# ANKARA ÜNİVERSİTESİ

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**FİNAL PROJESİ RAPORU**

**(BLM3522-A) BULUT BİLİŞİM VE UYGULAMALARI**

**Perizat SAGYNBEKOVA (21290895)**

**Derda SİNA GÜNAY (20291274)**

**Rüstem TUKHBETOV (21291001)**

**GitHub (**<https://github.com/Impasbaa/BulutBilisim>**)**

**YouTube (**<https://youtu.be/JVRNKY-bnjI?si=hqfi95IjIc2smjRv>**)**

**25.06.2025**

**PROJE 7: IOT VE AKILLI ŞEHİR UYGULAMASI**

**Backend Dili:** Python (IoT cihazları ile veri gönderme/simülasyon için)

**Bulut Platformu:** AWS (Amazon Web Services)

**Problem:** Python ile AWS Tabanlı Akıllı Aydınlatma Uygulaması

**AMAÇ:** Akıllı şehir uygulamaları, şehir yaşamını daha sürdürülebilir, verimli ve kullanıcı dostu hale getirmeyi amaçlayan çözümlerdir. Bu projede, akıllı aydınlatma sistemi ele alınmıştır. Python ile geliştirilen IoT tabanlı Akıllı Aydınlatma Sistemi, hareket sensörleri ve ışık şiddeti sensörlerinden gelen verileri AWS IoT Core üzerinden işleyerek, sokak lambalarının otomatik olarak açılıp kapanmasını ve parlaklık ayarını optimize edecektir. Amaç; sokak lambalarının enerji verimliliğini artırmak, gereksiz çalışmayı engellemek ve uzaktan yönetilebilir hale getirmektir.

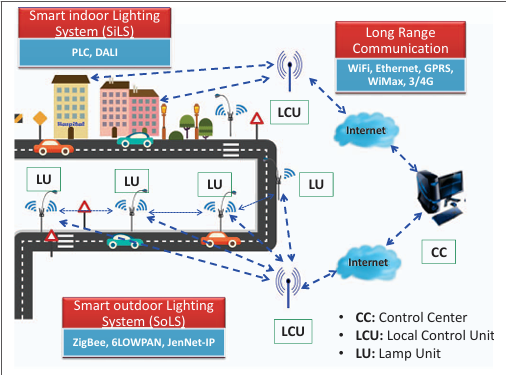
**GİRİŞ**

Günümüzde kentleşmenin hızla artması, şehirlerdeki yaşam kalitesinin iyileştirilmesi için daha akıllı ve verimli teknolojilere olan ihtiyacı da artırmaktadır. Bu bağlamda, akıllı şehir kavramı; enerji, ulaşım, sağlık gibi temel altyapıların bilgi ve iletişim teknolojileri (ICT) ile entegre edilmesini hedefleyen yenilikçi bir yaklaşımdır. Nesnelerin İnterneti (IoT) ise bu dönüşümün merkezinde yer almakta, farklı cihazların birbiriyle bağlantılı ve otonom şekilde çalışmasını mümkün kılmaktadır.

Şehir altyapısı içerisinde önemli miktarda enerji tüketen sistemlerden biri olan aydınlatma, akıllı şehir uygulamalarının öncelikli hedeflerinden biridir. Yapılan araştırmalara göre, sadece kamusal alanlardaki aydınlatma sistemleri bile şehirlerin toplam enerji tüketiminin yaklaşık %10’unu oluşturmaktadır. Bu enerji israfının önüne geçmek amacıyla geliştirilen Akıllı Aydınlatma Sistemleri (Smart Lighting Systems – SLS), hareket ve ışık sensörleri gibi bileşenlerle desteklenerek, aydınlatma yoğunluğunu çevresel koşullara göre otomatik olarak ayarlayabilmektedir

