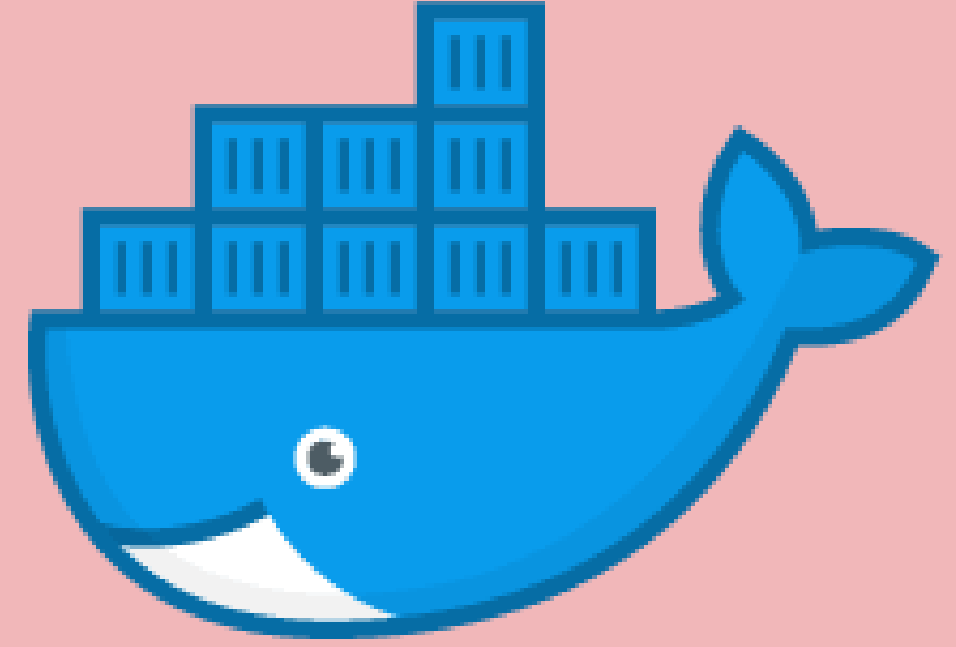


Docker Nedir ve Temel Kullanımı

Ufuk Vural

Docker nedir?

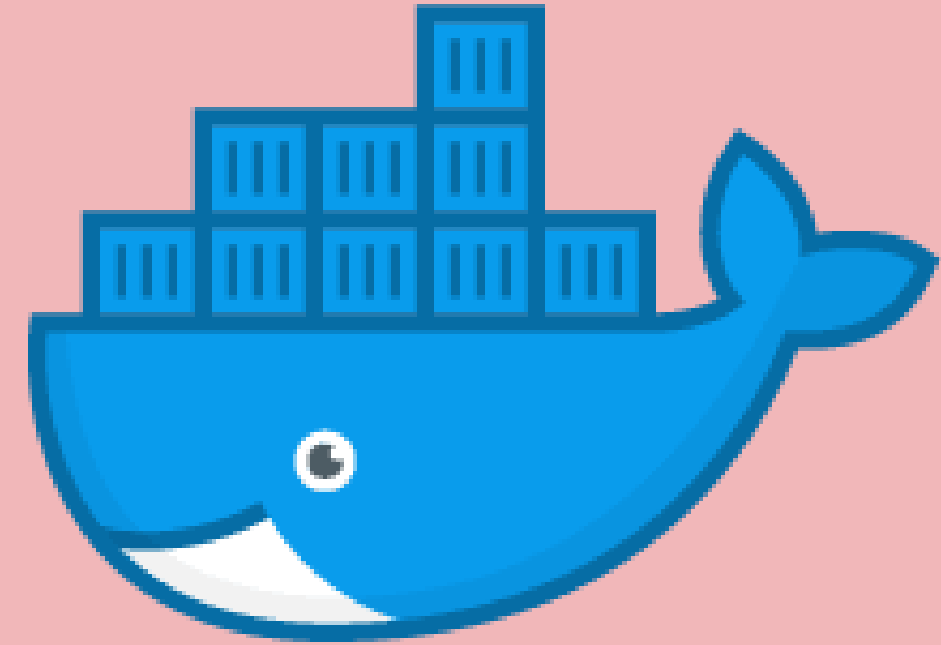
- Docker, uygulamaları ve bağımlılıklarını birlikte paketleyerek çalıştıran bir konteyner platformudur.
- Sanal makineler gibi çalışır ama daha hafif ve hızlıdır.
- Uygulama geliştirme, test etme ve dağıtım süreçlerini kolaylaştırır.



