Docker compose localinizde image oluşturdunuz.Ve Docker composede build alıp kullanmak istiyordunuz.

nginx.conf

server{  
 listen 80;  
  
 location / {  
 proxy\_pass http://web;  
 proxy\_redirect off;  
 proxy\_set\_header Host $host;  
 proxy\_set\_header X-Real-IP $remote\_addr;  
 proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  
 proxy\_set\_header X-Forwarded-Host $server\_name;  
 }  
}

Dockerfile

**FROM** nginx:1.14  
  
**COPY** nginx.conf **/**etc**/**nginx**/**conf.d**/**default.conf

nginx.conf dosyasını nginx imajının içerisine koyuluyor

**version**: **"3.1"  
  
services**:  
 **proxy**:  
 **build**:  
 **context**: .  
 **dockerfile**: Dockerfile  
 **ports**:  
 - **"80:80"  
 web**:  
 **image**: httpd  
 **volumes**:  
 - ./html:/usr/local/apache2/htdocs/

Dockerfile compose a tanıtmak için build tagı kullanılıyor

Context :Dosyaların bulunduğu dizin.Dockerfile vs.

. bulunduğu dizin demek

dockerfile :Buraya build alacağımız docker file nin ismini giriyoruz

web konteynarındada bulunduğumuz dizine html adında bir template koyup bu template web konteynarındaki htdocs dizini volume ile mapledik.

docker compose-up

dediğimizde localhosta bootstrap templatenin göründüğünü görürüz.

Templatede bir yeri değiştirdiğimizde direk konteynara yansır.Konteynarı deploy etmekle uğraşmayız.

docker compose-down

Tüm konteynarlarımızı kapattık ve sildik

docker image ls

dediğimizde dockercompose\_proxy imagenin silinmediğini görürüz

docker-compose down --help

ilgili komutları listeler

docker-compose down --rmi local

Tüm konteynarları ve imageleride sildi

docker image ls

dediğimizde docker imagesi artık yok olduğunu görürüz.imageler çok fazla yer tutuyorsa silmek gerekir.