

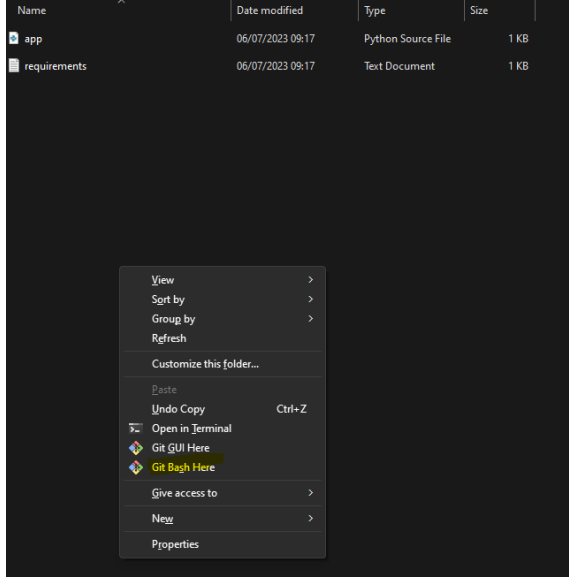
BESTCLOUDFOME CASE STUDY REPORT

Merhabalar. Case ile alakalı işlemler aşağıda sıralanmıştır.

İlk olarak masaüstünde “akademi2023” dosyası oluşturulmuştur.

DOSYALARIN GİT REPOİTORY EKLENMESİ

1. Dosya içerisinde git bash başlatılır.



2. “git status” ile .git control sağlanır.

```
PC@DESKTOP-1I5PUME MINGW64 ~/akademi2023
$ git status
fatal: not a git repository (or any of the parent directorie
s): .git
```

3. “git init” ile .git oluşturulur.

```
PC@DESKTOP-1I5PUME MINGW64 ~/akademi2023
$ git init
Initialized empty Git repository in C:/Users/PC/akademi2023/.git/
```

4. “git status” ile tekrar control sağlanır. Ekleyeceğimiz 2 dosya burada görülür.

```
PC@DESKTOP-1I5PUME MINGW64 ~/akademi2023 (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  app.py
  requirements.txt

nothing added to commit but untracked files present (use "git add"
to track)
```

5. “git add .” komutu ile dosyalar yüklenmeye hazır.

```
PC@DESKTOP-1I5PUME MINGW64 ~/akademi2023 (master)
$ git add .
warning: in the working copy of 'app.py', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'requirements.txt', LF will be replaced by CRLF the next time Git touches it
```

6. “git commit -m “Dosyalar localden yüklendi.”” komutu ile mesaj bilgisi girilir.

```
PC@DESKTOP-1I5PUME MINGW64 ~/akademi2023 (master)
$ git commit -m "Docyalar localden yüklendi."
[master (root-commit) cb83bcc] Docyalar localden yüklendi.
2 files changed, 14 insertions(+)
create mode 100644 app.py
create mode 100644 requirements.txt
```

7. “git log” ile kontrol sağlanır.

```
PC@DESKTOP-1I5PUME MINGW64 ~/akademi2023 (master)
$ git log
commit cb83bcc3f60453758789cc9c88074248e0a4a4e4 (HEAD -> master)
Author: Yasin Temur <tahayasintemur@gmail.com>
Date: Sat Jul 15 16:51:58 2023 +0300

    Docyalar localden yüklendi.
```

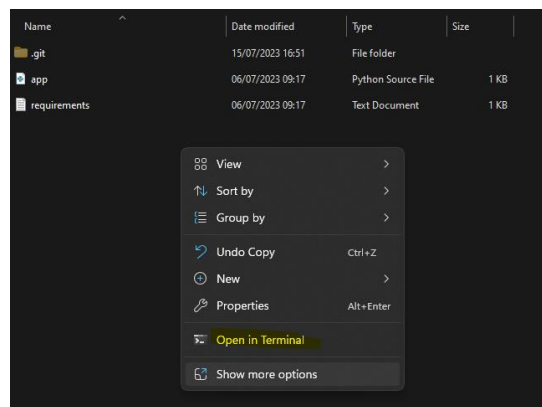
Not: Bu adımdan sonraki ekran çıktıları git repository gönderilirken oluşan çıktılarıdır. Bundan önceki ekran görüntüleri terminal “clear” ile temizlendiğinden tekrar yazılarak alınmıştır.

8. “git push <https://github.com/yasintemur/bcfmakademi2023.git>” komutu ile dosyalar oluşturduğum git repository gönderilir.

```
PC@DESKTOP-1I5PUME MINGW64 ~/Desktop/academi2023 (master)
$ git push https://github.com/yasintemur/bcfmakademi2023.git
info: please complete authentication in your browser...
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 452 bytes | 452.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/yasintemur/bcfmakademi2023.git
 * [new branch]      master -> master
```

DOCKERFILE OLUŞTURMA

1. Masaüstünde oluşturduğumuz “akademi2023” klasörü içerisinde powershell terminal ekranına geçeriz.



2. Terminal ekranında “ls” komutu ile var olan dosyalar görülür. Ardından “code .” ile VS Code uygulamasına geçiş yapılır.

