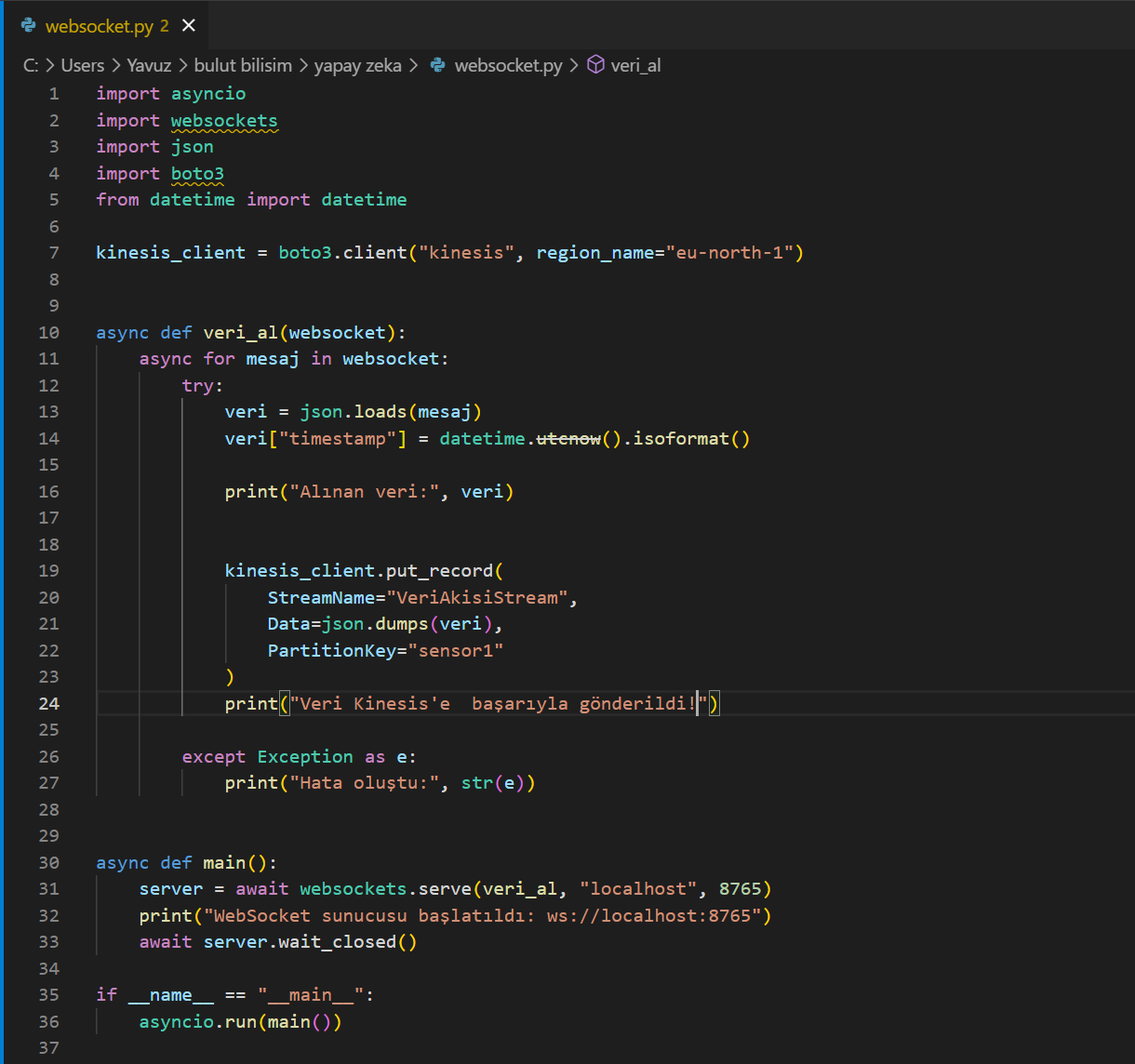
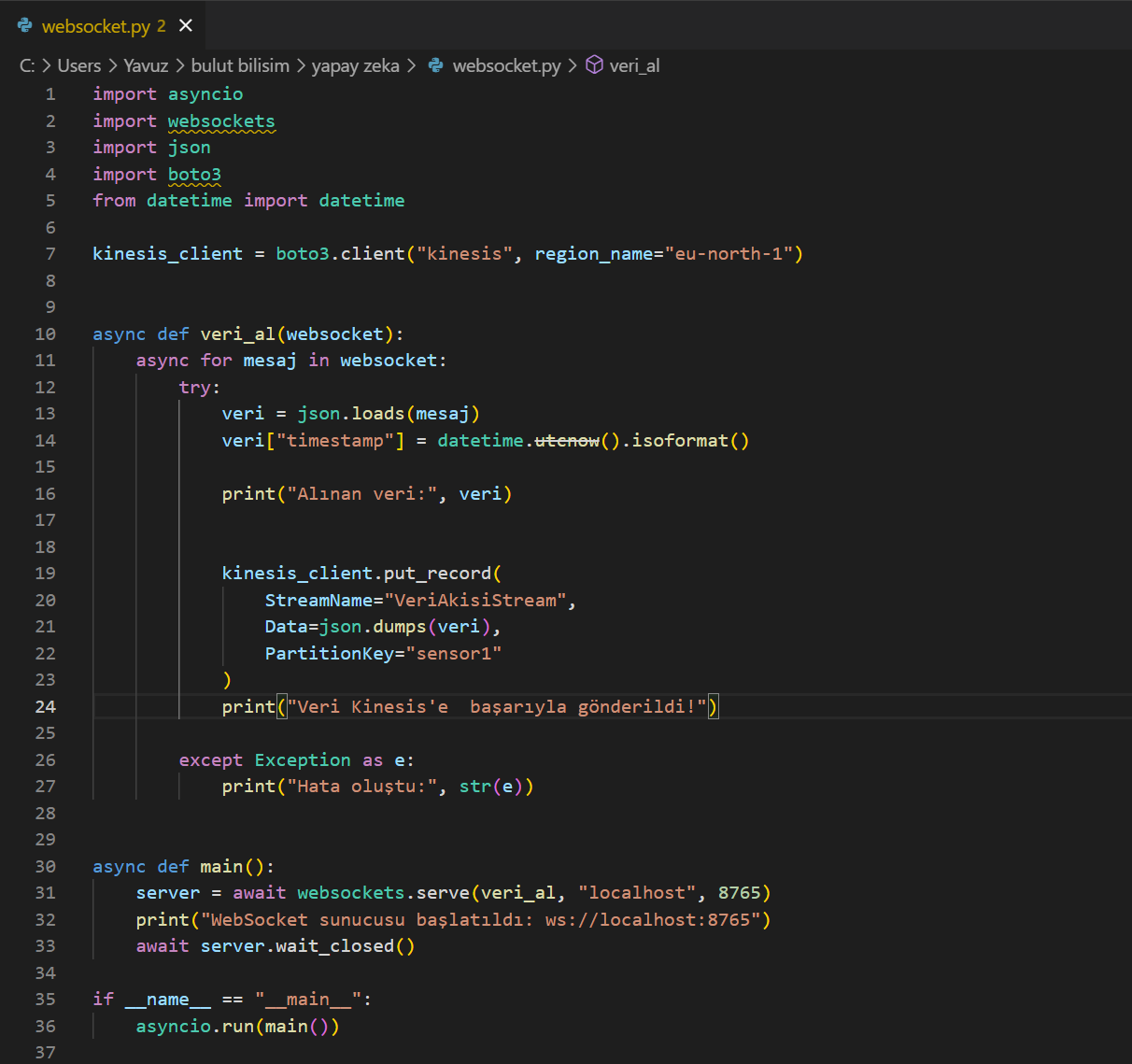
**WEBSOCKET KODU:**

