Docker Compose Konfigurasyonu

CLI(Terminal kkomutları)

Docker Compose neden kullanılır

Konternardan oluşan servisler var.Konteynar arası konfigurasyon yapılır.

Konteynarın environment,port,network ve volume ayarlarını çok kolay bir şekilde yapabililrsiniz.

Kolay bir şekilde ortamın kurulmasını sağlar.

Docker Compose u ikiye ayırabiliriz

1.Yaml dosyası:konfigurasyon içindiri

Konteynarları servis olarak belirtebilirsiniz.

Portları,networkleri belitebilirsiniz

Volumeleri belirtebilirsiniz

2.CLI docker-compose komutu :Yönetmek ve çalıştırmak için

Local,dev ve test ortamlarını nasıl kontrol edebiliriz ve bunları yönetebiliriz

Docker compose un konfigurasyon dosyasına bakalım

version: Zorunlu alandır.1 den başlayıp 3.1 e kadar gelir.Ben minumum 2.0 kullanmanızı tavsiye ederim

services :konteynar bilgilerini giriyoruz

servisname : Konteynardaki –name alanına karşılık geliyor

image :Konternarın hangi imageyi kullanacağı.Mongodb için mongo,nginx nginx gibi

command : Bu alan zorunlu değil isteğe bağlı.konteynar çalıştığında bash komutları çalıştırmak istersek bu komutu kullanacağız.

volumes : Volume tanımı yapacak isek bu yere yapıyoruz. Bu alan zorunlu değil isteğe bağlı.Konternardaki -v komutuna karlışık gelir.

environment : Environment variableleri ayarlıyoruz.Konteynardaki -e komutuna karşılık geliyor.Database ayağa kaldırırken username ve passwordünü set edebiliyoruz. Bu alan zorunlu değil isteğe bağlı

servicename2:Bir başka konteynar için kullanılıyor.Mesela node js

servicename3 : Bir başka konteynar için kullanılıyor.Mesela proxy server nginx gibi

volumes : Volume bilgileri gelecek.Optionaldır.docker volume create ile yarattığınız bilgiler.Konteynarda tanımladığınız volumeyi aşağıda oluşturmanız gerekecek.

Bu alan zorunlu değil isteğe bağlı

networks :network bilgileri gelecek.Optionaldır.docker network create ile oluşturduğunuz networkler.Bu alan zorunlu değil isteğe bağlı

Services tagı ile birden fazla servisi tek bir servişmiş gibi çalıştırabiliriz

*docker container run -p 80:80 -v ${PWD}:/site –name test suayb/test*

komutunun docker-compose dosyasını yazalım

docker-compose.yml

**version**: **'3.1'***#docker run -p 80:4000 -v ${PWD}:/test suayb/test***services**:  
 **test**:  
 **image**: suayb/test  
 **volumes**:  
 - .:/test  
 **ports**:  
 - **'80:80'**

test :--name alanına karşılık geliyor.yani konteynar adına

image :komutun sonunda yazdığımız imaj adıyla aynı

volumes alanına volumeleri geliyoruz.

. : bulunduğumuz dizin demek. *${PWD}* ile aynı

Ports :portları giriyoruz.-p komutu ile aynı işlevi yapıyor

Daha karmaşık bir örnek inceleyelim

**version**: **'3.1'  
services**:  
 **wordpress**:  
 **image**: wordpress  
 **ports**:  
 - 8080:80  
 **environment**:  
 **WORDPRESS\_DB\_HOST**: mysql  
 **WORDPRESS\_DB\_NAME**: wordpress  
 **WORDPRESS\_DB\_USER**: wp  
 **WORDPRESS\_DB\_PASSWORD**: pass  
 **volumes**:  
 - ./wordpress-data:/var/www/html  
  
 **mysql**:  
 **image**: mysql  
 **environment**:  
 **MYSQL\_ROOT\_PASSWORD**: pass  
 **MYSQL\_DATABASE**: wordpress  
 **MYSQL\_USER**: wp  
 **MYSQL\_PASSWORD**: pass  
 **volumes**:  
 - mysql-db:/var/lib/mysql  
  
**volumes**:  
 **mysql-db**:

Bu örnekte 2 tane konteynarımız var.wordpress ve mysql olmak üzere

Mysql konteynarı

image : Image olarak mysqli kullanıyor.

environment : environment değişkenleri olarak MYSQL\_USER da kullanıcı adını ,ROOT\_PASSSWORD de de pasword alanını giriyoruz.MYSQL\_DATABASE de yaratılacak databsei olan wordpressi girdik.

Volumes olarakda mysql\_db yi ayarlamış olduk.

Wordpress konteynarı içinde

image : Image olarak wordpressi kullanıyor.

ports : Localde 8080 de konteynarda 80 de çalışıyor

environment : environment değişkenleri olarak db host environmentandan mysql konteynarına bağlanıyor.DB\_NAME de database adı,DB\_USER da kullanıcı adı ,DB\_PASSSWORD de de pasword alanını giriyoruz.

Volumes olarakda wordpress-data yı ayarlamış olduk.

Son olarak

Volumes da mysql\_db yi tanımlayarak mysql\_db volumenin oluşturulmasını sağladık.

Docker compose ile daha fazla bilgi almak için <https://docs.docker.com/compose/gettingstarted/> sayfasını ziyaret edebilirsiniz.