Docker Nedir? Ne İşe Yarar?

04 Mart 2022 14 dakikada okuyabilirsiniz.

Docker nedir? sorusunun yanıtı şirketlerin ve yazılımcıların uygulama geliştirme yöntemlerini değiştiren önemli bir atılımı barındırıyor.

Docker'dan önce, farklı bir makinede bir proje yürütmek isteyenlerin, kitaplıklar ve veritabanları gibi tüm bileşenleri kurması gerekiyordu. Tek bir proje üzerinde çalışmak söz konusuysa bu durum bir sorun teşkil etmeyebilir ama aynı anda kurulamayan ve çalıştırılamayan bileşenler gerektiren paralel projeler için ciddi bir karmaşa da söz konusu olabilir. Örneğin, projelerinizden biri SQL gerektiriyorsa ve başka bir proje MariaDB gerektiriyorsa, diğerini başlatmak için birini kaldırmanız gerekecektir.

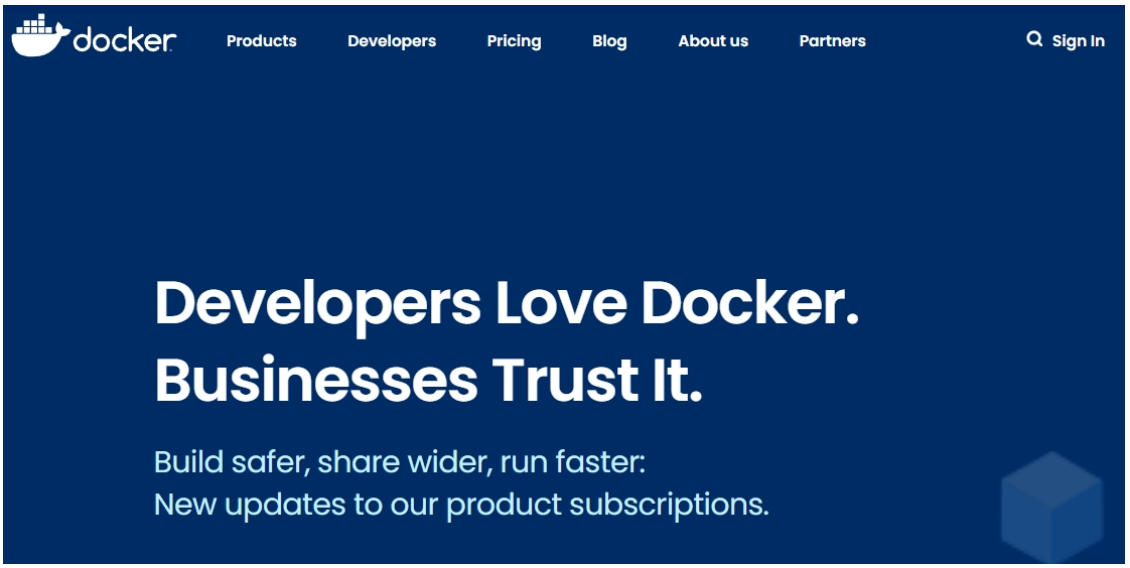
Her projenin/uygulamanın tüm bileşenleriyle ayrı bir

container'da izole edilebildiği, aynı makinede aynı anda birden fazla uygulamayı çalıştırabilmenizi sağlayan çözüm olarak karşımıza çıkan Docker'ın, son yıllarda birçok geliştirme ekibi ve projesi için neredeyse olmazsa olmaz hale gelmesinin bunun gibi pek çok nedeni var.

Docker Nedir?

Docker, uygulamaların oluşturma, yönetme, çalıştırma ve dağıtma sürecini basitleştirmeye yardımcı olan açık kaynaklı bir yazılım platformudur.

Şu ana kadar 7 milyondan fazla uygulama için kullanılan Docker, kurulu olduğu bilgisayarın işletim sistemini sanallaştırarak uygulama katmanında bir soyutlama oluşturur ve uygulamalarınızı altyapınızdan ayırmanıza olanak tanır.



Birçok şirketin uygulama oluşturma, çalıştırma ve yönetme sürecini basitleştirmek için kullandığı Docker, Linux ve Windows üzerinde çalışan açık kaynaklı bir yazılımdır.

2008 yılında Paris'te Solomon Hykes tarafından DotCloud olarak kurulan Docker, ilk kez Mart 2013'te PyCon'da tanıtılmış ve Microsoft, IBM ve Red Hat gibi dev isimlerin yanı sıra yenilikçi teknolojilere yatırım yapmak isteyenlerin de hızla

dikkatini çekmeyi başarmıştır.

Container'lara dayalı uygulamalar oluşturmaya yarayan ücretsiz, açık kaynaklı bir platform olan Docker'ın ticari versiyonunu satan şirket; Docker Inc'dir. Podman gibi alternatif platformlar mevcut olsa da Docker bu alanın lider oyuncusudur.