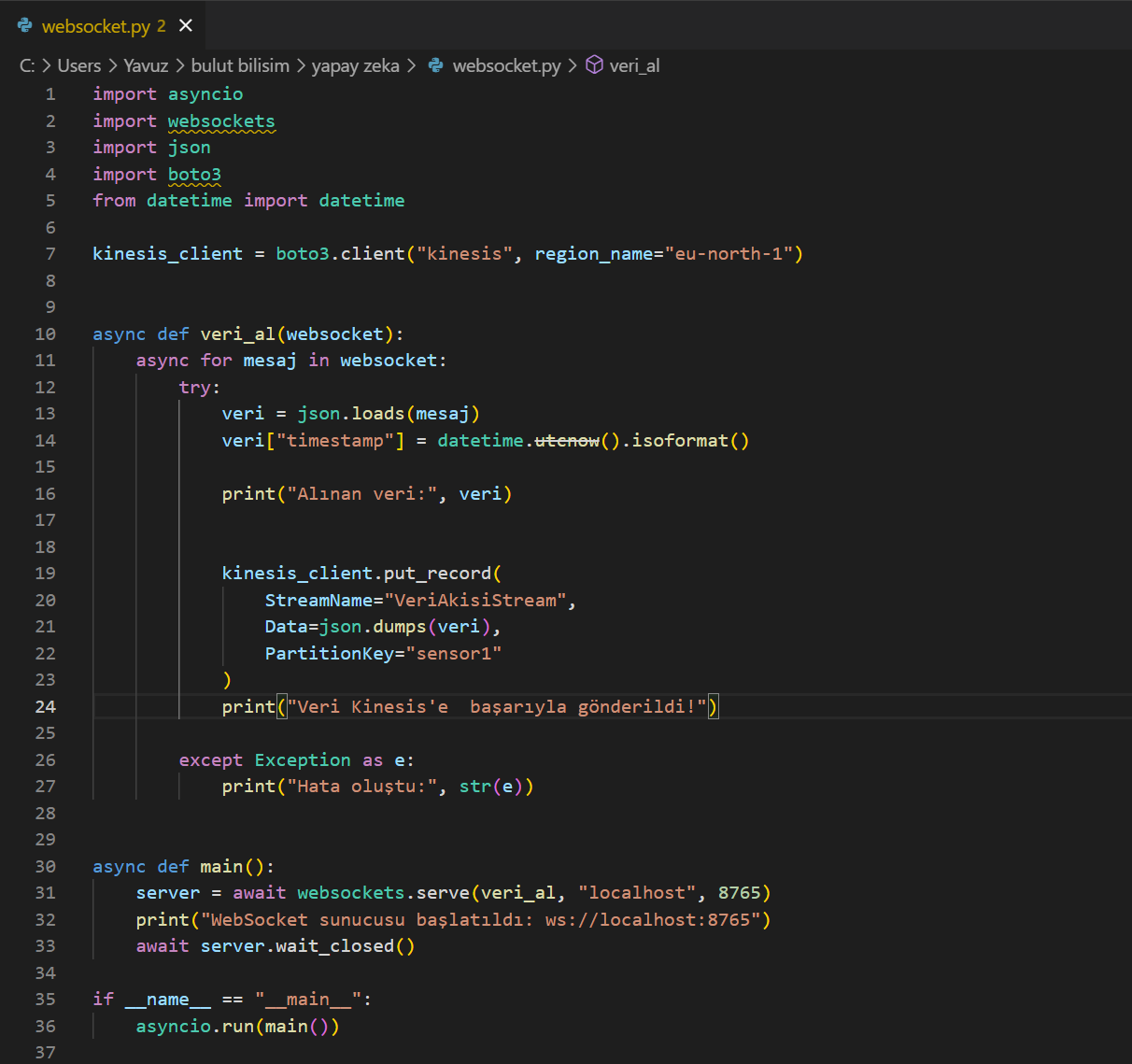
Burada Python ile oluşturacağımız websocket’i Amazon Kinesis’e bağlayarak gelen canlı veri akışını bulut ortamına aktarmak için kullanacağız



