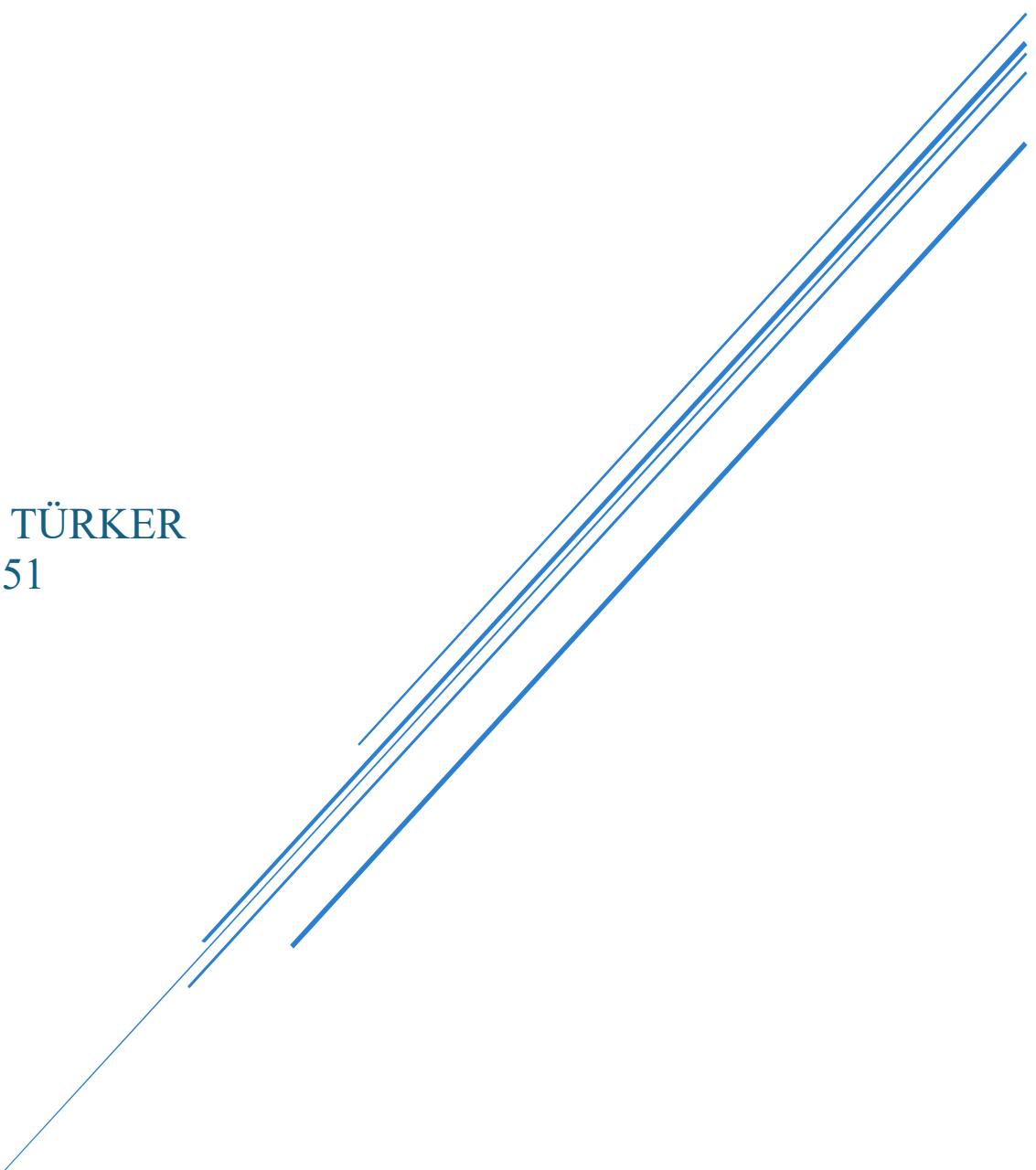


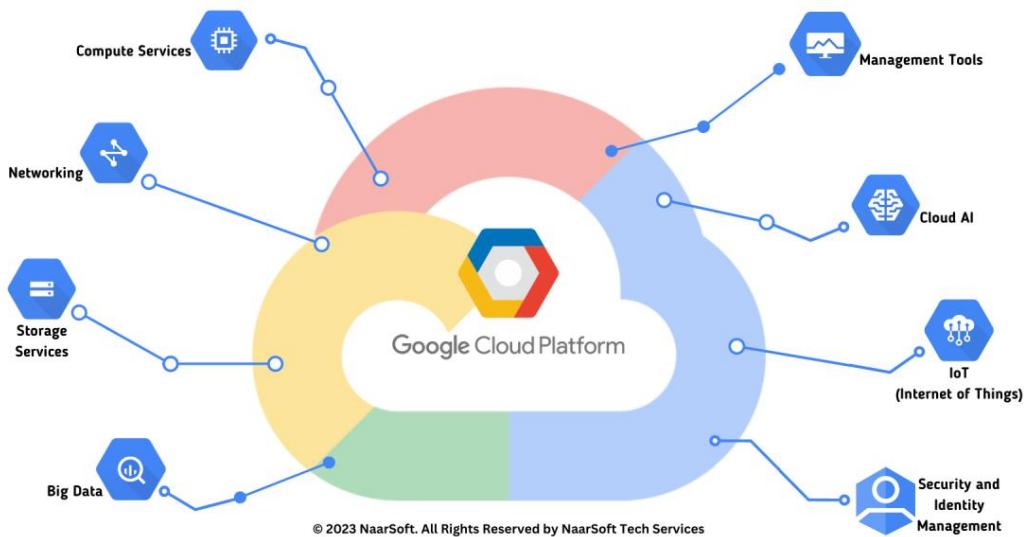
**ISE 465 – BULUT BİLİŞİM
DERSİ 2. ÖDEV 2025-2026
GÜZ DÖNEMİ**

FURKAN TÜRKER
B221200051



1. Giriş ve Amaç

Bu çalışmanın amacı, seçilen bir web uygulamasının Google Cloud platformuna taşınarak bulut ortamında çalışır hâle getirilmesi ve bulut servislerinin (dağıtım, erişim kontrolü, kimlik yönetimi, ölçeklendirme, izleme) kullanılarak uygulamanın yönetilebilir ve güvenli şekilde yayınlanmasının sağlanmasıdır. Ödev kapsamında Python/Flask ile geliştirilen basit bir web uygulaması konteynerleştirilmiş, Google Cloud Build ile imaj hâline getirilmiş ve Google Cloud Run üzerinde yayınlanmıştır. Ayrıca IAM yetkilendirmesi ile erişim kısıtlamaları uygulanmış, Service Account kullanımı ile minimum yetki prensibi doğrultusunda güvenlik artırılmıştır.



2. Bulut Platformu Seçimi

Bu projede **Google Cloud Platform (GCP)** tercih edilmiştir. Bunun temel sebepleri:

- Sunucusuz mimari (serverless) desteği
- Otomatik ölçeklendirme (auto scaling)
- Docker tabanlı uygulamaları doğrudan çalıştırılabilmesi
- Cloud Run ile hızlı deployment imkânı
- Gelişmiş IAM ve güvenlik altyapısı
- Öğrenciler için ücretsiz kota imkânı

Cloud Run servisi sayesinde herhangi bir sanal makine yönetimine ihtiyaç duyulmadan uygulama doğrudan çalıştırılmıştır.

Aynı zamanda basit arayüz ve kullanım kolaylığı ile diğer platformların önüne geçmiştir.