Docker'ın temel özelliklerine, kullanım amaçlarına ve temel bileşenlerine göz atmadan önce container kavramını açıklamakta yarar var.

Container

Container teknolojisi, daha çok bulut bilişimde kullanılan ve VM gereksinimlerine ihtiyaç duymadan farklı işletim sistemi platformlarında uygulamaları çalıştırmak için tasarlanmış bir tür sanallaştırma platformudur. Özellikle test aşamasında olmak üzere yazılım geliştirmenin çeşitli aşamalarını kolaylaştırmak için kullanılmaktadır.

Container'lar, uygulamaların kolaylıkla ve tek tip dağıtılmasına olanak tanıyan, uygulamaların yaşayabileceği yalıtılmış ortam yaratmanın bir yolu olarak tanımlanabilirler. Taşınabilirdirler, herhangi bir işletim sisteminde yeniden kullanılabilirler.

Container teknolojisi uzun süredir varlığını sürdürse de 2013'te Docker'ın piyasaya sürülmesi, container'ları uygulama ve yazılım geliştirme alanında endüstri standardı haline getirmiştir.

VM'lerin daha hafif sürümleri olarak düşünebileceğimiz container'lar ana bilgisayarın işletim sistemini paylaşıp kendi işletim sistemlerine ihtiyaç duymadıklarından VM'lerden farklıdır. Container'larla, temeldeki bilgisayarı bir VM gibi

sanallaştırmak yerine, yalnızca işletim sistemi sanallaştırılır.

Container'larla uygulamalar, tek bir amaca hizmet eden en küçük bileşen parçalarına bölünebilir ve bu bileşenler birbirinden bağımsız olarak geliştirilebilir ve dağıtılabilir. Örneğin, müşterilerin ürünler satın almasına olanak tanıyan bir uygulamanız olduğunu varsayalım. Arama çubuğunuz, alışveriş sepetiniz, satın alma butonunuz gibi bileşenlerin her biri kendi Container'ında bulunabilir.

Docker container'ları, tek bir işletim sistemi üzerinde birden çok uygulama çalıştırmak istediğiniz durumlar için uygundur.

Container'lar, fiziksel bir sunucunun ve ana bilgisayar işletim sisteminin (genellikle Linux veya Windows) üzerine oturur. Her container, ana bilgisayar işletim sistemi çekirdeğini ve genellikle ikili dosyaları ve kitaplıkları paylaşır. Paylaşılan bileşenler salt okunurdur.

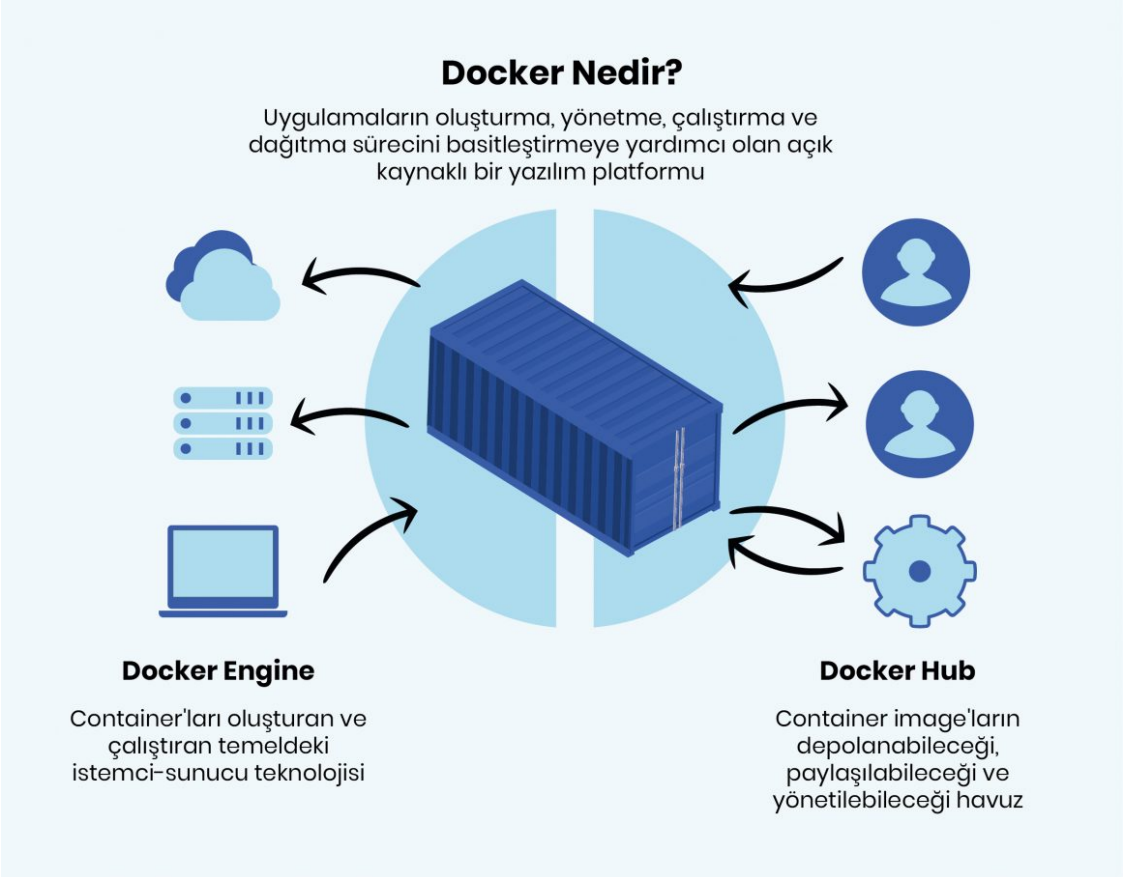
Kitaplıklar gibi işletim sistemi kaynaklarının paylaşılması, işletim sistemi kodunu yeniden oluşturma ihtiyacını önemli ölçüde azaltır; bir sunucu, tek bir işletim sistemi kurulumuyla birden çok iş yükünü çalıştırabilir. Container'lar bu nedenle son derece hafiftir; boyutları yalnızca megabayt

