



Univerzitet u Zenici
Politehnički fakultet
2022/23



Operativni sistemi

Prva godina I. ciklus

Dokumentacija projekta

-POSTAVLJANJE PORTAINER OKRUŽENJA-

Članovi/ce tima:

- Esma Dizdarević
- Arduana Halilović
- Amra Drijenčić
- Melika Gluhić

Uvod

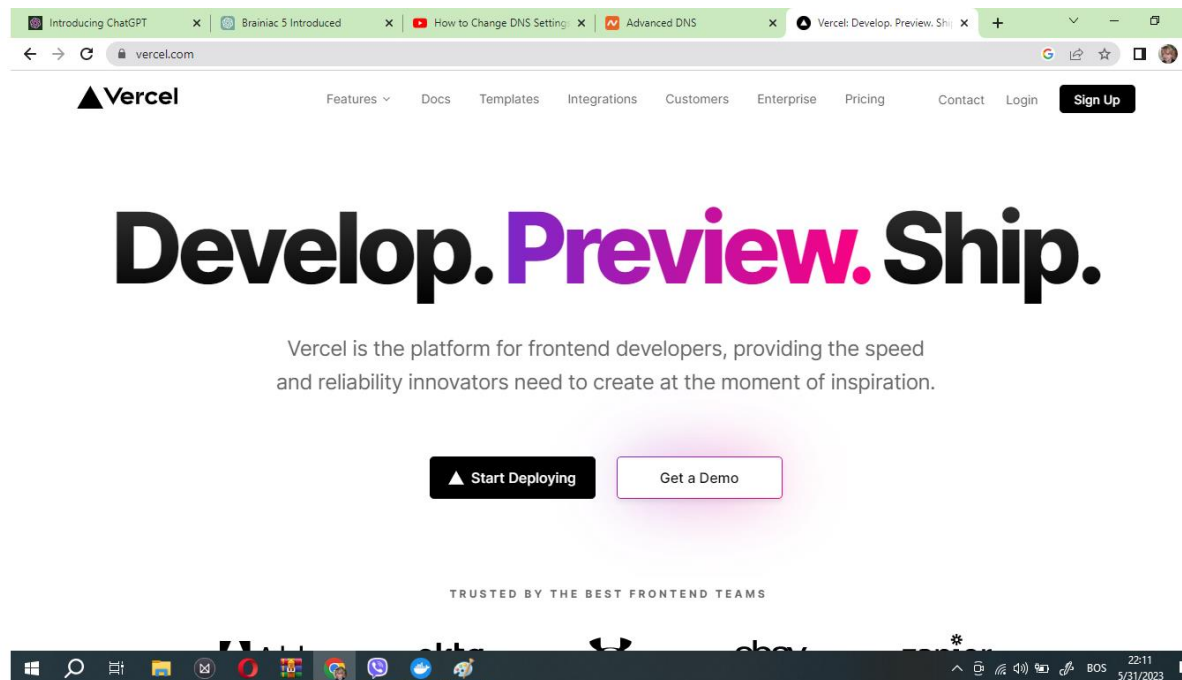
Postavljanje efikasnog okruženja za upravljanje kontejnerima je ključno za razvoj, implementaciju i održavanje modernih web aplikacija. Jedan od popularnih alata koji omogućava jednostavno upravljanje kontejnerima je Portainer. Portainer je intuitivan grafički korisnički interfejs (GUI) koji olakšava upravljanje Docker kontejnerima i pruža mogućnosti za pregled, nadzor i administraciju kontejnerskih okruženja.

Metodologija/Izrada/Koraci rada

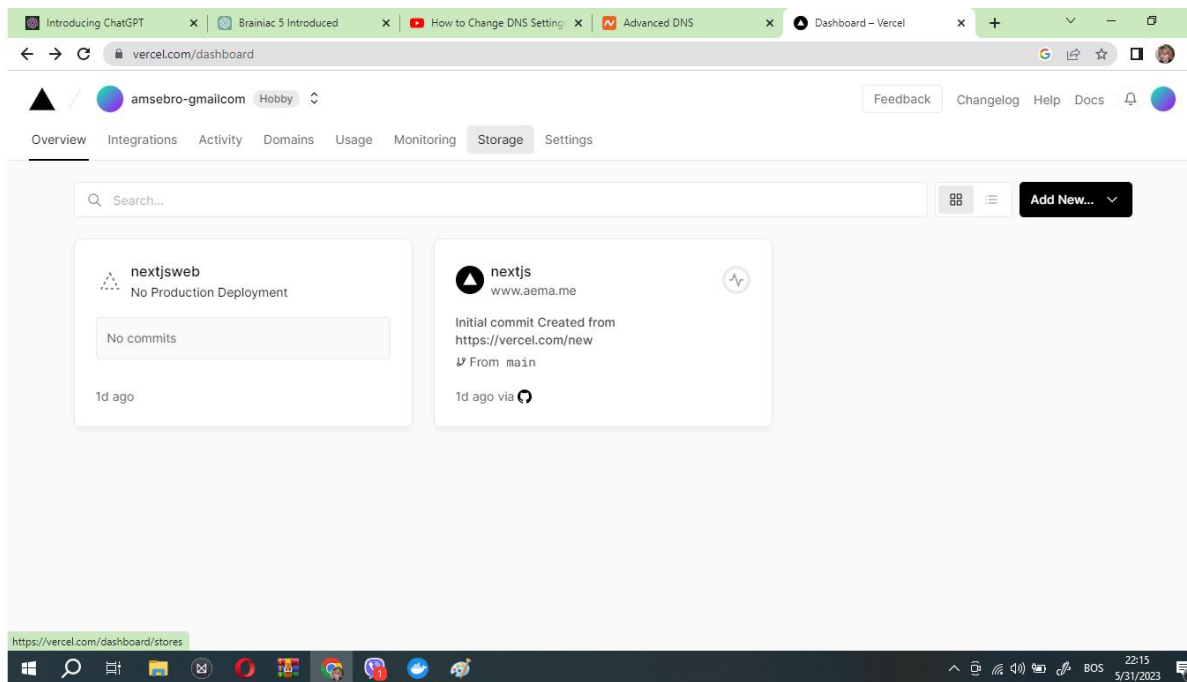
1. Vercel
2. Namecheap
3. Povezivanje templatea i domene
4. Instalacija Dockera i
5. Instalacija Portainera
6. Kreiranje docker slike