docker

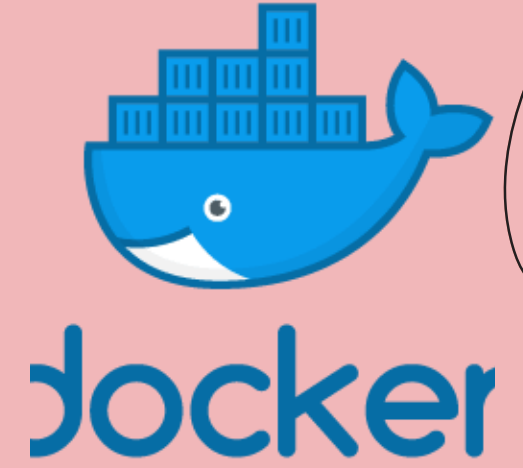
Docker'ın Avantajları

- Platformdan bağımsız çalışma
- Hızlı dağıtım ve başlatma süresi
- Geliştirici ekipler için standart bir ortam sağlar
- Daha kolay versiyonlama ve taşınabilirlik



docker

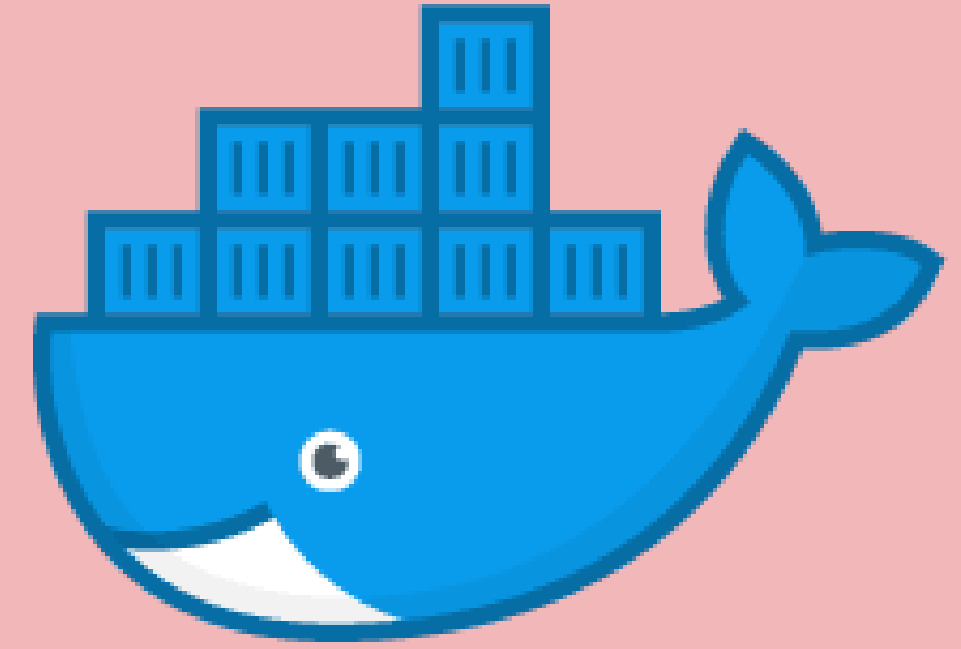
Docker'ın Avantajları



Özellik	Sanal Makine	Docker Konteyner
Boyut	GB	MB
Açılış Süresi	Dakikalar	Saniyeler
Sistem	Kendi işletim sistemi	Ana sistemin çekirdeğini kullanır
Performans	Daha düşük	Daha yüksek

