

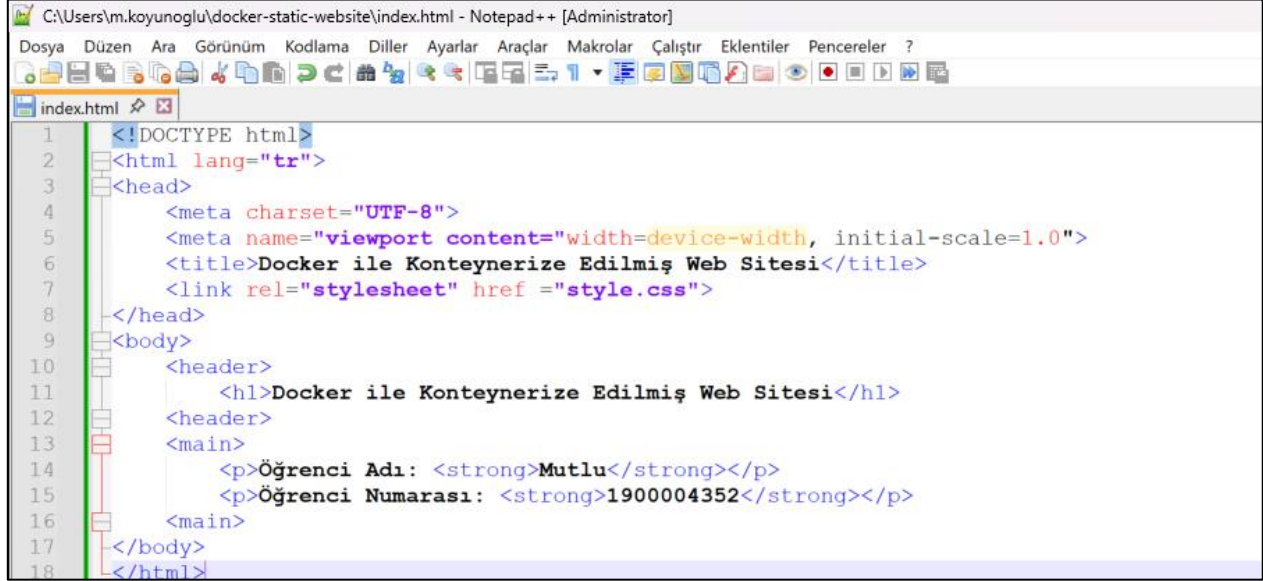
İSTANBUL KÜLTÜR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
2024-2025 AKADEMİK YILI

CSE0559-YAZILIM MİMARİSİ VE ENTEGRASYONU

MUTLU KOYUNOĞLU
1900004352

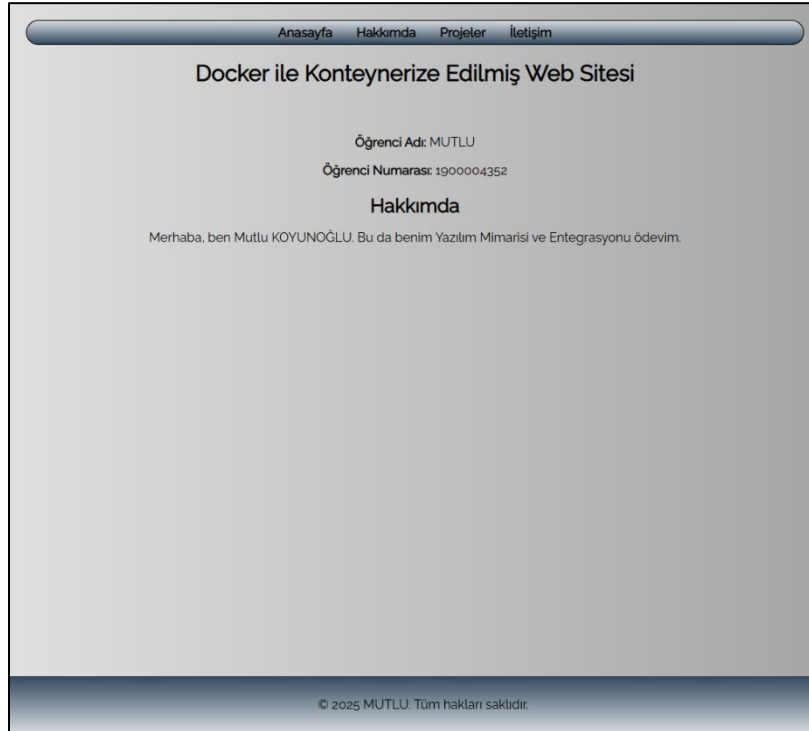
Çalışma Süreci

İlk olarak yapılacak çalışmanın bulunacağı klasör oluşturulmuştur. Dosya yolu olarak “C:\Users\m.koyunoglu\docker-static-website” seçilmiştir. Devamında “index.html” dosyası ve bunun paralelinde bir “style.css” dosyası oluşturulmuştur. Bu dosyaların düzenlenmesinde **Notepad++** uygulaması kullanılmıştır. İlk etapta “index.html” içeriği aşağıdaki gibidir.



```
1 <!DOCTYPE html>
2 <html lang="tr">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport content="width=device-width, initial-scale=1.0">
6 <title>Docker ile Konteynerize Edilmiş Web Sitesi</title>
7 <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10 <header>
11 <h1>Docker ile Konteynerize Edilmiş Web Sitesi</h1>
12 <header>
13 <main>
14 <p>Öğrenci Adı: <strong>Mutlu</strong></p>
15 <p>Öğrenci Numarası: <strong>1900004352</strong></p>
16 </main>
17 </body>
18 </html>
```

Birkaç adım sonra işin içine “style.css” girmesiyle aşağıdaki gibi bir site oluşturulmuştur.



