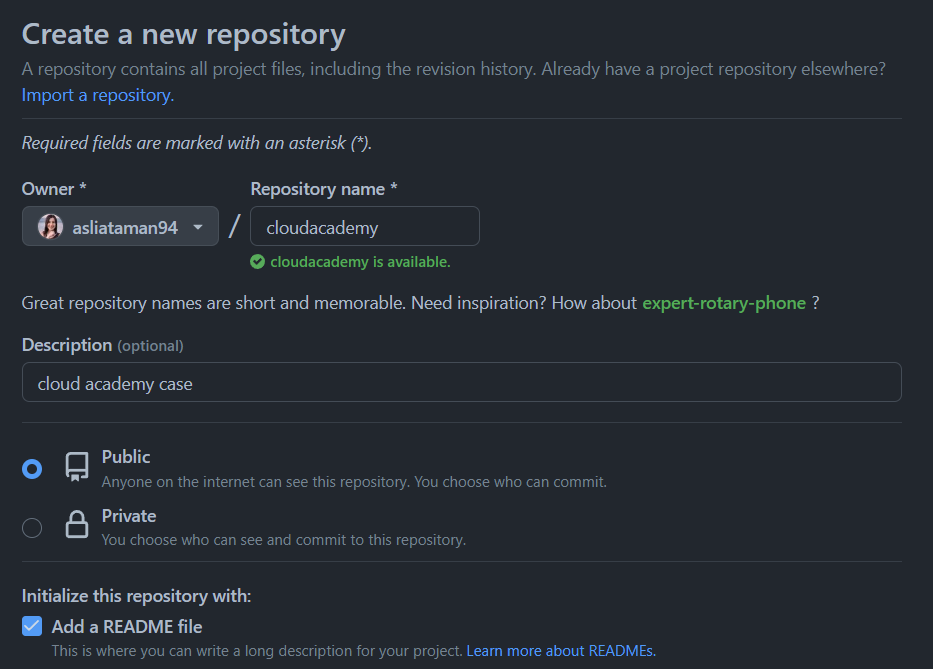
**ACADEMY ’23 CASE STUDY**

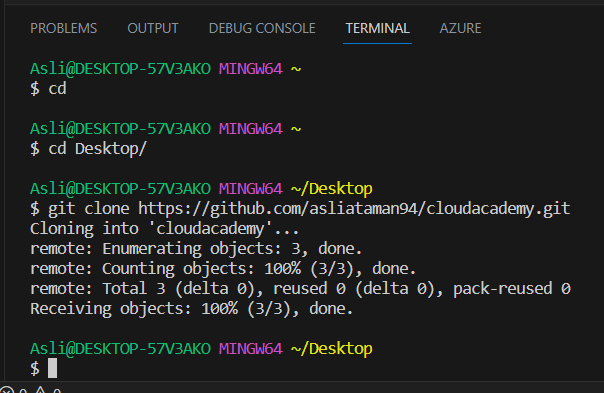
**Uygulama:** Python ile geliştirilmiş bir uygulama, web browser üzerinde "{"msg":"BC4M"}" mesajını gösteren bir API.

**GitHub Proje Reposu:** https://github.com/asliataman94/cloudacademy

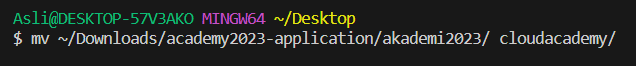
**● Kodun Git Reposuna Yüklenmesi:**



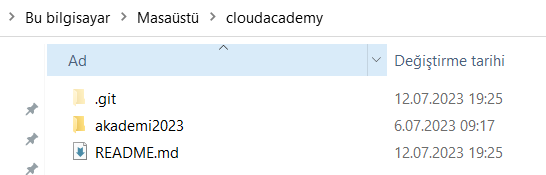
Github’da cloudacademy isimli public repo açıldı.



