ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Bulut Bilişim Ve Uygulamaları

Proje 7: IoT ve Akıllı Şehir Uygulaması

Emrah Oğuz Ordu
21290298

Video Linki: https://youtu.be/vbN5od3n2Z0?si=jligzsCqDopFY4rL
GitHub Uygulama Linki: https://github.com/oguzordu/IoT-ve-Akilli-Sehir-Uygulamasi

Bu projede, akıllı bir şehirdeki sokak lambası sensöründen gelen verilerin toplanması, bulutta işlenmesi ve depolanması simüle edilmiştir. Amaç, AWS bulut platformunun IoT, serverless işlem (Lambda) ve NoSQL veritabanı (DynamoDB) servislerini kullanarak basit ama işlevsel bir IoT veri akış hattı kurmaktır. Python ile geliştirilen bir simülatör, MQTT üzerinden veri gönderirken, AWS servisleri bu veriyi alıp işleyerek DynamoDB'ye kaydeder.

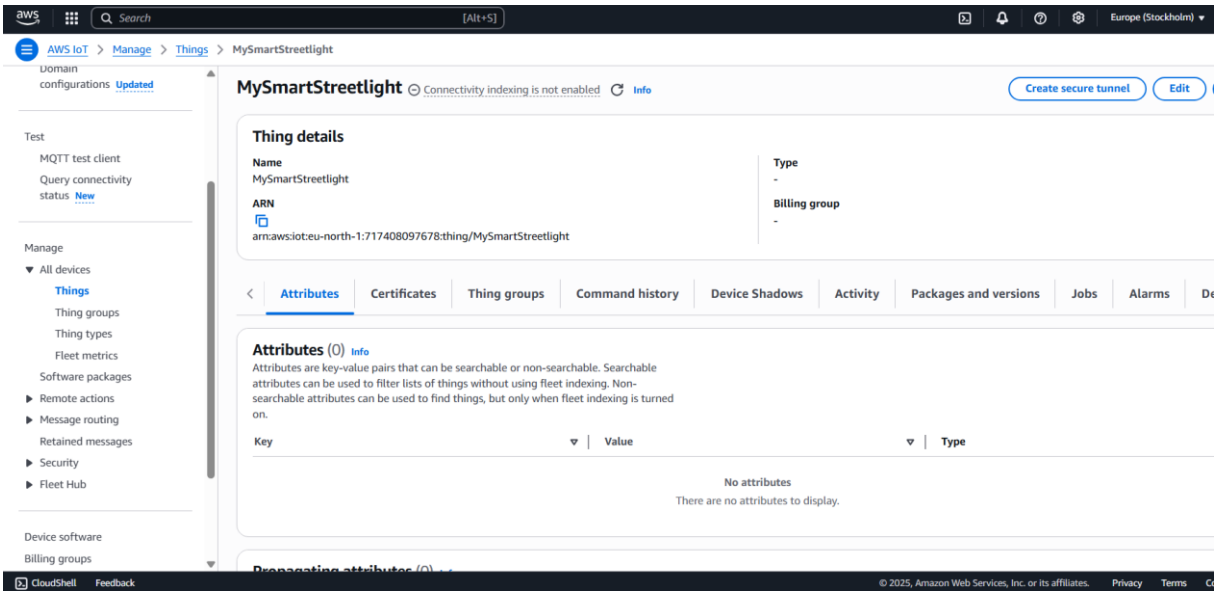
2. Kullanılan Teknolojiler ve AWS Servisleri

- **Programlama Dili:** Python (Cihaz simülatörü ve Lambda fonksiyonu için)
- **Veri İletim Protokolü:** MQTT
- **Bulut Platformu:** Amazon Web Services (AWS)
- **Temel AWS Servisleri:**
- **AWS IoT Core:** Cihaz bağlantılarını, kimlik doğrulamasını, yetkilendirmeyi ve mesaj yönlendirmeyi yönetir.
- **AWS Lambda:** IoT Core'dan gelen verileri işlemek için sunucusuz (serverless) işlem gücü sağlar.
- **AWS DynamoDB:** İşlenen verilerin depolandığı ölçeklenebilir NoSQL veritabanıdır.
- **AWS IAM:** Servislerin birbirleriyle güvenli bir şekilde etkileşimde bulunması için gerekli izinleri yönetir.
- **AWS CloudWatch:** Lambda fonksiyonlarının loglarını izlemek ve hata ayıklamak için kullanılır.