Sürecin ilerlemesiyle siteye birkaç fotoğraf/bilgi/haberi slider olarak koyma fikri gelmiştir. Slider çalışmaları için **“script.js”** dosyası sürece dahil edilmiştir. **ChatGPT** vasıtasıyla üretilen iki adet fotoğraf kullanılarak slider çalışmasının ilk adımları aşağıdaki gibi atılmıştır.

```
k.html style.css script.js
<!DOCTYPE html>
<html lang="tr">
<head>

<link href="https://fonts.googleapis.com/css2?family=Raleway&display=swap" rel="stylesheet">
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Docker ile Konteynerize Edilmiş Web Sitesi</title>
<link rel="stylesheet" href="style.css">

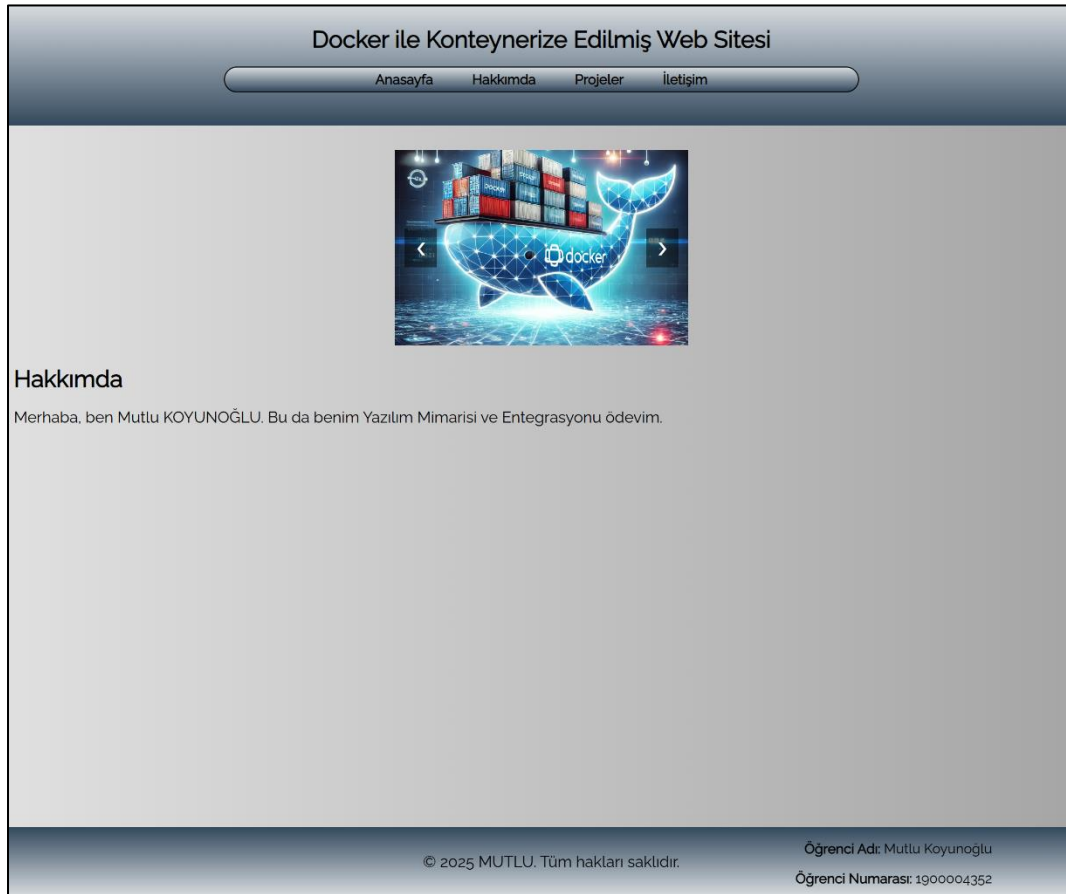
</head>
<body>
<header>
<script src="script.js"> </script>
<h1>Docker ile Konteynerize Edilmiş Web Sitesi</h1>

<nav>
<ul>
<li><a href="#">Anasayfa</a></li>
<li><a href="#">Hakkımda</a></li>
<li><a href="#">Projeler</a></li>
<li><a href="#">İletişim</a></li>
</ul>
</nav>
</header>
<main>
<div class="slider">
<div class="slides">
<div class="slide active">

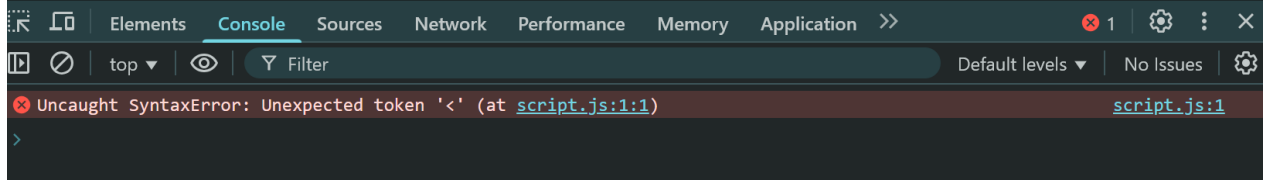
</div>
<div class="slide">

</div>
</div>
<button class="prev">#10094</button>
<button class="next">#10095</button>
</div>
<section class="about">
<h2>Hakkımda</h2>
<p>Merhaba, ben Mutlu KOYUNOĞLU. Bu da benim Yazılım Mimarisi ve Entegrasyonu ödevim.</p>
</section>
</main>
<footer>
<p>#copy; 2025 MUTLU. Tüm hakları saklıdır.</p>
<section class="intro">
<p><strong>Öğrenci Adı:</strong> Mutlu Koyunoğlu</p>
<p><strong>Öğrenci Numarası:</strong> 1900004352</p>
</section>
</footer>
</body>
</html>
```

```
1 <script>
2 // Mevcut slide indeksini tutar
3 let slideIndex = 0;
4 // Tüm slide'ları seç
5 const slides = document.querySelectorAll('.slide');
6 // Toplam slide sayısı
7 const totalSlides = slides.length;
8 // Slide'ları içeren kapsayıcı
9 const slidesContainer = document.querySelector('.slides');
10
11 // Slide'ı gösterme fonksiyonu
12 function showSlide(index) {
13 // Her slide genişliğini baz alarak container'ı kaydırıyoruz
14 slidesContainer.style.transform = 'translateX(' + (-index * 300) + 'px)';
15 }
16
17 // Sonraki düğmesine tıklanınca
18 document.querySelector('.next').addEventListener('click', function() {
19 slideIndex = (slideIndex + 1) % totalSlides;
20 showSlide(slideIndex);
21 });
22
23 // Önceki düğmesine tıklanınca
24 document.querySelector('.prev').addEventListener('click', function() {
25 slideIndex = (slideIndex - 1 + totalSlides) % totalSlides;
26 showSlide(slideIndex);
27 });
28 </script>
```

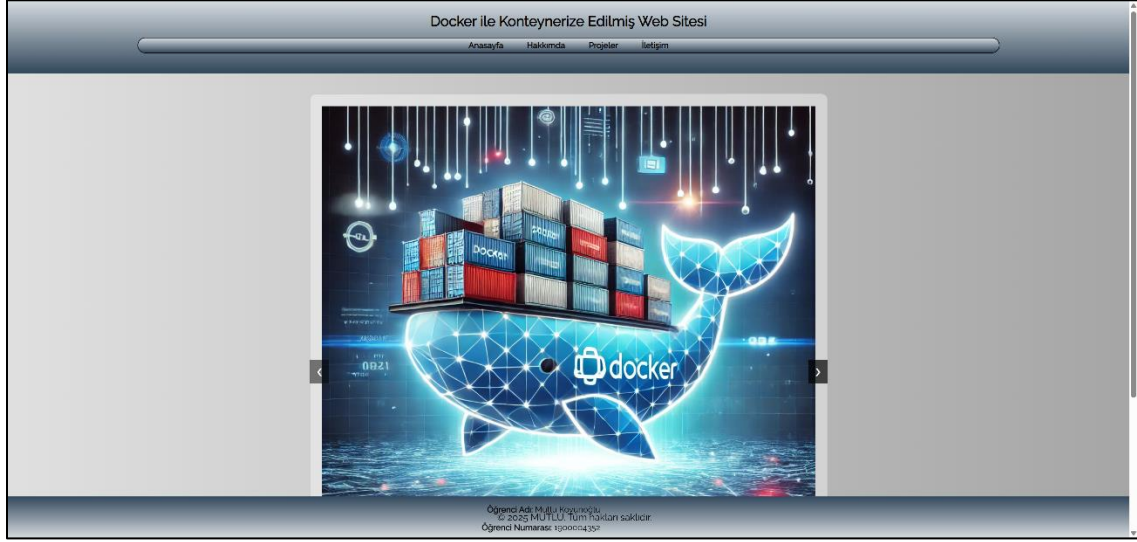


Burada görüleceği üzere “Öğrenci Adı ve Numarası” bilgisi footer alanının sağ tarafına alınmıştır. Bunun haricinde ilk etapta slider özelliğini gerçekleştirme için fotoğraflar üzerine konan slider butonlarının çalışmadığı fark edilmiştir. Console üzerinden ilgili hata yakalanmıştır.



“script.js” dosyasında yapılan hata tespit edilmiş ve problem düzeltilmiştir.

Sürecin devamında bu sliderları yan yana koymanın daha estetik duracağı düşünülmüş ve o yolda yeni adımlar atılmıştır.