1. Vercel

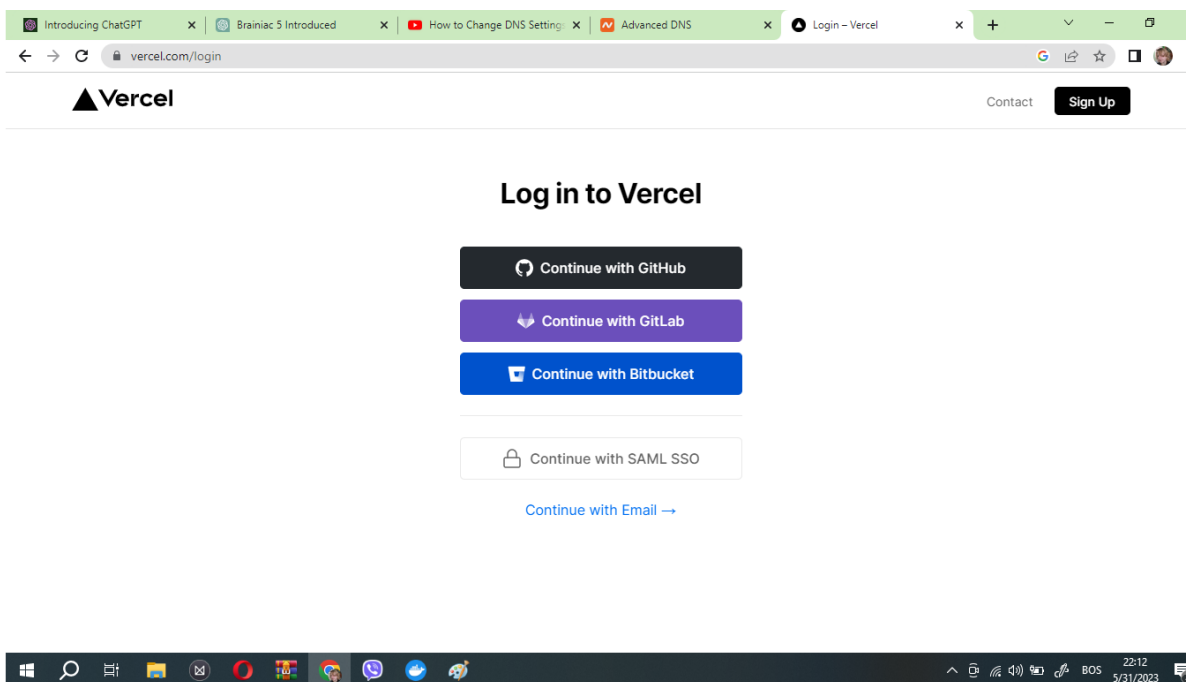
Vercel je platforma koja omogućuje brzo i jednostavno objavljivanje web stranica i aplikacija. Ona podržava razvojne timove i pojedince da izgrade, testiraju i distribuiraju svoje projekte na globalnoj mreži. Vercel pruža hosting uslugu koja automatizira postupak objavljivanja web stranica i aplikacija, omogućujući brzo prikazivanje projekta na internetu. Da bi se koristio Vercel, treba se prijaviti na njihovu platformu putem korisničkog računa. Nakon prijave, moguće je odabrati gotove template (predloške) koje Vercel nudi ili koristiti vlastiti dizajn. Template je već izrađena osnova koju je moguće prilagoditi svojim potrebama.