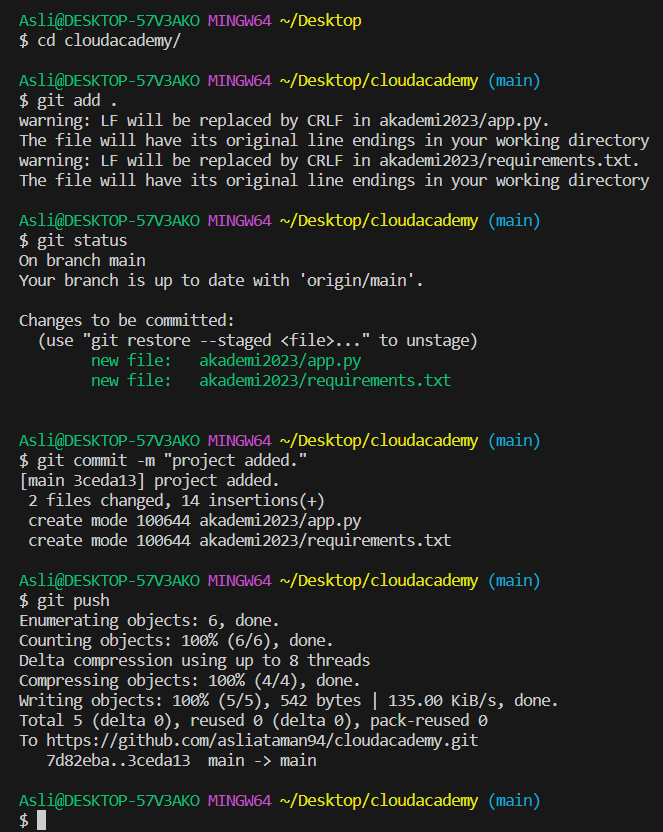
Remotedaki repo locale klonlandı.



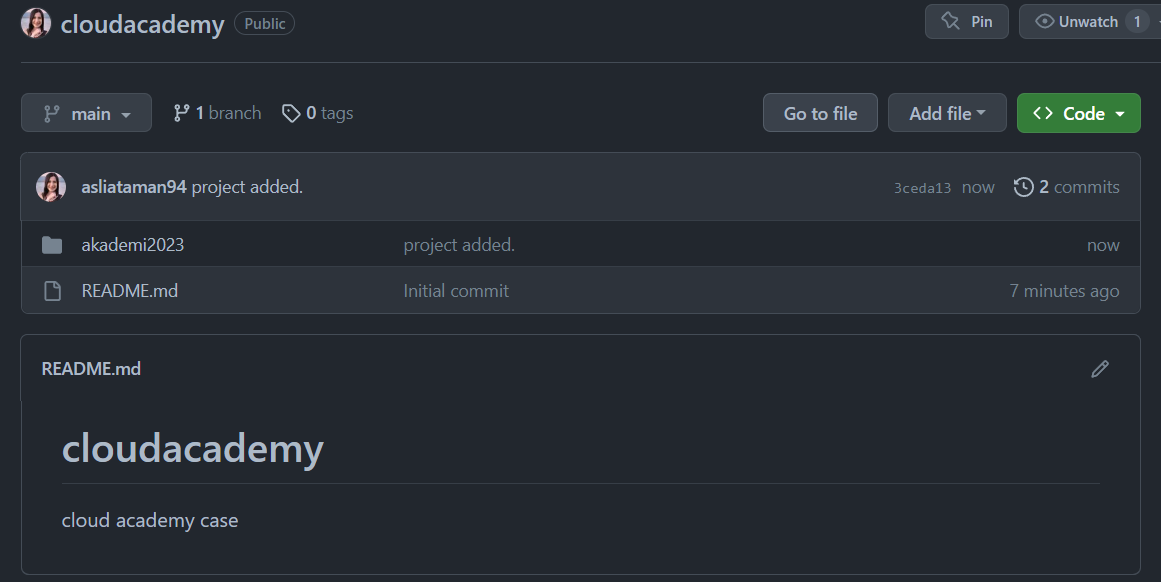
Uygulama kodu downloads klasöründen lokaldeki repoya taşındı.



Lokal repository klasörü bu şekildedir ve uygulamanın kodu akademi2023 klasörünün içindedir.

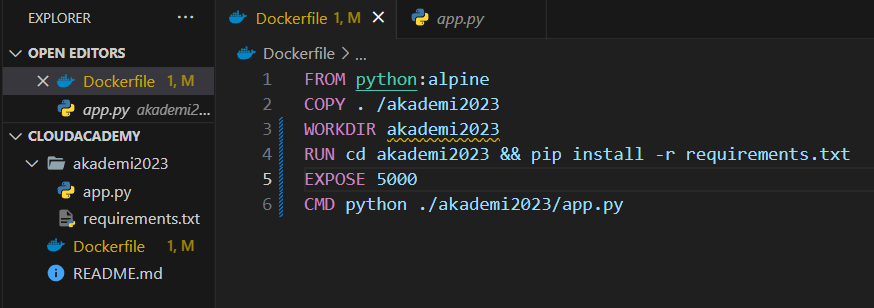


Uygulama kodu klasörü ile lokal repodaki değişiklikler remote repoya pushlandı.



