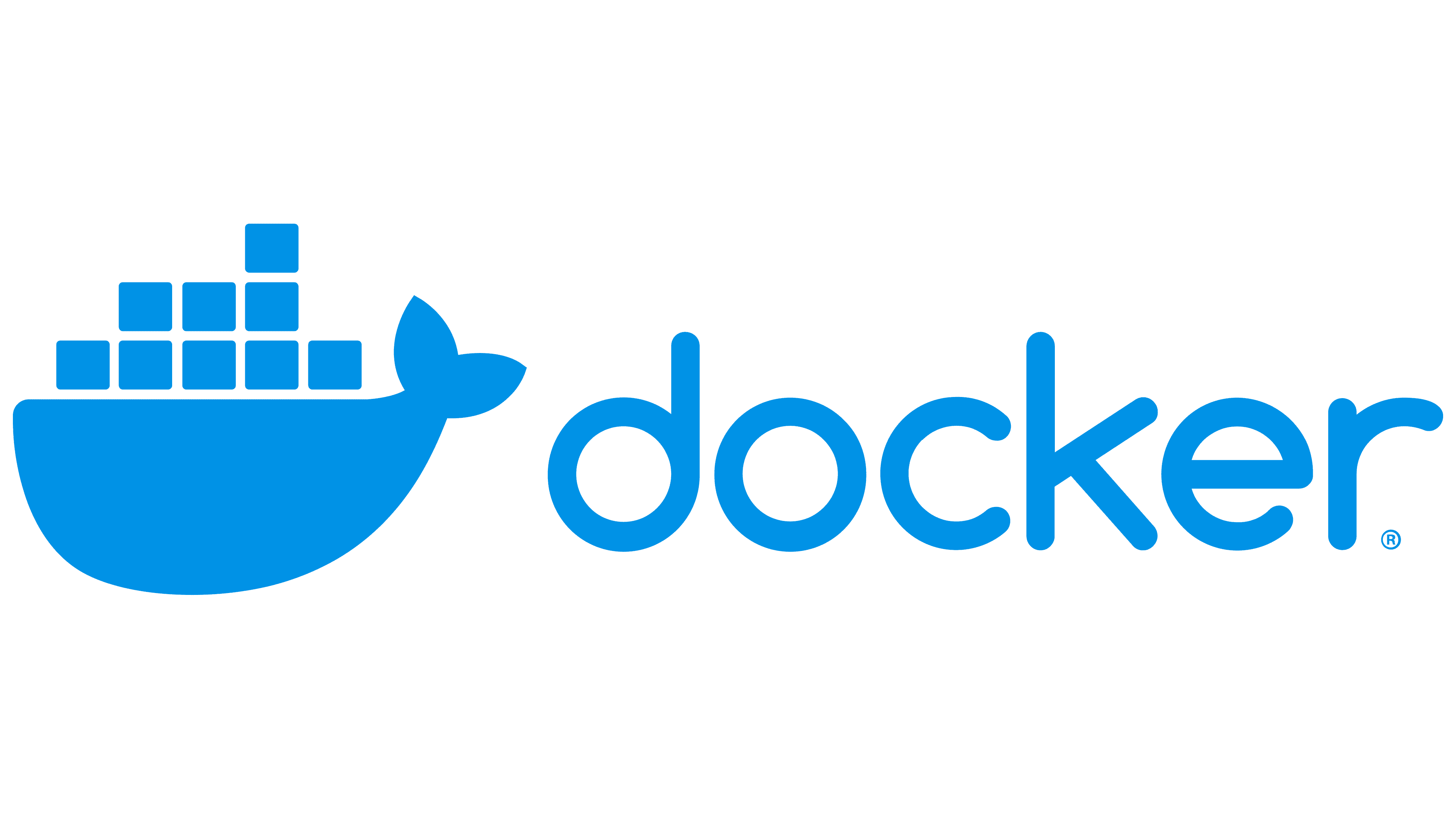
**Docker**

**Nedir?**

Docker, uygulamalarınızı hızla derlemenize, test etmenize ve dağıtmanıza imkan tanıyan bir yazılım platformudur. Docker, yazılımları kitaplıklar, sistem araçları, kod ve çalışma zamanı dahil olmak üzere yazılımların çalışması için gerekli her şeyi içeren [container](https://aws.amazon.com/tr/containers/) adlı standartlaştırılmış birimler halinde paketler. Docker'ı kullanarak her ortama hızla uygulama dağıtıp uygulamaları ölçeklendirebilir ve kodunuzun çalışacağından emin olabilirsiniz.