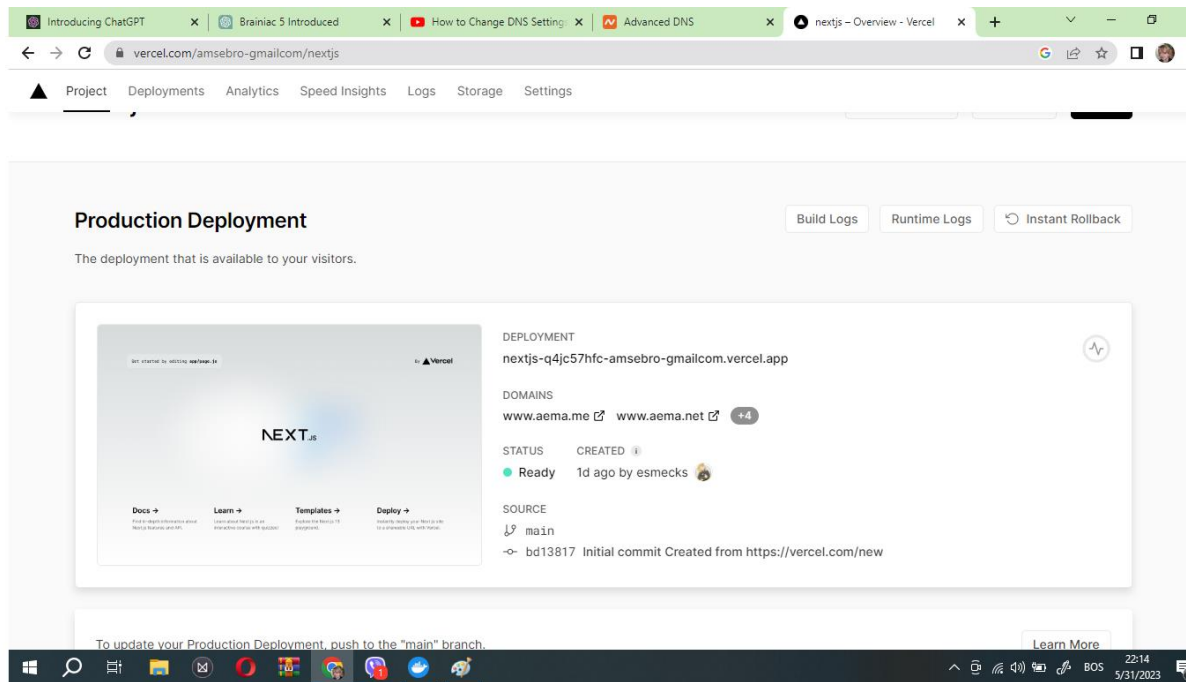
Slika 1 : Vercel početna



Slika 2 : Vercel prijava



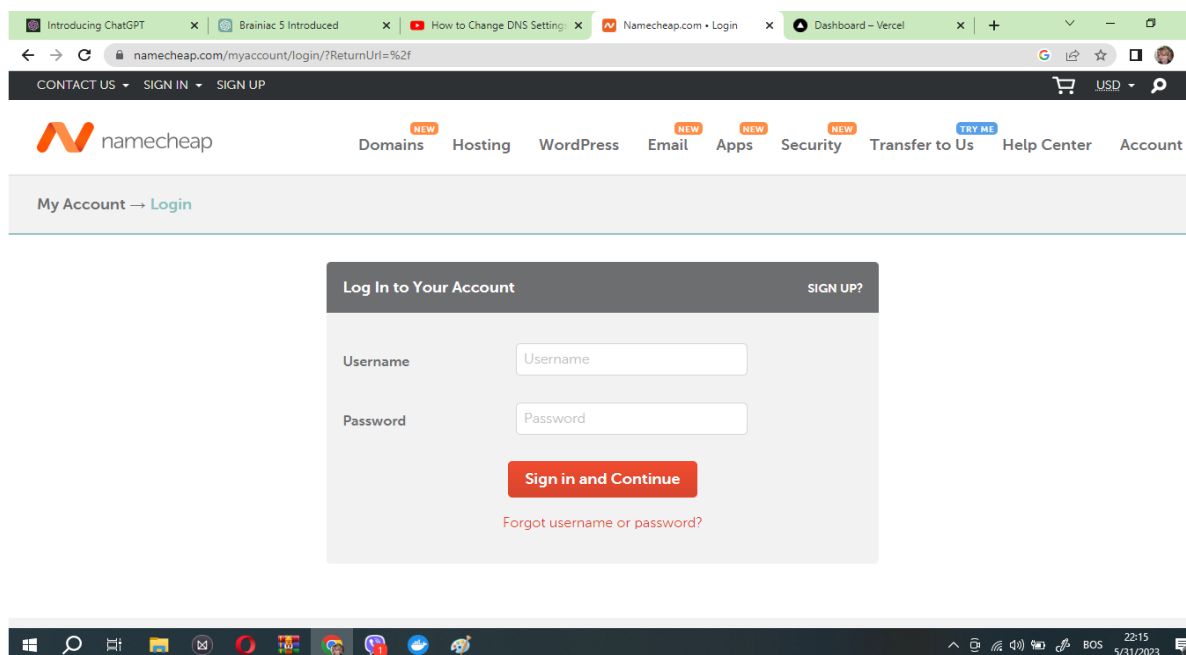
Slika 3 : Login Vercel



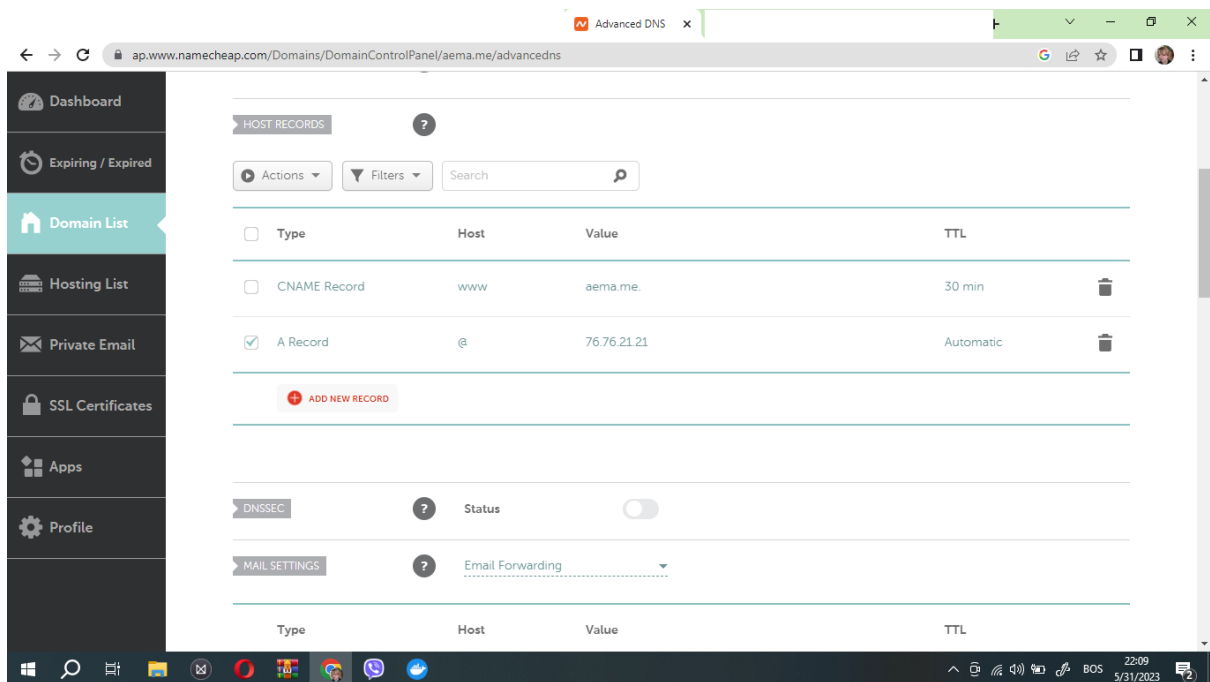
Slika 4 : Vercel produkcija

2. Namecheap

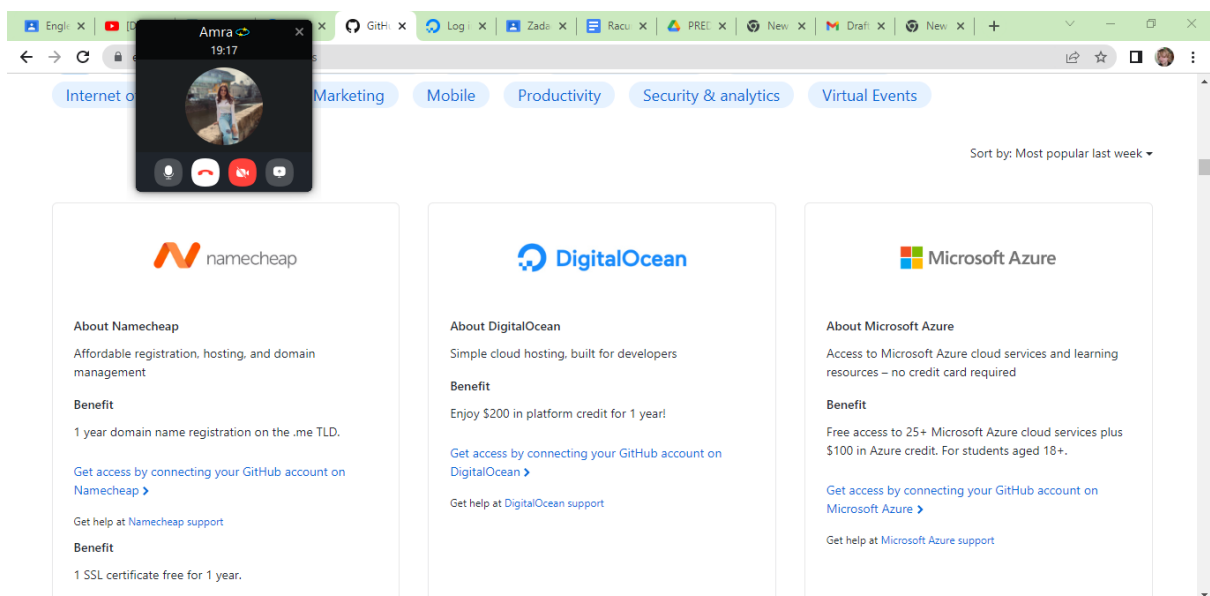
Namecheap je registracija domena i pružatelj hosting usluga. Za posjedovanje vlastite domene, potrebno je registrovati je kod pružatelja domenskih usluga poput Namecheap-a. Registracija domene pruža jedinstvenu web adresu (URL) za web stranicu ili aplikaciju. Na web stranici Namecheap-a moguće je provjeriti dostupnost željene domene i kupiti je ako je dostupna. Potrebno je izvršiti postupak registracije domene i konfigurisati je tako da se poveže s hosting providerom ili platformom poput Vercela.



Slika 5 : Namecheap početna



Slika 6 : Namecheap prijava



Slika 7 : Student paket

3. Povezivanje templatea i domene

Kada se odabere template na Vercelu i registruje domena na Namecheap-u, sljedeći korak je povezivanje templatea i domene. To znači da je potrebno konfigurirati postavke DNS-a (Domain Name System) kako bi se usmjerili posjetitelji s domene na template na Vercelu. U postavkama domene na Namecheap-u, treba promijeniti DNS zapise (kao što su A zapisi ili CNAME zapisi) kako bi usmjerili domenu na IP adresu ili adresu URL templatea na Vercelu. Kada se ovo konfigurira, posjetitelji koji upišu domenu u web pregledniku bit će preusmjereni na web stranicu koja se temelji na odabranom templateu.

4. Instalacija Dockera

Docker je platforma koja omogućuje pakiranje, distribuciju i izvršavanje aplikacija u kontejnerima. Instaliranje Dockera je preduvjet za korištenje Portainera. Docker se može instalirati na svom računar s operativnim sistemom koji podržava Docker. Nakon instalacije Dockera, može se koristiti Command Prompt ili terminal za pokretanje Portainera kao kontejnera.