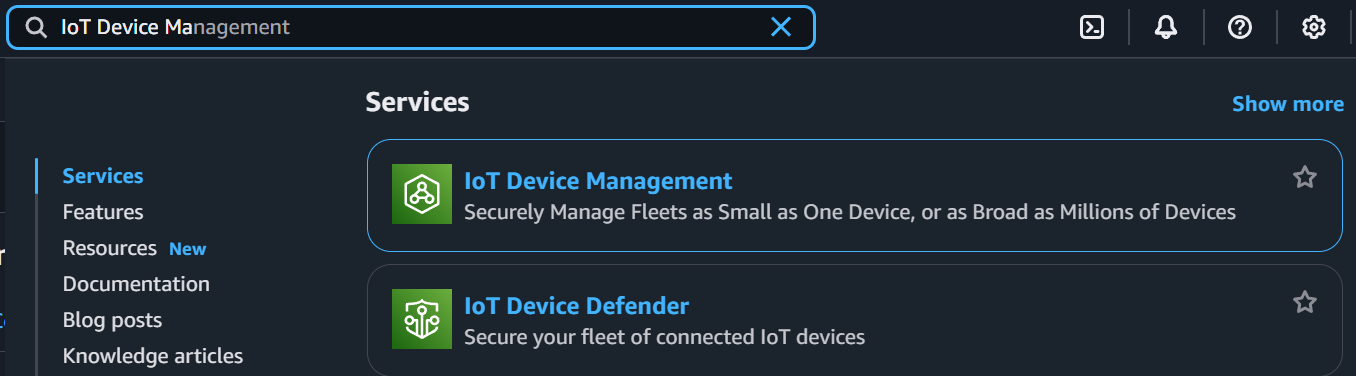
Bu projede, AWS platformu kullanılarak Python dili ile geliştirilen simülatif bir IoT tabanlı akıllı aydınlatma uygulaması sunulmaktadır. Uygulamada, bir akıllı lambadan (simülasyon cihazı) toplanan sensör verileri MQTT protokolü üzerinden AWS IoT Core'a iletilmekte, oradan da Lambda fonksiyonu aracılığıyla işlenip DynamoDB veritabanına kaydedilmektedir. Bu yapı sayesinde veriler hem analiz edilebilmekte hem de gerçek zamanlı olarak izlenebilmektedir.

