Bulutta Sanallaştırılmış Uygulama

Eraycan Çoban  
 *Bilişim Sistemleri Mühendisliği*  
 Kocaeli UniversityKocaeli,Türkiye  
eraycancoban@gmail.com

***Özet*—Bu rapor bulutta sanallaştırılmış bir uygulamanın adımlarını ve sanallaştırma hakkında bilgi içeren bir dökümandır.http://ec2-16-16-70-158.eu-north-1.compute.amazonaws.com:5000/**

***Anahtar Kelimeler—Sanallaştırma,Hyper-v,AWS,Bulut***

# Sanallaştırma

This Bu döküman sanallaştırma ve sanallaştırmanın uygulanması üzerinde yapılan çalışmalardır.

## Depolama Sanallaştırma

Depolama sanallaştırma, depolama altyapısını sanal makineler veya uygulamalar için daha etkili, esnek ve yönetilebilir hale getiren bir teknoloji ve sanallaştırma formudur. Bu, depolama kaynaklarının sanal düzeyde bölünebilmesine, yönetilebilmesine ve paylaşılabilmesine olanak tanır. Depolama sanallaştırma, işletmelerin depolama altyapılarını optimize etmelerine, verimliliklerini artırmalarına ve daha hızlı yanıt vermelerine yardımcı olabilir.

## Uygulama Sanallaştırma

Uygulama sanallaştırma, bir uygulamanın çalışma ortamını, bağımsız olarak yürütülebilen ve izole edilebilen bir sanal konteyner içine yerleştiren bir sanallaştırma formudur. Bu, uygulamaların bir bilgisayar sistemine veya sunucuya kurulu olduğu ortamdan bağımsız olarak çalışabilmesini sağlar. Uygulama sanallaştırma, geleneksel kurulumlara kıyasla daha hızlı dağıtım, daha etkili kullanım ve daha kolay yönetim gibi avantajlar sunabilir.

## Ağ Sanallaştırma

Ağ sanallaştırma, ağ kaynaklarını sanal bir ortamda birleştirme ve yönetme sürecidir. Bu, ağ üzerindeki cihazları, bağlantıları ve hizmetleri sanal katmanlar içinde izole etmeyi ve yönetmeyi sağlar. Ağ sanallaştırma, ağ altyapısını daha esnek, ölçeklenebilir ve yönetilebilir hale getirir. Sanal ağlar, sanal anahtarlar, sanal yönlendiriciler ve sanal ağ fonksiyonları gibi kavramlar içerir. Bu, özellikle bulut bilişim ortamlarında ve geniş alan ağlarında kullanılır.

## Masaüstü Sanallaştırma

Masaüstü sanallaştırma, masaüstü işletim sistemlerini ve uygulamalarını merkezi bir sunucu üzerinde barındırarak kullanıcılara uzaktan erişim sağlayan bir teknolojidir. Bu, kullanıcılara farklı cihazlardan aynı masaüstü deneyimini yaşatır. Masaüstü sanallaştırma, güvenlik, merkezi yönetim, yedekleme ve uygulama uyumluluğu gibi avantajlar sağlar. Kullanıcılar, masaüstü sanallaştırma sayesinde bir web tarayıcısı veya özel bir istemci aracılığıyla uzak bir sunucuya bağlanarak işlerini sürdürebilirler.

