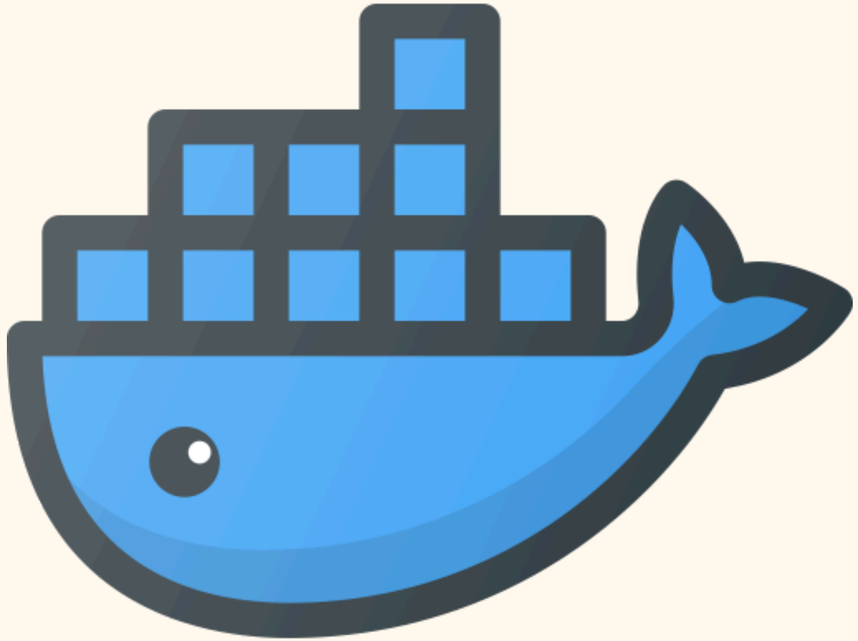
DOCKER ve KONTEYNER TEKNOLOJİLERİ

KEMAL KEREM ACAR
21360859004
Bilgisayar Mühendisliği-3.Sınıf
23 Mayıs 2024

İÇİNDEKİLER

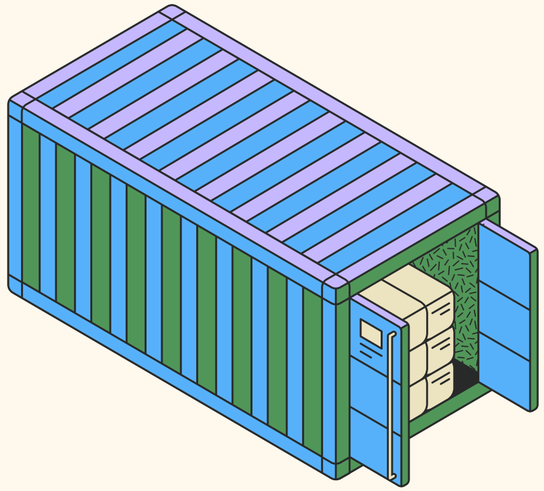
Docker Nedir

Docker, yazılım uygulamalarını hafif, taşınabilir ve izole edilmiş konteynerler içinde çalıştırmak için kullanılan bir platformdur.



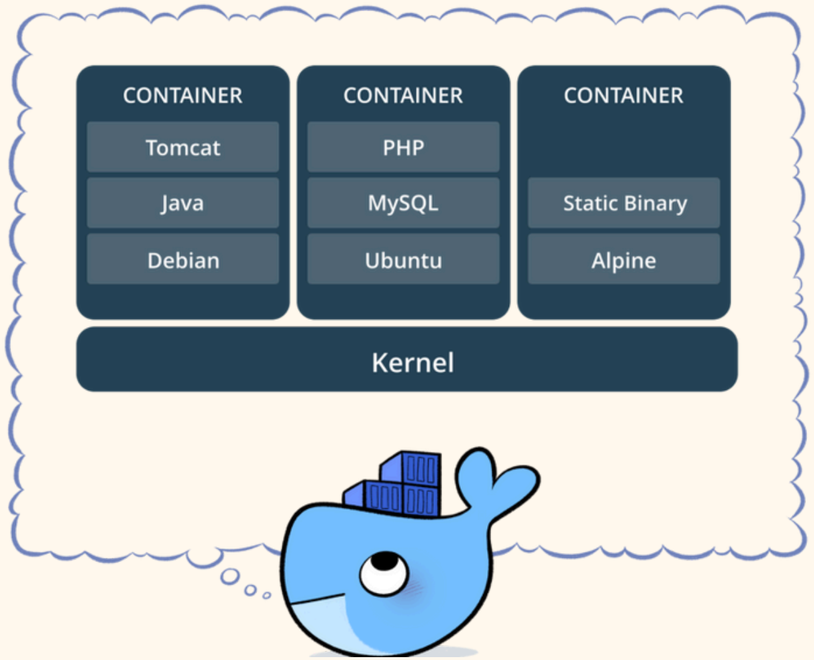
Temelde, bir uygulamanın çalışması için gerekli tüm kod, çalışma zamanı, sistem araçları ve kütüphaneleri içeren bir konteyner oluşturur. Bu konteyner, işletim sisteminden bağımsız olarak çalışabilir ve herhangi bir ortamda tutarlı bir şekilde çalışır.