Docker Bileşenleri

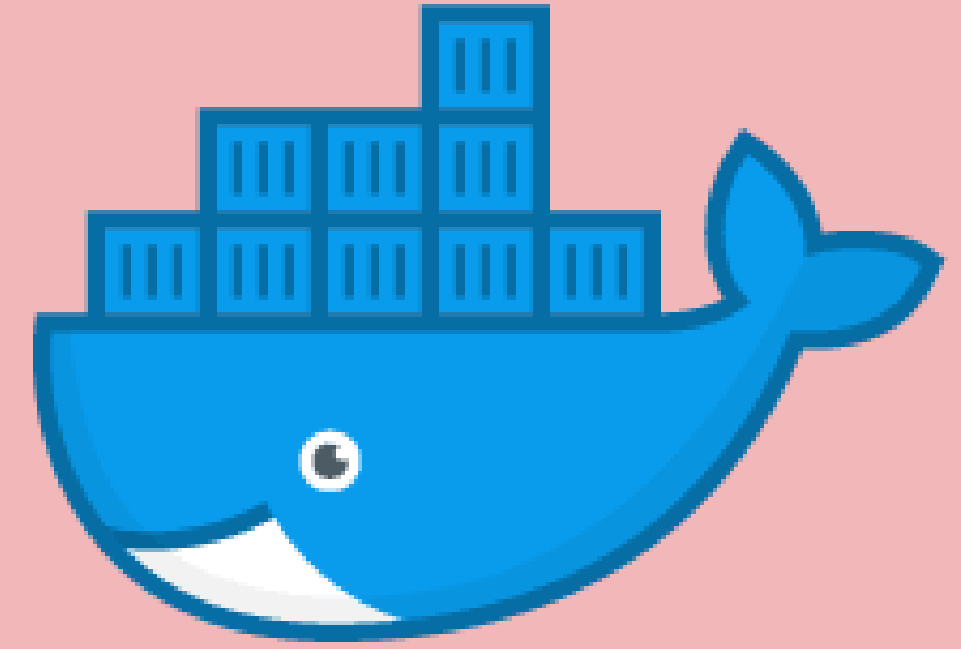
- Image: Konteynerin şablonudur. Değişmez.
- Container: İmajdan oluşturulan çalışan ortamdır.
- Dockerfile: Bir imajı tanımlamak için kullanılan betik.
- Docker Hub: Resmi imajların paylaşıldığı depo.
- Docker CLI: Komut satırından Docker'ı yönetmeyi sağlar.



docker

Dockerfile Nedir?

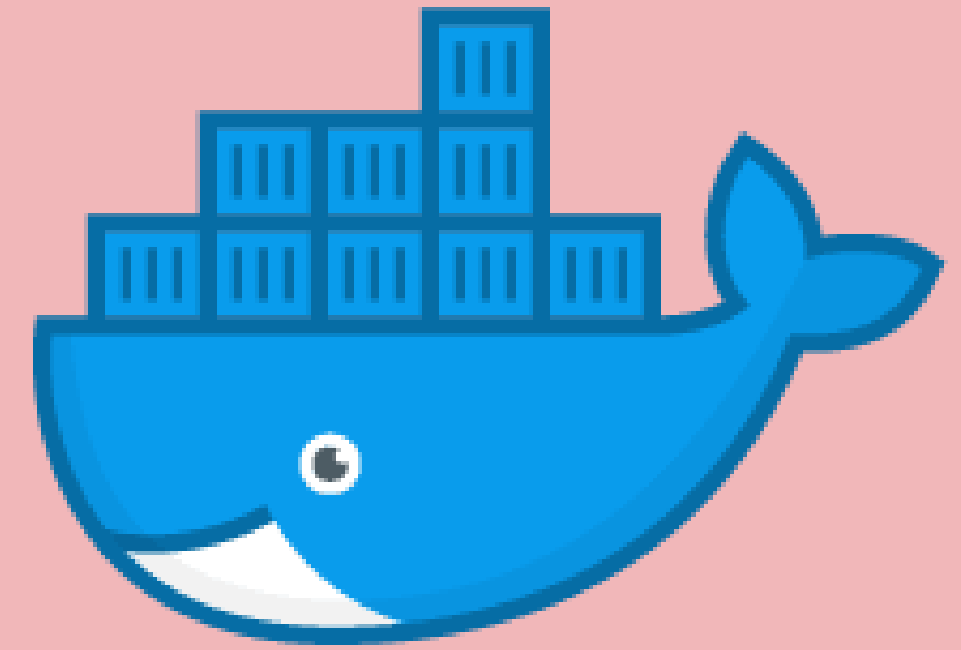
- Dockerfile Nedir?
- Dockerfile, imaj oluşturmak için kullanılan tarif dosyasıdır.
- İçeriği genelde:
 - Base image (örn: python:3.9)
 - Gerekli dosyaların kopyalanması
 - Bağımlılıkların yüklenmesi
 - Uygulamanın çalıştırılması
 - Örnek bir Python uygulaması
- Dockerfile'ı:



docker

Örnek bir Python uygulaması Dockerfile'ı:

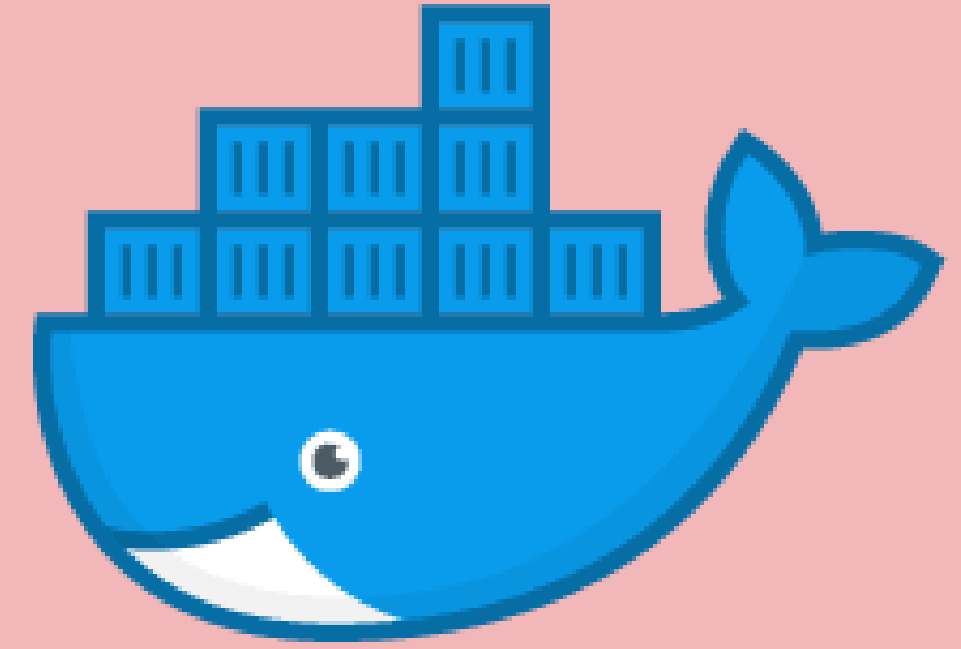
- FROM python:3.9
- WORKDIR /app
- COPY . /app
- RUN pip install -r requirements.txt
- CMD ["python", "app.py"]



docker

Uygulamanın Konteynerleştirilmesi - Adım Adım

- Python dosyalarını hazırla (app.py, requirements.txt)
- Aynı klasöre bir Dockerfile oluştur
- Terminalde aşağıdaki komutları sırayla çalıştır:
- `docker build -t my-python-app .`
- `docker run -p 5000:5000 my-python-app`