**1)** Burada öncelikle gerekli olan kütüphaneleri import ederek başlıyoruz. Daha sonrasında ise boto3 (AWS’ye bağlantı kurmak için gerekli basit kütüphaneleri kullanır) ile Kinesis’e bağlantı kuruyoruz ve region\_name kısmında ise bulut ortamında kullanacağımız bölgeyi yazıyoruz.

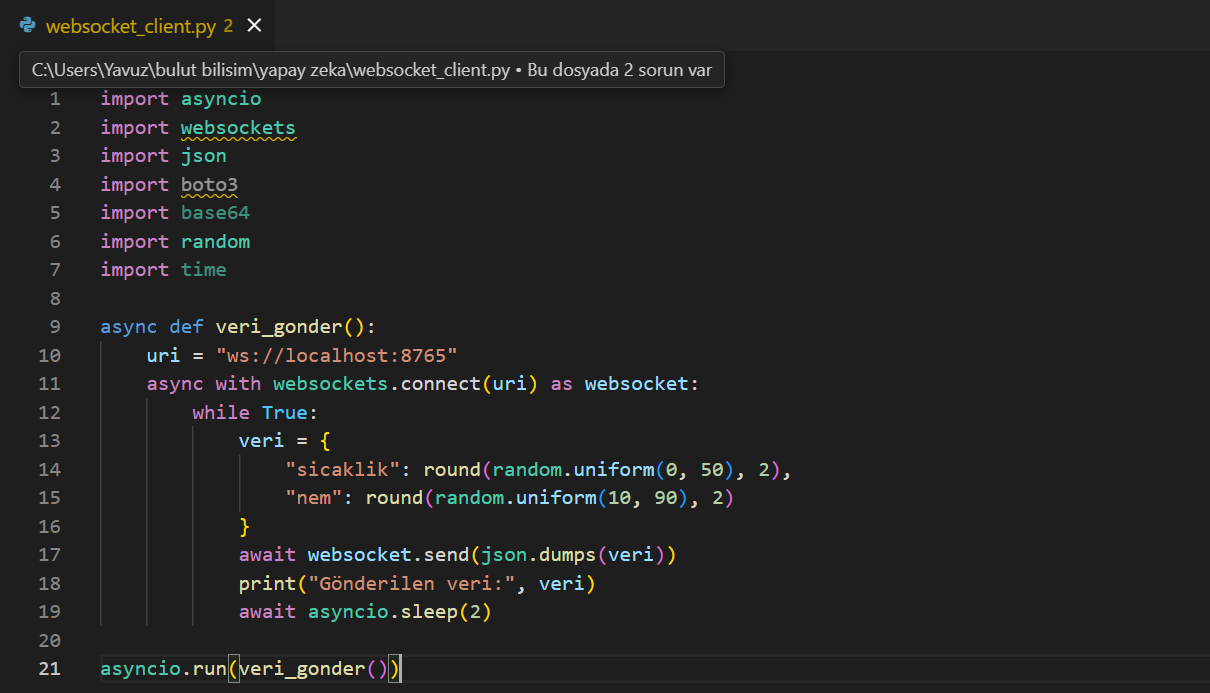


**2)** Gelen verileri alarak bu verilere “timestamp” yani veriyi aldığımız saati eklemek ve veriyi Kinesis’e yollamak için veri\_al fonksiyonunu oluşturuyoruz. Bu fonksiyon gelen veriyi json türüne çevirir bu sayede Kinesis’e yollarken ve Lambda ile bulut ortamında işleme yaparken bize kolaylık sağlar. Daha sonra terminalde alınan veriyi ekrana bastırır, VeriAkisiStream isimli önceden oluşturduğumuz Kinesis ortamına gönderir. Tüm işlemler başarı ile tamamlandığında ekrana “Veri Kinesis’e başarıyla gönderildi” mesajını yazarak fonksiyonu bitirir.