büyüklüğündedir ve başlamaları yalnızca birkaç saniye sürer. Bunun pratikteki anlamı, container'lı tek bir sunucuya bir VM ile yapabileceğinizden iki ila üç kat daha fazla uygulama koyabilmektir.

> Altyapı bakımını, güncellemesini ve desteğini basitleştirmeye yarayan Docker container'ları temel olarak, çalıştıkları ortamla pek ilgilenmezler, bu da farklı işletim sistemleri ve donanım platformlarıyla birçok farklı ortamda çalıştırılabilecekleri anlamına gelir.

Geliştiriciler genellikle dizüstü bilgisayarlarında kod yazarlar ve ardından bu kodu bir sunucuya aktarırlar. Bu ortamlar yazılım sürümleri, izinler, veritabanı erişimi gibi bileşenler arasında farklılıklara, hatalara yol açabilir. Container'lar çevre tutarsızlığı sorununu çözer.

Docker'ın Temel Özellikleri Nelerdir?



Uygulamaların farklı platformlarda geliştirilmesi ve test edilmesi gereken durumlar için Docker container'ları idealdir.

Docker, yeni nesil sanallaştırmanın bir tezahürüdür. Aynı işletim sisteminde çalışabilen veya uzaktan bağlanabilen Docker arka plan programı ile bir istemci-sunucu mimarisi kullanır.

- Docker, uygulamaların geliştirilmesini, test edilmesini ve devreye alınmasını basitleştirir.
- Docker, bileşenleri daha hızlı birleştirmemizi sağlar ve kodu gönderirken oluşabilecek hataları ortadan kaldırır. Örneğin, aynı sistemde aynı uygulamanın iki farklı sürümünü çalıştıran iki Docker container'ımız olabilir.
- Docker, kodu mümkün olan en kısa sürede test etmemize yardımcı olur.
- Kullanımı basittir, platformun sağladığı tüm işlemleri yapabileceğiniz bir komut satırı arabirimi (CLI) ile birlikte gelir.
- Docker, bir uygulamanın kodunu ve tüm bağımlılıklarını geliştiricinin dizüstü bilgisayarından sunucuya taşımayı

kolaylaştırır.
- Çoğunlukla Docker container'ları bir dakika içinde başlatılabilir.
- Docker konteynerleri her yerde çalışır. Containerları masaüstlerine, fiziksel sunuculara, sanal makinelere, veri merkezlerine, genel ve özel bulutlara dağıtabiliriz ve her yerde aynı kapsayıcıları çalıştırabiliriz.

Docker Ne İçin Kullanılır?

2013 yılında piyasaya sürülen Docker, bulutta yerel ve hibrit çözümler kullanan şirketlerin artmasıyla giderek daha popüler hale gelmiştir. Docker, aslen yazılım geliştirme projelerine hizmet etmek ve uygulamaların tutarlı teslimatını artırmak için kullanılmaktadır.

