



## **CS450 CLOUD COMPUTING**

### **PROJEKTNI ZADATAK:**

**Deploy aplikacija na lokalnom kladu računaru**

### **STUDENT:**

Dušan Stanković, 3611

[dusan.stankovic.3611@metropolitan.ac.rs](mailto:dusan.stankovic.3611@metropolitan.ac.rs)

U Nišu, 12.06.2023

## SADRŽAJ:

1 Uvod .....	4
2 Opšti podaci o poslovnom sistemu .....	4
2.1 Karakteristike sistema .....	4
2.1.1 Centralizovano skladištenje podataka .....	4
2.1.2 Deljenje fajlova .....	4
2.1.3 Remote deploy .....	5
2.1.4 Poboljšana produktivnost .....	5
2.1.5 Bezbednost podataka .....	5
3 Analiza postojećeg informacionog sistema organizacije .....	5
3.1 Hardver postojećeg sistema .....	6
3.2 Softver postojećeg sistema .....	6
4 Uporedni pregled tehnologija .....	7
4.1 Trenutni sistem .....	7
4.1.1 Skladištenje projekata .....	7
4.1.2 Deljenje projekata .....	7
4.1.3 Resursi .....	7
4.1.4 Slanje dokumenata .....	7
4.1.5 Podešavanje okruženja .....	7
4.2 sistem koji će se implemenitrati .....	7
4.2.1 Skladištenje projekata .....	7
4.2.2 Deljenje projekata .....	7
4.2.3 Resursi .....	7
4.2.4 Slanje dokumenata .....	7
4.2.5 Podešavanje okruženja .....	7
5 Predlog rešenja za unapređenje poslovnog sistema .....	8
5.1 Cloud skladištenje projekata .....	8
5.2 Centralizovani pristup projektima .....	8
5.3 Izvršavanje projekata na centralnom serveru .....	8
5.4 Centralizovano skladištenje i deljenje dokumenata .....	9
5.5 Podešavanje okruženja .....	9
6 Implementacija rešenja za unapređenje poslovnog sistema .....	9
6.1 Instalacija i podešavanje ubuntu server 22.04.2 LTS .....	9

6.2 Instalacija i podešavanje softvera za cloud .....	
6.3 deploy backend i frontend aplikacija .....	26
7 Analiza i prikaz postignutih rezultata .....	33
8 Zaključak .....	34
9 Literatura .....	35

## SPISAK SLIKA:

Slika 1: Logotip BMU .....	<b>Error! Bookmark not defined.</b>
----------------------------	-------------------------------------

# 1 UVOD

U današnjem poslovnom okruženju, mnogi samostalni developeri se suočavaju sa izazovom korišćenja više računara tokom svog radnog dana. Slična situacija javlja se i kod mene u firmi, gde koristim i desktop i laptop računar u svakodnevnom poslovanju.

Međutim, ovakav pristup može stvoriti nekoliko problema. Glavni problem je to što nemam stalni pristup istim fajlovima i podacima sa oba računara a čuvanje duplih kopija podataka na oba računara je veoma neefikasno trošenje resursa.

Pored toga, ovaj sistem će pružiti mogućnost daljinskog razvijanja (remote deploy) frontend i backend aplikacija. To će mi omogućiti brzo testiranje i implementaciju novih funkcionalnosti bez potrebe za direktnim pristupom mašini na kojoj se aplikacija izvršava.

Glavni cilj mog projekta je implementacija jedinstvenog računarskog sistema koji će mi omogućiti pristup i čuvanje fajlova sa bilo koje mašine koju koristim. Ovaj sistem će se sastojati od centralizovanog računara koji će biti dostupan samo uz pomoć SSH ključeva.

## 2 OPŠTI PODACI O POSLOVNOM SISTEMU

Centralizovani pristup i upravljanje fajlovima u višeračunarskom okruženju omogućava efikasno deljenje i pristup fajlovima između više računara. Ovaj sistem je posebno dizajniran organizacije koje koriste više računara u svom radnom okruženju kako bi poboljšali produktivnost i olakšali upravljanje podacima.

### 2.1 KARAKTERISTIKE SISTEMA

U nastavku ovog poglavlja biće detaljnije opisane karakteristike ovakvog sistema, i koje su njihove prednosti, kao i način na koji pomažu pri obavljanju takvog oblika poslovanja. Uz ove karakteristike, poslovni sistem centralizovanog pristupa i upravljanja fajlovima u višeračunarskom okruženju pruža praktično i efikasno rešenje za izazove koje profesionalci i organizacije susreću pri korišćenju više računara u svakodnevnom poslovanju.

#### 2.1.1 CENTRALIZOVANO SKLADIŠTENJE PODATAKA

Sistem omogućava korisnicima da skladište sve svoje fajlove i podatke na centralnom računaru koji je pristupačan samo uz pomoć SSH ključeva. Ovo osigurava sigurnost i zaštitu podataka, smanjuje rizik od gubitka podataka i omogućava jednostavan pristup podacima sa bilo koje mašine u mreži.

#### 2.1.2 DELJENJE FAJLOVA

Korisnici mogu jednostavno deliti fajlove između računara u okviru sistema. To znači da je moguće brzo i efikasno razmenjivati dokumente, slike, video zapise i

druge vrste fajlova između računara, bez potrebe za lokalnim skladištenjem kao ni korišćenjem eksternih uređaja ili slanja fajlova putem email-a.

### 2.1.3 REMOTE DEPLOY

Korišćenje ovog sistema omogućava korisnicima da efikasnije upravljaju svojim radnim okruženjem. Pristup istim fajlovima sa različitih računara eliminiše potrebu za prenosom podataka i omogućava neprekidno radno iskustvo. Takođe, rasterećivanje lokalnih mašina pomaže u poboljšanju performansi i brzine rada.

### 2.1.4 POBOLJŠANA PRODUKTIVNOST

Sistem omogućava daljinsko razvijanje frontend i backend aplikacija. Korisnici mogu brzo testirati, implementirati i ažurirati svoje aplikacije bez potrebe za direktnim pristupom mašini na kojoj se aplikacija izvršava. Ovo ubrzava proces razvoja i smanjuje vreme potrebno za implementaciju novih funkcionalnosti.

### 2.1.5 BEZBEDNOST PODATAKA

Sistem pruža visok nivo bezbednosti podataka putem autentifikacije putem SSH ključeva. Ovo obezbeđuje da samo ovlašćene osobe imaju pristup fajlovima i smanjuje rizik od neovlašćenog pristupa podacima.

## 3 ANALIZA POSTOJEĆEG INFORMACIONOG SISTEMA ORGANIZACIJE

Trenutni informacioni sistem organizacije ima kopije projekata koje se čuvaju na više računara radi pristupa i rada na njima. Ova praksa često rezultira višestrukim kopijama istog projekta na različitim uređajima. Jedna od mana ovog pristupa je zauzimanje prostora na svakom računaru, što može dovesti do dupliranja ili zastarelosti verzija projekata na različitim uređajima.

Dodatno, ručno prenošenje fajlova između računara je neophodno kako biste imali pristup ažuriranim verzijama projekata. Ovaj proces može biti zamoran, posebno ako često radite na različitim računarima ili delite projekte sa drugim članovima tima.

Ukupno gledano, trenutni informacioni sistem organizacije ima nekoliko nedostataka. Zauzimanje prostora na svakom računaru mogu ograničiti efikasnost rada na različitim računarima. Ručno prenošenje fajlova takođe može biti vremenski zahtevno i otežati proces deljenja ažuriranih verzija projekata.

Unapređenje informacionog sistema organizacije kroz centralizovani sistem upravljanja fajlovima adresaće ove mane pružajući efikasnije rešenje za skladištenje, pristup i deljenje projekata.

### **3.1 HARDVER POSTOJEĆEG SISTEMA**

Spisak svih računara koje koristim u svrhe poslovanja:

- Desktop računar 1:
  - o 16GB RAM-a
  - o Ryzen 5 procesor sa 6 jezgara i 12 niti
  - o 512GB SSD-a
- Desktop računar 2:
  - o 16GB RAM-a
  - o Intel i7 procesor sa 14 jezgara i 20 niti
  - o 1TB SSD-a
- Laptop:
  - o 12GB RAM-a
  - o Intel i5 procesor sa 4 jezgara i 8 niti
  - o 512GB SSD-a

### **3.2 SOFTVER POSTOJEĆEG SISTEMA**

Softver se, u ovom slučaju, odnosi, osim na konkretan softver (aplikacije računara), i na same operativne sisteme svih računara. Računari koji se koriste za rad sa klijentima imaju Arch Linux operativni sistem

Svaki od računara obavezno ima:

- WPS Writer
- WPS Spreadsheets
- WPS PDF
- WPS Presentation
- JetBrains IntelliJ
- JetBrains Webstorm
- Bitwarden Password Manager
- Insomnia
- Slack
- Zoom
- Git
- OpenVPN

Kao što se vidi kroz spisak neophodnog softvera na svakom računaru za ovu normalno poslovanje sa klijentima kao i ažurno regulisanje dokumenata, tako da oslobađanje memorije na svim računarima predstavlja jedan od najbitnijih faktora

ovog sistema, jer pored toga čuvanje podataka na jednom cloud računaru pruža i uniformno skladište podataka gde ne moramo brinuti o neusaglašenim dokumentima i podacima.

## **4 UPOREDNI PREGLED TEHNOLOGIJA**

### **4.1 TRENUTNI SISTEM**

#### **4.1.1 SKLADIŠTENJE PROJEKATA**

Projekti se čuvaju lokalno na svakom računaru.

#### **4.1.2 DELJENJE PROJEKATA**

Ručno prenošenje fajlova između računara ili upotreba Git-a za verzionisanje projekata.

#### **4.1.3 RESURSI**

Pokretanje projekata zahteva resurse na svakom računaru.

#### **4.1.4 SLANJE DOKUMENATA**

Dokumenti se ručno šalju mejlom sa jednog računara na drugi.

#### **4.1.5 PODEŠAVANJE OKRUŽENJA**

Svaki računar mora biti posebno podešen sa odgovarajućim alatima i okruženjem za razvoj.

### **4.2 SISTEM KOJI ĆE SE IMPLEMENITRATI**

#### **4.2.1 SKLADIŠTENJE PROJEKATA**

Projekti će biti smešteni na centralnom serveru u cloud okruženju.

#### **4.2.2 DELJENJE PROJEKATA**

Svi korisnici će imati pristup istim projektima putem centralnog servera. Neće biti potrebe za ručnim prenošenjem fajlova.

#### **4.2.3 RESURSI**

Projekti će se izvršavati na centralnom serveru, oslobađajući resurse i memoriju na lokalnim računarima.

#### **4.2.4 SLANJE DOKUMENATA**

Skenirani dokumenti će biti čuvani na centralnom serveru i lako dostupni za deljenje sa drugim korisnicima.

#### **4.2.5 PODEŠAVANJE OKRUŽENJA**

Korisnici će imati pristup centralnom serveru koji je već podešen sa svim neophodnim alatima i okruženjem za razvoj.

## **5 PREDLOG REŠENJA ZA UNAPREĐENJE POSLOVNOG SISTEMA**

Za implementaciju predloženog sistema, na centralnom serveru će biti instaliran Ubuntu Server operativni sistem bez grafičkog interfejsa (GUI). Ova odluka je doneta s ciljem oslobađanja resursa servera i optimizacije performansi.

Korisnici će biti u mogućnosti da se povežu sa serverom putem SSH protokola koristeći Command Line Interface (CLI). SSH omogućava sigurno i enkriptovano povezivanje sa serverom i pruža korisnicima mogućnost da obavljaju različite zadatke, uključujući deploy, build i druge manipulacije podacima.

Kroz SSH, korisnici će moći da pristupe svim potrebnim alatima i resursima na serveru kako bi izvršavali zadatke vezane za razvoj aplikacija. Na primer, moguće je pokretanje build procesa, implementacija promena i nadgledanje aplikacija koje su smeštene na serveru.

Osim toga, korisnici će imati privilegije za restartovanje servera i izmenu drugih podešavanja u slučaju grešaka ili potrebe za prilagođavanjem okruženja. Ovo će im omogućiti brzu reakciju i efikasno rešavanje eventualnih problema koji se mogu javiti.

Pristup putem SSH-a i korišćenje CLI-a pruža korisnicima fleksibilnost i kontrolu nad serverom, omogućavajući im da obavljaju različite operacije bez potrebe za grafičkim interfejsom.

Ovo su neki od ključnih aspekata centralizovanog rešenja:

### **5.1 CLOUD SKLADIŠTENJE PROJEKATA**

Projekti će biti smešteni na centralnom serveru u cloud okruženju. Ovo će eliminisati potrebu za lokalnim kopijama projekata na svakom računaru, oslobađajući prostor i smanjujući rizik od dupliranja ili zastarelih verzija.

### **5.2 CENTRALIZOVANI PRISTUP PROJEKTIMA**

Svi korisnici će imati pristup istim projektima putem centralnog servera. To znači da neće biti potrebe za ručnim prenošenjem fajlova između računara ili upotrebe Git-a za deljenje. Svi korisnici će raditi sa ažurnim verzijama projekata.

### **5.3 IZVRŠAVANJE PROJEKATA NA CENTRALNOM SERVERU**

Umesto da se projekti pokreću na lokalnim računarima, oni će se izvršavati na centralnom serveru. Ovo će smanjiti opterećenje resursa na lokalnim računarima i poboljšati njihovu brzinu rada.



## **5.4 CENTRALIZOVANO SKLADIŠTENJE I DELJENJE DOKUMENATA**

Skenirani dokumenti će biti čuvani na centralnom serveru i lako dostupni za deljenje sa drugim korisnicima. Umesto ručnog slanja mejlom, korisnici će moći jednostavno pristupiti dokumentima i deliti ih sa drugima.

## **5.5 PODEŠAVANJE OKRUŽENJA**

Centralni server će biti već podešen sa svim neophodnim alatima i okruženjem za razvoj projekata. Korisnici će imati pristup već konfigurisanom okruženju, eliminisajući potrebu za pojedinačnim podešavanjem na svakom računaru.

Implementacija ovog rešenja omogućiće efikasnije upravljanje projektima, smanjenje dupliranja podataka, olakšan pristup i deljenje projekata između korisnika, kao i smanjenje ručnih koraka u radnom procesu. Centralizacija fajlova, izvršavanje projekata na centralnom serveru i cloud skladištenje donose brojne prednosti u efikasnosti, upravljanju resursima i brzini rada.