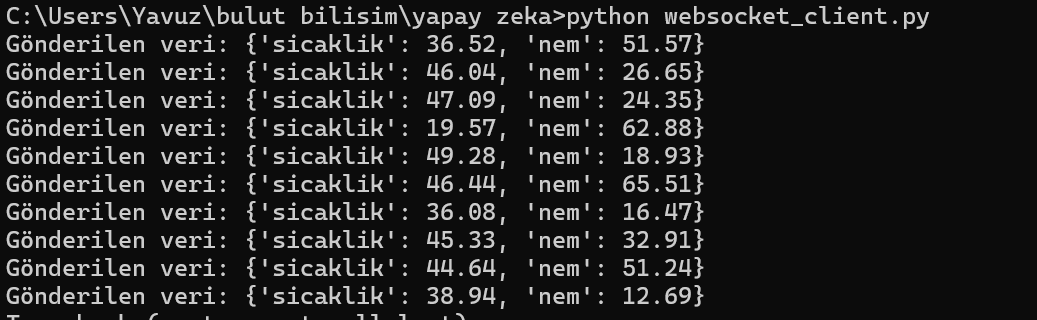
**3)** Main kısmındaki kodda ise sunucuyu başlatır ve fonksiyonu çalıştırması için gerekli parametreleri yazarız. Eğer sunucu başarılı bir şekilde başlarsa ekrana “Websocket sunucusu başlatıldı: ws://localhost:8765” mesajını bastırır. Son olarak sunucunun biz kapatana kadar aktif çalışmasını istediğimizden await kodunu ekleriz.