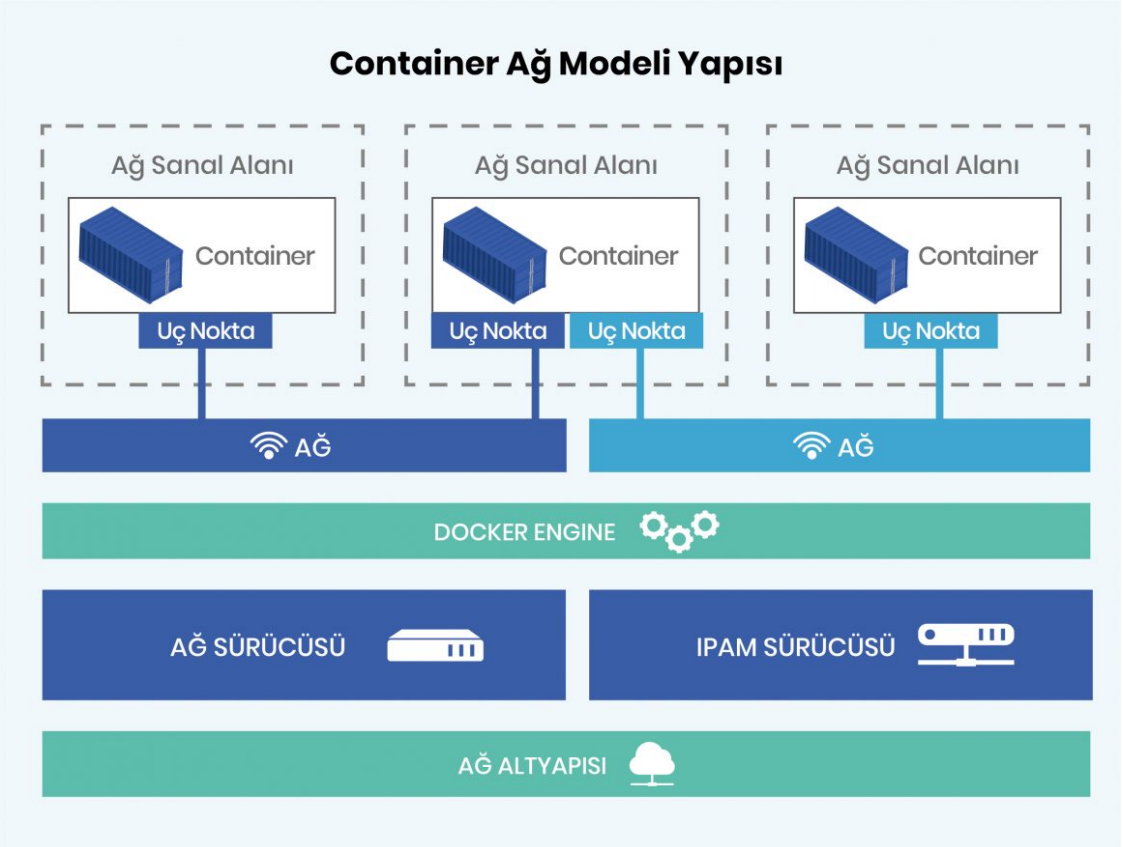
DevOps'tan, yazılıma, BT hizmetlerinden finans ve sağlık hizmetlerine yayılan Docker'ın en önemli kullanım alanlarını şu şekilde listeleyebiliriz:

- Docker, , yazılım geliştirme yaşam döngüsünü (SDLC) basitleştirme ve standartlaştırma amacıyla kullanılır.
- Geliştiricilerin, otomatik ve manuel testleri sorunsuz bir şekilde yürütmeleri için yerel yazılı kodu meslektaşları ve test uzmanlarıyla kolaylıkla paylaşmasına olanak tanır. Docker, özellikle ekibe katılan yeni geliştiricilerin ilk günden itibaren verimli olabilmesi için kurulum ve yüklemeyi otomatik hale getirmesi açısından oldukça avantajlıdır. Her şeyi manuel olarak yapmak yerine, geliştirme ortamını tek bir komutla çalıştırabilirler. Bu elbette ki çok zaman kazandırır.
- Geliştirme ve test ortamları arasında hataların düzeltilmesini ve birleştirilebilirliği kolaylaştırır. Yazılımınız ne kadar karmaşık hale gelirse, çalışması için gerekli olan tüm parçalarını takip etmek o kadar zor olur. Docker olmadan, proje kurulumundaki tüm değişikliklerin diğer

geliştiricilere iletilmesi ve belgelenmesi gerekir. Aksi takdirde, kod sürümleri çalışmayı durdurabilir ve nedeni anlaşılmayabilir. Docker ile, yazılımın gerekli tüm bileşenleri Docker yapılandırma dosyalarında (Dockerfile ve docker-compose.yml gibi) belirtilir.

- Bulutta, yerel veri merkezlerinde ve hibrit platformlarda; yerel dizüstü bilgisayarlardan, birincil sunuculardan, klasik sanal sunuculardan dinamik ve yüksek taşınabilirlik sağlayan iş yükleri oluşturmaya yarar. Çalıştırdığınız her ortam için ek yapılandırma, kurulum ve ayarlamalar gerekmez. Uygulamanız her ortamda tutarlı ve öngörülebilir bir şekilde çalışabilir.
- İşletmenin ihtiyaçlarına göre iş yüklerini ve uygulamaları gerçek zamanlı olarak büyütüp küçültmeye ve yüksek yoğunluklu ortamlarda birden fazla iş yükünü çalıştırmaya olanak tanır. Küçük web siteleri ve uygulamalar, karmaşık barındırma altyapısı gerektirmez ancak bir işletme büyüyüp geliştiğinde, sunucu gereksinimleri de değişir. Hızlı tempolu iş ortamlarında, hem web sitenizin çökmemesini sağlamak hem de altyapı maliyetlerinin uygun olmasını sağlamak için web altyapısının hızla uyum sağlayacak kadar esnek olması gerekir. Docker container'ları hemen hemen her sunucu ortamında başlatılabilir. Böylece ihtiyaçlarınız değiştiğinde yazılımınız başka bir yere kolaylıkla yerleştirilebilir.
- Farklı teknolojileri denemek istiyorsanız oldukça kullanışlıdır. Yeni bir veritabanı veya programlama dilini denemek için Docker topluluğu tarafından oluşturulmuş hazır bir Docker şablonu bulma İhtimaliniz oldukça yüksektir. Docker Hub, kullanmak istediğiniz hemen hemen tüm teknolojiler için büyük bir Docker image deposudur.