*Akıllı şehirlerde akıllı aydınlatma sistemleri [1]:*

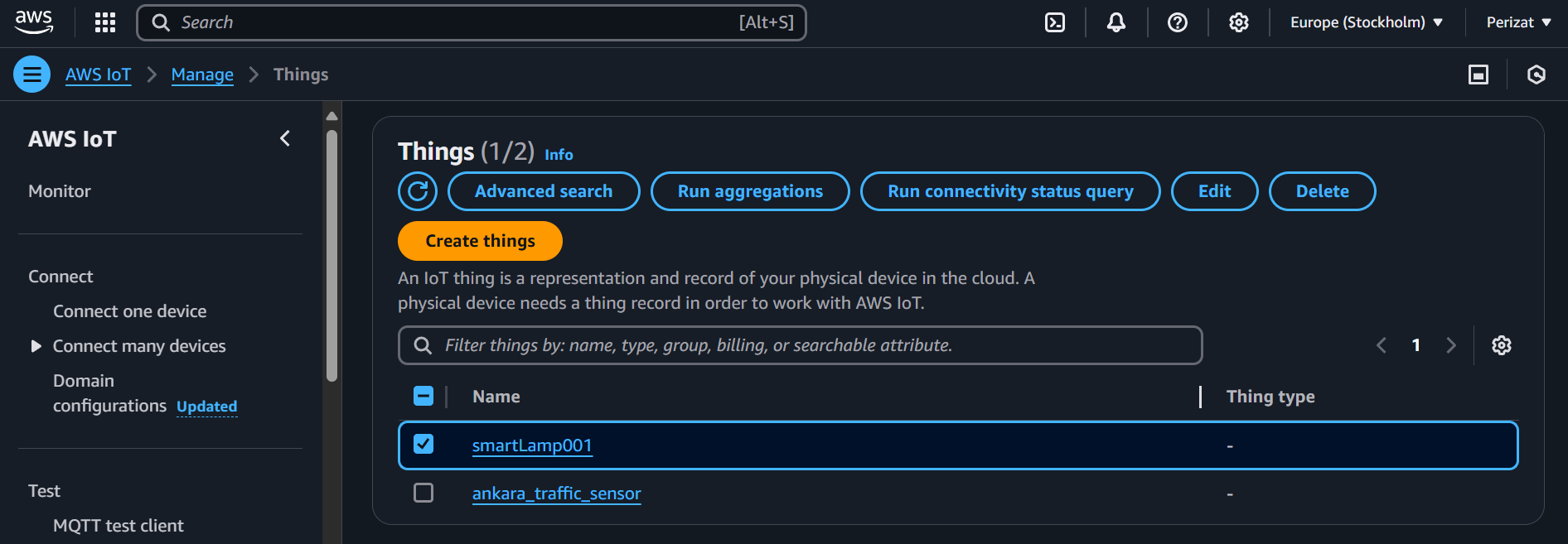
****

**1. Her Sokak Lambası Bir Iot Cihazı Gibi Düşünülür**

İnsan varlığını algılayan hareket sensörü (PIR), ortam aydınlığını ölçen ışık şiddeti sensörü (LDR) ve aydınlatma ayarı yapan LED Strip (PWM Control) gibi gerçek fiziksel IoT cihazları (Raspberry Pi/ESP32) elimizde bulunmadığından bu projede AWS IoT Core ile yazılımsal sanal cihaz simülasyonu yapıyor olacağız.



*smartLamp001 adında bir nesne oluşturuyoruz:*



Cihaz oluştururken gerekli **.pem** dosyalarını (private key, public key, certificates) indiriyoruz.

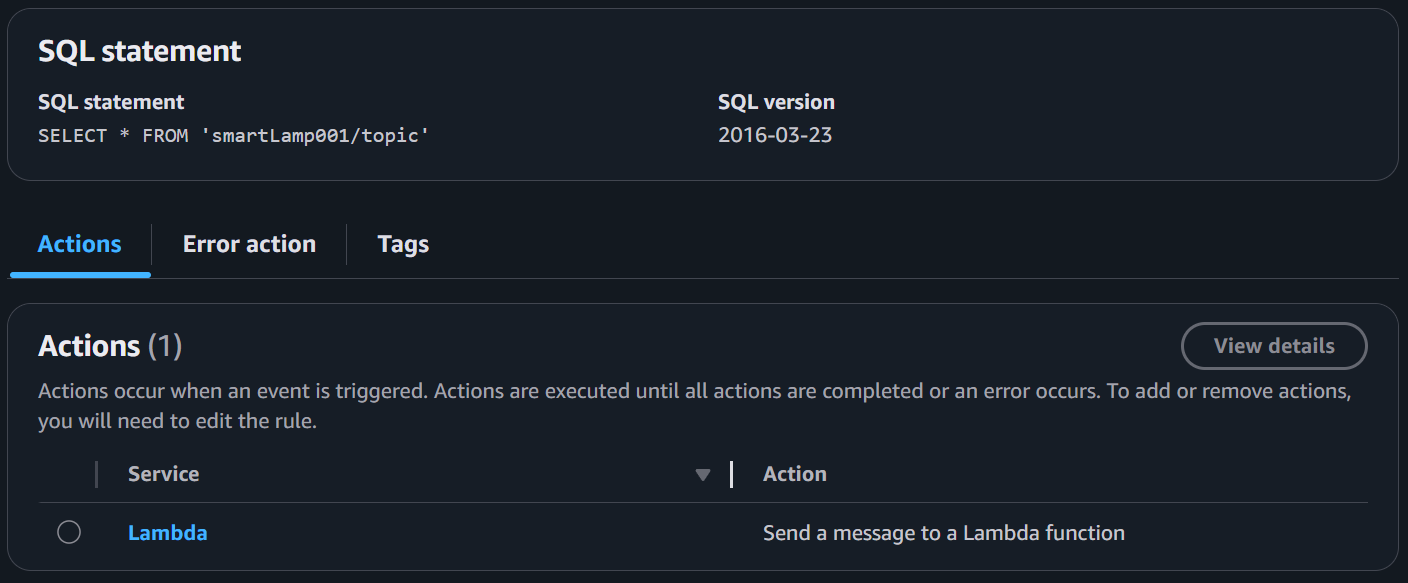
**2. Lambalar, Işık Seviyesi (Lux), Hareket Algılama ve Enerji Tüketimi Gibi Verileri AWS Iot Core’a Gönderir**

*AWS IoT Core yapılandırması yaparken IoT cihazlarından gelen verileri (MQTT mesajları) alıp, başka bir yere (Lambda, DynamoDB, S3, vs.) aktarma yapmak için yeni bir Rule oluşturuyoruz:*