**WEBSOCKET\_CLİENT KODU:**

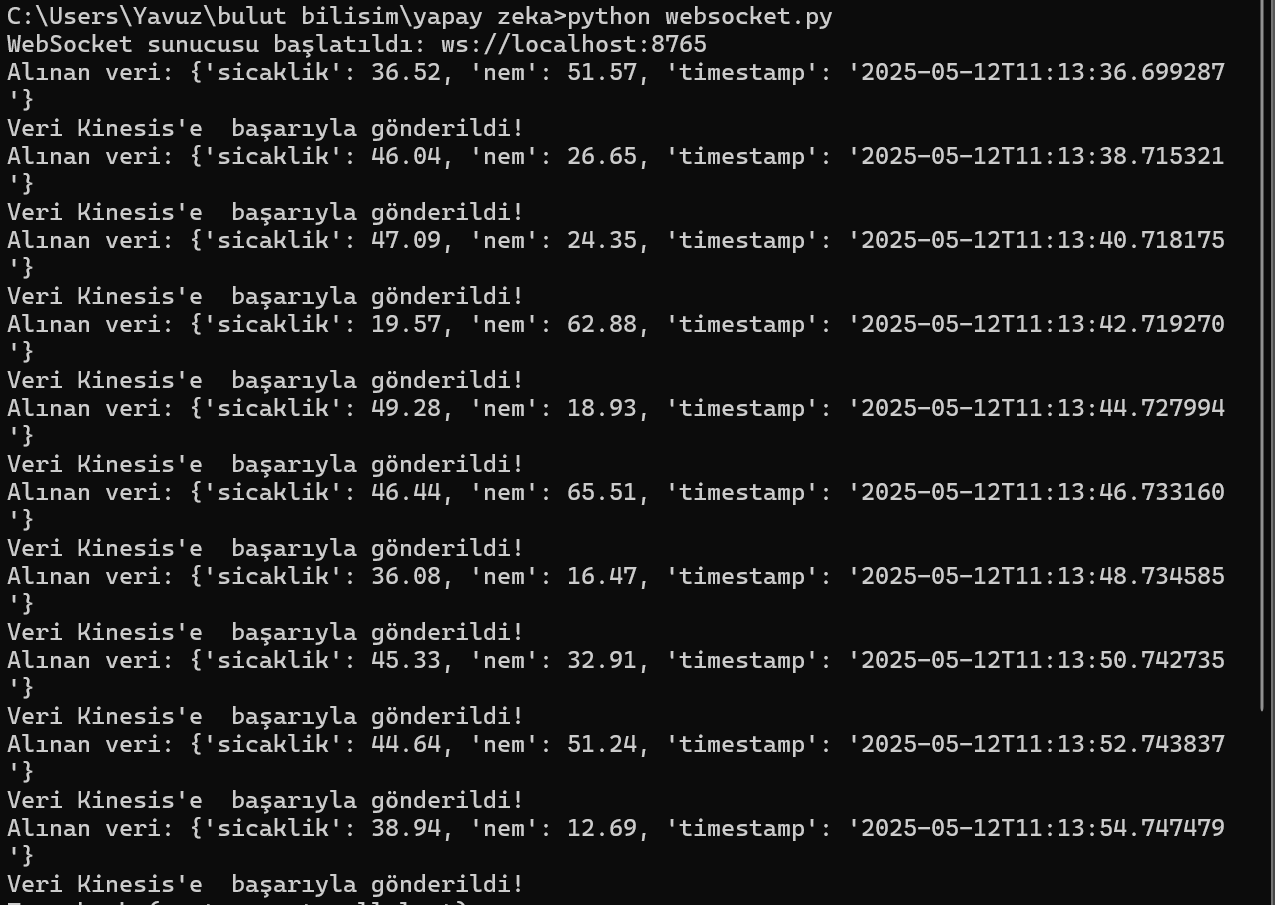


Elimizde canlı veri üreten, sıcaklık ve nem değeri ölçen bir sensör olmadığından, bir client kodu yazarak websocket sunucumuza her 2 saniyede bir random şekilde veri gönderen bir fonksiyon yazıyoruz. Burada öncelikle client’ımızı websockete bağlıyoruz. Daha sonra veri\_gonder fonksiyonu sonsuz döngüde sıcaklık ve nem verilerini yolluyoruz. Veriler başarılı bir şekilde gönderildiğinde ekrana gönderilen verinin değerini bastırıyoruz ve 2 saniyelik bekleme süremizi ekleyerek sistemi veri ile yüklememeye dikkat ediyoruz. Son olarak veri\_gonder fonksiyonunun döngüsünü başlatan satırı ekliyoruz.

**CLİENT VE WEBSOCKET KODLARININ ÇALIŞMASI:**



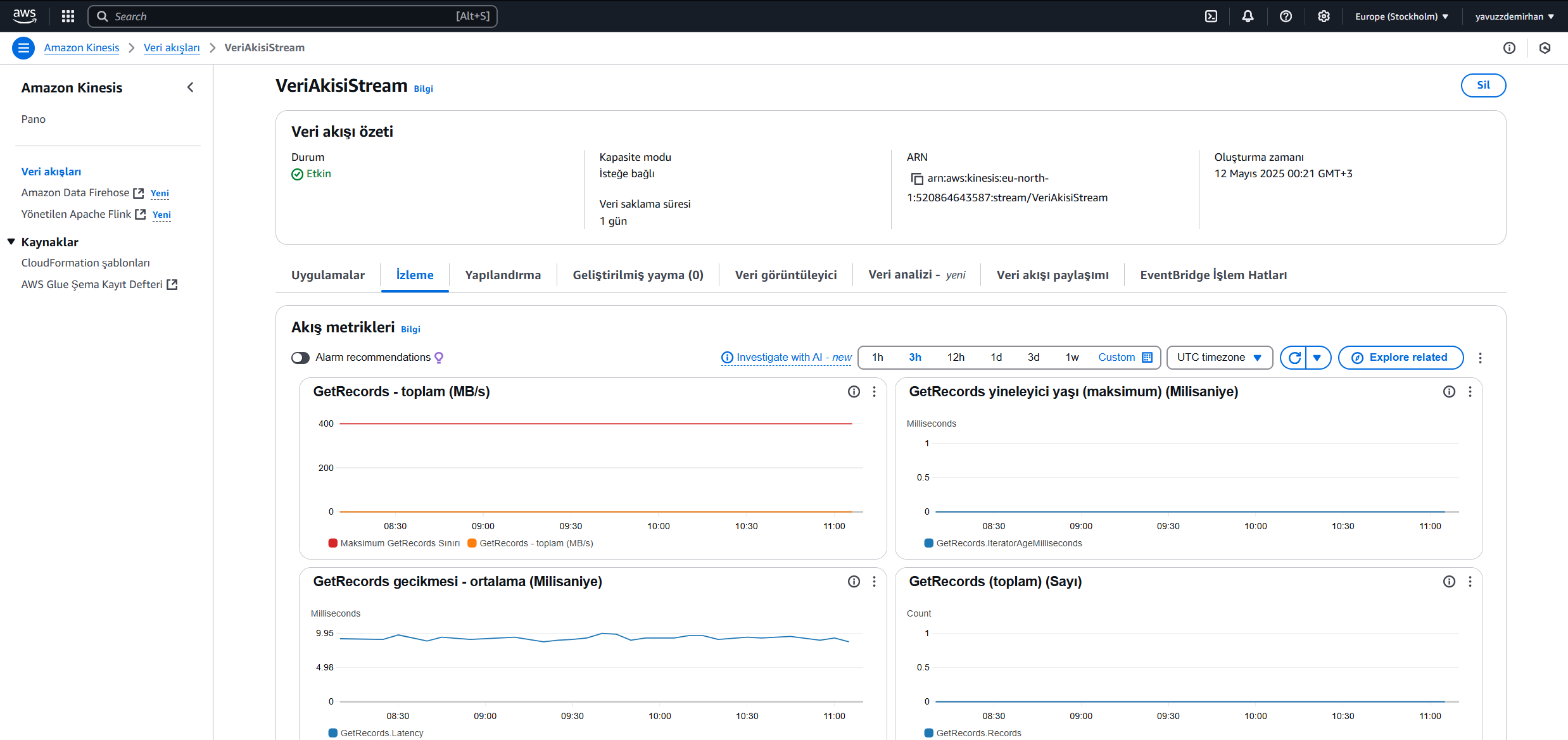
Websocket sunucusunu terminal ile başlattıktan sonra client programını çalıştırıp oluşturduğumuz 10 tane canlı veriyi her 2 saniyede bir olmak üzere Websocket sunucusuna gönderiyoruz.