3. Kullanılan Teknolojiler ve Servisler

- **Uygulama Teknolojisi:** Python, Flask
- **Konteynerleştirme:** Dockerfile tabanlı containerization
- **Bulut Platformu:** Google Cloud Platform (GCP)
- **Build Servisi:** Cloud Build
- **Container Registry:** Artifact Registry
- **Çalıştırma Ortamı:** Cloud Run (Fully Managed)
- **Erişim Kontrolü:** IAM (Cloud Run Invoker rolü)
- **Kimlik Yönetimi:** Service Account
- **Gözlemeleme:** Cloud Logging / Cloud Run Metrics

Teknoloji	Kullanım Amacı
• Python	Uygulama geliştirme
• Flask	Web servis geliştirme
• Docker	Uygulamanın konteynerleştirilmesi
• Google Cloud Run	Uygulamanın bulutta çalıştırılması
• Google Cloud Build	Docker imajının oluşturulması
• Artifact Registry	Container imajlarının saklanması
• IAM	Yetkilendirme ve erişim kontrolü
• Service Account	Güvenli kimlik yönetimi
• Cloud Logging	Uygulama loglarının izlenmesi

3. Uygulama Seçimi ve Tanımı

1. Uygulama Seçimi

Bu projede, bulut ortamına taşınmaya uygun, hafif ve ölçeklenebilir bir web uygulaması geliştirilmiştir. Uygulama, Python programlama dili ve Flask web framework’ü kullanılarak oluşturulmuştur.

Flask tercih edilme sebepleri:

- Hafif ve öğrenilmesi kolay bir framework olması
- Mikroservis mimarisine uygun yapıda olması
- Container tabanlı ortamlarda sorunsuz çalışması
- Cloud Run gibi sunucusuz platformlara kolayca entegre edilebilmesi

Uygulama, temel olarak HTTP isteklerine yanıt veren bir web servisi olarak tasarlanmış ve sistemin çalıştığını doğrulamak amacıyla /health endpoint'i eklenmiştir. Bu endpoint sayesinde uygulamanın ayakta olup olmadığı kolayca test edilebilmektedir.

Bu yapı, gerçek hayatı kullanılarak REST tabanlı servis mimarisini temsil etmektedir.

Geliştirilen Flask uygulaması iki temel endpoint içermektedir:

- / : Basit bir metin yanıtı döndürür.
- /health : Servisin ayakta olduğunu doğrulamak için JSON yanıt döndürür.

main.py dosyası Cloud Run uyumluluğu için aşağıdaki kritik noktaları sağlar:

- Cloud Run'in verdiği **PORT** environment variable okunur.
- Uygulama **0.0.0.0** host'u üzerinde dinlenir.

Kod (özet):

- app.run(host="0.0.0.0", port=port)
- port = int(os.environ.get("PORT", "8080"))

```

BULUTB
> .venv
Dockerfile
main.py
requirements.txt
main.py
1  from flask import Flask
2
3  app = Flask(__name__)
4
5  @app.get("/")
6  def home():
7      return "Hello from Cloud Run, HELLO FURKAN! GÜNCELLEME 2"
8
9  @app.get("/health")
10 def health():
11     return {"status": "ok"}
12
13 if __name__ == "__main__":
14     # Cloud Run, PORT environment variable kullanır
15     import os
16     port = int(os.environ.get("PORT", "8080"))
17     app.run(host="0.0.0.0", port=port)
18

```

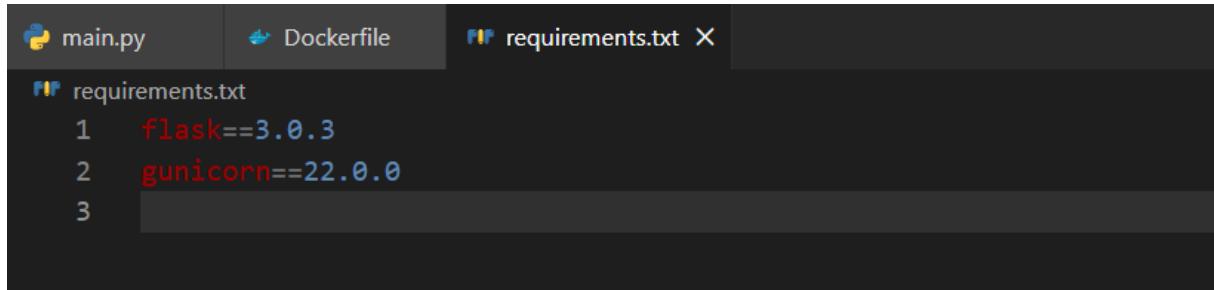
4. Konteynerleştirme (Dockerfile ve requirements.txt)

4.1 requirements.txt Oluşturulması

Uygulamanın bağımlılıklarını belirtmek için proje kök dizinine requirements.txt eklenmiştir. Bu dosyada Flask ve gunicorn bağımlılıkları tanımlanmıştır.

Örnek içerik:

- Flask
- gunicorn

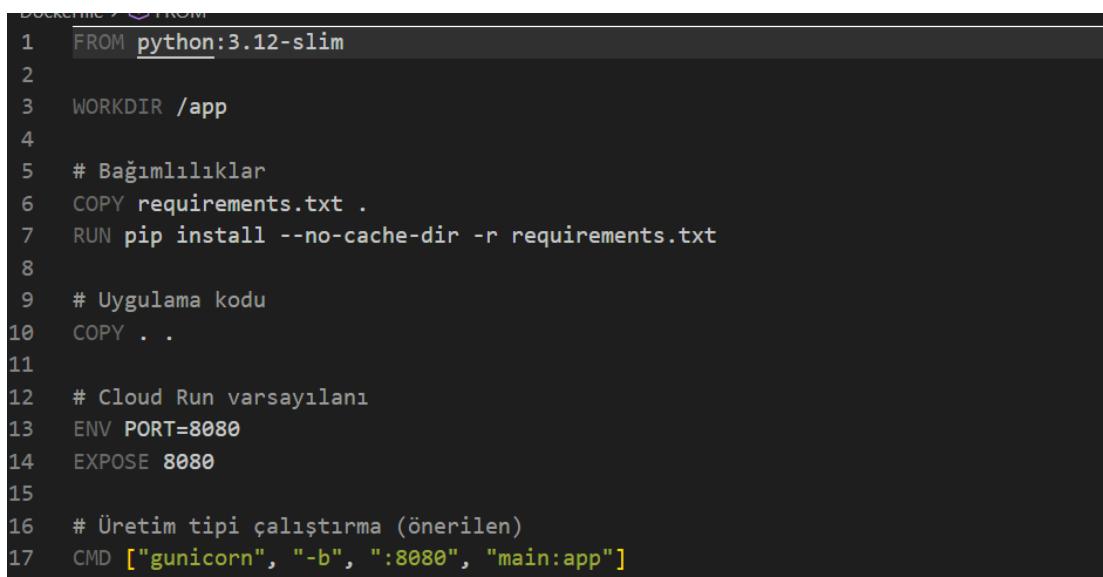


```
requirements.txt
1 flask==3.0.3
2 gunicorn==22.0.0
3
```

4.2 Dockerfile Oluşturulması

Uygulamanın konteyner olarak çalıştırılabilmesi için proje kök dizinine uzantısız şekilde Dockerfile dosyası eklenmiştir. Dockerfile aşağıdaki işlemleri yapmaktadır:

1. Python taban imajını kullanır (`python:3.12-slim`)
2. Çalışma dizinini ayarlar
3. requirements.txt üzerinden bağımlılıkları kurar
4. Uygulama kodlarını kopyalar
5. 8080 portunu açar
6. Uygulamayı gunicorn ile başlatır (`main:app`)