Github’da reponun son görüntüsü bu şekildedir.

**● Dockerfile Oluşturulması:**



FROM python:alpine

COPY . /akademi2023

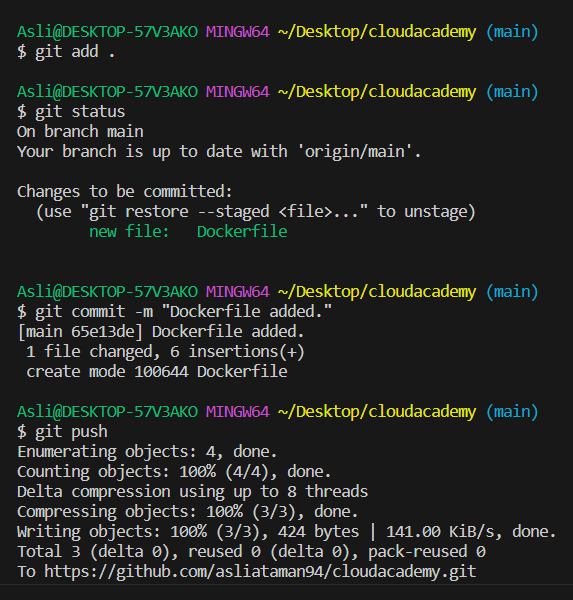
WORKDIR akademi2023

RUN cd akademi2023 && pip install -r requirements.txt

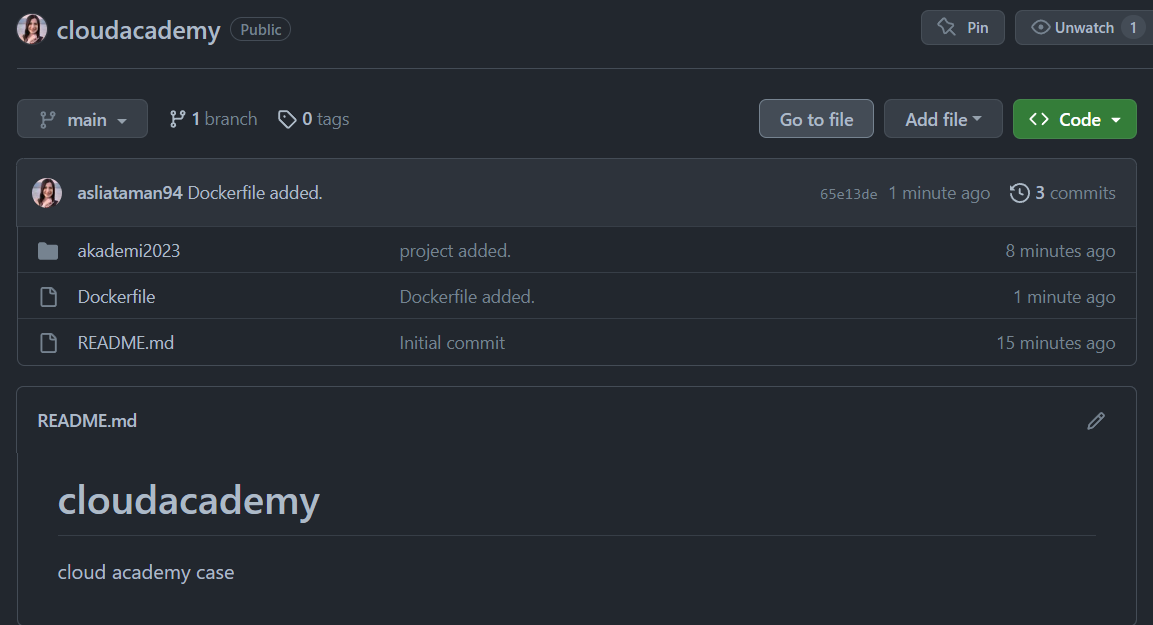
EXPOSE 5000

CMD python ./akademi2023/app.py

Uygulamayı çalıştıracak Dockerfile yazıldı. Uygulamanın küçük boyutlu olması için Python alpine image ı kullanıldı. İmage ın içine uygulama dosyaları kopyalandı. Pip install ile projenin gerekli kütüphaneleri yüklendi. Uygulamanın yayın yapacağı port app.py dosyası içerisinde 5000 olduğu görüldü. Expose ile 5000 portundan yayın yapacağını duyuruldu. Image ayağa kalktığında CMD ile Python dosyamızı çalıştıracaktır.

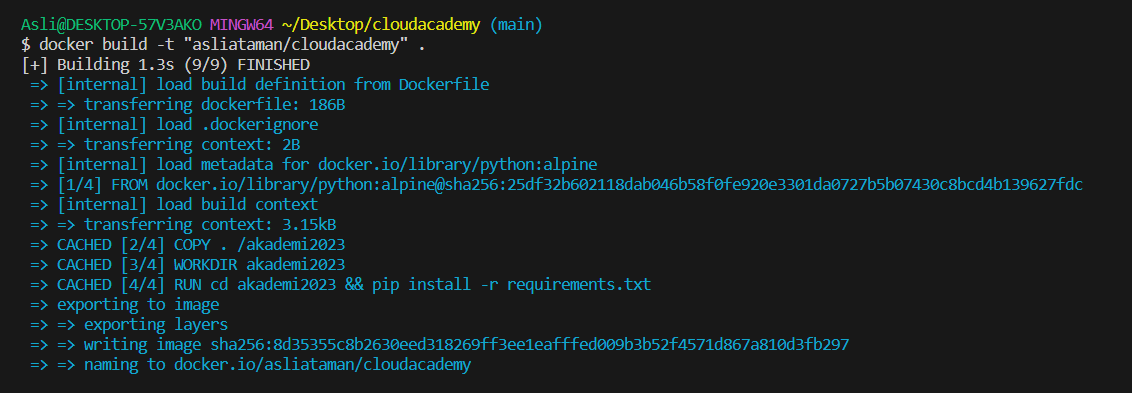


Yazılan Dockerfile dosyası Github reposuna gönderildi.



Github Reposunun son durumu bu şekildedir.

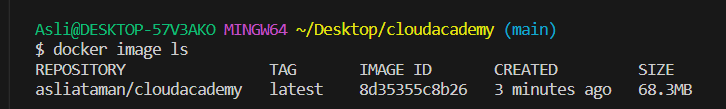
**● Docker Image Oluşturulması ve Docker Image Registry’de Saklanması:**



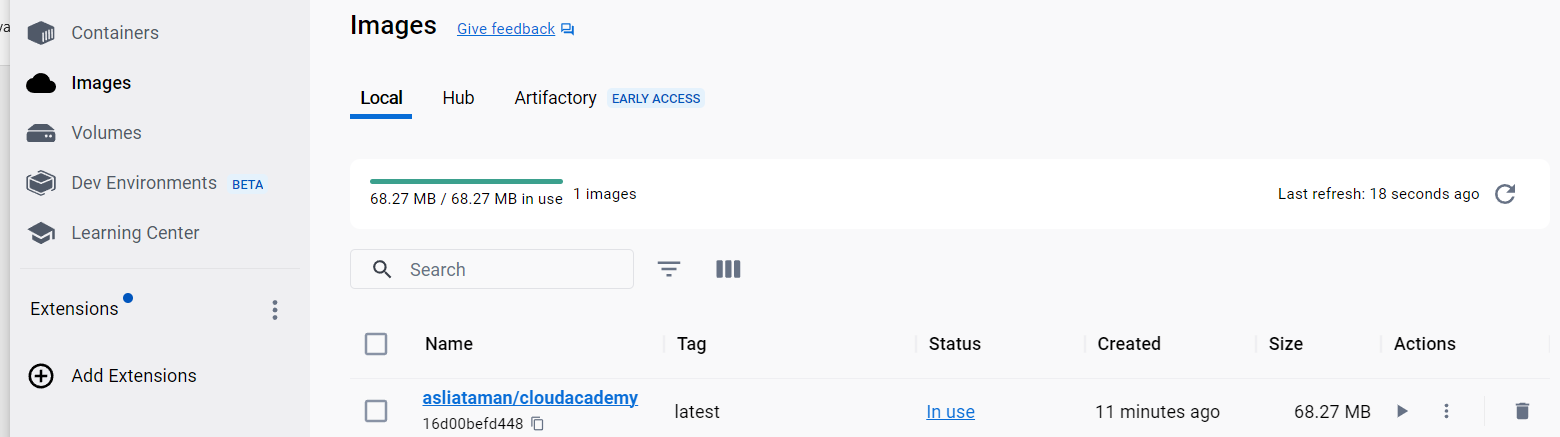
‘docker build -t “asliataman/cloudacademy” .’ komutu ile Docker image oluşturuldu.

-t ile tag vererek kendi DockerHub ismimi girdim. Komut sonundaki “.” İle bu Dockerfile içinde kopyalanacak dosya var ise bu dizinden al ve çalıştır demektir.

Bu işlem birkaç kez denendiği için cacheten alarak işlemleri gerçekleştirdi. İlk denemede app.py klasöründeki porta dikkat etmeden web uygulamasını 80 portundan yayımlama işlemi denendi, daha sonra 5000 portu ile uygulama çalıştırıldı.



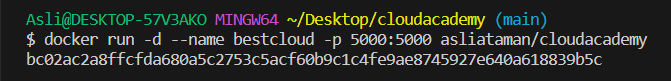
“docker image ls” komutu ile docker image in oluştuğu görüldü.



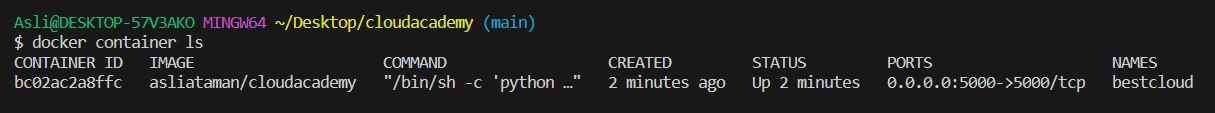
Lokalimdeki Docker uygulamasında image sayfasının görüntüsü bu şekildedir.

Konteynerın çalıştığından ve uygulamanın geldiğinden emin olduktan sonra DockerHub’a gönderilecektir.

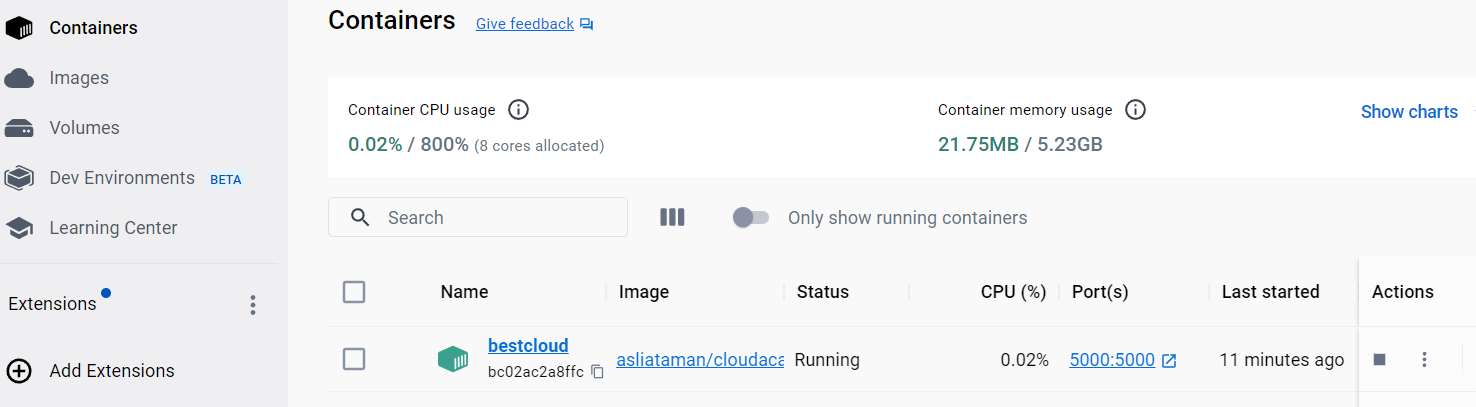
**● Docker Container Çalıştırılması:**



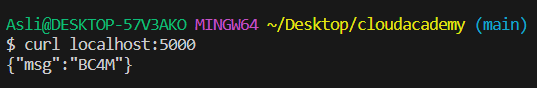
“docker run -d –name bestcloud -p 5000:5000 asliataman/cloudacademy” komutu ile oluşturduğumuz image dan bestcloud isminde bir konteyner oluşturularak çalıştırıldı.

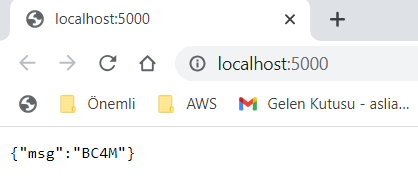


“docker container ls” komutu ile konteynerın oluştuğu ve 2 dakikadır çalıştığı görüldü.

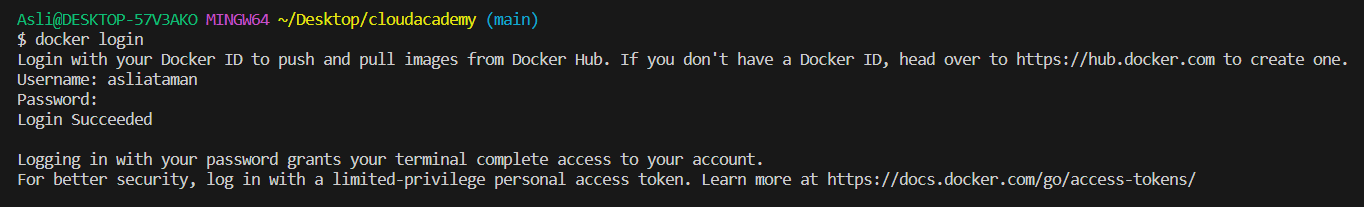


Lokalimdeki Docker uygulamasında container sayfasının görüntüsü bu şekildedir.





Uygulamanın çalıştığından emin olduk artık Docker Hub’a pushlama işlemini gerçekleştirebiliriz.



“docker login” komutu ile credential bilgileri girildi.

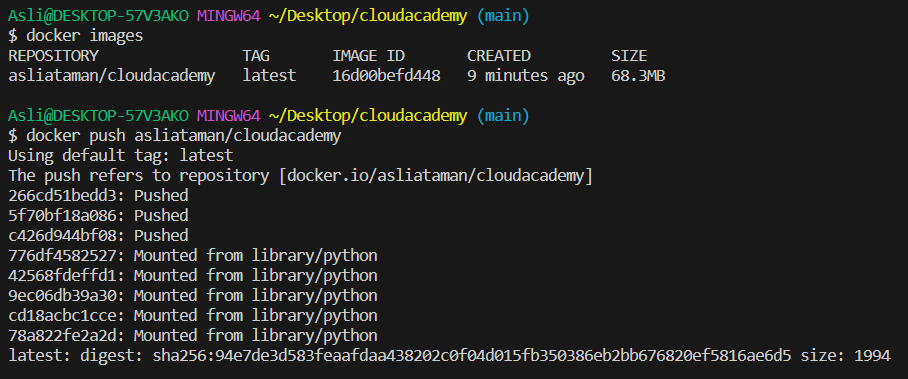
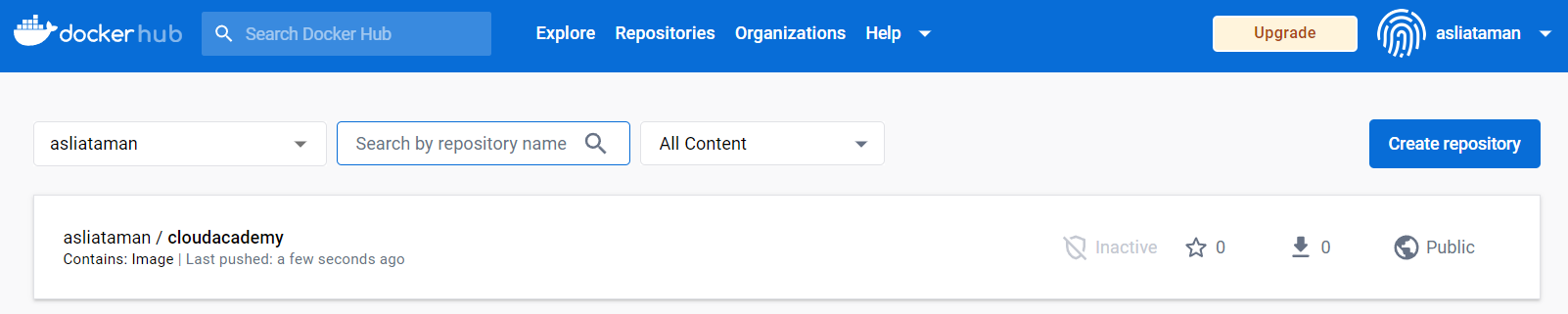


Image ismi kontrol edilip “docker push asliataman/cloudacademy” komutu ile DockerHub’a image gönderildi.



DockerHub’da image görüntüsü şu şekildedir.

Bu Word döküman dosyası da repoya eklenmiştir.