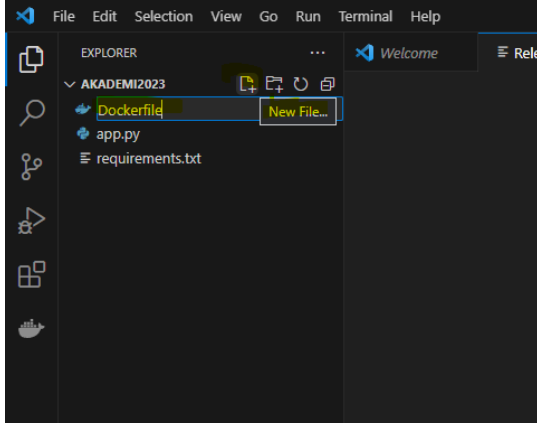
```
Windows PowerShell
PS C:\Users\PC\akademi2023> ls

Directory: C:\Users\PC\akademi2023

Mode                LastWriteTime         Length Name
----                -
-a----          06/07/2023   09:17           203 app.py
-a----          06/07/2023   09:17            13 requirements.txt

PS C:\Users\PC\akademi2023> code .
```

3. VS Code editörü üzerinden “New File” tıklanır ve yeni dosya adına D harfi büyük şekilde “Dockerfile” yazılır. Aksi taktirde docker dosyayı tanımayacaktır.



4. Oluşturulan Dockerfile dosyasına komutlar yazılır.

- ✓ FROM python:alpine : Alpine image kullanılarak image temeli oluşturulmuştur. Küçük ve python destekli olması kullanım açısından en büyük anataj olmuştur.
- ✓ COPY . /uyg : dizinde varolan dosyaların image içerisinde “uyg” klasörüne kopyalanması istenmektedir. Klasör yoksa otomatik oluşturulacaktır.
- ✓ WORKDIR /uyg : “uyg” klasörüne geçilerek komutların bu klasörden çalıştırılması amaçlanmaktadır.
- ✓ RUN pip install -r requirements.txt : “requirements.txt” dosyası içerisinde belirtilen Flask sürümünü yüklenmesi komutu çalıştırılmaktadır.
- ✓ EXPOSE 5000 : İmagedan oluşturulacak Containerların hangi port üzerinden erişebileceğini, yayın yapacağını belirtmektedir.
- ✓ CMD python ./app.py : İmagedan container oluşturulduğunda varsayılan olarak çalıştırılacak komut belirtilmektedir.

```
EXPLORER
AKADEMI2023
  app.py
  Dockerfile
  requirements.txt

Dockerfile U X
Dockerfile > ...
1 FROM python:alpine
2 COPY . /uyg
3 WORKDIR /uyg
4 RUN pip install -r requirements.txt
5 EXPOSE 5000
6 CMD python ./app.py
```

5. Dockerfile en son kayıt edilir.

HAZIR DOCKERFILE İLE IMAGE OLUŞTURMA VE CONTAINER OLUŞTURARAK TEST ETME

Not: İmage oluşturma işlemi daha önceden yapıldığından ekran görüntüsü alabilmek için adımlar tekrar yapılmıştır. Bundan kaynaklı image id farklıdır.

1. İlk olarak Windows üzerinde Powershell işlemleri yaptığımız için “Docker Desktop for Windows” yüklü olması gerekir. Yüklü ve hazır olduğu düşüncesiyle devam ediyoruz.
2. Powershell açılır.
3. “docker image build -t yasintemur/uygulama” komutu ile oluşturduğumuz Dockerfile dosyasındaki komutları çalıştırarak image oluşturulur.