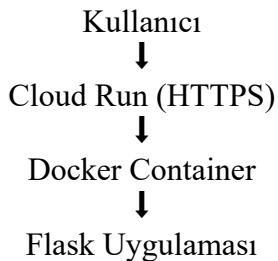


```
Dockerfile
1 FROM python:3.12-slim
2
3 WORKDIR /app
4
5 # Bağımlılıklar
6 COPY requirements.txt .
7 RUN pip install --no-cache-dir -r requirements.txt
8
9 # Uygulama kodu
10 COPY ..
11
12 # Cloud Run varsayılanı
13 ENV PORT=8080
14 EXPOSE 8080
15
16 # Üretim tipi çalıştırma (önerilen)
17 CMD ["gunicorn", "-b", ":8080", "main:app"]
```

5. Uygulama Mimarisi

Aşağıda uygulamanın genel mimari yapısı açıklanmıştır:

Mimari Akış:



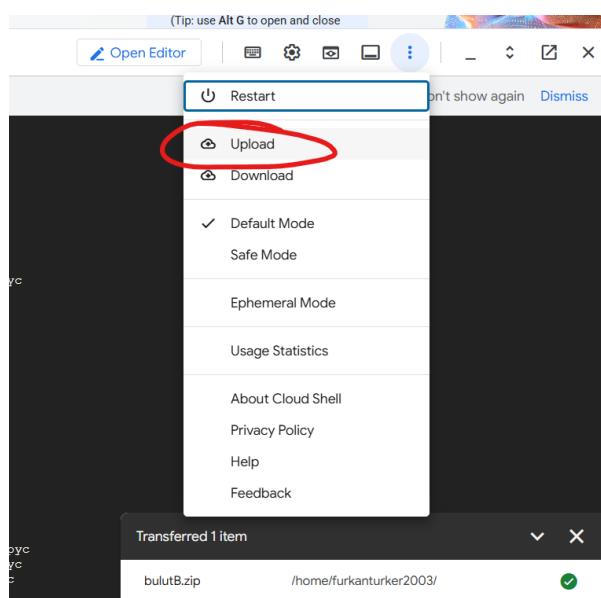
Uygulama mimarisi, modern bulut mimarilerine uygun olacak şekilde aşağıdaki bileşenlerden oluşmaktadır:

- Kullanıcıdan gelen HTTP istekleri Cloud Run tarafından karşılanır.
- Cloud Run, ilgili Docker container'ı çalıştırır.
- Flask uygulaması isteği işler.
- Yanıt tekrar kullanıcıya iletilir.

6. Kaynak Kodun Cloud Shell'e Aktarılması

Docker kurulumu yerel ortamda bulunmadığı için build işlemleri Google Cloud üzerinde yapılmıştır. Bu kapsamda:

1. Proje klasörü zip hâline getirilmiştir.
2. Google Cloud Console üzerinden **Cloud Shell** açılmıştır.
3. Zip dosyası Cloud Shell'e yüklenmiş ve açılmıştır.
4. Build işlemi doğru dizinde yapılacak şekilde proje klasörüne geçilmiştir.



```

creating: bulutB/bulutB/.venv/Scripts/
inflating: bulutB/bulutB/.venv/Scripts/activate
inflating: bulutB/bulutB/.venv/Scripts/activate.bat
inflating: bulutB/bulutB/.venv/Scripts/Activate.ps1
inflating: bulutB/bulutB/.venv/Scripts/deactivate.bat
inflating: bulutB/bulutB/.venv/Scripts/pip.exe
inflating: bulutB/bulutB/.venv/Scripts/pip3.10.exe
inflating: bulutB/bulutB/.venv/Scripts/pip3.exe
inflating: bulutB/bulutB/.venv/Scripts/python.exe
inflating: bulutB/bulutB/.venv/Scripts/pythonw.exe
inflating: bulutB/bulutB/Dockerfile
inflating: bulutB/bulutB/main.py
inflating: bulutB/bulutB/requirements.txt
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$ ls
bulutB
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$ ^C
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$

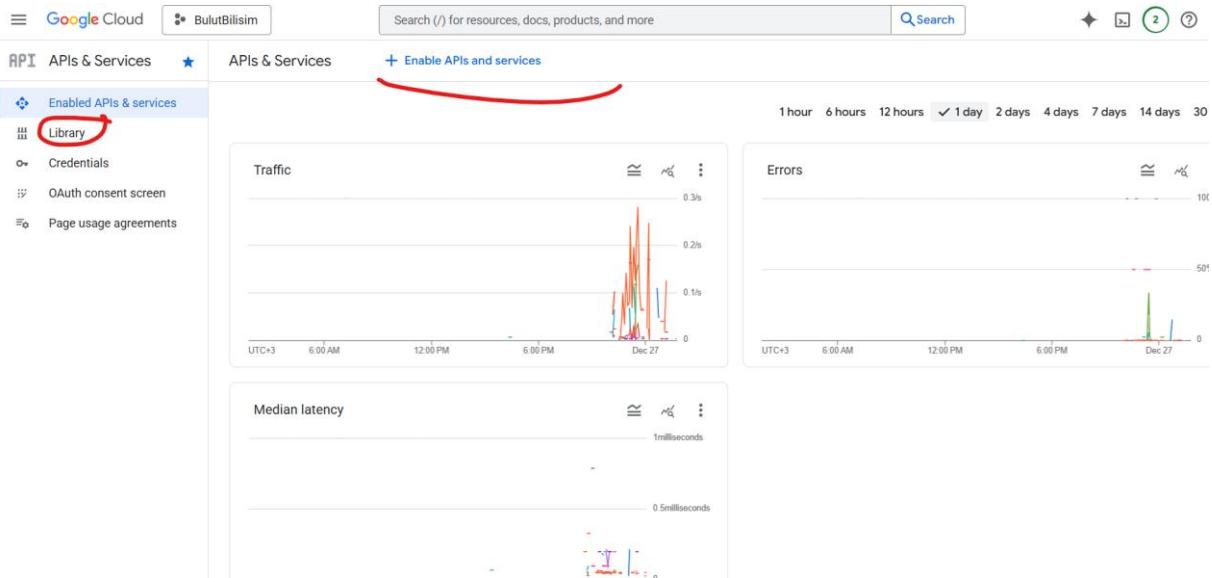
```

7. Google Cloud Build ile Container Image Üretilmesi

7.1 Gerekli API'lerin Aktifleştirilmesi

Cloud Run, Cloud Build ve Artifact Registry servislerinin kullanılabilmesi için ilgili API'ler etkinleştirilmiştir:

- Cloud Run API
- Cloud Build API
- Artifact Registry API



7.2 Artifact Registry Repository Oluşturulması

Container imajlarının saklanacağı repository oluşturulmuştur. Bölge olarak europe-west1 kullanılmıştır. Öğrenmek için hem Cloud Shell üzerinden hemde Google Cloud Build servisi kullanılarak Docker imajı oluşturulmuş ve Artifact Registry'ye gönderilmiştir.

```
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$ gcloud config get-value project  
Your active configuration is: [cloudshell-12785]  
bulutbilisim-482017  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$
```

```
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$ REGION=europe-west1  
gcloud config set run/region $REGION  
Updated property [run/region].  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$
```

```
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$ gcloud services enable run.googleapis.com cloudbuild.googleapis.com artifactregistry.googleapis.com  
Operation "operations/acat/p2-85130803826-3d3b808e-3ec9-40f6-8ff8-ed304e153785" finished successfully.  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$
```

```
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$ gcloud artifacts repositories create webrepo \  
--repository-format=docker \  
--location=$REGION  
Create request issued for: [webrepo]  
Waiting for operation [projects/bulutbilisim-482017/locations/europe-west1/operations/a6da9147-f2e9-48a8-b66b-a49d9043e244] to complete...working
```

