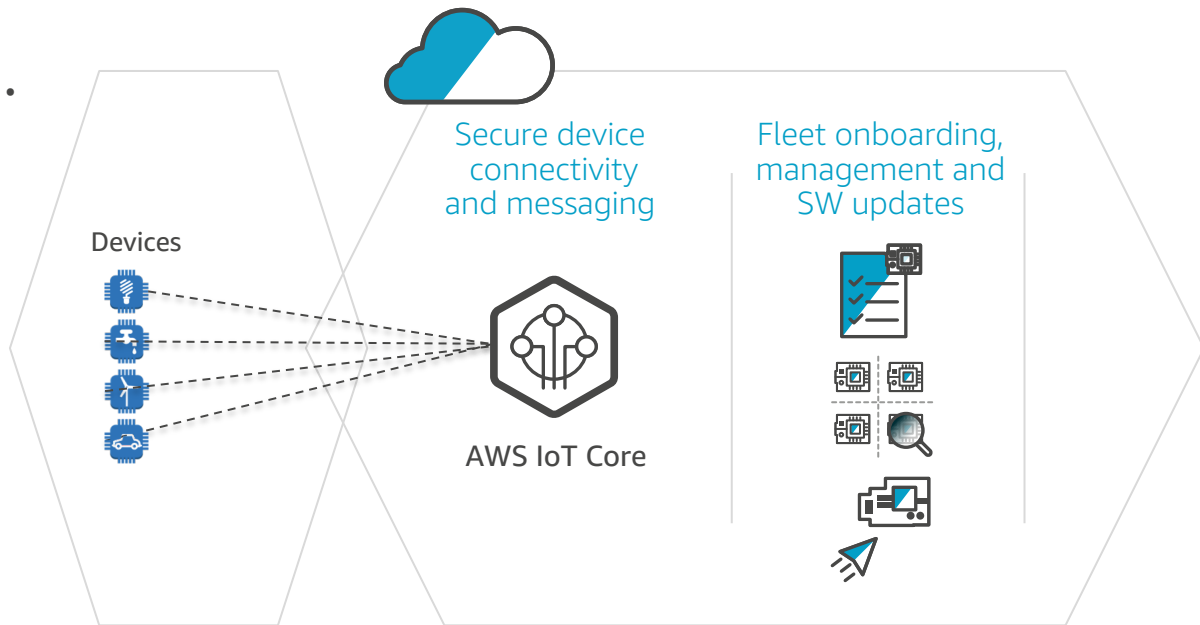
Browse AWS IoT Starter Kits that were made for connecting to AWS IoT and getting started with the service.

[Browse starter kits](#)

At Scale - How To Provision Devices?

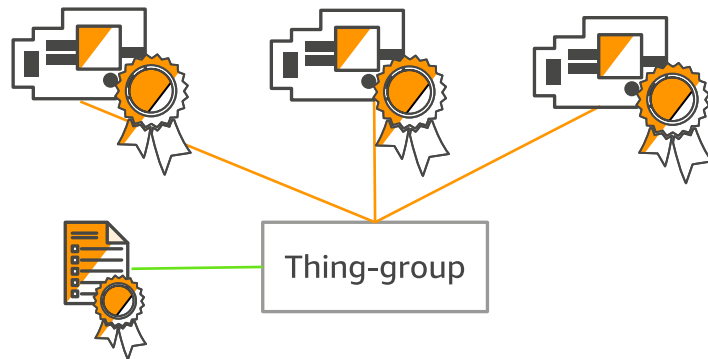
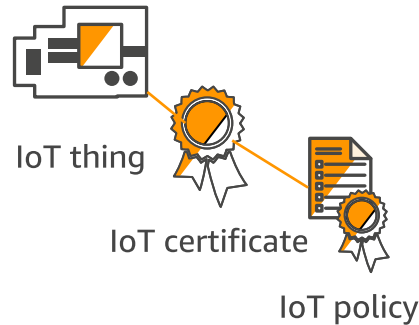
Architecture is developed...

How Do I onboard my devices???