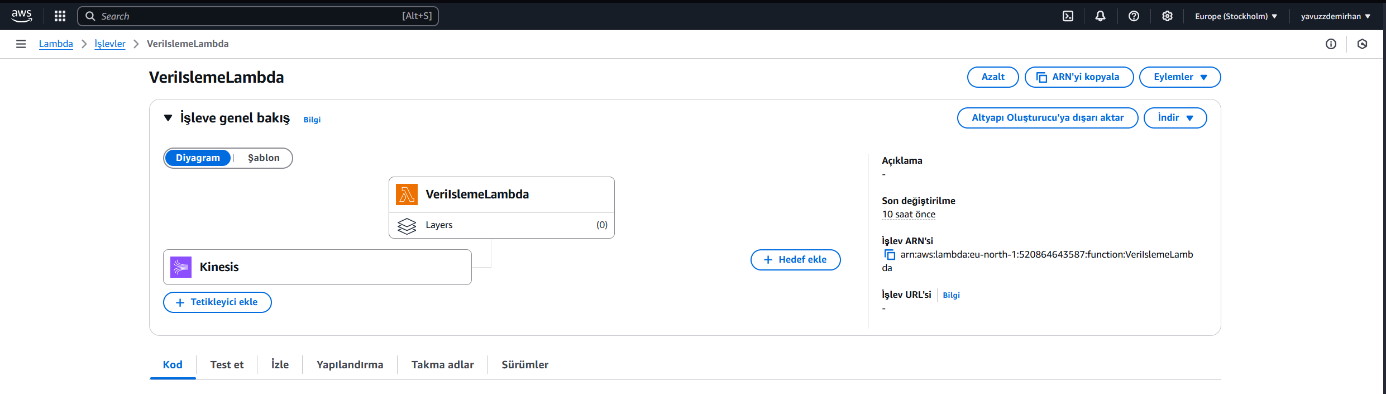
Burada ise Websocket sunucusunun başarılı bir şekilde başladığı, gelen verileri aldığı ve bunlara “timestamp” yani aldığı an bilgisayardaki süreyi yazdığını görmekteyiz. Alınan tüm veriler Amazon Kinesis’e gönderilerek bulut ortamına aktarılmış oldu. Şimdi bu verileri Amazon Lambda ile işlemeye ve bulut ortamında kaydetmeye geçebiliriz.

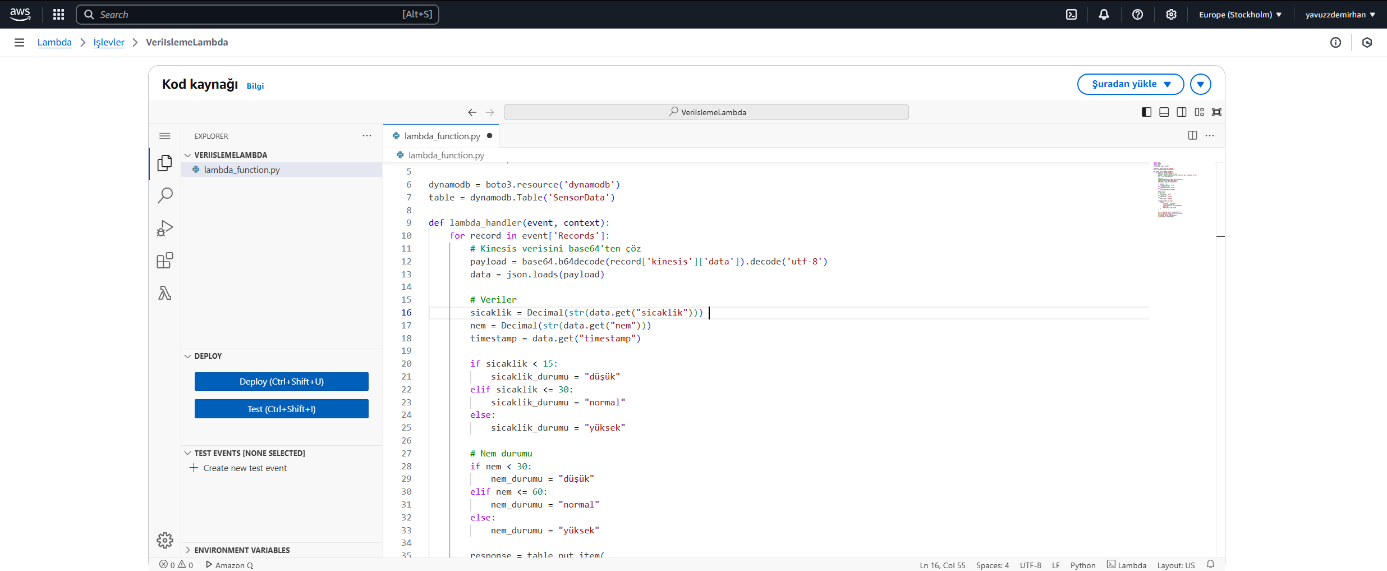
**AMAZON BULUT ORTAMINDA KULLANILAN UYGULAMALAR:**