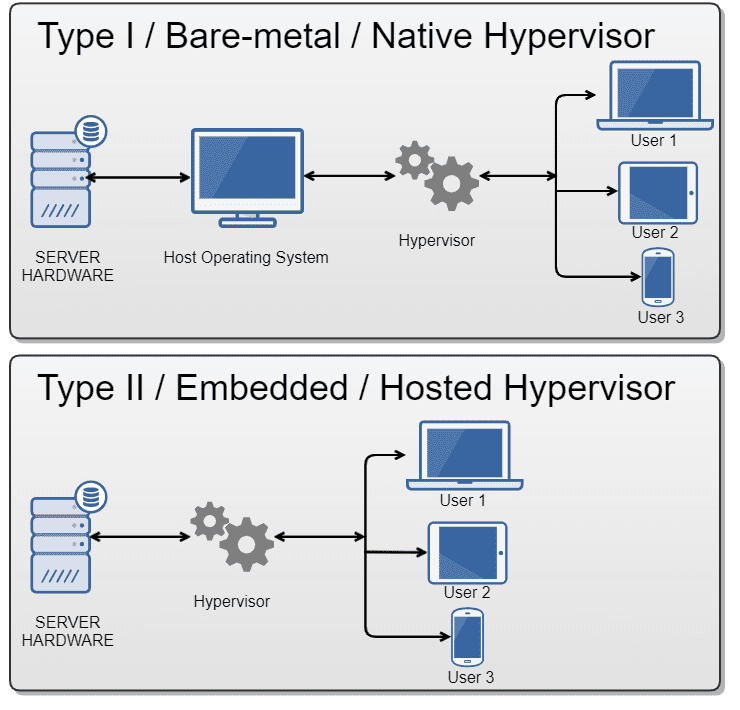
## Sunucu Sanallaştırma

Sunucu sanallaştırma, bir fiziksel sunucu üzerinde çalışan birden çok sanal sunucuyu barındırma sürecidir. Bu, her bir sanal sunucunun bağımsız olarak çalışabilmesini ve kaynakları diğer sanal sunuculardan izole etmesini sağlar. Sunucu sanallaştırma, donanım kaynaklarının daha etkili kullanılmasını, hızlı dağıtımı ve yönetim kolaylığını sağlar. Popüler sunucu sanallaştırma platformları arasında VMware vSphere, Microsoft Hyper-V ve KVM bulunmaktadır.

# Sanallaştırma Teknolojileri

## Hyper-Visor

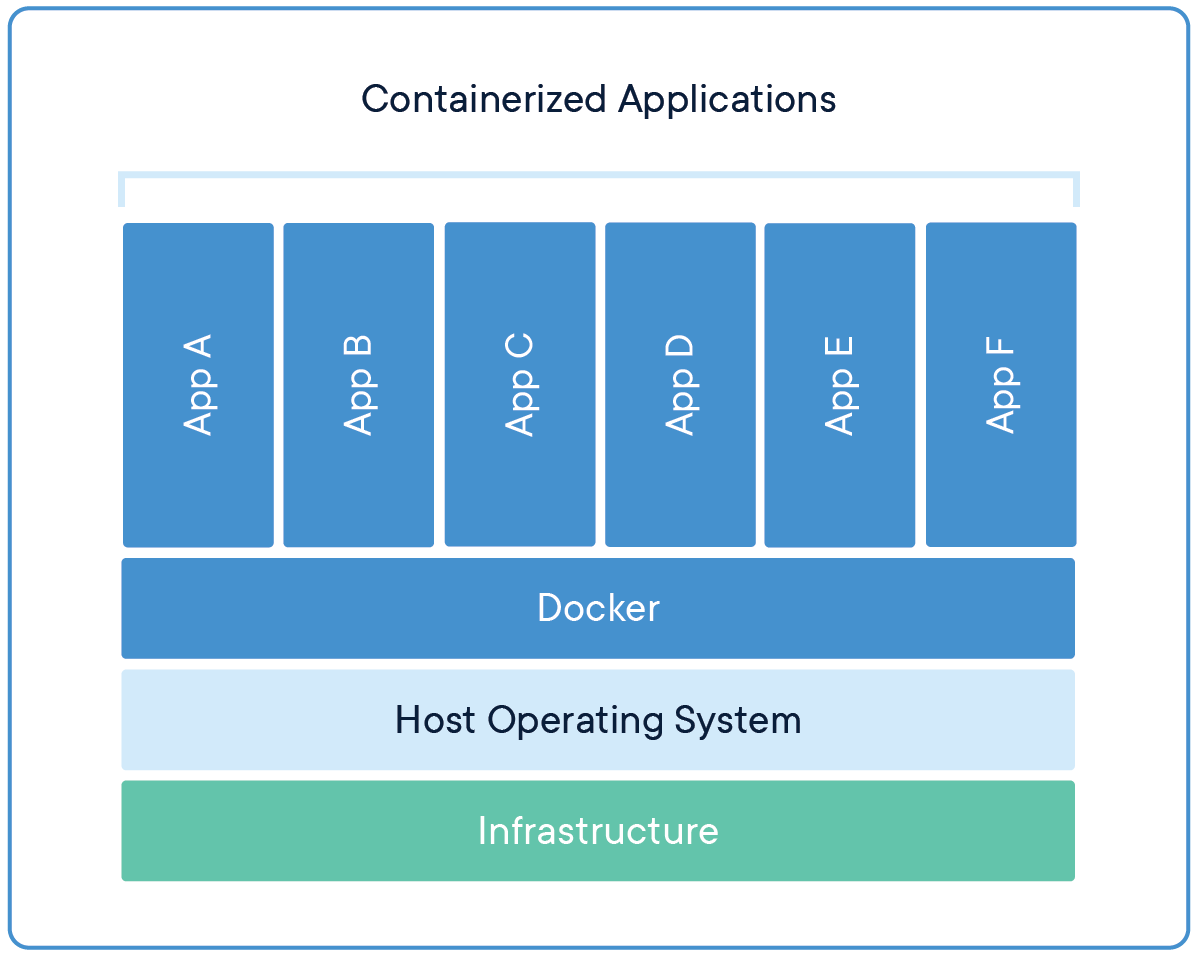
Sanal makinelerin oluşturulmasını, yönetilmesini ve izlenmesini sağlayan yazılım veya donanım tabanlı bir katmandır. Tip 1 hypervisor doğrudan donanım üzerinde çalışırken, Tip 2 hypervisor bir işletim sistemi üzerinde çalışır.



