



AWS IoT Core

김성한

Sr. Solutions Architect
AWS

고객이 AWS IoT로 실현하는 가치



산업 공정의 성능 및 생산성 향상



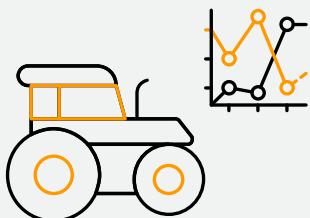
환자 건강 및 웰빙
애플리케이션을 원격으로
모니터링



재고 관리 및 창고 운영



가정, 건물 및 도시에서 더
스마트한 제품 및 사용자 경험
구축



더 큰 효율성으로 더 건강한
작물 재배



지속 가능한 에너지 자원을
보다 효율적으로 관리



커넥티드 및 자율주행
차량으로 운송 혁신



가정, 사무실, 공장 현장의 안전
강화

AWS IoT architecture



Analytics
Services

어떻게 하면 IoT 데이터를 이해하고 비즈니스 문제를 해결하기 위한 조치를 취할 수 있을까요?



Connectivity
and Control
Services

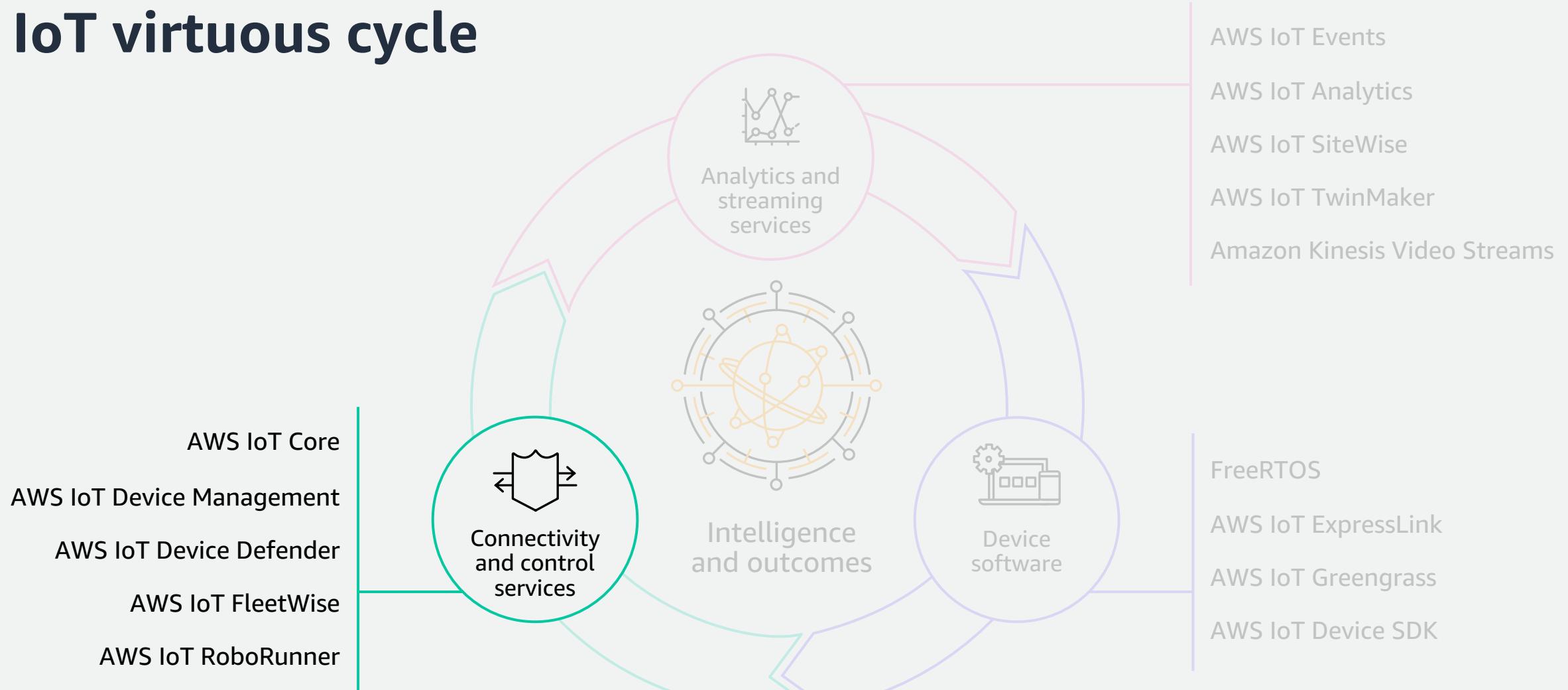
규모에 맞게 디바이스를 연결, 관리 및 보호하려면 어떻게 해야 합니까?



Device
Software

기본적으로 AWS와 함께 동작하는 엣지에서 동작하는 기기를 어떻게 만들 수 있나요?

IoT virtuous cycle

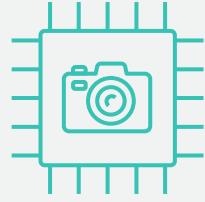


Common challenges of connected products



Turning data into value

IoT 운영 비용을 관리 및 절감하면서 혁신적인 서비스로 새로운 수익원 창출



Deploying connected devices

수백만 대의 장치를 안전하게 수집, 저장 및 유지 관리하기 위한 인프라 구축, 확장 및 유지 관리



Managing and scaling large fleets

장치 관리를 단순화하는 도구를 사용하여 전 세계적으로 수백만 대의 장치로 확장할 수 있는 기능

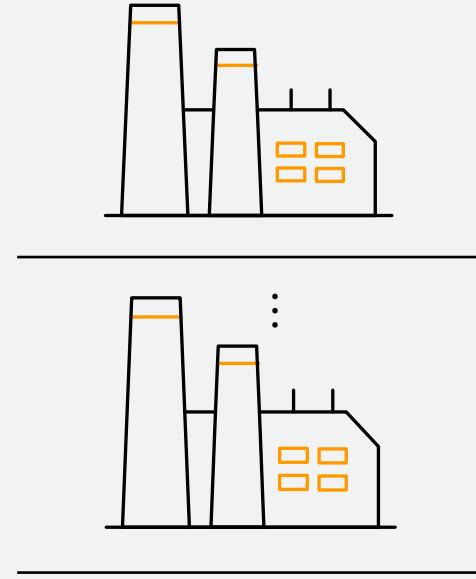


Managing security and privacy

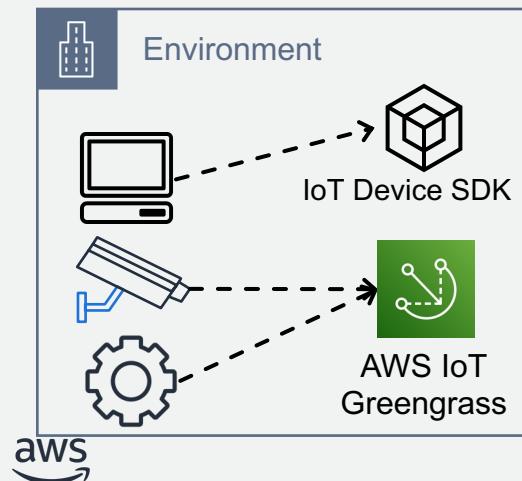
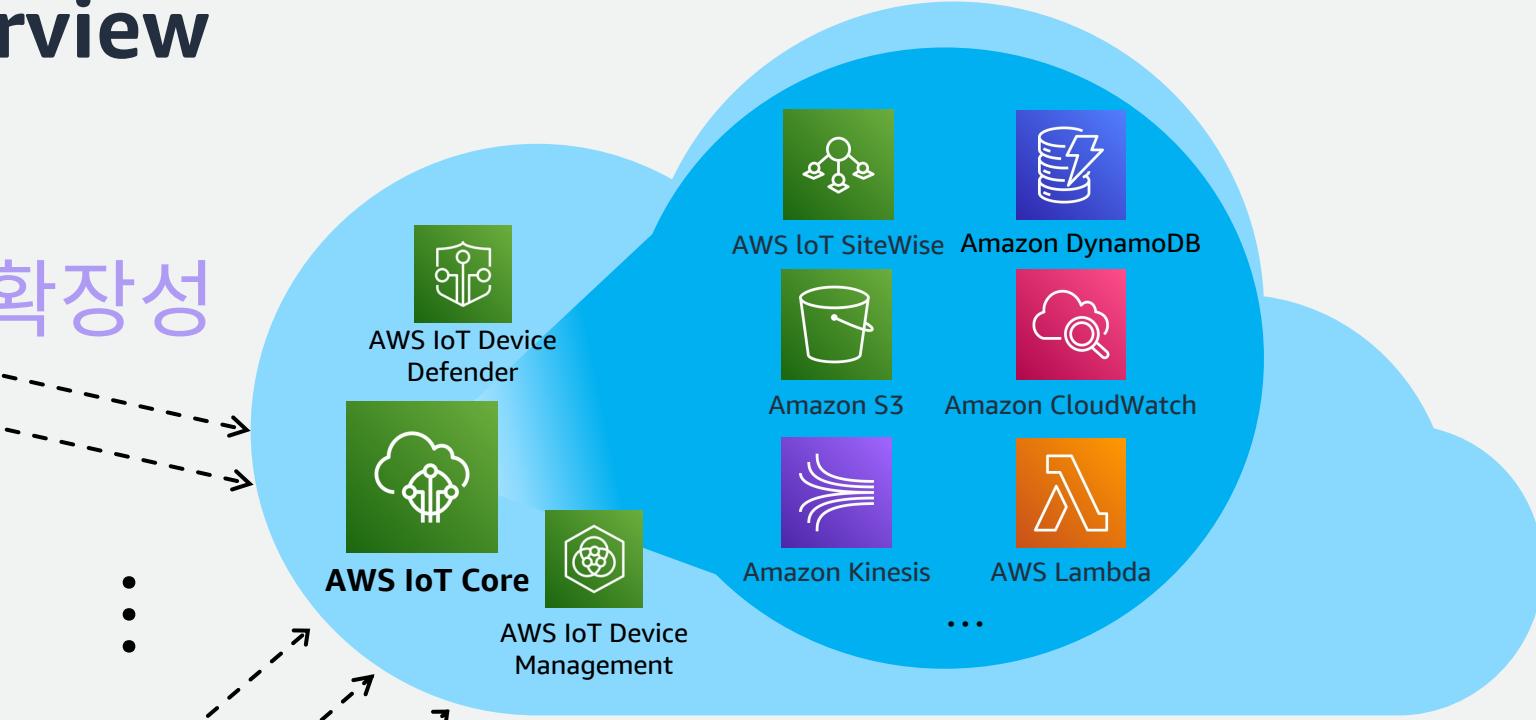
장치, IoT 데이터 및 고객 정보를 안전하게 유지하고 위협으로부터 보호하여 고객 문제를 최소화합니다.

다양한 AWS 서비스와 연결

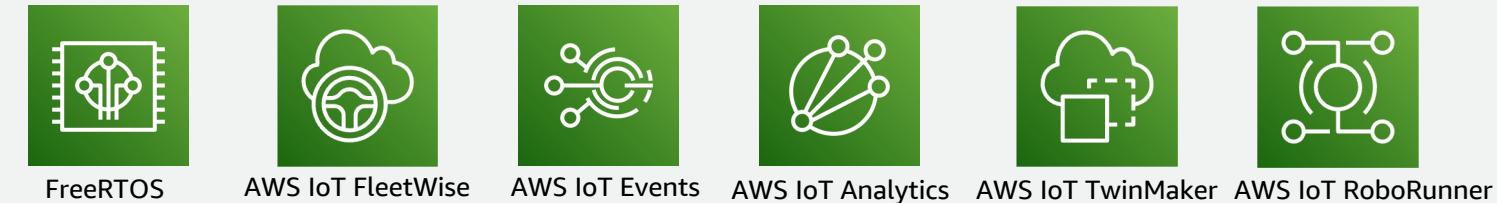
AWS IoT Overview



확장성



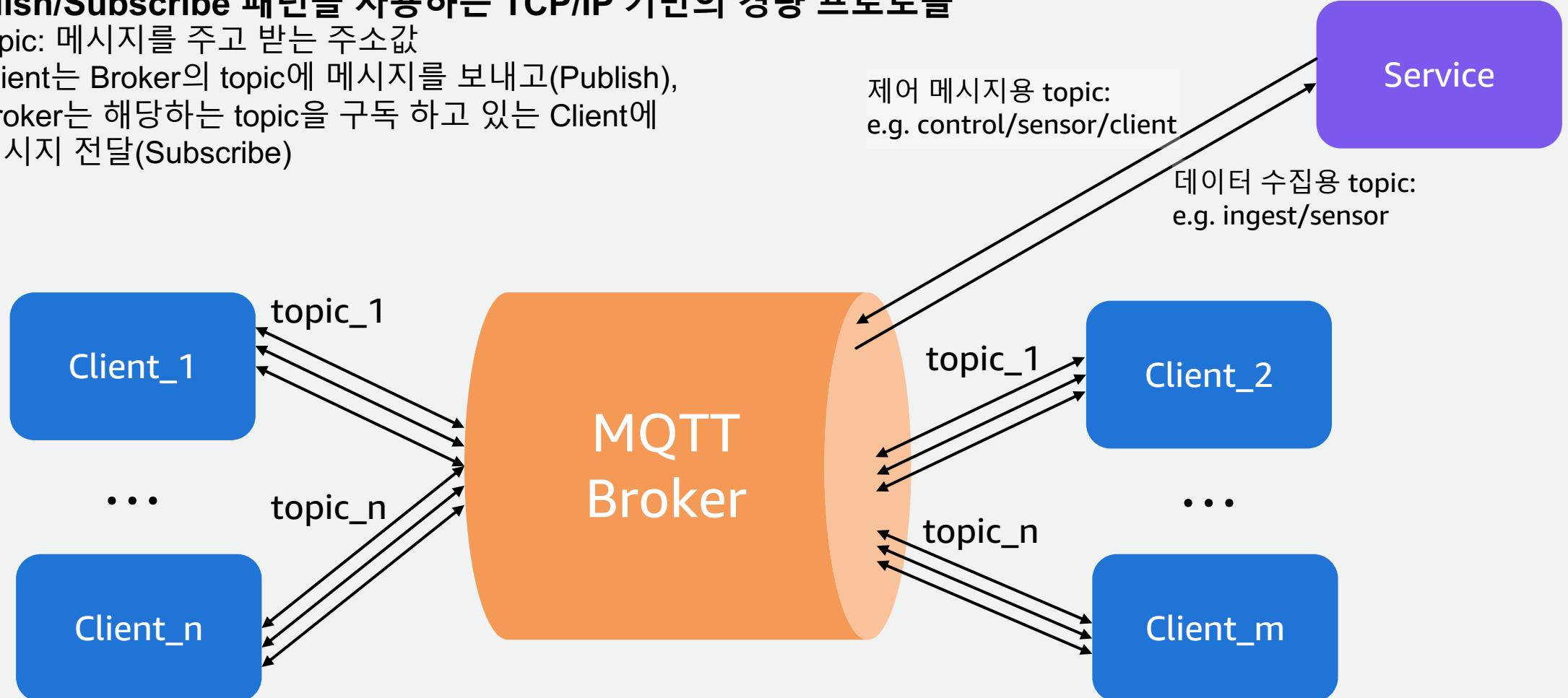
보안성



MQTT(Message Queuing Telemetry Transport)

Publish/Subscribe 패턴을 사용하는 TCP/IP 기반의 경량 프로토콜

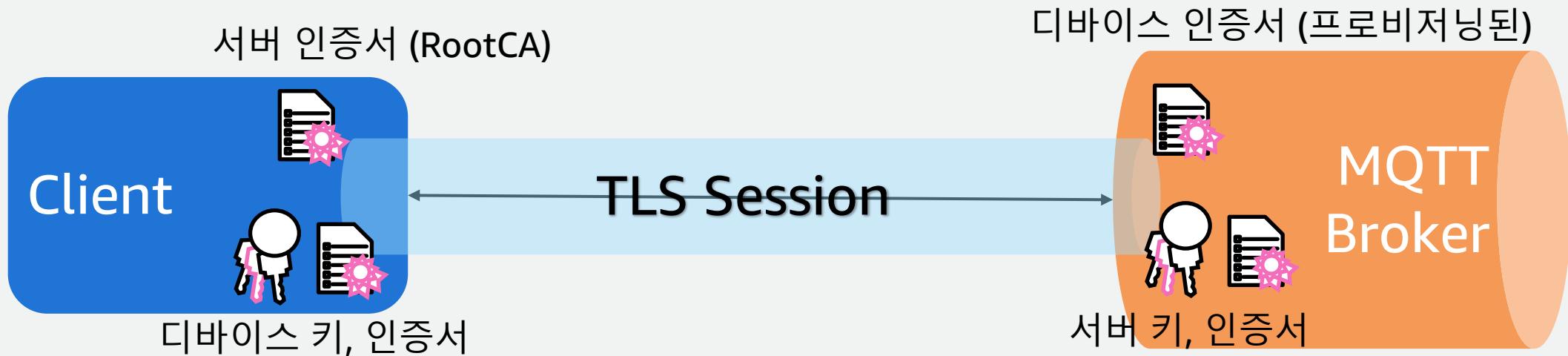
- topic: 메시지를 주고 받는 주소값
- Client는 Broker의 topic에 메시지를 보내고(Publish),
- Broker는 해당하는 topic을 구독 하고 있는 Client에 메시지 전달(Subscribe)



<https://aws.amazon.com/ko/what-is/mqtt/>

MQTT(Message Queuing Telemetry Transport) - 1

- TLS Handshake를 통한 TLS Session 구축
- TLS 상호 인증을 위한 인증서 프로비저닝 과정 필요
- 추가적인 Custom 인증 추가 가능 (Custom Authorizer)
 - TLS 서버 인증 + Custom 인증
 - TLS 상호 인증 + Custom 인증



