AWS IoT 디바이스 프로비저닝 방식 비교

AWS IoT 디바이스 프로비저닝은 다양한 규모와 보안 수준에 맞춰 IoT 디바이스를 AWS IoT Core에 등록하고 인증서를 배포하는 방식이다.

1. Provisioning with the API

- **기술적 설명**: CreateThing, CreateKeysAndCertificate, AttachPolicy 등의 API를 사용해서 직접 프로비저닝 수행
- **사용 목적**: 개발자가 수동으로 혹은 자동화 스크립트를 만들어 등록할 수 있음
- **예시**: 개발 초기 단계에서 몇 대의 테스트 디바이스를 등록하거나 CI/CD 파이프라인에서 자동 등록할 때

---

2. Single device provisioning

- **기술적 설명**: 개별 디바이스를 AWS IoT 콘솔이나 API를 통해 하나씩 수동 등록
- **사용 목적**: 수량이 적거나 테스트용 디바이스일 때
- **예시**: 실험실에서 새로운 센서 장비를 등록할 때

---

3. Bulk provisioning

- **기술적 설명**: JSON/CSV 파일로 수천 개의 디바이스 정보를 업로드해 한 번에 등록
- **사용 목적**: 대량 디바이스를 한꺼번에 등록할 때
- **예시**: 공장에서 수출 전 디바이스를 사전 등록하고 인증서를 미리 발급받을 때

---

4. Fleet provisioning

- **기술적 설명**: 디바이스가 처음 연결될 때 등록 템플릿과 보안 토큰을 이용해 자동 등록 및 인증서 생성

- **구성 요소**: 프로비저닝 템플릿, 디바이스 CSR/인증서, Claim 인증서 사용
- **사용 목적**: 수백~수천 개의 디바이스가 현장에 배치될 때 자동 등록되게 하기 위함
- **예시**: 전국에 설치되는 스마트미터기가 처음 전원 켜질 때 자동으로 AWS에 등록

---

5. Bring Your Own CA (BYOCA)

- **기술적 설명**: AWS에 사용자 고유의 CA를 등록하고, 이 CA로 서명된 디바이스 인증서를 기반으로 인증 및 등록
- **사용 목적**: 자체 PKI 체계를 이미 가지고 있는 기업이 기존 인증서를 계속 활용하고자 할 때
- **예시**: 의료기기 제조사가 자체 발급한 인증서를 이용해 병원에 납품하는 디바이스를 등록

---

6. Just-in-Time Provisioning (JITP)

- **기술적 설명**: 디바이스가 연결될 때 CA 인증서로 서명된 클라이언트 인증서를 AWS가 확인하고, 사전 설정된 프로비저닝 템플릿에 따라 자동 등록
- **차이점**: 등록 즉시 Thing, 인증서, 정책이 생성되며 사용자 개입 없음
- **필수 조건**: CA 등록 + 템플릿 설정
- **예시**: 배터리 내장형 센서가 배송 중에도 전원이 꺼져 있다가, 현장에서 처음 연결될 때 자동으로 등록됨

---

7. Just-in-Time Registration (JITR)

- **기술적 설명**: JITP와 비슷하지만, 자동 등록 전에 수동 승인을 필요로 함 (등록 후 수동 승인)
- **사용 목적**: 보안상 모든 디바이스 등록을 관리자가 검토한 후 진행하고 싶을 때
- **예시**: 정부기관 납품용 IoT 장비로, 보안상 등록 승인이 관리자 검토 후에 이뤄져야 하는 경우

- **차이 요약표:**

방식	등록 시점	자동화	인증서 제공 방식	적합 환경
API	언제든지	낮음	직접 발급	개발/테스트
Single	사전	낮음	직접 발급	소량 등록
Bulk	사전	중간	일괄 발급	대량 사전 등록
Fleet	최초 연결 시	높음	자동 발급	자동화 배포
BYOCA	연결 시	중간~높음	외부 인증서	자체 CA 있는 환경
JITP	최초 연결 시	높음	외부 인증서	자동 등록 필요
JITR	최초 연결 시	중간	외부 인증서	수동 승인 필요 환경

CA와 CSR 설명

1. CA (Certificate Authority, 인증 기관)

- **정의**: 인증서를 발급해주는 신뢰할 수 있는 기관
- **역할**: 디바이스나 서버가 사용하는 인증서가 신뢰할 수 있는지를 보장해줌
- **예시**:
    - 공인 CA: DigiCert, Let's Encrypt, GlobalSign 등
    - 사설 CA (Private CA): 조직 내부에서 직접 구축한 CA, AWS Private CA도 가능

실제 AWS IoT에서는?

- AWS에 CA를 등록해 놓으면, 이 CA로 서명된 인증서를 가진 디바이스는 **자동으로 신뢰**되고 등록될 수 있다.

- 이를 통해 **Just-in-Time Provisioning (JITP)**, **Fleet Provisioning**, **BYOCA** 등이 가능하다.

---

2. CSR (Certificate Signing Request, 인증서 서명 요청)

- **정의**: 인증서를 발급받기 위해 CA에 보내는 요청 파일
- **포함 정보**: 공개키(public key), 조직 정보, 도메인 이름 등
- **형식**: 일반적으로 .csr 파일이며, PEM(Base64) 형식

CSR 생성 과정:

1. 디바이스나 서버가 공개키/개인키 쌍을 생성
2. 공개키와 정보로 CSR 파일 생성
3. 이 CSR을 CA에 보내 인증서를 발급 요청

예시 (OpenSSL 사용):

openssl req -new -newkey rsa:2048 -nodes -keyout device.key -out device.csr

---

정리

용어	의미	주요 역할
CA	인증서 발급자	디바이스 인증서가 신뢰 가능한지 판단
CSR	인증서 발급 요청서	공개키와 메타정보를 포함해 CA에 인증서 발급 요청

디바이스 프로비저닝에서 이 둘은 다음과 같은 관계를 가진다:

디바이스가 CSR을 생성 → 등록된 CA에 제출 → CA가 서명된 인증서 발급 → AWS IoT에 등록