Unapređenje poslovnog sistema kroz implementaciju centralizovanog sistema upravljanja fajlovima u cloud okruženju će transformisati način na koji se projekti čuvaju, pristupaju i deluju u vašoj organizaciji.

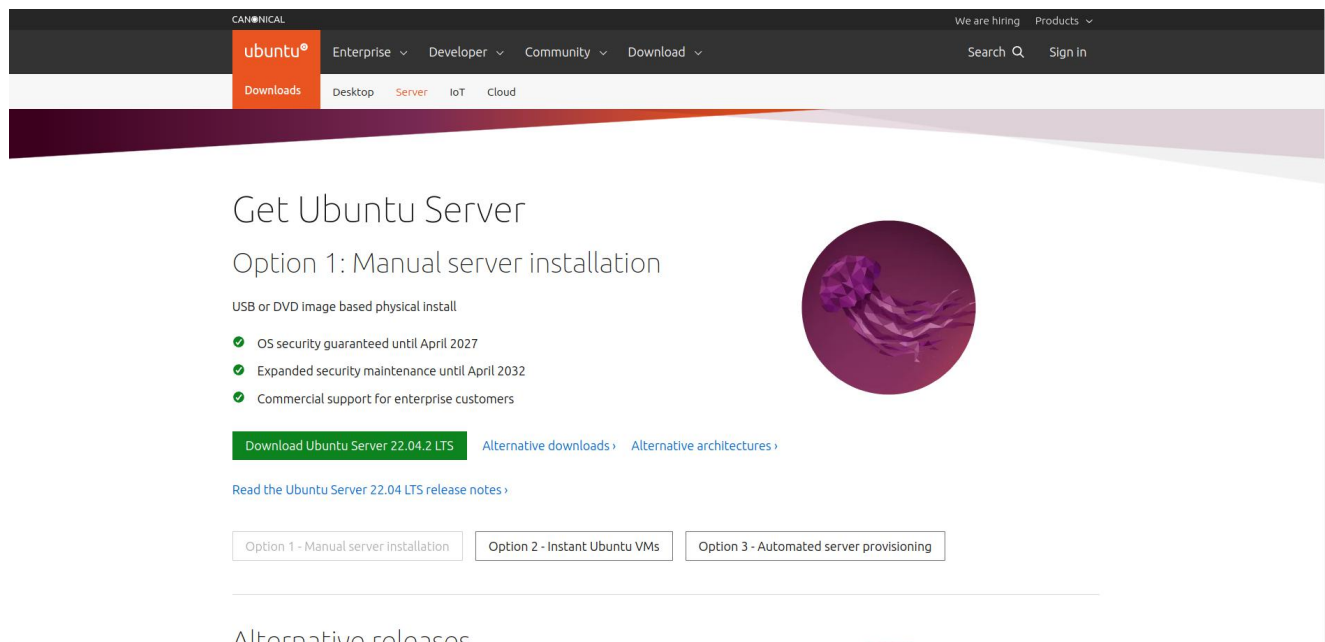
## **6 IMPLEMENTACIJA REŠENJA ZA UNAPREĐENJE POSLOVNOG SISTEMA**

Praktični primer primene predloženog rešenja za unapređenje izabranog poslovnog sistema. Mašina na kojoj će biti podignut sistem ima sledeću konfiguraciju:

- Procesor sa četiri jezgara i osam niti
- 8GB RAM-a
- SSD od 256GB za operativni sistem
- HDD od 3TB za čuvanje podataka

### **6.1 INSTALACIJA I PODEŠAVANJE UBUNTU SERVER 22.04.2 LTS**

Prvi korak je da preuzmemo ISO fajl Ubuntu Server-a sa zvaničnog sajta. Biramo poslednju verziju koja ima LTS (Long Term Support) podršku. (slika 1)



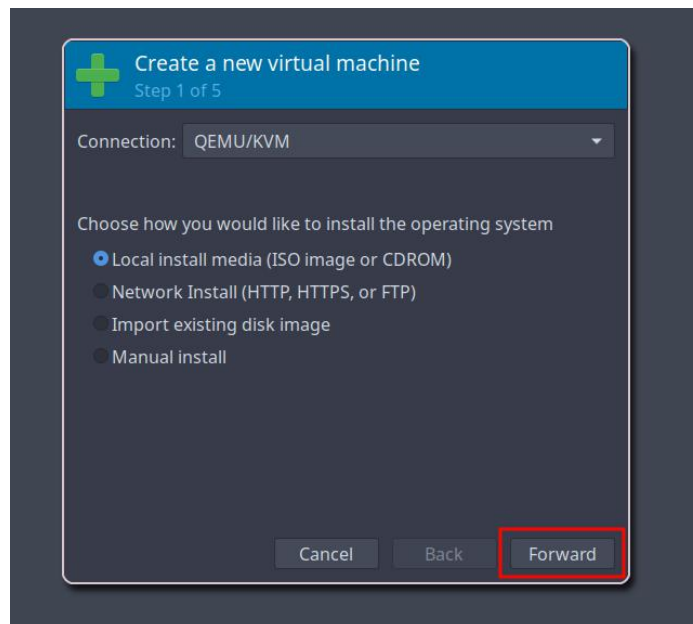
Slika 1. Preuzimanje Ubuntu Server ISO fajla

U virt-manageru biramo opciju "create a new virtual machine" u gornjem levom uglu gui-a. (slika 2)



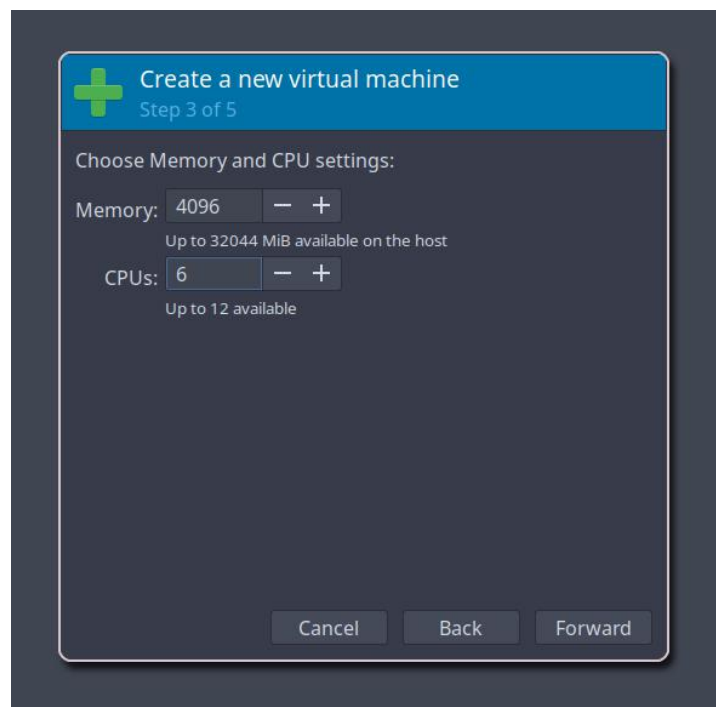
Slika 2. Odabiranje opcije za kreiranje nove virtualne mašine unutar virt managera

Biramo opciju za instaliranje koristeći ISO image. (slika 3)



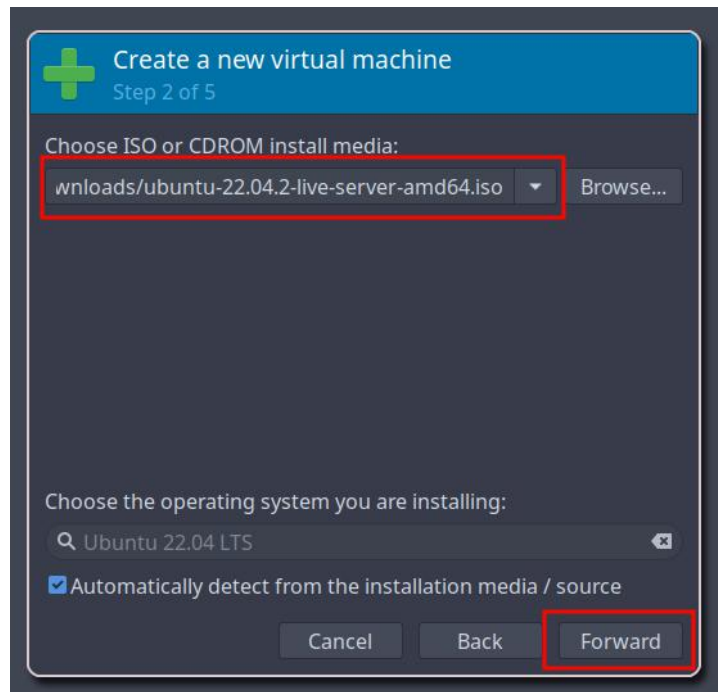
Slika 3. Odabiranje opcije za kreiranje nove virtualne mašine unutar virt menagera

Dodeljujemo 4GB RAM memorije i 6 jezgara procesora za ovu virtualnu mašinu jer više RAM-a nisam mogao da izdvojim sa trenutnog računara, a 4 je sasvim dovoljno za svrhe prikazivanja kako bi ovakav cloud funkcionisao, a 6 jezgara radi bržeg instaliranja paketa. (slika 4)



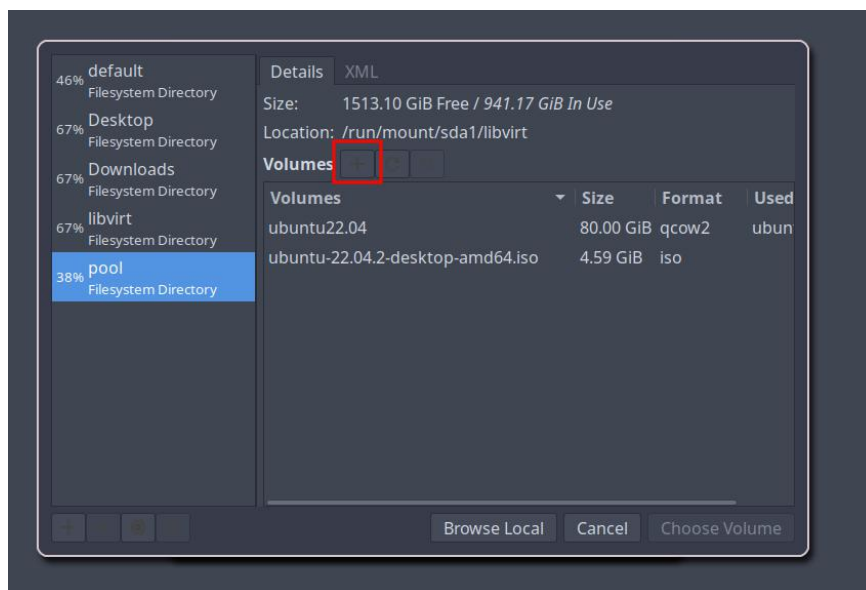
Slika 4. Alociranje resursa za virtualnu mašinu  
Projektni zadatak: Naziv projektnog zadatka

Sledeći korak je da odaberemo ISO fajl sa lokalnog računara koji smo preuzeli sa Ubuntu zvaničnog sajta. (slika 5)



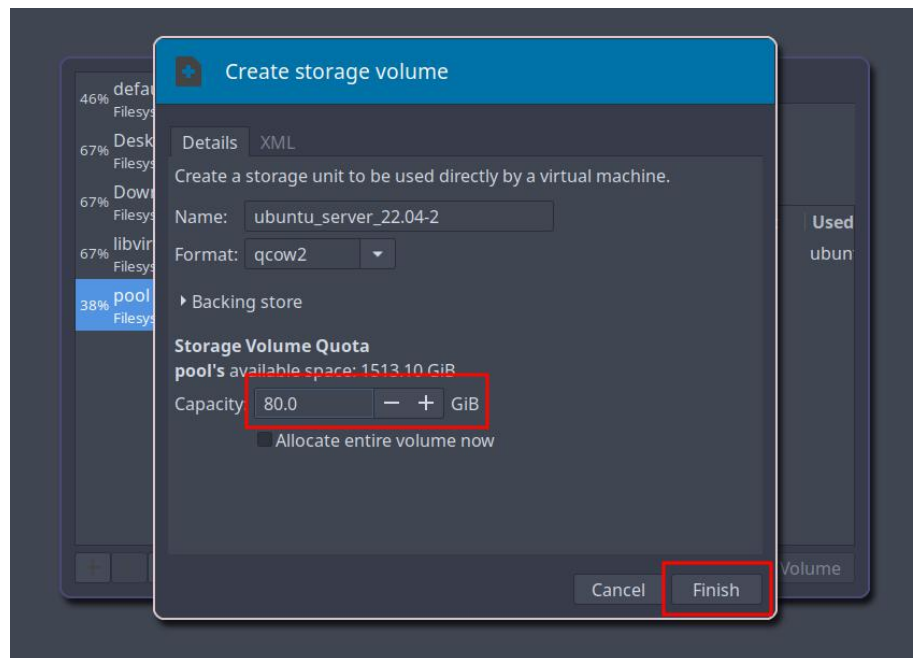
Slika 5. Odabiranje ISO-a za instalaciju Ubuntu Servera

U ovom koraku kreiramo Volume sa koga izdvajamo željenu količinu memorije koja će služiti kao particija na kojoj je instaliran Ubuntu Server. (slika 6)



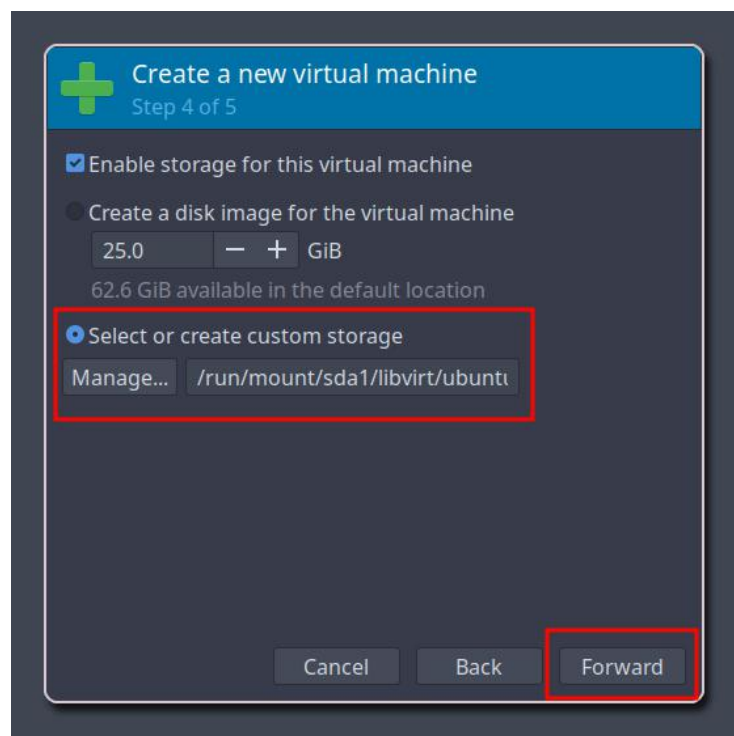
Slika 6. Odabiranje pool-a za novi volume

