



Ankara Üniversitesi

Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü

Bulut Bilişim ve Uygulamaları Proje Dokümanı

Tuğrul Özgün

22290082

<https://github.com/Forcipus/IoT-City>

Timur Malkoç

22290719

<https://github.com/TimurMalkc/Cloud-Final-ECommerce>

Proje 7 : Azure IoT Hub Tabanlı Akıllı Şehir Sensör Simülasyonu ve Gerçek Zamanlı Veri Analizi

Proje Amacı

Bu proje, bir akıllı şehir senaryosunda sıcaklık ve nem gibi çevresel verilerin simüle edilmiş IoT cihazları aracılığıyla toplanmasını, bu verilerin Azure IoT Hub üzerinden buluta iletilmesini ve gelen verilerin gerçek zamanlı olarak işlenip analiz edilmesini amaçlamaktadır. Amaç, IoT sistemlerinin temel bileşenlerini kullanarak uçtan uca bir veri akışı mimarisi oluşturmaktır.

Kullanılan Teknolojiler ve Araçlar

- **Programlama Dili:** Python
- **Bulut Platformu:** Microsoft Azure
 - Azure IoT Hub
 - Azure Event Hub (IoT Hub'ın Event Hub uyumlu uç noktası)
- **Veri Analizi:** Python ile anlık ortalama sıcaklık ve nem hesaplamaları
- **Cihaz Simülasyonu:** Python betiği ile rastgele veri üretimi

1. Azure IoT Hub Oluşturulması

- Microsoft Azure portalına giriş yapılarak, bir "IoT Hub" kaynağı oluşturuldu.
- IoT Hub içindeki "Devices" sekmesinden bir cihaz kaydedildi. Bu cihazın adı ve **connection string** bilgisi daha sonra simülasyon kodunda kullanılmak üzere not edildi.

2. IoT Cihazı Emulatörü (Veri Gönderimi)

- Python ile `iot_simulator.py` adında bir dosya oluşturuldu.
- Bu dosyada:
 - Azure IoT Device SDK (`azure.iot.device`) kullanılarak doğrudan Azure IoT Hub'a bağlantı sağlandı.
 - Rastgele sıcaklık ve nem verileri üreten bir algoritma yazıldı.
 - Her 5 saniyede bir bu veriler JSON formatında Azure IoT Hub'a gönderildi.
 - Kod çalıştırıldığında terminalde mesajların başarıyla gönderildiği görüldü

The screenshot shows the Azure IoT Explorer (preview) interface. The top window is a terminal titled 'Komut İstemi' (Command Prompt) showing the execution of a Python script 'iot_simulator.py'. The terminal output displays ten JSON objects, each representing a temperature and humidity reading. Below the terminal, the main interface is divided into two panes. The left pane contains a sidebar with 'Cloud-to-device message', 'Module identities', and 'IoT Plug and Play components'. The right pane, titled 'Show system properties', displays a list of received events. Each event is a JSON object with a 'body' containing 'temperature' and 'humidity' values, and an 'enqueuedTime' timestamp. The events are listed chronologically from top to bottom.

```
C:\Users\EEemreÖZ\Desktop\iot>python iot_simulator.py
Gönderiliyor: {"temperature": 33.12, "humidity": 42.55}
Gönderiliyor: {"temperature": 21.96, "humidity": 62.77}
Gönderiliyor: {"temperature": 20.69, "humidity": 51.55}
Gönderiliyor: {"temperature": 31.45, "humidity": 67.75}
Gönderiliyor: {"temperature": 25.13, "humidity": 33.43}
Gönderiliyor: {"temperature": 32.29, "humidity": 51.4}
Gönderiliyor: {"temperature": 28.52, "humidity": 31.2}
Gönderiliyor: {"temperature": 34.4, "humidity": 54.4}
Gönderiliyor: {"temperature": 21.22, "humidity": 39.35}
Gönderiliyor: {"temperature": 23.32, "humidity": 59.94}
```

Cloud-to-device message

Module identities

IoT Plug and Play components

Show system properties

Receiving events...

Thu May 29 2025 13:47:35 GMT+0300 (GMT+03:00):

```
{
  "body": {
    "temperature": 34.65,
    "humidity": 31.68
  },
  "enqueuedTime": "Thu May 29 2025 13:47:35 GMT+0300 (GMT+03:00)"
}
```

Thu May 29 2025 13:47:30 GMT+0300 (GMT+03:00):

```
{
  "body": {
    "temperature": 26.43,
    "humidity": 51.64
  },
  "enqueuedTime": "Thu May 29 2025 13:47:30 GMT+0300 (GMT+03:00)"
}
```

Thu May 29 2025 13:47:25 GMT+0300 (GMT+03:00):

```
{
  "body": {
    "temperature": 30.18,
    "humidity": 49.81
  },
  "enqueuedTime": "Thu May 29 2025 13:47:25 GMT+0300 (GMT+03:00)"
}
```

3. Azure IoT Explorer ile Doğrulama

- Azure tarafından sunulan masaüstü aracı **Azure IoT Explorer** kullanılarak gönderilen veriler görüntülendi.
- Cihaz bağlantısı kurularak gelen mesajların formatı ve frekansı gerçek zamanlı izlendi.

4. Event Hub Uyumlu Uç Nokta ile Veri Dinleme

- IoT Hub içerisindeki **Event Hub uyumlu uç nokta** bilgileri ve erişim anahtarları alındı.
- azure-eventhub kütüphanesi kullanılarak bir Python dosyası (azure_eventhub_listener.py) oluşturuldu.
- Bu kod:
 - Event Hub'a bağlandı.
 - Her gelen mesajı parse ederek sıcaklık ve nem değerlerini terminalde yazdırdı.
 - Tüm gelen mesajların ortalamasını hesaplayarak analiz yaptı.

1. IoT Cihazı (Simülasyon: Python kodu ile MQTT üzerinden veri gönderimi)



2. Azure IoT Hub (Veriyi alır ve Event Hub'a yönlendirir)



3. Event Hub Uyumlu Uç Nokta (Veriyi dış sistemlere açar)



4. Python Listener (azure-eventhub-listener.py ile veriyi dinler ve işler)

5. Gerçek Zamanlı Veri Analizi

- Kod çalışırken her gelen veri üzerinde:
 - Anlık olarak sıcaklık ve nem verileri gösterildi.
 - Tüm veriler bir listede saklandı.
 - Bu liste üzerinde mean() hesaplamalarıyla ortalama değerler anlık olarak terminalde verildi.
- Bu kısım, veri görselleştirme yerine ilk analiz katmanını olarak işlev gördü.

```
Seç Komut İstemi
^C
C:\Users\EEmreÖZ\Desktop\iot>python azure_eventhub_listener.py
Veri dinleniyor... Ctrl+C ile durdurabilirsiniz.
Partition: 0 | Data: {"temperature": 22.92, "humidity": 54.49}
Ortalama Sıcaklık: 22.92 °C, Ortalama Nem: 54.49 %
Partition: 0 | Data: {"temperature": 26.97, "humidity": 55.1}
Ortalama Sıcaklık: 24.95 °C, Ortalama Nem: 54.80 %
Partition: 0 | Data: {"temperature": 28.5, "humidity": 41.28}
Ortalama Sıcaklık: 26.13 °C, Ortalama Nem: 50.29 %
Partition: 0 | Data: {"temperature": 32.35, "humidity": 69.68}
Ortalama Sıcaklık: 27.69 °C, Ortalama Nem: 55.14 %
Partition: 0 | Data: {"temperature": 25.18, "humidity": 38.7}
Ortalama Sıcaklık: 27.18 °C, Ortalama Nem: 51.85 %
Partition: 0 | Data: {"temperature": 31.32, "humidity": 30.32}
Ortalama Sıcaklık: 27.87 °C, Ortalama Nem: 48.26 %
Partition: 0 | Data: {"temperature": 32.86, "humidity": 46.84}
Ortalama Sıcaklık: 28.59 °C, Ortalama Nem: 48.06 %
Partition: 0 | Data: {"temperature": 30.41, "humidity": 36.38}
```

6. Grafik Çizimi ve Görselleştirme

Veri analizinin bir sonraki adımı olarak, simülasyon sonucu elde edilen sıcaklık ve nem verilerinin grafiksel olarak görselleştirilmesi gerçekleştirildi. Bu sayede verilerin zamana göre değişimi incelenerek daha anlamlı çıkarımlar elde edildi.

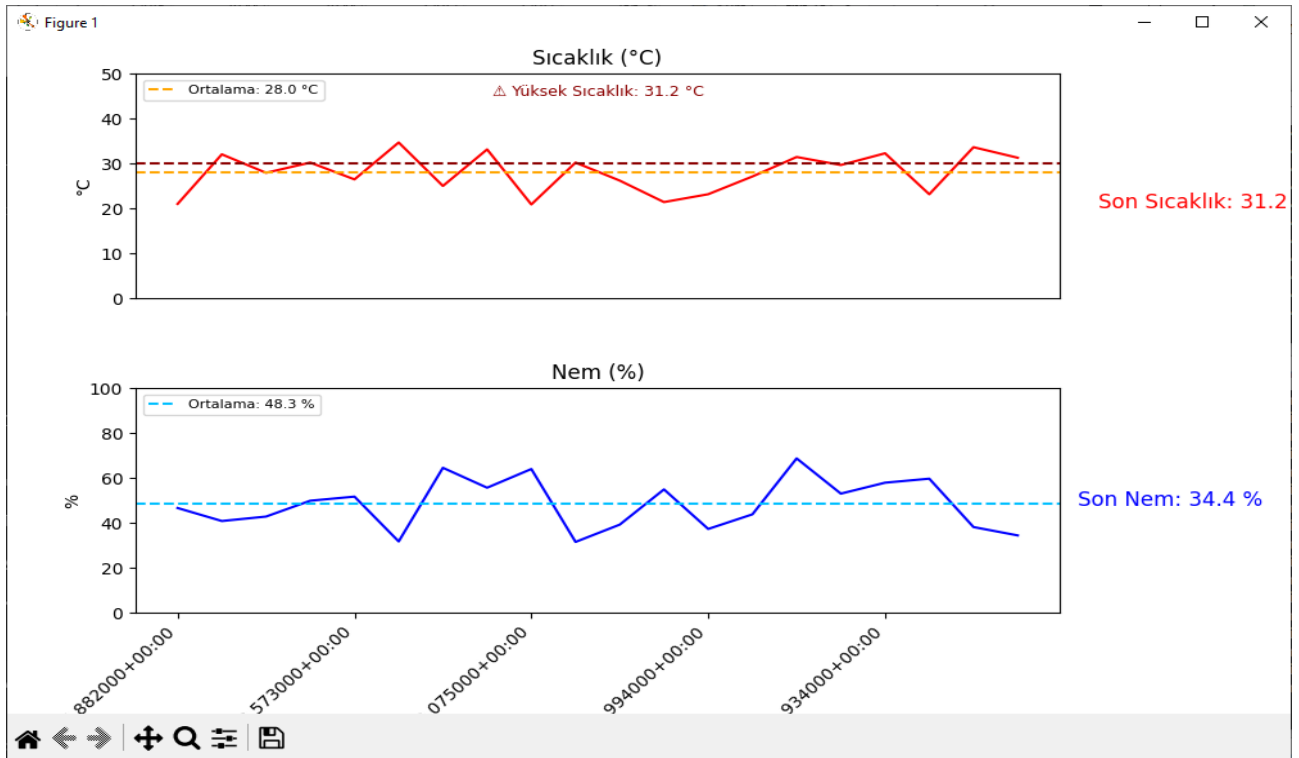
A. Python ile Grafik Çizimi (grafik_çiz.py)

- matplotlib ve json kütüphaneleri kullanılarak basit bir görselleştirme aracı geliştirildi.
- azure_eventhub_listener.py betiği ile toplanan sıcaklık ve nem verileri bir .json dosyasına kaydedildi.
- Ardından bu JSON dosyası grafik_çiz.py dosyası tarafından okunarak çizgi grafik şeklinde görselleştirildi.

B. MATLAB ile İleri Seviye Görselleştirme

MATLAB, bilimsel veri analizi ve istatistiksel grafikler oluşturmak için güçlü bir araçtır. Bu projede Python ile kaydedilen veriler MATLAB ortamına aktarılmış ve burada daha gelişmiş analizler yapılmıştır.

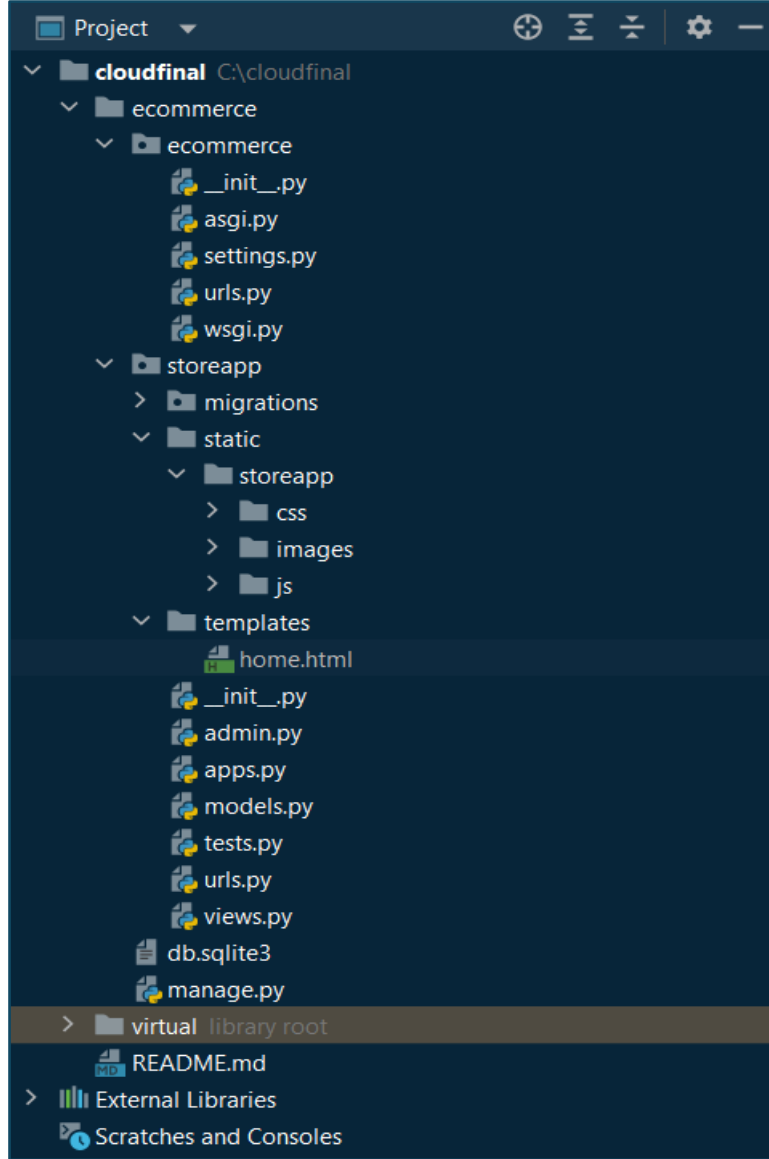
- Python'da oluşturulan JSON dosyası, CSV formatına dönüştürüldü.
- MATLAB'de CSV dosyası içe aktarıldı:
- İkili eksen kullanan detaylı bir grafik oluşturuldu:



- Grafik aldığı JSON dosyasındaki verileri gerçek zamanlı tabloya aktardı.
- Eğer sıcaklık ve nem belirlenen değerin üzerinde ise uyarı verdi
- O ana kadar girilen verilerden yola çıkarak hesaplanan ortalama sıcaklık ve nemi gösterdi

Proje 4: E-Ticaret Uygulaması (Otomatik Ölçeklendirme ve Yönetim)

Proje içerisinde back-end dili olarak Python, back-end kütüphanesi olarak Django, veritabanı olarak MySQL ve bulut platformu olarak Google Cloud kullanılmıştır. E-Ticaret sitesinin görünümü internet üzerinden hazır olarak alınıp bir Django uygulaması üzerine aktarılmıştır. Python üzerinde html dosyası templates klasörüne; diğer css, js ve png dosyaları da static klasörüne atılmış, bağlantılar yeni dosya yollarına göre düzenlenerek back-end kısmı tamamlanmıştır.



Google Cloud üzerinde bulunan Compute Engine hizmeti kullanılarak websitesi için bir instance ve instance group oluşturulmuştur.

VM instances							
Filter Enter property name or value							
<input type="checkbox"/> Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
<input type="checkbox"/>	instance-20250528-215021	us-central1-c			10.128.0.3 (nic0)	34.46.195.28 (nic0)	SSH ▾ ⋮
<input type="checkbox"/>	instance-group-1-xkgh	us-central1-c		instance- ▾	10.128.0.4 (nic0)	34.133.117.112 (nic0)	SSH ▾ ⋮

Oluşturalan instance group haricinde bir back-end config ve health check hazırlanmıştır. Bu hizmetlerin üçü kullanılarak Google Cloud Load Balancer hizmeti ile otomatik ölçeklendirme yapılmıştır.

Backend service details

Edit

Delete

backendcong

General properties

Load balancer type	Global external Application Load Balancer (EXTERNAL_MANAGED)
Endpoint protocol	HTTP
In use by	finalloadbalancer
Timeout ?	30 seconds
IP address selection policy ?	Only IPv4
Health check	final-healthcheck View health check details
Backend security policy	default-security-policy-for-backend-service-backendcong
Session affinity	None
Cloud CDN	Enabled View details
Connection draining timeout	300 seconds
Custom request headers ?	Currently there are no custom request headers configured
Custom response headers ?	Currently there are no custom response headers configured
Logging	Disabled
Sample rate	0
Backend authentication	Disabled

Health checks

Create health check

Refresh

Delete

Health checks determine if applications on your VMs respond to requests. They're used for load balancing and with autohealing in managed instance groups. [Learn more](#)

Filter

Enter property name or value

<input type="checkbox"/>	Name ↑	Scope	Region	Host	Path	Protocol	Port	In use by
<input type="checkbox"/>	final-healthcheck	Global				TCP	80	backendcong

Load balancing

+ Create load balancer

Refresh

Delete

Learn

Load balancers

Backends

Frontends

Service LB policies

Filter

Enter property name or value

?

<input type="checkbox"/>	Name	Load balancer type	Access type	Protocols	Region	Backends	Actions
<input type="checkbox"/>	finalloadbalancer	Application	External	HTTP		✔ 1 backend service (1 instance group, 0 network endpoint groups)	⋮

Compute Engine instance ile sağlanan SSH (secure shell) kullanılarak websitesine bir MySQL veri tabanı eklenmiştir.

```
(venv) timurmlkc@instance-20250528-215021:~/Cloud-Final-ECommerce$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.42 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Bütün özelliklerin geliştirilmesi tamamlandıktan sonra SSH üzerinden Nginx ve Gunicorn yardımı ile istenilen ip adresi üzerinde bir sunucu açılmıştır.

```
ssh.cloud.google.com/v2/ssh/projects/midterm-commerce/zones/us-central1-c/instances/instance-20250528-215021?authuser=2&hl=en_...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

(venv) timurmlkc@instance-20250528-215021:~/Cloud-Final-ECommerce$ sudo systemctl status gunicorn
● gunicorn.service - gunicorn daemon for Django app
   Loaded: loaded (/etc/systemd/system/gunicorn.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-05-29 22:48:58 UTC; 7min ago
   Main PID: 2610 (gunicorn)
     Tasks: 4 (limit: 1136)
    Memory: 104.8M
       CPU: 1.610s
   CGroup: /system.slice/gunicorn.service
           └─2610 /home/timurmlkc/venv/bin/python3 /home/timurmlkc/venv/bin/gunicorn --access-logfile - --work>
           └─2612 /home/timurmlkc/venv/bin/python3 /home/timurmlkc/venv/bin/gunicorn --access-logfile - --work>
           └─2613 /home/timurmlkc/venv/bin/python3 /home/timurmlkc/venv/bin/gunicorn --access-logfile - --work>
           └─2614 /home/timurmlkc/venv/bin/python3 /home/timurmlkc/venv/bin/gunicorn --access-logfile - --work>

May 29 22:49:12 instance-20250528-215021 gunicorn[2613]: Not Found: /images/ad-image-1.png
May 29 22:49:12 instance-20250528-215021 gunicorn[2613]: - - [29/May/2025:22:49:12 +0000] "GET /images/ad-image>
May 29 22:49:12 instance-20250528-215021 gunicorn[2613]: Not Found: /images/ad-image-2.png
May 29 22:49:12 instance-20250528-215021 gunicorn[2613]: - - [29/May/2025:22:49:12 +0000] "GET /images/ad-image>
May 29 22:49:12 instance-20250528-215021 gunicorn[2614]: Not Found: /images/ad-image-3.png
May 29 22:49:12 instance-20250528-215021 gunicorn[2614]: - - [29/May/2025:22:49:12 +0000] "GET /images/ad-image>
May 29 22:49:12 instance-20250528-215021 gunicorn[2613]: Not Found: /images/ad-image-4.png
May 29 22:49:12 instance-20250528-215021 gunicorn[2613]: - - [29/May/2025:22:49:12 +0000] "GET /images/ad-image>
May 29 22:49:13 instance-20250528-215021 gunicorn[2612]: Not Found: /images/bg-leaves-img-pattern.png
May 29 22:49:13 instance-20250528-215021 gunicorn[2612]: - - [29/May/2025:22:49:12 +0000] "GET /images/bg-leave>
log file: udo nginx -t
[1]+  Stopped                  sudo systemctl status gunicorn
(venv) timurmlkc@instance-20250528-215021:~/Cloud-Final-ECommerce$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
(venv) timurmlkc@instance-20250528-215021:~/Cloud-Final-ECommerce$ sudo systemctl restart nginx
(venv) timurmlkc@instance-20250528-215021:~/Cloud-Final-ECommerce$
```

Google Cloud tarafından sağlanan ip adresi ile websitesine erişilebilmektedir.

