

DevOps Teknolojileri ile Yazılım Geliştirme

İsim - Soy isim: Habib Şako

Öğrenci No: 02200201011

Giriş:

DevOps, yazılım geliştirme (Development) ve işletim (Operations) süreçlerini bütünleştiren bir kültür, yaklaşım ve uygulama setidir. Temel amacı, yazılımın daha hızlı, güvenilir ve sürekli bir şekilde teslim edilmesini sağlamaktır. Geleneksel olarak, yazılım geliştirme ve işletim ekipleri farklı süreçler, araçlar ve hedeflere sahip olabilirler; ancak DevOps, bu ekipler arasındaki iş birliğini artırarak, otomasyonu teşvik ederek ve sürekli geri bildirim döngülerini güçlendirerek bu ayrımı azaltmaya çalışır.

DevOps, yazılımın planlama, geliştirme, test etme, sürüm ve izleme süreçlerini kapsar. Bu yaklaşım, hızlı bir şekilde yenilik yapmayı, müşteri geri bildirimlerini hızlı bir şekilde ele almayı ve geliştirme ile işletim arasındaki iletişimi ve iş birliğini güçlendirmeyi hedefler. Bu sayede, yazılım projeleri daha hızlı ve daha güvenilir bir şekilde hayata geçirilebilir.

Projede kullandığım teknolojiler:

- **IDE:** Visual Studio Code (Eklentiler: Docker, Kubernetes, Synk)
- **Git Repo:** Github
- **Code Scan:** Snyk
- **Image:** Docker File
- **Image Registry:** DockerHub
- **Image Scan:** Snyk
- **Kubernetes Cluster:** Kubernetes
- **CI/CD:** Jenkins
- **Monitoring:** Prometheus, Grafana
- **Diğer teknolojiler:** Helm, chocolatey

Not: Jenkins adımlarında snyk taramasında problemi çözemediğim için Fail the build if errors occur seçeneğindeki seçili tiki kaldırmak zorunda kaldım. Snyk taramasını hem adımların en başında hem de docker push ve login işlemlerinden önceye koydum.

Özet:

Yazmış olduğum kod Flask ve Flask-RESTful kütüphanelerini kullanarak bir basit API (Application Programming Interface) uygulamasını oluşturuyor. Kodun temel amacı, / endpoint'ine yapılan HTTP GET isteğine belirli bir JSON yanıtı vermektir. Bu yanıt, farklı kategoriler altında bulunan aldığım dersleri ve derslere ait bazı terimleri göstermektedir.

/endpoint'i tüm kategorilere ait bilgileri döndürürken, /<string:category_id> endpoint'i sadece belirli bir kategoriye ait bilgileri döndürmektedir. Örneğin, /1 endpoint'i Bulut Bilişim kategorisinin bilgilerini verecek şekildedir. Aşağıda kodu görmektesiniz:

```
C:\Users\habib\Desktop> cd .\2> .\api.py
1  from flask import Flask
2  from flask_restful import Api, Resource
3
4  app = Flask(__name__)
5  api = Api(app)
6
7  class Product(Resource):
8      def get(self, category_id=None):
9          all_categories = {
10              'Bulut Bilişim': ['Google Cloud', 'Docker', 'Jenkins', 'Snyk', 'DockerHub', 'ArgoCD'],
11              'Oyun Geliştirme': ['Unity', 'C#', 'Hyper Casual Game'],
12              'Açık Kaynak Yazılım Geliştirme': ['Git ve Github', 'NGINX', 'Flask', 'API'],
13              'Büyük Veri Teknolojileri': ['Hadoop', 'Hive', 'HBASE', 'MONGODB']
14          }
15
16          if category_id is None:
17              return all_categories
18          elif category_id in all_categories:
19              return {category_id: all_categories[category_id]}
20          else:
21              return {'error': 'Kategori bulunamadı.'}
22
23  api.add_resource(Product, '/', '/<string:category_id>')
24
25  if __name__ == '__main__':
26      app.run(host="0.0.0.0", port=80, debug=True)
27
```

GitHub Repo Oluşturma ve Commit:

GitHub hesabıma bağlamış olduğum Visual Studio Code uygulamamda yazmış olduğum kodu VSCode ile public bir repo oluşturarak kodu gönderdim. Daha sonra yaptığım değişiklikleri commitleyerek projedeki değişiklikleri github'a aktardım. Projemin github linki: <https://github.com/HabibSako/CI-CD-Pipeline/blob/main/ornek.py>

Snyk:

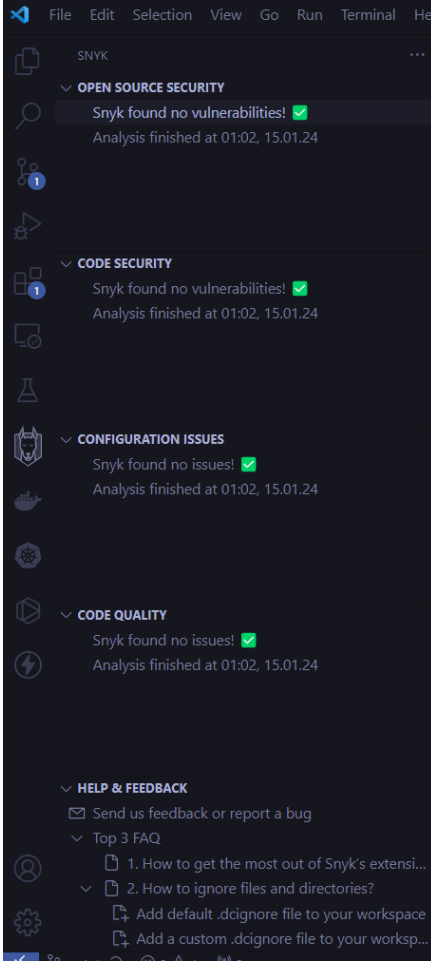
Npm install snyk -g diyerek cmd snyk kurulumunu yaptım ve snyk auth işlemi ile snyk bağladım.

```
C:\Users\habib>npm install snyk -g
[ ] \ reify:snyk: http fetch GET 200 https://registry.npmjs.org/snyk/-/snyk-1.1276.0.tgz 58855ms (cach
```

VSCode ile Snyk Security Eklentisi:

IDE üzerindeki eklentiler kısmında Snyk Security eklentisini IDE'ye ekledim. Sonrasında Snyk kullanabilmek için hesap açtım. Hesap açtıktan sonra VSCode üzerinden yazmış olduğum kodları tarayabildim ve çıktılar aldım. Daha önceden açık kaynak yazılım geliştirme dersinde fork yaptığım ve üzerine yeni kod kısımları eklemiş olduğum projeyi ilerletmek istedim ancak yazdığım koddaki açıkları bazılarını giderebilsem de veri tabanı bağlantısı yapılırken ki açıkları gideremedim. Bahsettiğim bu kod <https://github.com/HabibSako/Rest->

[API/blob/main/docker%2027.12.2023/ornek.py](https://api.github.com/repos/2027.12.2023/ornek.py) adresinde bulunmaktadır. Bundan dolayı yazmış olduğum kodu daha basit bir şekilde indirgeyerek üst başlıktaki git reposuna aktardım. Bu koddaki Snyk taramasından aldığım en son ekran görüntüsü:



Docker Kurulumu:

Dockerı windows cihazıma kurmak için <https://www.docker.com/products/docker-desktop/> adresinden windows için exe dosyasını indirdim ve gerekli adımları takip ederek kurulumu tamamladım. Ayrıca docker desktop sanal makinesinin kurulumunu da yaptım.