Docker Temel Bileşenleri



Container oluşturmak ve başlatmak için gereken araçları daha önce mümkün olandan daha akıcı ve basitleştirilmiş bir şekilde paketlemenin yeni bir yolunu bulan Docker; Dockerfile, Docker Compose, Docker Images, Docker Daemon, Docker Hub, Docker Engine gibi bileşenlere sahiptir.

Dockerfile

Her Docker container bir Dockerfile ile başlar. Bu; metin dosyası, işletim sistemi, diller, çevresel değişkenler, dosya konumları, ağ bağlantı noktaları ve çalıştırması gereken diğer bileşenler dahil bir Docker image oluşturmak için bir dizi talimat sağlayan ögedir.

Dockerfile içindeki her talimat, görüntüde yeni bir katman oluşturur. Dockerfile'ı değiştirmemiz gerektiğinde değişen katmanlar yeniden oluşturulur. Bu nedenle diğer sanallaştırma teknolojileriyle karşılaştırıldığında görüntüler çok hafif, küçük ve hızlıdır.

Docker Image

Docker image, Docker container oluşturmak için talimatlar içeren salt okunur bir şablondur. Docker yaşam döngüsünün en çok inşa edilen parçasıdır. Çoğunlukla, bir image, bazı ek özelleştirmelerle birlikte başka bir image'ı temel alır.

Kendi görüntülerimizi oluşturabilir veya yalnızca başkaları tarafından oluşturulan ve kayıt dizininde yayınlananları kullanabiliriz.

Docker run utility, bir container'ı başlatan komuttur. Her kap, bir image'ın bir örneğidir ve aynı image'ın birden çok örneği aynı anda çalıştırılabilir.

Docker kayıt defteri, Docker image'larını tutar. Özel kayıt defterimizi çalıştırabiliriz. Docker pull ve docker run komutlarını çalıştırdığımızda, gerekli image'lar yapılandırılmış kayıt dizinimizden kaldırılır. Docker push komutunu kullanarak image, yapılandırılmış kayıt dizinimize yüklenebilir.

Docker Hub

Docker Hub, container image'larının depolanabileceği, paylaşılabileceği ve yönetilebileceği bir havuzdur. Bunu Docker'ın kendi GitHub sürümü olarak düşünebilirsiniz.

Docker Engine

Docker Engine, Docker'ın çekirdeğidir. Container'ları oluşturan ve çalıştıran temeldeki istemci-sunucu teknolojisidir. Container'ları yönetmek için dockerd adlı bir arka plan programı, programların Docker arka plan programı ile iletişim kurmasına olanak tanıyan API'ler ve bir komut satırı arabirimi içerir.

Container inşa etmenin ve çalıştırmanın ağır iş yükü, Docker motorunun omuzlarındadır ve uygulamalarımızı yönettiğimiz gibi altyapımızı da yönetebiliriz.

Docker Compose

Docker Compose, çok container'lı Docker uygulamalarını tanımlamak ve çalıştırmak için YAML dosyalarını kullanan bir komut satırı aracıdır. Yapılandırmanızdan tüm hizmetleri oluşturmanıza, başlatmanıza, durdurmanıza ve yeniden oluşturmanıza ve çalışan tüm hizmetlerin durumunu ve günlük çıktısını görüntülemenize olanak tanır.

Docker Desktop

Bileşen parçalarının tümü, Docker'ın Masaüstü uygulamasına sarılarak, derlemek ve paylaşmak için kullanıcı dostu bir yol sağlar.

Docker Daemon

Docker Daemon, ana bilgisayarda çalışan ve API çağrılarını dinleyen (Docker istemcisi aracılığıyla) arka plan hizmetidir, image'ları yönetir, container'ları oluşturur, çalıştırır ve dağıtır. Deamon, istemcinin konuştuğu işletim sisteminde çalışan ve aracı rolünü oynayan süreçtir.