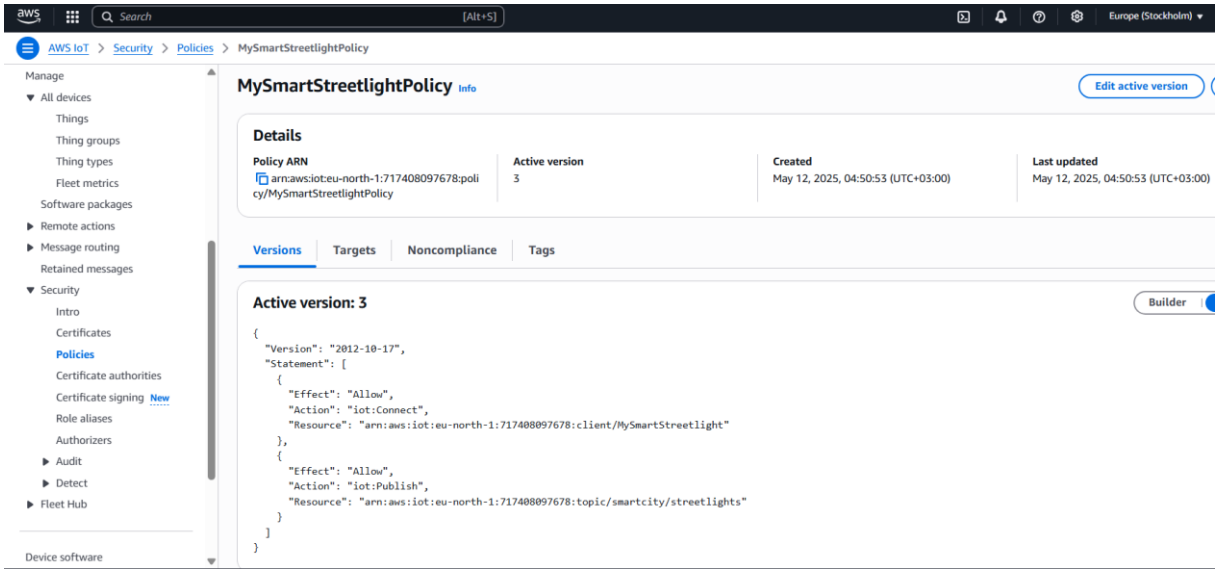
3. Sistem Mimarisi ve Veri Akışı

1. **simulated_device.py (Python Script):** Lokal makinede çalışarak periyodik olarak sokak lambası durumu (ON/OFF) ve zaman damgası içeren JSON formatında veri üretir.
2. **MQTT ile AWS IoT Core'a Veri Gönderimi:** Script, AWS IoT Device SDK kullanarak ürettiği veriyi MQTT üzerinden AWS IoT Core'daki smartcity/streetlights konusuna (topic) yayınlar.
3. **AWS IoT Core Yapılandırması:**

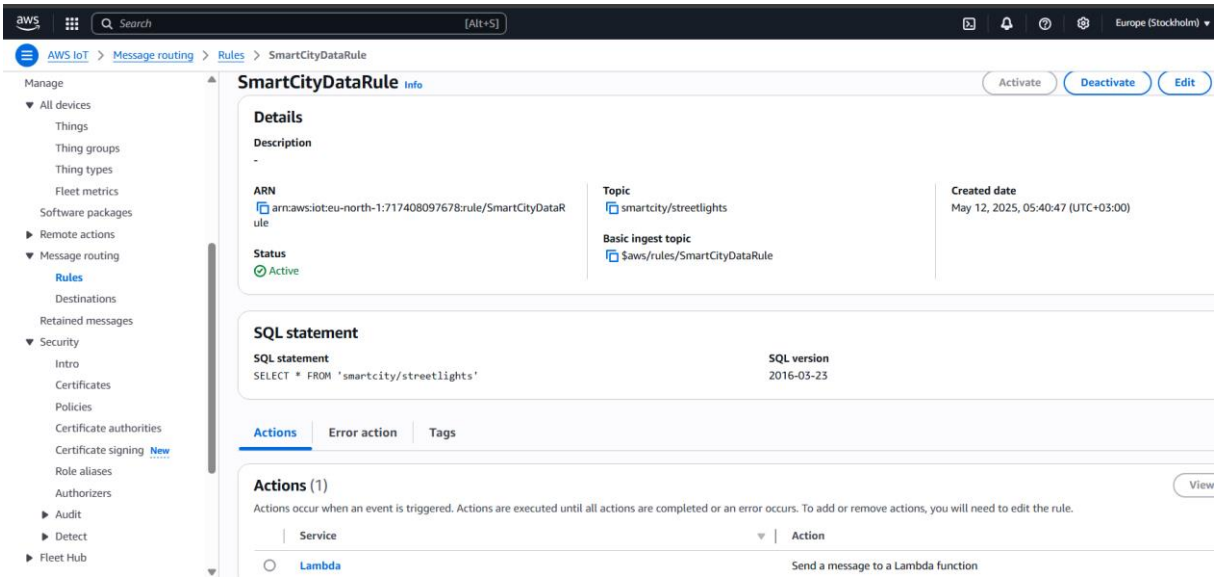
- **Thing (Cihaz):** MySmartStreetlight adında bir "Thing" oluşturularak script temsil edilir. Gerekli sertifikalar (cihaz sertifikası, özel anahtar, root CA) bu Thing ile ilişkilendirilir.