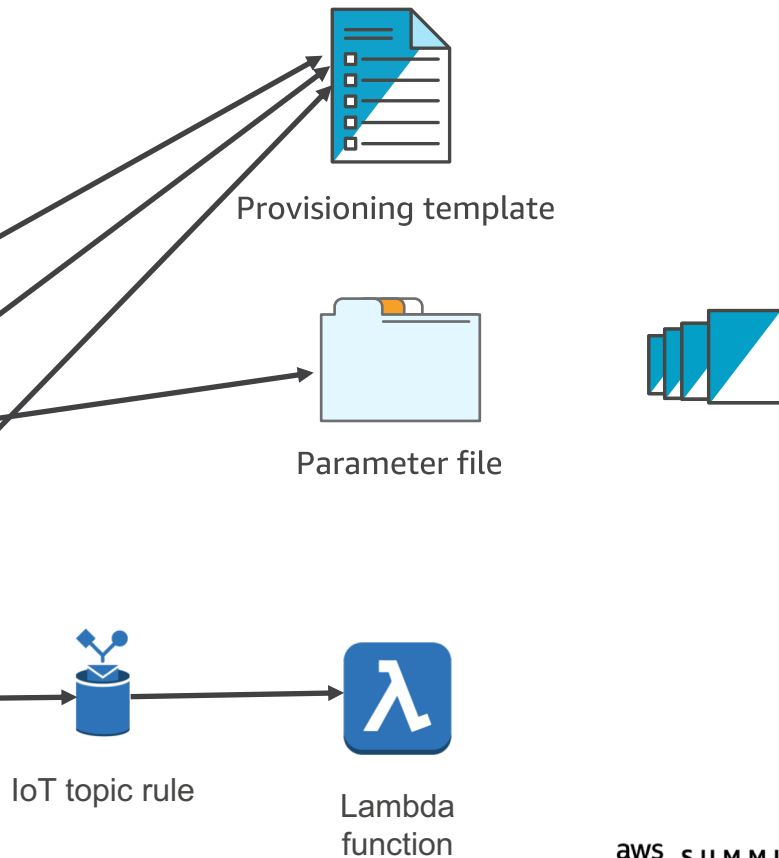
When a Device is Provisioned

- (Created in the device registry)
- Device certificate registered with AWS IoT Core
- (Certificate attached to the device)
- IoT Policy attached to the device through:
 - Certificate
 - Thing group



AWS IoT Provisioning Options

- API Calls
- Single Device Provisioning
- Bulk Device Provisioning
- Just-in-Time Provisioning
- Just-in-Time Registration



Device Onboarding – API

Using scripts with SDKs and call the API

- create-thing
- create-keys-and-certificate or register-certificate for BYOCA
- create-policy
- attach-principal-policy
- attach-thing-principal

Provisioning Template

```
"Parameters" : {  
  "ThingName" : {  
    "Type" : "String",  
    "Default" : "lightbulb",  
    "AllowedPattern" : "[a-zA-Z0-9-]*",  
    "MaxLength" : 128  
  },  
  "SerialNumber" : {  
    "Type" : "String",  
    "Default" : "v1",  
    "AllowedPattern" : "[a-zA-Z0-9-]*",  
    "MaxLength" : 128  
  },  
  "Location" : {  
    "Type" : "String",  
    "Default" : "WA",  
    "AllowedPattern" : "[a-zA-Z0-9-]*",  
    "MaxLength" : 128  
  },  
  "CSR" : {  
    "Type" : "String",  
    "Default" : "",  
    "AllowedPattern" : "[a-zA-Z0-9-]*",  
    "MaxLength" : 128  
  }  
},  
  
"Resources" : {  
  "thing" : {  
    "Type" : "AWS::IoT::Thing",  
    "Properties" : {  
      "ThingName" : {"Ref" : "ThingName"},  
      "AttributePayload" : {  
        "version" : "v1",  
        "serialNumber" : {"Ref" : "SerialNumber"}  
      },  
      "ThingTypeName" : "lightBulb-versionA",  
      "ThingGroups" : ["v1-lightbulbs", {"Ref" : "Location"}]  
    }  
  },  
  
  "certificate" : {  
    "Type" : "AWS::IoT::Certificate",  
    "Properties" : {  
      "CertificateSigningRequest": {"Ref" : "CSR"},  
      "Status" : "ACTIVE"  
    }  
  }  
}
```