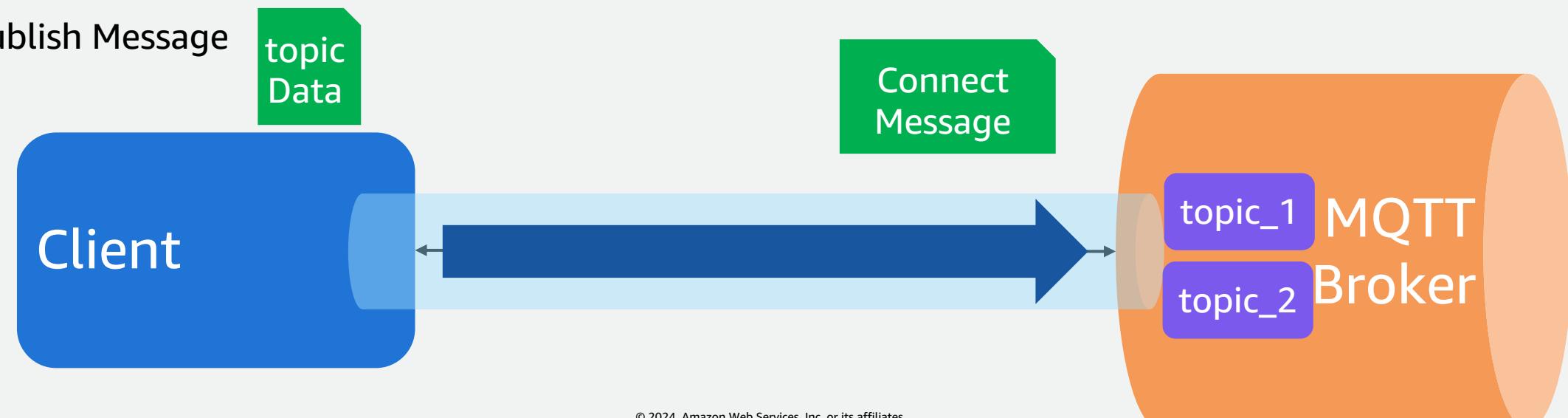
MQTT(Message Queuing Telemetry Transport) - 2

Connect

- Connect Message 전달로 실제 Client 연결 시작
- clean session 설정이 false이면서 QoS1일 때, 미수신 시 재전달 기능이 동작함
- LWT 메시지(유언장)은 이 때 Connect 메시지에 포함하여 전달

Publish

- Publish Message에 설정된 topic 정보로 포함된 데이터를 전달
- QoS 1일 경우, Broker로부터 PubAck Message를 받음
- QoS 1일 경우, Broker로부터 PubAck를 못받을 경우, 재전달 (중복 메시지 전달 가능성 있음)



<https://aws.amazon.com/ko/what-is/mqtt/>

MQTT(Message Queuing Telemetry Transport) - 3

Subscribe

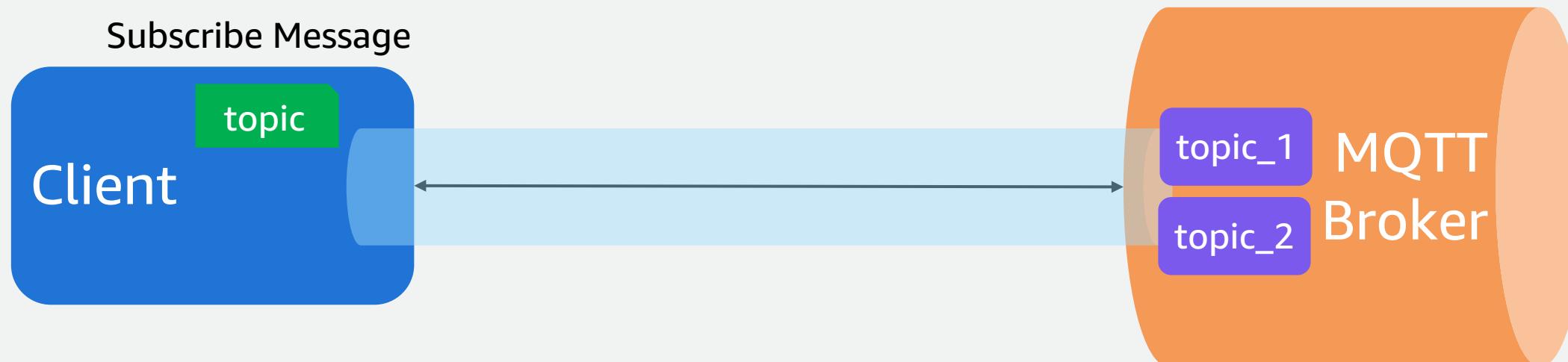
- Subscribe Message 설정된 topic 정보로 포함된 데이터를 broker로부터 전달받을 수 있도록 요청
- QoS 1일 경우, Broker로 SubAck Message를 전달
- QoS 1일 경우, Broker가 SubAck를 못받을 경우, Broker는 메시지를 재전달 (중복 메시지 전달 가능성 있음)
- 일반적으로 Connect 완료 후 다음 작업으로 진행



MQTT(Message Queuing Telemetry Transport) - 4

Retain

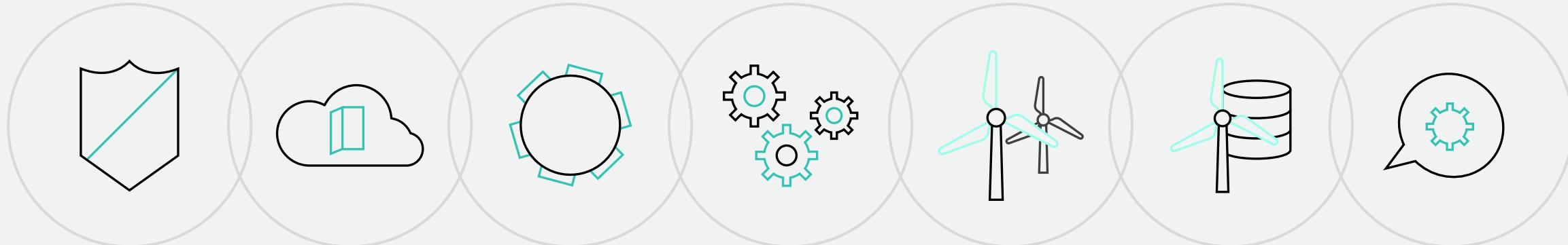
- Retain flag를 true로 하고 publish 된 메시지는 Broker에 해당 topic안에 메시지가 저장됨 (최신 데이터만)
- 해당 topic을 Subscribe할 때마다 데이터를 받아볼 수 있음
- MQTT3는 저장 기간이 따로 없고, MQTT5는 유효기간 있음 (최대 7일)
 - <https://docs.aws.amazon.com/iot/latest/developerguide/mqtt.html#mqtt5-message-expiry> 참고
- 별도의 권한 설정 필요
 - <https://docs.aws.amazon.com/iot/latest/developerguide/retained-message-policy-examples.html> 참고





AWS IoT Core

Connectivity and control services



Identity service

규모에 따른
디바이스 인증
관리 및 고유한
ID로 provision

Device gateway

IoT 워크로드에
최적화된 연결
관리

Message broker

IoT fleet 전체에서
안정적이고 빠른
통신

Rules engine

저렴한 비용으로
대량의 IoT
데이터를
수집하여 사전
처리하고 분석,
보고 및
시각화를 위해
20개 이상의
서비스에 제공

Device shadow

언제든지
디바이스 상태를
이해하고 제어

Registry

AWS 서비스에서
쉽게 사용할 수
있도록 디바이스
정의 및
카탈로그화

Device advisor

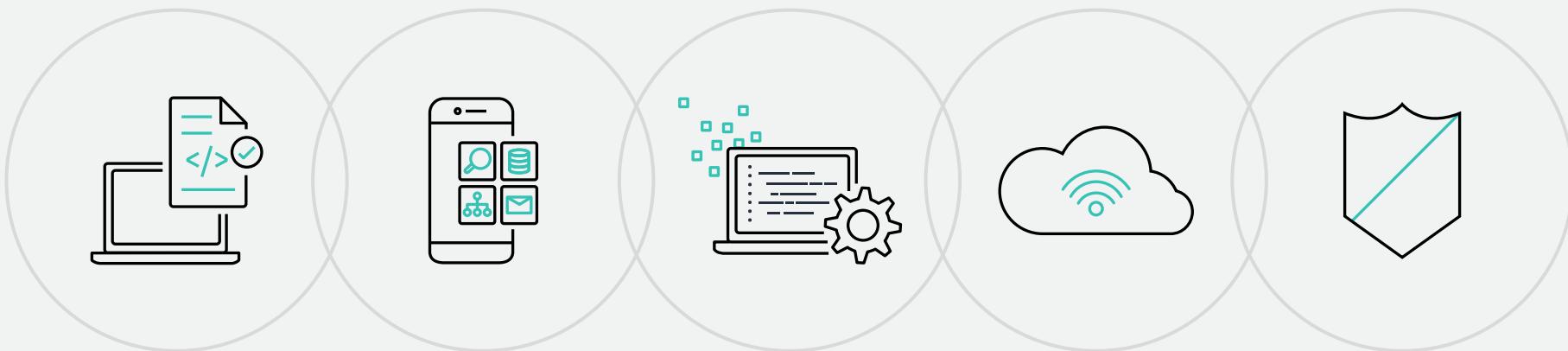
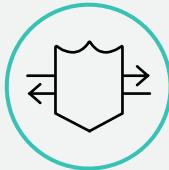
IoT 디바이스에
대한 완벽한 관리
테스트 기능



AWS IoT Core

AWS IoT Mobile SDK 및 AWS IoT Device SDK를 사용하면 디바이스를 AWS IoT Core에 연결하는 고유한 연결 기능을 구축할 수 있습니다.

Connectivity and control services



Abstract MQTT protocol

thing, topic, shadow, 및 job 등의 기능을 쉽게 활용

Mobile platform support

모바일 IoT 애플리케이션 구축을 위한 라이브러리는 iOS와 Android를 모두 지원합니다.

Common languages

JavaScript, Python, Java, Embedded C 및 C++를 활용한 프로그래밍 지원

Code flexibility

고객님 만의 고유한 연결 및 IoT 기능 구축

Security

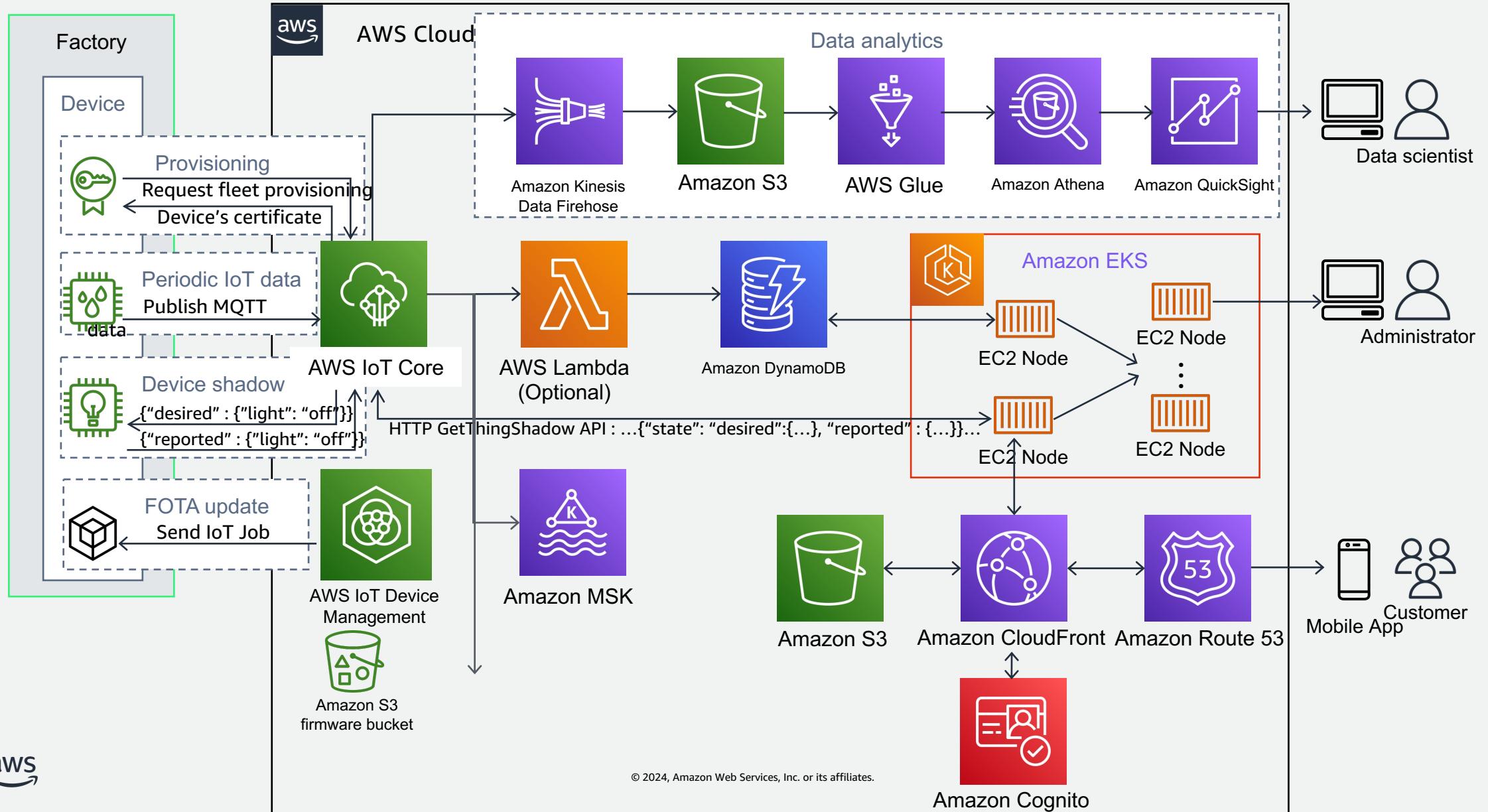
TLS 1.2를 사용한 End-to-end 클라우드 보안



IoT 디바이스 라이프사이클 관리: 주요 기능



Sample Architecture



Identity Service

규모에 따른 디바이스의 인증 관리 및
고유한 ID 제공

자체 root CA 및 client 인증서를 가져와서 사용하거나
AWS IoT Core가 사용자를 위한 인증서를 생성합니다

Fleet Provisioning을 사용하여
최초 연결 시 고유한 ID를 가진
대규모 디바이스를 자동 프로비저닝

Just-In-Time Registration, Just-In-Time
Provisioning을 활용한 자동 디바이스 등록

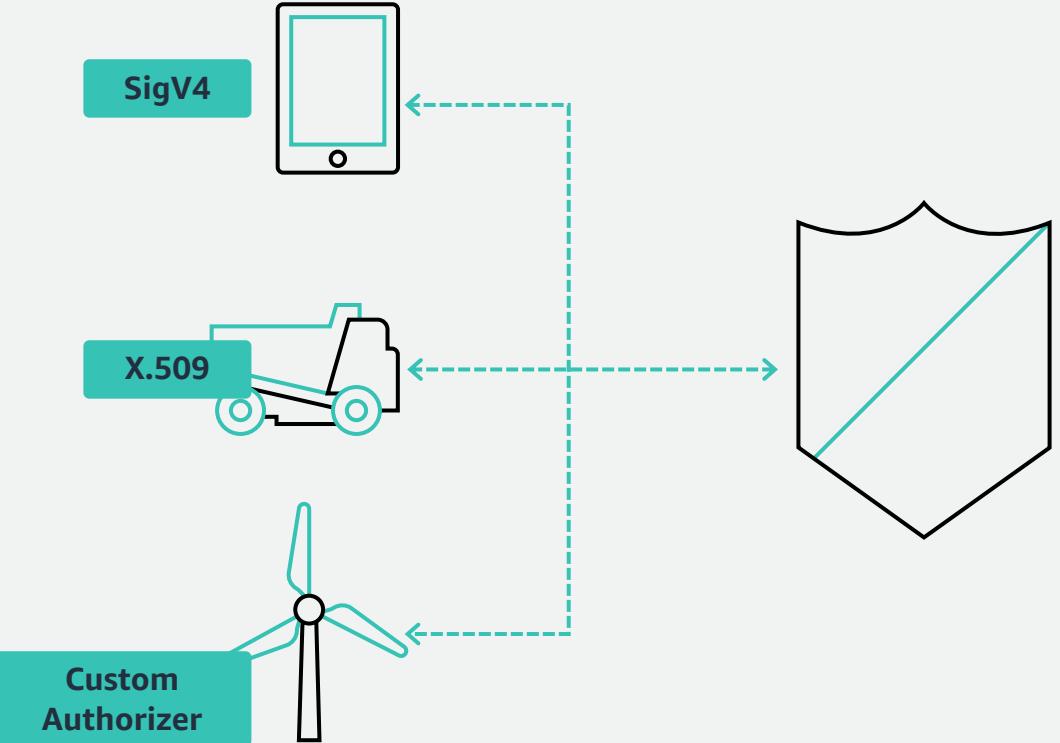
X.509, Cognito, SigV4 및
Lambda를 통한 Custom Authentication 지원

IoT Policy를 통한 유연하고 세분화된 액세스 제어

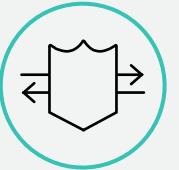
IoT Policy는 identities 또는 레지스트리 항목과 연결할 수
있습니다

MQTT topic 수준까지 액세스를 제어할 수 있습니다.