Odabiramo količinu memorije koju alociramo za volume koji kreiramo. (slika 7)



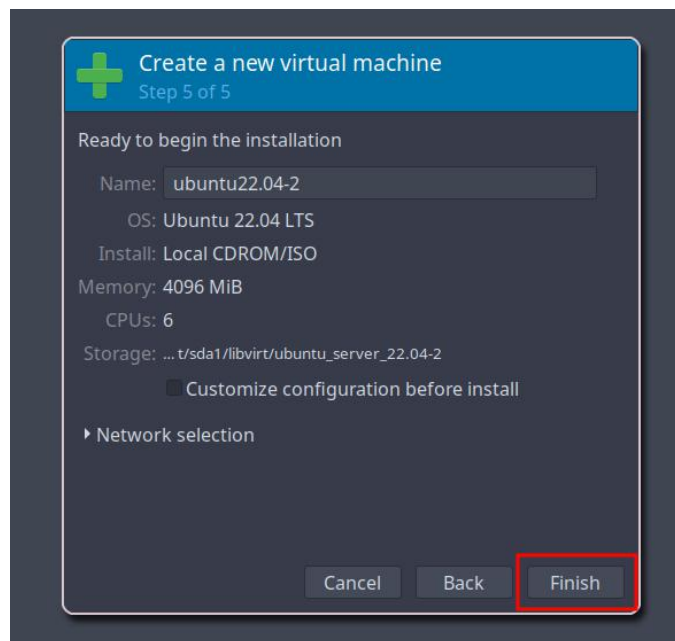
Slika 7. Odabiranje količine memorije za novi volume

Koristimo opciju kreiranja custom storage iz pool-a jer nema dovoljno memorije u home i root particiji na računaru sa koga se kreira virtualna mašina. (slika 8)



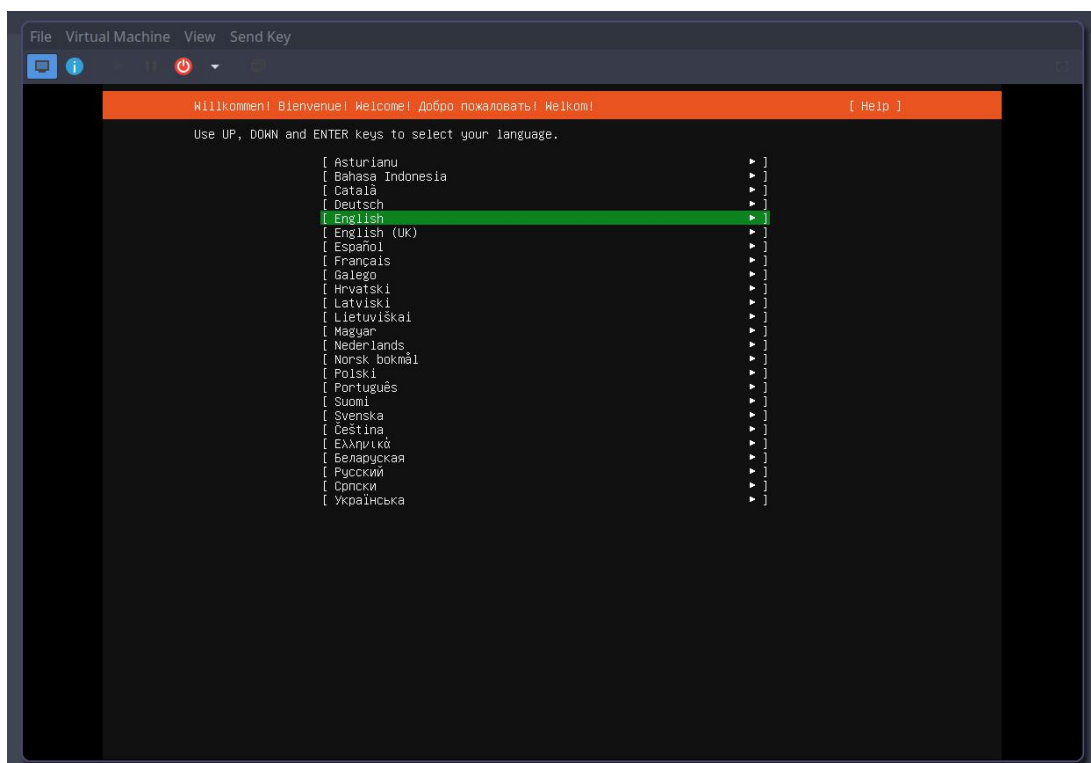
Slika 8. Odabiranje opcije za custom storage  
Projektni zadatak: Naziv projektnog zadatka

Na finalnom koraku pri podešavanju virtualne mašine je sumrizovan pregled odabranih podešavanja sa opcijom da to potvrdimo i kreiramo virtualnu mašinu. (slika 9)

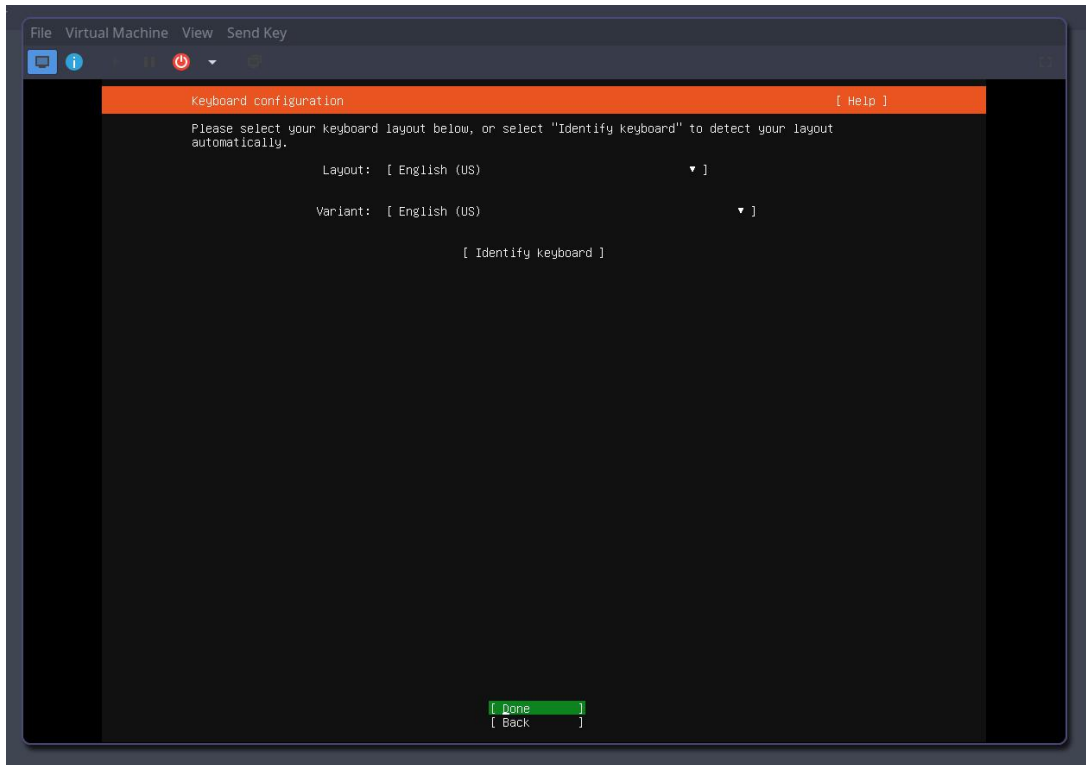


Slika 9. Potvrda sumrizovanih podešavanja

U narednih par slika možemo videti neka od osnovnih kao i složenijih podešavanja vezano za instalaciju samog OS-a. (slika 10)

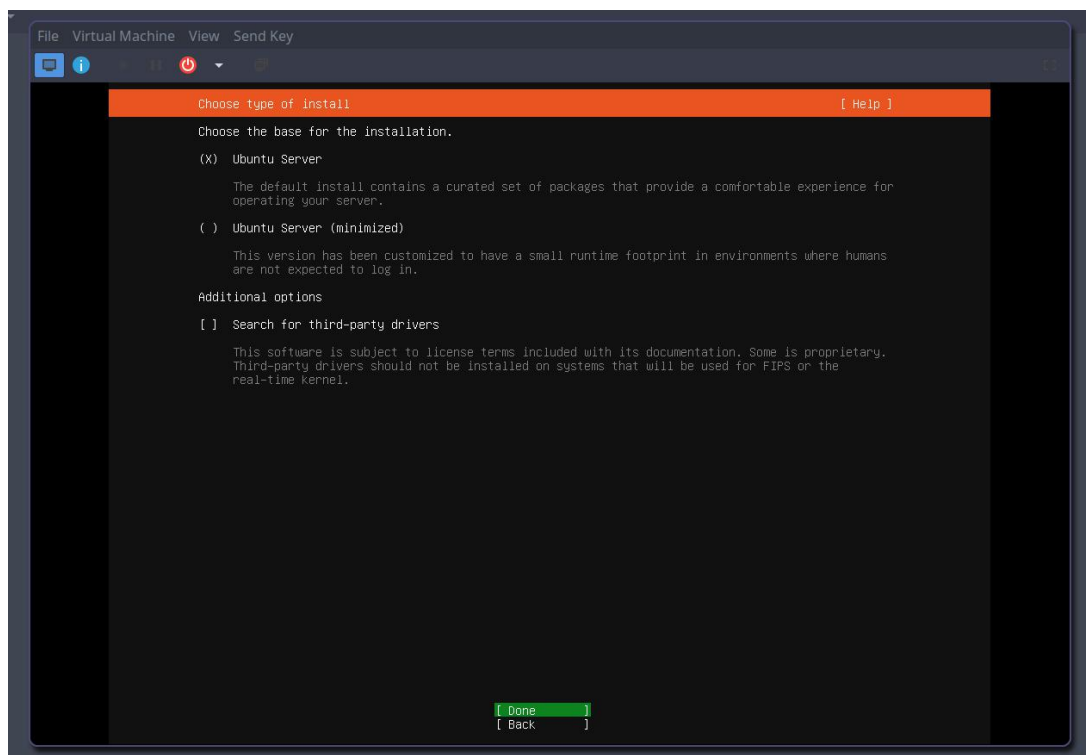


Slika 10. Odabir jezika



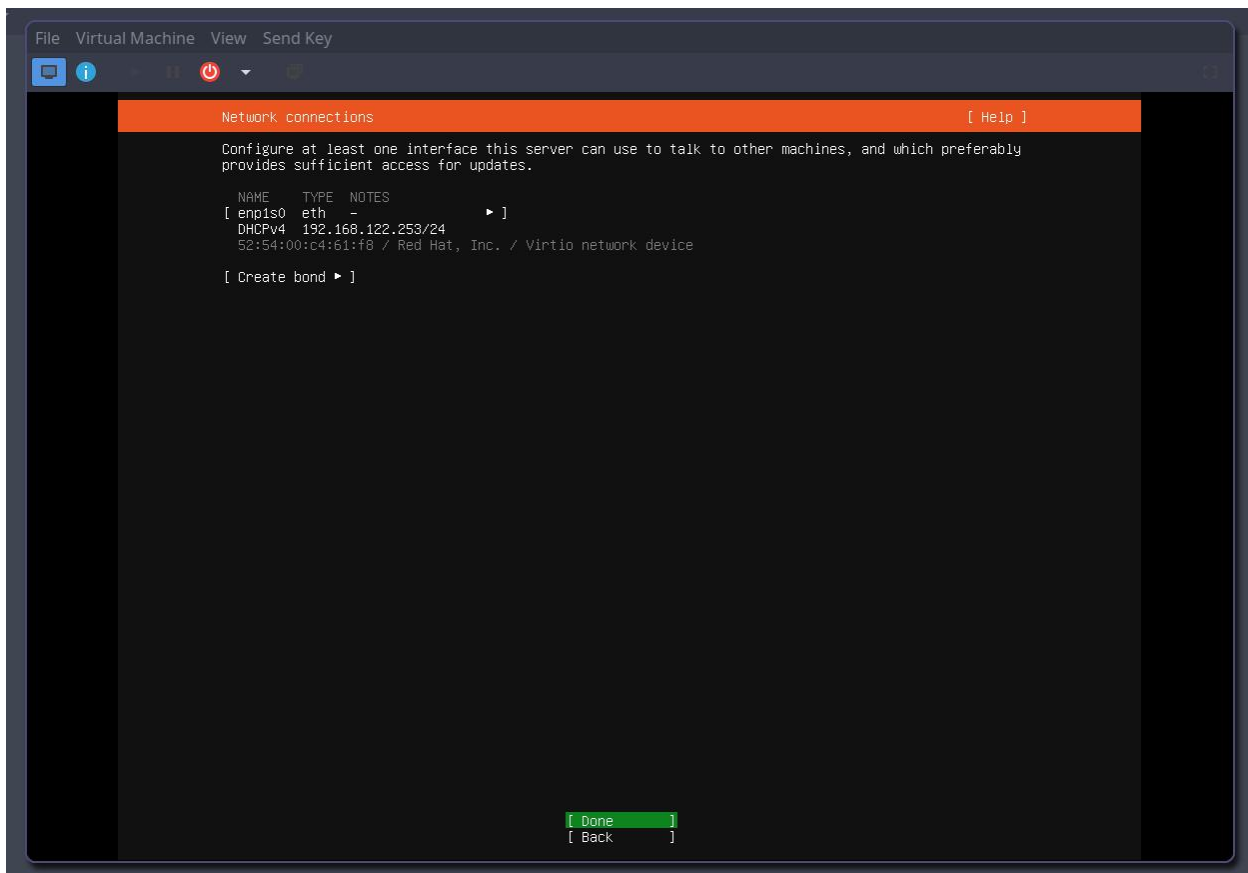
*Slika 11. Odabir keyboard layout-a*

Pri odabiru vrste instalacije imamo razne opcije, mi biramo default Ubuntu Server, kako bi dalje birali željene pakete koji su nam neophodni. (slika 12)