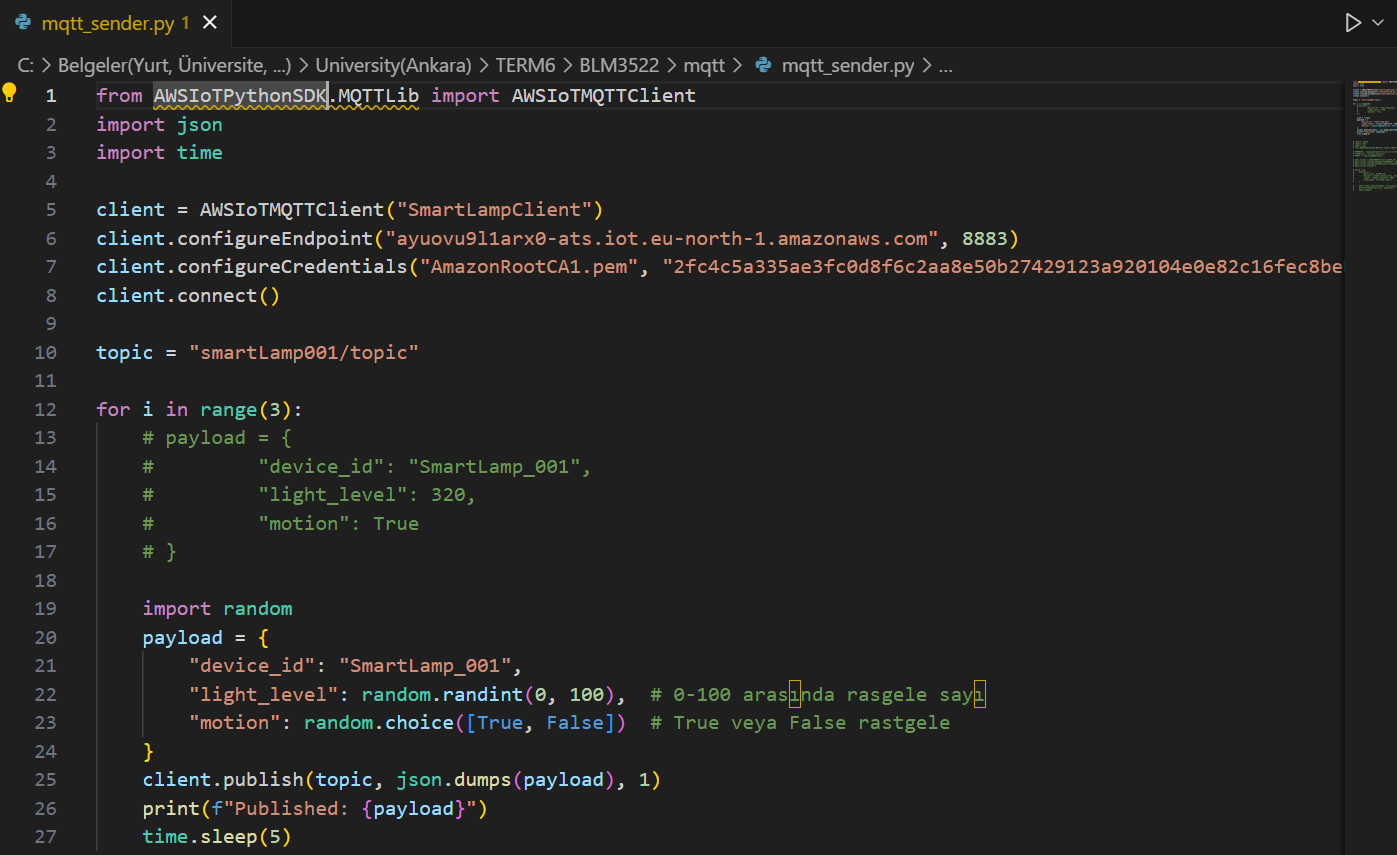
metin, yazılım, multimedya yazılımı, grafik yazılımı içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

*Aktarma action’ı (eylem) olarak lambda fonksiyonunu seçiyoruz ve gerekli SQL komutlarını gerekli yere yazıyoruz:*