Konteyner Teknolojileri



Konteyner, yazılım uygulamalarını ve bağımlılıklarını tek bir pakette bir araya getiren ve bu paketi her türlü ortamda tutarlı bir şekilde çalıştıran hafif bir sanal makinedir.

Kubernetes: Konteyner orkestrasyonunda liderdir ve büyük ölçekli uygulamaların yönetiminde yaygın olarak kullanılır.



Docker ve Kubernetes birbirleriyle uyumlu çalışabilir ve genellikle birlikte kullanılırlar. İkisi de konteyner teknolojileriyle ilgili farklı amaçlara hizmet eder ve birbirlerinin eksikliklerini tamamlarlar.

Docker’ın Avantajları

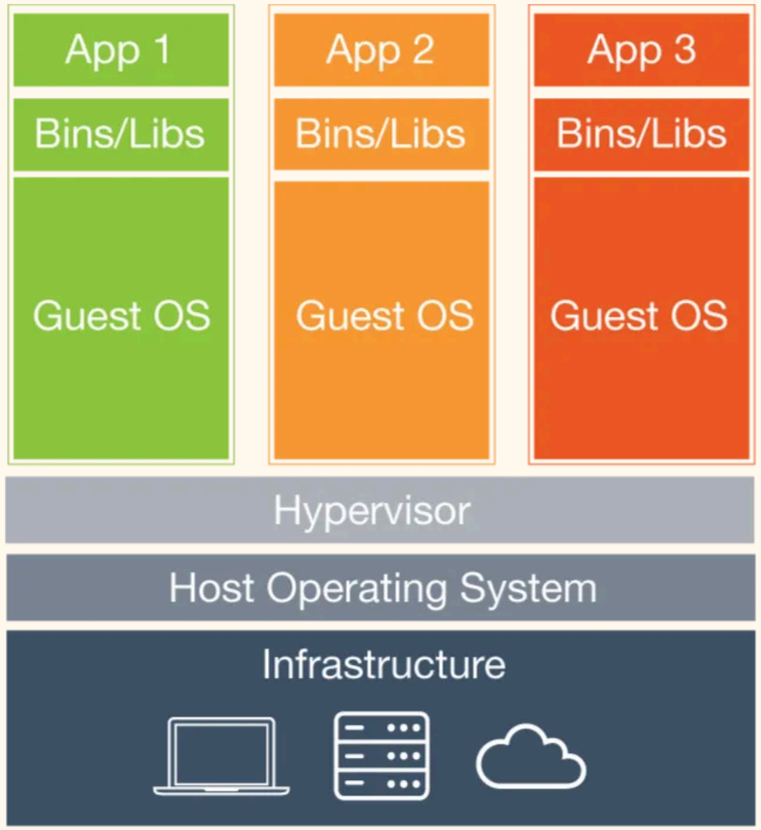
-Taşınabilirlik : Docker, uygulamaların farklı ortamlarda aynı şekilde çalışmasını sağlar.
Yerel bilgisayarlarından sunuculara ve hatta farklı bulut platformlarına kolayca taşımasını sağlar.