- **Policy (Politika):** Cihazın IoT Core'a bağlanmasına ve belirtilen konuya yayın yapmasına izin veren bir IAM politikası oluşturulur ve sertifikaya eklenir.

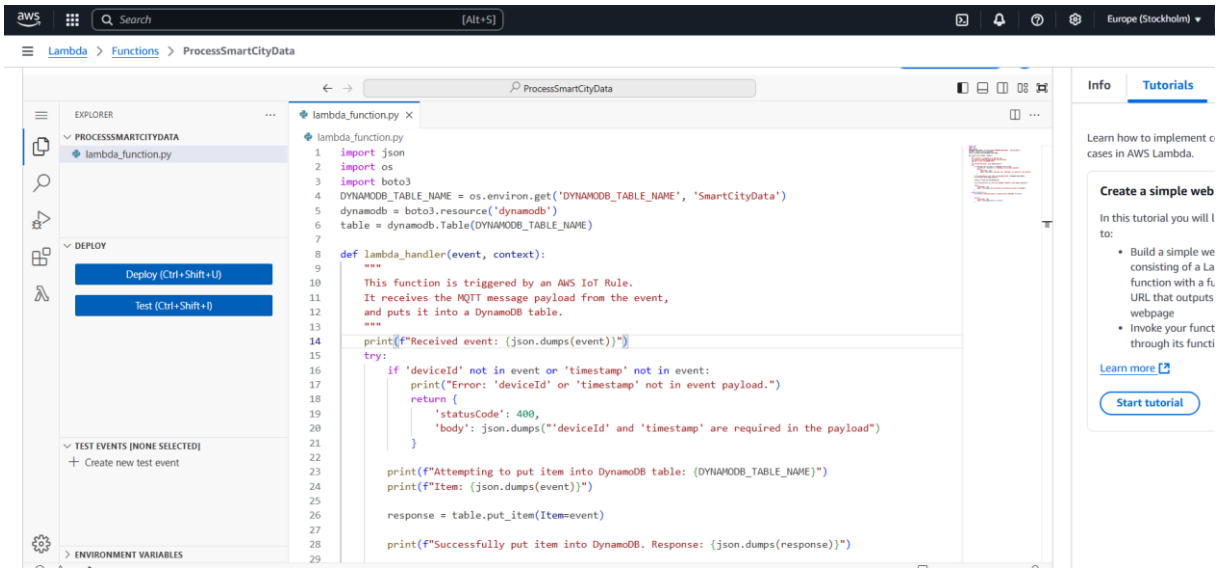


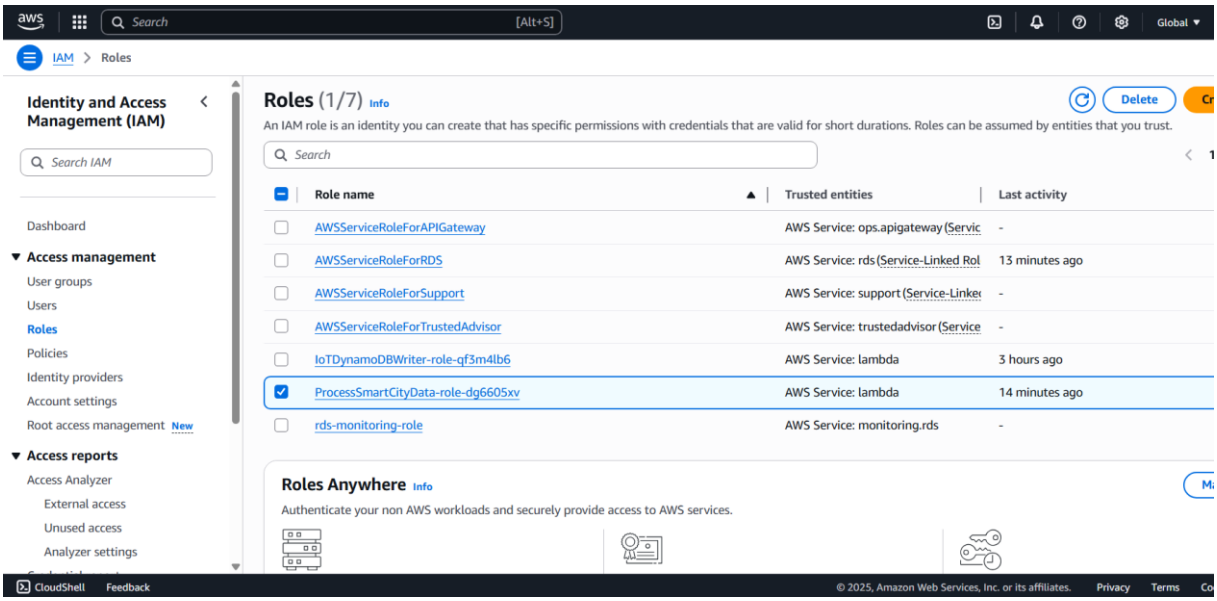
AWS IoT Rule (Kural): smartcity/streetlights konusuna gelen mesajları dinler ve bu mesajları ProcessSmartCityData Lambda fonksiyonuna iletir. SQL sorgusu: SELECT * FROM 'smartcity/streetlights'.



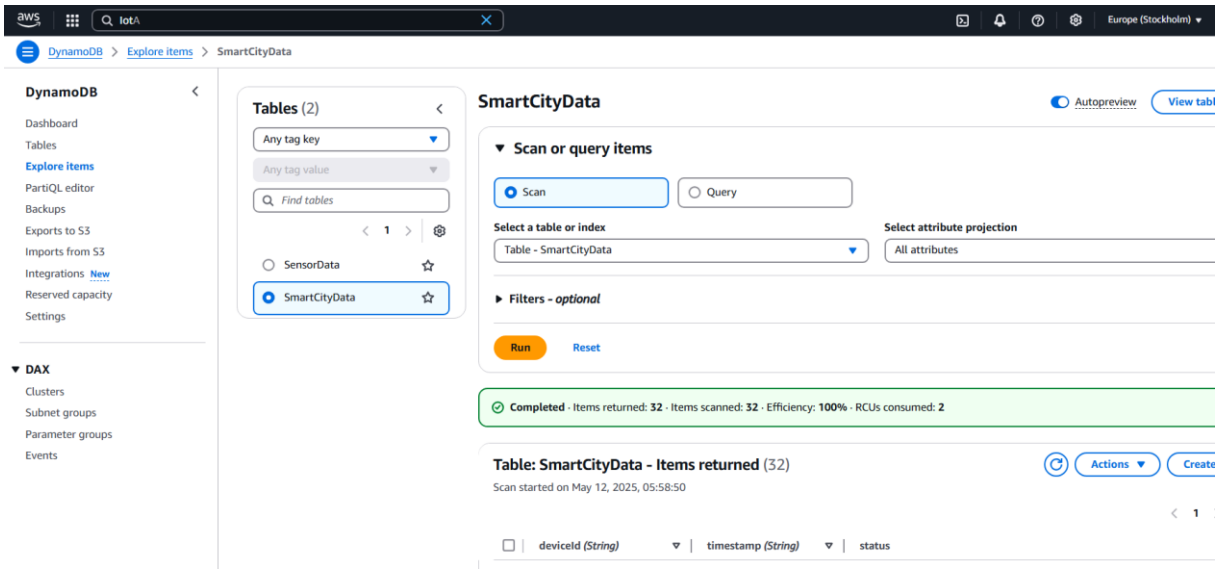
AWS Lambda (ProcessSmartCityData Fonksiyonu):

- IoT Rule tarafından tetiklenir ve gelen JSON verisini (event) alır.
- Bu veriyi, boto3 kütüphanesi aracılığıyla SmartCityData DynamoDB tablosuna yazar.
- Fonksiyonun çalışması ve olası hatalar CloudWatch Logs'a kaydedilir.



- 

- **AWS DynamoDB (SmartCityData Tablosu):**
- Lambda tarafından gönderilen sokak lambası verilerini depolar. Tablo yapısı: deviceId (String, Partition Key) ve timestamp (String, Sort Key).