HATA

```
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$ gcloud artifacts repositories create webrepo \  
--repository-format=docker \  
--location=$REGION  
Create request issued for: [webrepo]  
Waiting for operation [projects/bulutbilisim-482017/locations/europe-west1/operations/a6da9147-f2e9-48a8-b66b-a49d9043e244] to complete...done.  
Created repository [webrepo].  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$ gcloud builds submit \  
--tag $REGION-docker.pkg.dev$(gcloud config get-value project)/webrepo/flask-cloudrun:1.0 \  
.  
Your active configuration is: [cloudshell-12785]  
ERROR: (gcloud.builds.submit) Invalid value for [source]: Dockerfile required when specifying --tag  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$
```

Yanlış dizinde olduğum için hata aldım.:

```
/home/furkanturker2003/bulutB  
total 12  
drwxrwxr-x 3 furkanturker2003 furkanturker2003 4096 Dec 24 12:19 .  
drwxr-x--- 7 furkanturker2003 furkanturker2003 4096 Dec 24 12:19 ..  
drwxrwxrwx 3 furkanturker2003 furkanturker2003 4096 Dec 24 11:51 .gitlab-ci.yml  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$ find . -maxdepth 3 -iname "Dockerfile"  
./bulutB/Dockerfile  
furkanturker2003@cloudshell:~/bulutB (bulutbilisim-482017)$ cd bulutB  
ls -la  
total 24  
drwxrwxrwx 3 furkanturker2003 furkanturker2003 4096 Dec 24 11:51 .  
drwxrwxr-x 3 furkanturker2003 furkanturker2003 4096 Dec 24 12:19 ..  
-rw-rw-rw- 1 furkanturker2003 furkanturker2003 315 Dec 24 11:47 Dockerfile  
-rw-rw-rw- 1 furkanturker2003 furkanturker2003 370 Dec 22 17:01 main.py  
-rw-rw-rw- 1 furkanturker2003 furkanturker2003 32 Dec 22 17:02 requirements.txt  
drwxrwxrwx 5 furkanturker2003 furkanturker2003 4096 Dec 22 17:03 .venv  
furkanturker2003@cloudshell:~/bulutB/bulutB (bulutbilisim-482017)$
```

Hata

Sorun
Yanlış
dizindiriz
Daha
dizine
götürük

7.3 Cloud Build Submit ile İmajın Oluşturulması

Dockerfile kullanılarak Cloud Build üzerinden imaj üretilmiş ve Artifact Registry'ye push edilmiştir. Build'in başarılı olduğuna dair "STATUS: SUCCESS" çıktısı alınmıştır. Aynı zamanda denemek için diğer oluşturulan imaj Cloud Run servisine deploy edilmiştir. Deploy sonrası servis otomatik olarak çalışır duruma gelmiştir.

```
furkanturker2003@cloudshell:~/bulutB/bulutB (bulutbilisim-482017)$ gcloud run deploy flask-cloudrun \
--image europe-west1-docker.pkg.dev/bulutbilisim-482017/webrepo/flask-cloudrun:1.0 \
--platform managed \
--region europe-west1 \
--allow-unauthenticated
```

```
# Üretim tipi çalıştırma (önerilen)
CMD ["gunicorn", "-b", "8080", "main:app"]
furkanturker2003@cloudshell:~/bulutB/bulutB (bulutbilisim-482017)$ echo $REGION
europe-west1
furkanturker2003@cloudshell:~/bulutB/bulutB (bulutbilisim-482017)$ gcloud builds submit \
--tag $REGION-docker.pkg.dev/$(gcloud config get-value project)/webrepo/flask-cloudrun:1.0 \
.

Your active configuration is: [cloudshell-12785]
Creating temporary archive of 1478 file(s) totalling 18.3 MiB before compression.
Uploading tarball of [...] to [gs://bulutbilisim-482017_cloudbuild/source/1766579730.194231-62f129e712a6483f8038ca3827fcebcf.tgz]
Created [https://cloudbuild.googleapis.com/v1/projects/bulutbilisim-482017/locations/global/builds/0846c37d-86d4-465b-998b-64f51cf72965?project=85138083826]
Logs are available at [ https://console.cloud.google.com/cloud-build/builds/0846c37d-86d4-465b-998b-64f51cf72965?project=85138083826
Waiting for build to complete. Polling interval: 1 second(s).
```

----- REMOTE BUILD OUTPUT -----

```
644f9c869502: Pushed
77a2b55fbe8b: Pushed
1.0: digest: sha256:491f854db50bbfa13f15646e33c30f1ce37b6769a0f003999d2c1234dc448ec9 size: 1995
DONE
-----
ID: 0846c37d-86d4-465b-998b-64f51cf72965
CREATE_TIME: 2025-12-24T12:35:14Z
DURATION: 43S
SOURCE: gs://bulutbilisim-482017_cloudbuild/source/1766579730.194231-62f129e712a6483f8038ca3827fcebcf.tgz
IMAGES: europe-west1-docker.pkg.dev/bulutbilisim-482017/webrepo/flask-cloudrun:1.0
STATUS: SUCCESS
furkanturker2003@cloudshell:~/bulutB/bulutB (bulutbilisim-482017)$
```

8. Cloud Run Üzerine Deploy (Yayınlama)

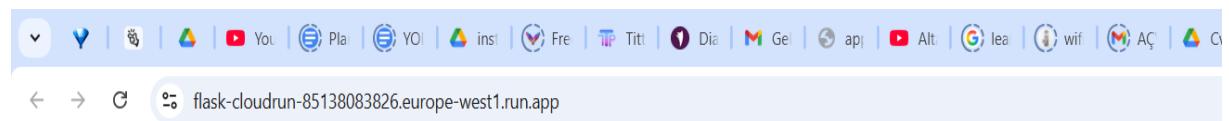
8.1 Cloud Run Servisi Oluşturma ve Deploy

Üretilen container image kullanılarak Cloud Run servisi oluşturulmuş ve internet üzerinden erişilebilecek şekilde yayınlanmıştır.

Deploy sonrası servis için bir URL üretilmiş ve testlerde kullanılmıştır.

```
furkanturker2003@cloudshell:~/bulutB/bulutB (bulutbilisim-482017)$ curl https://flask-cloudrun-85138083826.europe-west1.run.app
Hello from Cloud Run!furkanturker2003@cloudshell:~/bulutB/bulutB (bulutbilisim-482017)$ curl https://flask-cloudrun-85138083826.europe-west1.run.app/health
{"status":"ok"}
furkanturker2003@cloudshell:~/bulutB/bulutB (bulutbilisim-482017)$
```

The screenshot shows the Google Cloud Run interface. On the left, there's a sidebar with links: Overview, Services (selected), Jobs, Worker pools, and Domain mappings. The main area displays a service named 'flask-cloudrun' in the 'europe-west1' region. The URL is <https://flask-cloudrun-85138083826.europe-west1.run.app>. The service status is 'Deploying revision' with a progress bar at 100%. Below this, it says 'Updating service' and 'Creating revision' are completed. The 'Scaling' is set to 'Auto (Min: 0, Max: 3)'. The 'Observability' tab is selected, showing metrics for 'Request count' and 'Request latencies' over the last day. The request count chart shows a sharp spike around December 27th, reaching approximately 0.01/s. The latency chart shows a similar spike, with 95% of requests taking about 0.005s.