Connectivity and
control services



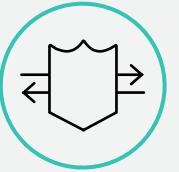
<https://docs.aws.amazon.com/iot/latest/developerguide/authentication.html>



상호 TLS 인증

- AWS IoT에서 지원하는 디바이스용 인증 방법
 - 인증서를 이용한 서버와 클라이언트 간의 양방향 인증 진행
- 디바이스에 x.509 인증서와 Private Key를 보관





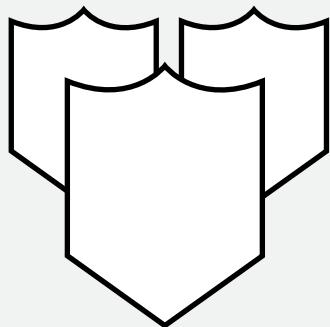
Device authentication and authorization

인증(authentication): 사용자(device) 신원 확인(식별) 절차

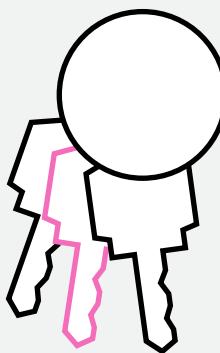
권한 부여(authorization): 특정 리소스에 대한 권한을 허용하는 과정

디바이스 인증서는 디바이스의 정보인 Thing과 IoT Policy에 연결

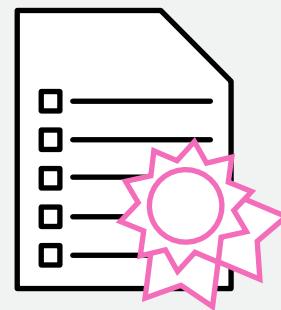
- 어떤 디바이스를 인증할 것인가? (Thing)
- 어떤 권한을 부여할 것인가? (IoT Policy)



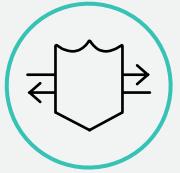
IoT thing



IoT certificate



IoT policy



IoT Policy

iot:ClientId: AWS IoT Core 메시지 브로커에 연결하는 데 사용되는 클라이언트 ID

aws:SourceIp: AWS IoT Core 메시지 브로커에 연결된 클라이언트의 IP 주소

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        { "Effect": "Allow",  
          "Action": ["iot:Connect"],  
          "Resource": [ "arn:aws:iot:us-east-1:123451234510:client/${iot:ClientId}" ]  
        },  
        { "Effect": "Allow",  
          "Action": ["iot:Publish"],  
          "Resource": [ "arn:aws:iot:us-east-1:123451234510:topic/my/topic/${iot:ClientId}" ]  
        }  
    ]  
}
```

Thing policy variables

디바이스가 MQTT 또는 WebSocket 프로토콜을 통한 MQTT를 통해 연결하는 경우에만 사용할 수 있는 policy variable 들

iot:Connection.Thing.ThingName

iot:Connection.Thing.ThingTypeName

iot:Connection.Thing.Attributes[**attributeName**]

iot:Connection.Thing.IsAttached: ["**true**"]: AWS IoT에 등록되고 보안 주체에 연결된 디바이스만 정책 내의 권한에 액세스할 수 있도록 함.

IoT Thing에 연결되지 않은 인증서를 제공하는 경우 AWS IoT Core에 연결하는 것을 방지

https://docs.aws.amazon.com/ko_kr/iot/latest/developerguide/thing-policy-variables.html



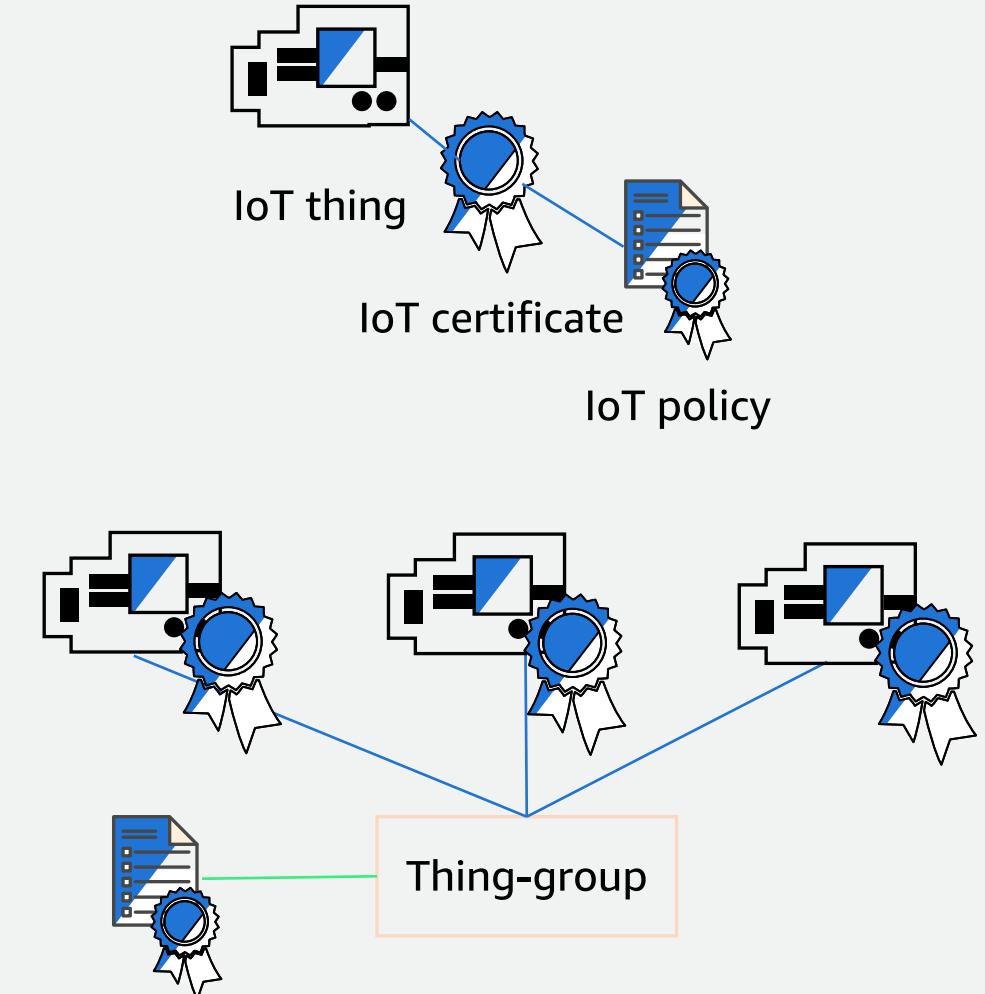
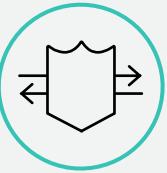
When a Device is Onboarded

- 디바이스 등록 (Thing)
- Device 인증서, private key 생성
- 디바이스에 인증서 적용
- 디바이스에 IoT Policy 적용
(또는 Thing Group)



Provisioning template

Connectivity and control services



Provisioning Template

- IoT Thing resources
- IoT Policy resources
- IoT Certificate resources
- Parameters

Provisioning API 요청

<https://docs.aws.amazon.com/iot/latest/developerguide/provision-template.html>

```
{  
  "Parameters": {  
    "SerialNumber": {  
      "Type": "String"  
    },  
    "AWS::IoT::Certificate::Id": {  
      "Type": "String"  
    }  
  },  
  "Resources": {  
    "certificate": {  
      "Properties": {  
        "CertificateId": {  
          "Ref": "AWS::IoT::Certificate::Id"  
        },  
        "Status": "Active"  
      },  
      "Type": "AWS::IoT::Certificate"  
    },  
    "policy": {  
      "Properties": {  
        "PolicyName": "EA1A0CDFCEDEAEE65D65B7AB012F0D7D89208F1EDA825C8"  
      },  
      "Type": "AWS::IoT::Policy"  
    },  
    "thing": {  
      "OverrideSettings": {  
        "AttributePayload": "MERGE",  
        "ThingGroups": "DO_NOTHING",  
        "ThingTypeName": "REPLACE"  
      },  
      "Properties": {  
        "AttributePayload": {},  
        "ThingGroups": [  
          "thinggroup",  
          "bulk-group"  
        ],  
        "ThingName": {  
          "Fn::Join": [  
            "",  
            [  
              "thing-",  
              {  
                "Ref": "SerialNumber"  
              }  
            ]  
          ]  
        }  
      },  
      "Type": "AWS::IoT::Thing"  
    }  
  }  
}
```

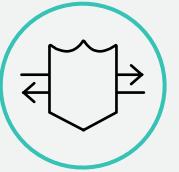
parameters



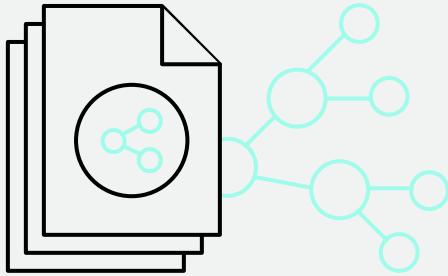
certificate

policy

Thing

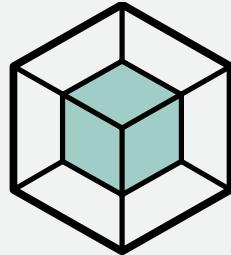


Fleet Provisioning—구성 요소



Provisioning template

고객이 초기 신뢰 기준(initial-trust criteria)을 정의할 수 있습니다.
구성 필요



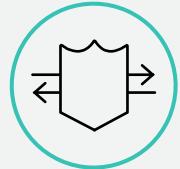
“Updated” IoT Device SDK

디바이스가 고유한 디바이스
구성을 가져올 수 있도록 합니다.



Mobile SDK

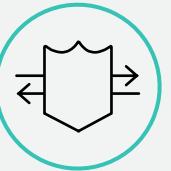
companion app이 고유한
디바이스 구성을 가져올 수
있습니다.



Device provisioning options

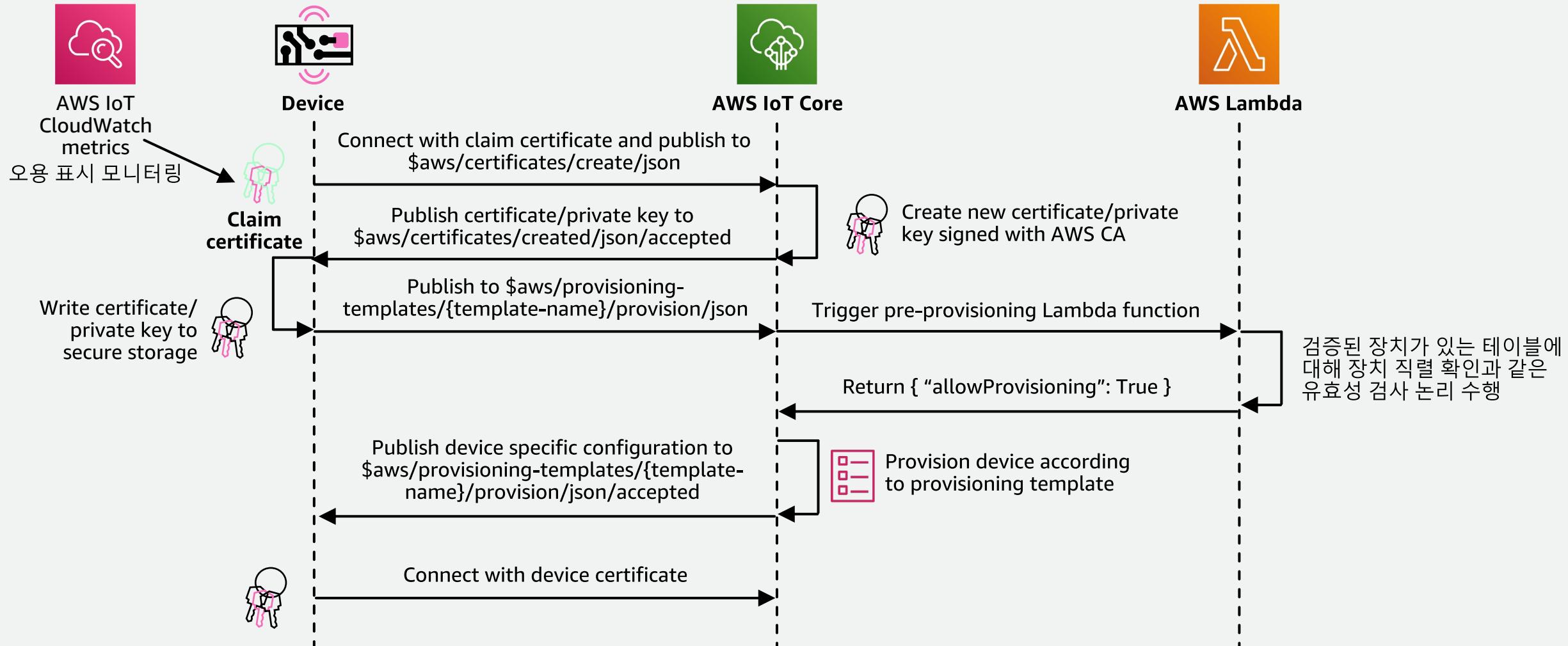
- Single thing provisioning
- Just-in-time provisioning (JITP)
- Fleet provisioning
 - By trusted user
 - By claim

Provisioning
Template 사용하여
대량 디바이스 등록
자동화 가능



Connectivity and
control services

Fleet provisioning by claim





IoT Greengrass device를 fleet provisioning하기

설치 작업

- Greengrass core software 다운로드
- Greengrass fleet provisioning plugin 다운로드
- Greengrass fleet provisioning plugin config 설정
 - 사전 확인한 Device의 region & endpoints
 - Claim 인증서 & private key 위치
 - Provisioning template
 - Parameter 정보
- Greengrass core software 실행

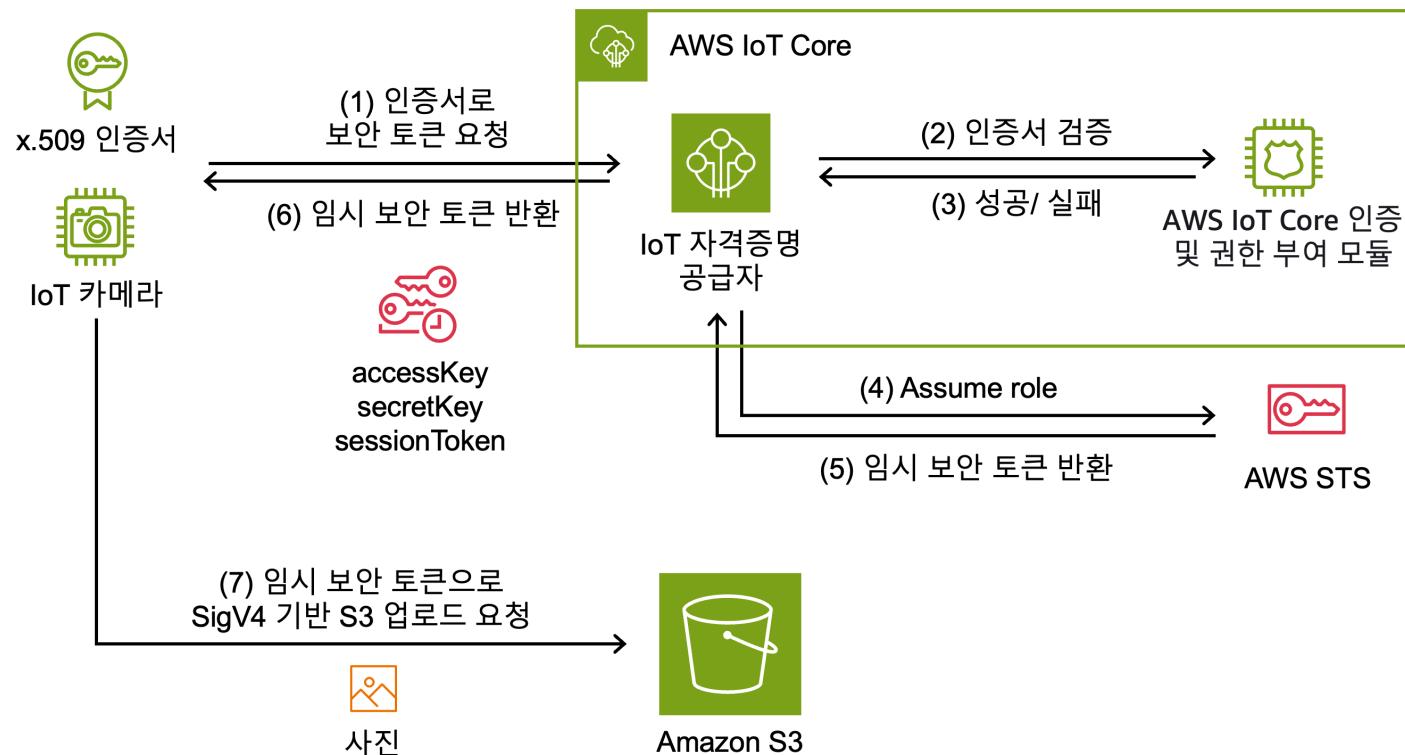
claim certificate

```
---  
services:  
  aws.greengrass.Nucleus:  
    version: "2.6.0"  
  aws.greengrass.FleetProvisioningByClaim:  
    configuration:  
      rootPath: "/greengrass/v2"  
      awsRegion: "us-west-2" Device의 region & endpoints  
      iotDataEndpoint: "device-data-prefix-ats.iot.us-west-2.amazonaws.com"  
      iotCredentialEndpoint: "device-credentials-prefix.credentials.iot.amazonaws.com"  
      iotRoleAlias: "GreengrassCoreTokenExchangeRoleAlias"  
      provisioningTemplate: "GreengrassFleetProvisioningTemplate"  
      claimCertificatePath: "/greengrass/v2/claim-certs/claim.pem.crt"  
      claimCertificatePrivateKeyPath: "/greengrass/v2/claim-certs/claim-privatekey.pem"  
      rootCaPath: "/greengrass/v2/AmazonRootCA1.pem"  
      templateParameters:  
        ThingName: "MyGreengrassCore"  
        ThingGroupName: "MyGreengrassCoreGroup"
```

parameters

Credential Provider

AWS IoT Core Credential Provider를 통해 고객은 최대 12시간 동안 유효한 제한된 권한의 임시 보안 토큰을 요청하고 (access key ID / secret access key 대신에) 해당 토큰을 사용하여 AWS request에 서명하고 인증할 수 있습니다.

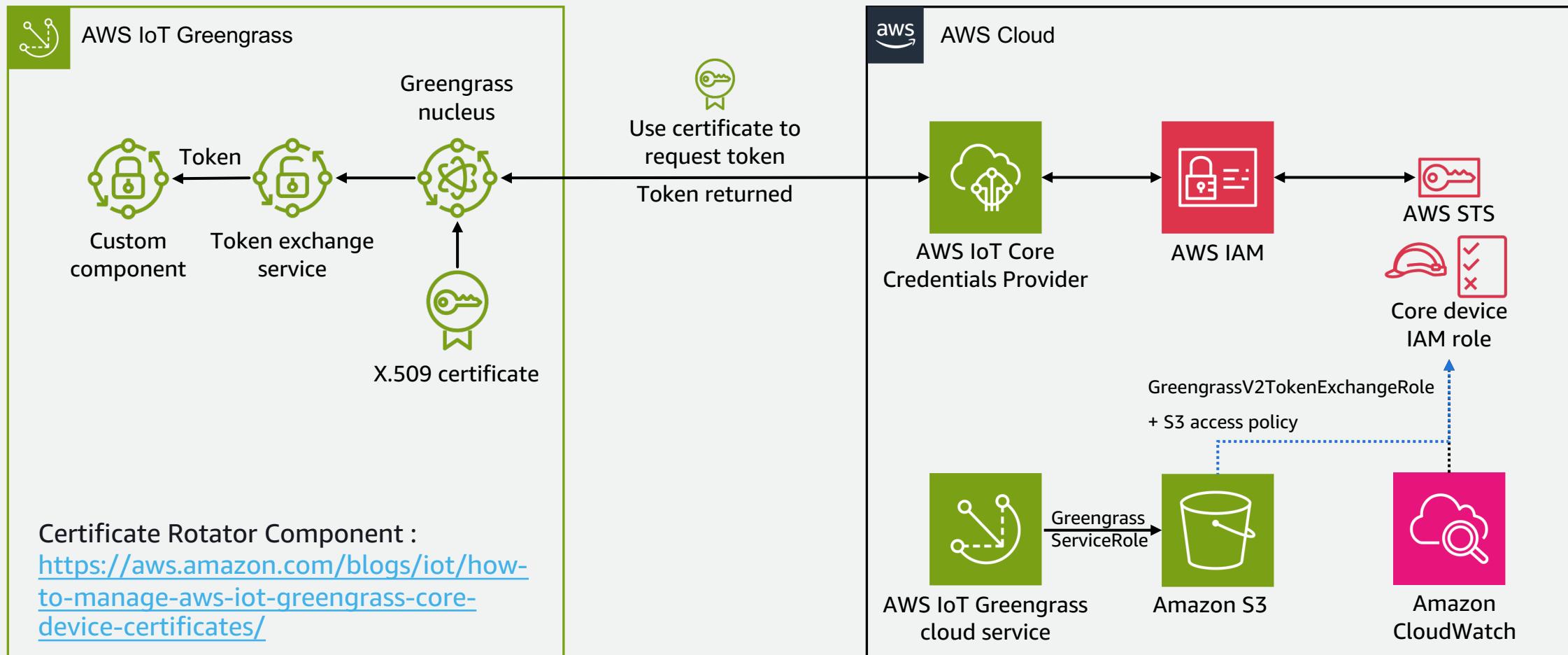


<https://aws.amazon.com/ko/blogs/security/how-to-eliminate-the-need-for-hardcoded-aws-credentials-in-devices-by-using-the-aws-iot-credentials-provider/>



<https://aws.amazon.com/ko/blogs/tech/uploading-files-to-s3-through-aws-iot-certificate-from-iot-devices/>

Identity and Access Management - Credential Provider



Custom Authorizer

매 인증 시마다 Lambda function 호출 및 인증 체크

토큰 기반 인증 형태 지원 (Lambda 비용 과금)

Policy 정보를 Lambda function 내에 정의해서 리턴

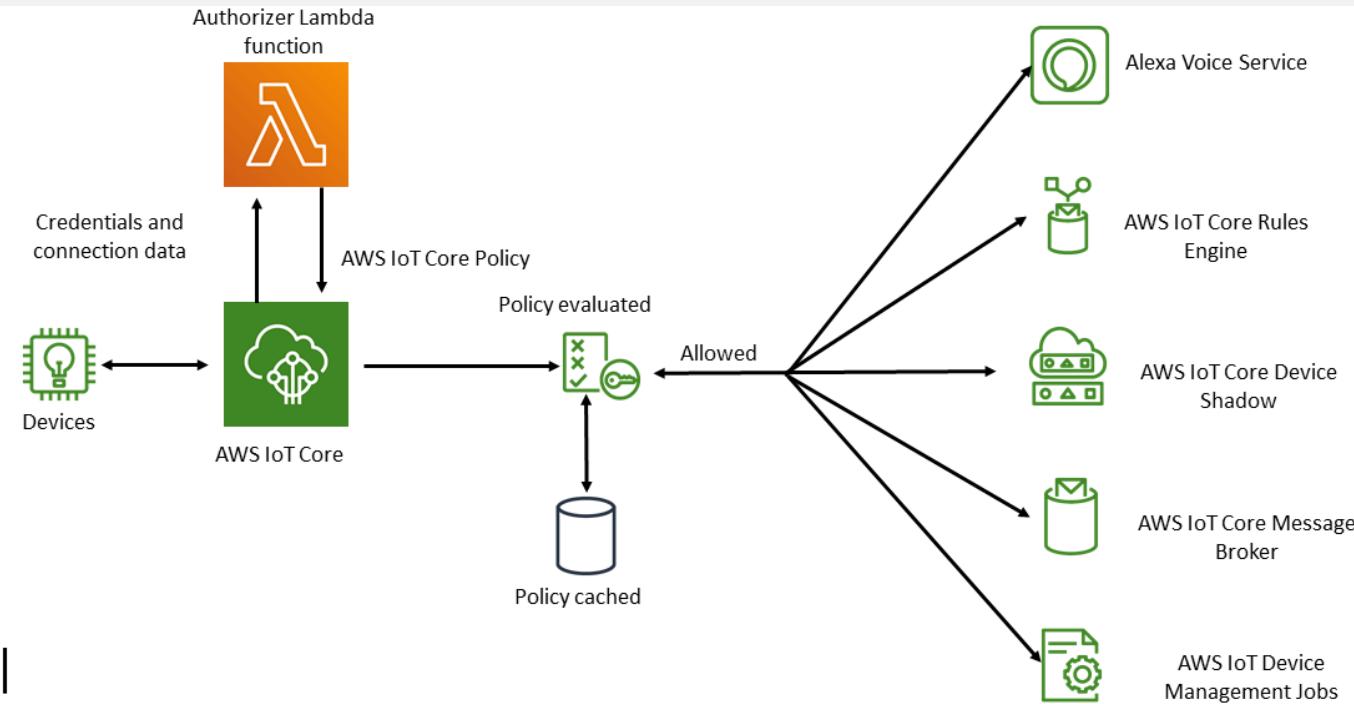
Credential 정보

MQTT 프로토콜 - 연결 요청할 때와

Refresh 시간마다 호출

HTTP 프로토콜 - Client의 Keep Alive 시간 및

HTTP Caching 기능을 적용하면 최대 5분 정도 유지



Device Gateway

IoT 워크로드에 최적화된 연결 관리

MQTT, WebSockets 및 HTTP를 포함하여 여러 프로토콜 지원

TLS 1.2를 통한 보안 통신

리소스가 제한된 디바이스에 최적화

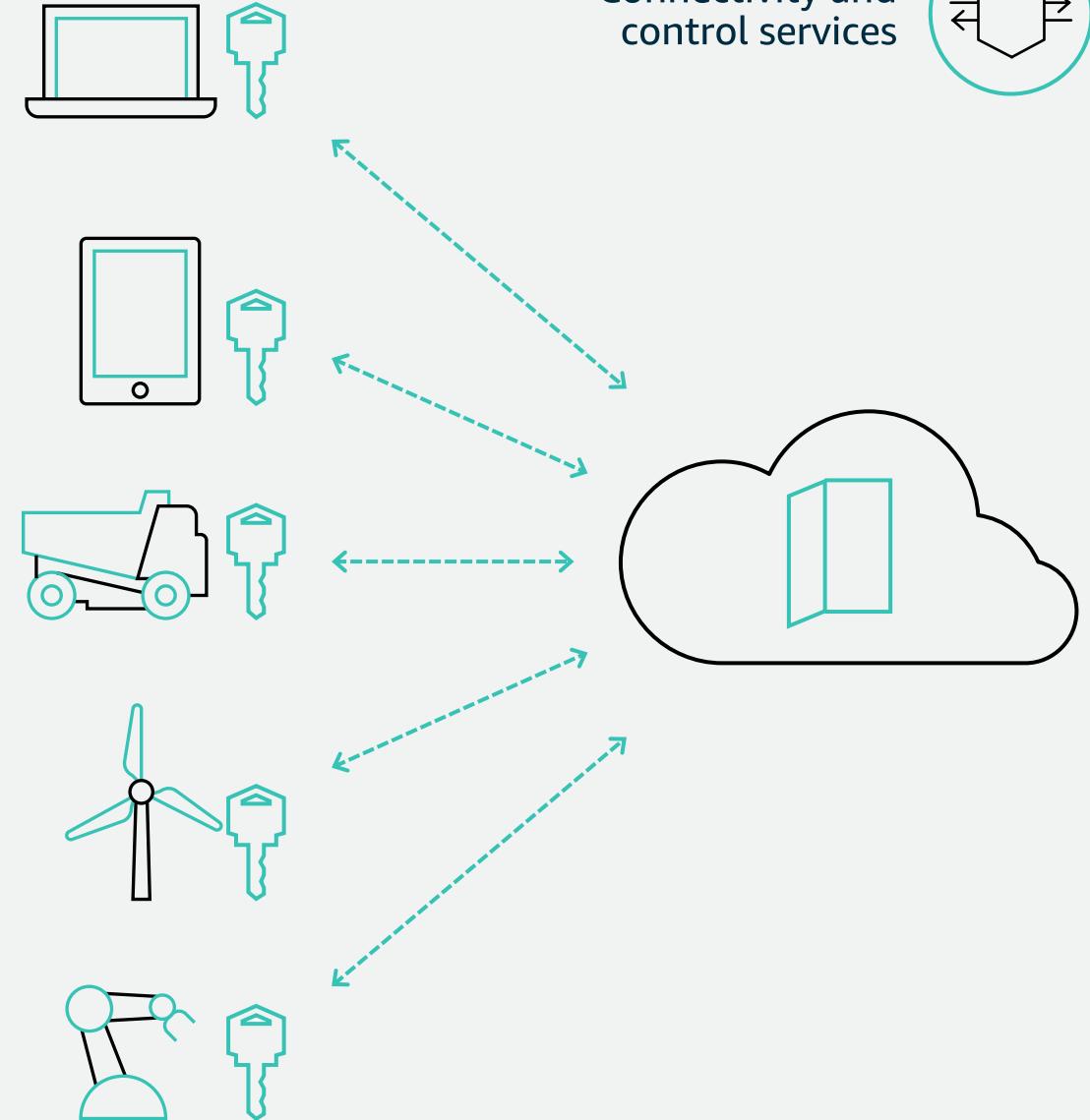
ECC 키교환 알고리즘 및 인증서 지원

Endpoints 설정

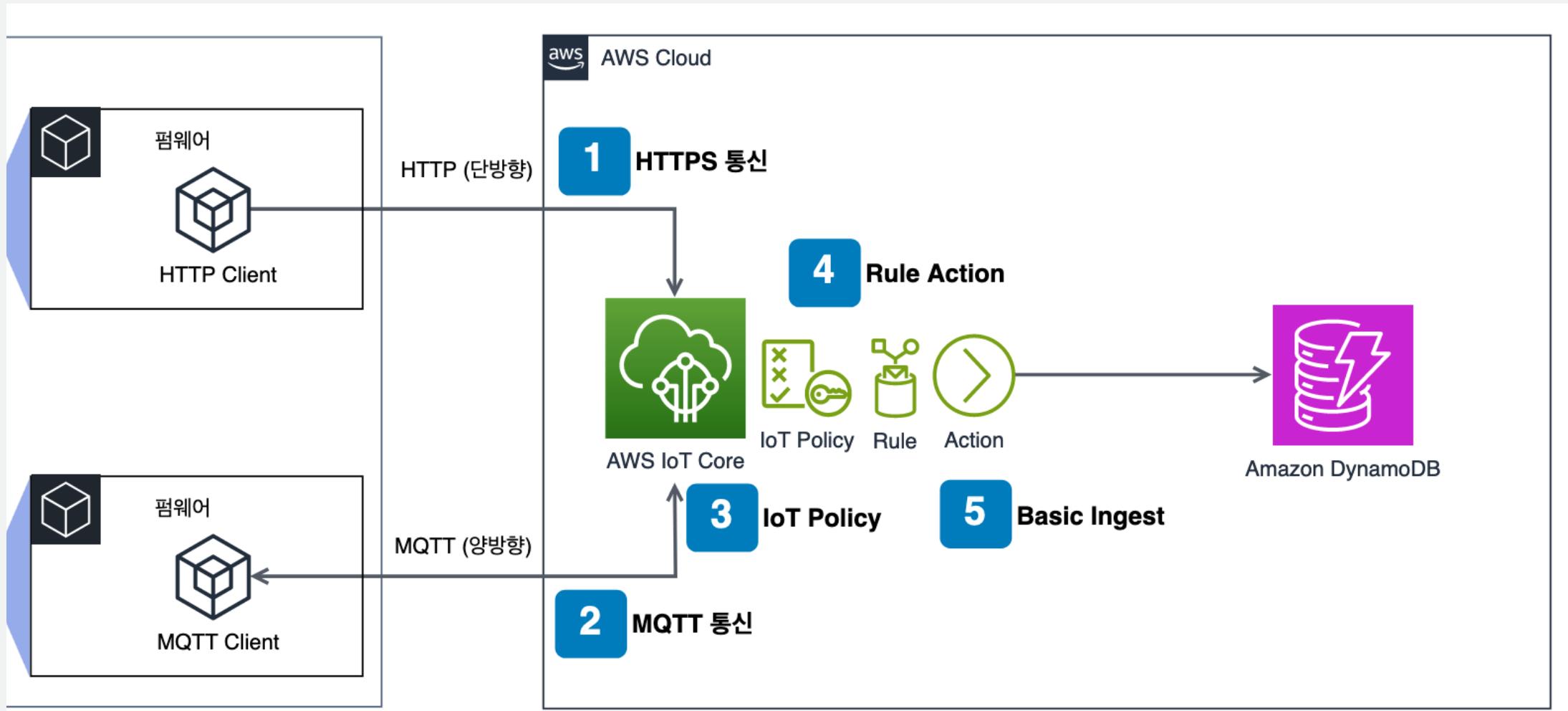
Custom domain과 서버 인증서 적용

Custom Authorizers 등록

Connectivity and control services



IoT Core 프로토콜 지원 예시



MQTT vs HTTP

- AWS IoT Core는 MQTT, WebSocket 및 HTTPS를 포함한 여러 프로토콜을 지원
 - MQTT: 연결 지향적이며, 일대다 혹은 다대다 통신, 그리고 양방향 통신에 적합하며, 연결에 대한 비용이 발생함.
 - HTTPS: 비연결 지향적이며, 단방향 메세지 전송에 적합하며 연결에 대한 비용이 없음
- 시나리오에 따라 MQTT 또는 HTTPS를 선택해서 혼용하는 것을 권장
 - MQTT: 자주 데이터를 보내거나 데이터 전송 간격이 짧고, 특히 디바이스 제어 시나리오에 적합
 - HTTPS: 간헐적으로 데이터를 보내거나 데이터 전송 간격이 길고, 특히 디바이스에서 데이터 수집 시나리오에 적합