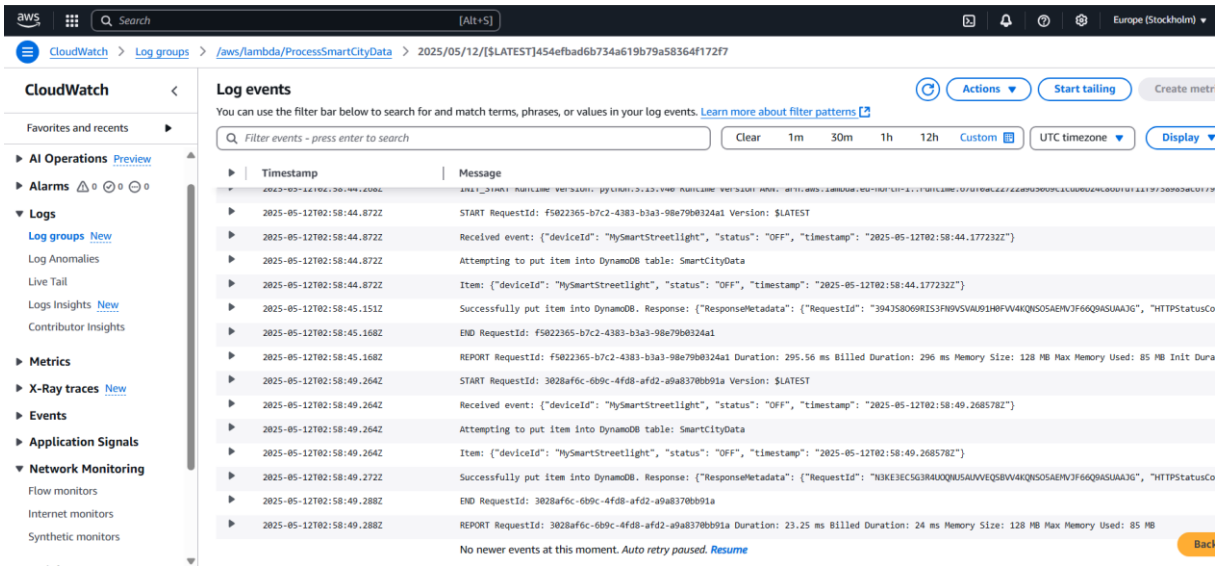
4. Uygulama Adımları ve Test Sonuçları

- **Cihaz Simülatörünün Çalıştırılması:** simulated_device.py scripti çalıştırıldı.

```
Connected to AWS IoT Core!
Simüle Edilmiş Cihaz Başlatıldı. Veri AWS IoT Core'a gönderiliyor...
CTRL+C ile durdurabilirsiniz.
Publishing message to topic 'smartcity/streetlights': {"deviceId": "MySmartStreetlight", "status": "ON", "timestamp": "2025-05-12T02:43:55.855905Z"}
Message 1 published successfully.
Publishing message to topic 'smartcity/streetlights': {"deviceId": "MySmartStreetlight", "status": "OFF", "timestamp": "2025-05-12T02:44:00.976992Z"}
Message 2 published successfully.
```

- **Mesajların IoT Core'da Gözlemlenmesi:** AWS IoT Core MQTT Test İstemcisi üzerinden smartcity/streetlights konusuna abone olunarak gelen mesajlar anlık olarak izlendi.

- **Lambda Fonksiyonunun Loglarının İncelenmesi:** CloudWatch Logs üzerinden Lambda fonksiyonunun tetiklendiği, veriyi aldığı ve DynamoDB'ye başarıyla yazdığına dair loglar kontrol edildi.



- **DynamoDB Verilerinin Doğrulanması:** SmartCityData DynamoDB tablosu kontrol edilerek Lambda tarafından yazılan verilerin (deviceId, status, timestamp) doğru bir şekilde depolandığı teyit edildi.

5. Sonuç ve Değerlendirme Bu proje ile AWS bulut servisleri kullanılarak basit bir IoT veri toplama, işleme ve depolama hattı başarıyla kurulmuştur. Simüle edilen cihazdan gönderilen veriler, AWS IoT Core üzerinden alınmış, bir Lambda fonksiyonu ile işlenerek DynamoDB'ye kaydedilmiştir. Proje, bulut teknolojilerinin IoT uygulamalarına entegrasyonunun temel prensiplerini göstermektedir.