Bu oluşan görüntünün estetikliğini geliştirmek açısından “style.css” ve “script.js” dosyalarında düzenlemeler sonrası aşağıdaki sonuca ulaşılmıştır.



2.adıma geçilmiş ve **Notepad++** ile bir **Dockerfile** Dosyası oluşturulmuştur.

```
C:\Users\m.koyunoglu\docker-static-website\ Dockerfile - Notepad++ [Administrator]
Dosya Düzen Ara Görünüm Kodlama Diller Ayarlar Araçlar Makrolar Çalıştır Eklentiler Pencerele...
index.html style.css script.js Dockerfile
1 # Nginx tabanlı imaj kullanıyoruz
2 FROM nginx:alpine
3
4 # Web sitesi dosyalarını Nginx'in servis ettiği dizine kopyala
5 COPY . /usr/share/nginx/html
6
7 # Container içinde 80. portu aç
8 EXPOSE 80
```

Web sunucusu (**Nginx**) kullanarak siteyi servis edecek şekilde yapılandırıldı. Ardından “**docker build -t mutlukoyunoglu.**” Komutu ile imaj oluşturuldu ve etiketlendi. 2. Adım tamamlanmış oldu.

```
Administrator: Komut İstemi
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\m.koyunoglu>cd C:\Users\m.koyunoglu\docker-static-website

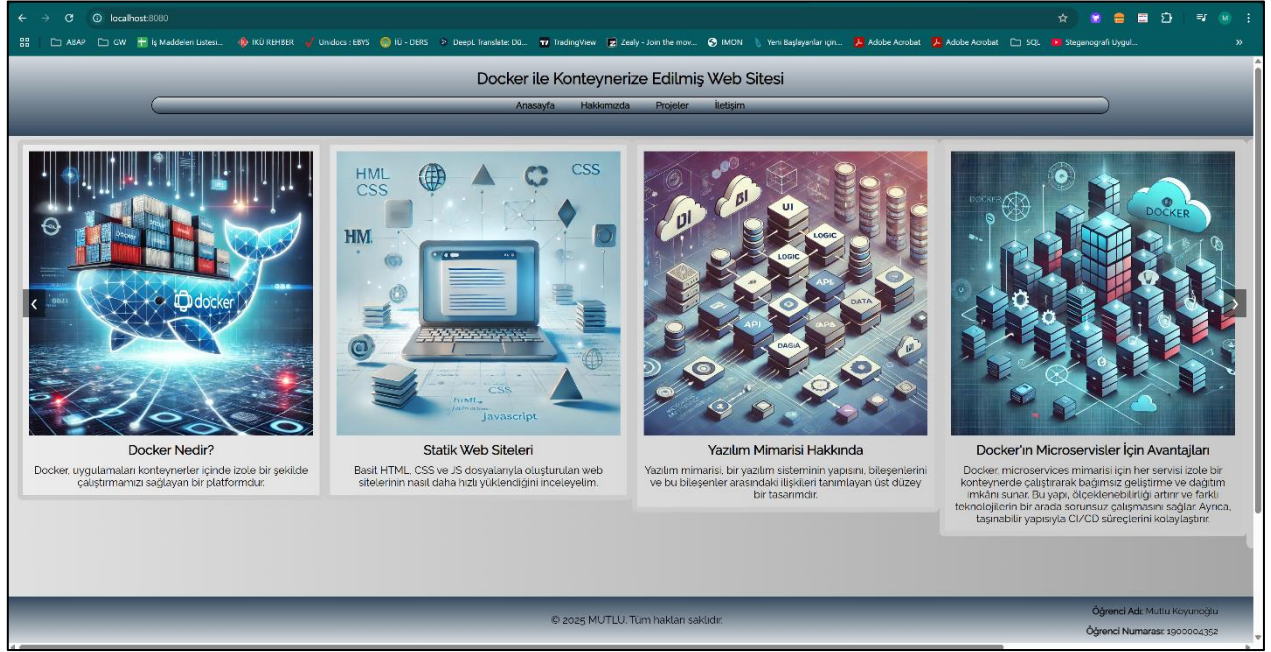
C:\Users\m.koyunoglu\docker-static-website>docker build -t mutlukoyunoglu .
[*] Building 19.0s (8/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 242B
=> [internal] load metadata for docker.io/library/nginx:alpine
=> [auth] library/nginx:pull token for registry-1.docker.io
=> [internal] load dockerignore
=> transferring context: 2B
=> [internal] load build context
=> transferring context: 1.79MB
=> [1/2] FROM docker.io/library/nginx:alpine@sha256:8ff102c5d78d254a6f0da862b3cf39eaf07f01ee0927fd21e219dbaf8bc8591
=> resolve docker.io/library/nginx:alpine@sha256:8ff102c5d78d254a6f0da862b3cf39eaf07f01ee0927fd21e219dbaf8bc8591
=> sha256:8c7e4c092ab716c95e1a8040943549b4ac5e6c5d1alc744996528e1b03e1e67a 15.38MB / 15.38MB
=> sha256:6d79cc0804d434876ce0f038c675d20532f28e476283a29e7e63bc0bf13a4ed6 1.40kB / 1.40kB
=> sha256:ab3286a7348303a1b09a5189f63f1414cc1de4e397088dcd07edb322df1fe9 1.21kB / 1.21kB
=> sha256:8d72c872a38f1ccf2428172acbe3b28018ff2d014f92de25b0210e46256deb 402B / 402B
=> sha256:984583bcf083fa6908b5c7834795a5a57a9b4dfe7408d5358474f5d309e25ace 955B / 955B
=> sha256:43f2ec46bdf5babe56d7e13b06b380f42cbf2103f6665c349f0b98dac8962 627B / 627B
=> sha256:ccc35e5d2d6dd115290f1074afc6ad1c474b6f94097434e6befd33b781d2 1.79MB / 1.79MB
=> sha256:f1822174bc91701f43d496d85011092101a832a93a388b79e99e69c2d5c870 3.64MB / 3.64MB
=> extracting sha256:f1822174bc91701f43d496d85011092101a832a93a388b79e99e69c2d5c870
=> extracting sha256:ccc35e5d2d6dd115290f1074afc6ad1c474b6f94097434e6befd33b781d2
=> extracting sha256:43f2ec46bdf5babe56d7e13b06b380f42cbf2103f6665c349f0b98dac8962
=> extracting sha256:984583bcf083fa6908b5c7834795a5a57a9b4dfe7408d5358474f5d309e25ace
=> extracting sha256:8d72c872a38f1ccf2428172acbe3b28018ff2d014f92de25b0210e46256deb
=> extracting sha256:ab3286a7348303a1b09a5189f63f1414cc1de4e397088dcd07edb322df1fe9
=> extracting sha256:6d79cc0804d434876ce0f038c675d20532f28e476283a29e7e63bc0bf13a4ed6
=> extracting sha256:8c7e4c092ab716c95e1a8040943549b4ac5e6c5d1alc744996528e1b03e1e67a
=> [2/2] COPY /usr/share/nginx/html
=> exporting image
=> exporting layers
=> exporting manifest sha256:d463b7d82ef3aab9e2dfee5c568cd7b2042e73834ae3d0588876398872f89ec
=> exporting config sha256:979e3d38ab0750c10008f28c211f3e8fe422bcf5d097f73f78cbb0afdee123
=> exporting attestation manifest sha256:8c278cdf280cd0b3aa00da1da3cec275c23491d5bcb5f4fbd0dc21bb3de8
=> exporting manifest list sha256:e08d1aa642ca2b5add8bec1667777dcf32dc256f13deaa4232dc7aba32ad27d
=> naming to docker.io/library/mutlukoyunoglu:latest
=> unpacking to docker.io/library/mutlukoyunoglu:latest

C:\Users\m.koyunoglu\docker-static-website>
```

3. Aşamada Oluşturulan image’den bir konteyner çalıştırılmıştır.

```
C:\Users\m.koyunoglu\docker-static-website>docker run -d -p 8080:80 mutlukoyunoglu
60ce5478b2f34dea7c2a87593ddd91a0fed9038d4b68a1e59e6b055dddfc5b4d0
```


Devamında **localhost:8080** linki üzerinden siteye erişim sağlanmıştır.



