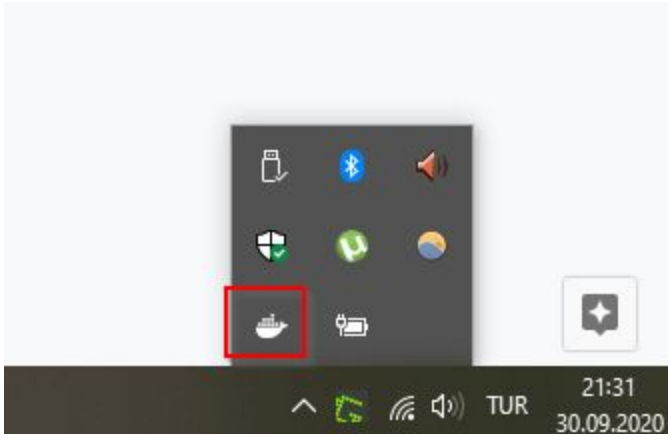
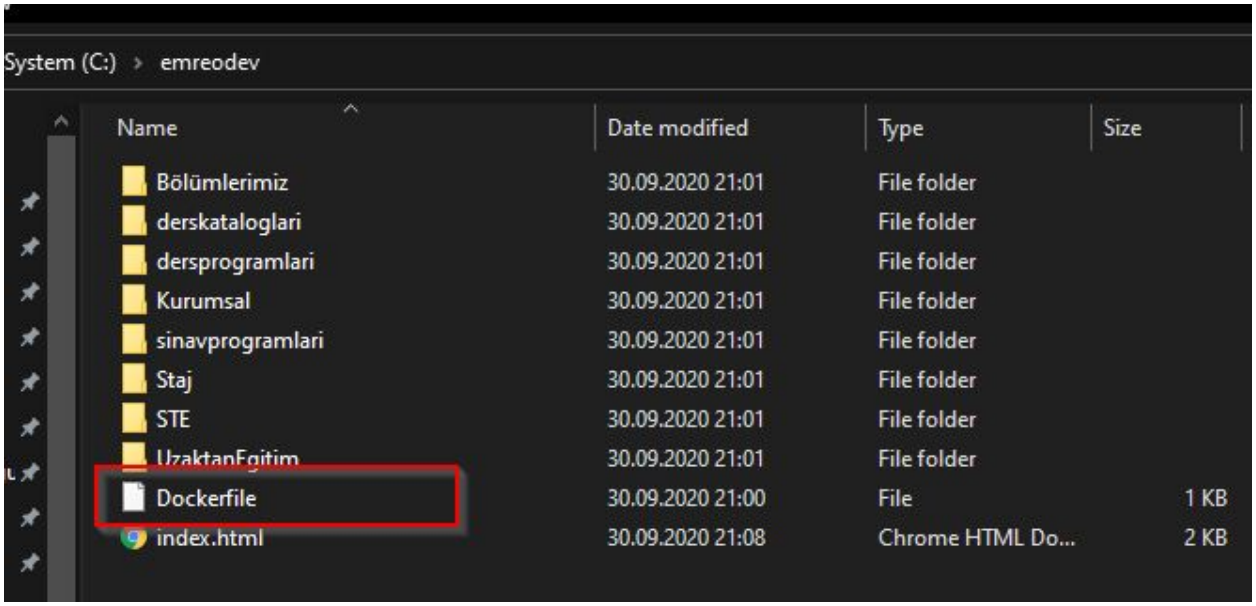
Docker Üzerinde Nginx Web Sunucusu ile Statik Site Kurulumu

1. Öncelikle bilgisayarımızda Docker kurulu olmalı ve Docker ın çalıştığından emin olalım; task bar daki simgesi şu şekilde olmalı. İlk açılıştaki yüklenme animasyonunun bitmesi gerekiyor.



2. Sitenin bulunduğu klasöre gidilir (örn: C:\emreodev) ve ismi **DockerFile** (Uzantısı olmayacak, ismi sadece bu) olan bir dosya yaratılır.

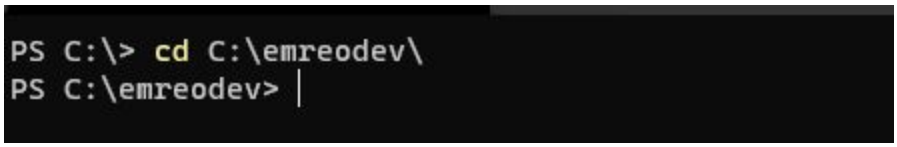


3. Bu dosya text editörde açılır ve içine şu aşağıdaki 2 satır yönerge yazılır. Bu dosyayı Docker'a ne yapacağını söyleyeceğimiz bir komut listesi şeklinde düşünebiliriz. Docker; sen bana bir sanal makine yarat ve onun içinde şunları yap gibi.

```
FROM nginx:alpine
COPY . /usr/share/nginx/html
```

4. Şimdi buradaki komutların anlamlarını açıklayalım.
    a. Birinci satırda Docker'a şunu diyoruz; Bana internetten (dockerhub'dan) nginx'in alpine sürümünü al getir, yani download et ve sanal makineme kur.
    b. İkinci satırda COPY kopyala komutudur. 2 parametresi vardır. Kaynak ve hedef. Kaynağımız bizim şu anda kullandığımız makinemiz, hedefimiz ise sanal makinenin içi gibi düşünebiliriz. COPY nin yanındaki nokta (.) yani kaynağımız; şu an içinde bulunduğumuz klasörde ne var ne yoksa herşeyi al demek. Noktanın yanındaki boşluktan sonra gelen "/usr/share/nginx/html" hedefimiz. Birinci satırda Nginx al kur dedik. Nginx'in dosyalarımızı hizmete almak için standart olarak kullandığı klasörün ismi. Yani burada yaptığımız özetle; benim dosyalarımı al, nginx in sanal makinedeki dizinine kopyala.