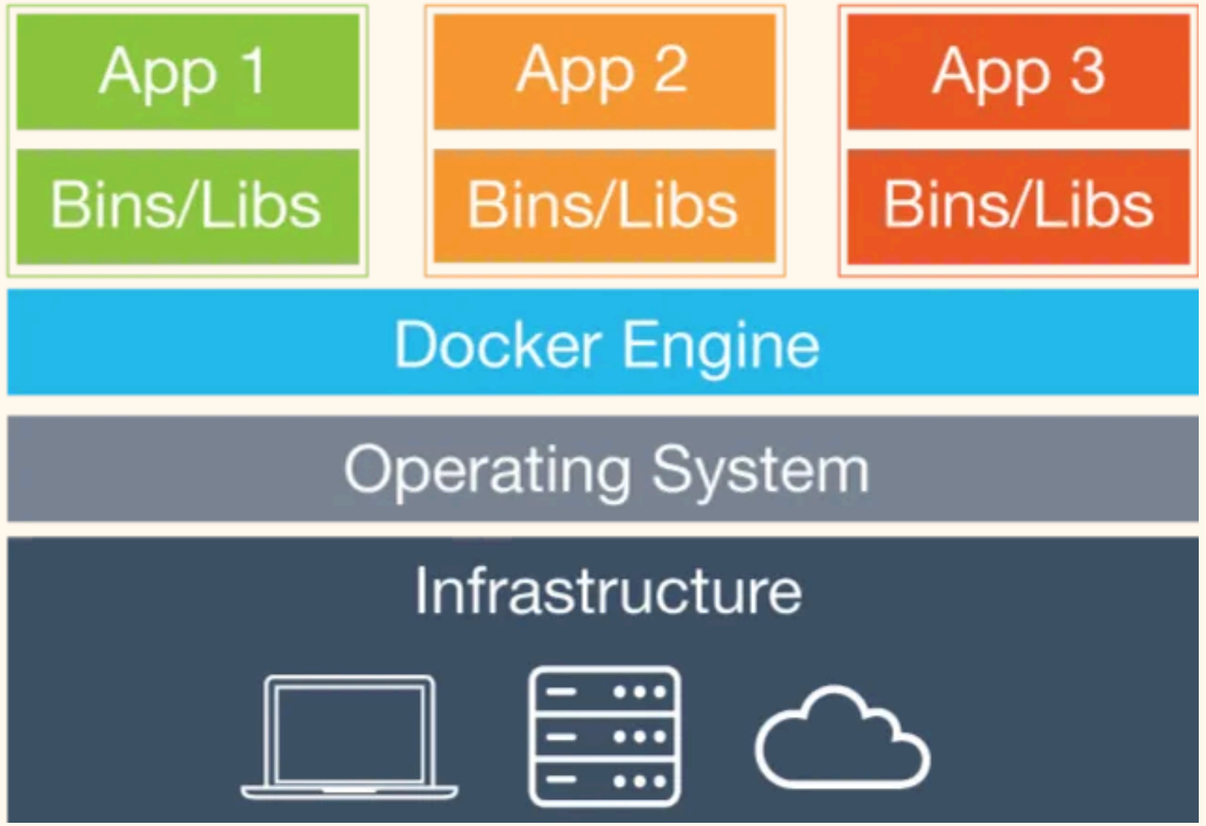
-Verimlilik: Daha az kaynak tüketimi ve hızlı dağıtım ile yüksek verimlilik sağlar.

-İzolasyon : Uygulamalar arasında birbirine müdahale olmadan güvenli bir şekilde çalıştırılmasını sağlar.

-Hızlı Dağıtım: Geliştiriciler, Docker konteynerleri aracılığıyla uygulamaları hızlıca paketleyebilir, test edebilir ve dağıtabilirler.