Sonraki adım için önce Konteyner bilgilerini öğrenmek için **“docker ps”** komutu yazılmıştır.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
60ce5478b2f3	mutlukoyunoglu	"/docker-entrypoint..."	2 minutes ago	Up 2 minutes	0.0.0.0:8080->80/tcp	awesome_lederberg

Öğrenilen ID ile **Docker logs** çıktıları alınmıştır.

```
C:\Users\m.koyunoglu\docker-static-website>docker logs 60ce5478b2f3
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/03/21 11:49:06 [notice] 1#1: using the "epoll" event method
2025/03/21 11:49:06 [notice] 1#1: nginx/1.27.4
2025/03/21 11:49:06 [notice] 1#1: built by gcc 14.2.0 (Alpine 14.2.0)
2025/03/21 11:49:06 [notice] 1#1: OS: Linux 5.15.167.4-microsoft-standard-WSL2
2025/03/21 11:49:06 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/03/21 11:49:06 [notice] 1#1: start worker processes
2025/03/21 11:49:06 [notice] 1#1: start worker process 39
2025/03/21 11:49:06 [notice] 1#1: start worker process 31
2025/03/21 11:49:06 [notice] 1#1: start worker process 32
2025/03/21 11:49:06 [notice] 1#1: start worker process 33
2025/03/21 11:49:06 [notice] 1#1: start worker process 34
2025/03/21 11:49:06 [notice] 1#1: start worker process 35
2025/03/21 11:49:06 [notice] 1#1: start worker process 36
2025/03/21 11:49:06 [notice] 1#1: start worker process 37
2025/03/21 11:49:06 [notice] 1#1: start worker process 38
2025/03/21 11:49:06 [notice] 1#1: start worker process 39
2025/03/21 11:49:06 [notice] 1#1: start worker process 40
2025/03/21 11:49:06 [notice] 1#1: start worker process 41
172.17.0.1 - - [21/Mar/2025:11:49:33 +0000] "GET / HTTP/1.1" 200 4135 "-" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36 "-"
172.17.0.1 - - [21/Mar/2025:11:49:33 +0000] "GET /style.css HTTP/1.1" 200 3312 "http://localhost:8080/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36 "-"
172.17.0.1 - - [21/Mar/2025:11:49:33 +0000] "GET /images/post1.jpg HTTP/1.1" 200 379808 "http://localhost:8080/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36 "-"
172.17.0.1 - - [21/Mar/2025:11:49:33 +0000] "GET /images/post2.jpg HTTP/1.1" 200 192380 "http://localhost:8080/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36 "-"
172.17.0.1 - - [21/Mar/2025:11:49:35 +0000] "GET /images/post3.jpg HTTP/1.1" 200 394768 "http://localhost:8080/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36 "-"
172.17.0.1 - - [21/Mar/2025:11:49:35 +0000] "GET /images/post4.jpg HTTP/1.1" 200 366520 "http://localhost:8080/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36 "-"
172.17.0.1 - - [21/Mar/2025:11:49:35 +0000] "GET /images/post5.jpg HTTP/1.1" 200 438212 "http://localhost:8080/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36 "-"
172.17.0.1 - - [21/Mar/2025:11:49:35 +0000] "GET /script.js HTTP/1.1" 200 1614 "http://localhost:8080/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36 "-"
2025/03/21 11:49:35 [error] 33#33: *4 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "localhost:8080", referer: "http://localhost:8080/"
172.17.0.1 - - [21/Mar/2025:11:49:35 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http://localhost:8080/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36 "-"
```

Container inspect bilgilerine ulaşmak için “**docker inspect 60ce5478b2f3**” komutuyla aşağıdaki çıktıya ulaşılmıştır.

```
C:\Users\m.koyunoglu\docker-static-website>docker inspect 60ce5478b2f3
[
  {
    "Id": "60ce5478b2f34dea7c2a87593ddd91a8fed9838d4b68a1e59eeb855ddfc3b4d8",
    "Created": "2025-03-21T11:49:05.302170481Z",
    "Path": "/docker-entrypoint.sh",
    "Args": [
      "nginx",
      "daemon off;"
    ],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 1676,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2025-03-21T11:49:05.821243835Z",
      "FinishedAt": "0001-01-01T00:00:00Z"
    },
    "Image": "sha256:e6d1aa62c2a2badd8bec166777dcf32dc256f13deaa4232dc7aba12ad27d",
    "ResolvConfPath": "/var/lib/docker/containers/60ce5478b2f34dea7c2a87593ddd91a8fed9838d4b68a1e59eeb855ddfc3b4d8/resolv.conf",
    "HostnamePath": "/var/lib/docker/containers/60ce5478b2f34dea7c2a87593ddd91a8fed9838d4b68a1e59eeb855ddfc3b4d8/hostname",
    "HostsPath": "/var/lib/docker/containers/60ce5478b2f34dea7c2a87593ddd91a8fed9838d4b68a1e59eeb855ddfc3b4d8/hosts",
    "LogPath": "/var/lib/docker/containers/60ce5478b2f34dea7c2a87593ddd91a8fed9838d4b68a1e59eeb855ddfc3b4d8/60ce5478b2f34dea7c2a87593ddd91a8fed9838d4b68a1e59eeb855ddfc3b4d8-json.log",
    "Name": "/awesome-lederberg",
    "RestartCount": 0,
    "Driver": "overlayfs",
    "Platform": "linux",
    "MountLabel": "",
    "ProcessLabel": "",
    "AppArmorProfile": "",
    "ExecIDs": null,
    "HostConfig": {
      "Binds": null,
      "ContainerIDFile": "",
      "LogConfig": {
        "Type": "json-file",
        "Config": {}
      },
      "NetworkMode": "bridge",
      "PortBindings": {
        "80/tcp": [
          {
            "HostIp": "",
            "HostPort": "8080"
          }
        ]
      },
      "RestartPolicy": {
        "Name": "no",
        "MaximumRetryCount": 0
      },
      "AutoRemove": false,
      "VolumeDriver": "",
      "VolumesFrom": null,
      "ConsoleSize": {
        "W": 51,
        "H": 209
      },
      "CapAdd": null,

