

YAVUZLAR WEB GÜVENLİĞİ & YAZILIM TAKIMI

DOCKER EĞİTİMİ ÖDEV RAPORU

1. Docker Komutları

```
leidorf@debian: ~  
File Edit View Search Terminal Help  
leidorf@debian:~$ docker search alpine  
NAME                DESCRIPTION                STARS    OFFICIAL  
alpine              A minimal Docker image based on Alpine Linux... 10941    [OK]  
alpinelinux/docker-cli Simple and lightweight Alpine Linux image wi... 11  
alpinelinux/alpine-gitlab-ci Build Alpine Linux packages with Gitlab CI 3  
alpinelinux/gitlab-runner-helper Helper image container gitlab-runner-helper ... 7  
alpinelinux/rsyncd 2  
alpinelinux/unbound 13  
alpinelinux/alpine-drone-ci Build Alpine Linux packages with drone CI 0  
alpinelinux/docker-alpine 0  
alpinelinux/ansible Ansible in docker 21  
alpinelinux/gitlab-runner Alpine Linux gitlab-runner (supports more ar... 7  
grafana/alpine Alpine Linux with ca-certificates package in... 7  
alpinelinux/docker-compose docker-compose image based on Alpine Linux 2  
alpinelinux/apkbuild-lint-tools Tools for linting APKBUILD files in a CI env... 0  
bellsoft/liberica-openjdk-alpine Liberica is a 100% open-source Java implemen... 58  
alpinelinux/darkhttpd 2  
alpinelinux/golang Build container for golang based on Alpine L... 3  
alpinelinux/alpine-docker-gitlab Gitlab running on Alpine Linux 0  
alpinelinux/build-base Base image suitable for building packages wi... 0  
alpinelinux/alpine-www The Alpine Linux public website (www.alpinel... 0  
bellsoft/liberica-openjre-alpine Liberica is a 100% open-source Java implemen... 22  
alpinelinux/docker-abuild Dockerised abuild 0  
bellsoft/liberica-openjdk-alpine-musl Liberica is a 100% open-source Java implemen... 24  
balenalib/amd64-alpine-node This image is part of the balena.io base ima... 2  
alpinelinux/mqtt-exec 0  
alpinelinux/netbox dockerized netbox 0  
leidorf@debian:~$ docker |
```

Şekil 1. Docker Search Komutu Kullanımı

“*docker search alpine*” komutu ile Docker Hub üzerinde “*alpine*” Image’ini aramamızı sağlar.

```
leidorf@debian: ~  
File Edit View Search Terminal Help  
leidorf@debian:~$ docker pull alpine  
Using default tag: latest  
latest: Pulling from library/alpine  
c6a83fedfae6: Pull complete  
Digest: sha256:0a4eaa0eefcf5f8c050e5bba433f58c052be7587ee8af3e8b3910ef9ab5f9f5  
Status: Downloaded newer image for alpine:latest  
docker.io/library/alpine:latest  
  
What's Next?  
1. Sign in to your Docker account → docker login  
2. View a summary of image vulnerabilities and recommendations → docker scout quickview alpine  
leidorf@debian:~$
```

Şekil 2. Docker Pull Komutu Kullanımı

“*docker pull alpine*” komutu ile Docker Hub üzerinden “*alpine*” adlı Image’ı indirmemizi sağlar.

```
leidorf@debian: ~  
File Edit View Search Terminal Help  
leidorf@debian:~$ docker network ls  
NETWORK ID          NAME                DRIVER            SCOPE  
c42a9aa36b07        bridge             bridge            local  
680fdf422afe        host               host              local  
91b986015d9e        none              null              local  
leidorf@debian:~$
```