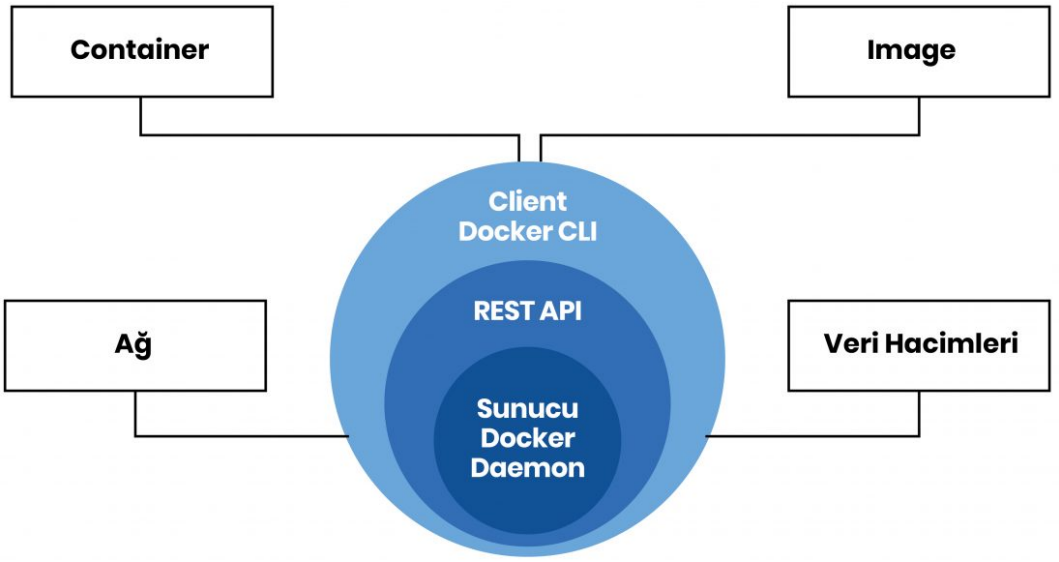
Docker Container

Bir image'ın örneğidir. Docker CLI kullanarak bir container oluşturabilir, çalıştırabilir, durdurabilir veya silebiliriz. Bir container'ı birden fazla ağa bağlayabilir, hatta mevcut durumuna göre yeni bir görüntü oluşturabiliriz. Yularıda da bahsettiğimiz gibi varsayılan olarak, bir container diğerlerinden ve sistem cihazından yalıtılmıştır.

Docker Client

Kullanıcının arka plan programı ile etkileşime girmesine izin veren komut satırı aracıdır. Client, bir ürünü kullanmanıza izin veren bir tür ağ geçidi veya arabirimdir. Bir kullanıcı ile ürünü çalıştıran bazı karmaşık uygulamalar arasında bulunan katmandır.

"Docker run" komutlarını kullandığımızda, istemci bu komutları dockerd'a gönderir. Docker tarafından kullanılan komut, Docker API'sine bağlıdır. Docker'da istemci birden fazla daemon işlemiyle etkileşime girebilir.



Container çalıştırmak gibi belirli bir görevi yürütmek üzere docker daemon ile iletişim kurmak, image oluşturmak; docker client istemcisi aracılığıyla yapılır. Docker CLI ile birlikte Docker daemon, daemon tarafından sunulan hizmetlere programlı olarak erişmek için kullanılabilecek bir REST API'sini de sunar.

Docker'ın Avantajları

Bugün, birçok şirketin uygulama oluşturma, çalıştırma ve yönetme sürecini basitleştirmek için kullandığı Docker'ın en

önemli avantajı, kodun daha hızlı gönderilmesini, test edilmesini ve dağıtılmasını sağlamasıdır.

Docker'ın bir diğer önemli avantajı ise tutarlılıktır. Geliştiricilerin bir uygulamayı tasarım ve geliştirmeden üretim ve bakıma kadar tutarlı bir ortamda çalıştırmasını sağlayan Docker sayesinde, uygulama farklı ortamlarda aynı şekilde davranarak üretim sorunlarını ortadan kaldırır. Öngörülebilir ortamlar mevcut olduğunda, geliştiriciler hata ayıklama ve yapılandırma/uyumluluk sorunlarını çözmek yerine uygulamaya kaliteli özellikler eklemeye daha fazla zaman ayırabilir.

Hız ve çeviklik,
Docker'ın en sevilen özelliklerindendir.

Her işlem için anında container oluşturmanıza ve bunları saniyeler içinde dağıtmanıza olanak tanır. İşletim sistemini başlatmanız gerekmediğinden, işlem yıldırım hızında yapılır. Anında kolaylıkla bir container oluşturabilir, yok edebilir, durdurabilir veya başlatabilirsiniz. YAML kullanarak bir yapılandırma dosyası oluşturarak dağıtımı otomatikleştirebilir ve altyapıyı kolaylıkla ölçeklendirebilirsiniz.

Docker, bir container image oluşturabileceğiniz ve bağımlı olmayan görevleri paralel olarak çalıştırırken bunu işlem hattı boyunca kullanabileceğiniz için CI/CD işlem hattınızın hızını ve verimliliğini artırır. Yeni bir değişikliğin ortamı bozması durumunda anında önceki sürüme geri dönmenizi sağlar.