8.2 Uygulama Doğrulama Testleri

Uygulamanın doğru çalıştığını doğrulamak için aşağıdaki kontroller yapılmıştır:

- Ana sayfa (/) kontrolü
- Sağlık kontrolü (/health) endpoint kontrolü

Sonuç olarak /health endpoint'inden `{"status": "ok"}` çıktısı alınmış ve servis “yayında” olarak doğrulanmıştır.

```
furkanturker2003@cloudshell:~/bulutB/bulutB (bulutbilisim-482017)$ curl https://flask-cloudrun-85138083826.europe-west1.run.app
Hello from Cloud Run!
furkanturker2003@cloudshell:~/bulutB/bulutB (bulutbilisim-482017)$ curl https://flask-cloudrun-85138083826.europe-west1.run.app/health
{"status": "ok"}
furkanturker2003@cloudshell:~/bulutB/bulutB (bulutbilisim-482017)$
```

9. Revision Yönetimi ve Güncelleme Süreci

Cloud Run her deploy işleminde yeni bir **revision** oluşturur. Bu sayede:

- Her sürüm ayrı bir revision olarak saklanır
- Geri dönüş (rollback) mümkündür

Revision listesi incelenerek yeni deployların revision olarak oluştugu doğrulanmıştır.

The screenshot shows the Cloud Run console for the 'flask-cloudrun' service. The 'Revisions' tab is selected. A table lists eight revisions, with the first one, 'flask-cloudrun-00008-5g6', highlighted as the active revision. The revision details pane on the right provides deployment history (1 hour ago), configuration (Container, Networking, Security), and runtime metrics (General, Billing, Startup CPU boost, Concurrency, Request timeout, Execution environment). The URL for this revision is https://flask-cloudrun-85138083826.europe-west1.run.app.

Name	Traffic	Deployed	Actions
flask-cloudrun-00008-5g6	100% (to latest)	1 hour ago	⋮
flask-cloudrun-00007-bg7	0%	2 hours ago	⋮
flask-cloudrun-00006-nwt	0%	2 hours ago	⋮
flask-cloudrun-00005-tjm	0%	3 hours ago	⋮
flask-cloudrun-00004-hc6	0%	3 hours ago	⋮
flask-cloudrun-00003-2qv	0%	3 days ago	⋮
flask-cloudrun-00002-ksm	0%	3 days ago	⋮
flask-cloudrun-00001-47b	0%	3 days ago	⋮

9.1 Uygulama Güncelleme Mantığı

Cloud Run üzerinde kod değişikliği “anlık” yansımaz; çünkü Cloud Run çalıştığı bileşen **container image**'dır. Bu nedenle güncelleme süreci şu şekildedir:

1. Kaynak kod güncellenir
2. Yeni imaj build edilir (Cloud Build)
3. Cloud Run servisi yeni imajla tekrar deploy edilir
4. Yeni revision aktif olur

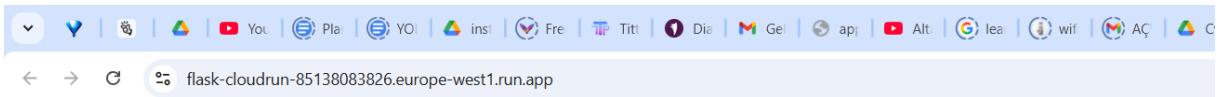
Bu akış kullanılarak uygulama güncelleme mekanizması doğrulanmıştır.

```
furkanturker2003@cloudshell:~/bulutB/bulutB (bulutbilisim-482017)$ Kodda guncelleme
furkanturker2003@cloudshell:~/bulutB/bulutB (bulutbilisim-482017)$ Yeni dekor
furkanturker2003@cloudshell:~/bulutB/bulutB (bulutbilisim-482017)$
furkanturker2003@cloudshell:~/bulutB/bulutB (bulutbilisim-482017)$
furkanturker2003@cloudshell:~/bulutB/bulutB (bulutbilisim-482017)$
furkanturker2003@cloudshell:~/bulutB/bulutB (bulutbilisim-482017)$ REGION=europe-west1
PROJECT_ID=bulutbilisim-482017
TAG=1.1

gcloud builds submit \
--tag $REGION-docker.pkg.dev/$PROJECT_ID/webrepo/flask-cloudrun:$TAG \
.
```

```
furkanturker2003@cloudshell:~/bulutB/bulutB (bulutbilisim-482017)$ gcloud run revisions list --service $SERVICE --region $REGION --project $PROJECT \
✓
REVISION: flask-cloudrun-00002-ksm
ACTIVE: yes
SERVICE: flask-cloudrun
DEPLOYED: 2025-12-24 13:13:53 UTC
DEPLOYED BY: furkanturker2003@gmail.com
) Yeni
✓
REVISION: flask-cloudrun-00001-47b
ACTIVE:
SERVICE: flask-cloudrun
DEPLOYED: 2025-12-24 12:38:45 UTC
DEPLOYED BY: furkanturker2003@gmail.com
furkanturker2003@cloudshell:~/bulutB/bulutB (bulutbilisim-482017)$ gcloud run services describe $SERVICE --region $REGION --project $PROJECT \

```



Hello from Cloud Run! FURKAN TÜRKER

10. IAM ile Erişim Kontrolü (Public → Private)

10.1 Public Erişimin Kapatılması

İlk kurulumda servis herkesin erişimine açıktı (unauthenticated). Güvenlik amacıyla Cloud Run servis izinleri üzerinden public erişim kaldırılmıştır.

Uygulamada “allUsers” yetkisi kaldırılarak anonim erişim engellenmiştir.

The screenshot shows the Google Cloud Platform interface for a Cloud Run service named 'flask-cloudrun'. The service is currently running in the 'europe-west1' region. The URL for the service is <https://flask-cloudrun-85138083826.europe-west1.run.app>. The scaling configuration is set to 'Auto' (Min: 0, Max: 3). On the left sidebar, under the 'Services' section, the 'flask-cloudrun' service is selected. The main content area displays the 'Security' tab of the service configuration. It includes sections for 'Authentication' (with options for public access, authentication via IAM or IAP, and custom audiences), 'Binary Authorization' (disabled), and 'Threat Detection'.

10.2 Kullanıcıya Yetki Verilmesi (Cloud Run Invoker)

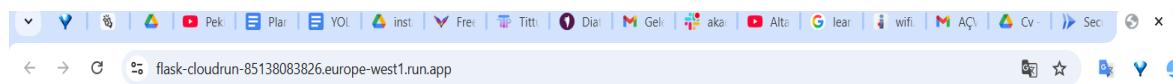
Public erişim kapatıldıktan sonra servise erişebilmek için kullanıcı bazlı yetki verilmiştir. Kendi Google hesabım, **Cloud Run Invoker** rolü ile servise eklenmiştir.

Bu işlem ile yalnızca yetkili kullanıcıların servis URL'sine erişmesi sağlanmıştır.

The screenshot shows the Google Cloud Platform IAM & Admin - Services page. A single service, 'flask-cloudrun', is selected. The right-hand panel shows the 'Assign roles' section for this service. Under 'Add principals', the email 'furkanturker2003@gmail.com' is listed as a new principal. Under 'Assign roles', the 'Role' dropdown is set to 'Cloud Run Invoker', which is described as 'Can invoke Cloud Run services and execute Cloud Run jobs.' At the bottom of the panel are 'Save' and 'Cancel' buttons.

11. URL/Endpoint (Default URL) Sorununun Yaşanması ve Çözümü

IAM ayarları sonrasında servis arayüzünde URL alanının boş göründüğü bir durum oluşmuştur. Bu durumun kaynağı, Cloud Run servisinin **default URL (run.app endpoint)** ayarının devre dışı bırakılması (disable) olmuştur.

