

Nginx ile Basit Bir Bootstrap Web Sitesini Docker Konteynerinde Çalıştırma

Adım 1: Basit Bir HTML ve Bootstrap Web Sitesi Oluşturma

1. **Proje Klasörü Oluşturun:** Proje dosyalarınızı saklayacağınız bir dizin oluşturun.
2. **index.html Dosyasını Oluşturun:** Proje dizininde `index.html` adında bir dosya oluşturun ve aşağıdaki içeriği ekleyin:

`index.html`

```
<!DOCTYPE html>
<html lang="tr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap Web Sitesi Demo</title>
  <link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.m
in.css" rel="stylesheet">
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="#">ibb-tech</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item active">
          <a class="nav-link" href="#">Anasayfa <span
class="sr-only">(şu anki)</span></a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Docker</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Kubernetes</a>
        </li>
      </ul>
    </div>
  </nav>
  <div class="container mt-5">
    <div class="jumbotron">
```

```
<h1 class="display-4">Docker Image!</h1>
<p class="lead">Docker imajı, bir uygulamanın çalışması için
gereken dosyaları, bağımlılıkları ve yapılandırmaları içeren salt okunur
bir şablondur. Docker imajı, birbirine bağlı bir dizi katmandan oluşur ve
taşınabilir olup, farklı ortamlar arasında (geliştirme, test, üretim)
tutarlı çalışmayı sağlar. Docker imajları, Docker Hub gibi container
registry'lerde depolanır ve paylaşılabilir.</p>
<hr class="my-4">
<p>İmajlar, Dockerfile kullanarak oluşturulur.
İmajlar salt okunur yapıya sahiptir; bir konteyner çalıştırıldığında,
imajın üzerine yazılabilir bir katman eklenir.
İmajlar, sürüm numaraları veya etiketlerle (tags) versiyonlanabilir
Docker imajları, konteyner çalıştırmak için temel alınır. Bir imajdan
birden fazla konteyner oluşturulabilir.
İmajlar, konteynerleri oluşturmak için kullanılır ve içerikleri
düzenlenemez, ancak silinebilirler
</p>
<a class="btn btn-primary btn-lg" href="#" role="button">Daha
fazla bilgi</a>
</div>
</div>
<script
src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.m
in.js"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min
.js"></script>
</body>
</html>
```

Adım 2: Dockerfile Oluşturma

Proje dizininde **Dockerfile** adında bir dosya oluşturun ve aşağıdaki içeriği ekleyin:

Dockerfile

```
# Resmi Nginx imajını temel olarak kullan
FROM nginx:alpine

# HTML dosyalarını Nginx'in html dizinine kopyala
COPY . /usr/share/nginx/html

# 80 portunu aç
```

```
EXPOSE 80
```

```
# Konteyner başlatıldığında Nginx'i çalıştır  
CMD ["nginx", "-g", "daemon off;"]
```

Adım 3: Docker İmajını Oluşturma

```
docker build -t bootstrap-web:latest .
```

Adım 4: Docker İmajını Doğrulama

İmajın başarılı bir şekilde oluşturulduğunu doğrulamak için tüm Docker imajlarını listeleyin:

```
docker images
```

Adım 5: Docker İmajın çalışıp çalışmadığını Test Etme

```
docker run -d -p 8080:80 bootstrap-web:latest
```

Adım 6: Docker Hub Repository'si Oluşturma

Docker Hub'da bir hesap oluşturduysanız, oturum açın ve yeni bir repository oluşturun:

- “Create Repository” butonuna tıklayın.
- Repository'nize bir ad verin, örneğin **bootstrap-web**.
- Görünürlüğü tercihinize göre “Public” veya “Private” olarak ayarlayın.

Adım 7: Docker İmajını Etiketleme

Docker Hub, imajlar için belirli bir adlandırma kuralı gerektirir. Docker imajınızı Docker Hub kullanıcı adınız ve repository adı ile etiketleyin:

```
docker tag bootstrap-web:latest  
your-dockerhub-username/bootstrap-web:latest
```

<your-dockerhub-username> kısmını Docker Hub kullanıcı adınızla değiştirin.

Adım 8: Docker Hub'a Komut Satırından Giriş Yapma

Docker Hub hesabınıza komut satırından giriş yapın:

```
docker login
```

Docker Hub kullanıcı adınızı ve şifrenizi girmeniz istenecektir.

Adım 9: Docker İmajını Docker Hub'a Gönderme

```
docker push <your-dockerhub-username>/bootstrap-web:latest
```

`<your-dockerhub-username>` kısmını Docker Hub kullanıcı adınızla değiştirin.

Adım 10: Docker Hub'da İmajı Doğrulama

Gönderme işlemi tamamlandıktan sonra, Docker Hub repository'nizde imajın başarılı bir şekilde yüklendiğini doğrulamak için:

- Web tarayıcınızda Docker Hub'a gidin.
- Profilinize gidin ve `bootstrap-web` repository'sini bulun.
- İmajın "Tags" bölümünde listelendiğini doğrulayın.

Adım 11: Docker Hub'dan İmajı Çekme ve Çalıştırma

İmajınızın Docker Hub'dan çekildiğinde çalıştığını doğrulamak için, Docker'ın kurulu olduğu herhangi bir makinede aşağıdaki komutu kullanabilirsiniz:

```
docker run -d -p 8080:80 <your-dockerhub-username>/bootstrap-web:latest
```

Web tarayıcınızı açın ve `http://localhost:8080` adresine gidin. Docker Hub imajından çalışan basit Bootstrap web sitenizi görmelisiniz.