Son zamanlarda popülerlik kazanan çoklu bulut ortamında, her bulut farklı yapılandırmalar, politikalar ve süreçlere sahiptir ve farklı altyapı yönetim araçları kullanılarak yönetilir. Ancak Docker container'ları herhangi bir ortamda taşınabilir (Container yok edildikten sonra kapsayıcı içindeki verilerin kalıcı olarak yok edildiği unutulmamalıdır. Gerekli verilerin yedeklediğinden emin olunması önerilir).

Docker ortamları oldukça güvenlidir. Docker container'larında çalışan uygulamalar birbirinden izole edilmiş, yalıtılmıştır. Her container kendi kaynaklarına sahiptir, diğer kapsayıcıların kaynaklarıyla etkileşime girmez. Kendilerine tahsis edilen kaynakları kullanır. Böylelikle trafik akışı üzerinde daha fazla kontrol sahibi olur, uygulamanın kullanım ömrü sona erdiğinde, container'ını silerek temiz bir uygulama kaldırma işlemi gerçekleştirebilirsiniz.

Docker altyapı maliyetlerini önemli ölçüde azaltmanıza da olanak tanır. Sanal makineler ve benzer teknolojilerle karşılaştırıldığında Docker, uygulamaları minimum maliyetle çalıştırmanızı sağlar. Küçük ekipler ve azaltılmış altyapı maliyetleriyle, işletim maliyetlerinden önemli ölçüde tasarruf edebilir ve yatırım getirinizi artırabilirsiniz.

Docker ve Sanal Makine Arasındaki Farklar Nedir?

Docker kapsayıcıları ve sanal makineler arasındaki farklar işletim sistemi desteği, güvenlik, taşınabilirlik ve performans şeklinde sıralanabilir.

Sanal makineleri ve Docker container'larını karşılaştırmak aslında doğru, adil bir yaklaşım değildir çünkü ikisi de farklı amaçlar için kullanılır. Docker'ın hafif mimarisi, daha az kaynak kullanma özelliği, onun verimlilik açısından daha iyi bir seçim olduğunu düşündürse de container'ların kaynak kullanımı ve sanal makinelere göre çok daha hızlı devreye alınabilmesi; içindeki yüke veya trafiğe göre değişiklik gösterir.

Docker Container

- İşletim sistemi düzeyinde sanallaştırma, daha düşük izolasyon
- Container'lar tek bir işletim sistemini paylaşır.
- Saniyeler içinde başlatılır.
- Daha hafiftirler (KBs/MBs).
- Taşımaktansa ortadan kaldırılır ya da yenisi oluşturulur.
- Daha az kaynak kullanır.
- Bulutta yerel uygulama geliştirme gerektiğinde idealdir.

VM

- Donanım düzeyinde sanallaştırma,, yüksek izolasyon
- Her VM kendi işletim sistemine sahiptir.
- Dakikalar içinde başlatılır.
- Birkaç GB gerektirirler.
- Yeni bir host'a kolaylıkla taşınır.
- Daha fazla kaynak kullanır.
- Sunucularda birden çok uygulama çalıştırmanız veya çeşitli işletim sistemleri kullanmanız gerektiğinde idealdir.

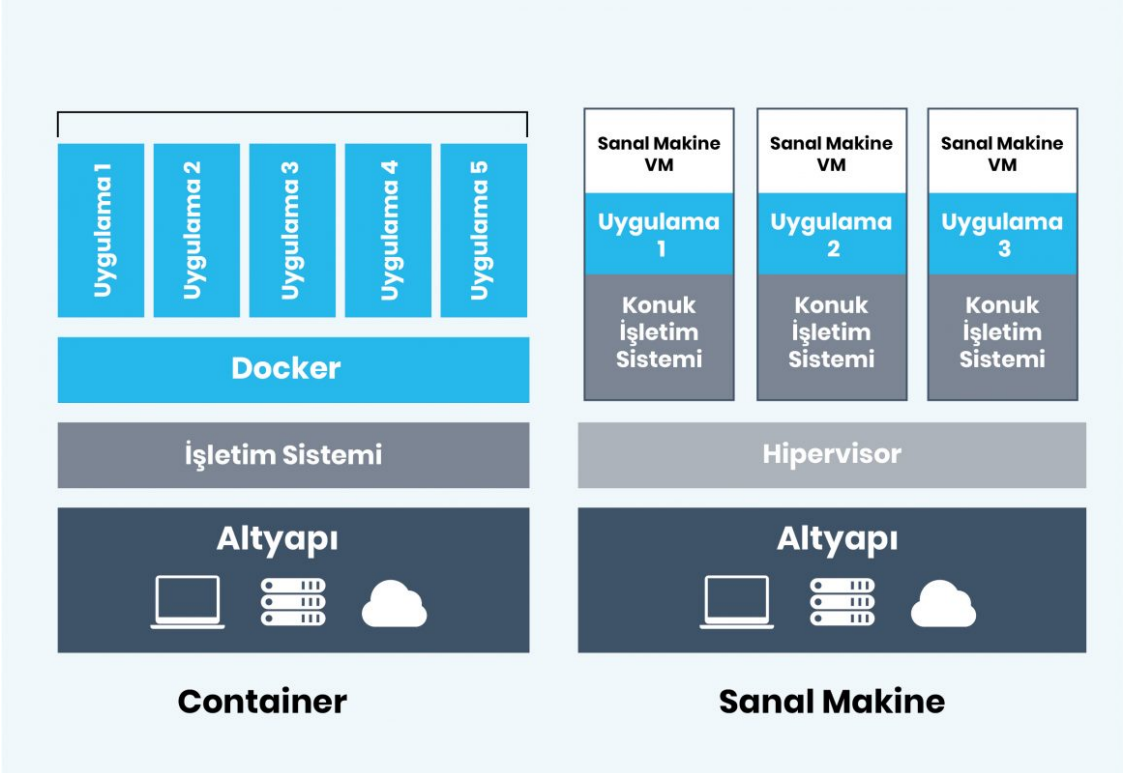
Yalıtılmış bir ortam yaratmanın bir başka yolu olan sanal makineler (VM), ana makinenin temel donanımının sanallaştırılmasıyla oluşturulur. VM'ler ve ana bilgisayar arasında ana bilgisayarın donanımını sanallaştıran ve aracı olarak hareket eden yazılım yani hipervisor katmanı bulunur.