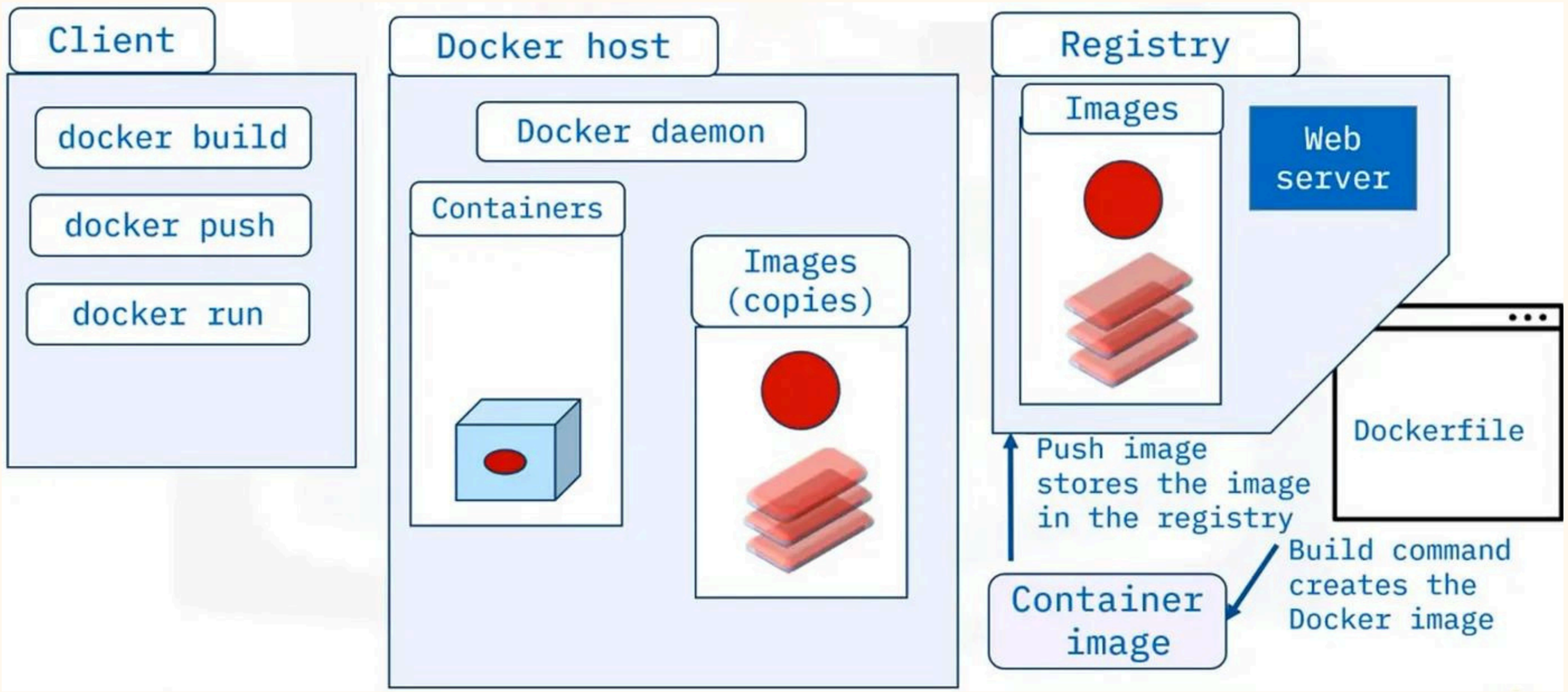
Sanal Makine ile Farkları





Soldaki şema bir sanal makinenin nasıl çalıştığını göstermektedir. Görüldüğü üzere her sanal makinenin kendi işletim sistemi kopyası bulunmaktadır. Docker ise işletim sistemi bağımlılığını ortadan kaldırmasıyla sanal makinelerden farklı bir yapı sunar.Docker üzerinde bulunduğu işletim sistemini birden fazla container ile paylaşır.