Görsel 2.1 Hyper-V Çeşitleri

## Konteynerlar

Uygulama ve bağımlılıklarını bir konteyner içinde paketleme ve bu konteynerı bir host sistemde çalıştırma işlemidir. Konteynerler, sanal makinelerden daha hafif ve daha hızlı başlatılabilir. Docker ve Kubernetes gibi araçlar popüler konteynerleme teknolojileridir.



Görsel 2.1 Hyper-V Çeşitleri

## Aralarındaki Farklar

### İzolasyon Seviyesi:

**Hypervisor**: Sanal makineler birbirinden tamamen izole edilmiştir. Bu, her sanal makinenin kendi işletim sistemini ve çekirdeğini içerdiği anlamına gelir.

**Konteynerlar**: Konteynerlar, bir host işletim sistemi çekirdeğini paylaşır. Bu, izolasyon seviyesinin daha düşük olduğu anlamına gelir, ancak aynı zamanda daha hafif ve hızlı başlatılabilir olmalarını sağlar.

### Kaynak Kullanımı:

**Hypervisor**: Sanal makineler, kendi işletim sistemleriyle birlikte gelen ek kaynakları (bellek, depolama) tüketir.

**Konteynerlar**: Konteynerlar, host işletim sistemi üzerinde çalıştıkları için daha hafif ve kaynak kullanımı daha etkili olabilir.

### Bağımlılık Yönetimi:

**Hypervisor**: Sanal makineler, kendi işletim sistemleri ve bağımlılıkları ile birlikte gelir. Her sanal makine, kendi kopyasını içerir.

**Konteynerlar**: Konteynerlar, uygulama ve bağımlılıklarını izole edilmiş bir ortamda barındırır. Konteyner, kendi dosya sistemini ve bağımlılıklarını içerir.

### Performans:

**Hypervisor**: Sanal makineler genellikle daha yavaş başlar ve daha fazla kaynak tüketir, çünkü her biri kendi işletim sistemini çalıştırır.

**Konteynerlar**: Konteynerlar hızlı başlar ve daha düşük kaynak tüketir. Konteynerlar genellikle daha yüksek performans sunar.

### Taşınabilirlik:

**Hypervisor**: Sanal makineler genellikle farklı hypervisor'lar arasında taşınabilir değildir.

**Konteynerlar**: Konteynerlar, bağımsız olarak çalışabilen ve farklı ortamlarda taşınabilen birimlerdir. Bu, daha iyi taşınabilirlik sağlar.