Şekil 3. Docker Network Komutu Kullanımı

“*docker network ls*” komutu ile Docker üzerinde oluşturulan tüm ağları listelememizi sağlar.

```
leidorf@debian: ~  
File Edit View Search Terminal Help  
leidorf@debian:~$ docker images  
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE  
deneme        latest    7337d58828fa   7 hours ago    471MB  
alpine        latest    324bc02ae123   47 hours ago   7.8MB  
nginx         latest    a72860cb95fd   4 weeks ago    188MB  
postgres      alpine    525e7b9d4e46   2 months ago   248MB  
sonarqube     latest    b728f044f72f   2 months ago   787MB  
hello-world   latest    d2c94e258dcb   14 months ago  13.3kB  
leidorf@debian:~$
```

Şekil 4. Docker Images Komutu Kullanımı

“*docker images*” komutu ile Docker üzerinden ana makinemize yüklenmiş tüm Docker Image’leri listeler.

```
leidorf@debian: ~  
File Edit View Search Terminal Help  
leidorf@debian:~$ docker run --name Yavuzlar alpine  
leidorf@debian:~$ docker run --name Docker-test hello-world  
Hello from Docker!  
This message shows that your installation appears to be working correctly.  
  
To generate this message, Docker took the following steps:  
1. The Docker client contacted the Docker daemon.  
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
   (amd64)  
3. The Docker daemon created a new container from that image which runs the  
   executable that produces the output you are currently reading.  
4. The Docker daemon streamed that output to the Docker client, which sent it  
   to your terminal.  
  
To try something more ambitious, you can run an Ubuntu container with:  
$ docker run -it ubuntu bash  
  
Share images, automate workflows, and more with a free Docker ID:  
https://hub.docker.com/  
  
For more examples and ideas, visit:  
https://docs.docker.com/get-started/
```

Şekil 5. Docker Run Komutu Kullanımı

“*docker run alpine*” komutu ile yüklediğimiz Docker Image’i çalıştırabiliriz. “*--name Yavuzlar*” komutu ile “*alpine*” Image’i ile çalışacak Docker Container’ının ismini belirtebiliriz.

```
leidorf@debian: ~  
File Edit View Search Terminal Help  
leidorf@debian:~$ docker run --rm --name Yavuzlar -p 8080:80 nginx  
/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  
2024/07/24 21:56:14 [notice] 1#1: using the "epoll" event method  
2024/07/24 21:56:14 [notice] 1#1: nginx/1.27.0  
2024/07/24 21:56:14 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)  
2024/07/24 21:56:14 [notice] 1#1: OS: Linux 6.6.26-linuxkit  
2024/07/24 21:56:14 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576  
2024/07/24 21:56:14 [notice] 1#1: start worker processes  
2024/07/24 21:56:14 [notice] 1#1: start worker process 29  
2024/07/24 21:56:14 [notice] 1#1: start worker process 30  
2024/07/24 21:56:14 [notice] 1#1: start worker process 31  
2024/07/24 21:56:14 [notice] 1#1: start worker process 32  
2024/07/24 21:56:14 [notice] 1#1: start worker process 33  
2024/07/24 21:56:14 [notice] 1#1: start worker process 34  
2024/07/24 21:56:14 [notice] 1#1: start worker process 35  
2024/07/24 21:56:14 [notice] 1#1: start worker process 36  
2024/07/24 21:56:20 [notice] 1#1: signal 28 (SIGWINCH) received  
2024/07/24 21:56:20 [notice] 1#1: signal 28 (SIGWINCH) received  
2024/07/24 21:56:20 [notice] 1#1: signal 28 (SIGWINCH) received
```