*Slika 12. Odabir vrste instalacije*  
*Projektni zadatak: Naziv projektnog zadatka*

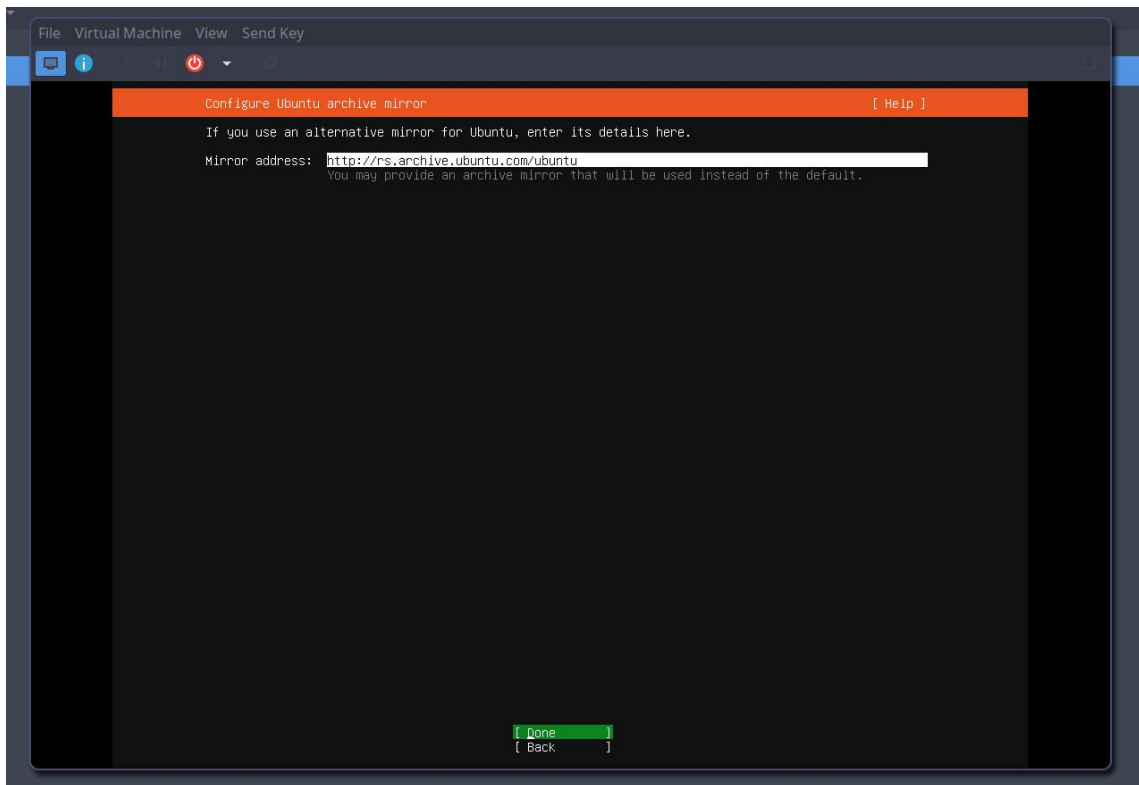
Pri ovom koraku biramo default network interface koji će se koristiti kako bi ova mašina imala internet konekciju i komunicirala sa drugim mašinama. (slika 13)



Slika 13. Podešavanje default network interface-a

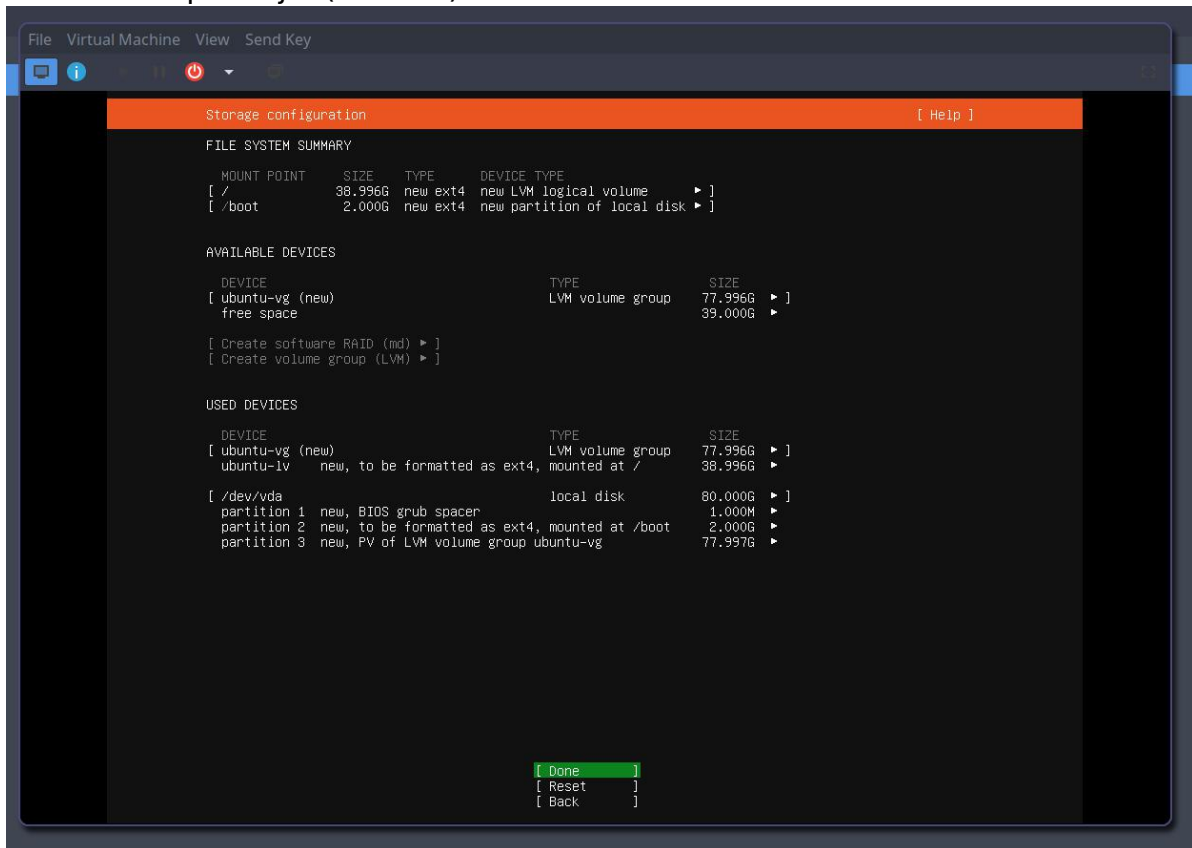


U ovom koraku biramo mirror adresu, to je adresa servera koji sadrže kopije ubuntu paketa i omogućavaju brže preuzimanje, u idealnom slučaju koristimo više mirror adresa koje su fizički što bliže našoj zemlji. (slika 14)



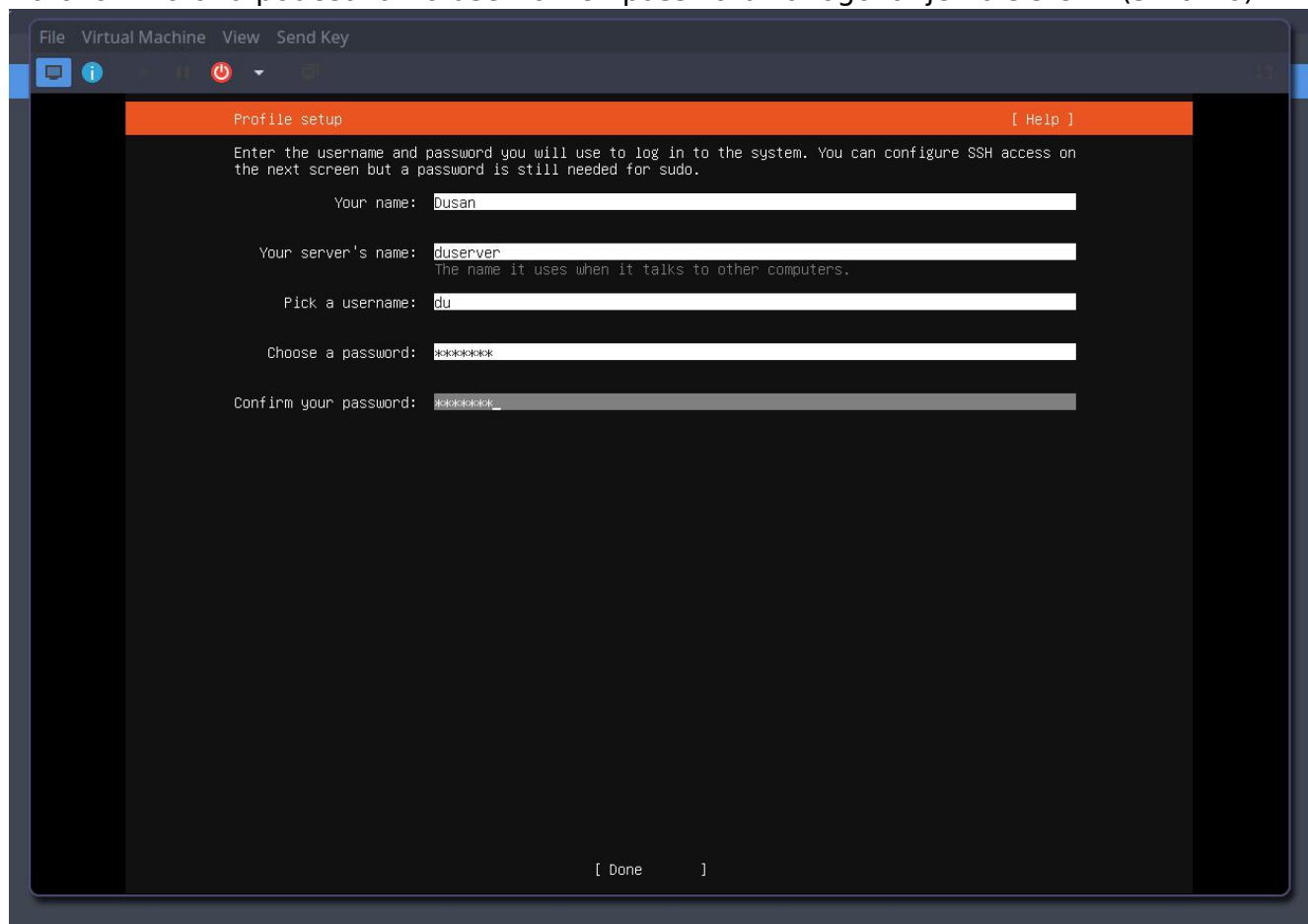
Slika 14. Podešavanje mirror adresa

Na ovom koraku podešavamo particije sistema, za potrebe demonstracije sistema ću sve smestiti u root particiju, ali u realnoj situaciji bi root bio na SSD-u od 256GB, a HDD od 3TB bi bio home particija. (slika 15)



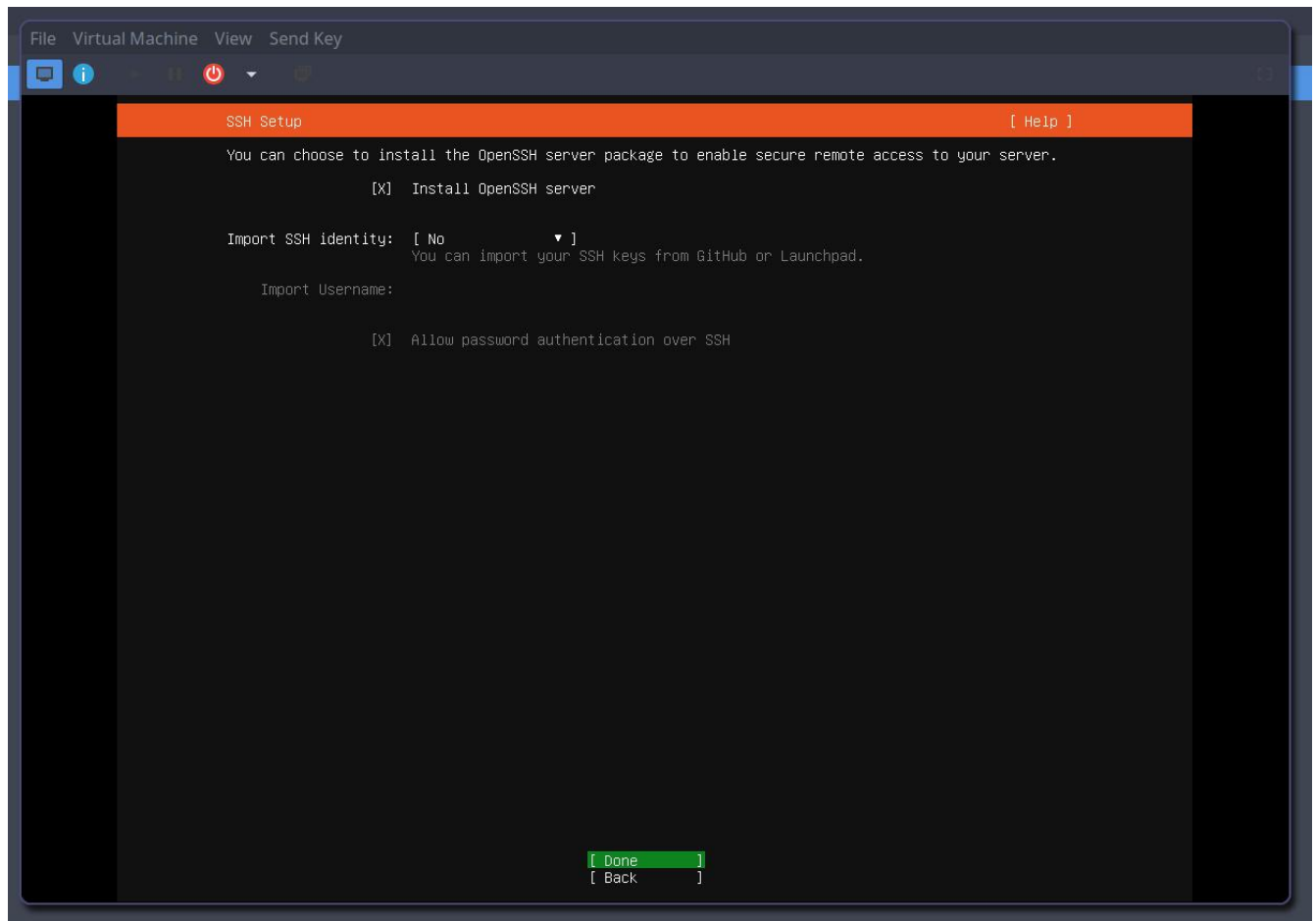
Slika 15. Podešavanje particija

Na ovom koraku podešavamo username i password za logovanje na sistem. (slika 16)