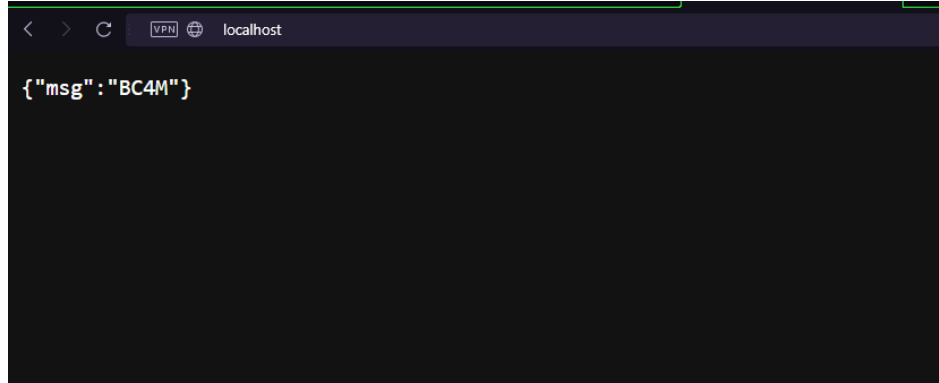
```
PS C:\Users\PC\akademi2023> docker image build -t yasintemur/uygulama
2023/07/15 17:52:25 http2: server: error reading preface from client //./pipe/docker_engine: file has already been closed
[*] Building 6.5s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 153B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/python:alpine
=> [auth] library/python:pull token for registry-1.docker.io
=> [internal] load build context
=> => transferring context: 29.75kB
=> CACHED [1/4] FROM docker.io/library/python:alpine@sha256:25df32b662118dab946b58f9e920e3301da09727b5b87438c8bcd0b139627fdc
=> [2/4] COPY ./app
=> [3/4] WORKDIR /app
=> [4/4] RUN pip install -r requirements.txt
=> exporting to image
=> => exporting layers
=> => writing image sha256:85ffc8d09a4a86e9934cb287f216501709c8c9e3dda5e491c7eb1816f66262f
=> => naming to docker.io/yasintemur/uygulama
PS C:\Users\PC\akademi2023>
```

4. Oluşturduğumuz image ile container oluşturarak test işlemine başlıyoruz.

“docker container run --name test -p 80:5000 yasintemur/uygulama”

```
PS C:\Users\PC\akademi2023> docker container run --name test -p 80:5000 yasintemur/uygulama
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
```

5. Bu şekilde istenilen ekran görüntüsünü elde etmekteyiz.



OLUŞTURDUĞUMUZ İMAGE DOCKER HUB'A YÜKLEME

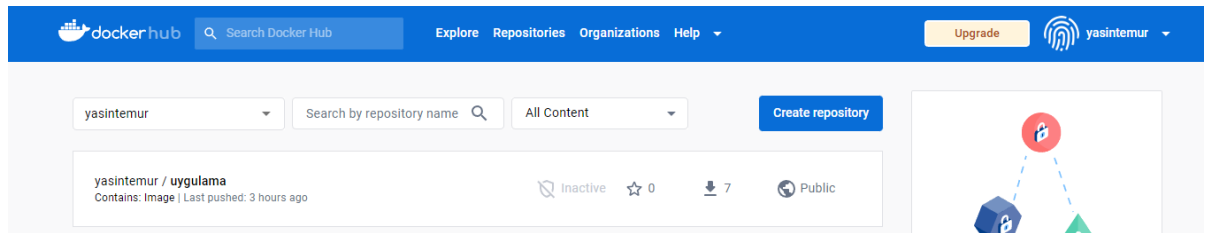
1. “docker image ls -a” ile sistemdeki image listelenir.

```
PS C:\Users\PC> docker image ls -a
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
yasintemur/uygulama latest      b04347c7efdc 12 hours ago  68.3MB
```

2. Oluşturduğumuz image dosyası “docker push yasintemur/uygulama” yazılarak docker hub gönderilir.

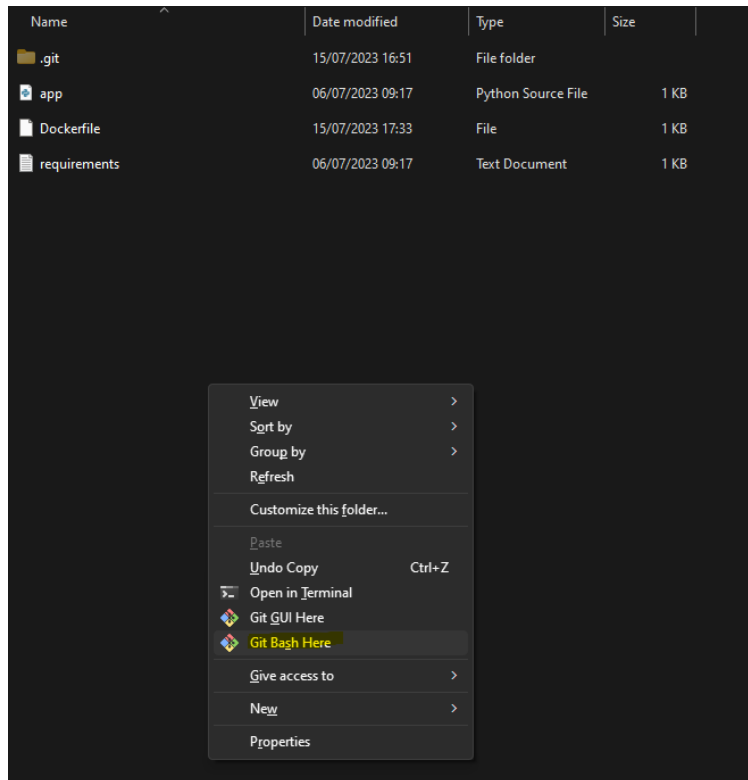
```
PS C:\Users\PC> docker tag b04347c7efdc yasintemur/uygulama:1.0
PS C:\Users\PC> docker push yasintemur/uygulama
Using default tag: latest
The push refers to repository [docker.io/yasintemur/uygulama]
2dc01a08f160: Pushed
5f70bf18a086: Pushed
f60e6cded245: Pushed
776df4582527: Mounted from library/python
42568fdeffd1: Mounted from library/python
9ec06db39a30: Mounted from library/python
cd18acbc1cce: Mounted from library/python
78a822fe2a2d: Mounted from library/python
```