Docker, **Linux Containers** (LXC) teknolojisinin üzerine inşa edilmiştir. Konteynerler, uygulamaları ve onların bağımlılıklarını bağımsız ve izole bir şekilde çalıştırmak için kullanılır. Bir Docker konteyneri, uygulama kodunu, gerekli kütüphaneleri ve konfigürasyon dosyalarını içerir. Bu konteynerler, herhangi bir ortamda - geliştirme makineleri, test sunucuları veya üretim sunucuları - tutarlı bir şekilde çalışabilirler. 

Docker'ın temel yapı taşları, **Docker İmajları** olarak adlandırılır. Bir Docker imajı, bir uygulamanın çalışması için gerekli olan tüm bileşenleri içerir. İmajlar, konteynerleri başlatmak için kullanılır. Docker imajları katmanlı yapıdadır, yani her katman bir önceki katmana eklenir ve sadece değişiklikler saklanır. Bu yapı, depolama verimliliğini artırır ve ağ üzerinden hızlı dağıtım sağlar.

**Docker konteynerleri**, ana işletim sistemi üzerinde çalışırken, kendi bağımsız dosya sistemine ve işletim sistemine sahiptirler. Bu izolasyon, uygulamaların birbirleriyle ve ana işletim sistemiyle çakışmasını engeller.

A diagram of a container and a file

Description automatically generated**Docker Ekosistemi**

Docker ekosistemi, çeşitli bileşenlerden oluşur:

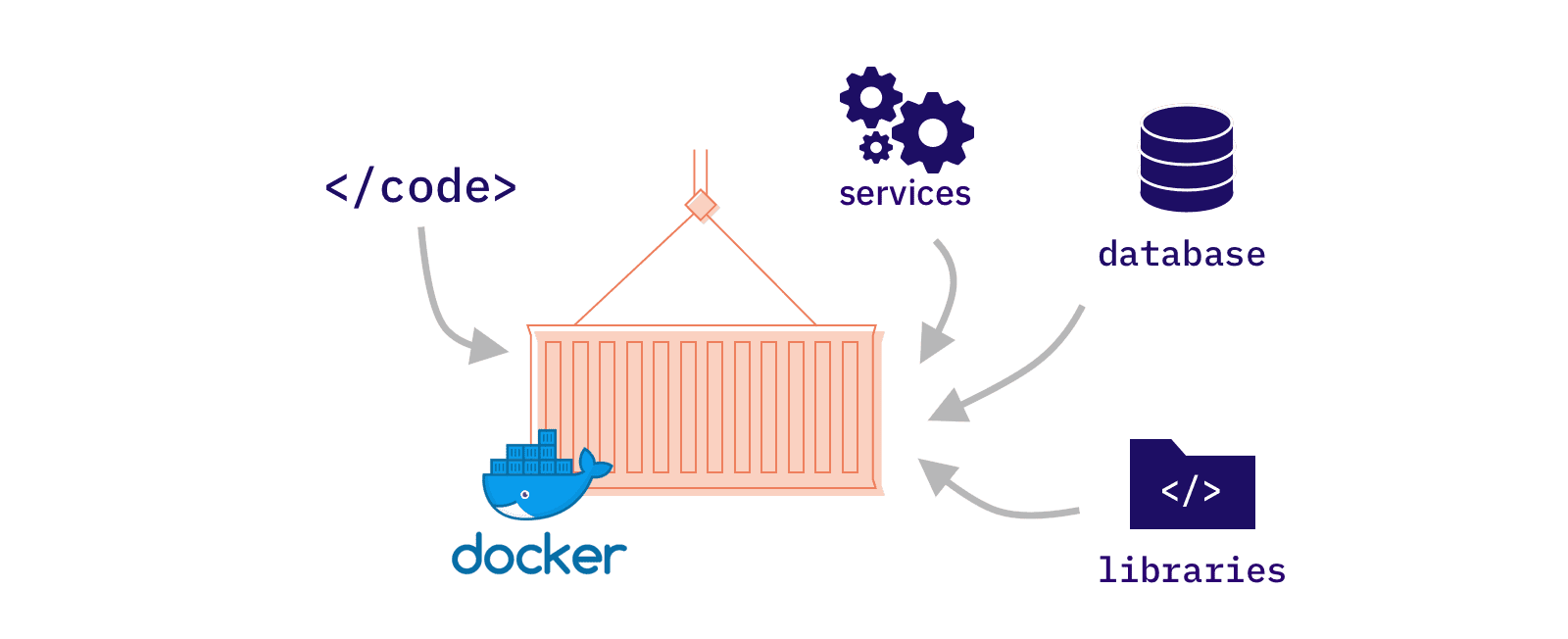
* **Docker Engine**: Docker konteynerlerini oluşturmak ve çalıştırmak için kullanılan çekirdek bileşendir.
* **Docker Hub**: Kullanıcıların Docker imajlarını paylaşabileceği ve keşfedebileceği çevrimiçi bir depodur.
* **Docker Compose**: Birden fazla konteyneri içeren uygulamaların tanımlanması ve yönetilmesi için kullanılan bir araçtır.
* **Docker Swarm**: Docker konteynerlerinin küme (cluster) ortamlarında orkestrasyonu için kullanılan bir araçtır.