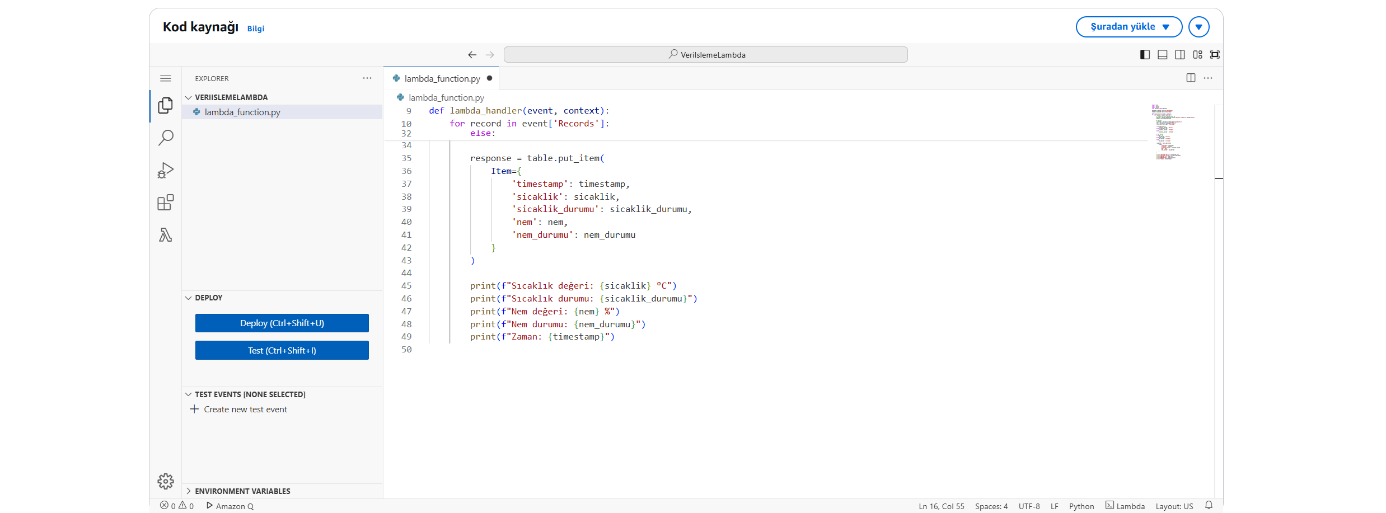
**1) Amazon Kinesis**



Burada daha önce Websocket sunucumuzu yapılandırırken de ismini gördüğümüz “VeriAkisiStrem” isimli Kinesis uygulamasını görüyoruz. Bu uygulama çalıştırdığımız Websocket sunucusundan gönderdiğimiz verileri alarak bir sonraki adımda göreceğimiz Lambda uygulamasına aktarıyor. Kinesis’in buradaki asıl amacı canlı verilerin sağlıklı ve düzenli bir şekilde almak ve bulut ortamda istediğimiz platformlara göndermektir.