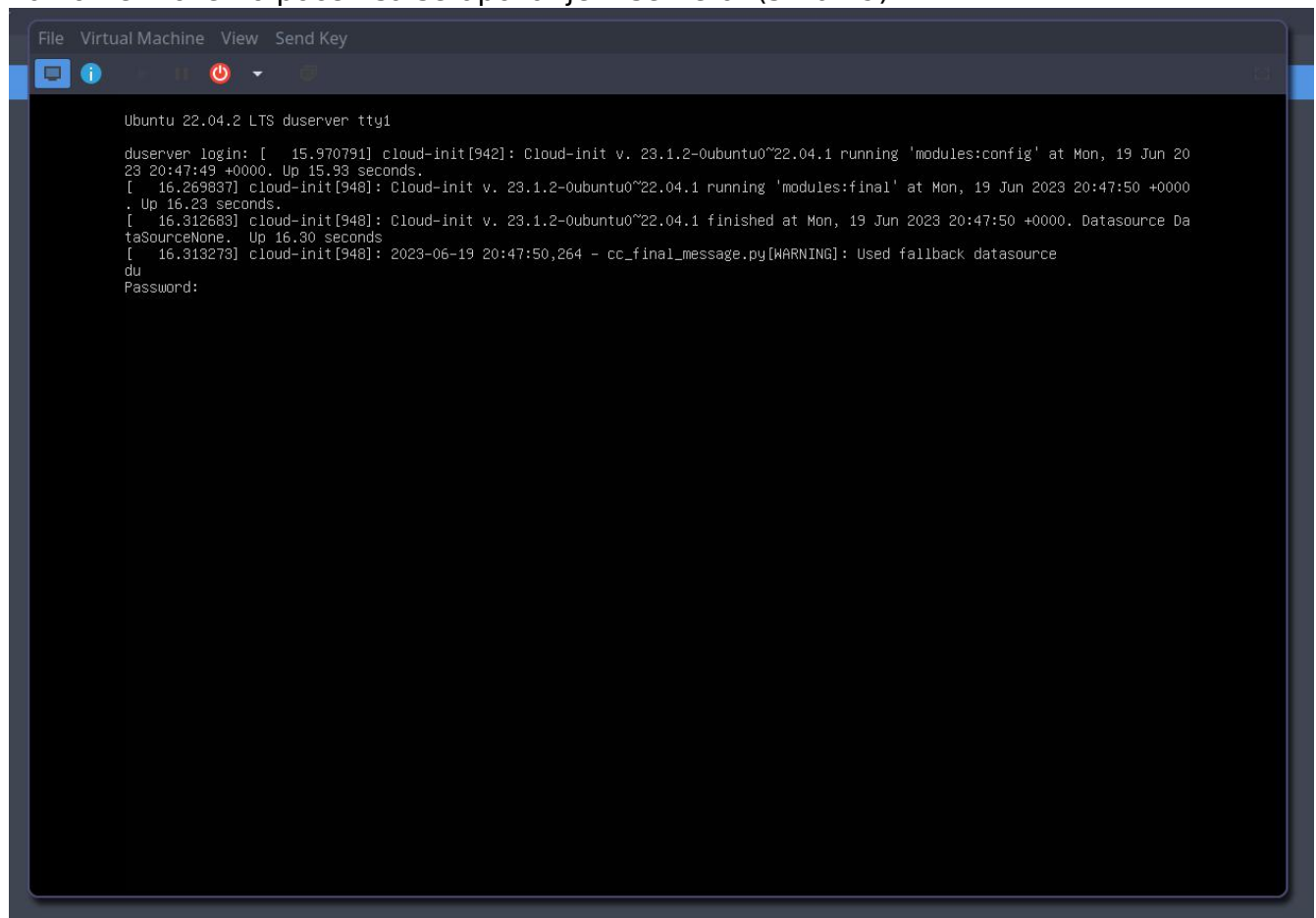
Slika 16. Podešavanje kredencijala

Na ovom koraku instaliramo OpenSSH server koji će nam omogućiti najbezbedniji način povezivanja sa računarima. (slika 17)



Slika 17. Instaliranje OpenSSH servera

U sledećem snimku ekrana vidimo početni ekran gde se logujemo tako što unesemo username i password (slika 18), i nakon toga možemo videti CLI(command line interface) na kome možemo početi sa setapovanjem servera. (slika 19)



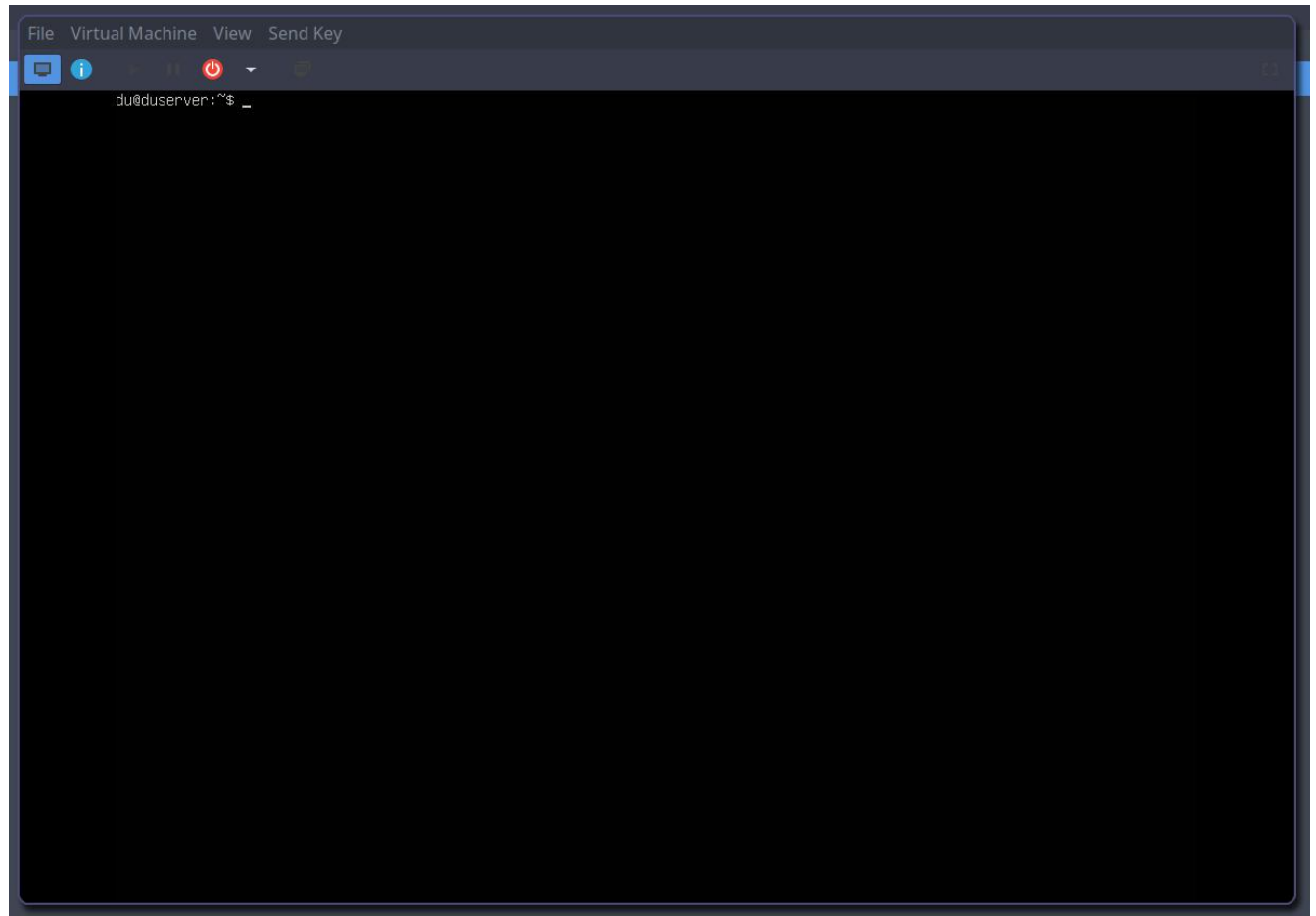
```

File Virtual Machine View Send Key

Ubuntu 22.04.2 LTS duserver tty1

duserver login: [ 15.970791] cloud-init[942]: Cloud-init v. 23.1.2-0ubuntu0~22.04.1 running 'modules:config' at Mon, 19 Jun 20
23 20:47:49 +0000. Up 15.93 seconds.
[ 16.269837] cloud-init[948]: Cloud-init v. 23.1.2-0ubuntu0~22.04.1 running 'modules:final' at Mon, 19 Jun 2023 20:47:50 +0000
. Up 16.23 seconds.
[ 16.312683] cloud-init[948]: Cloud-init v. 23.1.2-0ubuntu0~22.04.1 finished at Mon, 19 Jun 2023 20:47:50 +0000. Datasource Da
taSourceNone. Up 16.30 seconds
[ 16.313273] cloud-init[948]: 2023-06-19 20:47:50,264 - cc_final_message.py[WARNING]: Used fallback datasource
du
Password:
  
```

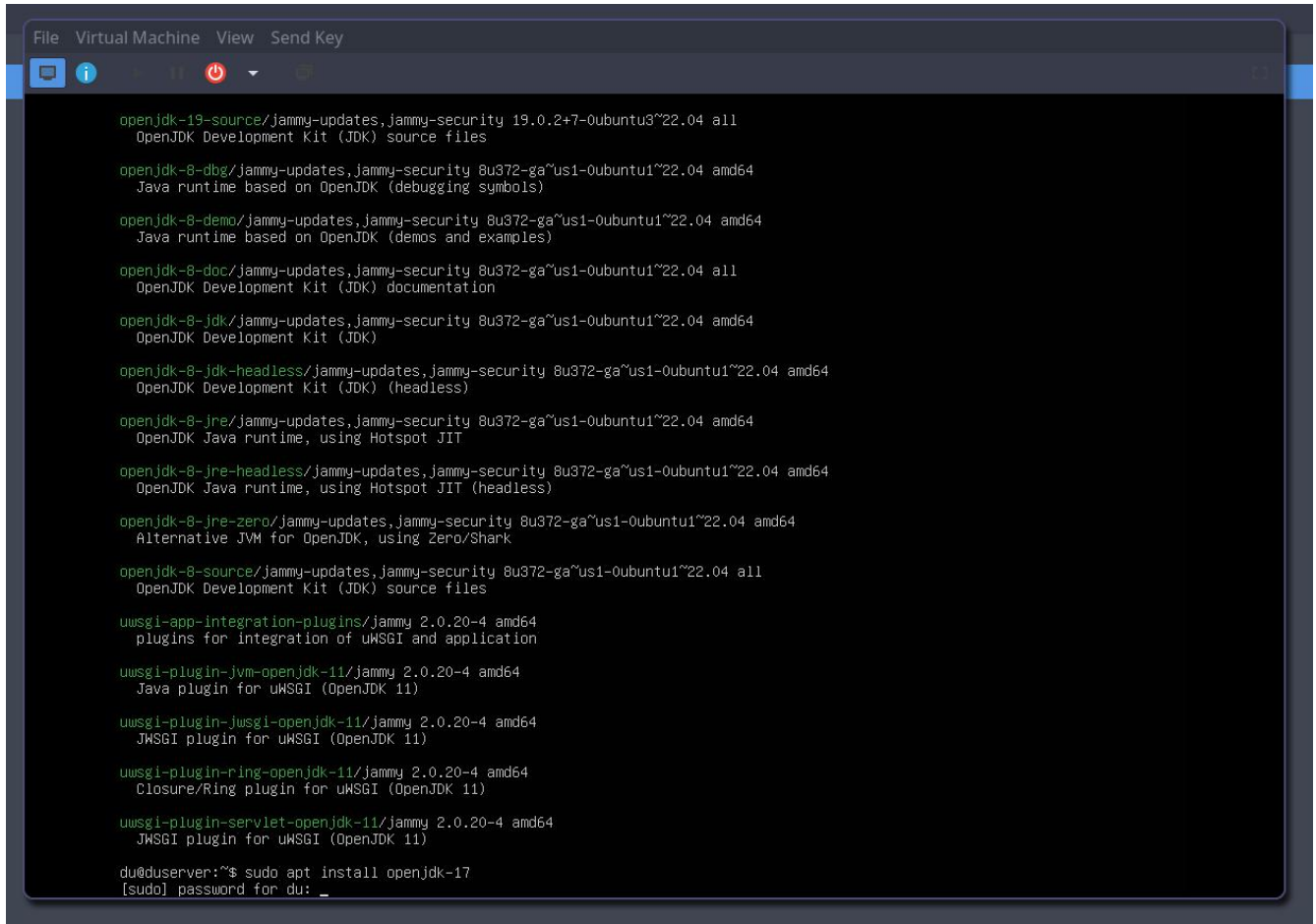
Slika 18. Logovanje na sistem



*Slika 19. CLI nakon logovanja na sistem*

## 6.2 INSTALACIJA I PODEŠAVANJE SOFTVERA ZA CLOUD

Prvi paket koji instaliramo je openjdk-17 kako bi pokretali springboot backend aplikacije. (slika 20 i 21)



```

openjdk-19-source/jammy-updates,jammy-security 19.0.2+7-0ubuntu3~22.04 all
  OpenJDK Development Kit (JDK) source files

openjdk-8-dbg/jammy-updates,jammy-security 8u372-ga~us1-0ubuntu1~22.04 amd64
  Java runtime based on OpenJDK (debugging symbols)

openjdk-8-demo/jammy-updates,jammy-security 8u372-ga~us1-0ubuntu1~22.04 amd64
  Java runtime based on OpenJDK (demos and examples)

openjdk-8-doc/jammy-updates,jammy-security 8u372-ga~us1-0ubuntu1~22.04 all
  OpenJDK Development Kit (JDK) documentation

openjdk-8-jdk/jammy-updates,jammy-security 8u372-ga~us1-0ubuntu1~22.04 amd64
  OpenJDK Development Kit (JDK)

openjdk-8-jdk-headless/jammy-updates,jammy-security 8u372-ga~us1-0ubuntu1~22.04 amd64
  OpenJDK Development Kit (JDK) (headless)

openjdk-8-jre/jammy-updates,jammy-security 8u372-ga~us1-0ubuntu1~22.04 amd64
  OpenJDK Java runtime, using Hotspot JIT

openjdk-8-jre-headless/jammy-updates,jammy-security 8u372-ga~us1-0ubuntu1~22.04 amd64
  OpenJDK Java runtime, using Hotspot JIT (headless)

openjdk-8-jre-zero/jammy-updates,jammy-security 8u372-ga~us1-0ubuntu1~22.04 amd64
  Alternative JVM for OpenJDK, using Zero/Shark

openjdk-8-source/jammy-updates,jammy-security 8u372-ga~us1-0ubuntu1~22.04 all
  OpenJDK Development Kit (JDK) source files

uwsgi-app-integration-plugins/jammy 2.0.20-4 amd64
  plugins for integration of uWSGI and application

uwsgi-plugin-jvm-openjdk-11/jammy 2.0.20-4 amd64
  Java plugin for uWSGI (OpenJDK 11)

uwsgi-plugin-jwsgi-openjdk-11/jammy 2.0.20-4 amd64
  JWSGI plugin for uWSGI (OpenJDK 11)

uwsgi-plugin-ring-openjdk-11/jammy 2.0.20-4 amd64
  Closure/Ring plugin for uWSGI (OpenJDK 11)

uwsgi-plugin-servlet-openjdk-11/jammy 2.0.20-4 amd64
  JWSGI plugin for uWSGI (OpenJDK 11)

du@duserver:~$ sudo apt install openjdk-17
[sudo] password for du: _