```

```

    "CgroupnsMode": "host",
    "Dns": [],
    "DnsOptions": [],
    "DnsSearch": [],
    "ExtraHosts": null,
    "GroupAdd": null,
    "IpcMode": "private",
    "Cgroup": "",
    "Links": null,
    "OomScoreAdj": 0,
    "PidMode": "",
    "Privileged": false,
    "PublishAllPorts": false,
    "ReadOnlyRootfs": false,
    "SecurityOpt": null,
    "UTSMode": "",
    "UsernsMode": "",
    "ShmSize": 67108864,
    "Runtime": "runc",
    "Isolation": "",
    "CpuShares": 0,
    "Memory": 0,
    "NanoCpus": 0,
    "CgroupParent": "",
    "BlkioWeight": 0,
    "BlkioWeightDevice": [],
    "BlkioDeviceReadBps": [],
    "BlkioDeviceWriteBps": [],
    "BlkioDeviceReadIops": [],
    "BlkioDeviceWriteIops": [],
    "CpuPeriod": 0,
    "CpuQuota": 0,
    "CpuRealtimePeriod": 0,
    "CpuRealtimeRuntime": 0,
    "CpusetCpus": "",
    "CpusetMems": "",
    "Devices": [],
    "DeviceCgroupRules": null,
    "DeviceRequests": null,
    "MemoryReservation": 0,
    "MemorySwap": 0,
    "MemorySwappiness": null,
    "OomKillDisable": false,
    "Pidfile": null,
    "PidsLimit": 0,
    "CpuCount": 0,
    "CpuPercent": 0,
    "IOMaxInops": 0,
    "IOMaxBandwidth": 0,
    "MaskedPaths": [
      "/proc/assound",
      "/proc/acpi",
      "/proc/kcore",
      "/proc/kmsg",
      "/proc/latency_stats",
      "/proc/timer_list",
      "/proc/timer_stats",
      "/proc/timedebg",
      "/proc/scsi",
      "/sys/firmware",
      "/sys/devices/virtual/powercap"
    ],
    "ReadOnlyPaths": [
      "/proc/bus",
      "/proc/fs",

```



```
Administrator: Komut İstemi X + v
{
  "/proc/bus",
  "/proc/fs",
  "/proc/i9n",
  "/proc/psa",
  "/proc/sysrq-trigger"
},
"GraphDriver": {
  "Data": null,
  "Name": "overLayfs"
},
"Mounts": [],
"Config": {
  "Hostname": "60ce5478b2f3",
  "Domainname": "",
  "User": "",
  "AttachStdin": false,
  "AttachStdout": false,
  "AttachStderr": false,
  "ExposedPorts": {
    "80/tcp": {}
  },
  "Tty": false,
  "OpenStdin": false,
  "StdinOnce": false,
  "Env": [
    "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
    "NGINX_VERSION=1.27.4",
    "PKG_RELEASE=1",
    "DYNWG_RELEASE=1",
    "NJS_VERSION=0.9",
    "NJS_RELEASE=1"
  ],
  "Cmd": [
    "nginx",
    "-g",
    "daemon off;"
  ],
  "Image": "mullkeyunoglu",
  "Volumes": null,
  "WorkingDir": "/",
  "Entrypoint": [
    "/docker-entrypoint.sh"
  ],
  "OnBuild": null,
  "Labels": {
    "maintainer": "NGINX Docker Maintainers <ubuntu@nginx.com>"
  },
  "StopSignal": "SIGQUIT"
},
"NetworkSettings": {
  "Bridge": "",
  "SandBoxIP": "du7688379fa59921f7e7e5df6e693b5a4f3724ce3b7e2fff8f9afc7881daf",
  "SandBoxkey": "/var/run/docker/netns/du7688379fa5",
  "Ports": {
    "80/tcp": [
      {
        "HostIp": "0.0.0.0",
        "HostPort": "8080"
      }
    ]
  },
  "HairpinMode": false,
  "LinkLocalIPv6Address": "",
  "LinkLocalIPv6Prefixlen": 0,

```

```
Administrator: Komut İstemi X + v
  "Bridge": "",
  "SandBoxIP": "du7688379fa59921f7e7e5df6e693b5a4f3724ce3b7e2fff8f9afc7881daf",
  "SandBoxkey": "/var/run/docker/netns/du7688379fa5",
  "Ports": {
    "80/tcp": [
      {
        "HostIp": "0.0.0.0",
        "HostPort": "8080"
      }
    ]
  },
  "HairpinMode": false,
  "LinkLocalIPv6Address": "",
  "LinkLocalIPv6Prefixlen": 0,
  "SecondaryIPAddresses": null,
  "SecondaryIPV6Addresses": null,
  "EndpointID": "7e1623938e3e7db63c836fa99dd6f9e5293aa5883b352364908165ef65f6877",
  "Gateway": "172.17.0.1",
  "GlobalIPv6Address": "",
  "GlobalIPv6Prefixlen": 0,
  "IPAddress": "172.17.0.2",
  "IPPrefixlen": 16,
  "IPv6Gateway": "",
  "MacAddress": "8e:5b:dd:98:16:94",
  "Networks": {
    "bridge": {
      "IPAMConfig": null,
      "Links": null,
      "Aliases": null,
      "MacAddress": "8e:5b:dd:98:16:94",
      "DriverOpts": null,
      "GwPriority": 0,
      "NetworkID": "f9ee9731a9d8-cf97888e2cc38686e124b81189a4d2b07048f33b6e33db",
      "EndpointID": "7e1623938e3e7db63c836fa99dd6f9e5293aa5883b352364908165ef65f6877",
      "Gateway": "172.17.0.1",
      "IPAddress": "172.17.0.2",
      "IPPrefixlen": 16,
      "IPv6Gateway": "",
      "GlobalIPv6Address": "",
      "GlobalIPv6Prefixlen": 0,
      "DNSNames": null
    }
  }
},
"ImageManifestDescriptor": {
  "MediaType": "application/vnd.oci.image.manifest.v1+json",
  "digest": "sha256:0ac3b7c82ef3aab9e2f6e5c568c7b2842e73834ae3d9588876398072f8ec",
  "size": 2081,
  "platform": {
    "architecture": "amd64",
    "os": "Linux"
  }
}
}
}
C:\Users\m.koyunoglu\docker-static-website\
```

4.adımda yapılmasını istenenler için bir “**default.conf**” dosyası oluşturulmuştur. Bu dosya, **reverse proxy** ayarlarını içermektedir. Bu konfigürasyon, **webapp** adındaki servise gelen tüm istekleri yönlendirir. **Docker Compose** içinde webapp servisine 3 kopya (replica) oluşturduğumuzda, istekler **round-robin** ile dağıtılacaktır.

```
C:\Users\m.koyunoglu\docker-static-website\default.conf - Notepad++ [Administrator]
Dosya Düzen Ara Görünüm Kodlama Diller Ayarlar Araçlar Makrolar Çalıştır Eklentiler Pencerele
index.html style.css script.js Dockerfile default.conf
1 upstream webapp {
2     # webapp servisin 80 portuna proxy
3     server webapp:80;
4 }
5
6 server {
7     listen 80;
8
9     location / {
10        proxy_pass http://webapp;
11    }
12 }
```

```
C:\Users\m.koyunoglu\docker-static-website\docker-compose.yml - Notepad++ [Administrator]
Dosya Düzen Ara Görünüm Kodlama Diller Ayarlar Araçlar Makrolar Çalıştır Eklentiler Pencerele
index.html style.css script.js Dockerfile default.conf docker-compose.yml
1 version: '3.8'
2 services:
3     webapp:
4         image: mutlukoyunoglu #daha önce oluşturulan docker image
5         container_name: webapp
6         expose:
7             - "80" #sadece container içi networkteki 80 portunu açar
8
9     loadbalancer:
10        image: nginx:alpine
11        container_name: loadbalancer
12        depends_on:
13            - webapp
14        ports:
15            - "8080:80" #hosttaki 8080'e gelen istekleri containerdaki 80 portuna yönlendirir
16        volumes:
17            - ./default.conf:/etc/nginx/conf.d/default.conf #default dosyası, loadbalancer conteneinar'ındaki nginx konf. olacak
18
19
```

Docker compose ile servisleri ayağa kaldırma işlemi için “**docker-compose up -d**” komutu çalıştırılmış ve ilgili portun **kullanıldığı bilgisi geri dönmüştür**. Mevcut konteynerleri kontrol etmek için “**docker ps**” komutu yazılmış ve **hala port 8080 kullanan (id:60ce5478b2f3) bir konteynerin hala çalıştığı tespit edilmiştir**.