Her sanal makine, ana işletim sisteminin üzerinde konuk işletim sistemine sahiptir. Container'lar ise izolasyon için farklı bir yaklaşımı benimser: VM'ler gibi bir ana makine üzerinde yaşamalarına ve onun kaynaklarını kullanmalarına rağmen kendi işletim sistemlerine sahip olmaları gerekmez, bir hipervisor (VMWare veya VirtualBox gibi) çalıştırmayı gerektirmez bu da onların VM'lerden çok daha hafif olmalarını sağlar.

Docker container'ları, tek bir işletim sistemi üzerinde birden çok uygulama çalıştırmak istediğiniz durumlar için uygundur. Ancak farklı işletim sistemleriyle çalışması gereken uygulamalarınız veya sunucularınız varsa, sanal makineler gereklidir.

Sanal makinelerde, ana bilgisayar çekirdeğinde güçlü bir izolasyon söz konusudur. Bu nedenle, VM'ler container'lara kıyasla daha güvenlidirler.

Docker kaynakları paylaşıldığı için, bir korsanın tek bir container'a erişerek kümedeki tüm kapsayıcılardan yararlanması söz konusu olabilir.



Sanal makinelerden farklı olarak, kaynakları kalıcı olarak container'lara tahsis etmeye gerek yoktur. Container'ları büyütmek ve çoğaltmak, sanal makinelere kıyasla kolaydır.

Docker container'ları ayrı işletim sistemlerine sahip olmadıkları için kolaylıkla taşınabilirler. Bir sanal makineyi taşımak, container'lara kıyasla daha zordur. Uygulamaların farklı platformlarda geliştirilmesi ve test edilmesi gereken durumlar için Docker container'ları idealdir.

VM'ler, sınırlı sayıda donanım ve yazılımdan yararlanma imkânını artırarak şirketlerin altyapı kaynaklarından en iyi şekilde yararlanmalarına yardımcı olur. Container'lar ise mikro hizmetleri ve DevOps uygulamalarını etkinleştirerek şirketlerin geliştirme kaynaklarından en iyi şekilde yararlanmasına yardımcı olur.

Yeni bir uygulama geliştiriyorsanız ve ölçeklenebilirlik ve taşınabilirlik için mikro hizmet mimarisi kullanmak istiyorsanız ya da en büyük önceliğiniz minimum sayıda sunucuda çalışan çok sayıda uygulama ve maksimum taşınabilirlikse

için container'larla yola çıkmanız önerilir. Container'ları sanal bir makinede çalıştırabilirsiniz.

Sanal Sunucu

Adı sanal performansı gerçek ve uygun bir hizmet için

Özet

VDS Server

İnovasyon ile hazırlanmış son teknoloji hizmet

Docker, container'ları kullanarak uygulamalar oluşturmayı, dağıtmayı ve yürütmeyi kolaylaştıran, Linux ve Windows üzerinde çalışan açık kaynaklı bir platformdur.

VPS Server

Eski hizmet, yeni teknolojiler ile güçlendirilmiş

Uygulamalarınızı altyapınızdan ayırmanıza olanak tanıyan

ma

yondan

eğiştiren bir

BUNLAR DA İLGİNİZİ ÇEKEBİLİR

İÇİNDEKİLER

Kubernetes Nedir?

WordPress Hosting Ne İşe Yarar?

Screaming Frog Nedir? Ne İşe Yarar?

En İyi Ücretsiz SEO Araçları

Veri Madenciliği (Data Mining) Nedir?

E-Ticaret Sitenizin Trafiğini Nasıl Artırabilirsiniz?