Dockerfile Dosyası:

- FROM komutunu kullanarak üzerinde çalışacağım temel image pull ederek eklenmesini sağladım.
- WORKDIR komutunu kullanarak çalışma alanını /app dizini olarak belirttim.
- COPY komutunu kullanarak yerel sistemdeki dosyaların çalışma dizinine “. /app “ kopyalanmasını sağladım. Bu sayede docker image oluşturulurken dosyalar kullanılabilir olacak.
- RUN komutu ile yazılacak olan terminal komutlarını sıraladım.
- EXPOSE komutu ile dış dünyaya bağlantı yapılacak adresi verdim.

- CMD komutu ile docker konteyneri başlatıldığında yapılacak komutu yazdım.

Aşağıda yazmış olduğum Dockerfile ekran görüntüsü bulunmaktadır:

```
Dockerfile > ...
1 FROM python:3.11
2
3 WORKDIR /app
4
5 COPY . /app
6
7 RUN pip install --upgrade pip
8 RUN pip install -r requirements.txt
9
10 EXPOSE 5003
11
12 CMD ["python", "ornek.py"]
13
```

Requirements.txt Dosyası:

Yazmış olduğum api için gerekli kütüphanelerini yazdım. Herhangi bir sürümünü belirtmedim. Bu yüzden otomatik olarak güncel versiyonlarını indirmesi sağladım.

```
requirements.txt
1 Flask
2 flask-restful
```

Docker Compose Dosyası:

- Version komutu ile kullanılan sürümü belirttim.
- Build ile Dockerfile bulunduğu dizini belirttim.
- Command ile Docker konteyneri başlatıldığında çalıştırılacak komutu belirttim.
- Ports ile docker konteynirinin portu ile dış dünyadaki portunu atadım.

```

docker-compose.yml > {} services > {} web > [ ] ports
docker-compose.yml - The Compose specification establishes a standard for the de
1  version: '3.8'
2
3  services:
4    web:
5      build: .
6      command: python ornek.py
7      volumes:
8        - ./app
9      ports:
10     - "5003:5003"

```

Docker Image Oluşturulması ve Container Ayağa Kaldırılması:

Gerekli dosyaları yazdıktan sonra terminale şu komutu yazarak image oluşturdum: [docker build -t habibsako/python-flask-api-project:0.0.1.RELEASE .]. Docker build komutu Docker imajını oluşturmak için kullandım. habibsako/python-flask-api-project 0.0.1.RELEASE etiket atama işlemi yaptım ve sürümünü belirttim. Son kısmındaki nokta ise Dockerfile dosyasının konumunu belirtiyor.

```

habib@Habib MINGW64 ~/Desktop/Yeni klasör/Bulut Bilişim (main)
$ docker build -t habibsako/python-flask-api-project:0.0.1.RELEASE .
[+] Building 21.8s (11/11) FINISHED
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                   0.0s
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 365B                              0.0s
=> [internal] load metadata for docker.io/library/python:3.11   3.0s
=> [auth] library/python:pull token for registry-1.docker.io    0.0s
=> [1/5] FROM docker.io/library/python:3.11@sha256:204d3ef2fd0fcd127c0a3735cd190865e68c3b0b5db2c333eba50a50f7833be1 0.1s
=> => resolve docker.io/library/python:3.11@sha256:204d3ef2fd0fcd127c0a3735cd190865e68c3b0b5db2c333eba50a50f7833be1 0.1s
=> [internal] load build context                                0.0s
=> => transferring context: 3.5kB                                  0.0s
=> CACHED [2/5] WORKDIR /app                                    0.0s
=> [3/5] COPY . /app                                           0.1s
=> [4/5] RUN pip install --upgrade pip                          11.9s
=> [5/5] RUN pip install -r requirements.txt                     6.2s
=> exporting to image                                           0.3s
=> => exporting layers                                           0.3s
=> => writing image sha256:a580215d91e0c9c9021bfd717aa7e82872eac86cb3007facdc47350ca738468 0.0s
=> => naming to docker.io/habibsako/python-flask-api-project:0.0.1.RELEASE 0.0s

What's Next?
View a summary of image vulnerabilities and recommendations -> docker scout quickview

```

Oluşturmuş olduğum image çalıştırmak için 5000 portundan yayın yaparak aşağıdaki ekran görüntüsündeki gibi komut yazdım ve docker konteynirini çalıştırdım:

```

habib@Habib MINGW64 ~/Desktop/Bulut Bilişim (main)
$ docker container run -d -p 5000:5000 habibsako/python-flask-project:0.0.1.RELEASE
c665b0bf9d1e274ce42748e21219302d259aa5c2d3ceac14c99e4d5e7cf337e3

```

Yazmış olduğum docker compose dosyasını build ettim:

```
habib@Habib MINGW64 ~/Desktop/Yeni klasör
$ pwd
/c/Users/habib/Desktop/Yeni klasör

habib@Habib MINGW64 ~/Desktop/Yeni klasör
$ cd Bulut\ Bilişim/

habib@Habib MINGW64 ~/Desktop/Yeni klasör/Bulut Bilişim (main)
$ docker-compose build
[+] Building 98.0s (11/11) FINISHED
=> [web internal] load build definition from Dockerfile
=> => transferring dockerfile: 365B
=> [web internal] load .dockerignore
=> => transferring context: 2B
=> [web internal] load metadata for docker.io/library/python:3.11
=> [web auth] library/python:pull token for registry-1.docker.io
=> [web 1/5] FROM docker.io/library/python:3.11@sha256:204d3ef2fd0fcd127c0a3735cd190865e68c3b0b5db2c333eba50a50f7833be1
=> => resolve docker.io/library/python:3.11@sha256:204d3ef2fd0fcd127c0a3735cd190865e68c3b0b5db2c333eba50a50f7833be1
=> [web internal] load build context
=> => transferring context: 4.08kB
=> CACHED [web 2/5] WORKDIR /app
=> [web 3/5] COPY . /app
=> [web 4/5] RUN pip install --upgrade pip
=> [web 5/5] RUN pip install -r requirements.txt
=> [web] exporting to image
=> => exporting layers
=> => writing image sha256:45224bcd9f5e364efd39fdc44041c0660d375995e28a316bfcd9acc86c71d00
=> => naming to docker.io/library/bulutbiliim-web

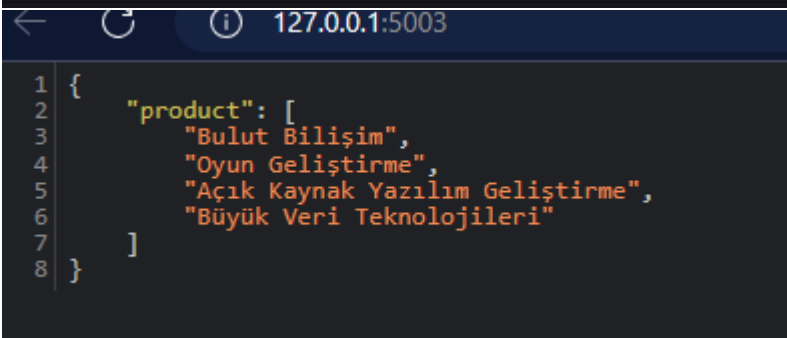
habib@Habib MINGW64 ~/Desktop/Yeni klasör/Bulut Bilişim (main)
$
```