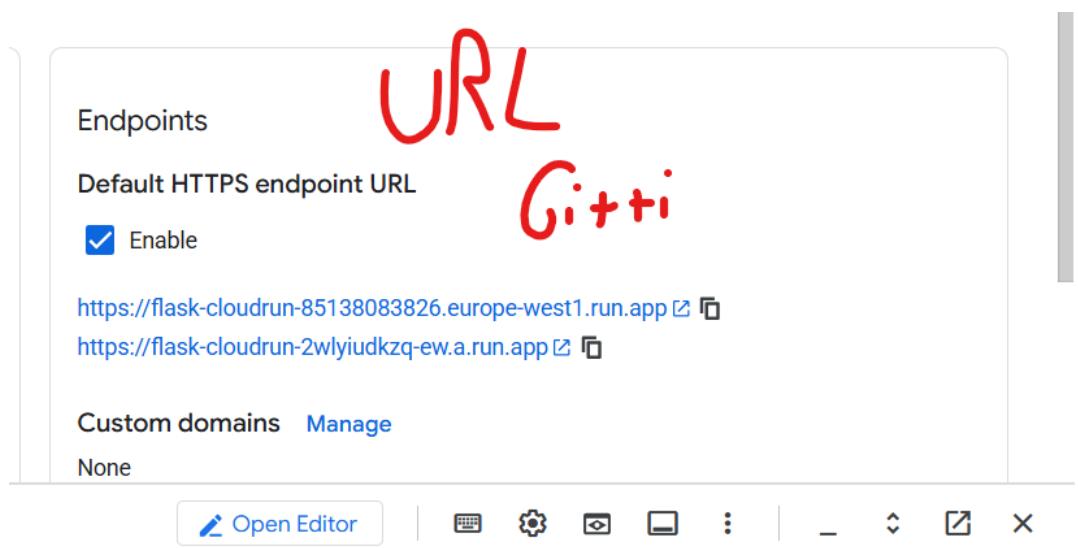


11.1 Sorun Belirtisi

- Cloud Run servis ekranında URL'nin görünmemesi
- Servise erişimde “404 Page not found” benzeri hatalar

11.2 Çözüm

Cloud Run servis ayarlarında default URL tekrar etkinleştirilmiştir (“Enable default endpoint/URL”). İşlem sonrası URL yeniden görünür hâle gelmiş ve endpoint erişimi normale dönmüştür.



Bu bölüm, bulut servislerinde güvenlik yapılandırmalarının servis erişimi üzerindeki etkisini göstermesi açısından raporda önemli bir yer tutmaktadır.

12. Otomatik Ölçeklendirme (Auto Scaling) Yapilandırması

Cloud Run servisi “Auto scaling” mantığıyla çalışacak şekilde yapılandırılmıştır. Yapılan ayarlar:

- Minimum instance: **0**
- Maksimum instance: **3**

Bu sayede:

- Trafik yokken kaynak tüketimi minimize edilir (0 instance)
- Trafik artınca Cloud Run otomatik olarak instance sayısını artırır

The screenshot shows the Cloud Run interface for deploying a revision named 'flask-cloudrun' to the 'europe-west1' region. Under the 'Second generation' section, the 'Revision scaling' settings are displayed. The 'Minimum number of instances' is set to 0 and the 'Maximum number of instances' is set to 3. A note below states: 'The service minimum instances is preferable for most use-cases. Only use this setting if you specifically require per-revision settings.' There is also a checked checkbox for 'Startup CPU boost' with a link to learn more.

Deploy revision to flask-cloudrun (europe-west1)

Show command line

Second generation
Network file system support, full Linux compatibility, faster CPU and network performance.

Revision scaling

Minimum and maximum numbers of instances for the new revision.

Minimum number of instances: 0

Maximum number of instances: 3

The service minimum instances is preferable for most use-cases. Only use this setting if you specifically require per-revision settings.

Startup CPU boost
Start containers faster by allocating more CPU during startup time. [Learn more](#)

Cloud SQL connections

13. Service Account Oluşturma ve Cloud Run'a Atama (Minimum Yetki)

13.1 Service Account Oluşturulması

Varsayılan hesap yerine, uygulamanın kimliğini izole etmek ve minimum yetki prensibini uygulamak amacıyla özel bir Service Account oluşturulmuştur.

The screenshot shows the Cloud Run service configuration for a revision named 'flask-cloudrun-00008-5g6'. The 'Security' tab is selected. The 'Identity & encryption' section shows the service account as 'serviceac-deneme@bulutbilisim-482017.iam.gserviceaccount.com'. Other settings include 'Encryption Key' as 'Google-managed' and 'Threat Detection' as 'Disabled'. Below this, there's a section for 'Software supply chain security insights' with a note about safeguarding artifact integrity across the lifecycle.

13.2 Cloud Run Servisine Service Account Atanması

Cloud Run servis ayarları üzerinden, uygulamanın çalıştığı runtime identity varsayılan hesap yerine oluşturulan Service Account olacak şekilde güncellenmiştir.

Service accounts					
+ Create service account					
Delete Manage access Refresh ?					
Service accounts for project "BulutBilisim"					
A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. Learn more about service accounts.					
Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accounts entirely. Learn more about service account organization policies.					
Filter Enter property name or value ? ☰					
Email	Status	Name ↑	Description	Actions	
<input type="checkbox"/> 85138083826-compute@googleapis.com	<input checked="" type="checkbox"/> Enabled	Default compute service account		⋮	
<input type="checkbox"/> serviceac-deneme@bulutbilisim-482017.iam.gserviceaccount.com	<input checked="" type="checkbox"/> Enabled	ServiceAc_deneme	Cloud Run Service Account	⋮	

Bu sayede uygulama, yalnızca gerekli izinlerle çalışacak şekilde yapılandırılmıştır.

14. Logging ve İzleme (Cloud Logging / Metrics)

14.1 Cloud Logging ile Logların İncelenmesi

Cloud Run servisine gelen istekler ve uygulama logları Cloud Logging üzerinden incelenmiştir. Bu sayede:

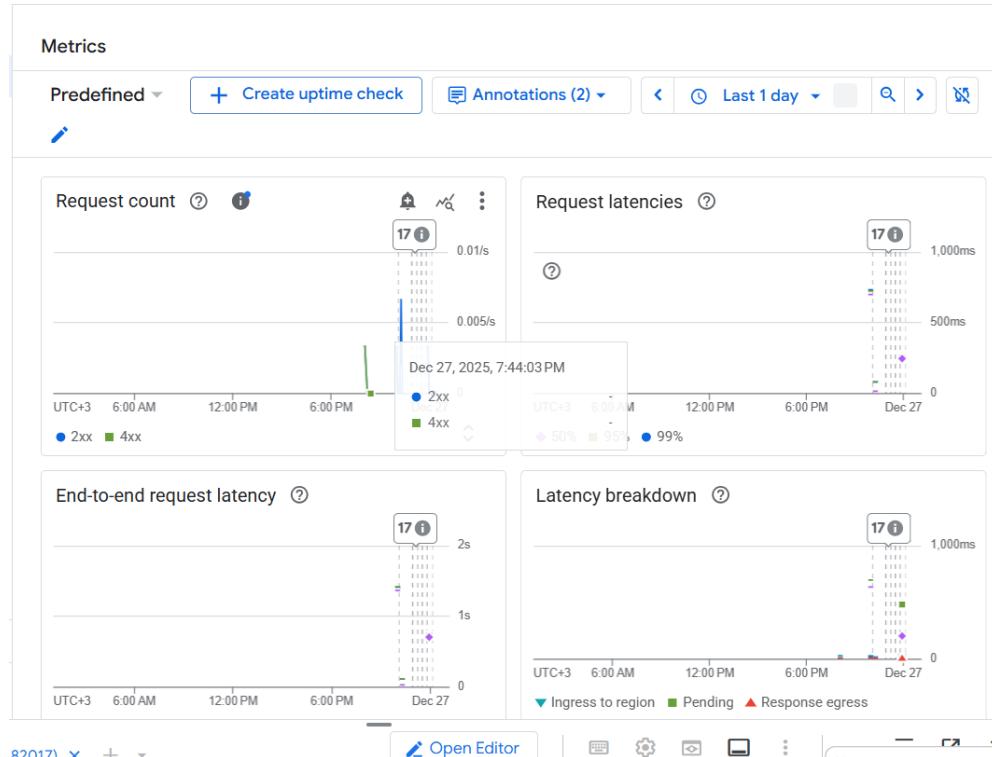
- HTTP istekleri
 - Status kodları
 - Hata mesajları
- gözlemlenebilir hâle gelmiştir.