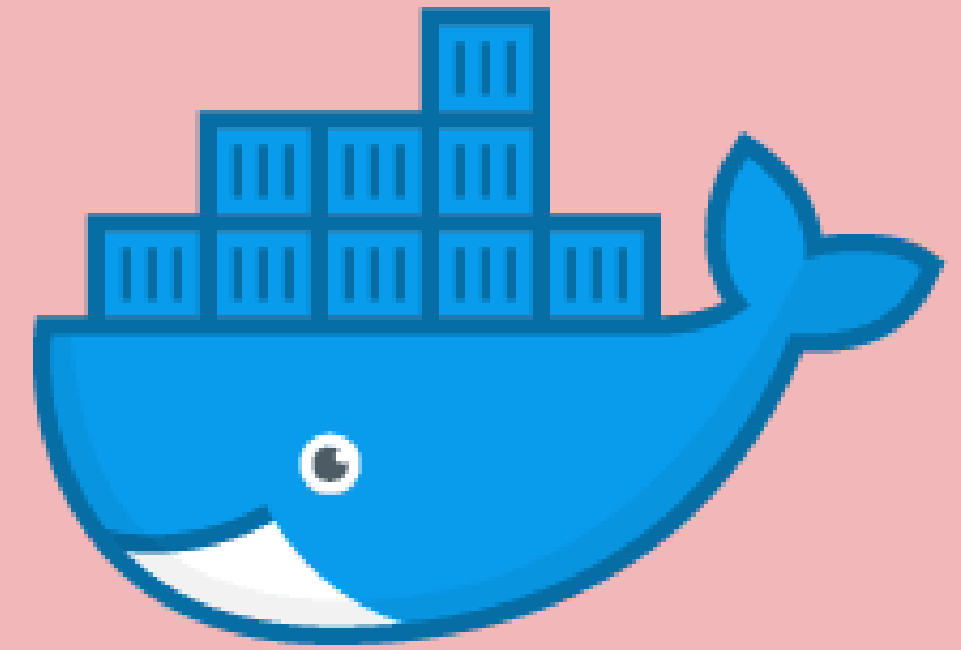


docker

Örnek Uygulama Yapısı

- proje/
 - |
 - |—— app.py
 - |—— requirements.txt
 - |—— Dockerfile

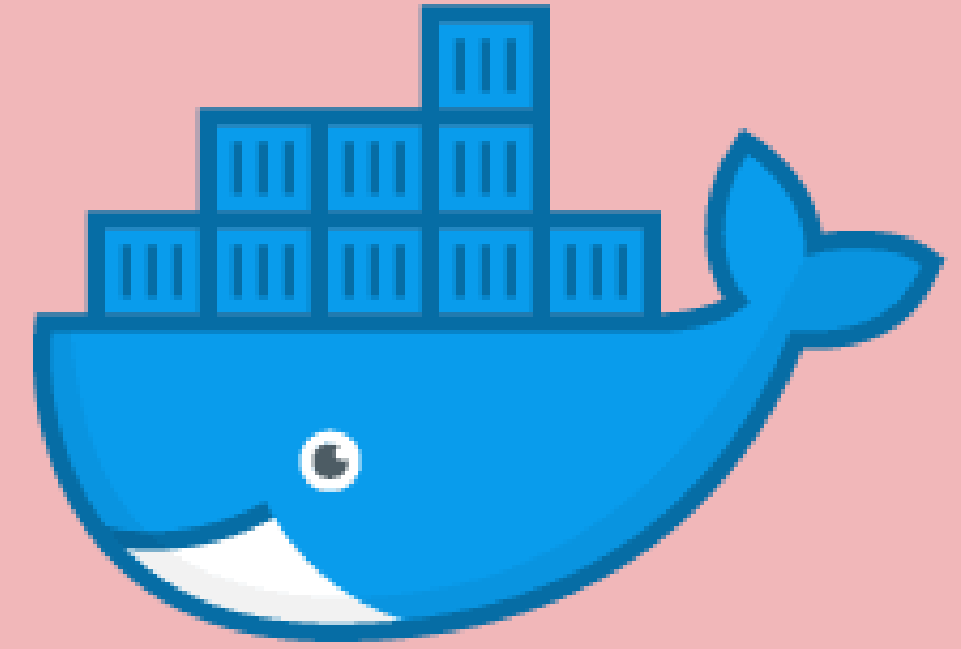
- app.py: Python uygulaman
- requirements.txt: Kütüphane bağımlılıkları
- Dockerfile: Docker yapı tanımı