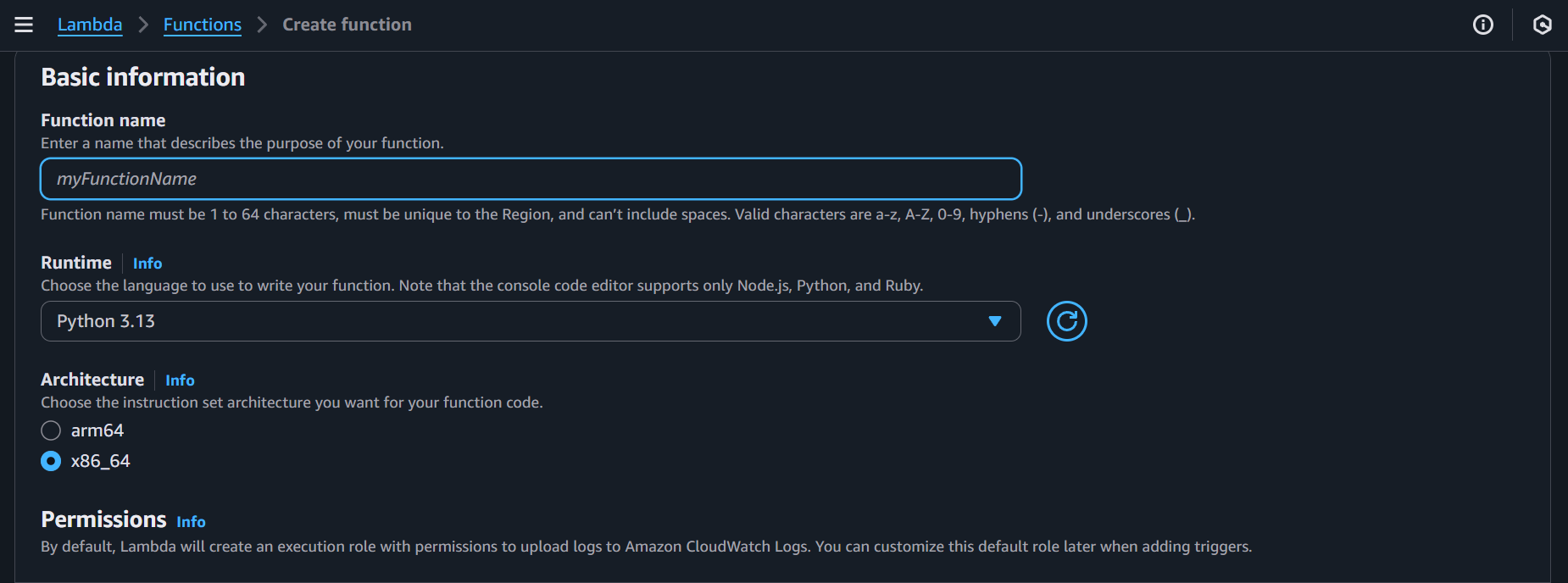


*Python ile MQTT veri gönderimi (simülasyon) için VSCode’da bir Python dosyası oluşturuyoruz ve gerekli MQTT kodlarını yazıyoruz:*



Ortamın ne kadar aydınlık olduğunu gösteren light\_level (ışık seviyesi) Lux (lüks) birimi ölçümünü kullanmaktadır.

*Lambda fonksiyonuna isim vererek ve Python dilini seçerek yeni bir function oluşturuyoruz:*



**3. Veriler AWS Lambda Aracılığıyla Analiz Edilir**

Eğer hava karanlık ve hareket algılandıysa → lamba açılır.