	MQTT	HTTPS
연결 지향성	연결 지향	비연결 지향
메시지 송수신 방향	양방향	단방향
메시지 헤더 크기	작은 편	큰 편
권장 시나리오	데이터 수집 및 제어	간헐적인 데이터 수집



Message Broker

언제든지 디바이스의 상태를 이해하고 제어

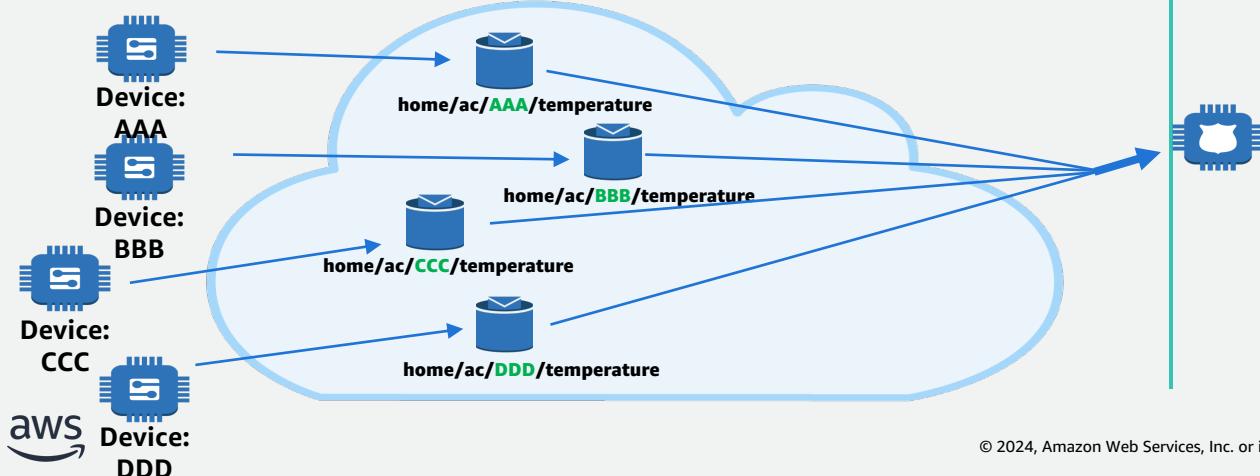
디바이스와 애플리케이션 간 양방향 메시지 스트리밍

오프라인 디바이스를 위하여 메시지 보존

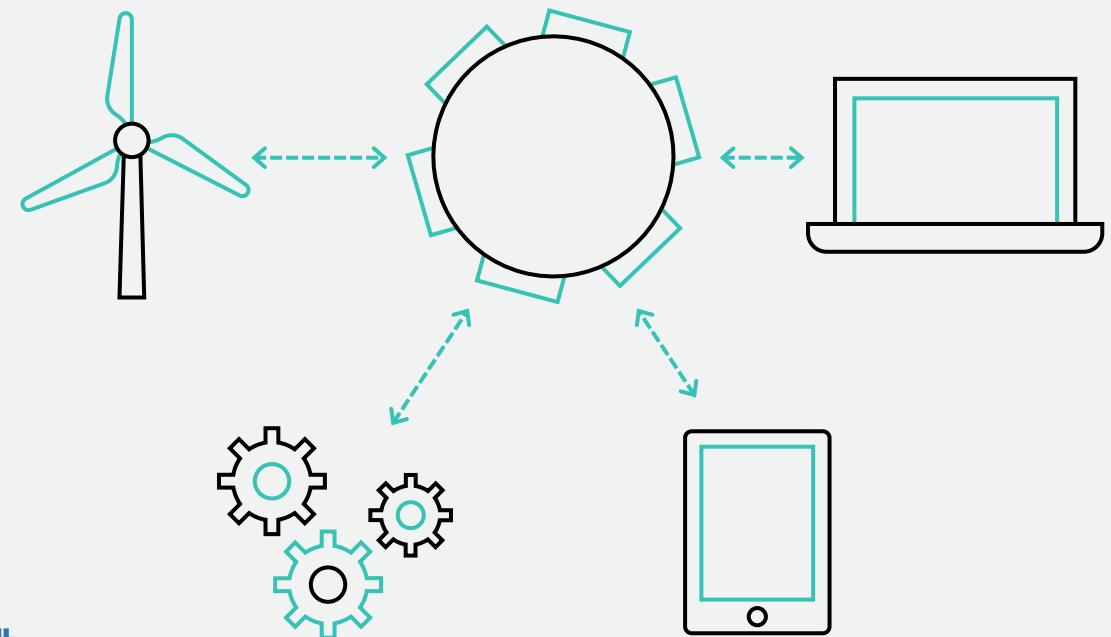
독립된 디바이스 및 애플리케이션을 위한
Publish/Subscribe 지원

QoS0 및 QoS1 messaging 지원

topic filter에 wildcard를 지원하는 사용자 지정 topic



Connectivity and control services



MQTT5

간편한 구현과 향상된 보안으로 MQTT5를 지원합니다

명령 및 제어 향상을 위한 Request/Response

규칙 엔진을 사용한 간편한 메시지 처리

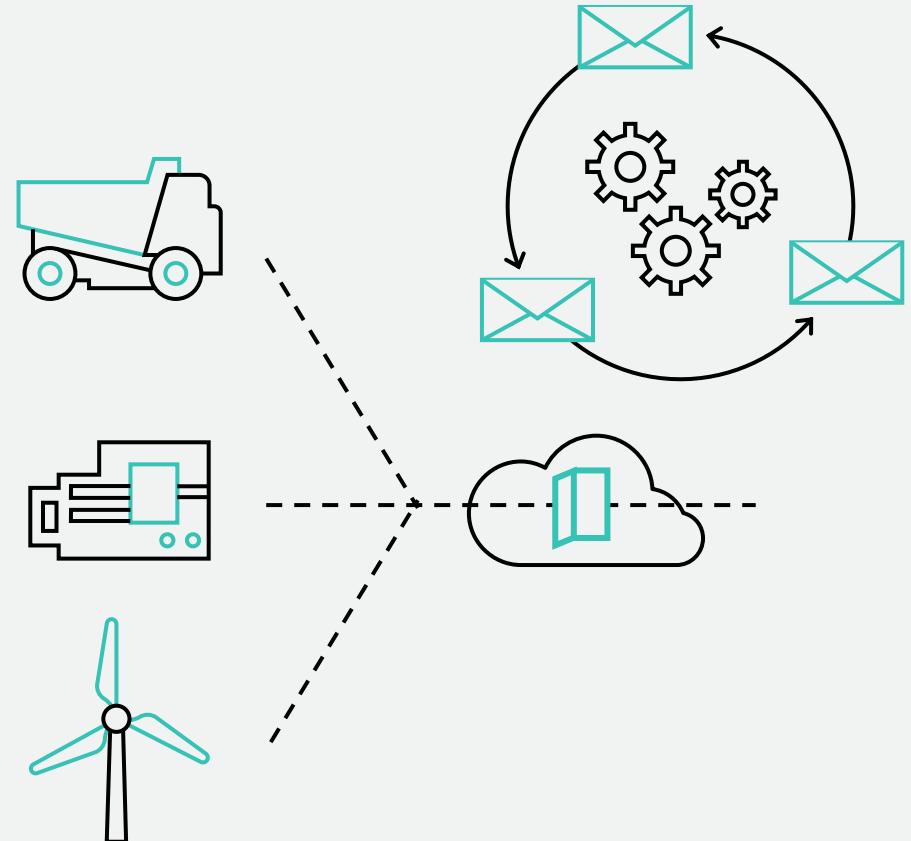
Topic Alias를 사용하여 대역폭 사용 감소

Reason Codes로 오류 처리 개선

Message/Session expiry timeouts 정의

Shared Subscriptions 사용하여 여러 구독자에게 메시지 로드 밸런싱

Connectivity and control services



Rules Engine

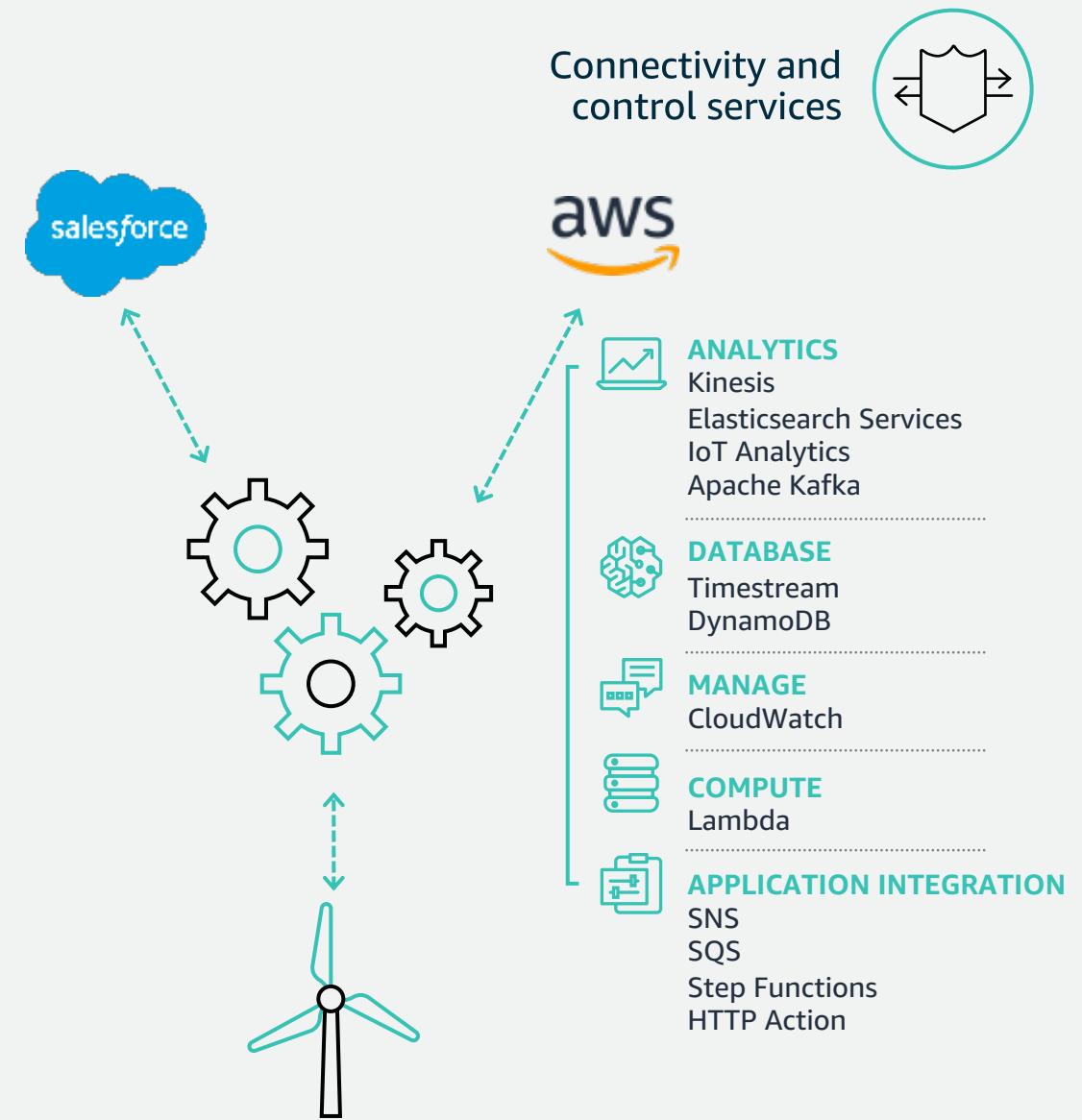
저렴한 비용으로 대량의 데이터를 수집하여 사전 처리한 후 분석, 보고 및 시각화를 위해 20개 이상의 서비스에서 사용할 수 있습니다.

Transform—수학, 문자열 조작, 날짜 등을 위한 기능이 내장되어 있습니다.

Filter—WHERE 절을 사용하여 원하는 데이터만 캡처합니다.

Enrich—Device Shadow 및 Amazon Machine Learning에서 또는 인라인 AWS Lambda 실행 및 DynamoDB 조회를 통해 외부 소스로부터 컨텍스트를 가져옵니다.

Route—데이터를 20개 이상의 AWS 서비스와 Salesforce와 같은 타사 서비스에 전송합니다.



Basic Ingest

Basic Ingest를 통한 메시징 비용 절감

Basic Ingest를 사용하면 AWS IoT 규칙 작업에서 지원하는 AWS 서비스로 디바이스 데이터를 안전하게 전송할 수 있으며, 이때 메시징 요금이 발생하지 않음

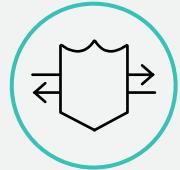
Basic Ingest 또는 AIA(Alexa Voice Service for AWS IoT Core) 항목을 사용하여 송신하거나 수신하는 메시지에 대해서는 메시지 요금이 발생하지 않음

Basic Ingest는 수집 경로에서 게시/구독 메시지 브로커를 제거해 데이터 흐름을 최적화

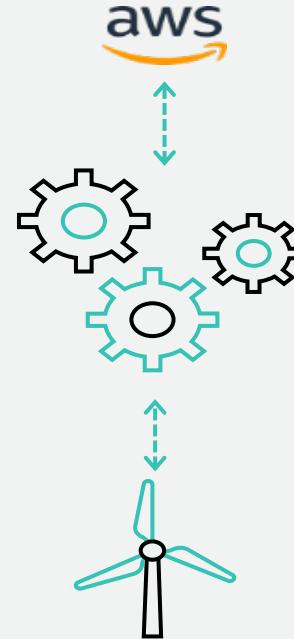
\$aws/rules/<rule-name>/<optional-customer-defined-segments> 형식으로 사용
(여기서 rule-name은 trigger할 AWS IoT 규칙의 이름)

예시) \$aws/rules/ruleName/topicName

Connectivity and control services

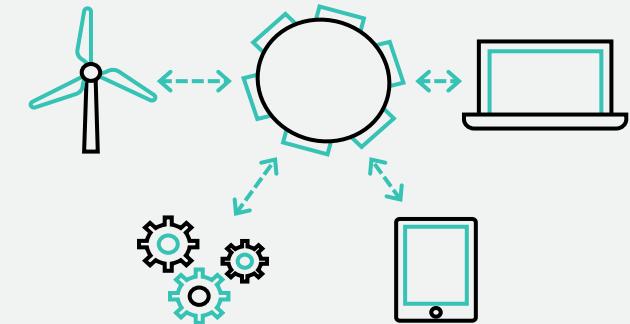


Basic Ingest (Rule Engine)



트리거된 rule 100만 개당/
실행된 action 100만 개당
\$0.18

Message Broker



메시지 100만 개당 \$1.20



Device Shadow

언제든지 디바이스의 상태를 이해하고 제어

디바이스의 마지막 상태를 보고합니다. 예를 들어 마지막으로 확인된 전구의 색상은 파란색입니다

디바이스의 상태 변경(예: 전구의 색상을 빨간색으로 변경)

MQTT를 통하여 상태 변경을 실시간으로 알림

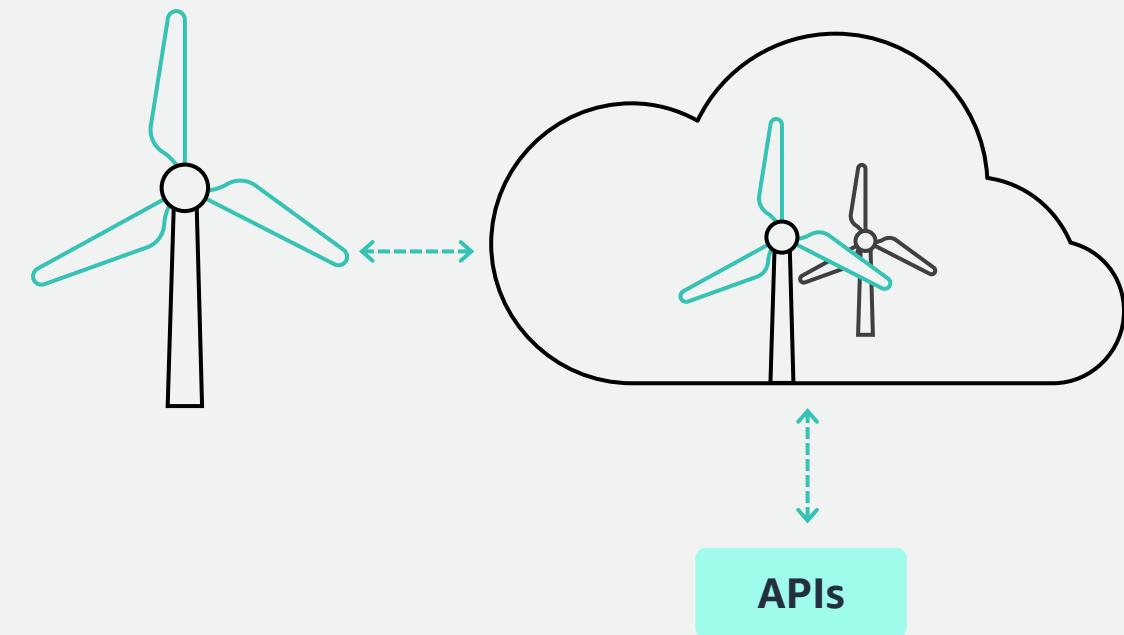
오프라인 디바이스와 비동기적으로 통신

Device SDK를 활용하여 디바이스에서 쉽게 구현

Application이 디바이스와 상호 작용할 수 있는 REST API 지원

App과 디바이스에서 실시간으로 상태 정보를 Get/Update/Delete할 수 있도록 Device Shadow MQTT topic 제공

Connectivity and control services



Device Shadow Example

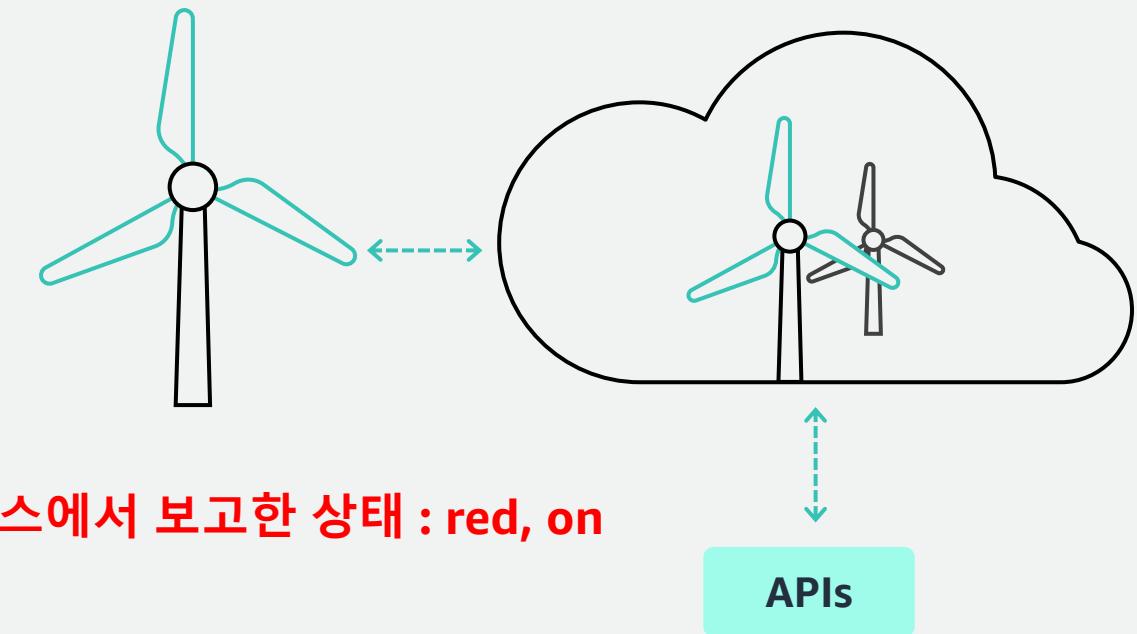
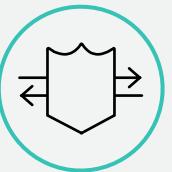
Device Shadow state

```
{  
  "state": {  
    사용자가 변경하고 싶은 상태 : blue, on  
    "desired": {  
      "status": "on",  
      "color": "blue"  
    },  
    현재 디바이스에서 보고한 상태 : red, on  
    "reported": {  
      "status": "on",  
      "color": "red"  
    },  
    Desired와 reported 간의 차이 상태 : blue  
    "delta": {  
      "color": "blue"  
    }  
  }  
}
```



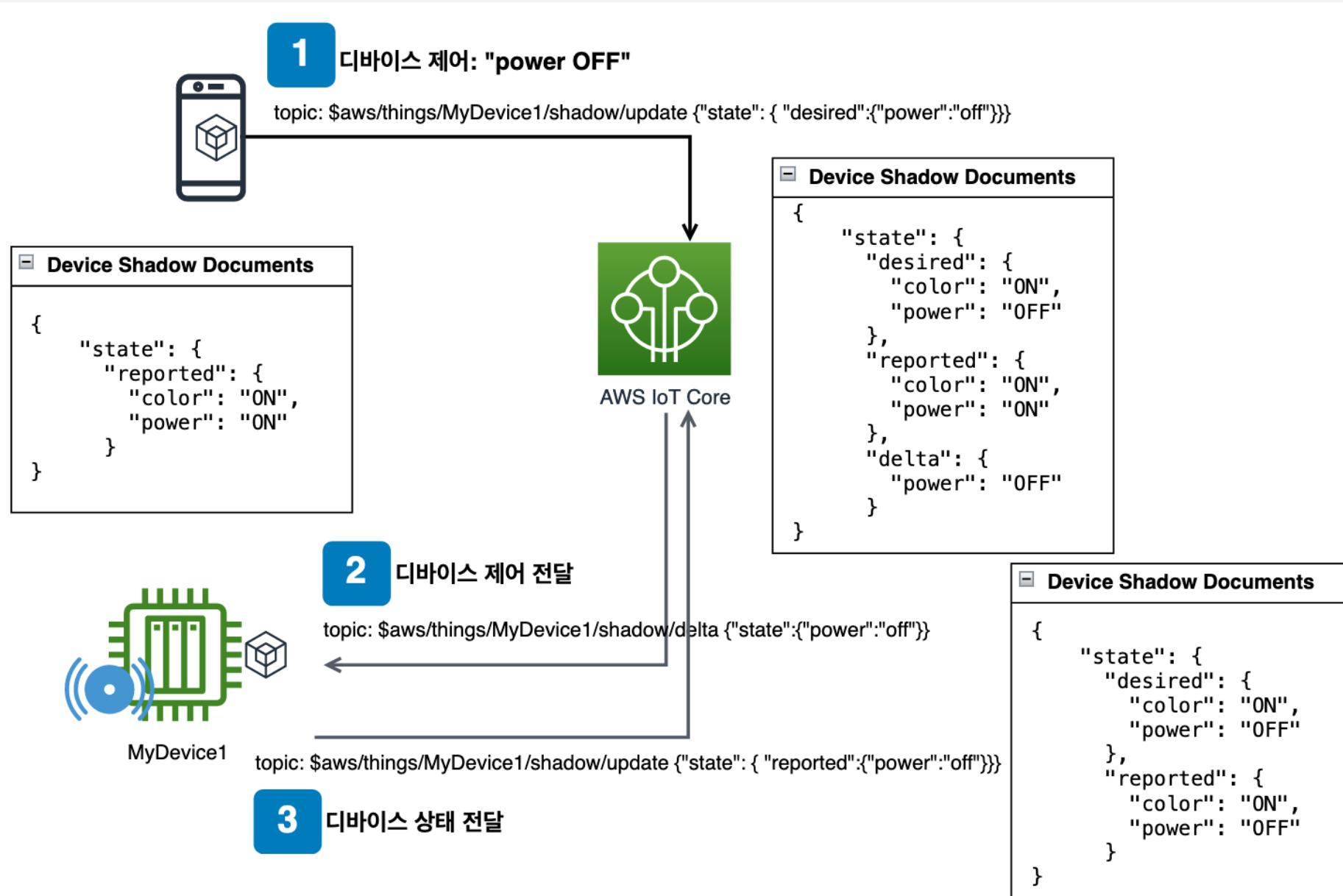
© 2024, Amazon Web Services, Inc. or its affiliates.

Connectivity and control services



Device Shadow Example

Connectivity and control services



Registry

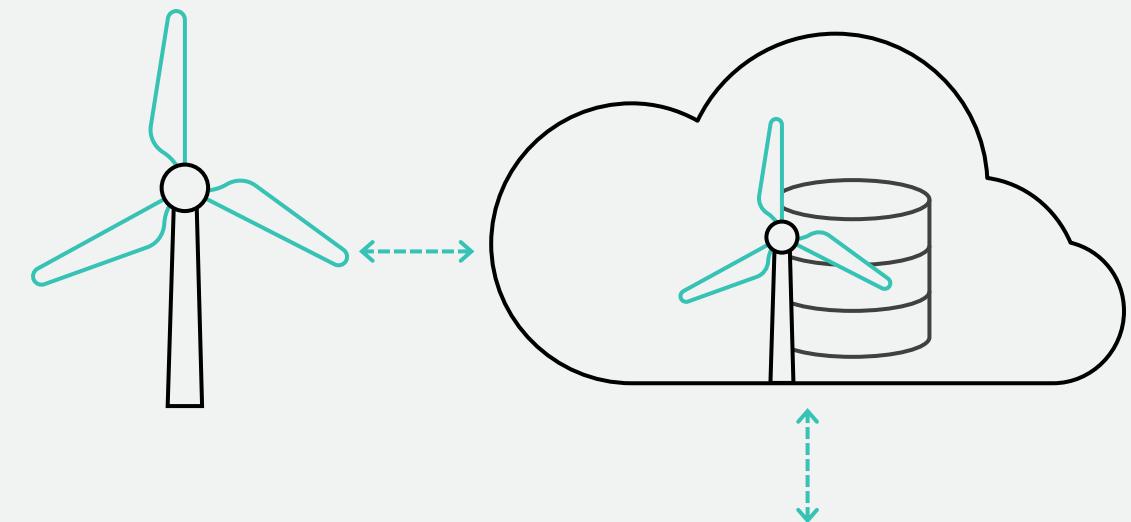
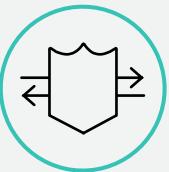
AWS 서비스에서 쉽게 사용할 수 있도록
디바이스 정의 및 카탈로그화

간단한 검색
(예를 들면, 2019년에 생성된 디바이스들은?)

디바이스 전반의 특성과 정책을 표준화할 수 있도록
Thing Type 정의
(예: 현대차와 기아차는 Thing Type Car임)

그룹(예: 차량 내 센서)을 정의하여
단순한 관리(작업 실행, 정책 설정 등) 가능

Connectivity and
control services



Static data
Serial #, meta data, etc.



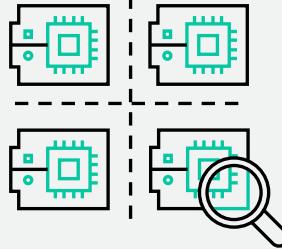
AWS IoT Device Management

Connectivity and control services



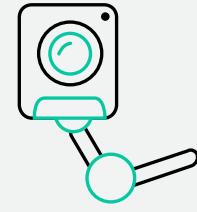
Bulk things registration

수백만 대의 디바이스에 애플리케이션 구축 및 관리



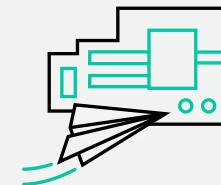
Fleet indexing and search

device fleet의 정상 여부 및 상태 파악



Device logging and monitoring

디바이스 로그를 수집하여 문제를 신속하게 식별하고 해결합니다.



Jobs

디바이스 그룹에 대한 작업 구성 및 시작



Secure tunneling

제한된 방화벽 뒤에 있는 디바이스에 안전하게 액세스



Fleet index and search

장치 집합의 상태 및 상태 이해

장치 속성의 조합을 기반으로 제품군의 장치 찾기

제조된 예: "현재 연결된 펌웨어 버전 1.2로 2015년 이후에 모든 장치 찾기"

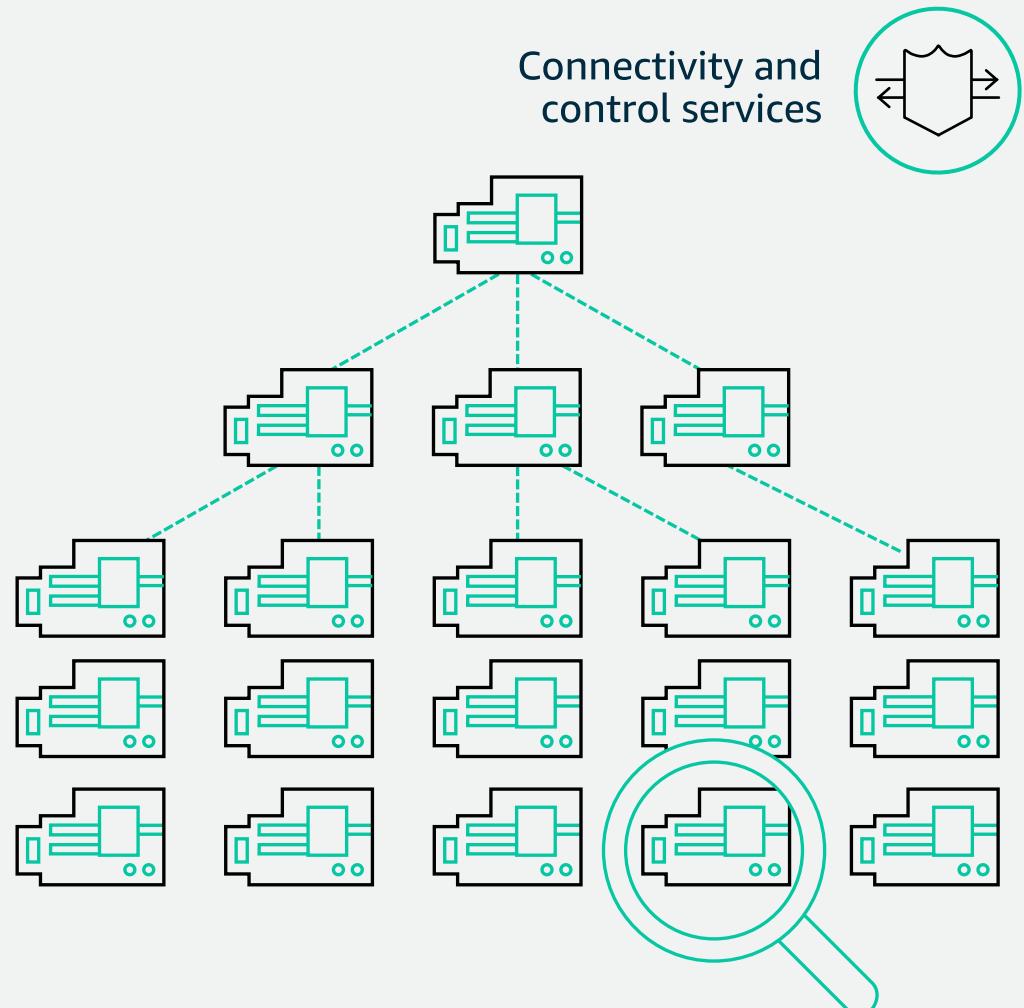
쿼리를 기반으로 동적으로 device 그룹을 업데이트하여 device 조직 자동화

예: "뉴욕의 모든 하드웨어 버전이 1.1인 전구 그룹화"

Amazon CloudWatch를 사용하여 정기적으로 fleet의 집계 정보 모니터링

간편한 사용 - 콘솔을 통한 one-click 활성화

<https://docs.aws.amazon.com/iot/latest/developerguide/managing-fleet-index.html>



`aws iot search-index --index-name "AWS_Things" --query-string "thingName:mything*"`

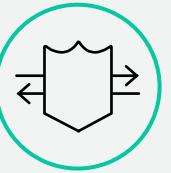
Fleet indexing

디바이스의 데이터를 사물과 사물 그룹을 대상으로 인덱싱, 검색 및 집계

- AWS IoT 레지스트리
 - *attributes, thingName, thingTypeName, thingGroupNames, ...*
- AWS IoT Device Shadow (namedShadow 포함)
 - *desired, reported, hasDelta*
- AWS IoT connectivity
 - *connected, disconnectReason, timestamp*
- AWS IoT Device Defender 위반
 - *violationCount, disconnectBehavior*

<https://docs.aws.amazon.com/iot/latest/developerguide/example-queries.html>





Connectivity and
control services

Jobs

Organize and trigger actions on groups of devices

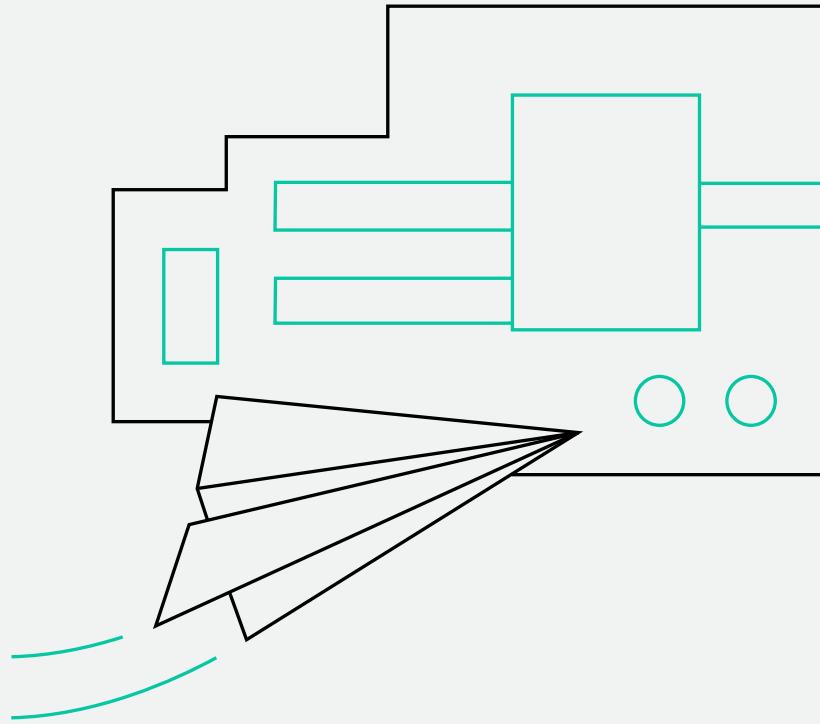
개별 장치에 대한 상태 업데이트를 수신하여
업데이트를 실행하는 동안 모니터링

대량으로 업데이트할 대상으로
device group 또는 단일 device 지정

배포 속도를 제어하고 Fail 기준을 설정하여 업데이트
시 적용 반경을 감소

device가 손상되는 것을 방지하기 위해 작업이
device로 전송되기 전에 디지털 서명

빈번한 원격 작업을 템플릿으로 저장하고,
대상을 선택하고, 필요에 따라 작업을 재사용



Job Template

AWS IoT > Manage > Remote actions > Job templates

Job templates Info

Jobs define remote actions to send to and run on devices that are connected to AWS IoT. Jobs that are frequently used can be converted to a job template for quicker deployment. To help accelerate implementation, AWS provides some public templates under job templates.

[Custom templates](#) [AWS managed templates](#)

Custom templates (2) Info

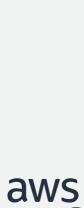
Templates created by you that are used to deploy a job to your fleet using the console or Fleet Hub.

Filter job templates by name

Name	Description	Created at
<input type="radio"/> DeviceErrorSimulation	Simulating errors on the device side	March 14, 2023, 10:55:41 (UTC+0900)
<input type="radio"/> FirmwareUpdate	Updating firmware to version: 2.0	March 14, 2023, 10:54:54 (UTC+0900)

Delete

Create job template



<https://docs.aws.amazon.com/iot/latest/developerguide/job-templates-managed.html>

© 2024, Amazon Web Services, Inc. or its affiliates.

AWS IoT > Manage > Remote actions > Job templates > Create job template

Create job template Info

Use job templates to create reusable AWS IoT job configurations.

Job template properties

Job template name

Enter a name for this template to help others understand what this job does, such as "reboot_device."

FirmwareUpdate

This field can only have alphanumeric, underline (_), and dash (-) characters.

Job template description

Enter a description to provide additional details about the job template to others.

Updating firmware to version: 2.0

2028 characters max.

Job document

The action that will be sent to your device using a job document that is either a JSON file stored in S3 or a JSON formatted document that is specified within the job.

Job file

A JSON file to upload to S3.

S3 URL

dv93yso/iot_job_firmware_update.json View Browse S3

No pre-signing URL found

To use pre-signing URL, a placeholder snippet is required in the job file.

Additional configurations - optional

An optional step to define your template's rollout speed, cancel configuration, and so on.

Rollout configuration

Rate

Constant rate

Exponential rate

Maximum per minute (1-1000)

5

Scheduling configuration - new

Job executions timeout configuration

Job executions retry configuration

Abort configuration

Tags - optional

Apply tags to your job templates to help organize and identify them. A tag consists of a case-sensitive key-value pair.

No tags are associated with the resource.

Add tag

You can add 50 more tags.

Cancel

Create job template

IoT Job Example

IoT Fleet Hub
Dashboard

shadow.reported.firmware_version ▾ | Q 1.0

Summary Device list

Devices (5 selected)

Name	Thing type	Thing groups	Connected	Last connection timestamp	Disconnect reason	Behavior violation count	Behavior violations	Classic shadow: reported.battery_stat_e_of_charge	Classic shadow: reported.firmware_version
fh_workshop_83767f5c-ced9-4262-abdc-bbb5964b0061	dispenserV1	AllWorkshopDevices, Firmware-1:0	<input checked="" type="checkbox"/> true	March 14, 2023 01:28 (UTC)	-	-	-	1	1.0
fh_workshop_7aac6eeb-f97c-413b-9d9e-763ed230f895	dispenserV1	Firmware-1:0, AllWorkshopDevices	<input checked="" type="checkbox"/> true	March 14, 2023 01:24 (UTC)	-	-	-	90	1.0
fh_workshop_9019489c-6870-47fd-853a-f969931162b6	dispenserV1	BostonDevices, Firmware-1:0, 1 more	<input checked="" type="checkbox"/> true	March 14, 2023 01:28 (UTC)	-	-	-	90	1.0
fh_workshop_19d1d7e1-961f-4ca5-a862-ffad2da3a5ff59	dispenserV1	Firmware-1:0, AllWorkshopDevices	<input checked="" type="checkbox"/> true	March 14, 2023 01:29 (UTC)	-	-	-	10	1.0
fh_workshop_f813d79b-2b89-485d-bcda-c2db6440b03	dispenserV1	AllWorkshopDevices, Firmware-1:0	<input checked="" type="checkbox"/> true	March 14, 2023 01:29 (UTC)	-	-	-	1	1.0

Run jobs

Run jobs on 5 devices

Job type

- Snapshot - Your job will complete after deploying to the selected devices.
- Continuous - Only available when running jobs on device groups.

Select job template

Choose a job template.

FirmwareUpdate

Description

Updating firmware to version: 2.0

Execution failure

Job name - optional

You can add a custom job name to help identify the job.

upgrade-test

64 characters maximum. Only letters, numbers, hyphens (-), and underscores (_).

Close Run

- 타겟 디바이스 또는 디바이스 그룹 설정하여 배포
- 배포 타입:
 - Snapshot: 지금 적용된 타겟 디바이스
 - Continuous: 타겟 디바이스 그룹에 추가될 때마다 실행



IoT Fleet Hub > Jobs

Jobs

All statuses	All types	Job name	Job status	Target	Job type	Last update	Created at
<input type="radio"/>	<input checked="" type="radio"/>	upgradeerror-test	<input checked="" type="radio"/> In progress	1 group	Snapshot	March 14, 2023 02:06 (UTC)	March 14, 2023 02:06 (UTC)
<input type="radio"/>	<input checked="" type="radio"/>	upgrade-test	<input checked="" type="radio"/> Completed	5 devices	Snapshot	March 14, 2023 02:04 (UTC)	March 14, 2023 02:03 (UTC)

Cancel job

IoT Job status Example

IoT Fleet Hub > Jobs > disconnect-test

disconnect-test

[Cancel job](#)

Job details

Job status	1 group	Target device	Timeout configuration	Recurrences (UTC)
In progress	-	-	-	-
Description	Snapshot	Execution type	Job execution failure configuration	Maintenance window duration
-	-	-	-	-
Created at	March 14, 2023 02:09 (UTC)			
Last updated at	March 14, 2023 02:10 (UTC)			

Execution overview

When canceling and deleting execution refresh this page if the summary does not match.

Succeeded	In progress	Queued	Failed
2	5	1	0
Timed out	Rejected	Canceled	Removed
0	0	0	0

Executions (8)

Devices currently processing this job.

[Cancel execution](#)

All executions (8) ▾	Device name	Retry attempts	Retries remaining	Execution status	Queued at	Last update
All executions (8) ▾	fh_workshop_a5703092-1cff-4ef4-8df6-f5027bc9ea1f	-	Retries not set	In progress	March 14, 2023 02:10 (UTC)	March 14, 2023 02:10 (UTC)
	fh_workshop_cf587299-df22-4375-b0fe-cd64f4a2e3df	-	Retries not set	In progress	March 14, 2023 02:10 (UTC)	March 14, 2023 02:10 (UTC)
	fh_workshop_0cf51b840-bb26-4fb0-9639-af54395f5d4	-	Retries not set	Succeeded	March 14, 2023 02:10 (UTC)	March 14, 2023 02:10 (UTC)
	fh_workshop_fc9c405-f9e9-4252-bdea-e9438702fd29	-	Retries not set	Queued	March 14, 2023 02:10 (UTC)	March 14, 2023 02:10 (UTC)
	fh_workshop_83767f5c-ced9-4262-abdc-bbb5964b0061	-	Retries not set	In progress	March 14, 2023 02:10 (UTC)	March 14, 2023 02:10 (UTC)
	fh_workshop_78c02f74-386e-48a7-b278-b25e8b3d9ec8	-	Retries not set	In progress	March 14, 2023 02:09 (UTC)	March 14, 2023 02:09 (UTC)
	fh_workshop_58bd6f0e-2e7b-4d8e-8956-40b2b225e6b8	-	Retries not set	In progress	March 14, 2023 02:09 (UTC)	March 14, 2023 02:09 (UTC)
	fh_workshop_2be44e56-b308-4d8c-9c66-69c458f8635f	-	Retries not set	Succeeded	March 14, 2023 02:09 (UTC)	March 14, 2023 02:09 (UTC)

IoT Fleet Hub > Jobs > upgrade-test

upgrade-test

[Cancel job](#)

Job details

Job status	5 devices	Target device	Timeout configuration	Recurrences (UTC)
Completed	-	-	-	-
Description	Snapshot	Execution type	Job execution failure configuration	Maintenance window duration
-	-	-	-	-
Created at	March 14, 2023 02:03 (UTC)			
Last updated at	March 14, 2023 02:04 (UTC)			

Execution overview

When canceling and deleting execution refresh this page if the summary does not match.

Succeeded	In progress	Queued	Failed
5	0	0	0
Timed out	Rejected	Canceled	Removed
0	0	0	0

Executions (5)

Devices currently processing this job.

[Cancel execution](#)

All executions (5) ▾	Device name	Retry attempts	Retries remaining	Execution status	Queued at	Last update
All executions (5) ▾	fh_workshop_f813d79b-2b89-485d-bcda-c22db6440b03	-	Retries not set	Succeeded	March 14, 2023 02:04 (UTC)	March 14, 2023 02:04 (UTC)
	fh_workshop_19d1d7e1-961f-4ca5-a862-ffa2d8a3ff59	-	Retries not set	Succeeded	March 14, 2023 02:03 (UTC)	March 14, 2023 02:03 (UTC)
	fh_workshop_9019489c-6870-47fd-853a-f969931162b6	-	Retries not set	Succeeded	March 14, 2023 02:03 (UTC)	March 14, 2023 02:03 (UTC)
	fh_workshop_7aac6eeb-f97c-413b-9d9e-763ed230f895	-	Retries not set	Succeeded	March 14, 2023 02:03 (UTC)	March 14, 2023 02:03 (UTC)
	fh_workshop_83767f5c-ced9-4262-abdc-bbb5964b0061	-	Retries not set	Succeeded	March 14, 2023 02:03 (UTC)	March 14, 2023 02:03 (UTC)



IoT Job – Workflow

1. 디바이스는 예약된 job 관련 topic 구독
2. AWS IoT Core -> **notify** topic으로 새 Job 알림
3. 디바이스 -> **get** topic으로 Job 세부사항 요청
4. 디바이스 -> **update** topic으로 "IN_PROGRESS" 상태 보고
 - 세부 진행 상황 % 형태로 전달 가능
5. Job 실행: 디바이스 별 필요한 작업 로직 구현
6. 디바이스 -> **update** topic으로 "SUCCEEDED" 상태 보고
7. AWS IoT Core -> **completed** topic으로 Job 완료 알림

```
$aws/things/THING_NAME/jobs/notify  
$aws/things/THING_NAME/jobs/notify-next  
$aws/things/THING_NAME/jobs/start-next  
  
$aws/things/THING_NAME/jobs/get  
$aws/things/THING_NAME/jobs/JOB_ID/get  
  
$aws/things/THING_NAME/jobs/JOB_ID/update  
  
$aws/events/job/JOB_ID/completed  
$aws/events/job/JOB_ID/canceled  
$aws/events/job/JOB_ID/deleted  
$aws/events/job/JOB_ID/cancellation_in_progress  
$aws/events/job/JOB_ID/deletion_in_progress
```



Secure tunneling

Securely access devices
behind restricted firewalls

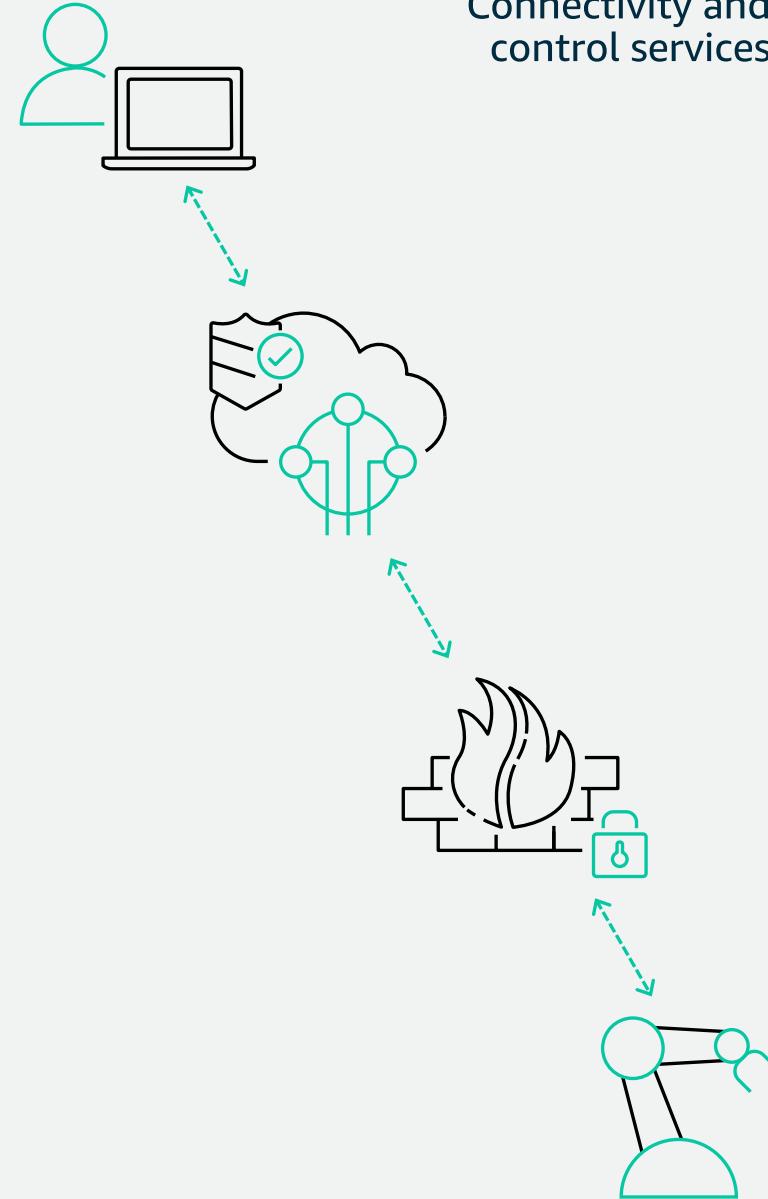
API 또는 콘솔을 사용하여 터널을 안전하게 열고
닫습니다.

원격 셸 또는 원격 데스크톱 작업을 사용하여 개별
장치에 액세스

IAM 권한과 최대 12시간의 구성 가능한 제한 시간을
사용하여 각 터널 세션에 대한 장치 권한 부여를
관리합니다.

MQTT 메시지를 통해 디바이스별 액세스 토큰
자동 수신

Connectivity and
control services



세분화된 장치 로깅 및 모니터링

장치 로그를 수집하여 문제 파악 및 해결

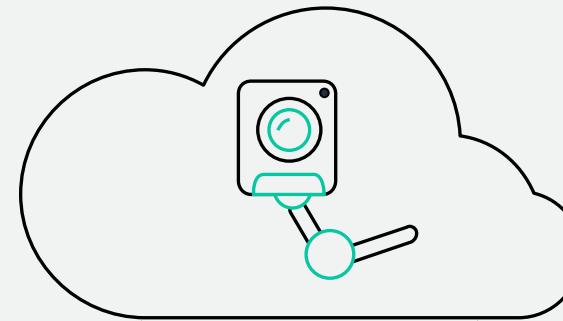
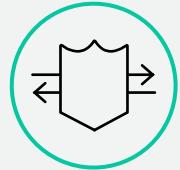
디바이스별 또는 디바이스 그룹별로 로깅 수준 구성

문제를 해결하려면 오작동하는 장치의 하위 집합에서 선택적으로 진단 수준을 높일 수 있습니다.

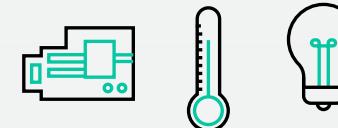
Amazon CloudWatch를 사용하여 경보 구성 및 로그 검색

- Error => 상용화 (Production)
 - Warn => 중소 규모의 부하 테스트
 - Info => 검증 테스트
 - Debug => 개발
 - Disabled
- aws

Connectivity and control services



Amazon CloudWatch



- Rule Metrics

- ParseError
- Action Failure

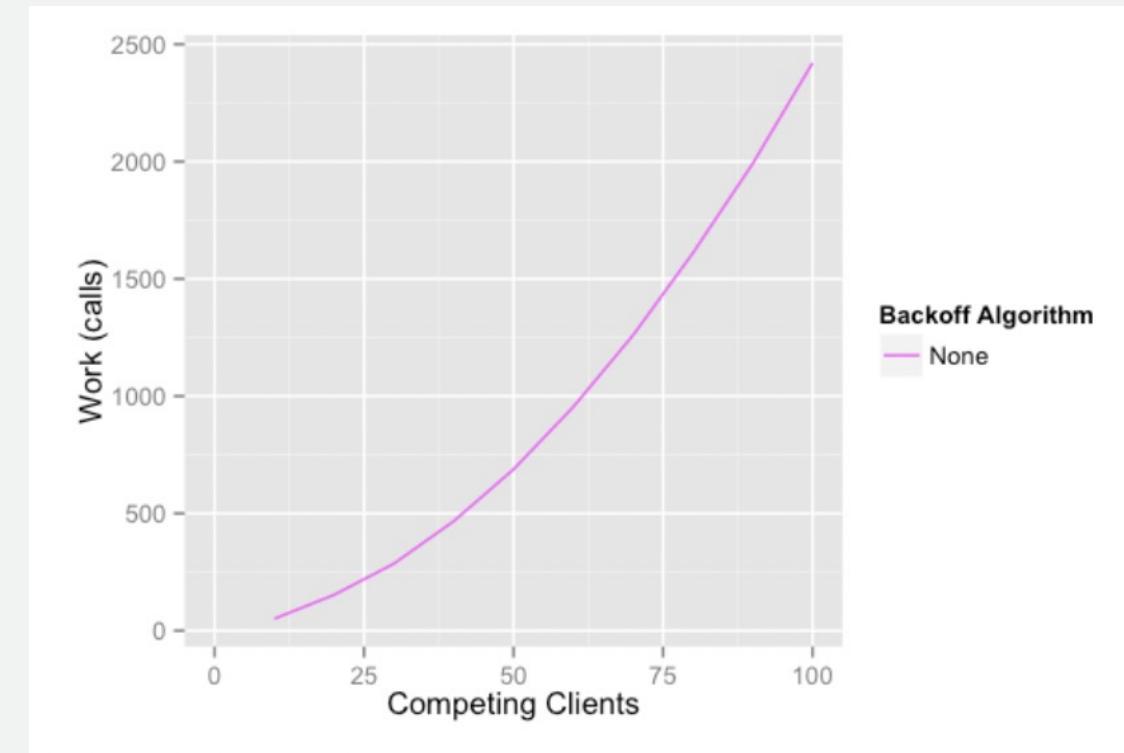
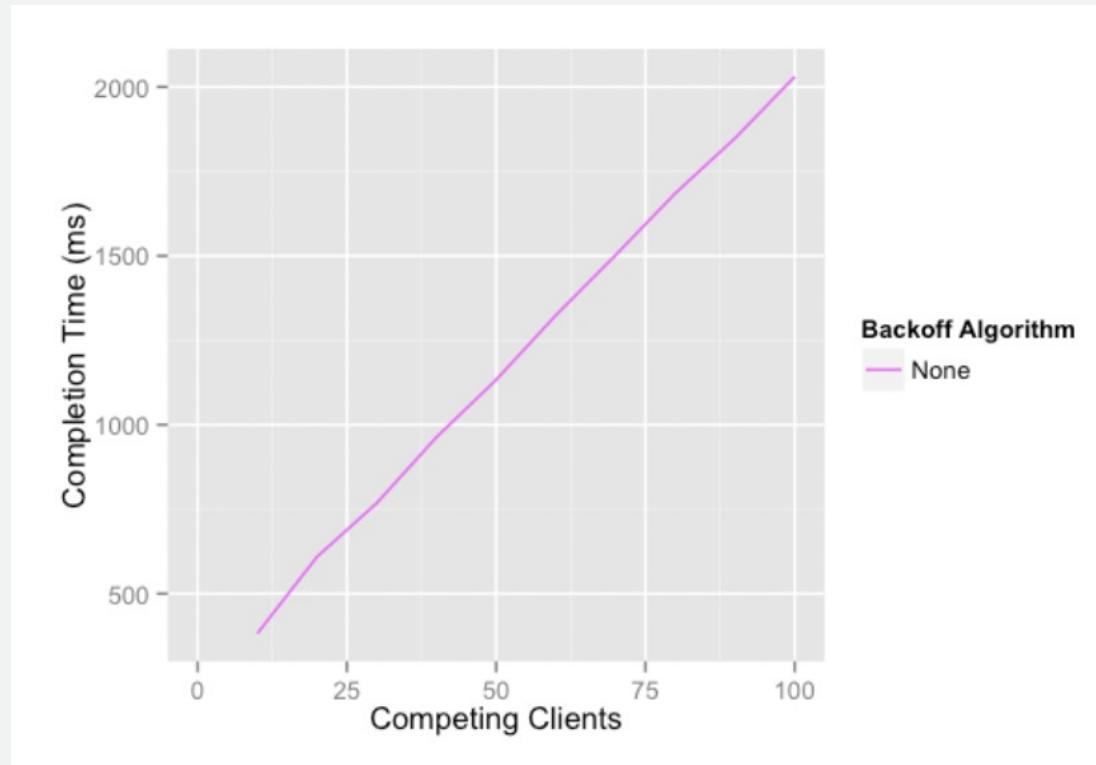
- Broker Metrics (Auth Errors)

- Connect.AuthError
- Publishin.AuthError
- PublishOut.AuthError
- Subscribe.AuthError

- Shadow Metrics

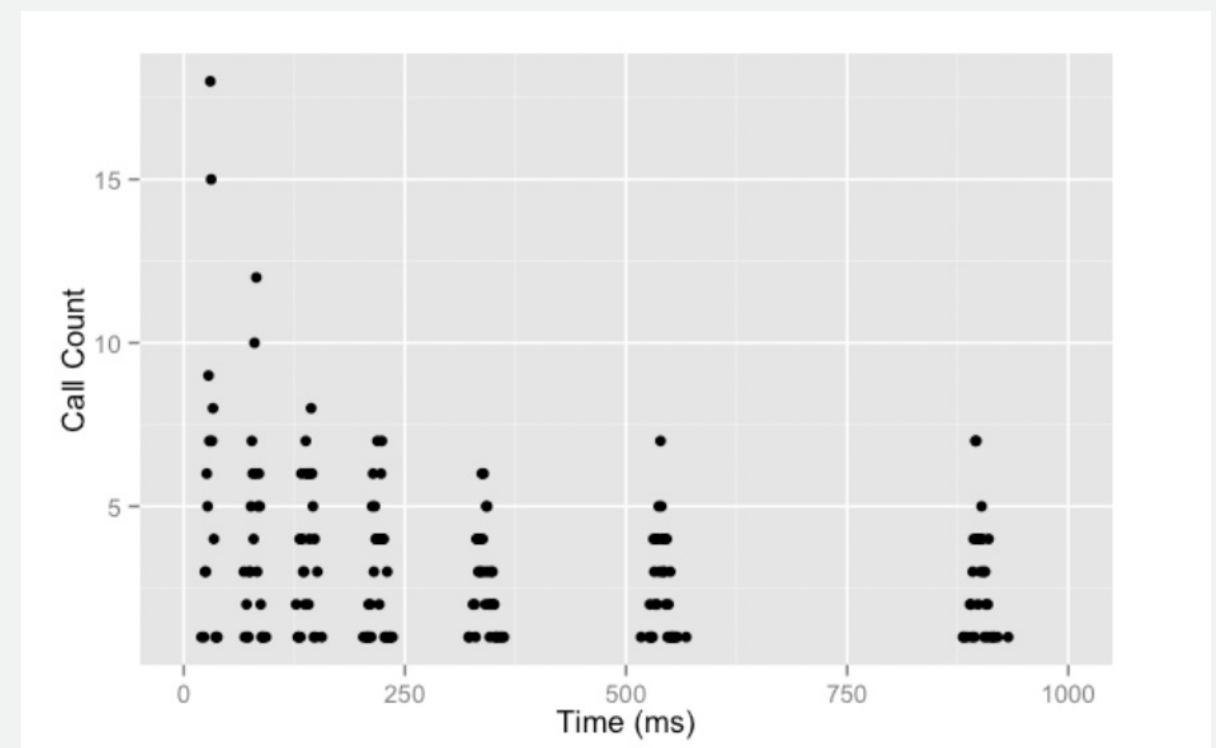
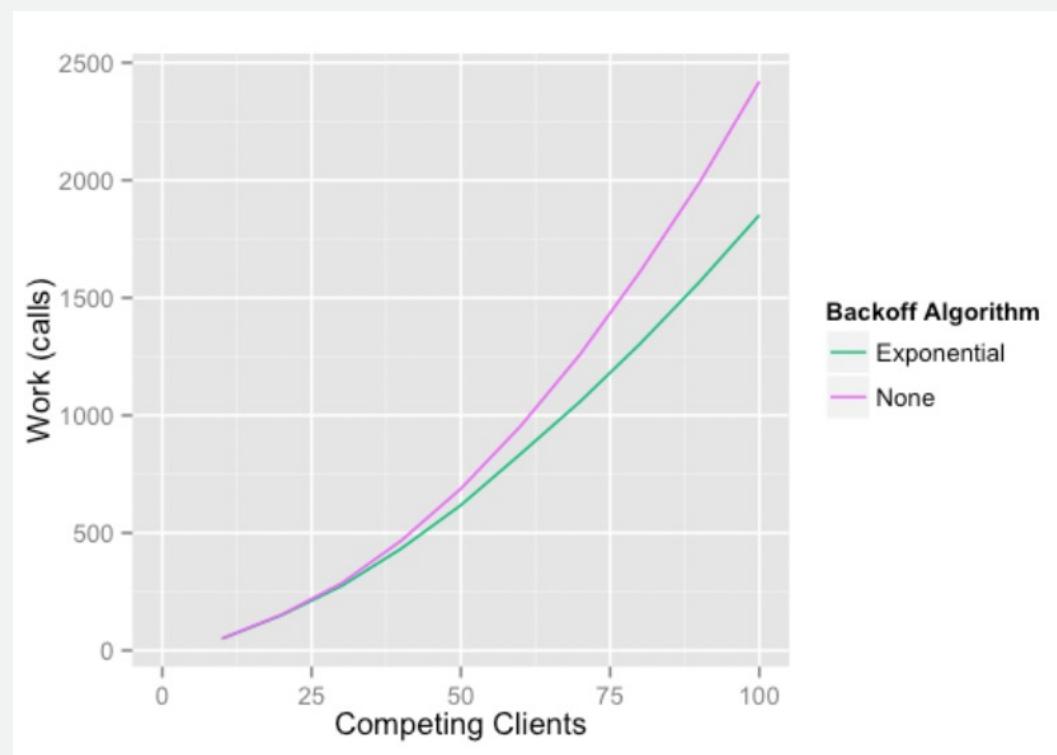
- DeleteThingShadow.Accepted
- GetThingShadow.Accepted
- UpdateThingShadow.Accepted

네트워크가 일시적으로 끊김으로 인한 Client의 동시 연결



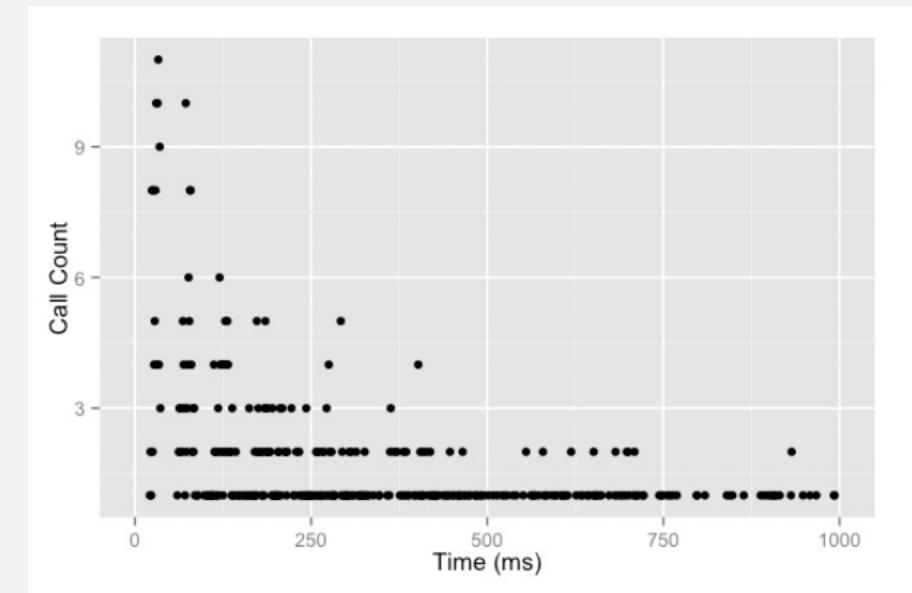
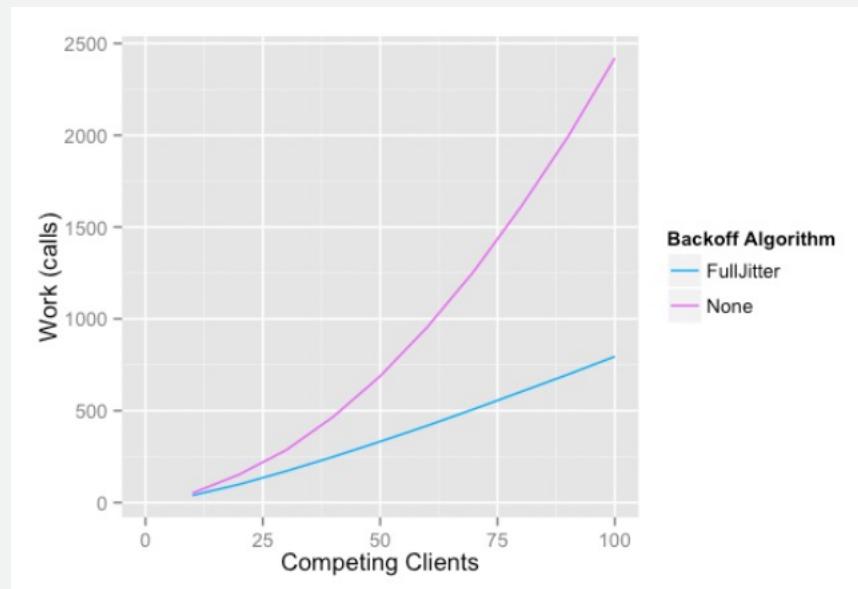
Adding exponentially backed-off: 점진적으로 시간 증가

`sleep = min(cap, base * 2 ** attempt)`



Adding Jitter: 랜덤하게 시간 간격 조정

`sleep = random_between(0, min(cap, base * 2 ** attempt))`

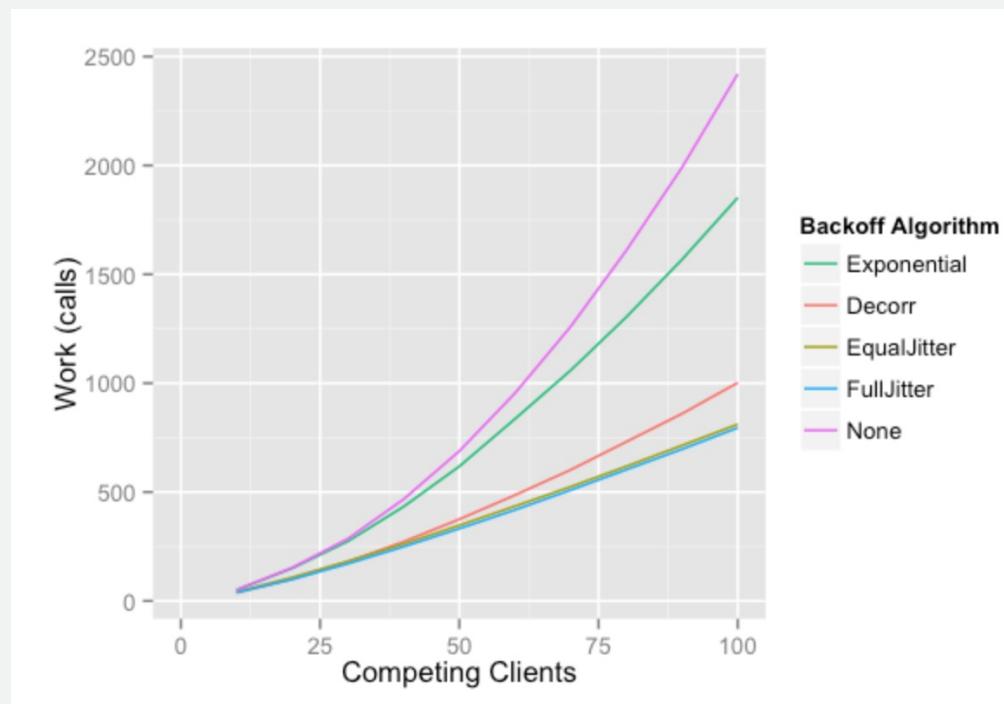


다양한 추가 구현 방안

Equal Jitter: 항상 일부 back-off와 jitter를 더 적은 양으로 유지

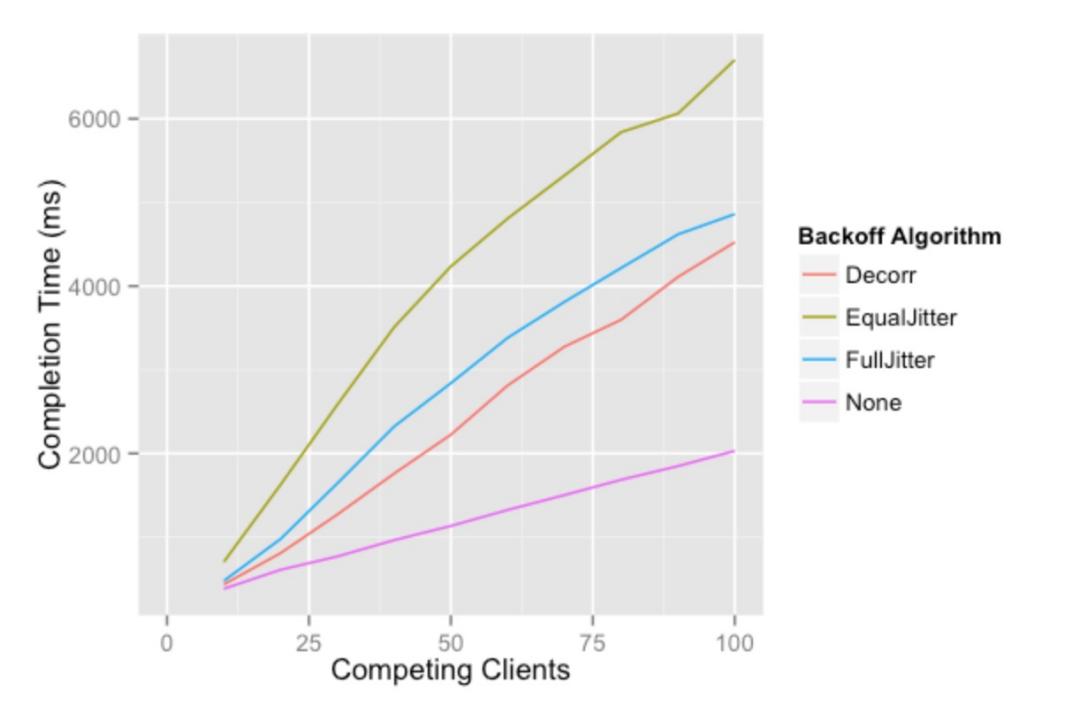
```
temp = min(cap, base * 2 ** attempt)
```

```
sleep = temp/2 + random_between(0, temp/2)
```



Decorrated Jitter: 마지막 랜덤 값을 기준으로 최대 jitter 증가

```
sleep = min(cap, random_between(base, sleep * 3))
```



참고 링크

<https://aws.amazon.com/ko/blogs/architecture/exponential-backoff-and-jitter/>

<https://github.com/aws-samples/aws-arch-backoff-simulator>

<https://github.com/aws/aws-iot-device-sdk-embedded-C#backoffalgorithm>

Quotas

Adjustable: No

512K Throughput per second per connection

128k Message Payload

50 Subscriptions per Connection

1 Connect requests per second per client ID

Adjustable: Yes

20,000 Inbound publish requests per second per account

500,000 Maximum concurrent client connections per account

20,000 Outbound publish requests per second per account

1 hour Persistent session expiry period

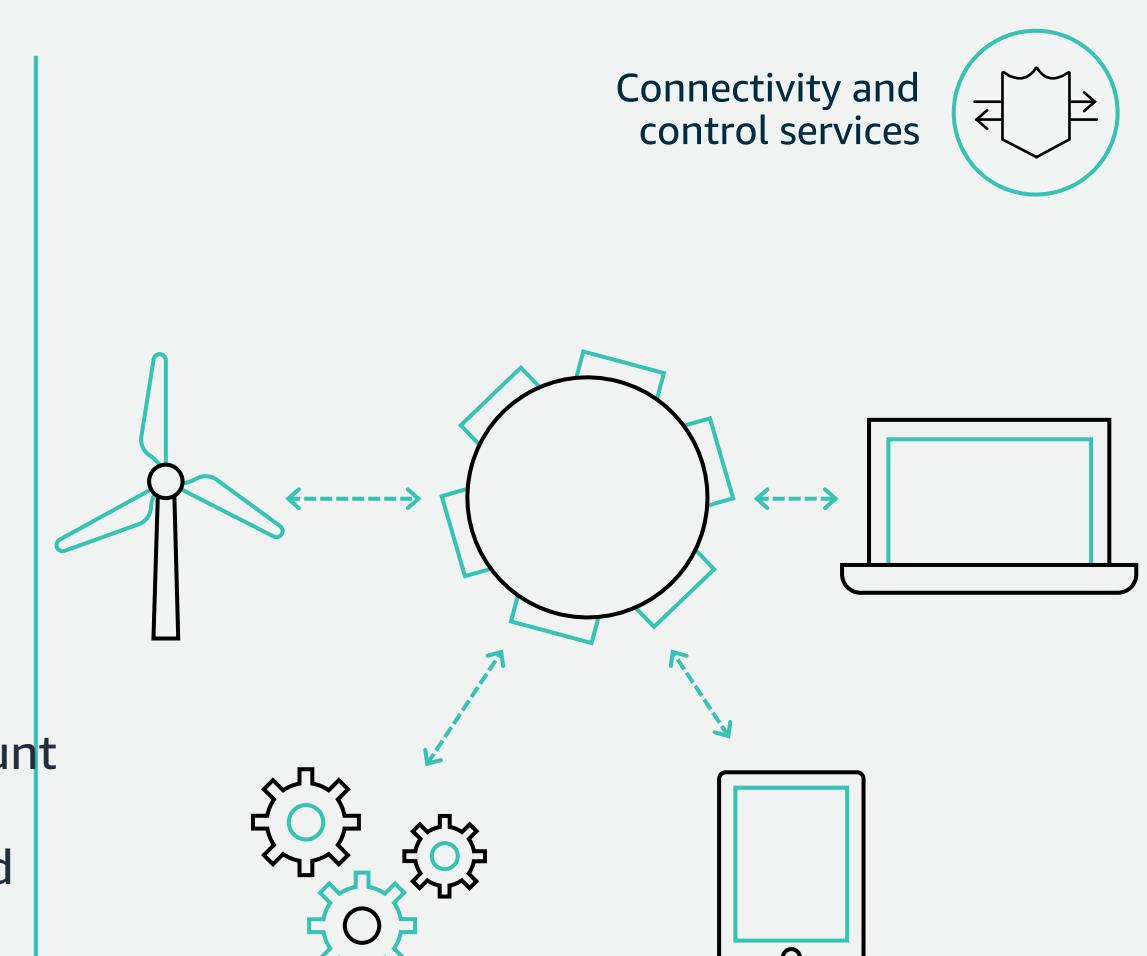
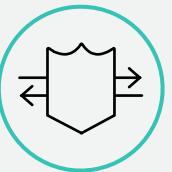
500,000 Subscriptions per account

500 Subscriptions per second per account

<https://docs.aws.amazon.com/general/latest/gr/iot-core.html>

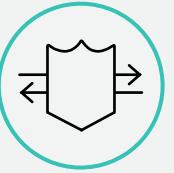


Connectivity and control services



Pricing for AWS IoT Core

Connectivity and control services

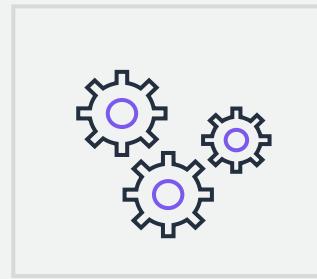


Connectivity



\$0.096 (per million minutes of connection)

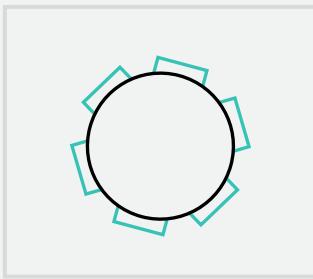
Rule Action



\$0.18 (per million rules triggered / per million actions executed)

* 5KB 단위 과금

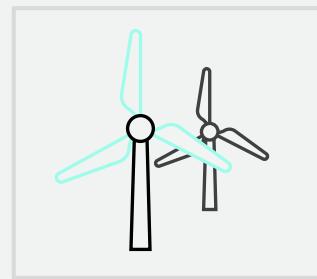
Message broker



\$1.20 (per million messages)

* 5KB 단위 과금

Device shadow & Registry



\$1.50 (per million operations)

*1KB 단위 과금

<https://aws.amazon.com/iot-core/pricing/>





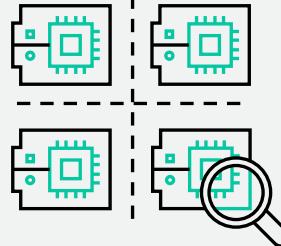
AWS IoT Device Management pricing

Connectivity and control services



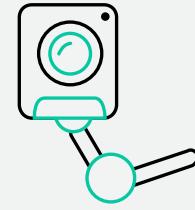
Bulk things registration

\$0.12
(per 1,000 things registered)



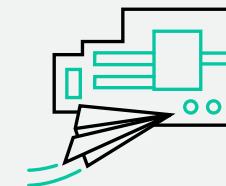
Fleet indexing and search

Index Updates: \$2.70
(per 1 million updates)
Search queries: \$0.06
(per 10,000 queries)



Device logging and monitoring

<https://aws.amazon.com/cloudwatch/pricing>



Jobs

\$0.0036
(per remote action / Month)



Secure tunneling

\$1.20
(per 1 tunnel opened)





Thank you!

김성한

senghank@amazon.com