5. Instalacija Portainera

1. Portainer koristi Docker API-je da bi komunicirao sa Docker Engine-om i upravljao kontejnerima, stoga je instalacija Dockera, koja je objašnjena u prethodnom dijelu, preduslov za instalaciju Portainera. **Svrha Portainera je lakše upravljanje Dockerom preko više uređaja, na način da se on lokalno postavi na uređaj zbog zaštite.**

Portainer se može postaviti na različite platforme, uključujući Windows i Linux

Postavljanje Portainera na Windows:

- Prvo je potrebno instalirati Docker za Windows. Moguće je preuzeti instalacijski program s web stranice Docker-a i slijediti upute za instalaciju.
- Nakon što je Docker instaliran, otvoriti PowerShell ili CMD prozor.
- Unesite sljedeću naredbu kako biste preuzeli Portainer sliku: `docker pull portainer/portainer`.
- Nakon što preuzimanje završi, pokrenite Portainer kontejner s ovom naredbom:
`docker run -d -p9000:9000 --name=portainer --restart=always -v/var/run/docker.sock:/var/run/docker.sock portainer/portainer`.
- Portainer će sada biti dostupan na <http://localhost:9000>. Otvoriti web preglednik i unijeti tu adresu kako bi pristupili Portaineru.

Postavljanje Portainera na Linux:

- Prvo je potrebno instalirati Docker na svoj Linux sistem. Postupak instalacije može se razlikovati ovisno o distribuciji Linuxa koja se koristi. Moguće je slijediti službeni dokumentaciju Docker-a za detaljnije upute.
- Nakon instalacije Dockera, otvoriti terminal i unijeti sljedeću naredbu za preuzimanje Portainer slike: `docker pull portainer/portainer`.
- Nakon preuzimanja, pokrenite Portainer kontejner s ovom naredbom: `docker run -d -p9000:9000 --name=portainer --restart=always -v/var/run/docker.sock:/var/run/docker.sock portainer/portainer`.
- Portainer će sada biti dostupan na <http://localhost:9000>. Otvoriti web preglednik i unijeti tu adresu kako bi pristupili Portaineru.

2. Kreiranje Docker volumena - Kreiranje Docker volumena je opcionalni korak, ali preporučuje se kako bismo sačuvali konfiguracijske podatke Portainera čak i ako kontejner bude izbrisan. Docker volumen je direktorij unutar Docker hosta koji se montira unutar

kontejnera. Kontejneri mogu pristupiti podacima u volumenu i izvršavati operacije čitanja i pisanja nad njima.



```
Command Prompt
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pc>docker volume create portainer_data
portainer_data
```

Slika 8 : Kreiranje volumena

Naredbom 'docker volume create portainer_data' prikazanom na slici iznad, kreirali smo volumen pod nazivom 'portainer_data'.

3. Pokretanje Portainer kontejnera - Koristimo naredbu u Command Promptu koja će preuzeti Portainer sliku iz Docker Huba i pokrenuti je kao kontejner s konfigurisanim vezama, postavkama portova i nazivom kontejnera.

Naredba koju koristimo je: `docker run -d -p 9000:9000 --name=portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer`.

`docker run`: Ova naredba pokreće Docker kontejner.

`-d`: Označava da se kontejner pokreće u pozadini (detached mode), što znači da će se izvršavanje nastaviti u pozadini nakon pokretanja.

`-p 9000:9000`: Mapira portove između hosta i kontejnera. U ovom slučaju, broj 9000 označava vanjski port na hostu, dok drugi broj 9000 označava port unutar kontejnera. To omogućava pristup Portainer sučelju preko port 9000 lokalnom računalu.

`--name=portainer`: Određuje naziv kontejnera. U ovom slučaju, naziv kontejnera je "portainer".

`--restart=always`: Postavlja način ponovnog pokretanja kontejnera u slučaju da se kontejner zaustavi. Vrijednost "always" označava da se kontejner uvijek ponovno pokreće nakon zaustavljanja.

`-v /var/run/docker.sock:/var/run/docker.sock`: Opcija između hosta i kontejnera. Omogućava Portaineru pristup upravljanju Docker kontejnerima na hostu.

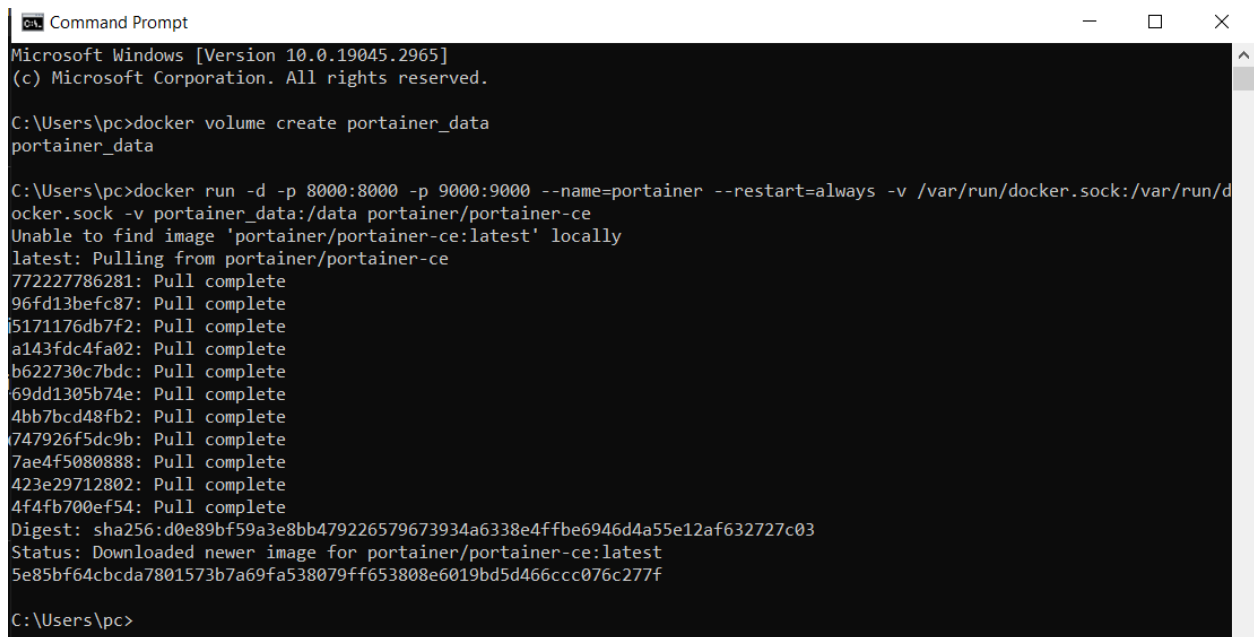
`-v portainer_data:/data`: Stvara vezu između nazvanog volumena "portainer_data" i direktorija unutar kontejnera /data. To omogućava kontejneru pristup i pohranu podataka u tom volumenu.