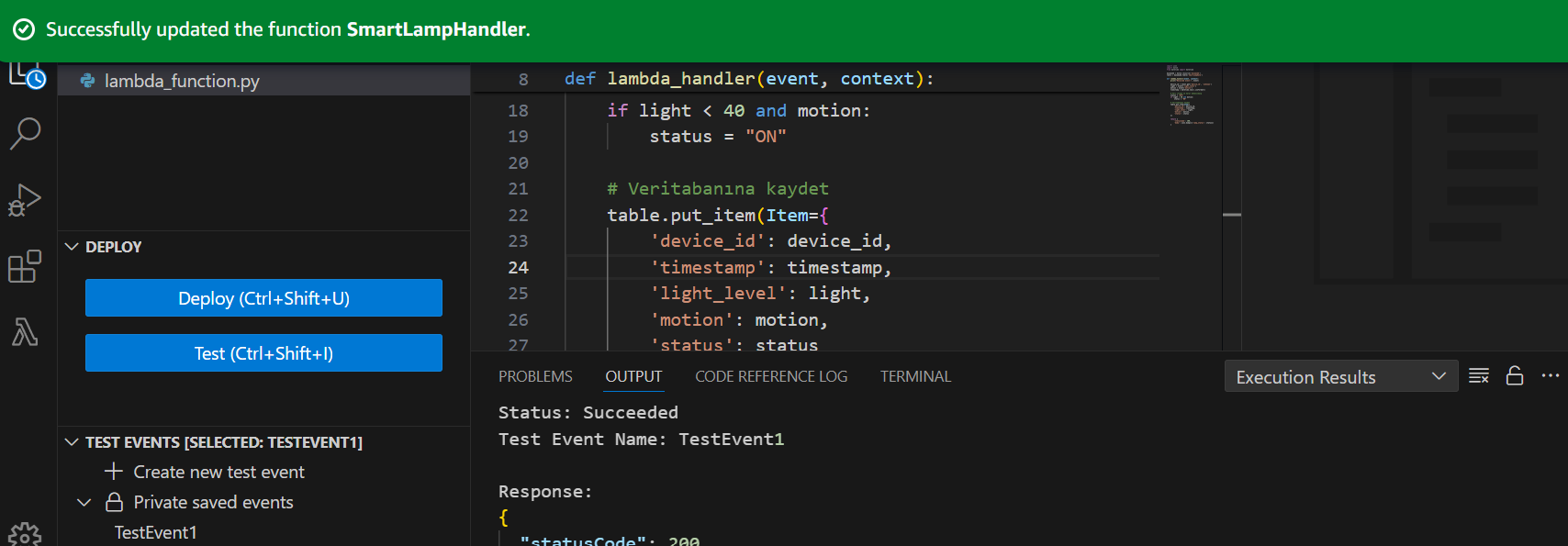
Eğer gündüzse veya hareket yoksa → lamba kapatılır.

*Akıllı aydınlatma sisteminizin beyni olarak çalışan Lambda fonksiyonu, sensör verilerini analiz ederek aydınlatma kurallarını uygular ve DynamoDB'ye alınan/işlenen verileri kaydeder:*

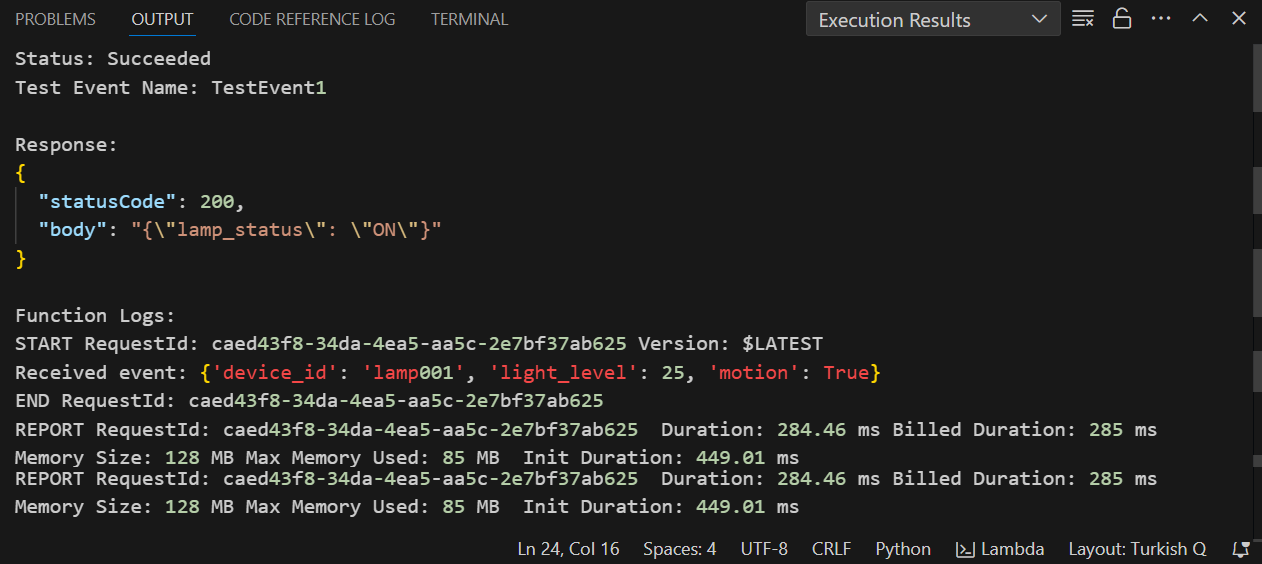
metin, ekran görüntüsü, yazılım içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