# Docker

Docker, yazılım uygulamalarını paketleme, taşıma ve çalıştırma için bir platform sağlayan açık kaynaklı bir konteynerizasyon teknolojisidir. Konteynerler, bir uygulamanın tüm bağımlılıklarını (kod, çalışma zamanı, sistem araçları, kütüphaneler ve ayarlar) tek bir paket içinde bir araya getirir. Docker, bu konteyner teknolojisini kullanarak uygulamaların farklı ortamlarda tutarlı bir şekilde çalışmasını sağlar.

Temel bileşenleri şunlardır:

### Docker Konteynerleri:

Docker, uygulamaları ve bağımlılıklarını bir araya getiren, izole edilmiş, hafif çalışma ortamları olan konteynerlere sahiptir. Bu konteynerler, işletim sistemi düzeyinde sanallaştırma kullanarak bir uygulamanın izolasyonunu sağlar.

### Docker Görüntüleri:

Docker konteynerlerini oluşturan şablonlar olarak düşünülebilecek Docker görüntüleri, bir uygulamanın çalışması için gereken tüm bileşenleri içerir. Bu görüntüler, paylaşılabilir ve kullanıma hazır paketler olarak depolanabilir.

### Docker Hub:

Docker Hub, kullanıcıların Docker görüntülerini depolayabileceği ve paylaşabileceği bir çevrimiçi servistir. Bu, kullanıcıların başkaları tarafından oluşturulan görüntüleri kullanmalarına ve kendi oluşturdukları görüntüleri paylaşmalarına olanak tanır.

Docker, yazılım geliştirme süreçlerini hızlandırır, uygulama dağıtımını kolaylaştırır ve çeşitli ortamlarda tutarlılığı sağlar. Ayrıca, daha hafif ve hızlı bir sanallaştırma çözümü sağlayarak kaynakları daha etkili bir şekilde kullanmaya olanak tanır. Bu nedenle, Docker, özellikle mikro hizmet mimarileri ve sürekli entegrasyon/deployment (CI/CD) süreçlerinin geliştirilmesi gibi modern yazılım geliştirme uygulamalarında yaygın olarak kullanılmaktadır.

# Bulut

"Bulut" terimi genellikle "cloud" olarak İngilizce'de ifade edilen "cloud computing" ya da "bulut bilişim" anlamına gelir. Bulut bilişim, bilgi işlem, veri depolama, ağ kullanımı ve diğer bilgisayar kaynaklarına internet üzerinden erişim sağlayan bir teknoloji modelidir. Bu hizmetler genellikle bir hizmet sağlayıcının sunucuları üzerinden sunulur ve kullanıcılar bu hizmetlere internet aracılığıyla erişir.

Bulut bilişim, fiziksel olarak uzak bir konumda bulunan sunucu ve veri merkezlerini kullanarak çeşitli hizmetleri sağlar. Bu hizmetler arasında web tabanlı e-posta, dosya depolama, veritabanı yönetimi, ağ servisleri ve hesaplamalı güç gibi birçok farklı alan bulunabilir. Kullanıcılar genellikle bu hizmetlere ihtiyaçları doğrultusunda abonelik temelli bir modelle erişir ve bu sayede kendi bilgisayarlarına büyük ölçüde bağımlı olmadan bu hizmetleri kullanabilirler.

Bulut bilişim, kaynakların paylaşımını artırarak iş süreçlerini ve verimliliği geliştirebilir. Ayrıca, kullanıcılara esneklik ve ölçeklenebilirlik avantajları sunar. Bulut bilişim, işletmelerin ve bireylerin büyük miktarda veriyi depolamasına, işlemesine ve paylaşmasına olanak tanır. Bu nedenle, günümüzde birçok organizasyon ve kişi bulut bilişim hizmetlerini kullanarak günlük iş süreçlerini yönetmektedir.

## AWS

Amazon Web Services (AWS), dünya genelinde lider bir bulut bilişim hizmet sağlayıcısıdır ve Amazon.com'un bir yan kuruluşudur. AWS, geniş bir hizmet yelpazesi sunarak şirketlere, geliştiricilere ve kuruluşlara çeşitli bulut tabanlı hizmetlerle iş süreçlerini optimize etme ve inovasyonlarını hızlandırma imkanı sağlar. AWS'nin sunduğu hizmetler arasında bilgi işlem gücü sağlama, veri depolama ve yönetme, veritabanı hizmetleri, yapay zeka ve makine öğrenimi, ağ altyapısı, güvenlik çözümleri, oyun geliştirme, analitik araçlar ve daha birçok özellik bulunmaktadır.