The screenshot shows the Cloud Run service details page for a service named "flask-cloudrun". The deployment status indicates "Deploying revision" with "Completed" status for "Updating service", "Creating revision", and "Routing traffic". The URL is https://flask-cloudrun-85138083826.europe-west1.run.app. The Observability tab is selected, showing logs, metrics, triggers, networking, security, and YAML sections. The logs section displays several log entries, including startup TCP probe success and worker booting messages.

14.2 Cloud Run Metrics ile Performans İzleme

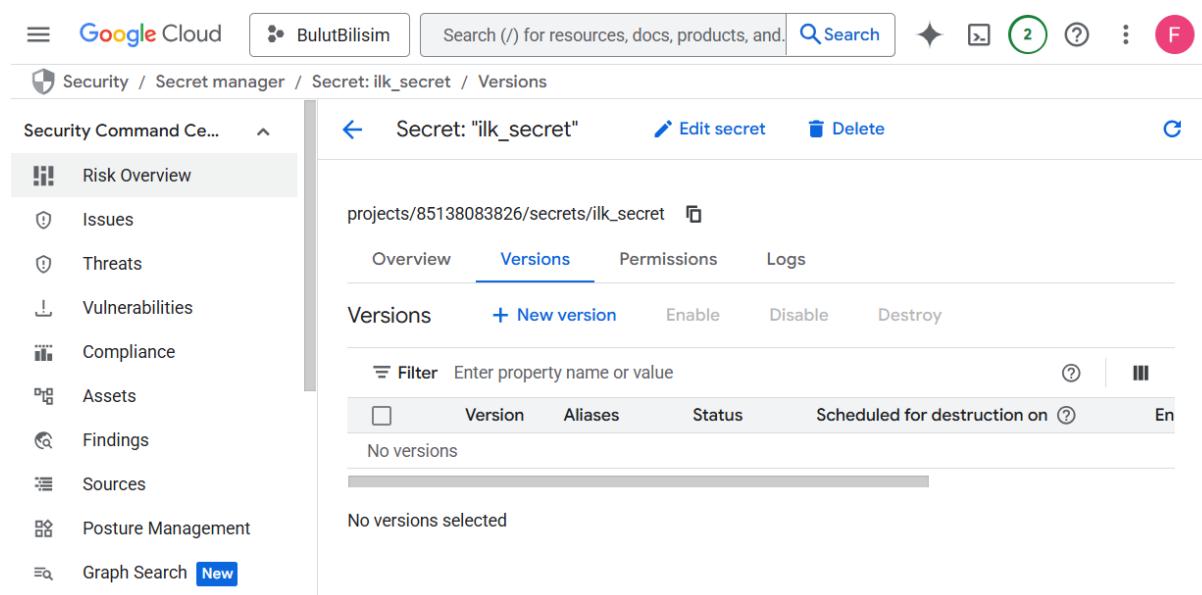
Cloud Run servisinin metrikleri üzerinden:

- Request sayısı
- Gecikme süreleri
- Kaynak kullanımı (CPU/memory) incelenmiştir.



15) Secrets Manager Ekleme

Hassas bilgiler kaynak kodda tutulmamış, Google Secret Manager üzerinden güvenli şekilde yönetilmiştir.



16. Otomasyon İçin Kullanılan Kodlar

Dockerfile

```
FROM python:3.12-slim

WORKDIR /app

COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

ENV PORT=8080
EXPOSE 8080

CMD ["gunicorn", "-b", ":8080", "main:app"]
```

Cloud Build Komutu

```
gcloud builds submit \  
--tag europe-west1-docker.pkg.dev/project-id/webrepo/flask-cloudrun:latest \  
.
```

Cloud Run Deploy Komutu

```
gcloud run deploy flask-cloudrun \  
--image europe-west1-docker.pkg.dev/project-id/webrepo/flask-cloudrun:latest \  
--region europe-west1
```

17. Sonuç ve Değerlendirme

Bu çalışma kapsamında Flask tabanlı web uygulaması konteynerleştirilmiş, Google Cloud Build ile container image üretilmiş ve Google Cloud Run üzerinde yayınlanmıştır. Uygulamanın çalıştığı /health endpoint'i ile doğrulama yapılmıştır. Ayrıca IAM üzerinden public erişim kaldırılmış, kullanıcı bazlı invoker yetkisi verilmiş ve Service Account ataması ile minimum yetki yaklaşımı uygulanmıştır. Cloud Run autoscaling özelliği ile ölçeklendirme yapılandırılmış; Logging ve Metrics kullanılarak izlenebilirlik sağlanmıştır. Son olarak default endpoint (URL) ayarları ile ilgili yaşanan sorun tespit edilmiş ve çözülmüştür.

18. Öğrenilen Dersler

Bu proje kapsamında aşağıdaki kazanımlar elde edilmiştir:

- Cloud Run üzerinde uygulama yayılama
- Docker ile uygulama paketleme
- IAM ile erişim kontrolü sağlama
- Service Account kullanımı
- Bulut ortamında hata ayıklama
- Revision mantığı ve versiyon yönetimi
- Serverless mimarinin avantajları

19. Olası İyileştirmeler

Gelecekte yapılabilecek geliştirmeler:

- CI/CD entegrasyonu (GitHub Actions)
- Custom domain eklenmesi
- HTTPS sertifikası yönetimi
- Uygulamaya veritabanı entegrasyonu
- Load test ve performans analizi

20. Statik Web Sitesinin Bulut Ortamında Yayınlanması

20.1. Statik Web Uygulamasının Tanıtımı

Bu çalışma kapsamında, dinamik Flask uygulamasına ek olarak, HTML, CSS ve JavaScript kullanılarak geliştirilmiş basit bir statik web sitesi de Google Cloud ortamında yayınlanmıştır. Bu web sitesi, herhangi bir sunucu tarafı işlem gerektirmeyen, yalnızca istemci tarafında çalışan bir yapıya sahiptir.

Amaç, farklı türdeki web uygulamalarının (dinamik ve statik) bulut ortamında nasıl yayınlanabileceğini göstermek ve Google Cloud servislerinin bu konudaki esnekliğini ortaya koymaktır.

20.2. Statik Web Sitesinin Yayınlanması Yöntemi

Statik web sitesi, **Google Cloud Storage** kullanılarak yayınlanmıştır.

Cloud Storage, yalnızca dosya depolamak için değil, aynı zamanda statik web sitelerinin barındırılması için de kullanılabilir.

Bu yöntem tercih edilmiştir çünkü:

- Sunucu kurulumu gerektirmez
- Düşük maliyetlidir
- Basit ve hızlıdır
- Küçük ölçekli web projeleri için idealdir

20.3. Cloud Storage Üzerinde Yayınlama Adımları

Aşağıdaki adımlar izlenerek statik web sitesi yayınlanmıştır:

1. Cloud Storage Bucket Oluşturulması

Google Cloud Console üzerinden bir Cloud Storage bucket oluşturulmuştur.