portainer/portainer: Ovo je naziv Docker slike koja će se koristiti za pokretanje kontejnera. U ovom slučaju, koristimo sliku "portainer/portainer" koja je službena slika Portainera iz Docker Huba.

Slika 9: Pokretanje Portainer kontejnera

Portainer će biti dostupan na adresi <http://localhost:9000> na lokalnom računalu

4. Nakon pokretanja, možemo pristupiti Portainer okruženju putem web preglednika na adresi <http://localhost:9000>. Prilikom prvog pristupa Portaineru, potrebno je kreirati



```
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pc>docker volume create portainer_data
portainer_data

C:\Users\pc>docker run -d -p 8000:8000 -p 9000:9000 --name=portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce
Unable to find image 'portainer/portainer-ce:latest' locally
latest: Pulling from portainer/portainer-ce
772227786281: Pull complete
96fd13befc87: Pull complete
5171176db7f2: Pull complete
a143fdc4fa02: Pull complete
b622730c7bdc: Pull complete
69dd1305b74e: Pull complete
4bb7bcd48fb2: Pull complete
747926f5dc9b: Pull complete
7ae4f5080888: Pull complete
423e29712802: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:d0e89bf59a3e8bb479226579673934a6338e4ffbe6946d4a55e12af632727c03
Status: Downloaded newer image for portainer/portainer-ce:latest
5e85bf64cbcd47801573b7a69fa538079ff653808e6019bd5d466ccc076c277f

C:\Users\pc>
```

administratorski račun, nakon čega možemo pristupiti Portainer okruženju i početi upravljati Docker kontejnerima.

6. Kreiranje Docker slike

Da bismo kreirali Docker sliku, koristimo niz komandi u CMD-u :

cd Desktop\moj-projekt - Premještamo se u folder na desktopu pod nazivom 'moj-projekt' u kojem će se klonirati github repozitorij.

git clone https://github.com/esmecks/nextjs - Naredba za kloniranje repozitorija u folder. Potrebno je instalirati git na racunar da bi naredba radila. Ova naredba će stvoriti mapu "nextjs" koja sadrži sadržaj repozitorija.

Slika 10: Kloniranje Github repozitorija

Zatim, ponovno otvorimo CMD ili terminal i navigiramo do mape "nextjs" u "moj-projekt" folderu.

git branch -a – Provjerava i ispisuje sve dostupne grane u repozitoriju.



```
Command Prompt
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pc>cd Desktop\moj-projekt

C:\Users\pc\Desktop\moj-projekt>git clone https://github.com/esmecks/nextjs
Cloning into 'nextjs'...
info: please complete authentication in your browser...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 18 (delta 0), reused 18 (delta 0), pack-reused 0
Receiving objects: 100% (18/18), 47.84 KiB | 569.00 KiB/s, done.

C:\Users\pc\Desktop\moj-projekt>
```

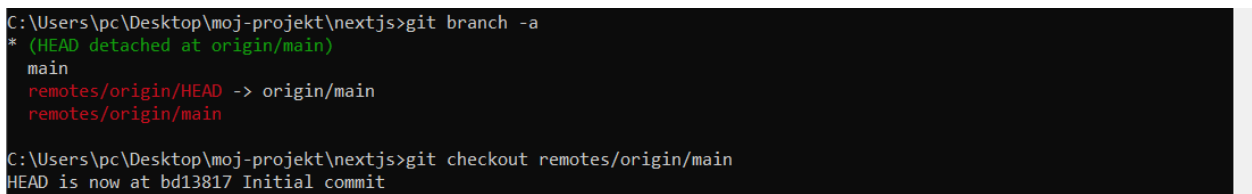
git checkout remotes/origin/main - Biramo granu na kojoj želimo raditi, u ovom slučaju na grani main koja je identična udaljenoj grani main na repozitoriju origin. Ova naredba omogućava pregled, izvršavanje promjena i commitova na odabranoj grani.

Slika 11: Iščitavanje dostupnih grana i biranje grane

echo. > Dockerfile - Stvaranje prazne datoteke pod nazivom Dockerfile.