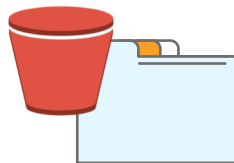

Single/Bulk Device Provisioning

```
{"ThingName": "foo", "SerialNumber": "123", "CSR": "csr1"}  
{"ThingName": "bar", "SerialNumber": "456", "CSR": "csr2"}
```

Parameters



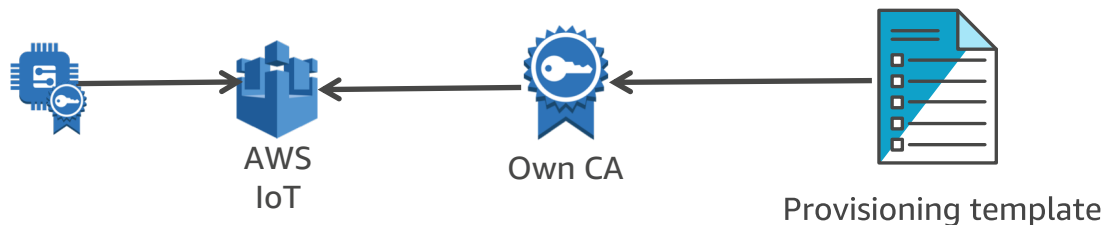
Provisioning template



Parameter file
on S3 Bucket

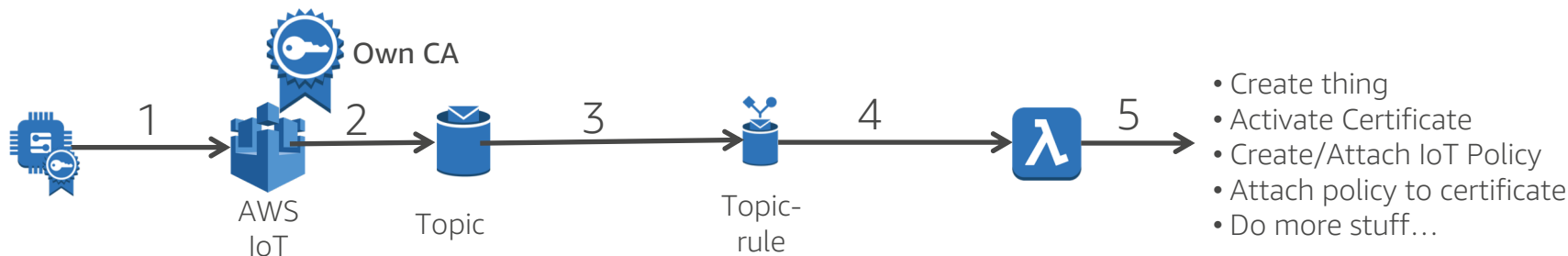
- Parameters with device information are used in the provisioning template
- Single: on "line" as parameter to register a thing
- Bulk: multiple parameter lines in an S3 bucket

Device Onboarding – JITP



- Own CA required
 - Provisioning Template attached to own CA
1. Device connects to AWS IoT, device certificate gets registered
 2. JITP provisions device according to the provisioning template

Device Onboarding – JITR



1. Device connects to AWS IoT, device certificate gets registered

2. AWS IoT publishes message to
`$aws/events/certificates/registered/<caCertificateID>`

3. Topic Rule is invoked

4. Topic Rule calls Lambda Function as action

5. Lambda provisions device

JITR vs. JITP

JITR	JITP
Topic rule and Lambda function. Code must be written and maintained	No code, only body template attached to CA
Provisioning more complex: Device connects, certificate registers with status PENDING_ACTIVATION, service sends MQTT message, rule triggers Lambda, Lambda does provisioning and optionally more stuff	Easy provisioning: Device connects, provisioning workflow run automatically
Flexible, different policies for different devices can be created/attached. Information from/to the provisioning process can be put/read from other systems, etc.	Static, same provisioning process for every device

Demo Time!

Please complete the session survey in
the summit mobile app.