```

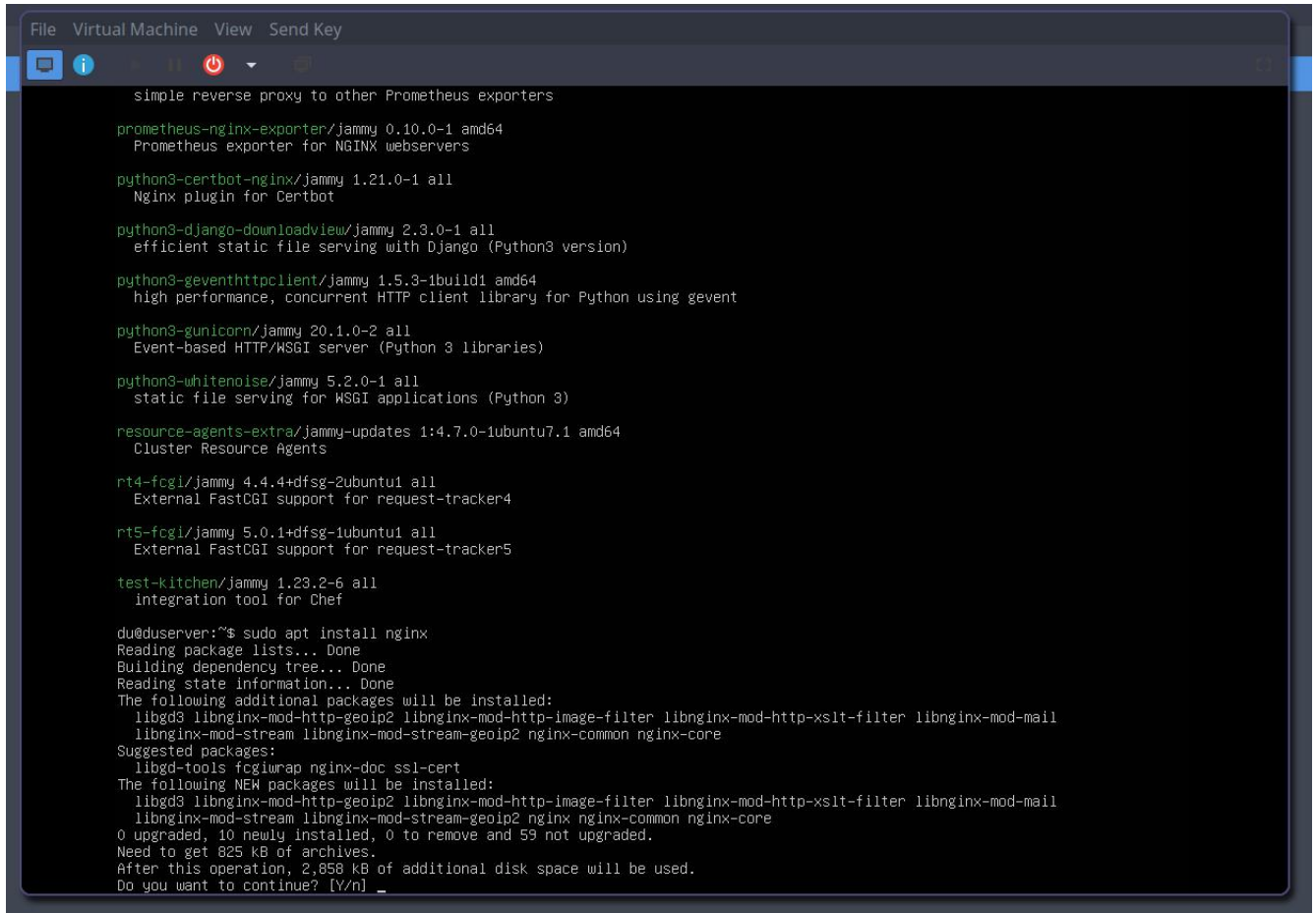
Slika 20. Pretraga openjdk paketa

```
File Virtual Machine View Send Key
Get:84 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libxv1 amd64 2:1.0.11-1build2 [11.2 kB]
Get:85 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libxxf86dga1 amd64 2:1.1.5-0ubuntu3 [12.6 kB]
Get:86 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 x11-utils amd64 7.7+5build2 [206 kB]
Get:87 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libatk-wrapper-java all 0.38.0-5build1 [53.1 kB]
Get:88 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libatk-wrapper-java-jni amd64 0.38.0-5build1 [49.0 kB]
Get:89 http://rs.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpixmap-1.0 amd64 0.40.0-1ubuntu0.22.04.1 [264 kB]
Get:90 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libxcb-render0 amd64 1.14-3ubuntu3 [16.4 kB]
Get:91 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libcairo2 amd64 1.16.0-5ubuntu2 [628 kB]
Get:92 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libcairo-gobject2 amd64 1.16.0-5ubuntu2 [19.4 kB]
Get:93 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libdatrie1 amd64 0.2.13-2 [19.9 kB]
Get:94 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libpciaccess0 amd64 0.16-3 [19.1 kB]
Get:95 http://rs.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libdrm-intel1 amd64 2.4.113-2~ubuntu0.22.04.1 [66.7 kB]
Get:96 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libgdk2.0-common all 2.24.33-2ubuntu2 [125 kB]
Get:97 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libthai-data all 0.1.29-1build1 [162 kB]
Get:98 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libthai0 amd64 0.1.29-1build1 [19.2 kB]
Get:99 http://rs.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpango-1.0-0 amd64 1.50.6+ds-2ubuntu1 [230 kB]
Get:100 http://rs.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpangocairo-1.0-0 amd64 1.50.6+ds-2ubuntu1 [54.0 kB]
Get:101 http://rs.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpangocairo-1.0-0 amd64 1.50.6+ds-2ubuntu1 [39.8 kB]
Get:102 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libxcursor1 amd64 1:1.2.0-2build4 [20.9 kB]
Get:103 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libxdamage1 amd64 1:1.1.5-2build2 [7,154 B]
Get:104 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libgtk2.0-0 amd64 2.24.33-2ubuntu2 [2,037 kB]
Get:105 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libgail18 amd64 2.24.33-2ubuntu2 [15.9 kB]
Get:106 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libgail-common amd64 2.24.33-2ubuntu2 [132 kB]
Get:107 http://rs.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgdk-pixbuf2.0-bin amd64 2.42.8+dfsg-1ubuntu0.2 [14.2 kB]
Get:108 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libgif7 amd64 5.1.9-2build2 [33.8 kB]
Get:109 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libgl1-amber-dri amd64 21.3.7-0ubuntu1 [4,433 kB]
Get:110 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libgtk2.0-bin amd64 2.24.33-2ubuntu2 [7,932 B]
Get:111 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 xorg-sgml-doctools all 1:1.11-1.1 [10.9 kB]
Get:112 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 x11proto-dev all 2021.5-1 [604 kB]
Get:113 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libice-dev amd64 2:1.0.10-1build2 [51.4 kB]
Get:114 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libpthread-stubs0-dev amd64 0.4-1build2 [5,516 B]
Get:115 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 librsvg2-2 amd64 2.52.5+dfsg-3 [3,020 kB]
Get:116 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 librsvg2-common amd64 2.52.5+dfsg-3 [17.7 kB]
Get:117 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libsm-dev amd64 2:1.2.3-1build2 [18.1 kB]
Get:118 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libxau-dev amd64 1:1.0.9-1build5 [9,724 B]
Get:119 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libxdmcp-dev amd64 1:1.1.3-0ubuntu5 [26.5 kB]
Get:120 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 xtrans-dev all 1.4.0-1 [68.9 kB]
Get:121 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libxcb1-dev amd64 1.14-3ubuntu3 [86.5 kB]
Get:122 http://rs.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libx11-dev amd64 2:1.7.5-1ubuntu0.2 [744 kB]
Get:123 http://rs.archive.ubuntu.com/ubuntu jammy/main amd64 libxt-dev amd64 1:1.2.1-1 [396 kB]
Get:124 http://rs.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 openjdk-17-jre amd64 17.0.7+7~us1-0ubuntu1~22.04.2 [203
kB]
Get:125 http://rs.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 openjdk-17-jdk-headless amd64 17.0.7+7~us1-0ubuntu1~22.
04.2 [71.0 MB]
Get:126 http://rs.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 openjdk-17-jdk amd64 17.0.7+7~us1-0ubuntu1~22.04.2 [2,7
42 kB]
Fetched 181 MB in 7s (24.8 MB/s)
Extracting templates from packages: 71%
```

Slika 21. Instaliranje openjdk-až



Sledeće što instaliramo je open source web server, nginx. (slika 22)



```

File Virtual Machine View Send Key

simple reverse proxy to other Prometheus exporters

prometheus-nginx-exporter/jammy 0.10.0-1 amd64
Prometheus exporter for NGINX web servers

python3-certbot-nginx/jammy 1.21.0-1 all
Nginx plugin for Certbot

python3-django-downloadview/jammy 2.3.0-1 all
efficient static file serving with Django (Python3 version)

python3-geventhttpclient/jammy 1.5.3-1build1 amd64
high performance, concurrent HTTP client library for Python using gevent

python3-gunicorn/jammy 20.1.0-2 all
Event-based HTTP/WSGI server (Python 3 libraries)

python3-whitenoise/jammy 5.2.0-1 all
static file serving for WSGI applications (Python 3)

resource-agents-extra/jammy-updates 1:4.7.0-1ubuntu7.1 amd64
Cluster Resource Agents

rt4-fcgi/jammy 4.4.4+dfsg-2ubuntu1 all
External FastCGI support for request-tracker4

rt5-fcgi/jammy 5.0.1+dfsg-1ubuntu1 all
External FastCGI support for request-tracker5

test-kitchen/jammy 1.23.2-6 all
integration tool for Chef

du@duserver:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libgd3 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail
  libnginx-mod-stream libnginx-mod-stream-geoip2 nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  libgd3 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail
  libnginx-mod-stream libnginx-mod-stream-geoip2 nginx nginx-common nginx-core
0 upgraded, 10 newly installed, 0 to remove and 59 not upgraded.
Need to get 825 kB of archives.
After this operation, 2,858 kB of additional disk space will be used.
Do you want to continue? [Y/n] _

```

Slika 22. Instaliranje nginx

## 6.3 DEPLOY BACKEND I FRONTEND APLIKACIJA

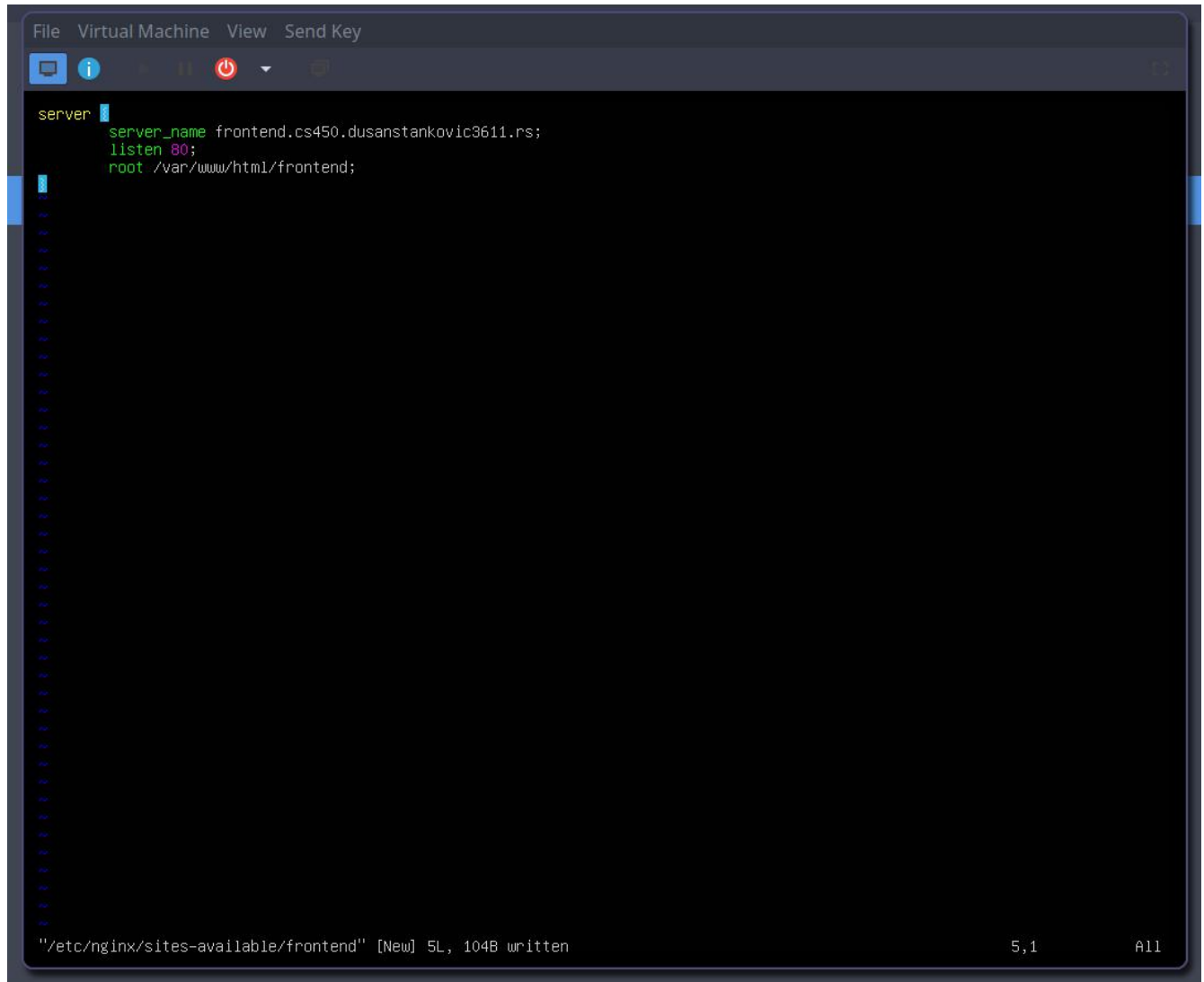
Kreiramo Nginx konfiguraciju za usmeravanje zahteva ka frontend aplikaciji sa sledećim postavkama:

U prvoj liniji konfiguracije servera, definišemo naziv servera koji je "frontend.cs450.cudanstankovic3611.rs". Ovo omogućava Nginx serveru da prepozna koji zahtevi treba da budu usmereni ka ovom serveru.