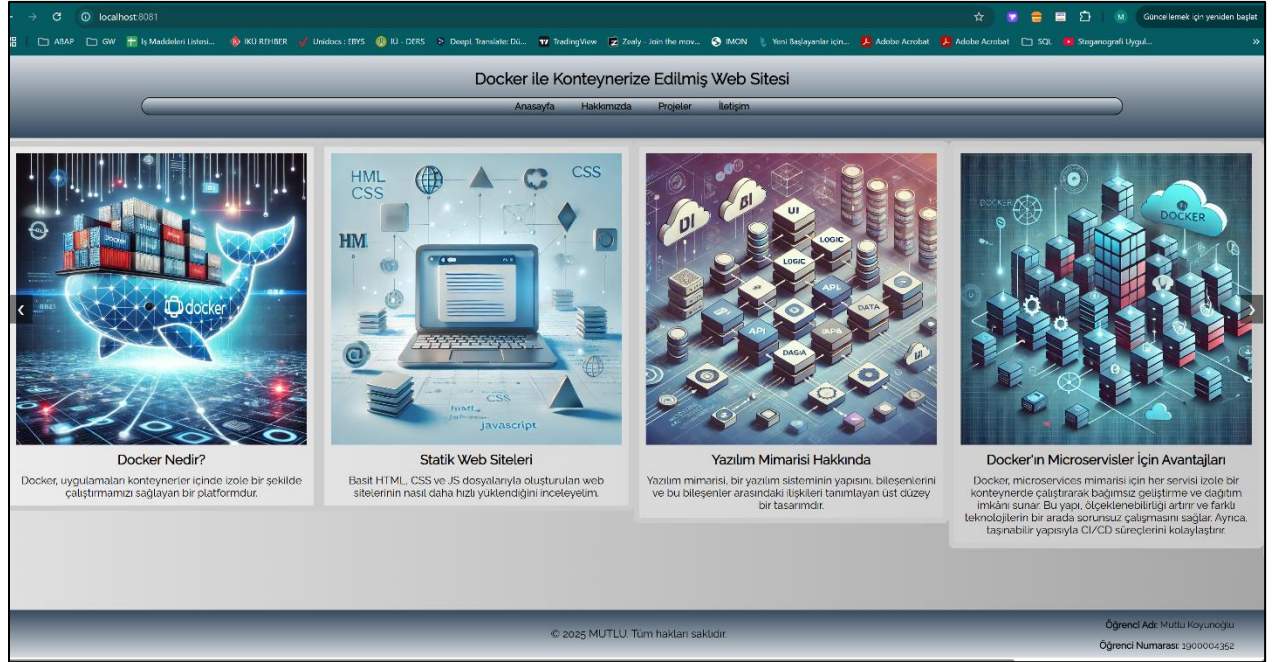
```
C:\Users\m.koyunoglu\docker-static-website>docker-compose up -d
time="2025-03-23T13:34:33+03:00" level=warning msg="C:\\Users\\m.koyunoglu\\docker-static-website\\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 4/4
  ✓ loadbalancer Pulled
[+] Running 2/3
  ✓ Network docker-static-website_default Created
  ✓ Container webapp Starting
  ✓ Container loadbalancer Starting
Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint loadbalancer (52121c9922691ddfa4e580b9ca0c72d54de4699a85d08cf3e8b6b631db50ee9): Bind for 0.0.0.0:8080 failed : port is already allocated

C:\Users\m.koyunoglu\docker-static-website>docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
web97a877235   mutlukoyunoglu "/docker-entrypoint..." 3 minutes ago Up 3 minutes    80/tcp              webapp
60ce5478b2f3   mutlukoyunoglu "/docker-entrypoint..." 47 hours ago  Up 47 hours    0.0.0.0:8080->80/tcp  awesome-lederberg
```

Bu sorunun çözülmesi için ya mevcut konteynerleri **durdurmak/silmek ya da farklı bir port kullanmak gerekmektedir**. Farklı bir port kullanmak için **“docker-compose.yml”** dosyası içindeki port bilgisini **8081:80** olarak güncellenmiştir. Güncelleme sonrası **“docker-compose up -d”** komutu ekrandaki bilgiyi dönmüştür.

```
C:\Users\m.koyunoglu\docker-static-website>docker-compose up -d
time="2025-03-23T13:51:46+03:00" level=warning msg="C:\Users\m.koyunoglu\docker-static-website\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 2/2
 ✓ Container webapp      Running
 ✓ Container loadbalancer Started
```

Tarayıcıda **“localhost:8081”** adresine giderek test gerçekleştirilmiştir.



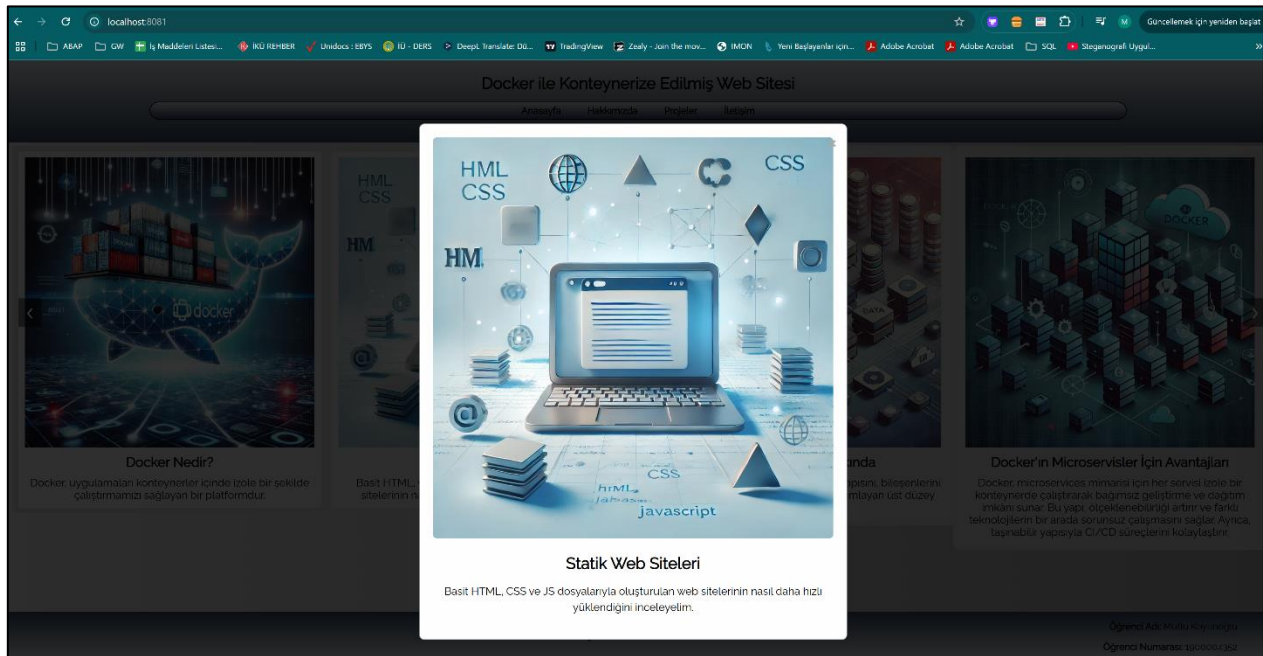
Bu aşamada tek kopya webapp servisi mevcuttur. Çoğaltma işlemini uygulamak için terminale **“docker-compose up -d --scale webapp=3”** komutu yazılmıştır. Aşağıdaki uyarı metinleriyle karşılaşılmıştır.

```
C:\Users\m.koyunoglu\docker-static-website>docker-compose up -d --scale webapp=3
time="2025-03-23T13:55:11+03:00" level=warning msg="C:\Users\m.koyunoglu\docker-static-website\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
WARNING: The "webapp" service is using the custom container name "webapp". Docker requires each container to have a unique name. Remove the custom name to scale the service.
```