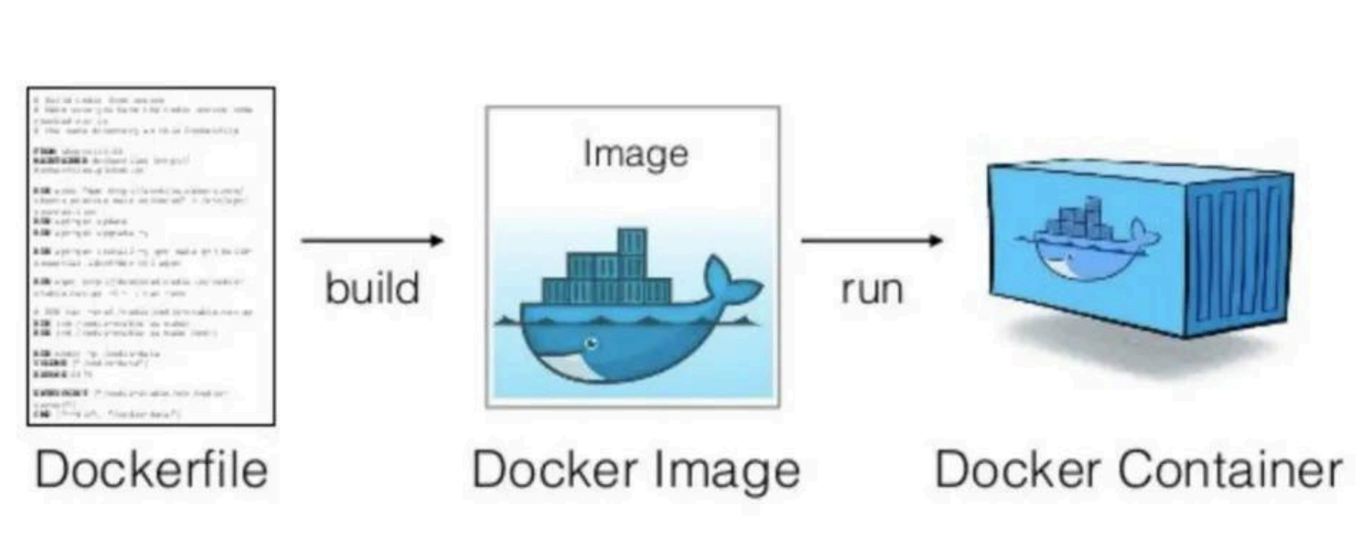
Docker Komutları



-build
Docker görüntüsü oluşturmak için kullanılır
-docker push
Yerel bir Docker imajını bir imaj deposuna yüklemek için kullanılır.
-run
Docker konteynerini çalıştırmak için kullanılır.
-daemon
Docker'ın arka planda çalışan bileşenidir ve Docker API isteklerini dinler, Docker objelerini yönetir ve Docker komutlarını işler.

WEB UYGULAMASI

Örneğin, basit bir web uygulaması seçebiliriz. Bu web uygulaması, frontend ve backend ile birlikte çalışan bir yapıya sahip olabilir. Örneğin, bir basit blog uygulaması düşünebiliriz. Frontend, kullanıcı arayüzünü oluşturan HTML, CSS ve JavaScript dosyalarından oluşurken, backend ise kullanıcıların blog yazılarını görebileceği ve yönetebileceği bir API sunabilir.



Dockerfile

```
# Base image
FROM node:alpine

# Çalışma dizinini belirle
WORKDIR /app

# package.json dosyasını kopyala ve bağımlılıkları yükle
COPY package.json .
RUN npm install

# Geri kalan kaynak kodunu kopyala
COPY . .

# Portu aç
EXPOSE 3000

# Uygulamayı çalıştırmak için komut
CMD ["npm", "start"]
```

Dockerfile dosyası

Dockerfile, bir Docker konteynerinin nasıl oluşturulacağını tanımlayan bir metin dosyasıdır.

Bu Dockerfile, Node.js tabanlı bir web uygulamasının Docker konteynerini oluşturmak için gerekli adımları tanımlar. İlk olarak, Alpine Linux tabanlı bir Node.js imajı kullanır. Ardından, uygulamanın bağımlılıklarını yükler, kodu kopyalar ve 3000 numaralı bir portu açar. Son olarak, uygulamayı başlatmak için **npm start** komutunu çalıştırır.

Docker Compose

```
version: '3'
services:
  frontend:
    build: .
    ports:
      - "3000:3000"
  backend:
    image: backend-image:latest
    ports:
      - "8080:8080"
```

Docker Compose dosyası

Docker Compose, birden fazla Docker konteynerinin tanımlanmasını, yapılandırılmasını ve yönetilmesini sağlayan bir araçtır.

Bu docker-compose.yml dosyası, frontend ve backend adında iki adet servis tanımlar. Frontend servisi, mevcut dizindeki Dockerfile kullanılarak oluşturulur ve 3000 numaralı bir portu host makineye bağlar. Backend servisi ise bir önceden oluşturulmuş bir imajı kullanır ve 8080 numaralı bir portu, host makineye bağlar.

docker-compose up

Docker komutlarını kullanarak Dockerfile'ı ve Docker Compose dosyasını kullanarak uygulamayı çalıştırabilirsiniz.