U drugoj liniji konfiguracije, koristimo "listen 80;" kako bismo postavili Nginx server da sluša na portu 80. Ovo je standardni HTTP port preko kojeg će se odvijati komunikacija između klijenta i servera.

Treća linija konfiguracije, "root /var/www/html/frontend;", definiše putanju do direktorijuma koji sadrži frontend aplikaciju. U ovom slučaju, direktorijum je "/var/www/html/frontend". Nginx će koristiti ovaj direktorijum kao izvor resursa koje će poslati klijentu prilikom zahteva.

Ove konfiguracijske postavke omogućavaju da Nginx server pravilno servira frontend aplikaciju sa nazivom servera "frontend.cs450.cudanstankovic3611.rs" na portu 80, koristeći resurse iz direktorijuma "/var/www/html/frontend". (slika 22)



The screenshot shows a terminal window with a dark background. The title bar at the top reads "File Virtual Machine View Send Key". Below the title bar, there are several icons: a terminal icon, an information icon, a power icon, and a refresh icon. The terminal content shows the following commands and output:

```
server
server_name frontend.cs450.dusanstankovic3611.rs;
listen 80;
root /var/www/html/frontend;
```

At the bottom of the terminal, a status bar indicates: `"etc/nginx/sites-available/frontend" [New] 5L, 104B written` on the left, and `5,1` and `All` on the right.

*Slika 23. Kreiranje nginx konfiguracije za usmeravanje zahteva ka frontend-u*

Kreiramo Nginx konfiguraciju za usmeravanje zahteva ka backend aplikaciji sa sledećim postavkama:

U prvoj liniji konfiguracije servera, definišemo naziv servera koji je "backend.cs450.dusanstankovic3611.rs". Ovo omogućava Nginx serveru da prepozna koji zahtevi treba da budu usmereni ka ovom serveru.

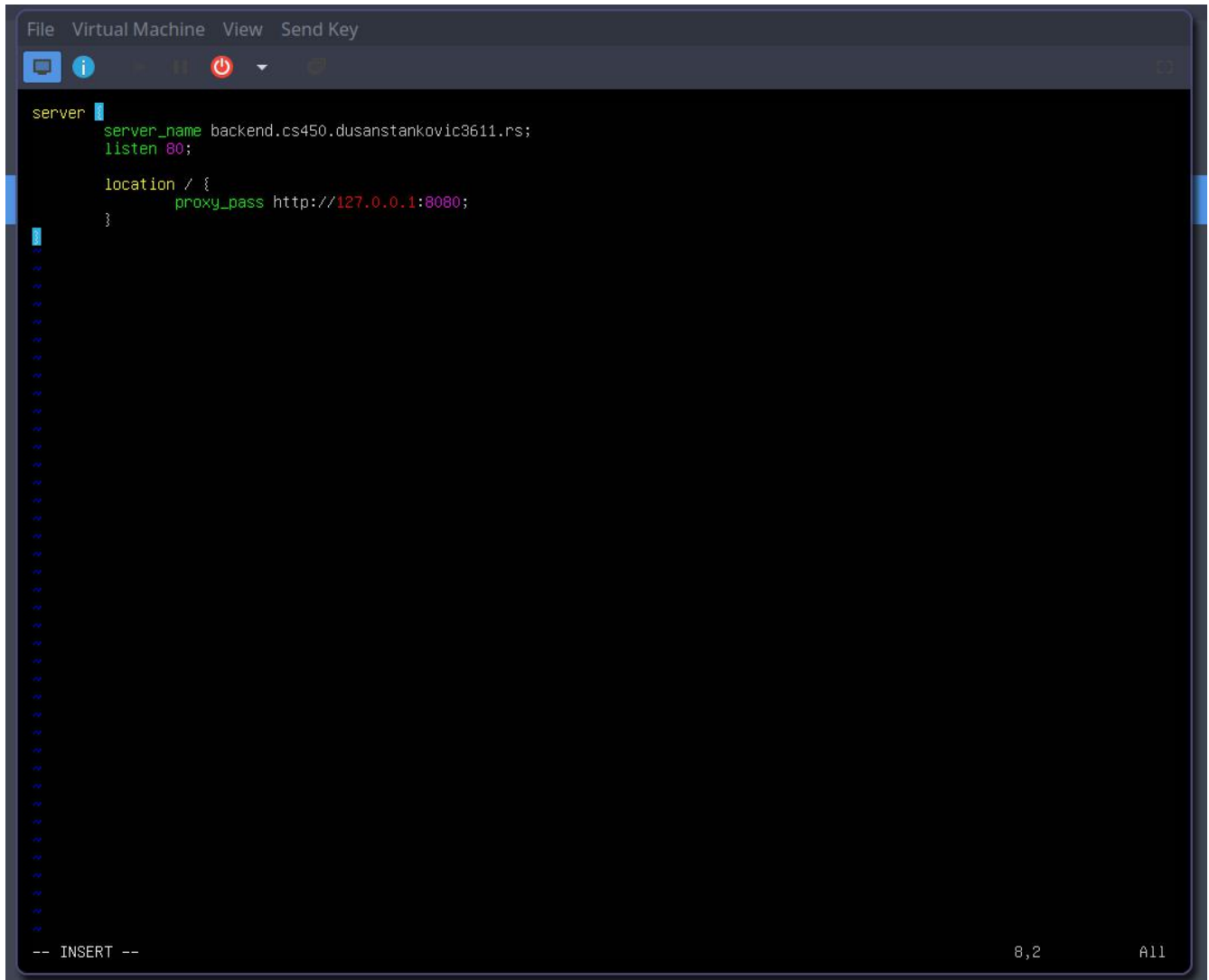
U drugoj liniji konfiguracije, koristimo "listen 80;" kako bismo postavili Nginx server da sluša na portu 80. Ovo je standardni HTTP port preko kojeg će se odvijati komunikacija između klijenta i servera.

Treća linija konfiguracije, location, definiše putanju koja će biti usmerena ka backend aplikaciji. U ovom slučaju, koristimo koren "/" kao putanju.

Unutar bloka "location /", koristimo "proxy\_pass http://127.0.0.1:8080;" kako bismo defisnali adresu backend servera na koji će Nginx usmeriti zahteve koji dolaze na ovu putanju. U ovom primeru, backend server je dostupan na adresi "http://127.0.0.1:8080".

*Projektni zadatak: Naziv projektnog zadatka*

Ove konfiguracione postavke omogućavaju da Nginx server pravilno usmerava zahteve koji dolaze na adresu "backend.cs450.dusanstankovic3611.rs" na portu 80 ka backend aplikaciji koja je dostupna na adresi "http://127.0.0.1:8080".



```

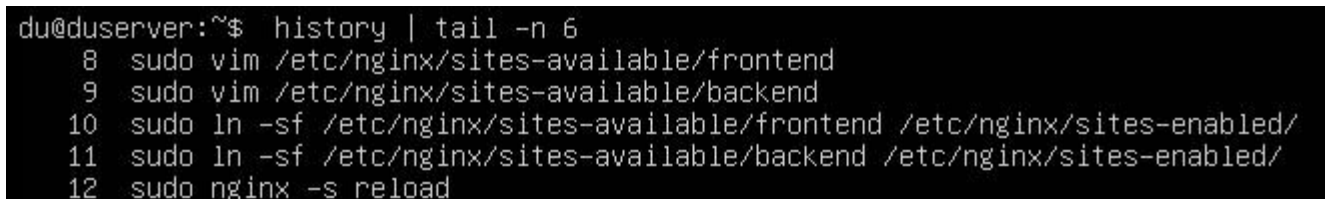
server {
    server_name backend.cs450.dusanstankovic3611.rs;
    listen 80;

    location / {
        proxy_pass http://127.0.0.1:8080;
    }
}

```

Slika 24. Kreiranje nginx konfiguracije za usmeravanje zahteva ka backend-u

Pravimo prečice obe konfiguracije u sites-enabled folder nginx-a kako bi uključili podizanje front i back end projekata, prečice se korišćenjem ln komande, gde flag -sf predstavlja soft link. (slika 25)



```

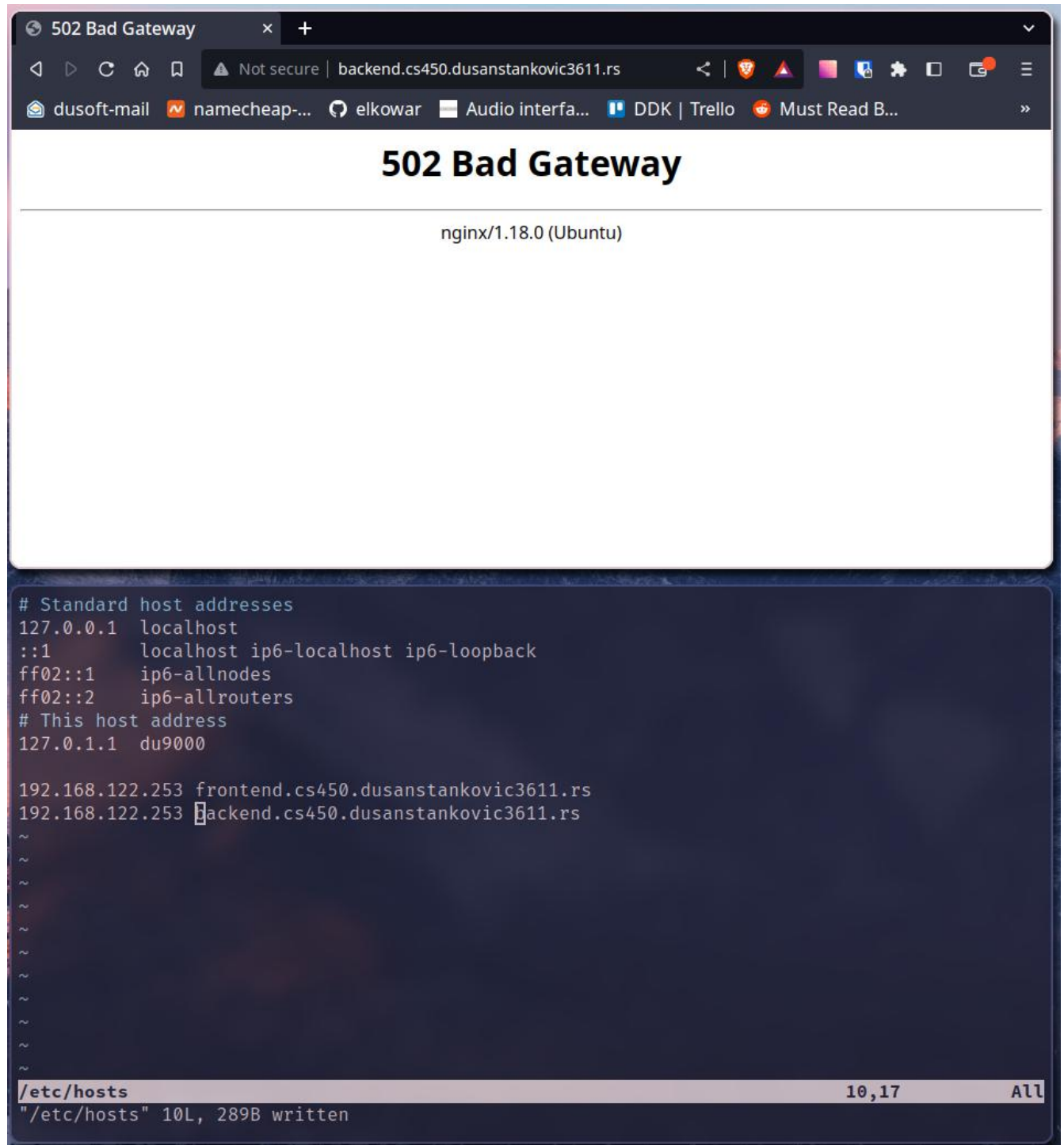
du@duserver:~$ history | tail -n 6
 8  sudo vim /etc/nginx/sites-available/frontend
 9  sudo vim /etc/nginx/sites-available/backend
10  sudo ln -sf /etc/nginx/sites-available/frontend /etc/nginx/sites-enabled/
11  sudo ln -sf /etc/nginx/sites-available/backend /etc/nginx/sites-enabled/
12  sudo nginx -s reload

```

Slika 25. Korišćenje komandi za kreiranje konfiguracija i kopiranje istih u sites-enabled

Ovde je prikazana mock konfiguracija koja nam omogućava da vežemo različite domene za IP adresu servera koji je podignut kao virtuelna mašina, bez potrebe da kupujemo domen i prolazimo kroz komplikovan proces njegove validacije.