Ve docker compose up diyerek ayağa kaldırdım.

```
habib@Habib MINGW64 ~/Desktop/Yeni klasör/Bulut Bilişim (main)
$ docker-compose up
[+] Running 2/2
  ✓ Network bulutbiliim default    Created
  ✓ Container bulutbiliim-web-1    Created
Attaching to web-1
web-1 | * Serving Flask app 'ornek'
web-1 | * Debug mode: off
web-1 | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
web-1 | * Running on all addresses (0.0.0.0)
web-1 | * Running on http://127.0.0.1:5003
web-1 | * Running on http://172.18.0.2:5003
web-1 | Press CTRL+C to quit
```

Yapmış olduğum işlemlerden sonra terminalde servisi ayağa kaldırmış oldum. Ayrıca 5003 portundan gözüken ekran görüntüsü de şu şekildedir:

```
web-1 | * Running on http://172.18.0.2:5003
web-1 | Press CTRL+C to quit
web-1 | 172.18.0.1 - - [14/Jan/2024 12:50:50] "GET / HTTP/1.1" 200 -
web-1 | 172.18.0.1 - - [14/Jan/2024 12:51:47] "GET / HTTP/1.1" 200 -
web-1 | 172.18.0.1 - - [14/Jan/2024 12:52:10] "GET / HTTP/1.1" 200 -
```



```
1 {
2   "product": [
3     "Bulut Bilişim",
4     "Oyun Geliştirme",
5     "Açık Kaynak Yazılım Geliştirme",
6     "Büyük Veri Teknolojileri"
7   ]
8 }
```

Ayağa kaldırmış olduğum servisi kapatarak dockerHub'a push işlemine geçtim.

```
habib@Habib MINGW64 ~/Desktop/Yeni klasör/Bulut Bilişim (main)
$ docker-compose down
[+] Running 2/2
✓ Container bulutbiliim-web-1 Removed
✓ Network bulutbiliim_default Removed

habib@Habib MINGW64 ~/Desktop/Yeni klasör/Bulut Bilişim (main)
$
```

Önce login terminalden login işlemini ardından push işlemini [docker push habibsako/python-flask-api-project:0.0.1.RELEASE] komutu ile gerçekleştirdim. Ayrıca docker desktop üzerinden image ve containerların ekran görüntüleri:

```
habib@Habib MINGW64 ~/Desktop/Yeni klasör/Bulut Bilişim (main)
$ docker login
Authenticating with existing credentials...
Login Succeeded
```

```
habib@Habib MINGW64 ~/Desktop/Yeni klasör/Bulut Bilişim (main)
$ docker push habibsako/python-flask-api-project:0.0.1.RELEASE
The push refers to repository [docker.io/habibsako/python-flask-api-project]
98dfb2cce62b: Pushed
57212f12405c: Pushed
9fad978a96f4: Pushed
0ff6fffd126da: Pushed
cd14e2bdd6c0: Mounted from library/python
3b61b4c903b3: Mounted from library/python
991f098ab280: Mounted from library/python
3479e3189b94: Mounted from library/python
671f9916da8d: Mounted from library/python
2b952ea61c4b: Mounted from library/python
0dfa23ffffa41: Mounted from library/python
aa904f36746c: Mounted from library/python
0.0.1.RELEASE: digest: sha256:e8e9c2555ac9cd28df06dd78adc4cc61fccc4fed09687bcdcdcafb0dd5038ef size: 2844
```

Containers [Give feedback](#)

Container CPU usage 0.01% / 1000% (10 cores available) Container memory usage 21.49MB / 7.45GB [Show charts](#)

Search Only show running containers

<input type="checkbox"/>	Name ↑	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	xenodochial_jang f88a0b1cd6dd	habibsako/python-flask-project2:0.0.1.RELEASE	Exited (137)	0%	5003:5000	29 minutes ago	▶ ⋮ 🗑
<input type="checkbox"/>	> bulutbiliim		Running (1/1)	0.01%		55 seconds ago	Start ▶ ⋮ 🗑

Search [≡](#)

<input type="checkbox"/>	Name	Tag	Status	Created	Size	Actions
<input type="checkbox"/>	bulutbiliim-web 45224bcd9f5e	latest	Unused	1 second ago	1.03 GB	▶ ⋮ 🗑
<input type="checkbox"/>	habibsako/python-flask-project2 fcd80c994593	0.0.1.RELEASE	In use	3 days ago	1.25 GB	▶ ⋮ 🗑
<input type="checkbox"/>	habibsako/python-flask-project 0f2cf0034b78	0.0.1.RELEASE	In use	3 days ago	1.25 GB	▶ ⋮ 🗑

DockerHub Hesabı ve Push:

DockerHub için oluşturmuş olduğum hesabım ve lokalimde oluşturmuş olduğum image pushladığım reposotiry incelemek için: <https://hub.docker.com/u/habibsako> adresini inceleyebilirsiniz.

The screenshot shows the Docker Hub interface for a repository named 'python-flask-api-project' under the user 'habibsako'. The page has a blue header with the Docker Hub logo and navigation links: 'Explore', 'Repositories' (active), and 'Organizations'. A search bar is on the right. Below the header, the breadcrumb path is 'habibsako / Repositories / python-flask-api-project / General'. The main content area is divided into several sections:

- General** (selected tab): Includes a prompt to 'Add a short description for this repository' with an 'Update' button. Below this, the repository name 'habibsako / python-flask-api-project' is shown. The 'Description' section states 'This repository does not have a description' with an edit icon. The 'Last pushed' time is '11 days ago'.
- Docker commands**: A section titled 'To push a new tag to this repository:' with a 'Public View' button and a code block showing the command: `docker push habibsako/python-flask-api-project:tagname`.
- Tags**: A section stating 'This repository contains 1 tag(s)'. It contains a table with the following data:

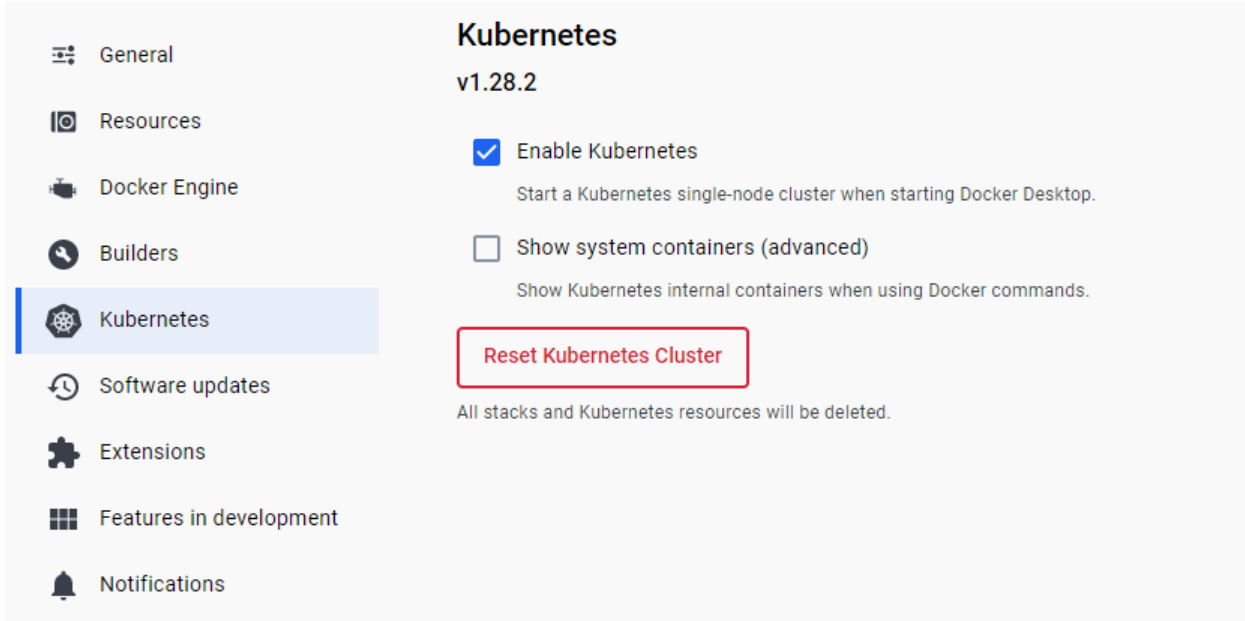
Tag	OS	Type	Pulled	Pushed
0.0.1.RELEASE	linux	Image	4 days ago	11 days ago

A 'See all' link is provided below the table.
- Automated Builds**: A section explaining that manually pushing images to Hub can be automated by connecting to GitHub or Bitbucket. It includes a link to 'Read more about automated builds' and an 'Upgrade' button.

Bu şekilde localde sağlıklı bir şekilde container-based bir uygulamayı çalıştırmış ve ağ üzerinden başarılı bir şekilde erişmiş oldum. Bundan sonraki adımlarda bu süreç otomatikleştirmeyi ve CI/CD sürecini yapmaya çalışacağım.

Kubernetes Kurulumu:

Kubernetes kullanmak için docker desktopta şu şekilde ayarlama yapıp yeniden başlattım. Ayrıca VSCode kubernetes eklentisi kurdum.



Sistem yeniden başladıktan sonra cmd ekranından sürüm kontrollünü yaptım. Ve kurulumu tamamladım.

```
Komut İstemi
C:\Users\habib>kubectl version
Client Version: v1.28.2
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
Server Version: v1.28.2

C:\Users\habib>kubectl get nodes
NAME                STATUS    ROLES          AGE    VERSION
docker-desktop      Ready    control-plane  23h    v1.28.2

C:\Users\habib>
```

Jenkins Kurulumu CI/CD:

Jenkins kurulumu için gerekli olan java jdk 17 indirdim ve kurulumunu yaptım. Jdk linki: <https://www.oracle.com/tr/java/technologies/downloads/#java17>

Daha sonra Jenkins kurulumunu tamamladım ve 8080 portunu atadım. <http://localhost:8080/> adresini açtım ve şu ekranla karşılaştım.

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

`C:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

Gerekli lokasyondan şifreyi alıp giriş yaptım. Önerilen pluginleri indirerek devam ettim. İndirme tamamlandıktan sonraki adımda hesap bilgilerimi oluşturdum ve Jenkins arayüzü karşıma geldi.

Yeni öge oluştur diyerek bir serbest stil proje oluşturdum ve isimlendirdim. Karşıma gelen ekranda github projesi seçeneğini seçerek github repomun adresini bu alana yapıştırdım. Kaynak kodu yönetimini git olarak seçip tekrardan github adresimi yapıştırdım. Branch main olarak düzelttim.

Github Hook Trigger:

Github hook trigger seçeneğini işaretledim ve github repomun ayarlar kısmına giderek webhook kısmını açtım. Aşağıdaki ekran görüntüsündeki gibi doldurdum. Add webhook tıkladım.

Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

Content type

application/json

Secret

Which events would you like to trigger this webhook?

☒ Just the push event.

☐ Send me **everything**.

☐ Let me select individual events.

☒ **Active**

We will deliver event details when this hook is triggered.

Add webhook

Daha sonra Jenkins ekranına dönerek projemi kaydettim. Yaptığım bir github commitinin Jenkins üzerindeki ekran görüntüsü:

</> Değişiklikler

Console Output

Sürüm Bilgisini Düzenle

Delete build '#38'

Timings

Git Build Data

← Önceki Yapılandırma

Summary

1. kod açığı giderildi. ([commit: 3cf3a30](#)) ([details](#))

Commit 3cf3a30977a0760c813cccc69bfc36a89df6caea by Habib Şako

kod açığı giderildi.

([commit: 3cf3a30](#))

[Dockerfile \(diff\)](#)

Snyk ile Jenkins Eklentisi Entegrasyonu:

Ana sayfadaki Jenkins'i yönet kısmına tıklayarak eklentiler bölümüne gittim. Snyk eklentisini indirdim.

İsim

Devrede

Snyk Security Plugin 4.0.2

Add the ability to test your code dependencies for vulnerabilities against Snyk database

[Report an issue with this plugin](#)

☒

☐

Ayarlar kısmından araçlar kısmına gelerek en altta bulunan Snyk Kurulumlarını şu şekilde doldurdum:

Kontrol Merkezi > Jenkins'i Yönet > Araçlar

Snyk kurulumları

Snyk ekle

≡ Snyk

Name

Snyk

☒ Install automatically ?

≡ Install from snyk.io

Version

latest

Update policy interval (hours)

24

OS platform architecture

Auto-detection

Add Installer ▾

Snyk ekle

Kaydet

Uygula

1°C

Kaydet butonuna tıkladım. Daha sonra ayarlar kısmından credential bölümüne girdim. Oradan global credential bölümünde Add Credentials butonuna tıkladım. Sonrasında yeni bir pencereden Snyk hesabıma gidip key bilgisini aldım. Aşağıda Snyk token bölümünün ekran görüntüsü:

Account Settings Log out

General

Notifications

Share With a Friend

Auth Token

Use this token to authenticate the Snyk CLI and in CI/CD pipelines. [Learn more about authenticating CLI in our docs.](#)

KEY	CREATED
<div>click to show</div>	09 January 2024, 00:33:56

Revoke & Regenerate

Aldığım token değeri ile birlikte Add Credentials bölümündeki bilgileri aşağıdaki ekran görüntüsündeki gibi doldurdum.

Update credentials

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Token ?



Concealed

ID ?

Snyk-token

Description ?



Save

Save tıklayarak kurulumu tamamladım.

Global credentials (unrestricted)

+ Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
 Snyk-token	Snyk-token	Snyk API token	

Ana sayfaya dönüp projemi seçtikten sonra konfigürasyon ekle diyerek Snyk şu şekilde adımlara ekledim. Hangi hata durumlarında çalışıp çalışmamasını seçtim. Ayarlamış olduğum ID'yi seçtim. Ekran görüntüsü:

≡

Invoke Snyk Security task

When issues are found ?