AWS, geniş global altyapısı sayesinde müşterilerine dünya çapında hızlı ve güvenilir bir erişim sunar. Bu, kullanıcıların uygulamalarını daha ölçeklenebilir hale getirmelerini, talebe göre kaynakları anında ayarlamalarını ve yüksek performans elde etmelerini sağlar. AWS aynı zamanda müşterilerine ödeme üzerine esnek bir model sunarak, sadece kullandıkları hizmetler için ödeme yapmalarına imkan tanır. Bu da işletmelerin maliyetleri optimize etmelerine ve bütçe planlamalarını daha etkili bir şekilde yönetmelerine yardımcı olur. AWS, geniş müşteri tabanı, güvenlik önlemleri ve sürekli yenilikleri ile dünya genelinde birçok sektörde güvenilen bir bulut bilişim çözümü olarak öne çıkmaktadır.

# Uygulama

## Amacı

Youtube videolarını çevrimdışı izlemek isteyen kişiler için videoların yerel olarak depolanmasını yani indirilmesini sağlamak böylece internet olmadığı zamanlarda dahi videoları izlemek.

## Araçlar

### Flask

Flask, Python tabanlı bir web uygulama çerçevesidir. Hafif ve modüler yapısıyla bilinir.Minimalist bir tasarıma sahiptir, bu da geliştiricilere ihtiyaçlarına göre özelleştirme olanağı tanır.Özellikle küçük ve orta ölçekli web uygulamaları için uygundur.

### Python

Python, genel amaçlı bir programlama dilidir ve web geliştirmenin yanı sıra birçok farklı alanda kullanılır.

Flask gibi web çerçeveleri (Django, Flask, FastAPI vb.) kullanılarak web uygulamaları geliştirmek için sıklıkla tercih edilir.

Python, basit ve okunabilir bir sözdizimine sahiptir, bu da geliştiricilerin hızlı ve etkili bir şekilde kod yazmalarına olanak tanır.

### Jquerry

jQuery, JavaScript tabanlı bir kütüphanedir ve web sayfalarındaki etkileşimleri ve DOM manipülasyonunu kolaylaştırır.

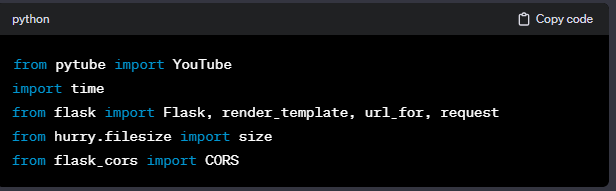
Farklı tarayıcılar arasındaki uyumluluğu artırmak ve karmaşık JavaScript kodlarını daha basit ve kısa bir şekilde yazmak için kullanılır.

Özellikle AJAX çağrıları ve animasyonlar gibi web sayfası etkileşimleri için popülerdir.

# Kodun Yapısı

## Python Dosyası

### Kullanılan Kütüphaneler:



**pytube**: YouTube videolarını indirmek için kullanılır.

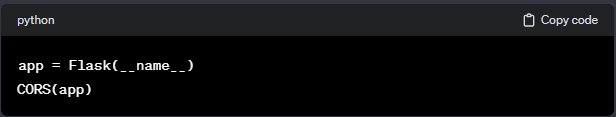
**time**: Zamanla ilgili işlemleri gerçekleştirmek için kullanılır.

**flask**: Web uygulaması oluşturmak ve yönetmek için kullanılır.

**hurry.filesize**: Dosya boyutlarını insan dostu bir formatta göstermek için kullanılır.

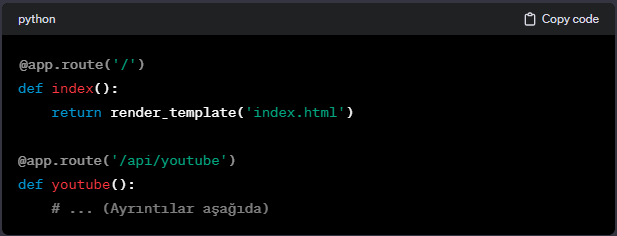
**flask\_cors:** CORS (Cross-Origin Resource Sharing) problemlerini çözmek için kullanılır.