Ovo je privremeno rešenje, i za realnu aplikaciju bismo kupili domen i konfigurisali javno dostupnu virtuelnu mašinu (VPS). Kao što se vidi na slici, odlaskom na "backend.cs450.dusanstankovic3611.rs" pristupamo nginx serveru koji smo prethodno konfigurisali, ali naravno dobijamo 502 Bad Gateway response, zbog toga što backend aplikacija još uvek nije raspoređena na virtuelnu mašinu. (slika 26)



Slika 26. Dodavanje mock hosts konfiguracije i pristupanje nginx serveru

Nakon pokretanja "npm run build" smo uspešno build-ovali frontend aplikaciju kao što se vidi na (slika 27).

```
> frontend-app-deploy-demo@0.0.0 build
> ng build

✓ Browser application bundle generation complete.
✓ Copying assets complete.
✓ Index html generation complete.
```

Initial Chunk Files	Names	Raw Size	Estimated Transfer Size
main.31f981fef87fde5b.js	main	93.97 kB	28.36 kB
polyfills.f20fb57fcf2cb57b.js	polyfills	33.12 kB	10.71 kB
runtime.507c9b3b66a784b0.js	runtime	926 bytes	522 bytes
styles.ef46db3751d8e999.css	styles	0 bytes	-
Initial Total		127.99 kB	39.58 kB

```
Build at: 2023-06-22T19:17:11.490Z - Hash: 9dbb66cdba8c09fb - Time: 12751ms
```

Slika 27. Build frontend aplikacije

Rsync je alat i biblioteka za sinhronizaciju i prenos podataka između lokalnih i udaljenih sistema. On je posebno koristan za efikasno kopiranje i ažuriranje fajlova između različitih lokacija.

Rsync se može koristiti za sinhronizaciju fajlova na istom računaru, između računara u lokalnoj mreži ili čak između udaljenih računara preko mreže. To radi tako što analizira fajlove na izvornoj i ciljnoj lokaciji i samo prenosi promenjene delove fajlova umesto celokupnog sadržaja. Ovo omogućava brže i efikasnije prenose, posebno kada se radi sa velikim fajlovima ili kada je mrežna veza sporija.

Flag koji koristimo je "-havz". Evo šta svaki od ovih flagova predstavlja:

"-h" flag omogućava da se prikazuju veličine fajlova u ljudski čitljivom formatu, sa prefiksima kao što su "K" (kilobajti), "M" (megabajti), "G" (gigabajti) itd. Ovo olakšava čitanje i razumevanje veličine fajlova.

"-a" flag predstavlja "archive" (arhiva) mod i koristi se za očuvanje sveobuhvatnih atributa fajlova tokom sinhronizacije. To uključuje informacije o vlasniku, grupi, dozvolama, vremenskim oznakama i drugim atributima fajlova. Ovaj flag osigurava da se svi relevantni atributi fajlova očuvaju tokom kopiranja.

"-v" flag predstavlja "verbose" (razgovorljiv) mod i omogućava da se prikazuju detaljnije informacije o toku kopiranja. Ovaj flag prikazuje informacije o svakom prenetom fajlu, kao i ukupan pregled napretka kopiranja.

"-z" flag predstavlja "compression" (kompresija) mod i omogućava kompresiju podataka tokom prenosa. Kompresija smanjuje veličinu prenetih podataka, što može biti korisno prilikom prenosa preko mreže sa ograničenim protokom ili kada je potrebno smanjiti opterećenje mreže.



Kombinacija ovih flagova ("-havz") omogućava čitljiv ispis veličina fajlova, očuvanje atributa fajlova, detaljno praćenje napretka i kompresiju podataka tokom prenosa pomoću rsync-a.

Pokretanjem komande "sudo mkdir /var/www/html/frontend" na serveru smo kreirali folder na kome će biti podignut frontend, i koristeći rsync biblioteku kopiramo build image sa lokalne mašine na server (slika 28)

```
du@du9000 ~/.local/src/uni/projects/CS450/frontend-app-deploy-demo $ rsync -havz dist/frontend-app-deploy-demo/ root@backend.cs450.dusanstankovic3611.rs:/var/www/html/frontend/
root@backend.cs450.dusanstankovic3611.rs's password:
sending incremental file list
./
3rdpartylicenses.txt
favicon.ico
index.html
main.31f981fef87fde5b.js
polyfills.f20fb57fcf2cb57b.js
runtime.507c9b3b66a784b0.js
styles.ef46db3751d8e999.css

sent 53.34K bytes  received 152 bytes  11.89K bytes/sec
total size is 144.83K  speedup is 2.71
du@du9000 ~/.local/src/uni/projects/CS450/frontend-app-deploy-demo $
```

Slika 28. rsync komanda za prenos image-a na server

Pokretanjem komande "mvn package" u root-u backend projekta kompajliramo i pakujemo kod u izvršivi .jar (slika 29).

```
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:3.3.0:jar (default-jar) @ backend ---
[INFO] Building jar: /home/du/.local/src/uni/projects/CS450/backend-app-deploy-demo/target/backend-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:3.1.1:repackage (repackage) @ backend ---
[INFO] Replacing main artifact /home/du/.local/src/uni/projects/CS450/backend-app-deploy-demo/target/backend-0.0.1-SNAPSHOT.jar with repackaged archive, adding nested dependencies in BOOT-INF/.
[INFO] The original artifact has been renamed to /home/du/.local/src/uni/projects/CS450/backend-app-deploy-demo/target/backend-0.0.1-SNAPSHOT.jar.original
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 5.056 s
[INFO] Finished at: 2023-06-22T21:58:13+02:00
[INFO] -----
du@du9000 ~/.local/src/uni/projects/CS450/backend-app-deploy-demo $
```

Slika 29. Build backend-a

S obzirom na to da je artifakt buildovanja backend projekta samo jedan .jar fajl ne moramo da koristimo rsync (rsync znatno ubrzava proces kopiranja više fajlova), već možemo jednostavno koristiti secure copy (scp). (slika 30)

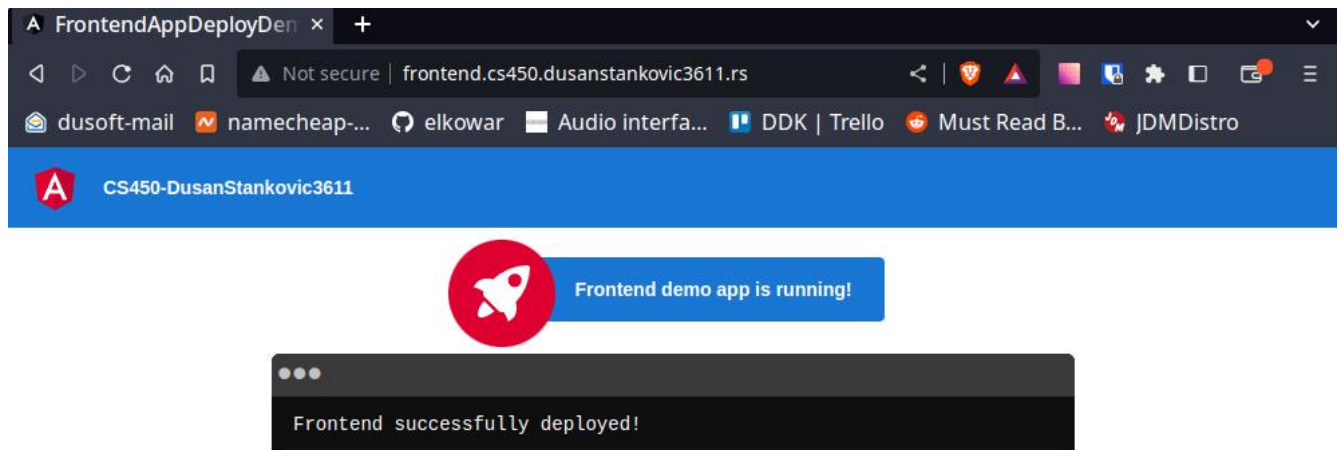


```
du@du9000 ~/.local/src/uni/projects/CS450/backend-app-deploy-demo master scp target/
backend-0.0.1-SNAPSHOT.jar root@backend.cs450.dusanstankovic3611.rs:
root@backend.cs450.dusanstankovic3611.rs's password:
backend-0.0.1-SNAPSHOT.jar 100% 18MB 128.8MB/s 00:00
du@du9000 ~/.local/src/uni/projects/CS450/backend-app-deploy-demo master
```

Slika 30. Kopiranje backend-a na server

## 7 ANALIZA I PRIKAZ POSTIGNUTIH REZULTATA

Nakon kopiranja izbuildovanog frontend projekta na server, sada možemo da pristupimo projektu sa lokalne mašine (slika 31).



Slika 31. Rezultat pristupanja podignutom frontendu na serveru sa lokalne mašine

Sada kada na serveru imamo upakovan .jar fajl, možemo ga pokrenuti koristeći komandu `java -jar [naziv paketa]`, na sledećem snimku ekrana vidimo uspešan build jednostavnog java spring boot projekta na serveru. (slika 32).

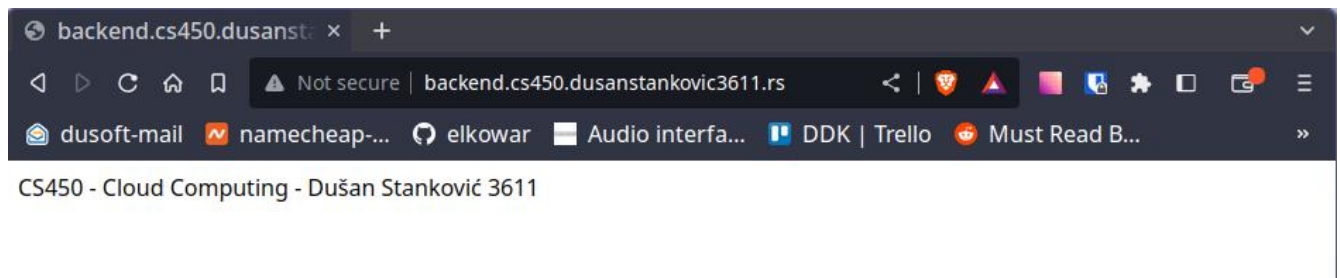
```
root@duserver:~# java -jar backend-0.0.1-SNAPSHOT.jar

:: Spring Boot :: (v3.1.1)

2023-06-22T20:01:36.983Z INFO 2178 --- [main] r.d.c.b.BackendAppDeployDemoApplication : Starting BackendAppDeployDe
moApplication v0.0.1-SNAPSHOT using Java 17.0.7 with PID 2178 (/root/backend-0.0.1-SNAPSHOT.jar started by root in /root)
2023-06-22T20:01:36.985Z INFO 2178 --- [main] r.d.c.b.BackendAppDeployDemoApplication : No active profile set, fall
ing back to 1 default profile: "default"
2023-06-22T20:01:37.854Z INFO 2178 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with por
t(s): 8080 (http)
2023-06-22T20:01:37.863Z INFO 2178 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-06-22T20:01:37.863Z INFO 2178 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [A
pache Tomcat/10.1.10]
2023-06-22T20:01:37.981Z INFO 2178 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedde
d WebApplicationContext
2023-06-22T20:01:37.984Z INFO 2178 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext:
initialization completed in 833 ms
2023-06-22T20:01:38.274Z INFO 2178 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s):
8080 (http) with context path ''
2023-06-22T20:01:38.307Z INFO 2178 --- [main] r.d.c.b.BackendAppDeployDemoApplication : Started BackendAppDeployDem
oApplication in 1.729 seconds (process running for 2.817)
```

Slika 32. Build .jar fajla na serveru

I na kraju vidimo rezultat java aplikacije kada pristupimo adresi virtuelne mašine. (slika 33)



Slika 33. Rezultat pristupanja podignutom backendu na serveru sa lokalne mašine

## 8 ZAKLJUČAK

U ovom projektu iz predmeta CS450 "Cloud Computing", implementirali smo rešenje za unapređenje poslovnog sistema u kompaniji. Identifikovali smo problem u korišćenju računara firme za sve potrebe poslovanja, gde je bilo potrebno imati pristup istim fajlovima i rasterećenje resursa pri radu.

Kroz analizu postojećeg informacionog sistema organizacije, uočili smo mane u čuvanju projekata na više računara, što je rezultiralo zauzimanjem prostora i gubitkom resursa pri pokretanju.

Da bismo rešili ove izazove, predložili smo implementaciju cloud computing arhitekture sa centralnim serverom zasnovanim na Ubuntu Server operativnom sistemu. Koristili smo

Projektni zadatak: Naziv projektnog zadatka

alate poput SSH i Nginx za omogućavanje pristupa računarima samo autorizovanim korisnicima i za usmeravanje zahteva ka frontendu i backendu aplikacija.

Kroz korake instalacije Ubuntu Servera, konfiguraciju SSH ključeva, upotrebu Nginx servera za serviranje frontend i backend aplikacija, kao i korišćenje alata poput rsync za sinhronizaciju podataka, postigli smo efikasniji rad, smanjenje resursa i olakšali pristup i upravljanje projektima.

Ovaj projekat nam je pružio priliku da se upoznamo sa cloud computing konceptima i tehnologijama, kao i da primenimo stečeno znanje u stvarnom poslovnom okruženju. Implementirano rešenje pruža efikasniju i skalabilniju infrastrukturu, smanjuje gubitak resursa i omogućava bolje upravljanje podacima. Kroz ovaj projekat, stekli smo dublje razumevanje cloud computinga i njegovog uticaja na poslovne sisteme.

U zaključku, ovaj projekat je bio koristan u razumevanju i primeni cloud computinga u kontekstu unapređenja poslovnog sistema. Kroz implementaciju centralizovanog servera i korišćenje različitih alata, postigli smo bolju organizaciju, efikasniji rad i smanjenje resursa, što će rezultirati poboljšanjem produktivnosti i performansi u našoj kompaniji.

## 9 LITERATURA

- [1.] Đokić, D. (2018/2019) 'Cloud Computing', Beograd: Univerzitet Metropolitan.
- [2.] <https://angular.io/docs>
- [3.] <https://www.baeldung.com/spring-boot>
- [4.] <http://nginx.org/en/docs/>
- [5.] <https://rsync.samba.org/documentation.html>
- [6.] <https://archlinux.org/>
- [7.] <https://help.ubuntu.com/lts/ubuntu-help/index.html>