Zatim, otvorimo Dockerfile sa Notepad i unutar pišemo:

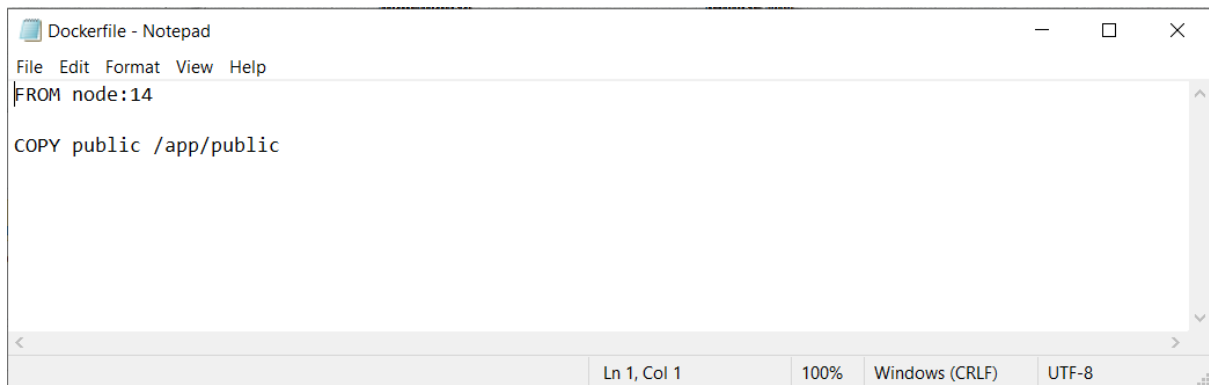
FROM node:14 - Ova linija označava da koristimo "node:14" kao osnovnu sliku za izgradnju Docker slike.



```
C:\Users\pc\Desktop\moj-projekt\nextjs>git branch -a
* (HEAD detached at origin/main)
  main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main

C:\Users\pc\Desktop\moj-projekt\nextjs>git checkout remotes/origin/main
HEAD is now at bd13817 Initial commit
```

COPY public /app/public - Naredba koja kopira datoteke i direktorije iz izvornog direktorija public u trenutni radni direktorij /app/public unutar Docker kontejnera.



Slika 12: Dockerfile – kopiranje datoteka u kontejner

docker build -t dokerslika . - Izgradnja docker slike pod nazivom dokerslika, pri čemu Docker mora biti otvoren da bi naredba radila.

```
C:\Users\pc\Desktop\moj-projekt\nextjs>docker build -t dokerslika .
[+] Building 1.1s (7/7) FINISHED
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 79B
=> [internal] load metadata for docker.io/library/node:14
=> [internal] load build context
=> => transferring context: 100B
=> [1/2] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa
=> CACHED [2/2] COPY public /app/public
=> exporting to image
=> => exporting layers
=> => writing image sha256:6509d00255bc73213f45fc853458e4fc57fb48b7366069b476ea46be3f2e7058
=> => naming to docker.io/library/dokerslika
```

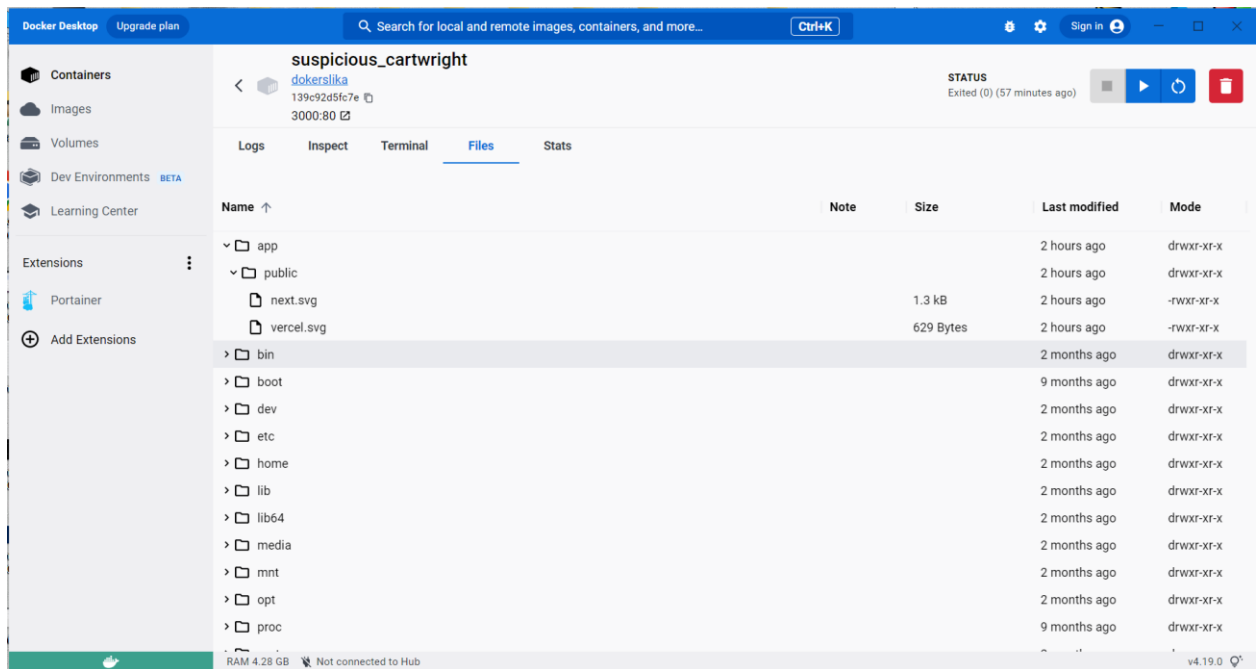
Slika 13: Izgradnja docker slike

docker run -d -p 3000:80 dokerslika - Naredba pokreće kontejner na temelju slike dokerslika s opcijama -d i -p 3000:80. Opcija -d označava da kontejner radi u pozadini (detached mode), dok opcija -p 3000:80 mapira host port 3000 na container port 80. To znači da će aplikacija koja se izvršava unutar kontejnera biti dostupna na host sistemu preko port-a 3000. '139c92d5fc7ec7757bc8d13d41be5c414352f723249110c221c6ac8c309e7ad5' je identifikator kontejnera (Container ID). Svaki put kada se kontejner pokrene, dodjeljuje mu se jedinstveni identifikator.