Bucket oluşturulurken:

- Bölge olarak **europe-west1**
- Erişim tipi olarak **public erişime izin verecek şekilde** ayarlar yapılmıştır.

The screenshot shows the Google Cloud Storage Buckets page. The left sidebar has 'Cloud Storage' selected under 'Storage Intelligence'. The main area shows a table of buckets:

Name	Created	Location type	Location	Default storage class	Last modified	Public
bulutbilisim-482017_cloudbuild	Dec 24, 2025, 3:35:31 PM	Multi-region	us	Standard	Dec 24, 2025, 3:35:31 PM	Subje
lokanta_static_site14	Dec 27, 2025, 12:51:05 PM	Multi-region	eu	Standard	Dec 27, 2025, 12:51:05 PM	Not p

2. Statik Web Site Ayarlarının Yapılması

Bucket yapılandırma ekranından:

- **Static website hosting** aktif edilmiştir
- Ana sayfa olarak index.html tanımlanmıştır

Bu ayar sayesinde bucket, bir web sunucusu gibi davranış göstermek üzere yapılandırılmıştır.

The screenshot shows the 'Bucket details' page for the 'lokanta_static_site14' bucket. The left sidebar has 'Objects' selected. The main area shows a tree view of the bucket's contents under 'Lokanta-Websitesi-main'. The tree includes 'lokanta_static_site14' at the root, which contains 'Lokanta-Websitesi-main/' and three sub-folders: 'images/', 'js/', and 'styles/'. The 'images/' folder contains files like 'about.html', 'bootstrap.html', 'contact.html', and 'galery.html'. The 'js/' folder contains 'index.html'. The 'styles/' folder contains 'menu.html', 'review.html', and another 'styles/' folder. A table below lists these objects by name, size, type, creation date, and storage class. On the right, there are tabs for 'Edit bucket' and 'Go to path'. A blue bar at the bottom indicates 'Uploading 45 files' with a progress bar and a list of files: 'about.html', 'bootstrap.html', and 'contact.html'.

3. Web Dosyalarının Yüklenmesi

GitHub üzerinde bulunan:

- index.html
- style.css
- script.js

dosyaları Cloud Storage bucket içeriğine yüklenmiştir.

4. Public Erişim Yetkisinin Verilmesi

Web sitesinin internet üzerinden erişilebilir olması için:

- allUsers kullanıcısına
- **Storage Object Viewer** yetkisi verilmiştir

Bu işlem ile dosyalar herkese açık hâle getirilmiştir.

The screenshot shows the 'Bucket details' page for 'lokanta_static_site14'. In the 'Permissions' section, under 'Grant access', there is a note: 'This bucket is not publicly accessible. If you know objects should never be exposed on the public internet, you should prevent public access to this bucket.' Below this is a 'Prevent public access' button. The 'New principals' field contains 'allUsers'. Under 'Assign roles', the 'Role' dropdown is set to 'Storage Object Viewer'. A warning message states: 'Adding "allUsers" makes this resource public and accessible to anyone on the internet.'

20.4. Web Sitesine Erişim

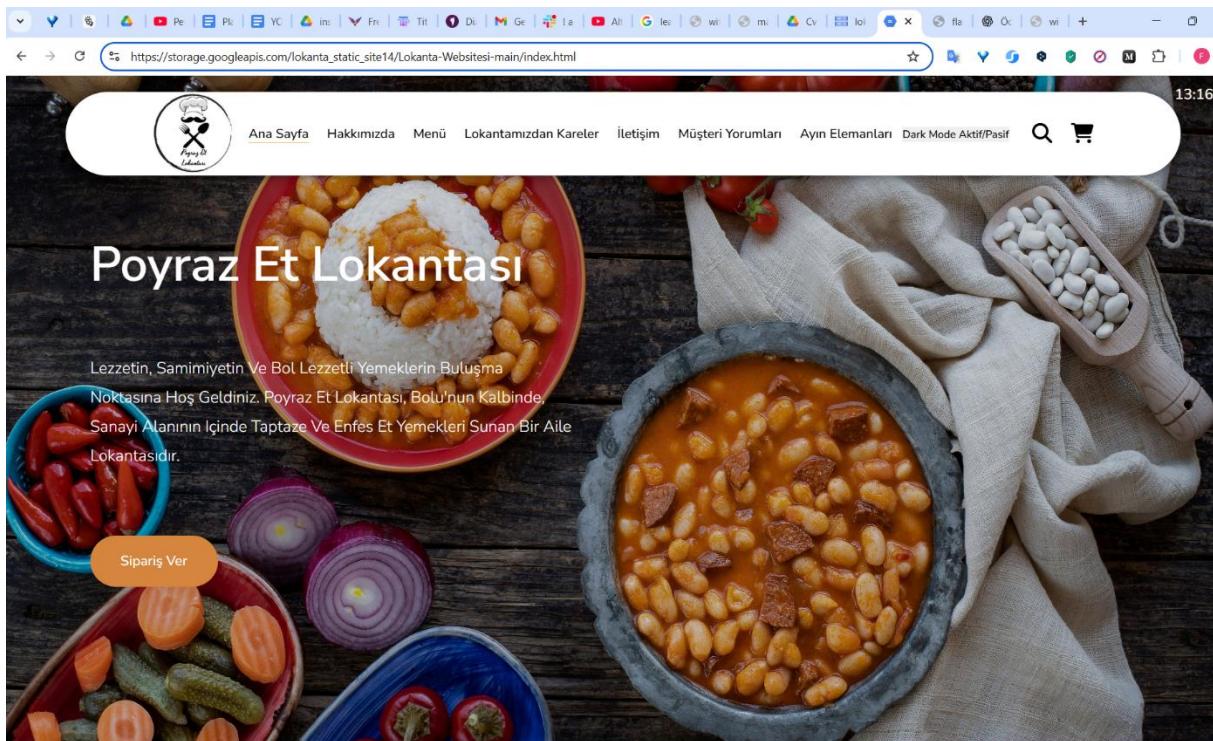
Yapilandırma tamamlandıktan sonra statik web sitesi aşağıdaki URL üzerinden erişilebilir hâle gelmiştir:

<http://<bucket-adı>.storage.googleapis.com>

Bu adres üzerinden:

- Ana sayfa görüntülenebilmekte
- CSS ve JavaScript dosyaları düzgün şekilde çalışmaktadır
-

The screenshot shows the 'Object details' page for 'index.html' in the 'lokanta_static_site14' bucket. The 'Overview' section shows the file type as 'text/html', size as '14.6 KB', and creation date as 'Dec 27, 2025, 1:04:34PM'. It also displays the 'Public URL' as https://storage.googleapis.com/lokanta_static_site14/Lokanta-Websitesi-main/index.html. The 'Permissions' section indicates 'Access granted to public principals'.



20.5. Yapılan Çalışmanın Katkısı

Bu çalışma ile:

- Statik ve dinamik web uygulamalarının farkı uygulamalı olarak görülmüştür
- Google Cloud Storage’ın web hosting amacıyla kullanılabilıldığı gösterilmiştir
- Cloud Run ile çalışan dinamik uygulamaya ek olarak statik bir web sitesi de bulut ortamında yayınlanmıştır
- Bulut servislerinin farklı kullanım senaryoları öğrenilmiştir

Bu sayede proje, yalnızca tek bir uygulamadan oluşmak yerine, farklı mimarileri kapsayan kapsamlı bir bulut çalışması hâline gelmiştir.

21. Kaynakça

- <https://console.cloud.google.com/>
- <https://cloud.google.com/>
- <https://gemini.google.com/?hl=tr>
- <https://coderspace.io/sozluk/google-cloud-platform/>