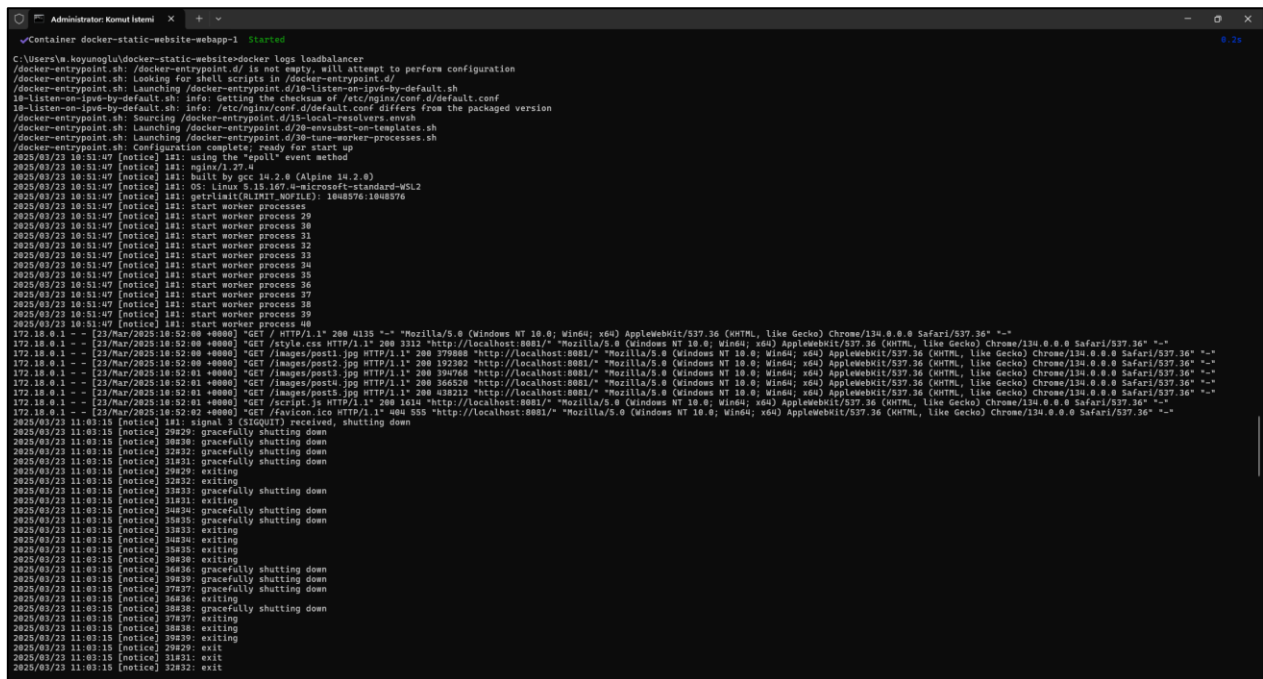
Araştırmalar sonucunda **Docker Compose’un yeni sürümlerinin “version: ‘3.8’”** satırına ihtiyaç **duymadığı öğrenilmiştir**. İlk olarak bu satır **“docker-compose.yml”** dosyasından kaldırılmıştır. Devamında yazan uyarının ise **standart olarak webapp servisine container_name tanımlandığı için çıktığı anlaşılmıştır**. **“container_name:webapp”** satırı dosyadan kaldırılmıştır. Devamında ilgili satır terminale tekrar girilmiş ve aşağıdaki sonuca ulaşılmıştır.

```
C:\Users\m.koyunoglu\docker-static-website>docker-compose up -d --scale webapp=3
[+] Running 5/5
 ✓ Container loadbalancer      Started
 ✓ Container docker-static-website-webapp-3 Started
 ✓ Container webapp            Recreated
 ✓ Container docker-static-website-webapp-2 Started
 ✓ Container docker-static-website-webapp-1 Started
```

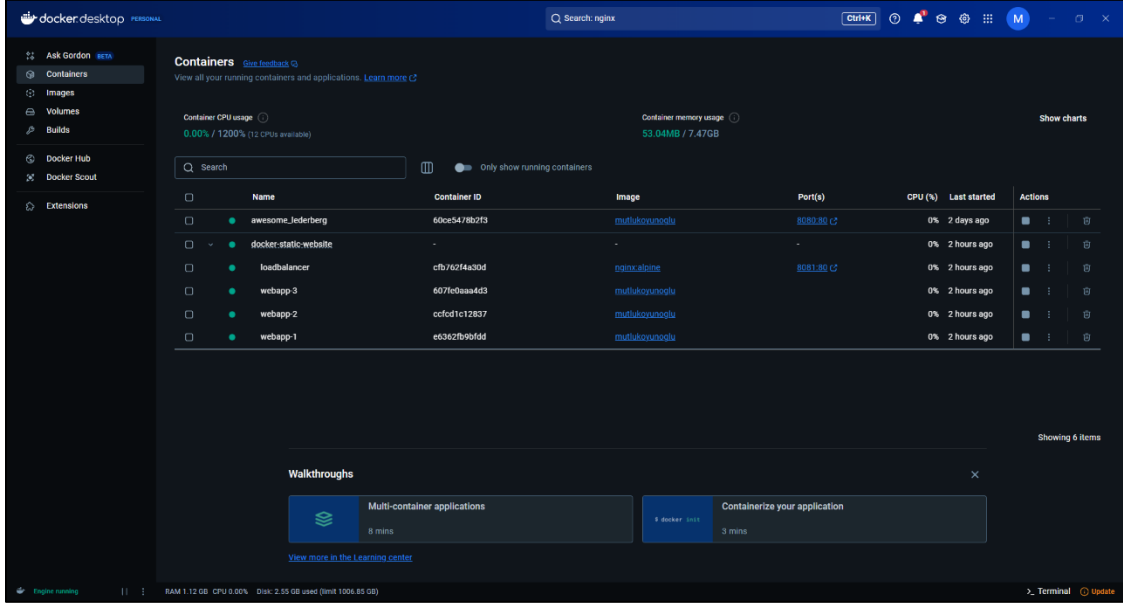
Load Balancer üzerinden siteye erişim test edilmesi için **“localhost:8081”** linkine gidilip yapılan geliştirmelerin durumu kontrol edilmiştir. Herhangi bir sorun görüntülenmemiştir.



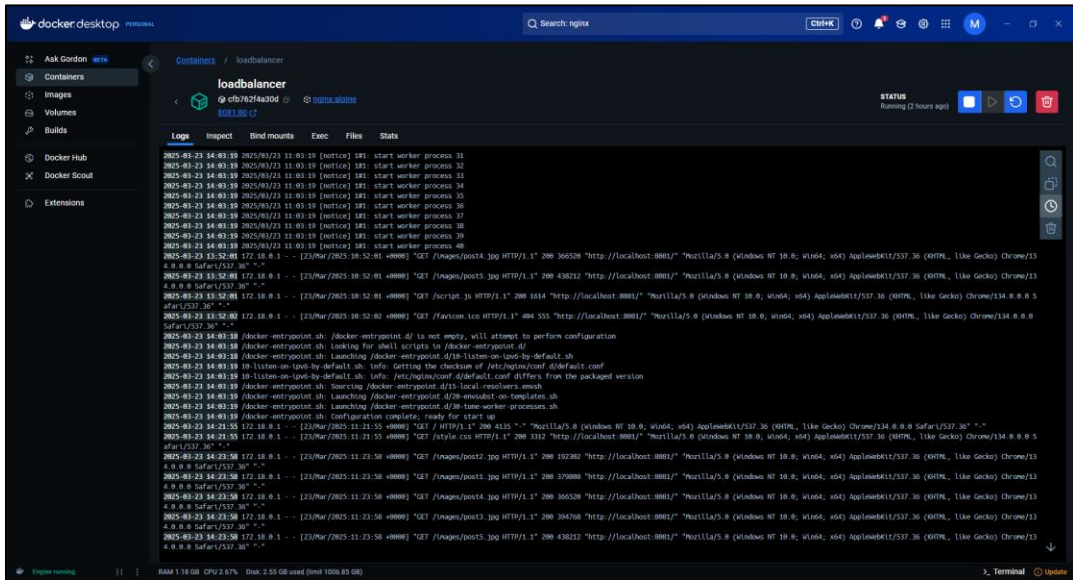
Terminal üzerinden **loadbalancer logları** da kontrol edilmiştir.