### Flask Uygulaması Oluşturma:



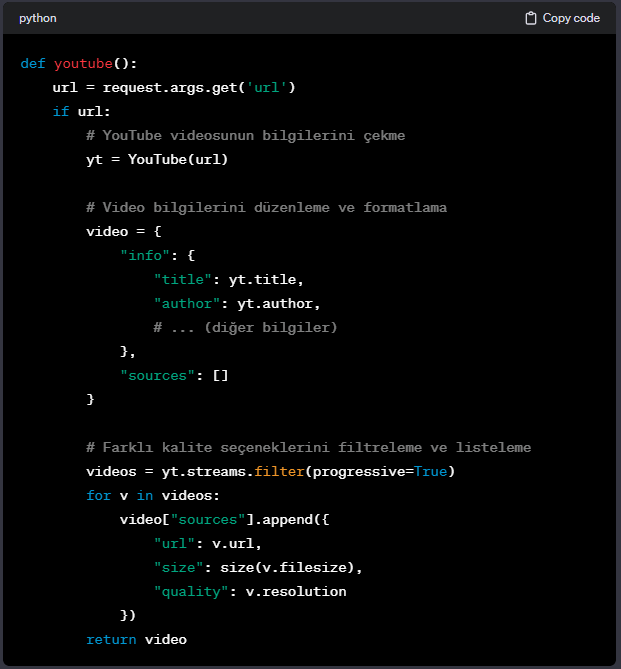
Flask uygulaması oluşturulur ve CORS desteği eklenir.

### Ana Sayfa ve API Endpoint'leri:



* / endpoint'i, kullanıcıların uygulamaya giriş yapmasını sağlayan ana sayfayı gösterir.
* /api/youtube endpoint'i, YouTube videosu bilgilerini ve indirme seçeneklerini sağlayan bir API endpoint'ini temsil eder.

### YouTube Fonksiyonu:

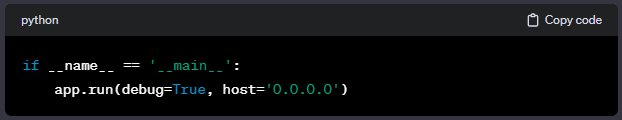


/api/youtube endpoint'i için ana işlevi temsil eder.

Kullanıcının sağladığı YouTube video URL'sini alır ve pytube kütüphanesi aracılığıyla video bilgilerini çeker.

Çekilen bilgileri düzenler ve bir JSON formatında geri döner.Video kaynaklarını (farklı kalite seçenekleri) listeler.

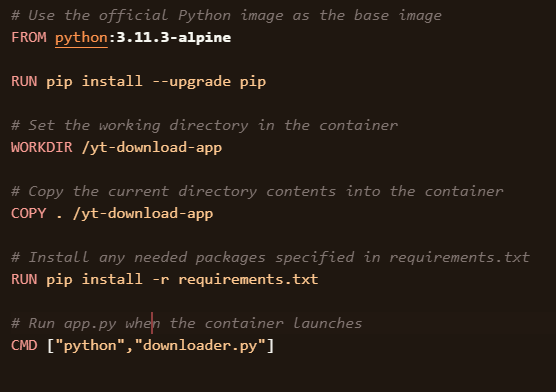
### Uygulamayı Başlatma:



Uygulamayı geliştirme modunda (debug=True) ve belirli bir host üzerinde çalıştırma komutudur.

Bu kod, Flask ve pytube kullanarak basit bir YouTube video indirme web uygulaması oluşturur. Kullanıcılar, uygulamaya bir YouTube video URL'si girerek videonun bilgilerini alabilir ve farklı kalite seçeneklerinde indirme bağlantılarına erişebilirler.

## Dockerfile



**FROM python:3.11.3-alpine:** Bu satır, temel olarak alpine Linux tabanlı Python 3.11.3 görüntüsünü kullanacağını belirtir. Alpine Linux, hafif bir Linux dağıtımıdır, bu da oluşturulan Docker görüntüsünün daha küçük olmasını sağlar.

**RUN pip install --upgrade pip:** Bu satır, pip (Python paket yöneticisi) aracılığıyla yüklü olan pip'in güncellenmesini sağlar.

**WORKDIR /yt-download-app:** Bu satır, çalışma dizinini /yt-download-app olarak ayarlar. Bundan sonraki talimatlar bu dizinde gerçekleşecektir.

**COPY . /yt-download-app:** Bu satır, Dockerfile'ın bulunduğu dizindeki tüm dosyaları (/yt-download-app içinde) Docker konteynerine kopyalar.

**RUN pip install -r requirements.txt:** Bu satır, requirements.txt dosyasındaki Python bağımlılıklarını yükler. Bu, uygulamanın çalışması için gerekli olan dış paketleri içerir.

**CMD ["python", "downloader.py"]:** Bu satır, Docker konteyneri başlatıldığında çalıştırılacak komutu belirtir. Bu örnekte, "python downloader.py" komutu çalıştırılır, bu da "downloader.py" adlı bir Python betiğinin başlatılmasını sağlar.

# AWS Servisleri

AWS ECS (Amazon Elastic Container Service), Amazon Web Services (AWS) tarafından sağlanan bir konteyner orkestrasyon hizmetidir. Bu hizmet, Docker konteynerlarını yönetmek ve dağıtmak için tasarlanmıştır. ECS, konteynerlarınızı ölçeklendirmek, güvenli bir şekilde çalıştırmak ve yönetmek için kullanılan bir platform sağlar.

ECS, aşağıdaki temel bileşenlere dayanır:

### Cluster (Küme):

ECS'de konteynerları barındıran ve yöneten bir grup EC2 örneği veya AWS Fargate görev tanımlarının bulunduğu bir kümedir. Bir kümede birden çok EC2 örneği veya Fargate görevi bulunabilir.

### Task Definition (Görev Tanımı):

Konteynerlarınızın nasıl çalıştırılacağını ve yapılandırılacağını tanımlayan bir JSON şablonudur. Görev tanımı, konteynerlarınızın kaynaklarını, bağımlılıklarını ve diğer konfigürasyon bilgilerini içerir.

### Service (Hizmet):

Bir ECS servisi, belirli bir görev tanımı temelinde çalışan ve ölçeklendirilebilen bir uygulamayı temsil eder. Servisler, belirli bir sayıda çalışan görevi koruyarak veya istenen bir kapasiteyi sürdürerek uygulamanın sürekli çalışmasını sağlar.

### Task (Görev):

Bir görev, bir görev tanımı temelinde çalışan bir konteyner örneğini ifade eder. Her görev bir veya birden çok konteyner içerebilir.

ECS'nin ana avantajları şunlardır:

* **Esnek Ölçekleme:** Uygulamanızın ihtiyaçlarına göre konteyner sayısını otomatik olarak ölçeklendirme yeteneği.
* **Yüksek Hizmetli Konteyner Çalışma Ortamı:** ECS, yüksek kullanılabilirlik ve dayanıklılık sağlamak için AWS'nin geniş altyapısını kullanır.
* **Integrasyon:** Diğer AWS hizmetleri ile entegrasyon sağlar. Örneğin, Amazon RDS gibi yönetilen veritabanlarına veya Amazon CloudWatch gibi izleme ve günlükleme hizmetlerine entegre olabilir.
* **Fargate Desteği:** ECS, EC2 örnek kullanmadan, AWS Fargate ile de konteynerları çalıştırabilir. Fargate, konteynerları yönetmek için bir hizmet olup, altta yatan altyapıyı yönetme ihtiyacını ortadan kaldırır.

ECS, mikro servis mimarileri, sürekli entegrasyon/dağıtım (CI/CD) süreçleri ve konteyner tabanlı uygulama dağıtımları için popüler bir seçenektir.

<https://drive.google.com/drive/folders/11kphqO6QXBz71vtD7T3DuvFPfBLWp_lw?usp=sharing>

<http://ec2-16-16-70-158.eu-north-1.compute.amazonaws.com:5000/>

Not:Siteyi 1 kere kullanın fatura tuzlu olmasın.