*Deploy düğmesine basarak yaptığımız değişiklikleri kaydederiz ve Test düğmesine basarak da kodun doğru çalışıp çalışmadığını görebiliriz:*

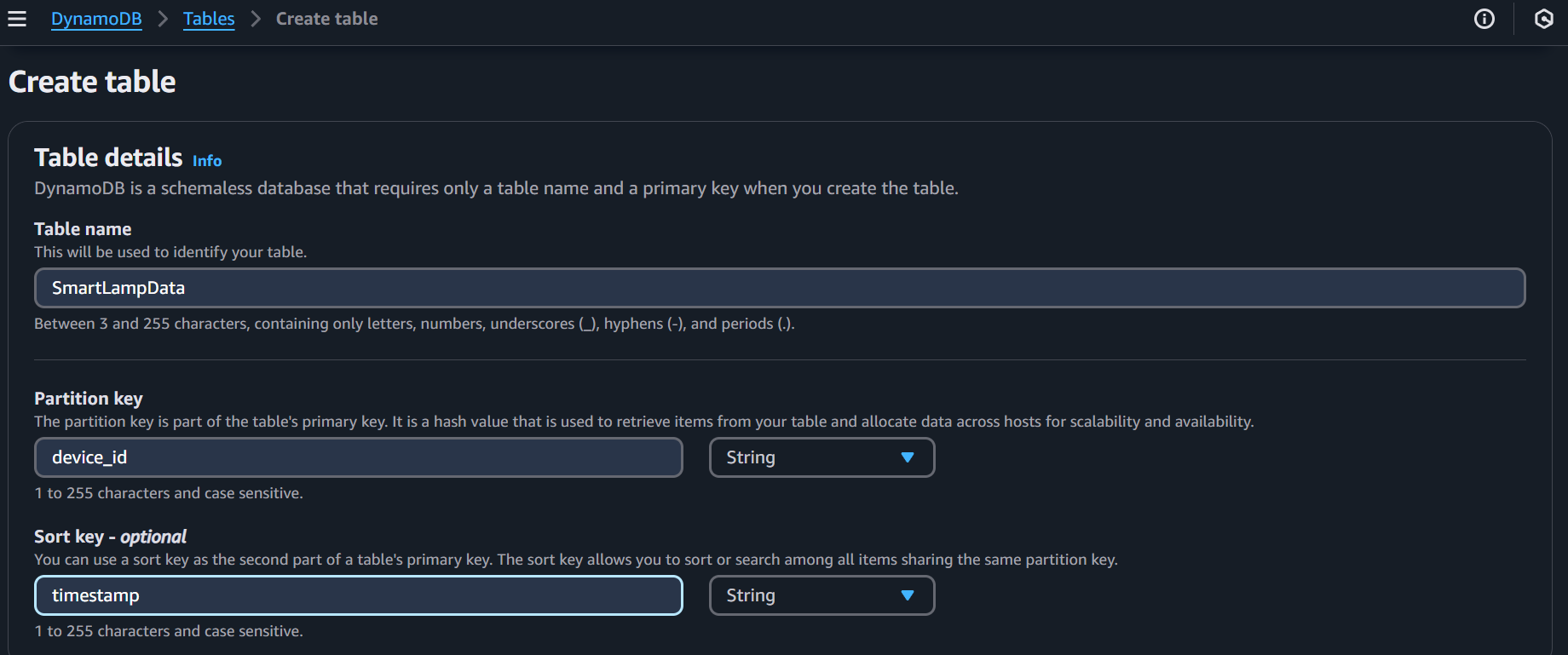


*Lambda fonksiyonunu test ederken elle verdiğimiz test verisinden (test event), light\_level = 25 ve motion = True olduğu için şartı sağlandı → lamba "****ON****" olarak belirlendi:*