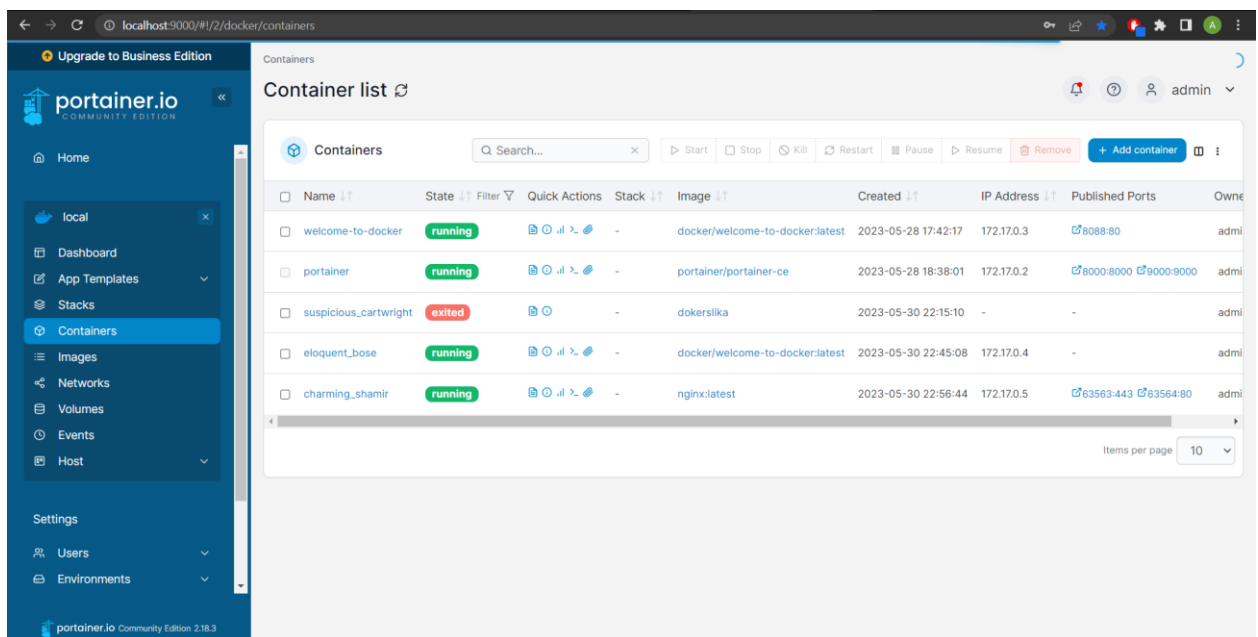
```
C:\Users\pc\Desktop\moj-projekt\nextjs>docker run -d -p 3000:80 dokerslika
139c92d5fc7ec7757bc8d13d41be5c414352f723249110c221c6ac8c309e7ad5
```

Slika 14: Pokretanje kreiranog kontejnera

Kreirana docker slika se sada nalazi u programu Docker Desktop sa direktorijem public i njegovim sadržajem.



Slika 15: Kreirana docker slika u Docker Desktop



Slika 16: Kreirana docker slika u Portaineru

Zaključak/Završna riječ/Zaključni sažetak

U projektu je korišten Vercel, popularna platforma za razvoj i hostovanje web aplikacija. Vercel je poslužio kao osnova za implementaciju koncepta. Za pristup i korištenje Vercel platforme, prvo je bila potrebna registracija i prijava putem korisničkih podataka. Nakon prijave, imali smo pristup širokom izboru predložaka (template) koje je Vercel pružio.

U procesu nabavke domene za projekat, tim je koristio Namecheap, popularnu platformu za registraciju domena. Kroz Namecheap smo istraživali dostupne domene i izvršili kupovinu odabrane domene.

Sljedeći korak je bio povezivanje te domene sa Vercel-om. To je postignuto kroz odgovarajuće konfiguracije DNS (Domain Name System) podešavanja. Kada je ta veza uspostavljena, dobijena je funkcionalna web stranicu koja je bila dostupna putem nabavljene domene.

Zatim je izvršena instalacija Docker-a i Portainer-a. Docker je platforma koja omogućava izolaciju i upravljanje aplikacijama u kontejnerima. Pored toga, dodan je Portainer, korisnički interfejs za upravljanje Docker okruženjem, kako bi imali intuitivan način za pregled i upravljanje kontejnerima i njihovim resursima.

Zadnji korak u projektu bilo je kreiranje Docker slike koristeći niz komandi u CMD terminalu.

Izjava o Autentičnosti Rada

Ovim putem, naš tim koji se sastoji od: Esme Dizdarević, Arduane Halilović, Amre Drijenčić i Melike Gluhić

Izjavljujemo da je ovaj projekat autentičan uz korištenje isključivo vlastitih izvora informacija i korištenjem odgovarajućih citata i referenci za sve dijelove koji su preuzeti iz drugih izvora.

Potvrđujemo:

- Da su svi dijelovi teksta koji su preuzeti iz drugih izvora citirani i referencirani u skladu s akademskim standardima.
- Da nije korišten nikakav oblik plagijata ili krađe intelektualnog vlasništva, te sam svjestan/na posljedica takvih djela.
- Da se slažemo da se ovaj rad može provjeriti na plagijate pomoću odgovarajućih alata za detekciju plagijata.

U slučaju da se utvrdi da je ovaj rad plagijat, svjestan/na sam da ću snositi odgovarajuće posljedice u skladu s pravilima o plagijatu i kršenju akademskih standarda na Politehničkom Fakultetu Univerziteta u Zenici.

Potpisi:

Datum: