AWS IoT Services running on STM32

AWS Solutions Architect
이세현

September 2020

AWS IoT architecture

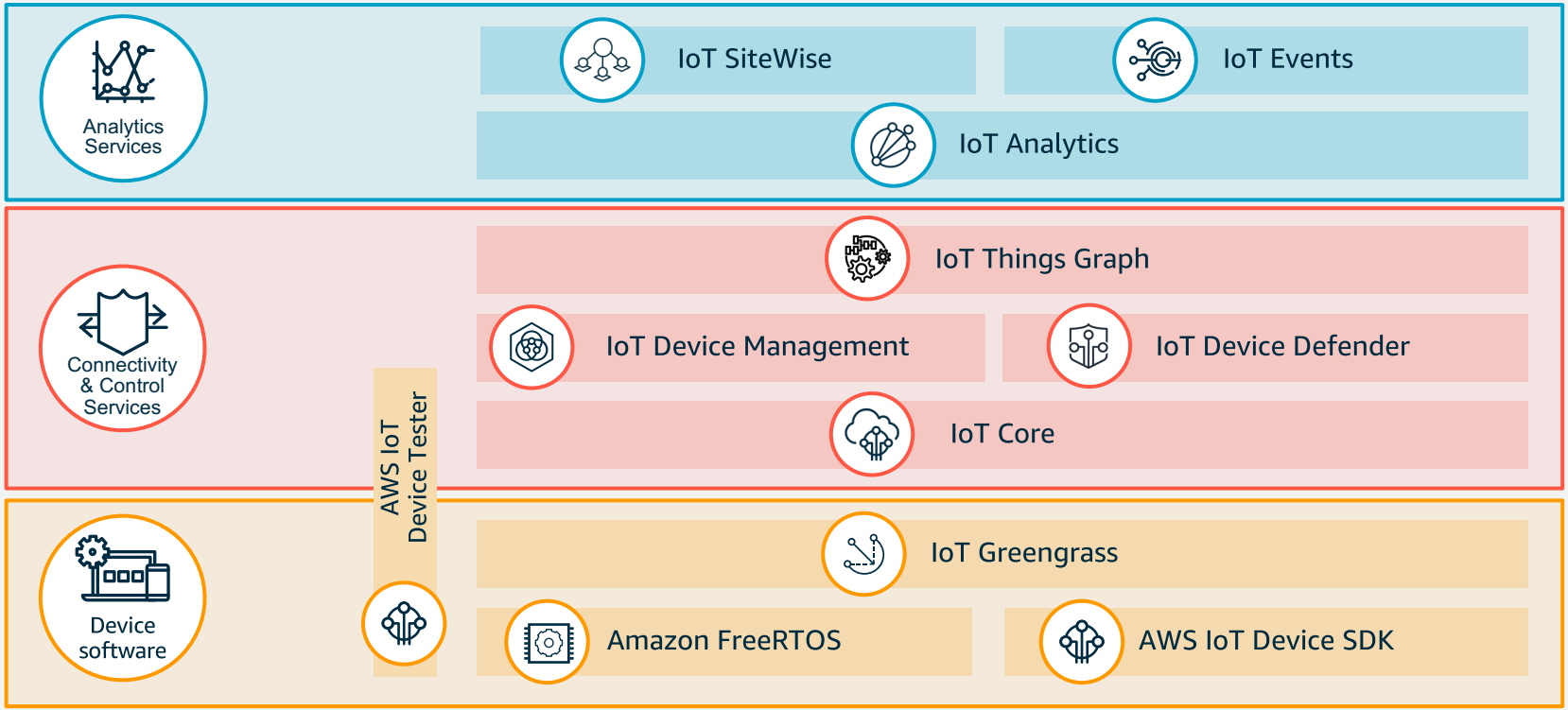
IoT 데이터에서 가치를 어떻게 추출합니까?

장치를 제어, 관리 및 보호하려면 어떻게 해야합니까?

장치를 연결하고 edge에서 작동하려면 어떻게 해야합니까?

aws

AWS IoT architecture



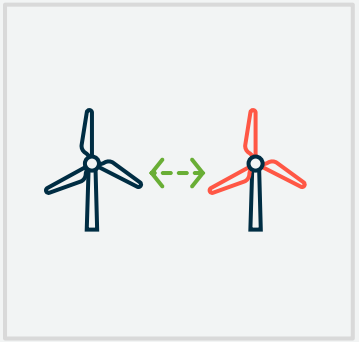
AWS IoT Core

AWS IoT Core

AWS IoT Core는 연결된 디바이스가 cloud 애플리케이션 및 기타 디바이스와
쉽고 안전하게 상호 작용할 수 있도록 하는 관리형 서비스입니다.

대규모로 디바이스를 AWS cloud 및 기타 디바이스에 안전하게 연결하기 위하여

연결된 장치의 데이터를 라우팅 및 처리하기 위하여



오프라인 상태에서도 응용 프로그램이 장치와 상호 작용할 수 있도록 하기 위하여

다른 AWS 서비스와 완전히 통합하여 데이터를 기반으로 추론하기 위하여
(Analytics, Databases, AI, etc.)

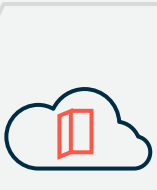
Connectivity
&control
services

aws

AWS IoT Core

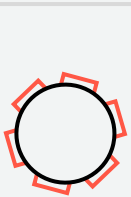
Identity Service

규모에 따라 기기 인증 관리 및 고유 한 ID로 provision



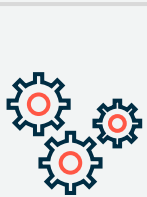
Device Gateway

IoT 워크로드에 최적화 된 연결 관리



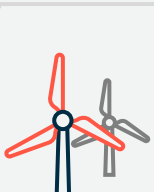
Message Broker

IoT fleet 전체에서 안정적이고 빠른 통신



Rules Engine

저렴한 비용으로 대량의 IoT 데이터를 수집하고 전처리& 분석,보고 및 시각화를 위해 10 개 이상의 서비스에 제공



Device Shadow

언제든지 장치 상태를 이해하고 제어

Registry

AWS 서비스에서 쉽게 사용할 수 있도록 디바이스 정의 및 카탈로그화

Connectivity &control services

aws

Message Broker

확장 가능하고 강력한 publish-subscribe broker

분리된 디바이스 및 application에 대한 Publish/Subscribe

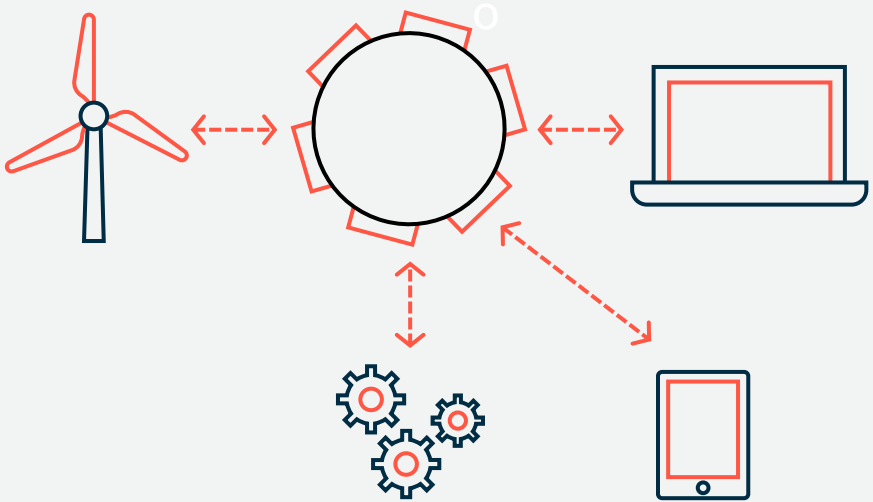
wildcard topic filter 지원

QoS0 와 QoS1 메시징

JSON과 Binary payload 지원

SDK 없이 모든 기능 사용 가능
+ 필요할 경우 일반적인 프로그래밍 언어를 지원하는 device SDK 제공

IoT Core 기능은 topic 및 reserved topic에 대해 구현되며 SDK를 사용하지 않아도 기능이 손실되지 않습니다.



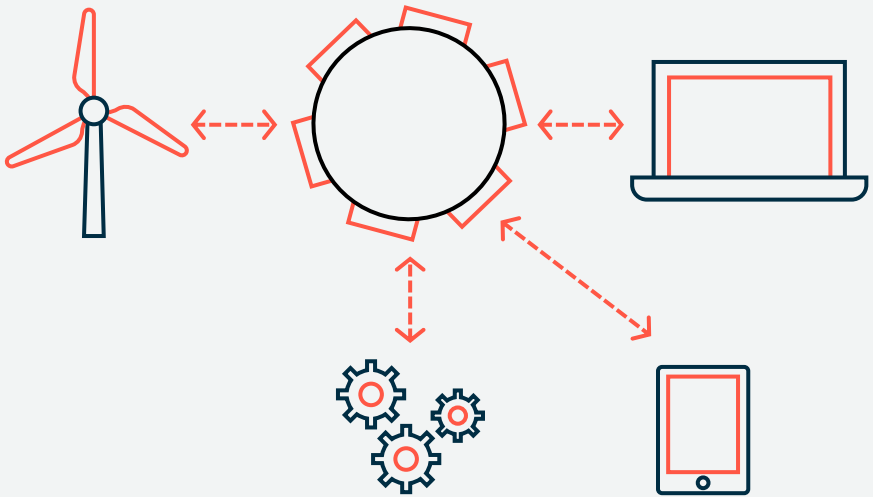
Connection Requirements

장치를 연결하는데 필요한 구성 요소:

x509 Device Certificate

Security Policy

Device를 등록



aws

Identity Service

규모에 따른 장치의 인증 관리 및 고유한 ID 제공

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

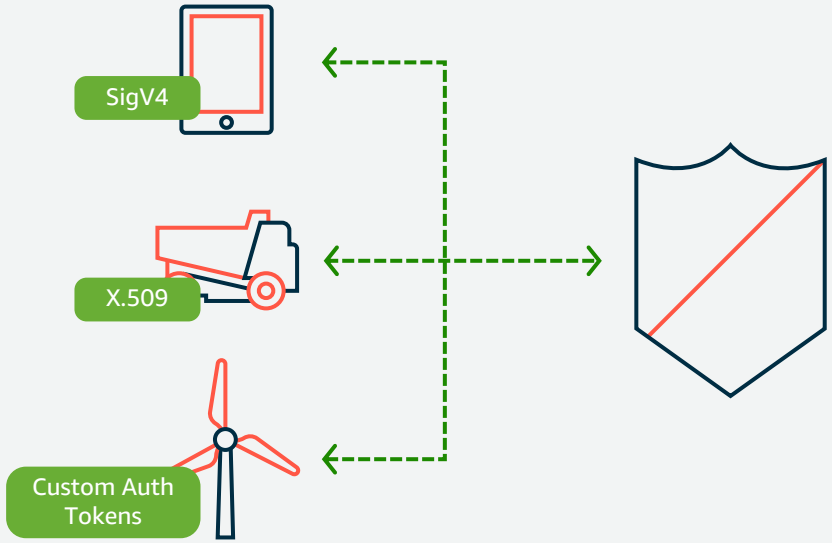
자동화된 device provisioning 방식 지원
- Just-In-Time Registration
- Just-In-Time Provisioning
- Fleet Provisioning

SigV4, X.509 그리고 token based authentication 지원 (Custom Authorizers를 통해서)

IoT정책을 통한 유연하고 세분화 된 액세스 제어

정책(policy)은 identities 또는 registry items과 연결될 수 있습니다.

MQTT topic level 단위로 엑세스를 제어할 수 있습니다.



Device Gateway

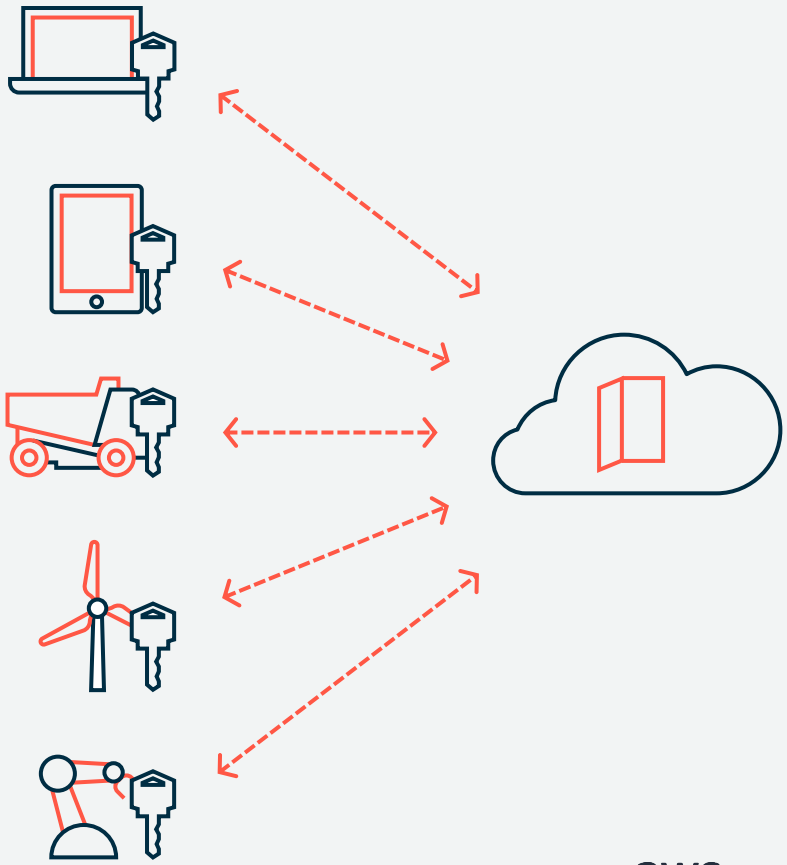
IoT 워크로드에 최적화 된 연결 관리

MQTT, WebSockets, HTTPS를 포함한 여러 프로토콜 지원

TLS 1.2 기반 암호화 통신

리소스가 제한적인 디바이스에 최적화됨

ECC Key Exchange 와 Certificates
Maximum Fragment Length Negotiation



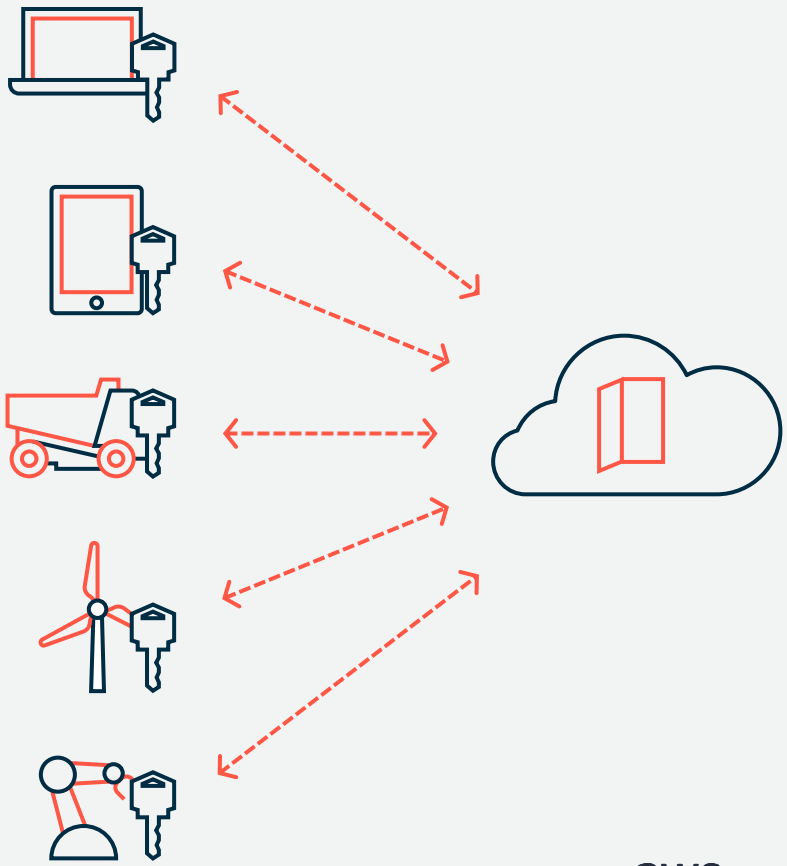
What can data look like?

사용자 지정 JSON payloads

```
PUBLISH turbines/ev-gen/123 (qos: 0)
 {
   "timestamp": "2016-11-29T10:00:00",
   "temperature": 125,
   "humidity": 95,
   ”rotor-freq": 6455,
   “output”: 480,
   “output-freq”: 60
 }
```

What about binary payloads?

Supported but limited



aws

Rules Engine

저렴한 비용으로 많은 양의 데이터를 수집하고 사전 처리하며 분석, 보고 및 시각화를 위한 10 개 이상의 서비스에 제공

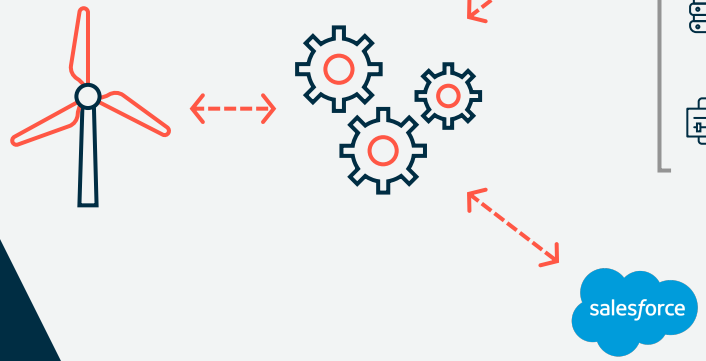
Transform—수학, 문자열 조작, 날짜 등을 위한 내장 함수

Filter—WHERE 절을 사용하여 필요한 데이터만 추출

Enrich—인라인 AWS Lambda 실행을 통해 Device Shadow 및 Amazon Machine Learning 또는 외부 소스에서 컨텍스트를 가져옵니다.

Route—10개가 넘는 AWS 서비스 및 Salesforce 등과 같은 3rd 서비스로 데이터를 전송

Direct Ingest



Analytics
Kinesis
Elasticsearch Services
IoT Analytics

Database
DynamoDB

Manage
CloudWatch

Compute
Lambda

Application Integration
SNS
SQS
Step Functions

aws

Device Shadow

언제든지 장치 상태를 이해하고 제어

디바이스의 최종 상태를 보고
예를 들면, 전구의 최종 상태는 빨간색(red)

디바이스의 상태를 변경; 예를 들면, 전구의 색상을 파란색(blue)으로 변경

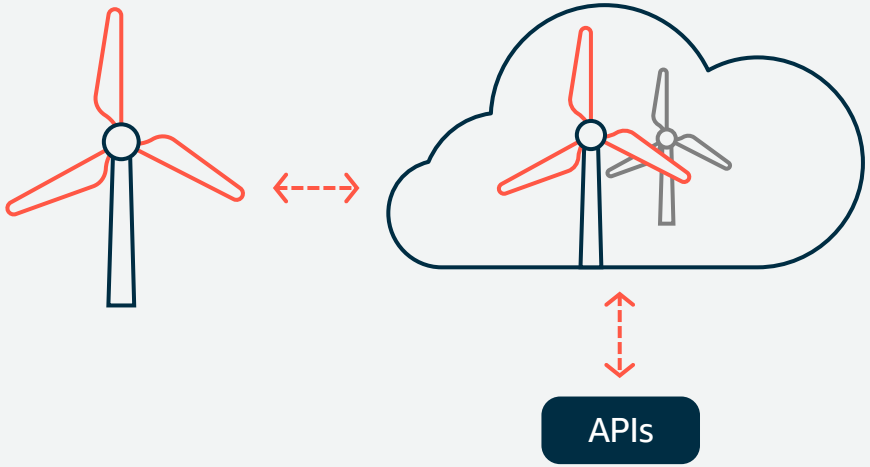
MQTT를 사용하여 상태 변경을 실시간으로 알림

오프라인 디바이스와의 비동기 통신

장치에서의 손 쉬운 개발을 위한 Device SDK 통합

Application이 디바이스와 상호 작용할 수 있는 REST API 지원

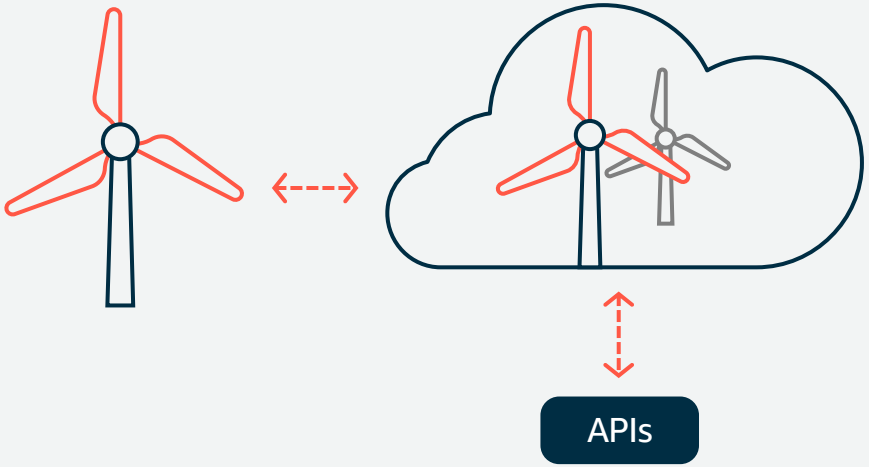
모든 명령 및 제어 작업을 위한 기본 기능



Shadow Example

The shadow is the most critical feature to understand....

```
"state" : {
        “desired" : {
            "lights": { "color": "RED" },
            "engine" : "ON"
        },
        "reported" : {
            "lights" : { "color": "GREEN"  },
               "engine" : "ON"
        },
        "delta" : {
            "lights" : { "color": "RED"  }
         } },
"version" : 10
}
```



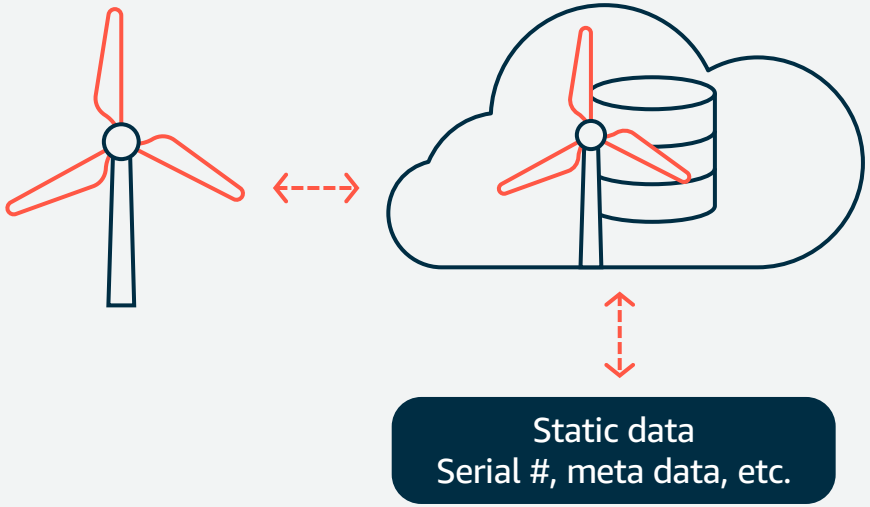
Registry

AWS 서비스에서 쉽게 사용할 수 있도록 디바이스 정의 및 카탈로그

간단한 검색 (예를 들면, 2010에 생성된 디바이스들은?)

여러 기기에서 속성 및 정책을 표준화하기 위해서 *ThingTypes* 정의 (예를 들면, Hyundai 및 Kia 를 Car 라는 ThingType으로 정의함)

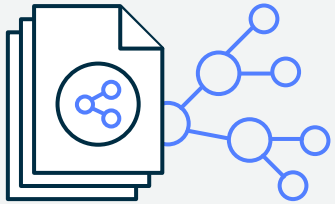
보다 간단한 관리 (작업 실행, 정책 설정 등)를 하기 위하여 *Group을 정의* (예를 들면, 차 안의 sensor들)



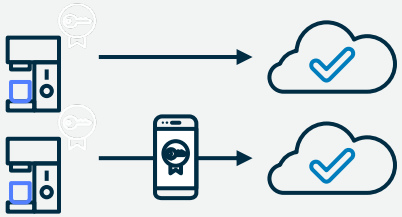
Fleet Provisioning for AWS IoT Core

New Feature!

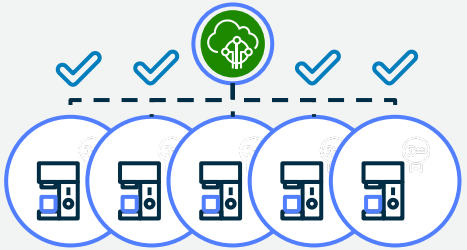
Automates device- and cloud-side configuration and authentication upon a device's first connection to AWS IoT Core.



Define templates for onboarding, and apply them to all devices for secure provisioning at scale.



Create unique identities to establish trust using the updated AWS IoT Device SDK or the AWS Mobile SDK in branded companion apps.



Automatically provision any number of devices upon first connection to AWS IoT Core.

aws

Configurable Endpoints for AWS IoT Core

Beta Feature!

Easily migrate devices to AWS IoT by maintaining different configurations across diverse device fleets with minimal impact on existing devices and applications.

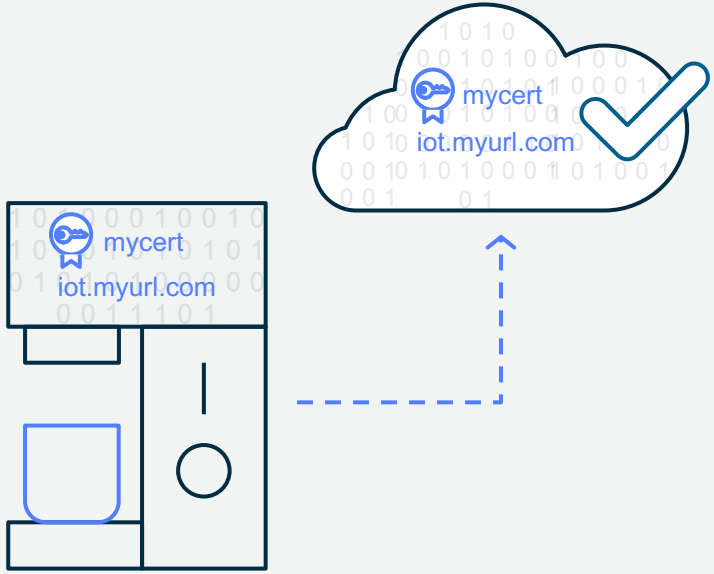
Create multiple IoT endpoints within a single AWS account and set up a unique configuration on each one.

Custom Domains

Continue to use your own domain names and associated server certificates after connecting to AWS IoT Core.

Custom Authorizer for MQTT Connections

Keep your own identity and access management systems for provisioning new device identities.

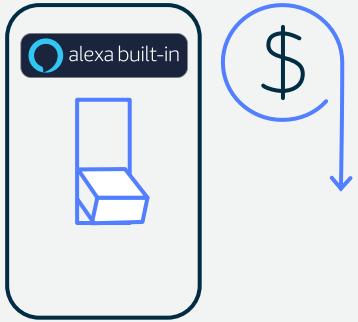


aws

Alexa Voice Service (AVS) Integration for IoT Core

New Feature!

Quickly and cost-effectively go to market with Alexa Built-in capabilities on new categories of products such as light switches, thermostats, and small appliances.



Lowers the cost of integrating Alexa Voice up to 50% by reducing the compute and memory footprint required.

Create new categories of Alexa Built-in products on resource constrained devices (e.g., ARM ‘M' class microcontrollers with <1MB embedded RAM).

Accelerate time to market with certified partner development kits that work with AVS Integration for IoT Core by default.

aws

AWS IoT Greengrass

AWS IoT Greengrass

Local Messages and Triggers

cloud 연결없이 장치 통신 가능

Local Actions

AWS Lambda로 device 프로그래밍 간소화

Data and State Sync

장치를 오프라인으로 작동 & 다시 연결할 때 데이터 동기화

Security

상호 인증 & cloud와 장치 간 인증

Local Resource Access

AWS Lambda 함수는 주어진 장치의 로컬 리소스에 액세스 할 수 있고 & 사용 가능

Machine Learning Inference

로컬에서 ML 추론

Connectors

외부 서비스에 연결하여 edge 장치 확장

Secrets Manager

edge devices 에 보안 자격 증명 배포

Over the Air Updates

AWS IoT Greengrass Core를 쉽게 업데이트

Device software

aws

Message Broker

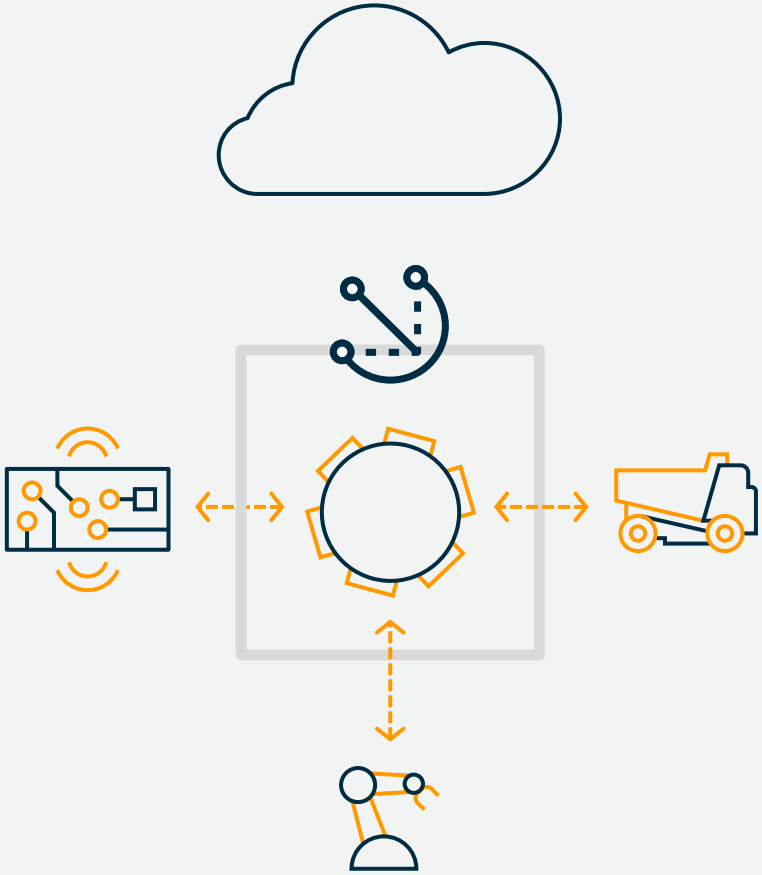
AWS IoT Core 기능을 복제하여 클라우드 연결없이 통신 할 수 있도록 로컬 네트워크의 디바이스 간 메시징을 활성화합니다.

MQTT pub/sub 메시지 패러다임을 edge까지 확장

Cloud에서 작성한 lambda function을 AWS IoT Greengrass Core에 배포하여 이벤트를 트리거하거나 이벤트에 응답

AWS IoT Greengrass Core 에서 IoT Device SDK를 사용하는 디바이스로, offline에서도 명령 및 제어 작업 가능

예를 들어 AWS IoT Greengrass Core는 토양의 낮은 수분을 감지하고, 클라우드에 연결하지 않고도 *smart greenhouse*에 더 많은 물을 분사하는 작업을 트리거 할 수 있습니다.



Lambda - Locally

로컬 AWS Lambda 기능으로 임베디드 소프트웨어 개발 간소화

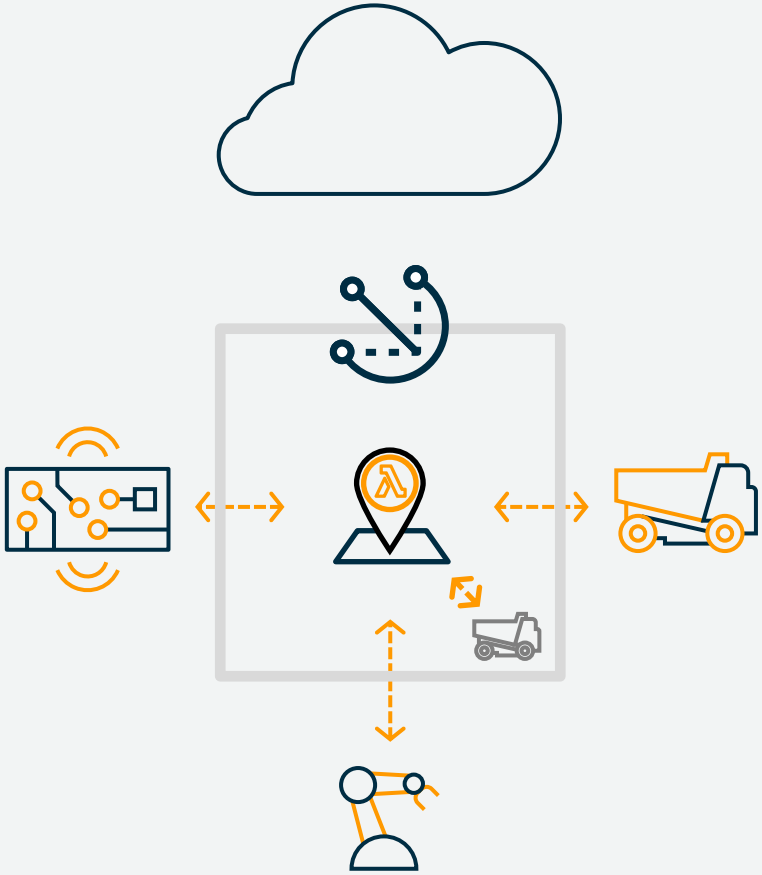
이벤트 기반 람다 함수를 cloud에서 작성하고 device에 배포

Run AWS Lambda functions written in Python 2.7/3.7, Node.js 12.x, Java, C/C++

메시징 및 shadow 업데이트로 AWS Lambda 함수 호출

예를 들어 AWS IoT Greengrass Core는 토양의 낮은 수분을 감지하고, 클라우드에 연결하지 않고도 smart greenhouse에 더 많은 물을 분사하는 작업을 트리거

Containers and Non-Containers



aws

Shadows

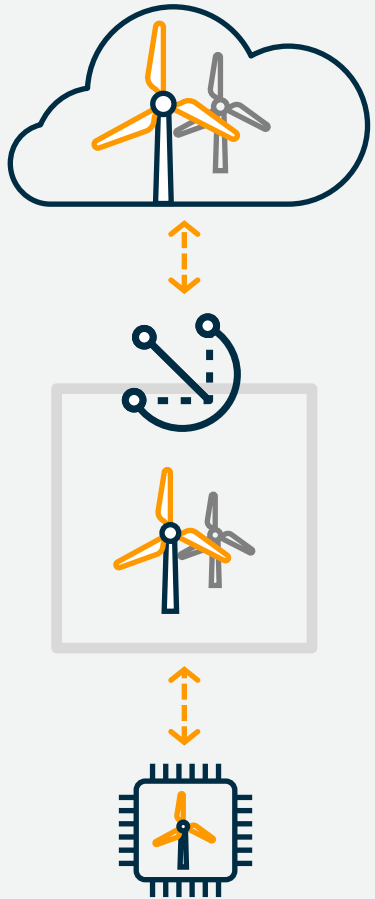
간헐적인 연결 중에 장치를 작동하고 다시 연결될 때 클라우드와 데이터를 동기화

논리적 방식으로 장치의 shadow 상태를 JSON 문서로 정의 할 수 있습니다.

Shadow 상태를 로컬로 cloud에 동기화 할 수 있습니다

AWS IoT Greengrass Core에서 실행되는 AWS Lambda 함수는 MQTT 메시지를 통해 섀도우 상태를 업데이트 할 수 있습니다

Cloud version과 동일한, 이 shadow는 명령 및 제어 작업을 이해하는 데 중요한 개념입니다.



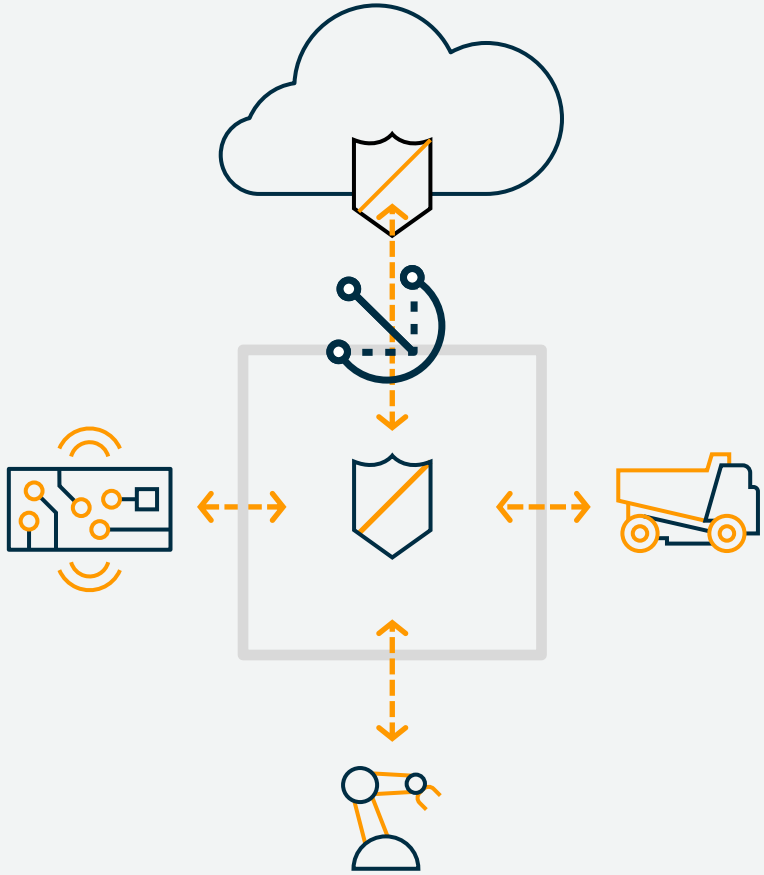
Security

로컬 및 클라우드 통신을 위한 장치 데이터를 인증 및 암호화

로컬과 cloud 모두에서, TLS mutual authentication 지원

장치의 인증서는 클라우드의 SigV4 자격 증명과 연결될 수 있습니다

로컬 Lambda 함수에 사용된 secret이나 private device key를 암호화하기 위한 hardware 기반 root of trust 설정



aws

Local Resource Access

AWS Lambda 함수는 지정된 디바이스의 로컬 리소스에 액세스하고 사용할 수 있습니다.

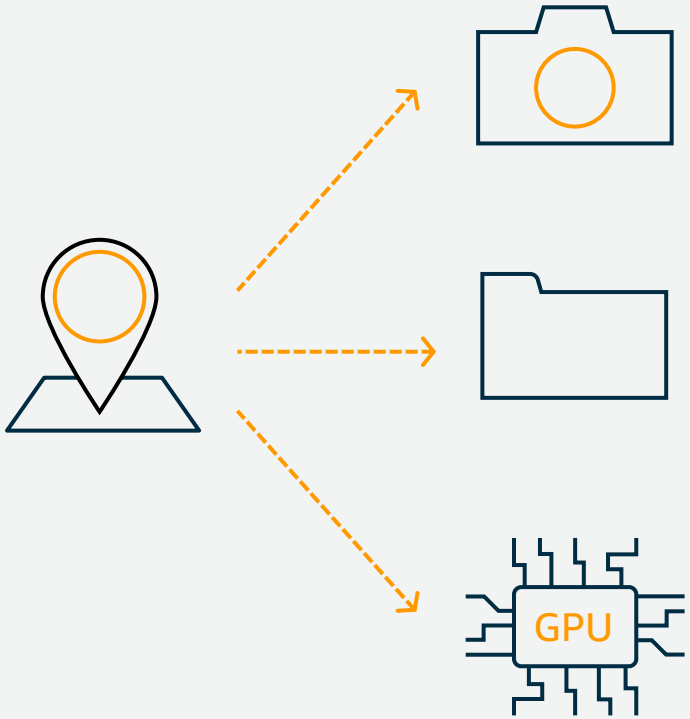
Lambda 함수들이 기기의 로컬 리소스에 액세스 할 수 있도록 허용

센서 및 액추에이터 데이터를 처리하기 위해 GPIO에 액세스 할 수 있습니다.

Lambda 함수들은 운영 체제에서 로컬 파일 시스템을 활용할 수 있습니다

Lambda 함수들은 기계 학습을 위한 하드웨어 가속을 위해 GPU를 사용할 수 있습니다

데이터 저장을 위해 로컬 데이터베이스에 접근



aws

Machine Learning Inference

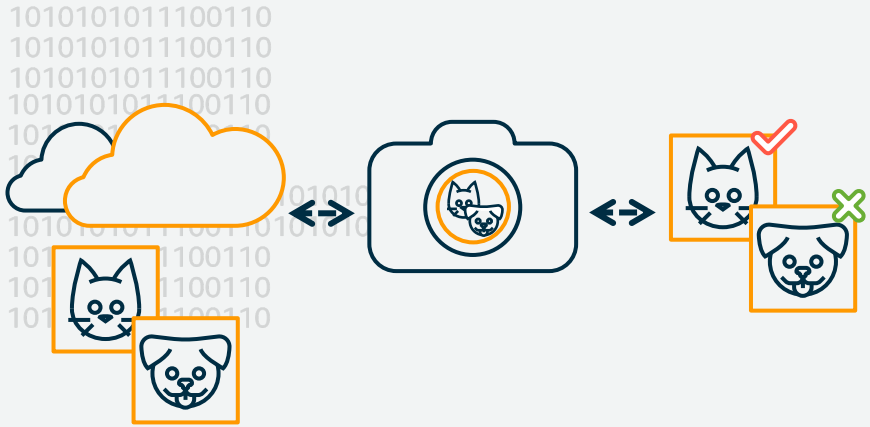
데이터 전송 비용이나 지연 시간 증가없이 로컬에서 ML 추론 수행

Amazon SageMaker 또는 EC2를 사용하는 다른 서비스를 사용하여 클라우드에서 모델 학습

ML 추론은 Apache MXNet 및 TensorFlow와 함께 동작합니다.

훈련된 모델을 device로 전송하고 데이터를 클라우드로 다시 전송하여 모델 정확도를 향상시킵니다.

Amazon SageMaker와의 통합은 model runtime footprint 를100 배 줄이고 추론 성능을 2 배 향상시킵니다



aws

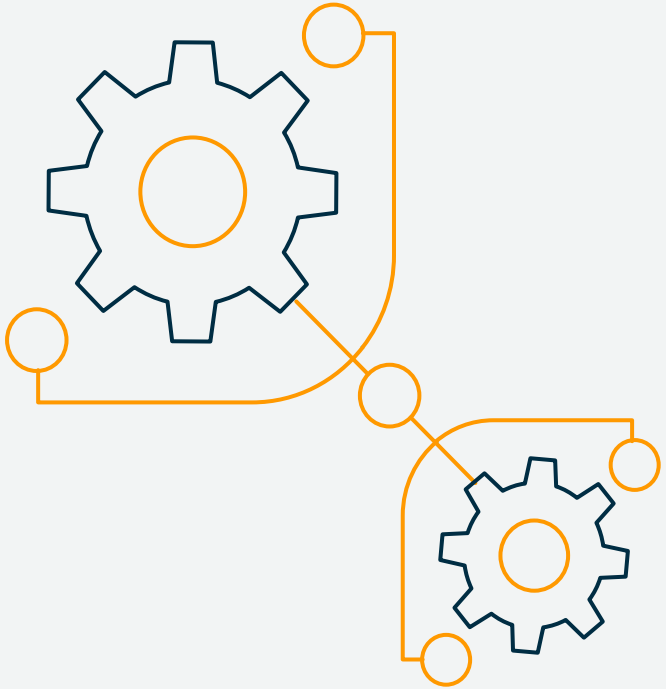
AWS IoT Greengrass Connectors

엣지 디바이스를 타사 서비스, 온 프레미스 소프트웨어 및 AWS 서비스에 빠르게 연결

사전 구축된 기능을 통해 AWS Kinesis Firehose, Amazon CloudWatch 및 Amazon Simple Notification Service와 같은 AWS 클라우드 서비스와 쉽게 연결할 수 있습니다 (SNS)

서비스 응용 프로그램 인 Twilio, ServiceNow 및 기타 소프트웨어와 사전 구축 된 통합

Connector를 빌딩 블록으로 사용하고 복잡한 애플리케이션에 통합



AWS IoT Greengrass Secrets Manager

edge devices에 secrets 배포

장치 자격 증명, 키, 엔드 포인트 및 구성과 같은 secret을 저장, 액세스, rotation 및 관리

Cloud에서 secret을 안전하게 관리하고 edge device에 로컬로 배포

클라우드에서 AWS Secrets Manager를 통해 device의 secret 관리

aws

Update Agent

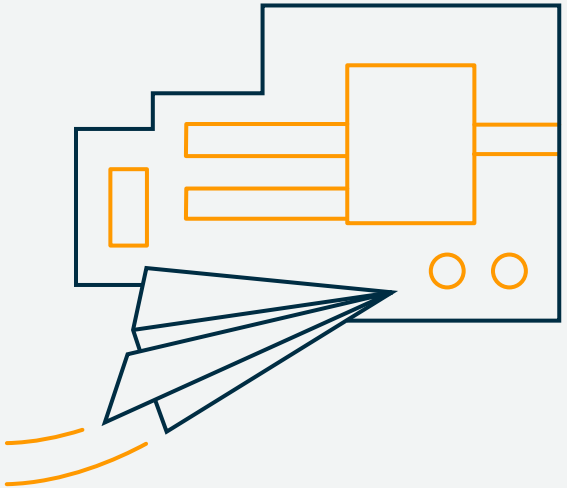
AWS IoT Greengrass 디바이스를 쉽게 업데이트하고 보안 업데이트, 버그 수정 및 기능을 배포

최신 AWS IoT Greengrass 소프트웨어, 보안 업데이트, 버그 수정 및 새로운 기능으로 AWS IoT Greengrass Core 디바이스를 원격으로 업데이트

한 번에 많은 AWS IoT Greengrass Core 장치를 대량으로 업데이트 할 수 있습니다

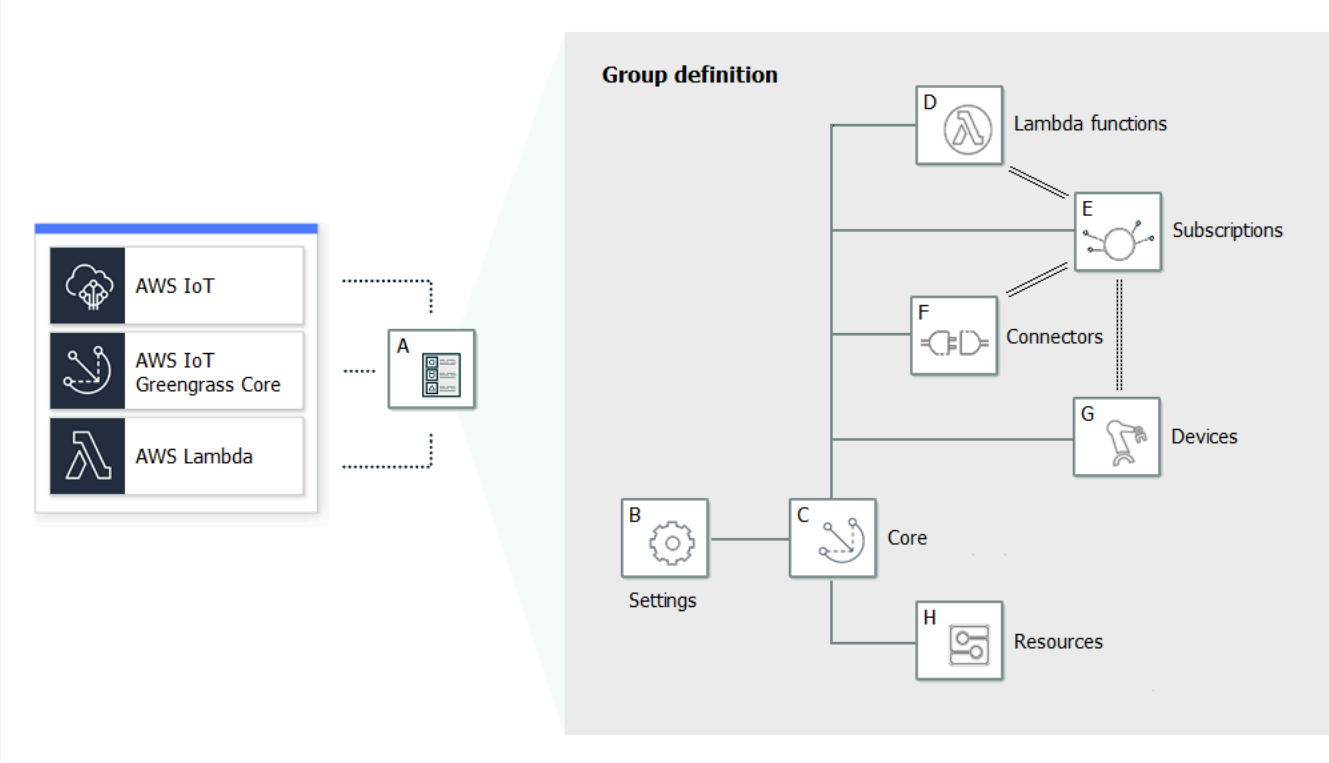
업데이트는 안전합니다. 업데이트가 실패하면 automatic revert가 실행됩니다

AWS IoT 콘솔에서 업데이트 상태를 추적 할 수 있습니다



aws

AWS IoT Greengrass components

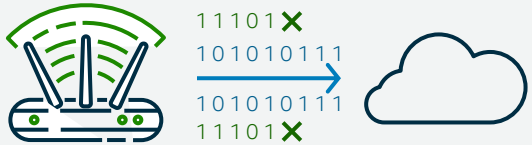


Stream Manager for AWS IoT Greengrass

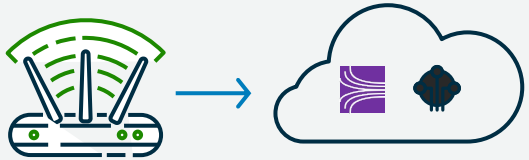
New Feature!

Preconfigure policies for collecting, processing, and exporting high-volume data streams on edge devices

Set policies for how data is processed and managed locally on devices with limited storage and compute.



Prioritize how data is streamed to the cloud with intermittent or limited connectivity.



Stream data from edge devices directly to AWS services such as Amazon Kinesis and AWS IoT Analytics.

aws

Container Support for AWS IoT Greengrass

New Feature!

Deploy containers seamlessly to your edge devices.

Move containers from the cloud to edge devices using AWS IoT Greengrass.

Enables both Docker and AWS Lambda components to operate seamlessly together at the edge.

Use AWS IoT Greengrass Secrets Manager to manage credentials for private container registries.

aws

Q&A

Thank you!