**Çalışma Prensibi**

Docker, kodunuzu çalıştırmanın standart bir yolunu sağlayarak çalışır. Docker, container'lara yönelik bir işletim sistemidir. Bir [sanal makinenin](https://aws.amazon.com/tr/ec2/) sunucu donanımını sanallaştırmasına (doğrudan yönetme gereksinimini ortadan kaldırma) benzer şekilde container'lar da bir sunucunun işletim sistemini sanallaştırır. Docker her sunucuya yüklenir ve container'ları oluşturmak, başlatmak veya durdurmak için kullanabileceğiniz basit komutlar sağlar.

**Docker neden kullanılmalı?**

Docker'ı kullanmanız kaynak kullanımını geliştirerek daha hızlı kod göndermenize, uygulama işlemlerini standart hale getirmenize, kodu sorunsuz bir şekilde taşımanıza ve tasarruf etmenize imkan tanır. Docker ile her yerde güvenilir bir şekilde çalışabilen tek bir nesneniz olur. Docker'ın basit ve kolay anlaşılan söz dizimi, size tam denetim sağlar. Docker'ın geniş çaplı olarak benimsenmiş olması, Docker ile kullanıma hazır güçlü bir araç ekosistemi ve uygulamalar olduğu anlamına gelir.

A screenshot of a computer

Description automatically generated

Docker container'larını modern uygulamalar ve platformlar oluşturmak için temel bir yapı taşı olarak kullanabilirsiniz. Docker, dağıtılmış mikro hizmet mimarileri oluşturup çalıştırmanızı, kodunuzu standartlaştırılmış sürekli entegrasyon ve teslim işlem hatlarıyla dağıtmanızı, yüksek oranda ölçeklenebilir veri işleme sistemleri oluşturmanızı ve geliştiricileriniz için tam olarak yönetilen platformlar oluşturmanızı kolaylaştırır.

-Docker kullanıcıları, Docker kullanmayanlardan ortalama 7 kat daha sık uygulama sunuyor. Docker, gereken sıklıkta yalıtılmış hizmetler sunmanıza imkan tanır.

-Küçük container'lı uygulamalar dağıtım yapmayı, sorunları belirlemeyi ve düzeltme için geri almayı kolaylaştırır.

-Docker temelli uygulamalar yerel geliştirme makinelerinden AWS'deki üretim dağıtımlarına sorunsuz bir şekilde taşınabilir.

-Docker container'ları her sunucuda daha fazla kod çalıştırmanızı kolaylaştırarak kullanımınızı geliştirir ve tasarruf etmenizi sağlar.