**2) Amazon Lambda**





,

**Canlı Sensör Sıcaklık ve Nem İşleme Projesi**

**BULUT BİLİŞİM VE UYGULAMALARI BLM3522 DERSİ VİZE ÖDEVİ**

**2.PROJE: GERÇEK ZAMANLI VERİ AKIŞI VE İŞLEME**

**(IOT VEYA WEBSOCKET UYGULAMASI)**

**Hazırlayanlar:**

**Muhammed Faruk Gözay 22290673**

**Batuhan Dede 22290502**

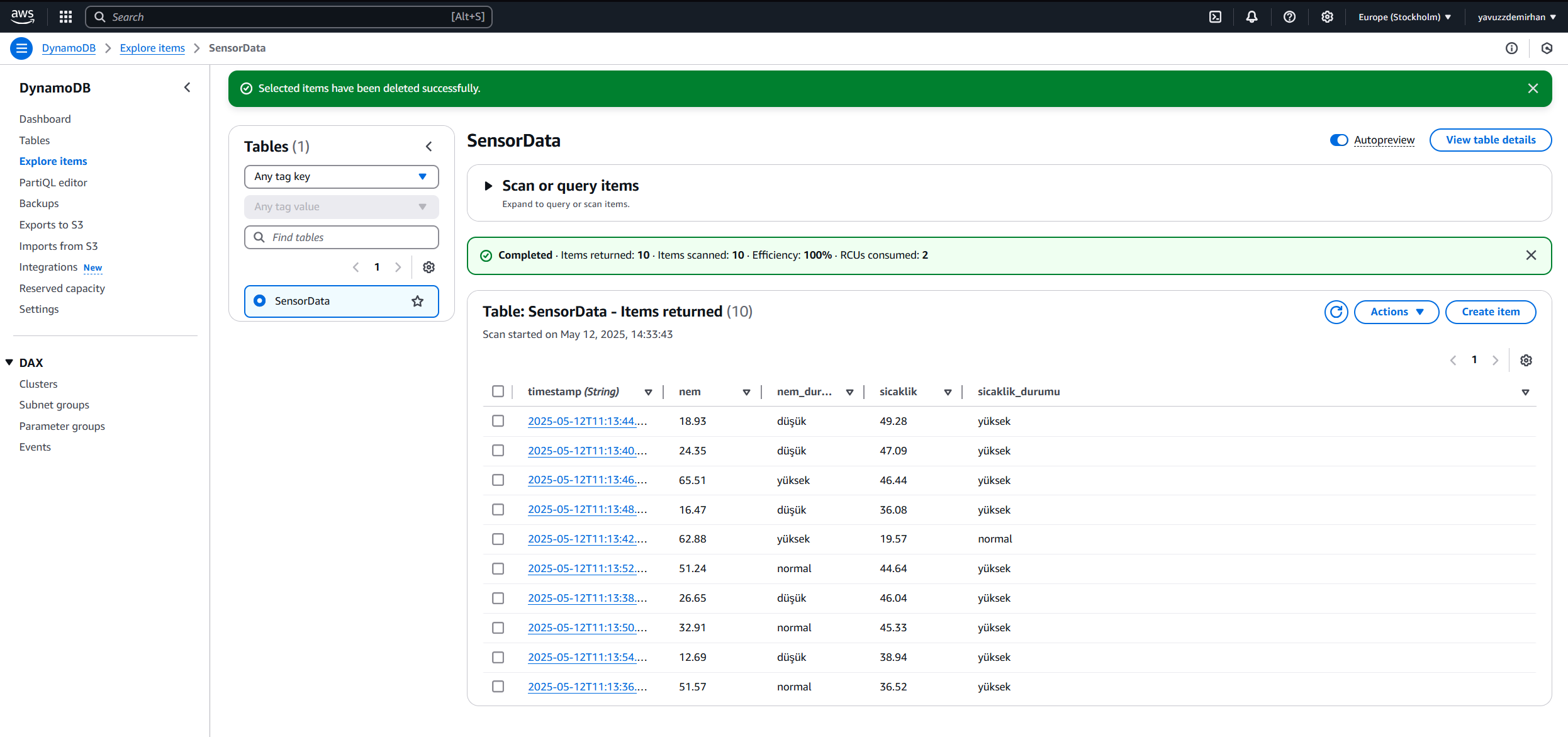
**Yavuz Selim Demirhan 22290565**

**Github Repo Adresi:**

**https://github.com/orgs/BLM3522-Grubu/repositories**

“VeriIslemeLambda” isimli Lambda uygulamasını gelen verileri işlemek ve bunları bir sonraki adımda göreceğimiz DynamoDB veri tabanına kaydetmek için kullanıyoruz. İlk fotoğrafta gördüğünüz üzere Lambda uygulamamızda tetikleyici olarak Kinesis yazdığını görebilirsiniz. Bu tetikleyici eklemek Kinesis’e veri geldiğinde Lambda’nın tetiklenerek aldığı verileri içerisine eklediğimiz kod ile işlemesini sağlamaktadır. 2. Ve 3. fotoğrafta da yazdığımız kodu görebilirsiniz. Bu kodda ise öncelikle gelen verileri sıcaklık, nem ve timestamp olarak üçe ayırıyoruz. Sıcaklık ve nem değerlerini kendi belirlediğimiz parametrelere göre işleyip bir çıktı elde ediyoruz. Daha sonra loglarda kodumuzun doğru çalıştığını kontrol etmek için print fonksiyonları ekliyoruz. Ayrıca DynamoDB’ye bu verileri kaydedeceğimiz için de put.table şeklinde fonksiyonumuzu kullanarak işlediğimiz verileri bulut ortamında bir veri tabanına kaydetmiş olacağız

**3) Amazon DynamoDB**



DynamoDB ile işlenen verileri tutmak adına “SensorData isimli bir tablo oluşturuyoruz ve bu tablonun birincil anahtarını “timestamp” yapıyoruz. Bunun sebebi gelen verileri geldikleri süre sırası ile kaydetmek istememiz. Bir önceki adımda yazdığımız Lambda kodu ile ilk client kodunu çalıştırdığımızda gönderdiğimiz 10 veriyi işleyerek bu tabloya gönderiyoruz. Bu tablo verileri Sıcaklık ve sıcaklık durumu, Nem ve nem durumu şeklinde veri tabanına dökerek işlediğimiz veriyi bulut ortamda tutmamızı sağlıyor.

**PROJENİN GENEL ÇALIŞMA PRENSİBİ:**

Canlı veri oluşturarak Websocket sunucusu ile bu verileri Amazon Kinesis aracılığı ile bulut ortamına aktarmak. Daha sonrasında ise Kinesis aracılığı ile verileri Lambdaya aktararak yapılacak tüm veri işleme yükünü bulut ortamda gerçekleştirmek. Son olarak alınan verileri ve işlenmiş değerlerini bulut ortamında tutmak amacıyla DynamoDB uygulamasını kullanarak tüm verileri güvenli şekilde tutmaktır.