3. Docker HUB üzerinden image görüntüleyebiliriz.



DOCKERFILE DOSYASINI GİT REPOSITORY YÜKLEME

1. Dosya içerisinde git bash başlatılır.



2. .git olduğundan direk “git add .” ile Dockerfile yüklemeye yazır hale getirilir.

```
PC@DESKTOP-1I5PUME MINGW64 ~/Desktop/academi2023 (master)
$ git add .
```

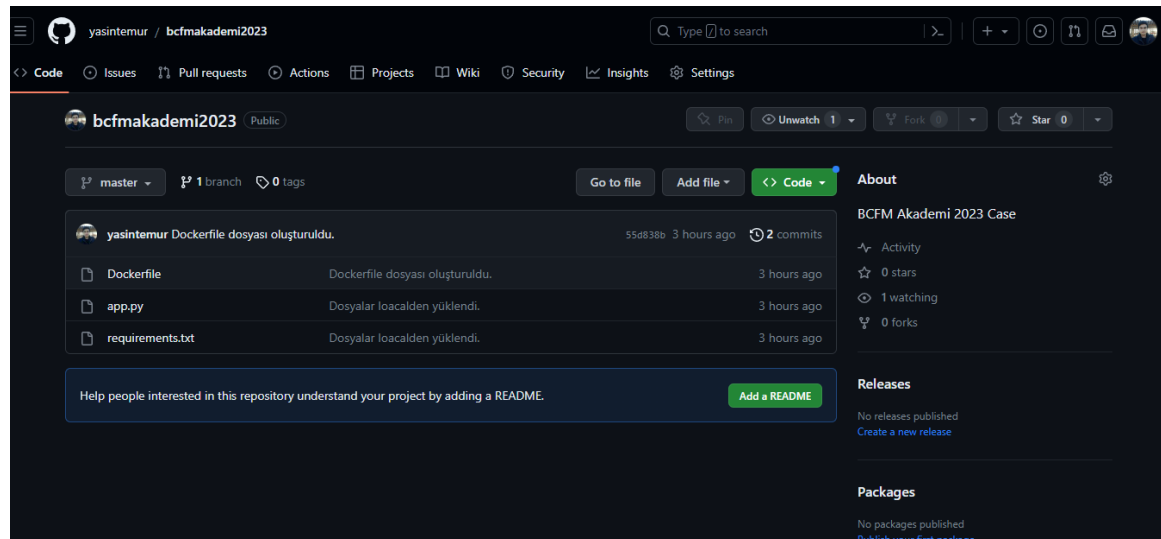
3. “git status” ile control edilir. Ardından “git commit -m “Dockerfile dosyası oluşturuldu.”” Komutu ile mesaj bilgisi girilir

```
PC@DESKTOP-1I5PUME MINGW64 ~/Desktop/academi2023 (master)
$ git commit -m "Dockerfile dosyası oluşturuldu."
[master 55d838b] Dockerfile dosyası oluşturuldu.
1 file changed, 6 insertions(+)
create mode 100644 Dockerfile
```

4. “git push <https://github.com/yasintemur/bcfmakademi2023.git>” komutu ile dosya oluşturduğum git repository gönderilir.

```
PC@DESKTOP-1I5PUME MINGW64 ~/Desktop/academi2023 (master)
$ git push https://github.com/yasintemur/bcfmakademi2023.git
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 448 bytes | 448.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/yasintemur/bcfmakademi2023.git
6654e0c..55d838b master -> master
```

5. GitHub üzerinden repository görüntüleyerek dosyaları görebiliriz.



The screenshot displays the GitHub interface for the repository `yasintemur / bcfmakademi2023`. The repository is public and has 0 stars and 0 forks. The commit history shows a single commit by `yasintemur` titled "Dockerfile dosyası oluşturuldu." with a message "Dockerfile dosyası oluşturuldu." and a timestamp of 3 hours ago. The commit details show three files: `Dockerfile`, `app.py`, and `requirements.txt`, all added 3 hours ago. The repository also has a README section with a button to "Add a README".