☒

 Fail the build, if severity at or above

Let the build continue

☐

☒

 Fail the build if errors occur ?☒

Snyk API token ?

Snyk-token

+ Add ▾

Target file ?

Organisation ?

Project name ?

Snyk installation ?

Snyk

Jenkins Eklentisi - Python:

Yazmış olduğum kod python dilinde olduğundan dolayı jenkins konfigürasyonunda hata almamak için ekledim.

Jenkins Yapılandırma Adımları:

Sırasıyla eklediğim adımlar:

CI süreci:

- Invoke Snyk Security task
- docker build -t habibsako/python-flask-api-project:0.0.1.RELEASE .
- docker container run -d -p 5000:5000 habibsako/python-flask-api-project:0.0.1.RELEASE
- docker-compose build
- docker-compose up

CD Süreci:

- docker login -u habibsako -p <bu alana docker şifremi yazdım>

- docker push habibsako/python-flask-api-project:0.0.1.RELEASE

☰ Windows batch komutu çalıştır ?

Command

See [the list of available environment variables](#)

```
docker build -t habibsako/python-flask-api-project:0.0.1.RELEASE .
```

Gelişmiş ▼

☰ Windows batch komutu çalıştır ?

Command

See [the list of available environment variables](#)

```
docker container run -d -p 5000:5000 habibsako/python-flask-api-project:0.0.1.RELEASE
```

Gelişmiş ▼

☰ Windows batch komutu çalıştır ?











Command

See [the list of available environment variables](#)

```
docker-compose build
```

Jenkins Yapılandırma Sonrası Oluşan Container ve Çıktısı:

Jenkinsde bulunan proje adımlarını tamamladıktan sonra yapılandır tuşuna bastıktan sonra localde docker desktopta oluşan container ve imageler çalıştı ve 5003 nolu porttan dışa yayın yaptı. Bulutbilsim containerı otomatik olarak oluştu ve başarılı bir şekilde çalıştı. Aldığım ekran görüntüleri:

<input type="checkbox"/>	 k8s_argocd-application-controller_argocd-application-contro 1a265730e4db	quay.io/argoproj/argocd	Running	0.04%	41 seconds ago	  
<input type="checkbox"/>	 bulutbilisim		Running (1/1)	0.01%	1 minute ago	  
<input type="checkbox"/>	 web-1 78c86d8fb2d4	bulutbilisim-web	Running	0.01%	5003:5003 1 minute ago	  

```
← ↺ ⓘ localhost:5003 A
1 {
2   "Bulut Bilişim": [
3     "Google Cloud",
4     "Docker",
5     "Jenkins",
6     "Snyk",
7     "DockerHub",
8     "ArgoCD"
9   ],
10  "Oyun Geliştirme": [
11    "Unity",
12    "C#",
13    "Hyper Casual Game"
14  ],
15  "Açık Kaynak Yazılım Geliştirme": [
16    "Git ve Github",
17    "NGINX",
18    "Flask",
19    "API"
20  ],
21  "Büyük Veri Teknolojileri": [
22    "Hadoop",
23    "Hive",
24    "HBASE",
25    "MONGODB"
26  ]
27 }
```

Bu servis ayağa kalktığı anda api olduğu için açık kalıyor ve jenkins çalışmaya devam ediyor. Server üzerindeki bir servis olduğundan hep açık kalıyor ancak jenkinsin başarılı sonuçlanabilmesi için uygulamanın kapatılması gerekiyor. Ve consolda 5003 portuna gittiğimde isteklerime 200 değerlerini gördüm ve çıktığı incellerseniz başarılı bir şekilde ayağa kalktığını ve dockerhub'a push yaptığını görebilirsiniz. Console outputum ve jenkinsde başarılı bir şekilde çıktının ekran görüntüleri:

Started by user Habib Şako

Running as SYSTEM

Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\BulutBilisim

Selected Git installation does not exist. Using Default

The recommended git tool is: NONE

No credentials specified

> git.exe rev-parse --resolve-git-dir

C:\ProgramData\Jenkins\.jenkins\workspace\BulutBilisim\.git # timeout=10

Fetching changes from the remote Git repository


```
> git.exe config remote.origin.url https://github.com/HabibSako/CI-CD-Pipeline #
timeout=10
Fetching upstream changes from https://github.com/HabibSako/CI-CD-Pipeline
> git.exe --version # timeout=10
> git --version # 'git version 2.39.2.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/HabibSako/CI-CD-Pipeline +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision efa1a2e6fe8448d89d8d3f728c3e9971b80aa55f
(refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f efa1a2e6fe8448d89d8d3f728c3e9971b80aa55f # timeout=10
Commit message: "grafana.yaml dosyası eklendi"
> git.exe rev-list --no-walk efa1a2e6fe8448d89d8d3f728c3e9971b80aa55f # timeout=10
[BulutBilisim] $ cmd /c call C:\Windows\TEMP\jenkins1465934001118054279.bat
```

```
C:\ProgramData\Jenkins\.jenkins\workspace\BulutBilisim>docker build -t habibsako/python-
flask-api-project:0.0.1.RELEASE .
#0 building with "default" instance using docker driver
```

```
#1 [internal] load .dockerignore
#1 transferring context: 2B done
#1 DONE 0.1s
```

```
#2 [internal] load build definition from Dockerfile
#2 transferring dockerfile: 203B done
#2 DONE 0.0s
```

```
#3 [auth] library/python:pull token for registry-1.docker.io
#3 DONE 0.0s
```

```
#4 [internal] load metadata for docker.io/library/python:3.11
#4 DONE 1.3s
```

```
#5 [1/5] FROM
docker.io/library/python:3.11@sha256:a2f76e7c73c241d15e867987f143dfbdcf280fb229ae1ecde98
850504fd3b234
#5 DONE 0.0s
```

```
#6 [internal] load build context
#6 transferring context: 6.45kB 0.0s done
#6 DONE 0.0s
```

```
#7 [2/5] WORKDIR /app
#7 CACHED
```

```
#8 [3/5] COPY . /app
#8 CACHED
```

```
#9 [4/5] RUN pip install --upgrade pip
#9 CACHED
```

```
#10 [5/5] RUN pip install -r requirements.txt
```

#10 CACHED

#11 exporting to image

#11 exporting layers done

#11 writing image

sha256:659a6565bd26f9ee198cccd248743828d2f526a3da4a47e4c2aba1b9870a6346 0.0s done

#11 naming to docker.io/habibsako/python-flask-api-project:0.0.1.RELEASE done

#11 DONE 0.0s

What's Next?