Şekil 6. Docker Run Komutu Kullanımı

Burada kullanılan komutlar şu şekilde çalışır: “*--rm*” Docker Container’ının çalışması durduğunda otomatik olarak kendini siler. “*-p 8080:80*” komutu ana makinenin 8080 portunu Docker Container’ının 80 portuna yönlendirir.

```
leidorf@debian: ~  
File Edit View Search Terminal Help  
leidorf@debian:~$ docker ps -a  
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                NAMES  
52d2425c3746   hello-world    "/hello"                About a minute ago    Exited (0) About a minute ago      
1aa416ba1812   alpine        "/bin/sh"               2 minutes ago    Exited (0) 2 minutes ago      
062bc31d04ec   alpine        "--name Yavuzlar"       2 minutes ago    Created                        thirsty_bose  
68e97ab8e573   sonarqube     "/opt/sonarqube/dock..." 2 months ago    Exited (255) 8 hours ago    0.0.0.0:9000->9000/tcp    sonarqube  
6c04c5085d44   postgres:alpine "docker-entrypoint.s..." 2 months ago    Exited (0) 2 months ago    sonarqube-db  
leidorf@debian:~$ docker stop Yavuzlar  
Yavuzlar  
leidorf@debian:~$ docker rm Yavuzlar  
Yavuzlar  
leidorf@debian:~$ docker ps -a  
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                NAMES  
52d2425c3746   hello-world    "/hello"                2 minutes ago    Exited (0) 2 minutes ago      
062bc31d04ec   alpine        "--name Yavuzlar"       2 minutes ago    Created                        thirsty_bose  
68e97ab8e573   sonarqube     "/opt/sonarqube/dock..." 2 months ago    Exited (255) 8 hours ago    0.0.0.0:9000->9000/tcp    sonarqube  
6c04c5085d44   postgres:alpine "docker-entrypoint.s..." 2 months ago    Exited (0) 2 months ago    sonarqube-db  
leidorf@debian:~$
```

Şekil 7. Docker Listeleme Komutu Kullanımı

"*docker ps -a*" ile mevcut durumda çalışan Docker Container'larını listeler.

2. Dockerfile ve Docker Compose Dosya Komutları

2.1. Dockerfile Dosya Komutları

"*FROM php:7.4-apache*" komutu, Docker Container'ının hangi Docker Image'i temel alacağını belirtir. Burada temel alınan Image "php:7.4-apache" olarak temel alınmış.

"*WORKDIR /var/www/html*" Docker Container'ının çalışma dizinini belirtir. Burada Docker'ın çalışma dizini "/var/www/html" dizini olarak belirtilmiştir.

"*COPY ./app .*" ana makineden, Docker Container içerisine dosya kopyalanmasını sağlar. Burada ana makinedeki "./app" klasörü Docker Container içerisindeki "." dizinine, yani Docker Container'ının temel dizinine kopyalar.

"*RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf*" komutu Docker Image üzerinde çalıştırılacak komutu belirtir. Burada "*echo*" aracılığıyla Docker Image içerisindeki

"*/etc/apache2/apache2.conf*" dosyasına ServerName, "*localhost*" olarak belirtilmiştir.

"*RUN apt-get update*" Docker Image üzerinde apt-get update komutu ile güncelleme sağlar.

"*RUN docker-php-ext-install pdo pdo_mysql*" Docker Image için PHP'nin PDO ve PDO MYSQL uzantıları kurar.

2.2. Docker Compose Dosya Komutları

"*version*" kısmı dosyanın Docker Compose versiyonunu belirtir.

"*services*" kısmı oluşturulacak Docker Container'ları tanımlar.

"*app*" web uygulamasının oluşturulacak Docker Container adıdır.

"*build*" Dockerfile kullanarak, Docker Container oluşturur.

"*depends_on*" kısmı bağımlılık olan Docker Container'ları belirtir. Burada bağımlılık olarak "*app*" Docker Container'ı belirtilmiştir.

"*ports*" ana makinenin 80 portunu Docker Container'ının 80 portuna yönlendirir.

"*networks*" servisin hangi ağda çalışacağını belirtir.

"*image*" kullanılacak olan Docker Image'ı belirtir.

"*environment*" ortam değişkenlerini tanımlar.

"*volumes*" verileri depolamak için kullanılan hacmi tanımlar.