**Docker Nedir?**

Docker en net tanımlamayla open source bir ‘container’ teknolojisidir.

Docker, aynı işletim sistemi üzerinde, yüzlerce hatta binlerce birbirinden izole ve bağımsız containerlar sayesinde sanallaştırma sağlayan bir teknolojidir.

Web uygulamalarımızın kolayca kurulumunu, testini, çalışmasını ve deploymentını sağlar. Bunun yanında sunucu maliyetlerini önemli ölçüde azaltır.

**Docker Daemon:**

Hypervisor’ün dockerdaki karşılığıdır. Bütün CPU ve RAM vb gibi işletim sistemine ait işlerin yapıldığı bölümdür.

**Nedir bu Virtualization(Sanallaştırma)?**

Sanallaştırma, gerçek donanımdan soyutlanmış bir layer(katman)’da bir bilgisayar sisteminin sanal bir kopyasını çalıştırma işlemidir.

Kısaca, bir bilgisayar sisteminde aynı anda birden fazla işletim sisteminin çalıştırılması anlamına gelir.Sanallaştırmayı kullanmamızın birçok nedeni vardır.

Bu nedenlerden en yaygın olanı ise, bilgisayar değiştirmek veya uygulamalarımızı farklı platformlarda test etmek ve ve vee linux ile windows işletim sistemlerinin ikisine’de ihtiyacın doğması.

Kendimden örnek vermek gerekirse; kişisel bilgisayarıma linux(herhangi birisi) kurulumu yapınca tam istediğim performansta çalışmaması durumunda sanal olarak linux kullanmam gösterilebilir.

**Hypervisor Nedir?**

Fiziksel makine ile sanal makine arasında görev yapan bir yazılım katmanıdır.

Hypervisor bir sanallaştırma teknolojisi bileşenidir. Günümüzde hypervisor yazılımı kullanılarak birçok sanallaştırma çözümü geliştirildi.

Hypervisor ile bilgisayarınız üzerinde sanal makineler çalıştırmanıza izin veren programlar geliştirildi.

2 Tipi vardır donanım ve işletim sistemi üzerindedir.

Type 1 Type 2

**Container Nedir:**

Docker Daemon tarafından Linux çekirdeği içerisinde birbirinden izole olarak çalıştırılan process’lerin her birine verilen isimdir.

Virtual Machine (Sanal Makina) analojisinde Docker’ı Hypervisor’e benzetirsek fiziksel sunucu üzerinde -

-halihazırda koşturulmakta olan her bir işletim sisteminin (sanal sunucunun) Docker’daki karşılığı Container’dır.

**Image Nedir:**

Containerlar layer halindeki Image’lardan oluşur. Docker Image ise containerlara kurulacak ve run edilecek olan uygulamaların veya OS’lerin image dosyalarıdır.

Örnek verecek olursak mysql, mongodb, redis, ubuntu, mariadb.. Yüzlercesi mevcut

Link:https://hub.docker.com/search?q=&type=image

**Docker Registry Nedir:**

Tıpkı GitHub gibi geliştiriciler açık kaynaklı olarak docker imagelerini yükleyerek ve DockerHub’ta paylaşarak imagelerin bizim de indirip kullanmamıza olanak sağlıyorlar.

Kısaca imagelar Docker Registrylerde tutuluyor. Örneğin siz postgres image’ını kullanmak istiyorsunuz; postgres image linkinden

/// docker pull postgres ///

komutu ile indirip artık bu image ile container oluşturabiliyorsunuz. Aslında github’a çok benziyor.

Siz de kendiniz imageleri oluşturup yükleyip başkalarının da sizin yarattığınız imageları kullanmalarını sağlayabiliyorsunuz.

İsterseniz Private olarak ta registerynizi tutabilirsiniz. Docker bu hizmeti de size sunuyor.

**Docker CLI Nedir:**

Kullanıcının Docker Daemon ile konuşmasını sağlayan, docker komutlarının çalıştırıldığı CLI ekranıdır.

**Docker Compose Nedir:**

Compose, birden fazla containere sahip docker uygulamalarını tanımlamak ve çalıştırmak için kullanılır. Compose ile uygulamanızın servislerini configure etmek için bir YAML dosyası kullanılır.

Ardından, tek bir komutla configure ettiğiniz ayarlar ile tüm servislerinizi oluşturup ve başlatabilirsiniz.

Compose tüm ortamlarda çalışır: production, staging, development, testing ve diğer CI iş akışları da dahil olmak üzere.

Compose kullanmak temel olarak üç adımlı bir işlemdir:

1- Uygulamanızın ortamını Dockerfile dosyası ile herhangi bir yerde yeniden üretilebilecek şekilde tanımlayın .

2- docker-compose.yml uygulamalarınızı izole ortamda beraber çalışacak şekilde yaml dosyası içinde set edebilirsiniz.

3- docker-compose up / komutuyla birlikte tüm uygulamanızı başlatır ve çalıştırabilirsiniz.

**Docker’ın Avantajları Nelerdir?**

1- Docker saniyeler içerisinde başlar

2- Daha hızlı deployment süreci

3- Daha Kolay Yönetim ve Ölçeklendirme

Kaynak : https://medium.com/batech/docker-nedir-docker-kavramlar%C4%B1-avantajlar%C4%B1-901b37742ee0