View a summary of image vulnerabilities and recommendations â†’ docker scout quickview

C:\ProgramData\Jenkins\.jenkins\workspace\BulutBilisim>exit 0

[BulutBilisim] \$ cmd /c call C:\Windows\TEMP\jenkins11266900744414591448.bat

C:\ProgramData\Jenkins\.jenkins\workspace\BulutBilisim>docker container run -d -p

5000:5000 habibsako/python-flask-api-project:0.0.1.RELEASE

e82b70b4772cfcbfcc4175ad01d88d7abd129d2d0c95af07651eacf1e918dcb2

C:\ProgramData\Jenkins\.jenkins\workspace\BulutBilisim>exit 0

[BulutBilisim] \$ cmd /c call C:\Windows\TEMP\jenkins17719170519200195466.bat

C:\ProgramData\Jenkins\.jenkins\workspace\BulutBilisim>docker-compose build

#0 building with "default" instance using docker driver

#1 [web internal] load .dockerignore

#1 transferring context: 2B done

#1 DONE 0.0s

#2 [web internal] load build definition from Dockerfile

#2 transferring dockerfile: 203B 0.0s done

#2 DONE 0.1s

#3 [web internal] load metadata for docker.io/library/python:3.11

#3 DONE 0.3s

#4 [web 1/5] FROM

docker.io/library/python:3.11@sha256:a2f76e7c73c241d15e867987f143dfbdcf280fb229ae1ecde98

850504fd3b234

#4 DONE 0.0s

#5 [web internal] load build context

#5 transferring context: 5.42kB 0.0s done

#5 DONE 0.1s

#6 [web 4/5] RUN pip install --upgrade pip

#6 CACHED

#7 [web 2/5] WORKDIR /app

#7 CACHED

#8 [web 3/5] COPY . /app

#8 CACHED

```
#9 [web 5/5] RUN pip install -r requirements.txt
```

```
#9 CACHED
```

```
#10 [web] exporting to image
```

```
#10 exporting layers done
```

```
#10 writing image
```

```
sha256:bd3988f388d7e389f070046a88c8779ccf8ab710125d86d36856571b2c160390 done
```

```
#10 naming to docker.io/library/bulutbilisim-web done
```

```
#10 DONE 0.0s
```

```
C:\ProgramData\Jenkins\.jenkins\workspace\BulutBilisim>exit 0
```

```
[BulutBilisim] $ cmd /c call C:\Windows\TEMP\jenkins149826158735164823.bat
```

```
C:\ProgramData\Jenkins\.jenkins\workspace\BulutBilisim>docker-compose up
```

```
Container bulutbilisim-web-1 Created
```

```
Attaching to web-1
```

```
web-1 | * Serving Flask app 'ornek'
```

```
web-1 | * Debug mode: off
```

```
web-1 | [31m[1mWARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.[0m
```

```
web-1 | * Running on all addresses (0.0.0.0)
```

```
web-1 | * Running on http://127.0.0.1:5003
```

```
web-1 | * Running on http://172.18.0.2:5003
```

```
web-1 | [33mPress CTRL+C to quit[0m
```

```
web-1 | 172.18.0.1 - - [30/Jan/2024 09:24:18] "GET / HTTP/1.1" 200 -
```

```
web-1 exited with code 0
```

```
C:\ProgramData\Jenkins\.jenkins\workspace\BulutBilisim>exit 0
```

```
[BulutBilisim] $ cmd /c call C:\Windows\TEMP\jenkins8176499781404583364.bat
```

```
C:\ProgramData\Jenkins\.jenkins\workspace\BulutBilisim>docker login -u habibsako -p Mydocker1
```

```
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
```

```
Login Succeeded
```

```
C:\ProgramData\Jenkins\.jenkins\workspace\BulutBilisim>exit 0
```

```
[BulutBilisim] $ cmd /c call C:\Windows\TEMP\jenkins9446619790550437419.bat
```

```
C:\ProgramData\Jenkins\.jenkins\workspace\BulutBilisim>docker push habibsako/python-flask-api-project:0.0.1.RELEASE
```

```
The push refers to repository [docker.io/habibsako/python-flask-api-project]
```

```
a25f149975d6: Preparing
```

```
a73899c9b0f0: Preparing
```

```
b499d6c01362: Preparing
```

```
1bf4c1aa4fa9: Preparing
```

```
423ef4d34be3: Preparing
```

```
2becd43a75e6: Preparing
```

```
b257916482ad: Preparing
```

```
f757a77eef71: Preparing
```

```
a876dfc51caa: Preparing
```

```
5bb1de08f5af: Preparing
```


```
0dfa23fffa41: Preparing
```

```
aa904f36746c: Preparing
```

```
b257916482ad: Waiting
f757a77eef71: Waiting
a876dfc51caa: Waiting
5bb1de08f5af: Waiting
0dfa23fffa41: Waiting
aa904f36746c: Waiting
2becd43a75e6: Waiting
423ef4d34be3: Mounted from library/python
1bf4c1aa4fa9: Pushed
b499d6c01362: Pushed
b257916482ad: Mounted from library/python
2becd43a75e6: Mounted from library/python
f757a77eef71: Mounted from library/python
a25f149975d6: Pushed
0dfa23fffa41: Layer already exists
aa904f36746c: Layer already exists
a876dfc51caa: Mounted from library/python
5bb1de08f5af: Mounted from library/python
a73899c9b0f0: Pushed
0.0.1.RELEASE: digest:
sha256:06cdf634d0e82ec6015bc6a8e1864f5388f63530282eddba4f1df7e3187225f2 size: 2844
```

```
C:\ProgramData\Jenkins\.jenkins\workspace\BulutBilisim>exit 0
Finished: SUCCESS
```

Sunucuyu durduktan sonra diğer adımları da sırayla gerçekleştirerek benim dockerhub hesabıma otomatik olarak push işlemini yaptı.

 **Jenkins**

Kontrol Merkezi > BulutBilisim >

Durum

</> Değişiklikler

Çalışma Alanı

Şimdi Yapılandır

Konfigürasyonu Düzenle

Projeyi Sil

GitHub Hook Log

GitHub

Adını Değiştir

✓ **BulutBilisim**

Permalinkler

- Son yapılandırma (#31), 2 dakika 21 saniye önce
- Son kararlı yapılandırma (#31), 2 dakika 21 saniye önce
- Son başarılı yapılandırma (#31), 2 dakika 21 saniye önce
- Son başarısız yapılandırma (#29), 7 dakika 13 saniye önce
- Last unsuccessful build (#29), 7 dakika 13 saniye önce
- Last completed build (#31), 2 dakika 21 saniye önce

Yapılandırma Geçmişi **Eğilim** ▾

Q Yapılandırmaları filtre... /

✓ #31 30 Oca 2024 12:23

✓ #30 30 Oca 2024 12:19


```
C:\Windows\System32>choco install kubernetes-helm
Chocolatey v2.2.2
Installing the following packages:
kubernetes-helm
By installing, you accept licenses for the packages.

kubernetes-helm v3.13.3 [Approved]
kubernetes-helm package files install completed. Performing other installation steps.
The package kubernetes-helm wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): y

Downloading kubernetes-helm 64 bit
  from 'https://get.helm.sh/helm-v3.13.3-windows-amd64.zip'
Progress: 100% - Completed download of C:\Users\habib\AppData\Local\Temp\chocolatey\ku
windows-amd64.zip (15.59 MB).
Download of helm-v3.13.3-windows-amd64.zip (15.59 MB) completed.
Hashes match.
Extracting C:\Users\habib\AppData\Local\Temp\chocolatey\kubernetes-helm\3.13.3\helm-v3
ramData\chocolatey\lib\kubernetes-helm\tools...
C:\ProgramData\chocolatey\lib\kubernetes-helm\tools
ShimGen has successfully created a shim for helm.exe
The install of kubernetes-helm was successful.
```