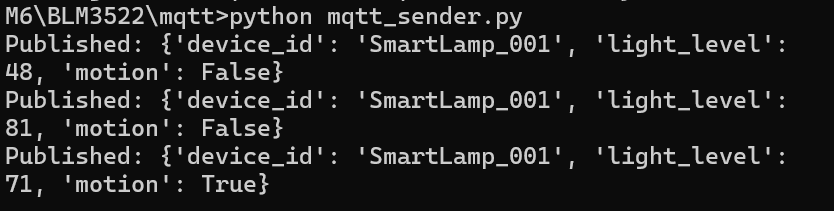


**4. Durum ve Log Verileri DynamoDb’ye Kaydedilir**

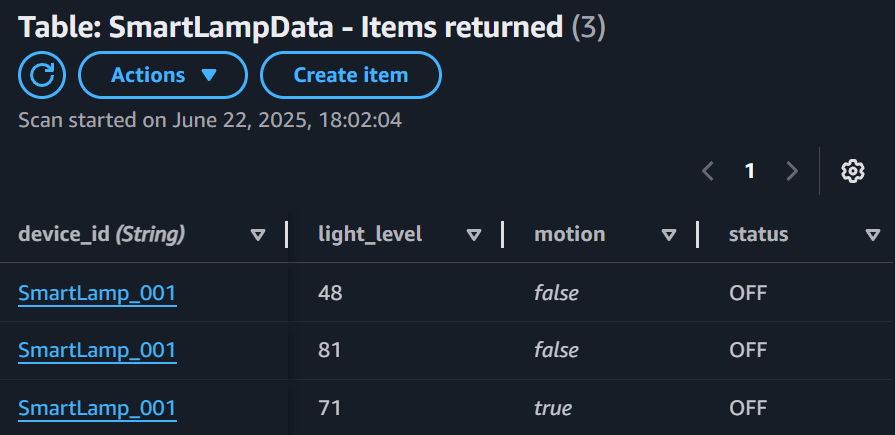
*AWS Management Console üzerinden DynamoDB servisini açarak yeni bir Table oluşturuyoruz:*

****

*mqtt\_sender.py dosyasını CMD üzerinden çalıştırıyoruz ve üretilen değerleri görebiliyoruz:*

****

*Aynı zamanda Explore Table İtems seçeneğine tıklayarak yeni gelen verileri canlı olarak gözlemleyebiliyoruz:*

****

**SONUÇ**

Bu proje kapsamında geliştirilen akıllı aydınlatma sistemi, temel bir IoT mimarisi üzerinden AWS bulut servisleriyle başarıyla entegre edilmiştir. Simülasyon ortamında MQTT protokolü aracılığıyla oluşturulan ışık şiddeti ve hareket verileri AWS IoT Core’a iletilmiş, IoT kuralları sayesinde Lambda fonksiyonuna yönlendirilmiş ve burada işlenerek DynamoDB veritabanında kaydedilmiştir. Python ile simüle edilen cihaz verileri sayesinde sistemin farklı senaryolarda nasıl tepki verdiği gözlemlenebilmiş, belirli eşiklerin (örneğin ışık seviyesi 40’ın altına düştüğünde ve hareket algılandığında) lambanın "ON" durumuna geçmesi sağlanmıştır.

Sistem, çevresel verilere göre sokak lambalarının otomatik olarak açılıp kapanmasını sağlamakta ve bu sayede hem enerji tasarrufu hem de uzaktan izlenebilirlik avantajı sunmaktadır. Gerçek fiziksel cihazlar (ör. Raspberry Pi, LDR sensör, PIR sensör) ile genişletildiğinde, proje pratik uygulamalara dönüştürülebilecek niteliktedir. Bulut bilişim kaynaklarının sunduğu ölçeklenebilirlik ve güvenilirlik, bu tarz akıllı şehir çözümleri için büyük bir avantaj sunmaktadır.

**KAYNAKLAR**

1. A. K. Sikder, A. Acar, H. Aksu, A. S. Uluagac, K. Akkaya and M. Conti, "IoT-enabled smart lighting systems for smart cities," 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2018, pp. 639-645, doi: 10.1109/CCWC.2018.8301744. keywords: {Lighting;Protocols;Smart cities;Intelligent sensors;Wireless communication;Smart Lighting;Internet of Things;Smart City}
2. AWS Documentation: <https://docs.aws.amazon.com/>
3. GitHub: <https://github.com/Impasbaa/BulutBilisim>
4. YouTube: <https://youtu.be/JVRNKY-bnjI?si=hqfi95IjIc2smjRv>