Backend ve Frontend

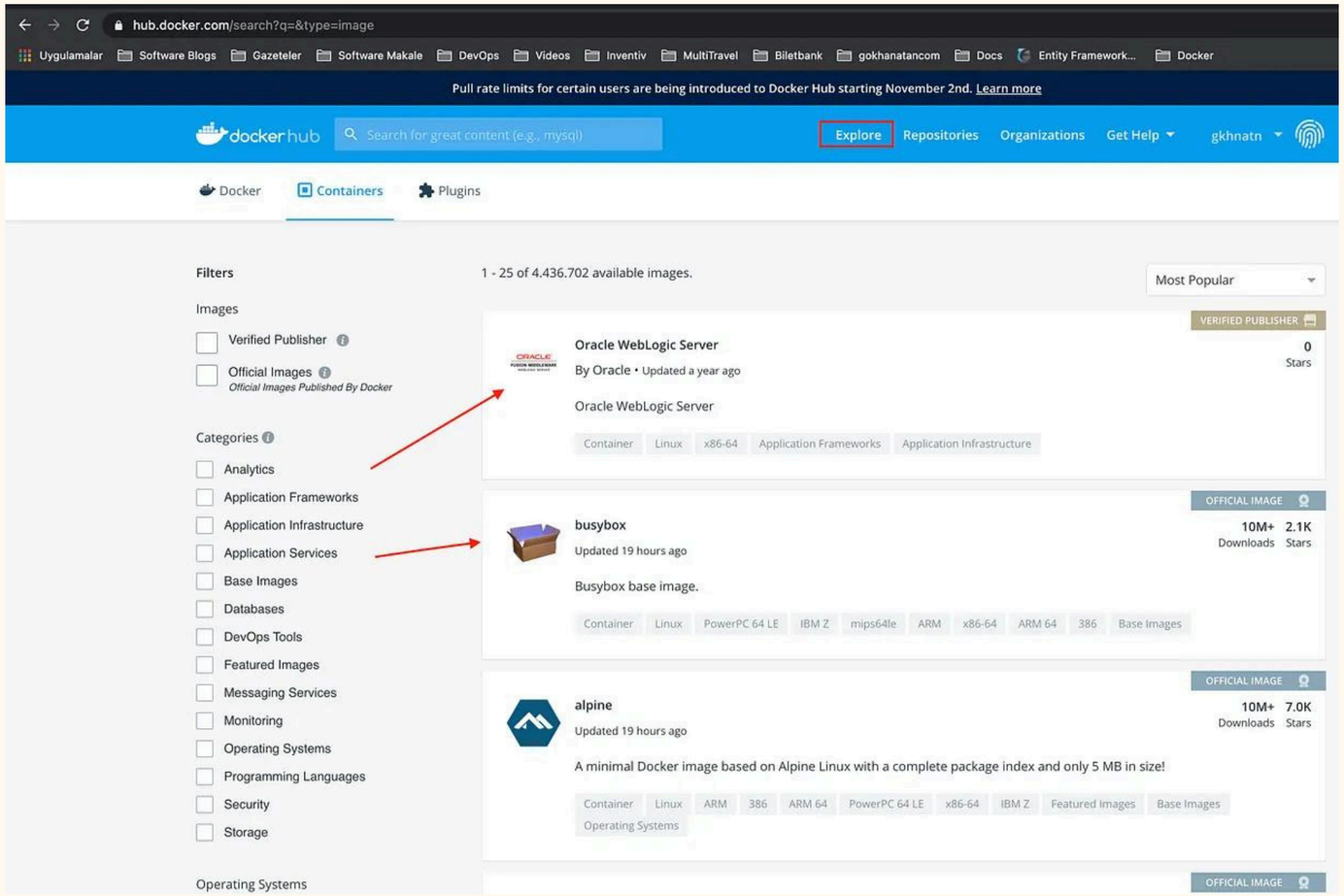
Frontend Uygulaması:

http://localhost:3000
Bu adresi ziyaret ettiğinizde, kullanıcı arayüzünü görüntüleyebilirsiniz. Örneğin, bir blog uygulaması için, blog yazılarına, yorumlara veya diğer içeriğe erişebileceğiniz bir ana sayfa görüntüleyebilirsiniz.

Backend API:

http://localhost:8080
Bu adresi ziyaret ettiğinizde, backend servisinizin API'sine erişeceksiniz. Örneğin, "/posts" gibi bir endpoint çağırarak, mevcut blog yazılarını veya "/users" gibi bir endpoint çağırarak kullanıcı bilgilerini alabilirsiniz.

Docker Hub



Docker Hub, Docker Image'larının paylaşılabileceği bir online servistir. Bu serviste Docker'ın kendisinin, ünlü şirketlerin ya da insanların oluşturduğu hazır Docker Image'ları bulunmaktadır.

Tarayıcınıza 'https://hub.docker.com/' yazarak Docker Hub'ı inceleyebilirsiniz.

KAYNAKLAR

1.https://docs.docker.com/

2.Docker: Up & Running

3.https://www.docker.com/blog/

SORULARINIZI ALABİLİRİM

DİNLEDİĞİNİZ İÇİN TEŞEKKÜR EDERİM