docker

Docker Komutlarına Giriş

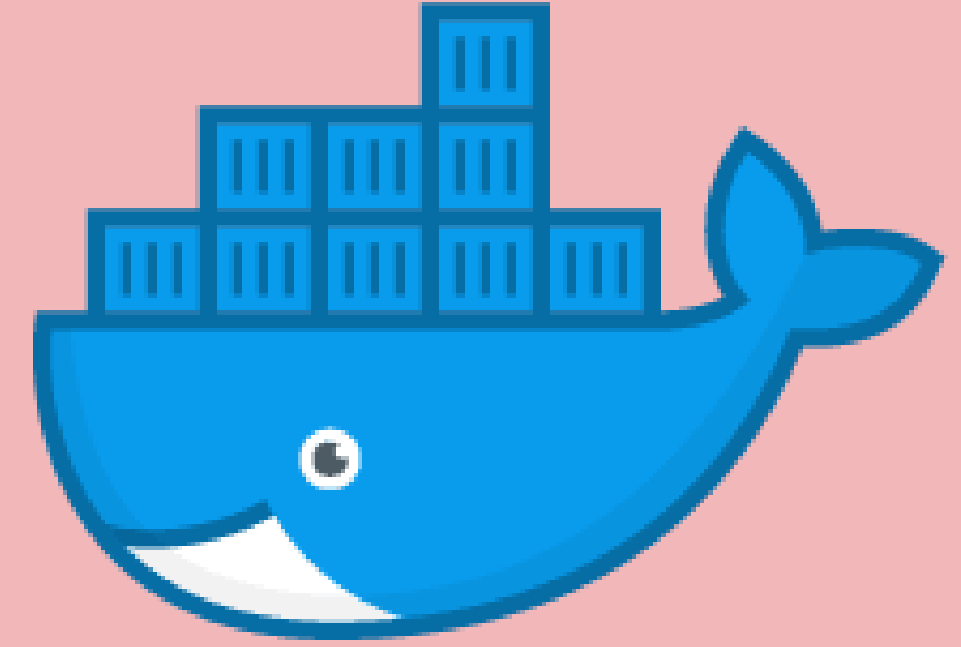
- `docker build -t isim .` → İmaj oluşturur
- `docker run -p 5000:5000 isim` → Konteyneri başlatır
- `docker ps` → Çalışan konteynerleri gösterir
- `docker stop <id>` → Konteyneri durdurur
- `docker images` → Mevcut imajları listeler
- `docker rm <id>` ve `docker rmi <image>` → Temizleme



docker

Docker'ın Avantajları

- Tutarlı çalışma ortamı
- Platformdan bağımsızlık
- Hızlı dağıtım ve başlatma
- Daha kolay CI/CD entegrasyonu
- Ölçeklenebilirlik ve mikro servis desteği