5. Bu işlemden sonda komut satırında c:\emreodev dizinine gidilir.



6. Docker uygulamaları sanal makinede container haline getirmek için Image lar kullanır. Image demek bir sanal makine kurulumu demek. Hani windows kurmak için bir cd kullanırız. Aynı onun gibi şimdi bilgisayarımızda bu sanal makinenin bir image'ını build edeceğiz.

7. C:\emreodev altında aşağıdaki komutu çalıştırıyoruz;

```
docker build -t emreodev:v1 .
```

8. Bu komutu açıklayalım;
    a. En baştaki "docker build" bir kalıp; Docker'a image build etmesi için komut veriyoruz.
    b. -t bir parametre (tag demek) t parametresi hemen sonrasında bizden bu image'ın ismi ne olsun'u girmemizi bekler. Buna "emreodev:v1" diyoruz. Bu emreodev isminde bir image yarat, bunun versiyonu v1 olsun demek

c. En sonradaki . (nokta) şu demek; docker file'ım içinde bulunduğum dizinde yani c:\emreodev de.

9. Komutu çalıştırıyoruz, docker bizim dockerfile ı buluyor ve işlemeye başlıyor. 4. Adımda anlattığım yönergeleri uygulayarak docker bize bir image oluşturur. (Docker ilk adımda söylediğimiz Nginx i bilgisayarda yoksa indirecek, dolayısıyla çok az bekleyeceğiz)

10. Successfully built 06863a94841d
    Successfully tagged emreodev:v1

    Mesajlarını gördüğümüzde image ımız oluşmuş demektir.

11. Şimdi bu image ı bir container haline getirip, bilgisayarımızda çalıştıracağız. Yani bir web sunucumuz olacak. Bunun için yine aynı dizinde şu komutu çalıştırıyoruz;

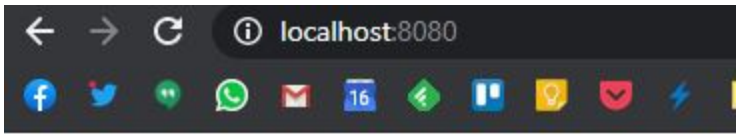
```
docker run -d -p 8080:80 emreodev:v1
```

12. Bu komutu açıklayalım;
    a. En baştaki "docker run" bir kalıp; Docker'a image ı container haline getirip ayağı kaldır komutu veriyoruz.
    b. Parametre -d docker a; bu komut çalıştıktan sonra container hep ayakta kalsın demek (daemon)
    c. Parametre -p port anlamına gelir, bizden bir port bilgisi bekler, burada 8080:80 şu anlama gelir; iki noktadan önceki 8080 benim makinedeki port, iki noktadan sonraki 80 ise sanal makinedeki port. Nginx sanal makinede ayağa kalkınca standart web sunucu portu olan 80 numaralı portu dinlemeye başlar. Burada şunu diyoruz özetle; ben kendi makinemde 8080 numaralı porta bir istek gönderirsem, sanal makineden 80 numaralı porta bunu yönlendir. Yani routing yapıyoruz.
    d. En sonraki ise şu an çalıştırmak istediğimiz image yani emreodev:v1

13. Komut çalışıp bitince şöyle bir ekran görmemiz lazım;

```
PS C:\emreodev> docker run -d -p 8080:80 emreodev:v1
146d9d05cbbcdff62158ec548c98fd7b14066e9b2cd39ad4eec8a655c962ae77
PS C:\emreodev>
```

14. Bu uygulamamız container haline geldi demek. Artık bilgisayarımızda bir browser açıp, localhost:8080 i çağırdığımızda sitemiz gelir, yani web sunucumuzda sitemiz ayağı kalkmış olur ve ana sayfamız gelir.



Ana Sayfa

15. Çalışan container ımızı görmek ve mesela çalışmasını durdurmak için önce container listeleme yaparız; mesela aşağıdaki listelemede emreodev:v1 image lı container arka planda 9 dakikadır çalışıyormuş.

docker container ls

```
PS C:\emreodev> docker container ls
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
  PORTS                  NAMES
146d9d05cbbc        emreodev:v1         "/docker-entrypoint.…"   9 minutes ago       Up 9 minutes
  0.0.0.0:8080->80/tcp   eloquent_burnell
PS C:\emreodev>
```

16. Sonra mesela siteyi durdurmak için listelemede gelen container id nin ilk 3 hanesini (**146**d9d05cbbc) kullanarak şu komutla sitemizin çalışmasını durdurabiliriz;

docker container stop 146

```
PS C:\emreodev> docker container stop 146
146
```

17. Artık localhost:8080 hizmet vermeyecektir.