Daha sonra internetten bulduğum bir kaynaktaki yaml dosyasını kendime göre revize etmeye çalıştım:

Cmd yönetici olarak çalıştırdım. Daha sonra sırasıyla bu adımları gerçekleştirdim:

Kubectl create namespace monitoring

```
C:\Windows\System32>kubectl create namespace monitoring
namespace/monitoring created
```

Daha sonra helm kullanarak şu komutu yazdım: helm search repo prometheus-community

Bu komut, Helm paket yöneticisi aracılığıyla, resmi olarak tanınan bir depo olan "prometheus-community" deposundaki mevcut Prometheus paketlerini aramama yardımcı oldu.

```
C:\Users\habib>kubectl get ns
NAME                STATUS   AGE
argocd              Active   4d6h
default             Active   5d6h
kube-node-lease     Active   5d6h
kube-public         Active   5d6h
kube-system         Active   5d6h
monitoring          Active   2d15h
```

```
C:\Users\habib>helm search repo prometheus-community
NAME                                CHART VERSION   APP VERSION     DESCRIPTION
prometheus-community/alertmanager  1.7.0           v0.26.0         The Alertmanager handles alerts sent by client ...
prometheus-community/alertmanager-snmp-notifier 0.3.0           v1.5.0          The SNMP Notifier handles alerts coming from Pr...
prometheus-community/jiraalert     1.7.0           v1.3.0          A Helm chart for Kubernetes to install jiraalert
prometheus-community/kube-prometheus-stack 56.2.0          v0.71.2         kube-prometheus-stack collects Kubernetes manif...
prometheus-community/kube-state-metrics 5.16.0          2.10.1          Install kube-state-metrics to generate and expo...
prometheus-community/prom-label-proxy 0.7.0           v0.8.0          A proxy that enforces a given label in a given ...
prometheus-community/prometheus    25.10.0         v2.49.1         Prometheus is a monitoring system and time seri...
prometheus-community/prometheus-adapter 4.9.0           v0.11.2         A Helm chart for k8s prometheus adapter
prometheus-community/prometheus-blackbox-exporter 8.10.0          v0.24.0         Prometheus Blackbox Exporter

prometheus-community/prometheus-cloudwatch-expo... 0.25.3          0.15.5          A Helm chart for prometheus cloudwatch-exporter
prometheus-community/prometheus-contrack-stats... 0.5.9           v0.4.17         A Helm chart for contrack-stats-exporter
prometheus-community/prometheus-consul-exporter 1.0.0           0.4.0           A Helm chart for the Prometheus Consul Exporter
prometheus-community/prometheus-couchdb-exporter 1.0.0           1.0             A Helm chart to export the metrics from couchdb...
prometheus-community/prometheus-druid-exporter 1.1.0           v0.11.0         Druid exporter to monitor druid metrics with Pr...
prometheus-community/prometheus-elasticsearch-e... 5.4.0           v1.7.0          Elasticsearch stats exporter for Prometheus
prometheus-community/prometheus-fastly-exporter 0.3.0           v7.6.1          A Helm chart for the Prometheus Fastly Exporter
prometheus-community/prometheus-ipmi-exporter 0.2.0           v1.7.0          This is an IPMI exporter for Prometheus.
prometheus-community/prometheus-json-exporter 0.11.0          v0.6.0          Install prometheus-json-exporter
prometheus-community/prometheus-kafka-exporter 2.8.0           v1.7.0          A Helm chart to export the metrics from Kafka i...
prometheus-community/prometheus-memcached-exporter 0.3.1           v0.14.2         Prometheus exporter for Memcached metrics
prometheus-community/prometheus-modbus-exporter 0.1.0           0.4.0           A Helm chart for prometheus-modbus-exporter
prometheus-community/prometheus-mongodb-exporter 3.5.0           0.40.0          A Prometheus exporter for MongoDB metrics
prometheus-community/prometheus-mysql-exporter 2.4.0           v0.15.1         A Helm chart for prometheus mysql exporter with...
prometheus-community/prometheus-nats-exporter 2.15.0          0.14.0          A Helm chart for prometheus-nats-exporter
prometheus-community/prometheus-nginx-exporter 0.2.0           0.11.0          A Helm chart for the Prometheus NGINX Exporter
prometheus-community/prometheus-node-exporter 4.26.0          1.7.0           A Helm chart for prometheus node-exporter
prometheus-community/prometheus-opencost-exporter 0.1.1           1.108.0         Prometheus OpenCost Exporter

prometheus-community/prometheus-operator 9.3.2           0.38.1          DEPRECATED - This chart will be renamed. See ht...
prometheus-community/prometheus-operator-admiss... 0.10.1          0.71.2          Prometheus Operator Admission Webhook
prometheus-community/prometheus-operator-crd 9.0.1           v0.71.2         A Helm chart that collects custom resource defi...
prometheus-community/prometheus-pgbouncer-exporter 0.1.1           1.18.0          A Helm chart for prometheus pgbouncer-exporter
prometheus-community/prometheus-pingdom-exporter 2.5.0           20190610-1      A Helm chart for Prometheus Pingdom Exporter
prometheus-community/prometheus-pingmesh-exporter 0.3.0           v1.1.0          Prometheus Pingmesh Exporter
prometheus-community/prometheus-postgres-exporter 5.2.0           v0.15.0         A Helm chart for prometheus postgres-exporter
```

Daha sonra “helm repo add prometheus-community <https://prometheus-community.github.io/helm-charts>” komutunu girdim. helm repo add prometheus-community <https://prometheus-community.github.io/helm-charts> komutu, prometheus-community adında bir Helm depo ekler ve bu depo, <https://prometheus-community.github.io/helm-charts> URL'sine sahiptir. Daha önceden yaptığım için already exists verdi. Ardından update komutunu çalıştırdım.

```
C:\Users\habib>helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
"prometheus-community" already exists with the same configuration, skipping

C:\Users\habib>helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "prometheus-community" chart repository
Update Complete. ☺Happy Helming!☺

C:\Users\habib>
```

Sonrasında “helm install prometheus prometheus-community/kube-prometheus-stack -n monitoring” komutunu çalıştırdım. Bu komut, Prometheus ve ilişkili bileşenlerin (Grafana, Alertmanager vb.) bir Kubernetes Cluster'da kurulumunu gerçekleştirir. Bu sayede, Prometheus ve Grafana gibi araçlar kullanılarak Kubernetes ortamındaki sistem ve uygulama metrikleri toplanabilir ve görselleştirilebilir hale geldi.

```
C:\Users\habib>helm install prometheus prometheus-community/kube-prometheus-stack -n monitoring
NAME: prometheus
LAST DEPLOYED: Wed Jan 31 00:48:14 2024
NAMESPACE: monitoring
STATUS: deployed
REVISION: 1
NOTES:
kube-prometheus-stack has been installed. Check its status by running:
  kubectl --namespace monitoring get pods -l "release=prometheus"

Visit https://github.com/prometheus-operator/kube-prometheus for instructions on how to create & configure Alertmanager
and Prometheus instances using the Operator.

C:\Users\habib>
```