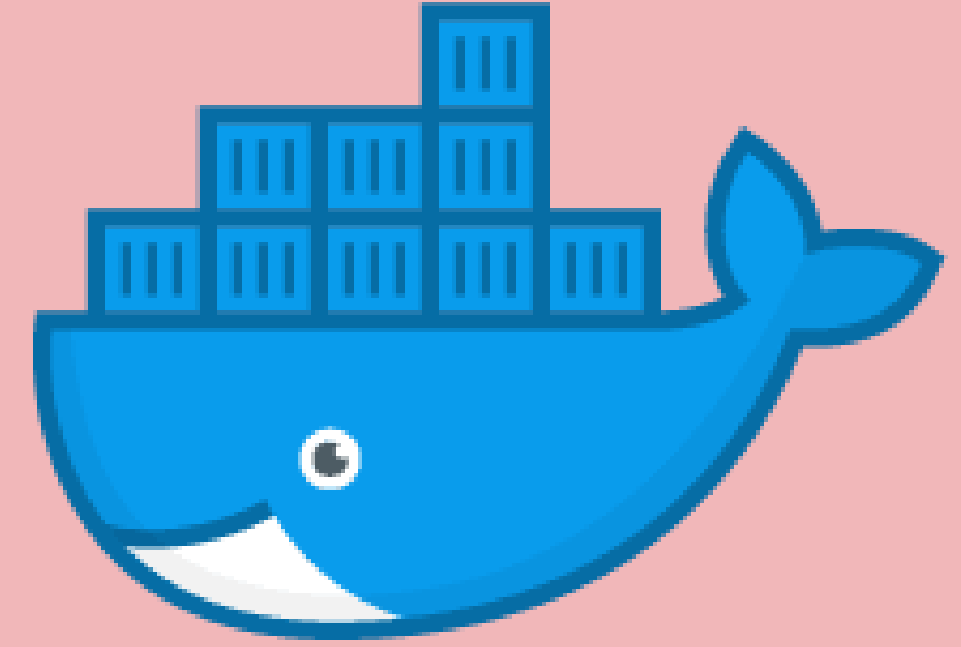


docker

Gerçek Hayatta Kullanım Senaryosu

Örnek:

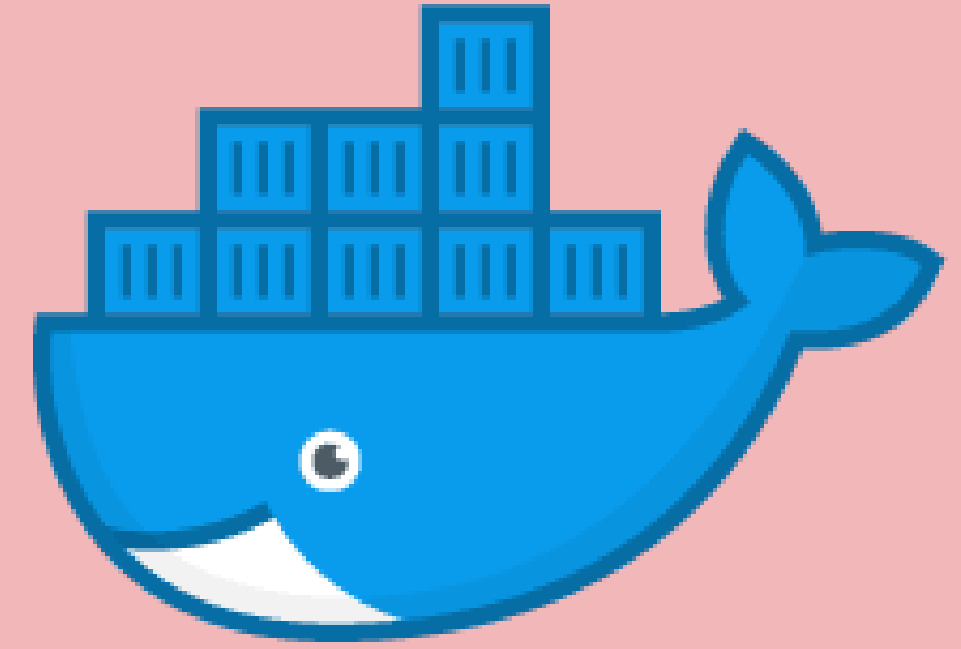
- Bir ekip Flask ile API geliştiriyor.
- Her geliřtiricide farklı işletim sistemi, Python sürümü var.
- Docker ile tüm ekip aynı ortamda geliştirme yapabiliyor.
- Canlı sunucuya birebir aynı imaj gönderiliyor.



docker

Özet

- Docker yazılım geliştirmeyi kolaylaştıran güçlü bir araçtır.
- Konteynerler sayesinde ortam tutarsızlıkları ortadan kalkar.
- Dockerfile ile uygulamalar kolayca taşınabilir hale gelir.
- Python uygulamaları dakikalar içinde konteynerleştirilebilir.



docker



teşekkür ederim.

Ufuk Vural