**Kullanım Alanları ve Avantajları**

Docker, yazılım geliştirme ve dağıtım süreçlerini hızlandırır, taşınabilirliği artırır ve kaynakları daha verimli kullanmayı sağlar. Bu özellikler, Docker'ı günümüzün yazılım dünyasında önemli bir teknoloji haline getirmiştir ve birçok organizasyon tarafından benimsenmiştir. Docker'ın başlıca avantajları şunlardır:

* **Taşınabilirlik**: Docker konteynerleri, herhangi bir platformda aynı şekilde çalışabilir, bu da geliştiriciler ve operasyon ekipleri arasındaki uyumu artırır.
* **İzolasyon**: Konteynerler, uygulamaların ve bağımlılıklarının birbirlerinden ve ana sistemden izole edilmesini sağlar.
* **Verimlilik**: Docker, kaynak kullanımını optimize eder ve aynı sunucuda birden fazla uygulamanın verimli bir şekilde çalışmasını sağlar.
* **Hız**: Docker konteynerlerinin başlatılması ve durdurulması, geleneksel sanal makinelerden çok daha hızlıdır.

A group of people and containers

Description automatically generated

Docker, Linux veya Windows'ta yerel olarak çalışabilir. Linux kapsayıcıları (nginx sunucusu gibi) doğrudan bir Linux makinesinde çalıştırılabilir. Kapsayıcıyı Mac veya Windows'ta çalıştırıyorsanız, sanal bir makineye ihtiyacınız olacak veya Windows durumunda Windows Subsystem [for Linux'tan](https://docs.microsoft.com/en-us/windows/wsl/about) (WSL) yararlanabilirsiniz. Windows kapsayıcıları doğrudan Windows makinelerinde çalıştırılabilir.

[Neyse ki, Docker hizmetleri Docker Desktop veya](https://www.docker.com/get-started)[Rancher Desktop](https://rancherdesktop.io/) gibi başka bir araç kullanılarak Mac, Windows veya Linux makinelerinde kolayca çalıştırılabilir . Her ikisi de görüntüler ve kapsayıcılarla çalışmak için kullanılabilen kapsayıcı yönetim istemcileri (isterseniz bir "motor") sağlar.

 Bireysel olarak veya bir ekibin parçası olarak çalışsanız da, Docker geliştirme ortamlarının kurulumunu hızlandırabilir. Bu avantaj önemsiz görünse de, web geliştiricilerine önemli ölçüde yardımcı olur. Ek olarak, Docker uygulama çakışmalarını ortadan kaldırmaya yardımcı olabilir. En son çerçeve sürümüne yükseltmeye çalışırken uyumluluk sorunlarıyla karşılaşırsanız, izole edilmiş kapsayıcılar bir çözüm sağlayabilir. Dahası, Docker kodun ve tüm ortamının geliştirme, hazırlama ve üretim gibi farklı ortamlar arasında sorunsuz bir şekilde aktarılmasını sağlar. Sonuç olarak, bu avantajlar daha hızlı yazılım gönderimine katkıda bulunur.

**Windows Kurulum**

**Adım 1: Docker Desktop'ı İndirin**

* Docker Desktop'ı Docker’ın resmi web sitesinden indirin.( <https://docs.docker.com/desktop/install/windows-install/> )

**Adım 2: Docker Desktop'ı Yükleyin**

* İndirdiğiniz Docker Desktop **Installer.exe** dosyasını çalıştırın ve kurulum sihirbazını takip edin.
* Kurulum sırasında, “**Use the WSL 2 based engine**” seçeneğini işaretli bırakın.

**Adım 3: Docker'ı Başlatın**

* Docker Desktop’ı kurduktan sonra, masaüstünüzdeki Docker simgesine çift tıklayarak Docker’ı başlatın.
* Docker’ın yüklenip çalıştığını görmek için Windows terminali (Cmd, PowerShell veya Windows Terminal) açın ve şu komutu çalıştırın:

docker --version

**Kaynakça:**

[**https://aws.amazon.com/tr/docker/**](https://aws.amazon.com/tr/docker/)

[**https://en.wikipedia.org/wiki/Docker\_(software)**](https://en.wikipedia.org/wiki/Docker_(software))

[**https://tr.wikipedia.org/wiki/Docker**](https://tr.wikipedia.org/wiki/Docker)

[**https://blog.codewithdan.com/docker-for-developers-understanding-the-core-concepts/**](https://blog.codewithdan.com/docker-for-developers-understanding-the-core-concepts/)

[**https://medium.com/batech/docker-nedir-docker-kavramlar%C4%B1-avantajlar%C4%B1-901b37742ee0**](https://medium.com/batech/docker-nedir-docker-kavramlar%C4%B1-avantajlar%C4%B1-901b37742ee0)

[**https://chatgpt.com/**](https://chatgpt.com/)