“kubectl get all -n monitoring” komutunu çalıştırarak podların durumunu ve grafananın portuna baktım. Tüm podların çalışır duruma gelmesini bekledim.

```
C:\Users\habib>kubectl get all -n monitoring

NAME                                READY   STATUS             RESTARTS   AGE
pod/alertmanager-prometheus-kube-prometheus-alertmanager-0  0/2     Init:0/1           0           46s
pod/prometheus-grafana-77c588fccf-22zv5                      0/3     ContainerCreating  0           66s
pod/prometheus-kube-prometheus-operator-5987fcd6bf-cnb5h     1/1     Running            0           66s
pod/prometheus-kube-state-metrics-6db866c85b-jhc8z           1/1     Running            0           66s
pod/prometheus-prometheus-kube-prometheus-prometheus-0       0/2     Init:0/1           0           46s
pod/prometheus-prometheus-node-exporter-grj5p                1/1     Running            0           66s

NAME                                TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)
service/alertmanager-operated        ClusterIP      None             <none>           9093/TCP,9094/TCP,9094/UDP  46s
service/prometheus-grafana           ClusterIP      10.107.226.38   <none>           80/TCP
service/prometheus-kube-prometheus-alertmanager-67s          ClusterIP      10.110.246.204   <none>           9093/TCP,8080/TCP
service/prometheus-kube-prometheus-operator-67s              ClusterIP      10.97.6.187      <none>           443/TCP
service/prometheus-kube-prometheus-prometheus-67s            ClusterIP      10.109.186.46    <none>           9090/TCP,8080/TCP
service/prometheus-kube-state-metrics-67s                    ClusterIP      10.102.136.130   <none>           8080/TCP
service/prometheus-operated-46s          ClusterIP      None             <none>           9090/TCP
service/prometheus-prometheus-node-exporter-67s              ClusterIP      10.98.174.55     <none>           9100/TCP

NAME                                DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/prometheus-prometheus-node-exporter  1          1         1         1             1           kubernetes.io/os=linux  66s

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/prometheus-grafana  0/1     1             0           66s
deployment.apps/prometheus-kube-prometheus-operator  1/1     1             1           66s
deployment.apps/prometheus-kube-state-metrics        1/1     1             1           66s

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/prometheus-grafana-77c588fccf        1          1         0           66s
replicaset.apps/prometheus-kube-prometheus-operator-5987fcd6bf  1          1         1           66s
replicaset.apps/prometheus-kube-state-metrics-6db866c85b  1          1         1           66s

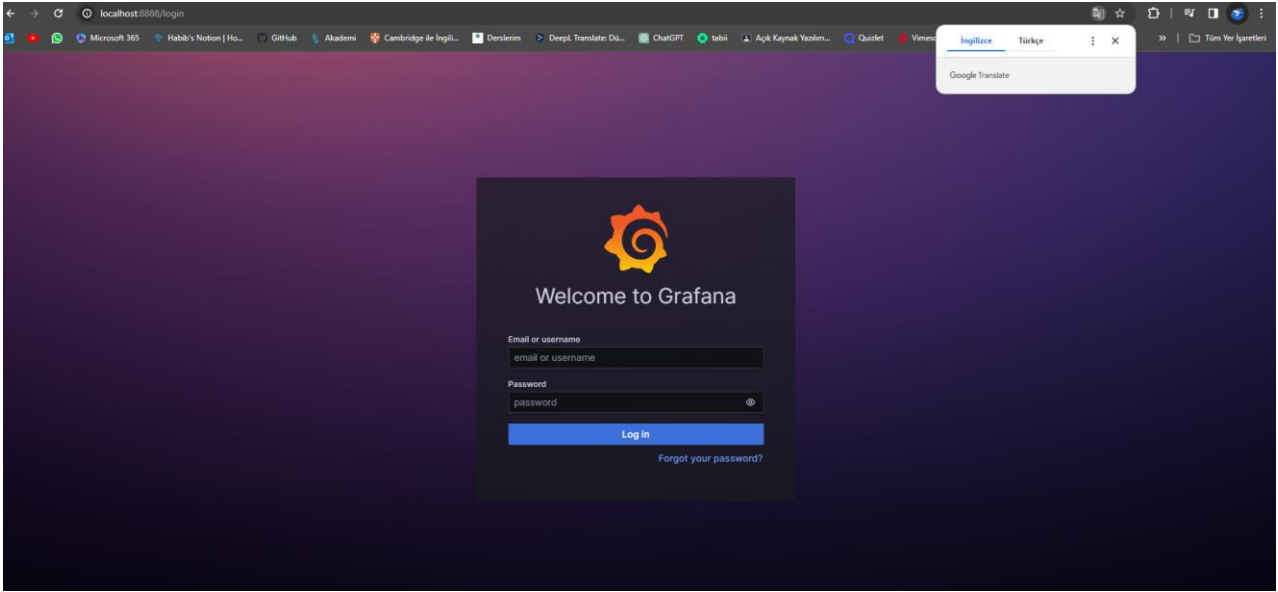
NAME                                READY   AGE
statefulset.apps/alertmanager-prometheus-kube-prometheus-alertmanager  0/1     46s
statefulset.apps/prometheus-prometheus-kube-prometheus-prometheus        0/1     46s

C:\Users\habib>
```

“kubectl port-forward service/prometheus-grafana 8888:80 -n monitoring” komutunu çalıştırarak port forwarding işlemi yaptım. 8888 portunu grafananın 80 portuna bağladım.

```
C:\Users\habib>kubectl port-forward service/prometheus-grafana 8888:80 -n monitoring
Forwarding from 127.0.0.1:8888 -> 3000
Forwarding from [::1]:8888 -> 3000
```

Localhost:8888 adresine giderek grafananın giriş ekranına geldim.



Daha sonra yeni terminal ekranı açtım ve önce kubectl “get secret -n monitoring” ardından “kubectl get secret prometheus-grafana -oyaml -n monitoring” komutlarını çalıştırarak default yaml dosyasına eriştim. Buradaki admin-password alanındaki kodu base64 göre decode değerini internetten bularak default şifreyi aldım. Grafana giriş ekranına gelerek admin ve şifre(prom-operator) girdim.

```
PS C:\Windows\system32> kubectl get secret prometheus-grafana -oyaml -n monitoring
apiVersion: v1
data:
  admin-password: cHJvbS1vcGVyYXRvcg==
  admin-user: YWRtaW4=
  ldap-toml: ""
kind: Secret
metadata:
  annotations:
    meta.helm.sh/release-name: prometheus
    meta.helm.sh/release-namespace: monitoring
  creationTimestamp: "2024-01-30T21:48:34Z"
  labels:
    app.kubernetes.io/instance: prometheus
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/name: grafana
    app.kubernetes.io/version: 10.2.3
    helm.sh/chart: grafana-7.2.5
  name: prometheus-grafana
  namespace: monitoring
  resourceVersion: "260648"
  uid: 523f55e8-d7a6-4405-b765-3704a445f1ee
type: Opaque
PS C:\Windows\system32>
```

En son olarak grafananın yan ekranından dashbord> Kubernetes / API server seçerek ekran görüntülerini aldım. Ekran görüntüleri:

