



Nastavni predmet	KONFIGURIRANJE RAČUNALNIH MREŽA I SERVISA
Naslov jedinice	Vježba 14: Konfiguracija i upravljanje Docker kontejnerima na Linux Mintu

CILJ VJEŽBE

Učenici će naučiti kako instalirati Docker infrastrukturu na Linux Mintu, pokrenuti osnovne kontejnere (ng-inx, Apache), konfigurirati mrežu za kontejnere i pristupiti servisima pokrenutim u kontejnerima s Windows 10 klijenta i/ili Linux Mint klijenta. Također će naučiti kako kreirati vlastiti Dockerfile odnosno kontejner te povezati više kontejnera u mrežu.

Potrebni resursi:

1. Linux Mint server:

- Instaliran i pokrenut u VMware Workstation Pro.
- Konfiguriran s NAT mrežom.
- Pristup internetu za preuzimanje Docker paketa.

2. Windows 10 klijent:

- Koristi se za pristup web serverima pokrenutim u kontejnerima.

3. Linux Mint klijent (opcionalno):

- Može poslužiti kao dodatni čvor za testiranje mrežne komunikacije između kontejnera.

IZVOĐENJE VJEŽBE

Pratite oprezno sve korake, pokažite profesoru izvedeno prema koracima u nastavku.

Koraci za izvođenje vježbe:

1. Inicijalne postavke

- 2) Pokrenuti VMware Workstation Pro
- 3) Postaviti snapshot u VMware-u za poslužitelj (Linux Mint) na **cleanInstall** kako bi započeli čistu konfiguraciju SNMP servisa
- 4) Pokrenuti Poslužitelj (Linux Mint)

User : **osboxes.org**

Password : **osboxes.org**

- 5) Na Poslužitelju (Linux Mint) u Network adapter postavkama odaberite **NAT** kako bi mogli pristupiti internetu kod instalacije Linux Mint paketa nadogradnje.
- 6) Na Poslužitelju (Linux Mint) postavite tipkovnicu na **Croatia (Layout)** zbog prilagodbe tipkovnice hrvatskoj verziji (ostale tipkovnice možete obrisati).

2. Ažurirajte repozitorije i instalirajte potrebne pakete:

NAPOMENA: U ovom dijelu vježbe pripremit ćemo Docker infrastrukturu koja će se nalaziti na Linux Mint poslužitelju i predstavlja platformu za prihvat kontejnera koji su mala nezavisna računala.

```
$ sudo apt-get update
```

```
$ sudo apt-get install apt-transport-https ca-certificates curl  
software-properties-common -y //sve u jednom redu
```

1) Dodajte Docker službeni GPG ključ:

GPG (GNU Privacy Guard) je alat za kriptografsku zaštitu podataka. Koristi se za potvrđivanje autentičnosti i integriteta softverskih paketa.

```
$ sudo install -m 0755 -d /etc/apt/keyrings
```

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo  
gpg --dearmor -o /etc/apt/keyrings/docker.gpg //sve u jednom redu
```

```
$ sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

2) Dodajte Docker repozitorij za Ubuntu 22.04 (Jammy Jellyfish):

NAPOMENA: Kada dodate Docker repozitorij u svoj sustav, možete koristiti paketni menadžer (npr. apt za Ubuntu/Debian ili yum za CentOS) za instalaciju Dockera. Bez repozitorija, morali biste ručno preuzimati i instalirati Docker pakete, što je manje praktično.

NAPOMENA: Ova linija sadrže više naredbi stoga treba biti pravilno unesena u Terminal za egzekuciju. „\“ predstavlja prelazak u drugi red, sve do „\“ je u jednom redu.

```
$ echo \  
"deb [arch=$(dpkg --print-architecture) signed-  
by=/etc/apt/keyrings/docker.gpg]  
https://download.docker.com/linux/ubuntu \  
jammy stable" | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

(pravilna primjena komande neće izazvati odgovor)

3) Ažurirajte listu paketa i instalirajte Docker:

```
$ sudo apt-get update  
$ sudo apt-get install docker-ce docker-ce-cli containerd.io  
docker-buildx-plugin docker-compose-plugin -y //sve u jednom redu
```

4) Provjerite instalaciju:

```
$ sudo docker --version
```

Očekivani odgovor oblika:

```
Docker version 28.0.0, build f9ced58
```

5) Dodajte korisnika u Docker grupu (kako biste izbjegli korištenje sudo za Docker naredbe):

```
$ sudo usermod -aG docker $USER  
$ newgrp docker
```

3. Pokretanje osnovnog Docker kontejnera (ng-inx)

NAPOMENA: U ovom dijelu vježbe instalirat ćemo kontejnere (računala) za web servise. Jedan kontejner će biti ng-inx web server, a drugi Apache web server. Također, kontejneri će imati pristup internetu, a putem vlastite mreže će moći komunicirati međusobno ako je potrebno.

Nginx (izgovara se "engine-x") je popularan web server i reverse proxy softver otvorenog koda. Razvijen je s fokusom na visoke performanse, stabilnost i nisku potrošnju resursa. Ng-inx se

široko koristi za posluživanje web stranica, ali i kao reverse proxy, load balancer, HTTP cache i za druge mrežne zadatke.

1) Preuzmite ng-inx Docker image:

```
$ docker pull nginx
```

2) Kreiranje nove mreže za kontejnere `my-network`:

```
$ docker network create my-network
```

Ovo stvara **Custom bridge** mrežu, ali s poboljšanim DNS-om za prepoznavanje imena kontejnera. Ima ugrađen **Docker DNS (razrješava imena kontejnera)**, što omogućava lakšu komunikaciju među kontejnerima.

3) Pokrenite ng-inx kontejner na novoj mreži `my-network`:

```
$ docker run -d --name my-nginx --network my-network -p 8080:80  
nginx //sve u jednom redu
```

Ova naredba pokreće ng-inx kontejner u pozadini (-d), mapira port 80 unutar kontejnera na port 8080 na hostu (-p 8080:80), daje mu naziv `my-nginx` i povezuje ga s `my-network` mrežom.

4) Provjerite pokrenute kontejnere:

```
$ docker ps
```

Trebali bi vidjeti `my-nginx` kontejner u listi.

5) Provjerite mrežne postavke kontejnera:

```
$ docker inspect my-nginx
```

Ova naredba prikazuje detalje o kontejneru, uključujući mrežne postavke.

4. Ažuriranje ping, nslookup i nano alata u ng-inx kontejneru:

1) Pristupite ng-inx kontejneru:

```
$ docker exec -it my-nginx /bin/bash
```

2) Ažurirajte paketnu bazu i instalirajte ping, dnsutils i nano:

```
$ apt-get update
```

```
$ apt-get install -y iputils-ping dnsutils
```

```
$ apt update && apt install nano -y
```

3) Provjerite instalaciju alata:

```
$ ping -c 4 google.com
```

```
$ nslookup google.com
```

Treballi bi dobiti ispravne odgovore.

5. Pokrenite Apache web server u kontejneru i testirajte komunikacije između kontejnera ng-inx i Apachija

1) Pokrenite drugi kontejner (npr. Apache) i povežite ga s mrežom my-network:

```
$ docker run -d --name my-apache --network my-network -p 8081:80  
httpd //sve u jednom redu
```

2) Testirajte komunikaciju između ng-inx i Apache kontejnera:

Pristupite my-nginx kontejneru:

```
$ docker exec -it my-nginx /bin/bash
```

Unutar kontejnera, pokušajte pingati my-apache kontejner:

```
$ ping my-apache
```

Ako je sve ispravno konfigurirano, dobit ćete ping odgovore.

6. Provjera pristupa web servisima ng-inx i Apache

NAPOMENA: Ovdje ćemo pristupiti našim servisima u kontejnerima, svakom servisu posebno.

Na Linux Mint serveru, otvorite **Firefox** web preglednik i unesite (kod pokretanja Firefox-a preskočite sve njegove setup postavke):

```
http://localhost:8080
```

Treballi biste vidjeti početnu stranicu ng-inx-a i ovakav odgovor:

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.

Commercial support is available at nginx.com.

Thank you for using nginx.

U drugoj kartici Firefox-a upišite:

<http://localhost:8081>

Trebali biste vidjeti početnu stranicu Apachija i ovakav odgovor:

It works!

Pokažite profesoru pristup web serverima u kontejnerima na Firefox-u.

7. Kreiranje vlastitog Dockerfilea

Što je Dockerfile?

Dockerfile je skripta koja sadrži naredbe za izgradnju Docker image-a. Svaka naredba u Dockerfile-u stvara novi sloj (layer) u Docker image-u. Dockerfile omogućuje automatizaciju procesa izgradnje image-a, što je korisno za konzistentno i ponovljivo stvaranje kontejnera.

1) Kreirajte direktorij za Dockerfile:

```
$ mkdir my-docker-app
```

```
$ cd my-docker-app
```

2) Kreiraj `index.html` s vlastitom porukom:

```
$ echo '<h1>Ja sam IME PREZIME, tehničar za računalstvo!</h1>' >
```

```
index.html
```

 //sve u jednom redu, IME PREZIME zamijeni svojim

3) Kreirajte Dockerfile:

```
$ nano Dockerfile
```

i dodajte sljedeći sadržaj:

```
# Koristi službenu Ubuntu sliku kao bazu
```

```
FROM ubuntu:latest
```

```
# Ažuriraj sustav i instaliraj Nginx
```

```
RUN apt-get update && apt-get install -y nginx
```

```
# Kopiraj prilagođeni index.html u ispravan direktorij
```

```
COPY index.html /var/www/html/index.html
```

```
# Otvori port 80
```

```
EXPOSE 80
```

```
# Pokreni Nginx u foreground modu
```

```
CMD ["nginx", "-g", "daemon off;"]
```

Spremite Dockerfile (Ctrl+X, Y, Enter).

4) Izgradite Docker image „ime“:

```
$ docker build -t my-ime-nginx .
```

Tako ste kreirali jedan vaš kontejner.

5) Pokrenite kontejner s vašim image-om:

```
$ docker run -d --name custom-nginx -p 8082:80 my-ime-nginx
```

6) Provjeri jel kontejner pokrenut:

```
$ docker ps
```

Ako vidiš custom-nginx u listi, znači da je sve u redu i kreirao si **Dokerfile**.

7) Pokreni Firefox i provjeri <http://localhost:8082>

Pokaži profesoru što si dobio.

8. Pristup ng-inx i Apachiju kroz Windows 10 klijent

1) Pronaći IP adresu Linux Mint servera:

```
$ ip a
```

 (pronaći IP adresu mrežnog sučelja ens33)

2) Provjeriti pristup iz Windowsa

Na Windows 10 klijentu otvori Command Prompt (cmd) i testiraj vezu:

```
$ ping <IPadresaLinuxMintServera>
```

Ako dobiješ odgovore, to znači da Windows klijent vidi Linux Mint server.

3) Pristupiti Nginx i Apache web serverima

Otvorite web preglednik (Chrome, Firefox ili Edge) i unesite sljedeće URL-ove:

Za ng-inx:

http://<IPadresaLinuxMintServera>:8080

Trebao bi se prikazati ng-inx početni ekran:

"Welcome to nginx!"

Za Apache:

http://<IPadresaLinuxMintServera>:8081

Trebao bi se prikazati Apache početni ekran:

"It works!"

POKAŽITE PROFI DA VAM STISNE PETICU! YUHUUUUUUU

Zaključak:

Ova vježba omogućuje učenici da steknu praktično iskustvo s Docker infrastrukturom, uključujući instalaciju Docker infrastrukture, pokretanje kontejnera, konfiguraciju mreže i kreiranje vlastitih Dockerfile-ova (kontejnera). Svi koraci se izvode na Linux Mint serveru, a pristup servisima se izvršava s Windows 10/Linux Mint klijenta.

9. Ako ste uspješno završili sve stavke vježbe dobit ćete odličnu ocjenu!

**Čestitamo na vašoj instalaciji kontejnera! Postavite računalo na
cleanInstall i Home.**