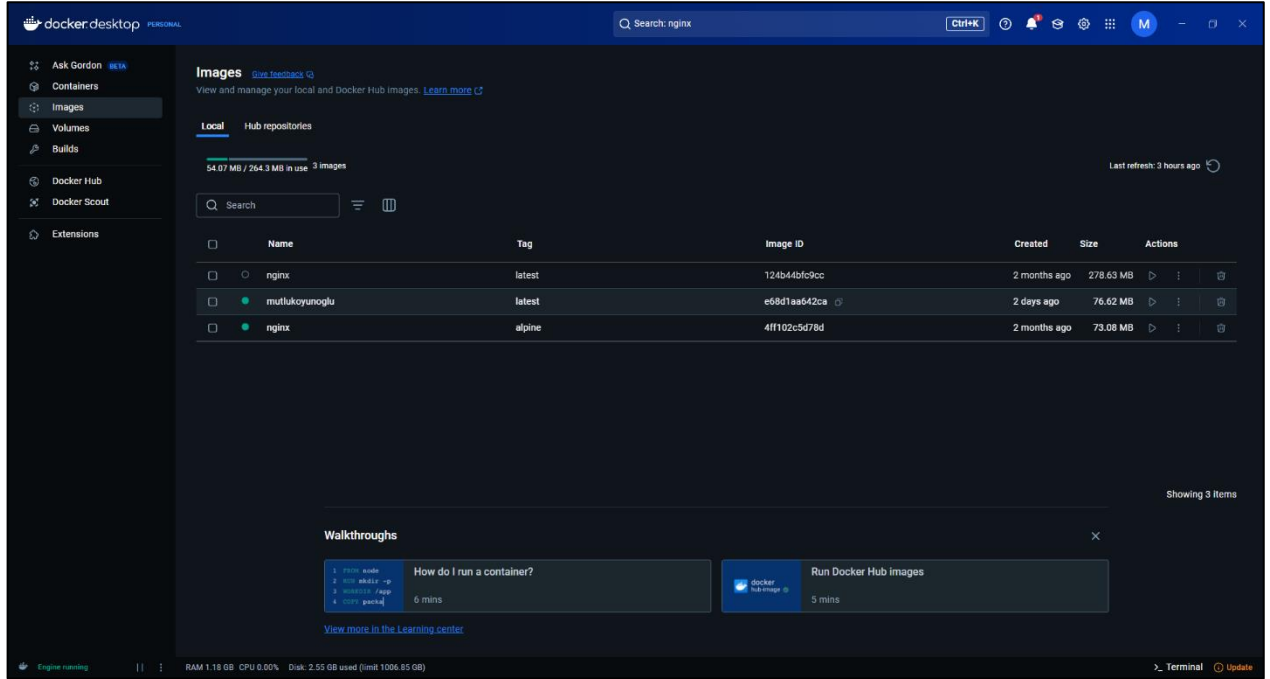
Proje yapılırken tüm komutlar **terminal** üzerinden gerçekleştirilmiştir. **Doğrudan Docker Desktop'un grafik arayüzü kullanılmamıştır.** Ancak, Docker Desktop sisteminde çalışıyor ve Docker Engine'i kontrol ediyor. Yani, terminal komutlarıyla imaj oluşturma, container çalıştırma, compose komutları gibi işlemler; Docker Desktop'un sağladığı altyapıyı kullandığı için doğrudan uygulamanın GUI arayüzüne ihtiyaç duyulmamıştır. Böylece, Docker Desktop arka planda çalışıyor olsa bile işlemler terminal üzerinden halledilmiştir. Ancak Desktop uygulaması üzerinden ekran görüntüleri de aşağıya eklenmiştir.



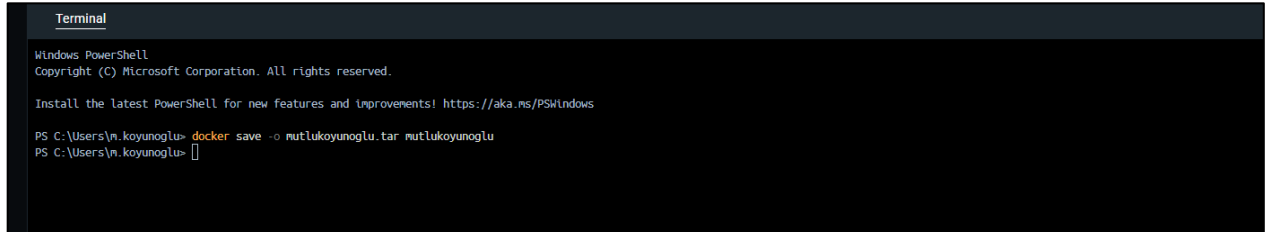
Üstteki fotoğrafta, terminalde oluşturulmuş konteynerleri (ilk oluşturulan 8080:80 portuna bağlı olan, sonrasında ki 8081:80'e bağlı olan loadbalancer ve webapplerin replicaları) görüntülenebilir. Her bir konteynerin üzerine tıklayarak detayları, logları ve kaynak kullanımı aşağıdaki gibi incelenebilir.



Images bölümünde, oluşturulan Docker image görüntülenebilir.



Docker Terminal üzerinden “**docker save -o mutlukoyunoglu.tar mutlukoyunoglu**” komutu ile imajı bir tar dosyası olarak paylaşılan belgeler arasına eklenmiştir.



Sonuçlar ve Öğrenilenler

Bu proje sayesinde, lisans eğitimim süresince öğrendiğim ancak uzun zamandır pratik yapmadığım web geliştirme bilgilerimi (.html, .css, .js) yeniden güncelleme fırsatı buldum. Öncelikle basit bir statik web sitesi oluşturarak HTML, CSS ve JavaScript kullanımı üzerinden temel tasarım ve içerik düzenleme becerilerimi tekrar gözden geçirdim. Ardından ilk kez Docker teknolojileriyle çalışmaya başladım ve bu statik web sitesini konteynerize etme sürecine odaklandım. Dockerfile oluşturmak, imaj üretmek, container başlatmak, ölçeklendirme işlemleri yapmak ve yük dengeleme mekanizmalarını adım adım uygulama şansı elde ettim. Ayrıca Docker Compose ile aynı imajın birden fazla kopyasını (replica) başlatarak, Nginx tabanlı bir reverse proxy aracılığıyla gelen isteklerin dağıtımını test ettim. Tüm bu süreç, container tabanlı uygulama dağıtımının sağladığı izole çalışma ortamı, hızlı kurulum ve ölçeklenebilirlik gibi avantajları anlamamı sağladı. Proje, modern yazılım dağıtım yöntemlerini pratikte deneyimlemek açısından benim için oldukça öğretici bir çalışma oldu.