****

**CS450**

**CLOUD COMPUTING**

**PROJEKTNI ZADATAK:**

**Deploy aplikacija na lokalnom klaud 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](#_Toc409241091)

[2 Opšti podaci o poslovnom sistemu 4](#_Toc662421998)

[2.1 Karakteristike sistema 4](#_Toc834825848)

[2.1.1 Centralizovano skladištenje podataka 4](#_Toc1871200843)

[2.1.2 Deljenje fajlova 4](#_Toc1355064758)

[2.1.3 Remote deploy 5](#_Toc1417431764)

[2.1.4 Poboljšana produktivnost 5](#_Toc540970012)

[2.1.5 Bezbednost podataka 5](#_Toc747555214)

[3 Analiza postojećeg informacionog sistema organizacije 5](#_Toc624373139)

[3.1 Hardver postojećeg sistema 6](#_Toc1020686444)

[3.2 Softver postojećeg sistema 6](#_Toc729320578)

[4 Uporedni pregled tehnologija 7](#_Toc477196616)

[4.1 Trenutni sistem 7](#_Toc898302229)

[4.1.1 Skladištenje projekata 7](#_Toc401282034)

[4.1.2 Deljenje projekata 7](#_Toc347011528)

[4.1.3 Resursi 7](#_Toc408771613)

[4.1.4 Slanje dokumenata 7](#_Toc713244268)

[4.1.5 Podešavanje okruženja 7](#_Toc1004342092)

[4.2 sistem koji će se implemenitrati 7](#_Toc1490693446)

[4.2.1 Skladištenje projekata 7](#_Toc1619466934)

[4.2.2 Deljenje projekata 7](#_Toc352998970)

[4.2.3 Resursi 7](#_Toc673921272)

[4.2.4 Slanje dokumenata 7](#_Toc879611919)

[4.2.5 Podešavanje okruženja 7](#_Toc512371042)

[5 Predlog rešenja za unapređenje poslovnog sistema 8](#_Toc1472092083)

[5.1 Cloud skladištenje projekata 8](#_Toc1370943878)

[5.2 Centralizovani pristup projektima 8](#_Toc1367814948)

[5.3 Izvršavanje projekata na centralnom serveru 8](#_Toc2146859943)

[5.4 Centralizovano skladištenje i deljenje dokumenata 9](#_Toc1443915406)

[5.5 Podešavanje okruženja 9](#_Toc674301088)

[6 Implementacija rešenja za unapređenje poslovnog sistema 9](#_Toc1183549726)

[6.1 Instalacija i podešavanje ubuntu server 22.04.2 LTS 9](#_Toc1853156497)

[6.2 Instalacija i podešavanje softvera za cloud 23](#_Toc1336723086)

[6.3 deploy backend i frontend aplikacija 26](#_Toc2018375575)

[7 Analiza i prikaz postignutih rezultata 33](#_Toc1576873693)

[8 Zaključak 34](#_Toc544304196)

[9 Literatura 35](#_Toc1288323691)

**SPISAK SLIKA:**

[Slika 1 . Preuzimanje Ubuntu Server ISO fajla 10](#_Toc1041251023)

[Slika 2 . Odabiranje opcije za kreiranje nove virtualne mašine unutar virt managera 10](#_Toc268912332)

[Slika 3 . Odabiranje opcije za kreiranje nove virtualne mašine unutar virt managera 11](#_Toc254653552)

[Slika 4 . Alociranje resursa za virtualnu mašinu 11](#_Toc1266550794)

[Slika 5 . Odabiranje ISO-a za instalaciju Ubuntu Servera 12](#_Toc255783478)

[Slika 6 . Odabiranje pool-a za novi volume 12](#_Toc1681637832)

[Slika 7 . Odabiranje količine memorije za novi volume 13](#_Toc1560323579)

[Slika 8 . Odabiranje opcije za custom storage 13](#_Toc1934259007)

[Slika 9 . Potvrda sumarizovanih podešavanja 14](#_Toc2026354498)

[Slika 10 . Odabir jezika 14](#_Toc841070720)

[Slika 11 . Odabir keyboard layout-a 15](#_Toc311488745)

[Slika 12 . Odabir vrste instalacije 15](#_Toc1230189638)

[Slika 13 . Podešavanje default network interface-a 16](#_Toc792509315)

[Slika 14 . Podešavanje mirror adresa 17](#_Toc43879961)

[Slika 15 . Podešavanje particija 18](#_Toc2116255213)

[Slika 16 . Podešavanje kredencijala 19](#_Toc1135294266)

[Slika 17 . Instaliranje OpenSSH servera 20](#_Toc76907269)

[Slika 18 . Logovanje na sistem 21](#_Toc620773955)

[Slika 19 . CLI nakon logovanja na sistem 22](#_Toc656238796)

[Slika 20 . Pretraga openjdk paketa 23](#_Toc1264509808)

[Slika 21 . Instaliranje openjdk-až 24](#_Toc1242932125)

[Slika 22 . Instaliranje nginx 25](#_Toc309120258)

[Slika 23 . Kreiranje nginx konfiguracije za usmeravanje zahteva ka frontend-u 27](#_Toc86784520)

[Slika 24 . Kreiranje nginx konfiguracije za usmeravanje zahteva ka backend-u 28](#_Toc1483134045)

[Slika 25 . Korišćenje komandi za kreiranje konfiguracija i kopiranje istih u sites-enabled 28](#_Toc74672185)

[Slika 26 . Dodavanje mock hosts konfiguracije i pristupanje nginx serveru 30](#_Toc2029040283)

[Slika 27 . Build frontend aplikacije 31](#_Toc1907617671)

[Slika 28 . rsync komanda za prenos image-a na server 32](#_Toc1921087236)

[Slika 29 . Build backend-a 32](#_Toc708866432)

[Slika 30 . Kopiranje backend-a na server 33](#_Toc559268186)

[Slika 31 . Rezultat pristupanja podignutom frontendu na serveru sa lokalne mašine 33](#_Toc875201846)

[Slika 32 . Build .jar fajla na serveru 34](#_Toc1687660587)

[Slika 33 . Rezultat pristupanja podignutom backendu na serveru sa lokalne mašine 34](#_Toc951285403)

# 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.

# 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.

## 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.

### 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.

### 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.

### 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.

### 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.

### 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.

# 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.

## Hardver postojećeg sistema

Spisak svih računara koje koristim u svrhe poslovanja:

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

## 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.

# Uporedni pregled tehnologija

## Trenutni sistem

### Skladištenje projekata

Projekti se čuvaju lokalno na svakom računaru.

### Deljenje projekata

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

### Resursi

Pokretanje projekata zahteva resurse na svakom računaru.

### Slanje dokumenata

Dokumenti se ručno šalju mejlom sa jednog računara na drugi.

### Podešavanje okruženja

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

## sistem koji će se implemenitrati

### Skladištenje projekata

Projekti će biti smešteni na centralnom serveru u cloud okruženju.

### Deljenje projekata

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

### Resursi

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

### Slanje dokumenata

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

### Podešavanje okruženja

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

# 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:

## 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.

## 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.

## 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.

## 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.

## 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.

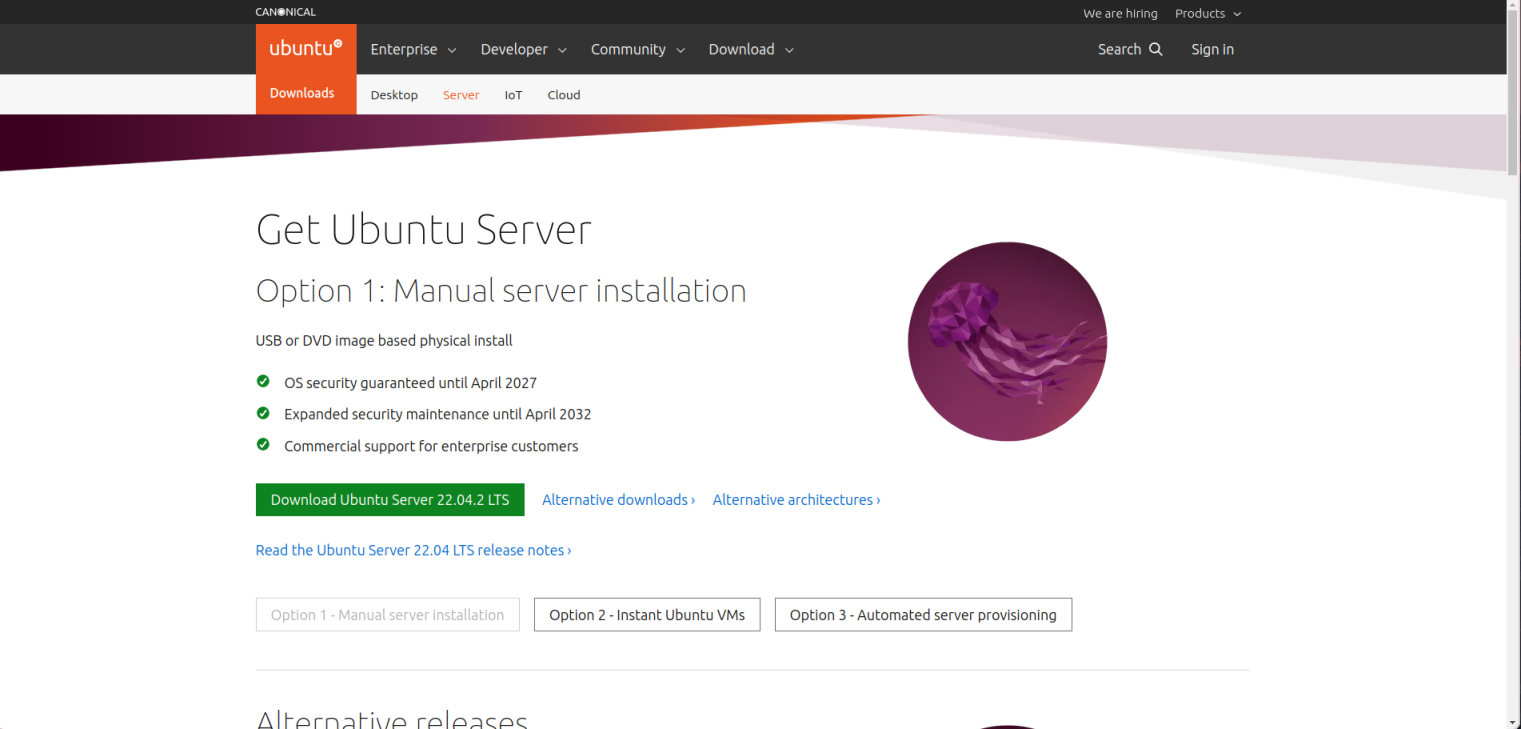
# Implementacija rešenja za unapređenje poslovnog sistema

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

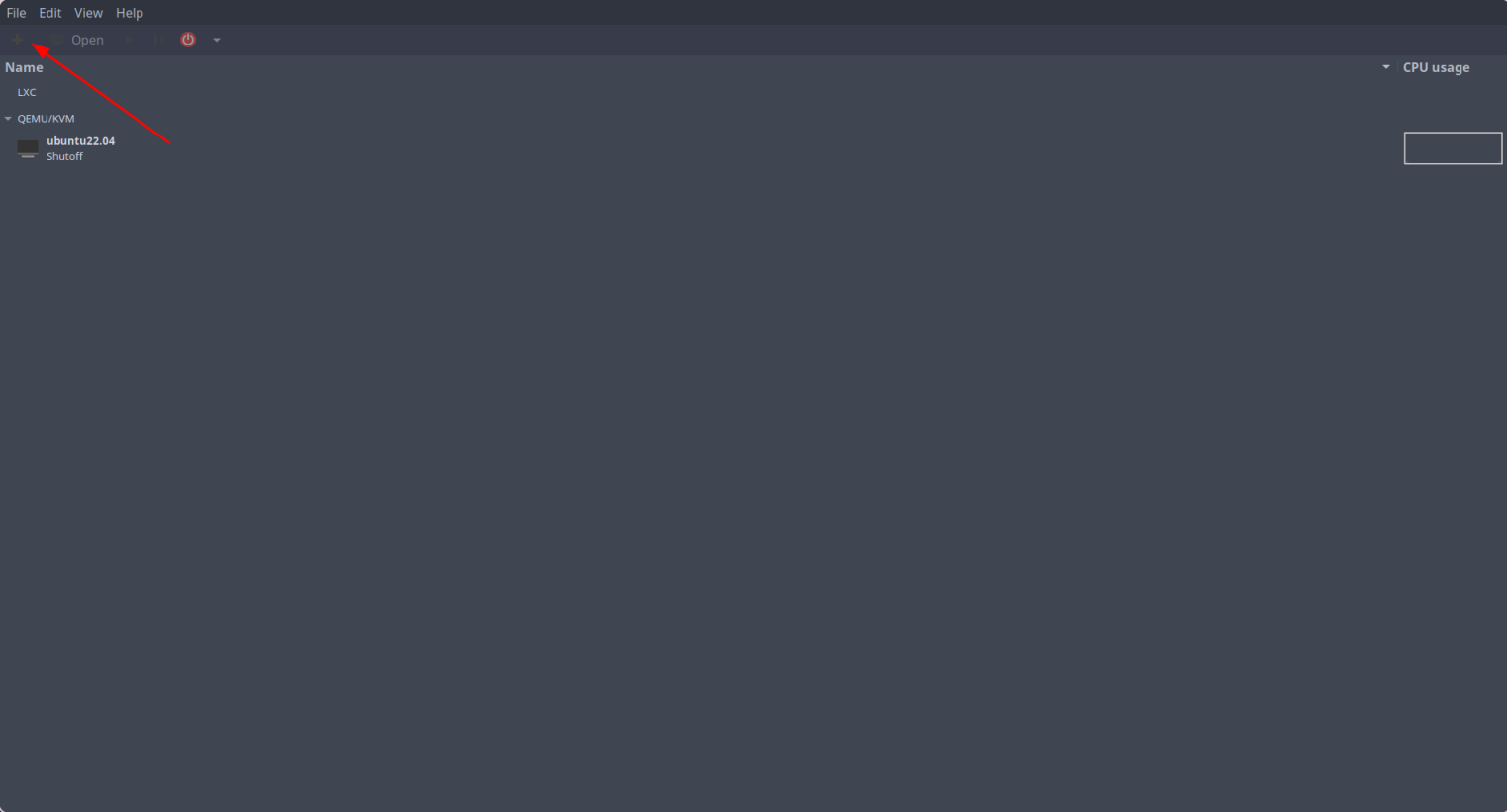
## Instalacija i podešavanje ubuntu server 22.04.2 LTS

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)



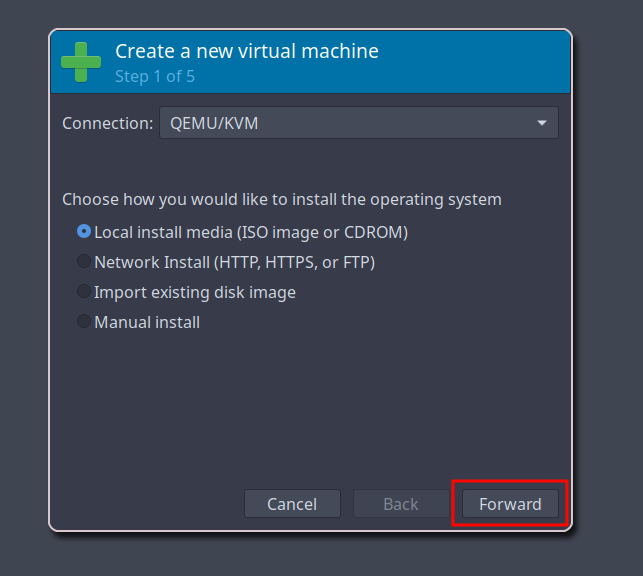
Slika 1. Preuzimanje Ubuntu Server ISO fajla

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



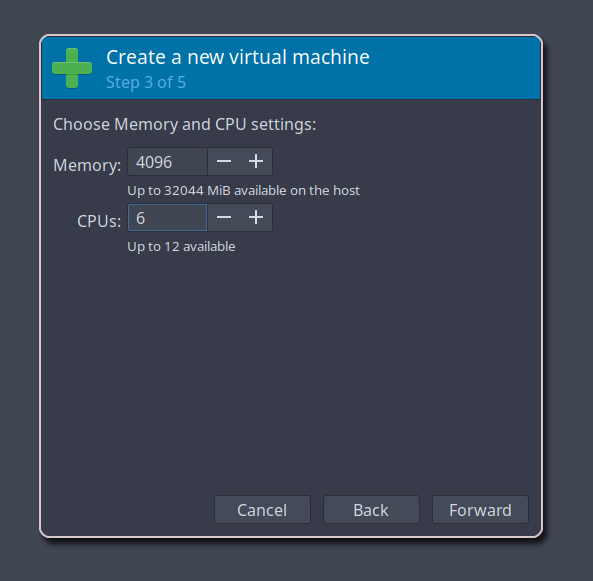
Slika 2. Odabiranje opcije za kreiranje nove virtualne mašine unutar virt managera

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



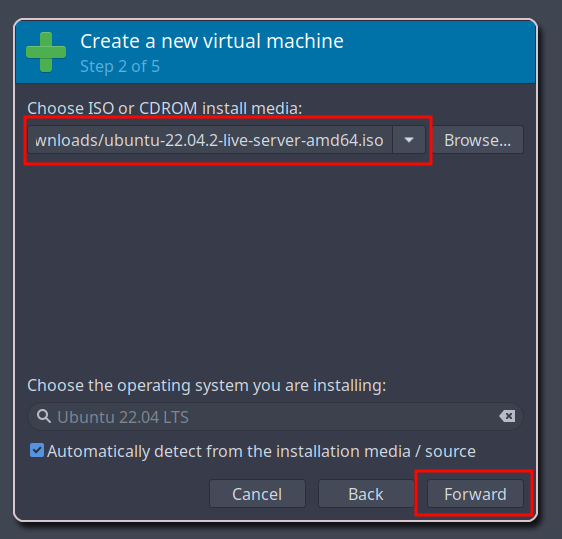
Slika 3. Odabiranje opcije za kreiranje nove virtualne mašine unutar virt managera

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)



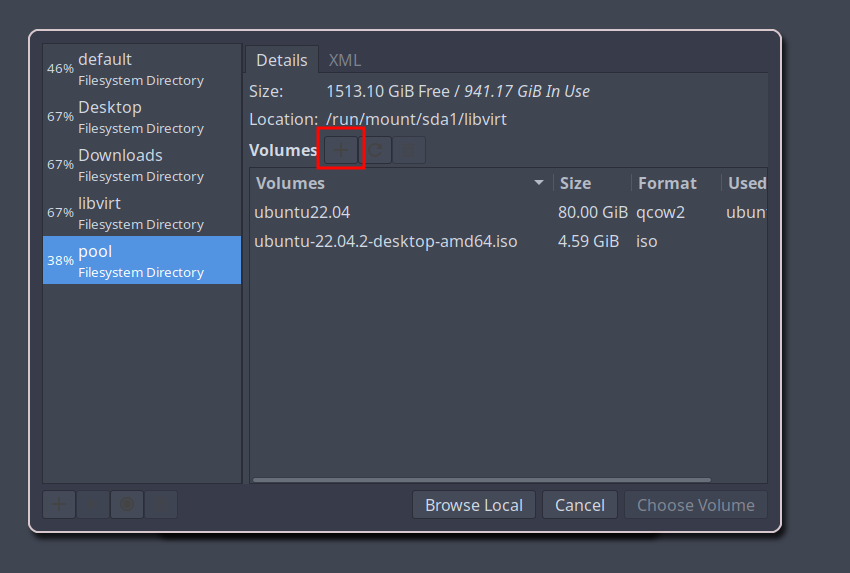
Slika 4. Alociranje resursa za virtualnu mašinu

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



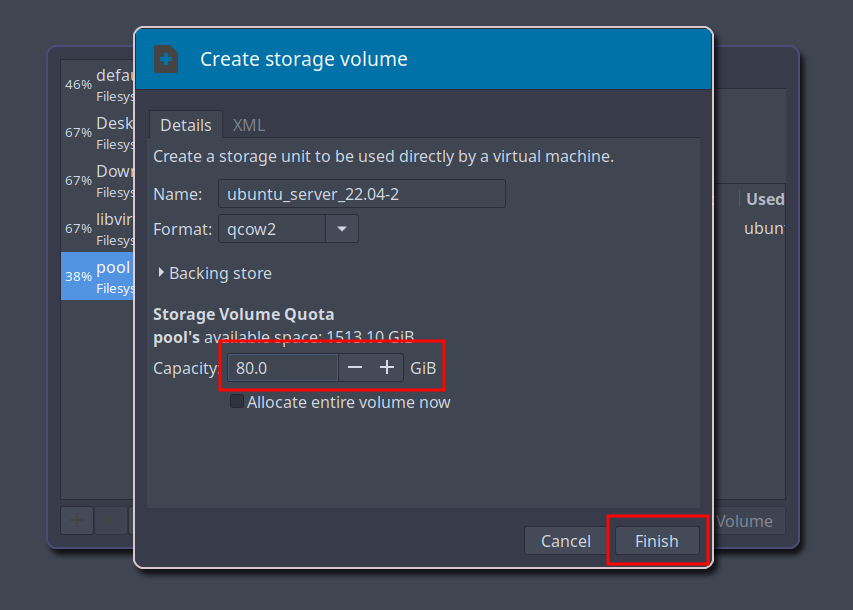
Slika 5. Odabiranje ISO-a za instalaciju Ubuntu Servera

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



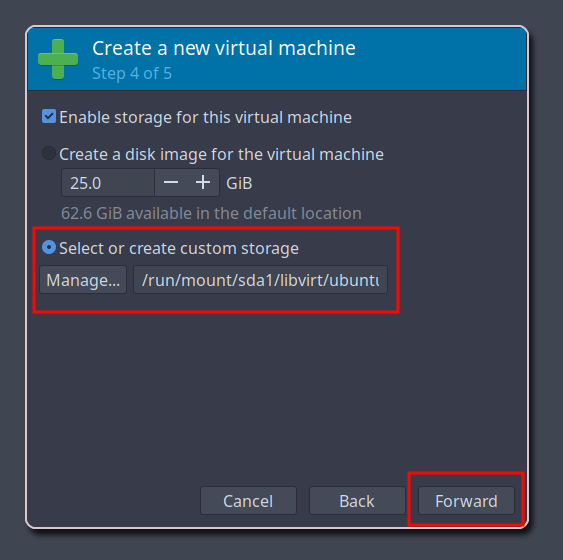
Slika 6. Odabiranje pool-a za novi volume

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



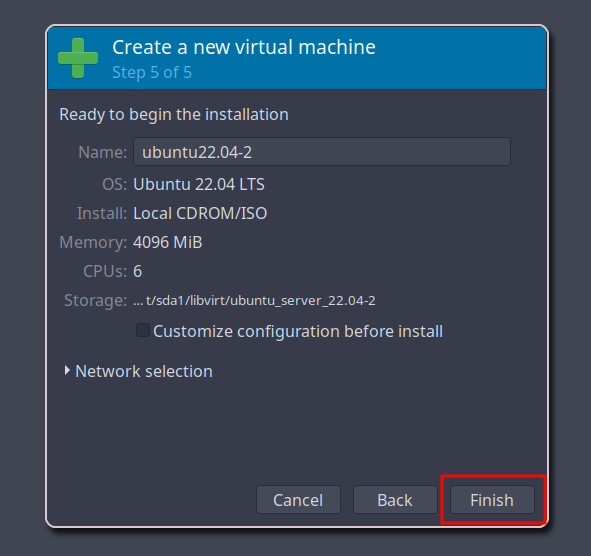
Slika 7. Odabiranje količine memorije za novi volume

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)



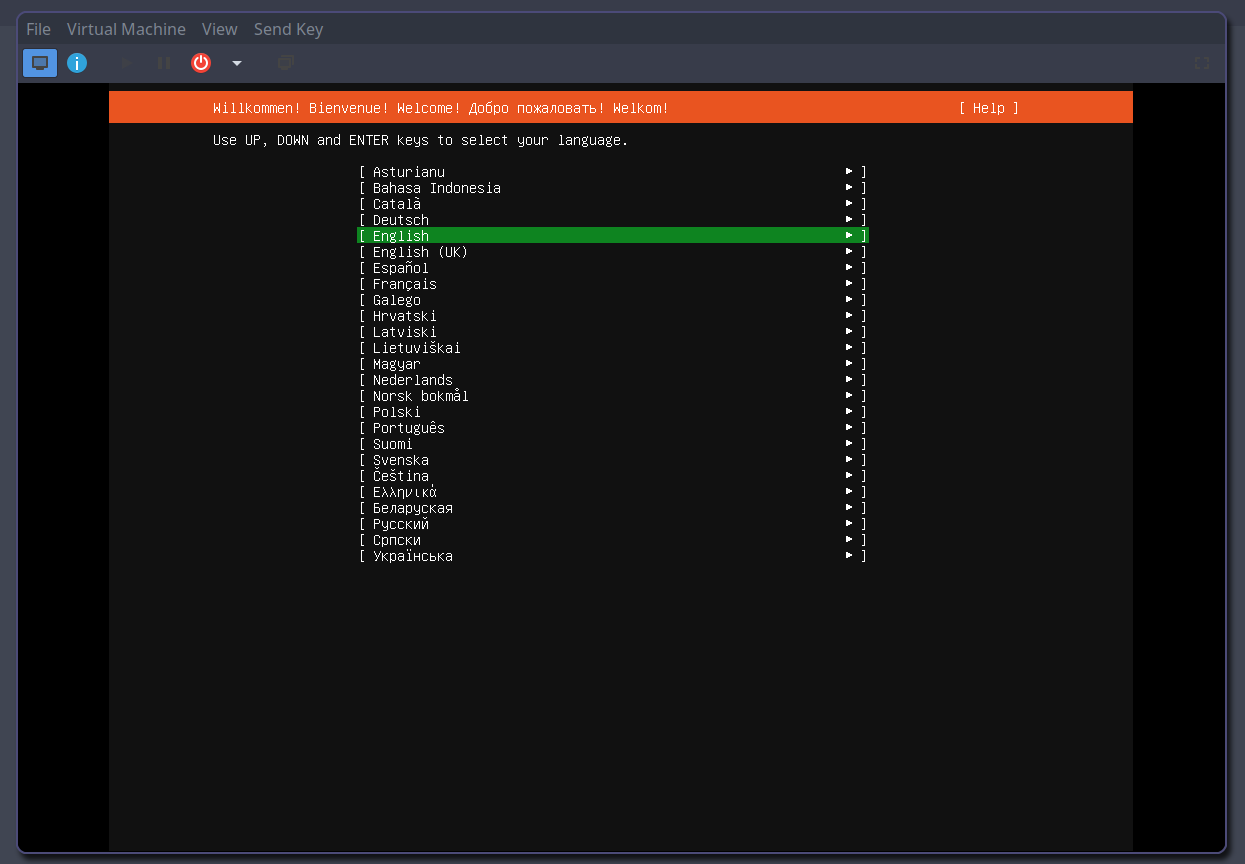
Slika 8. Odabiranje opcije za custom storage

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

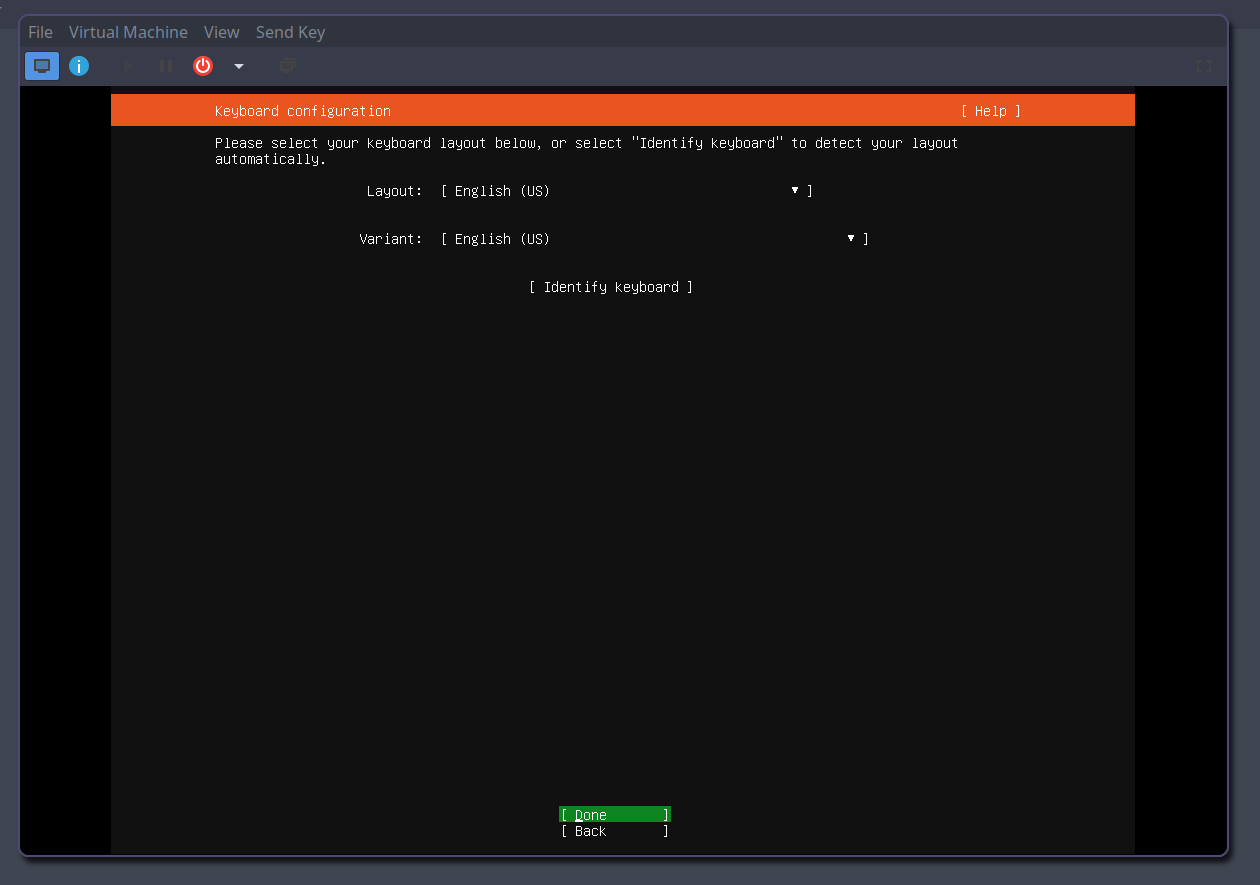


Slika 9. Potvrda sumarizovanih podešavanja

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

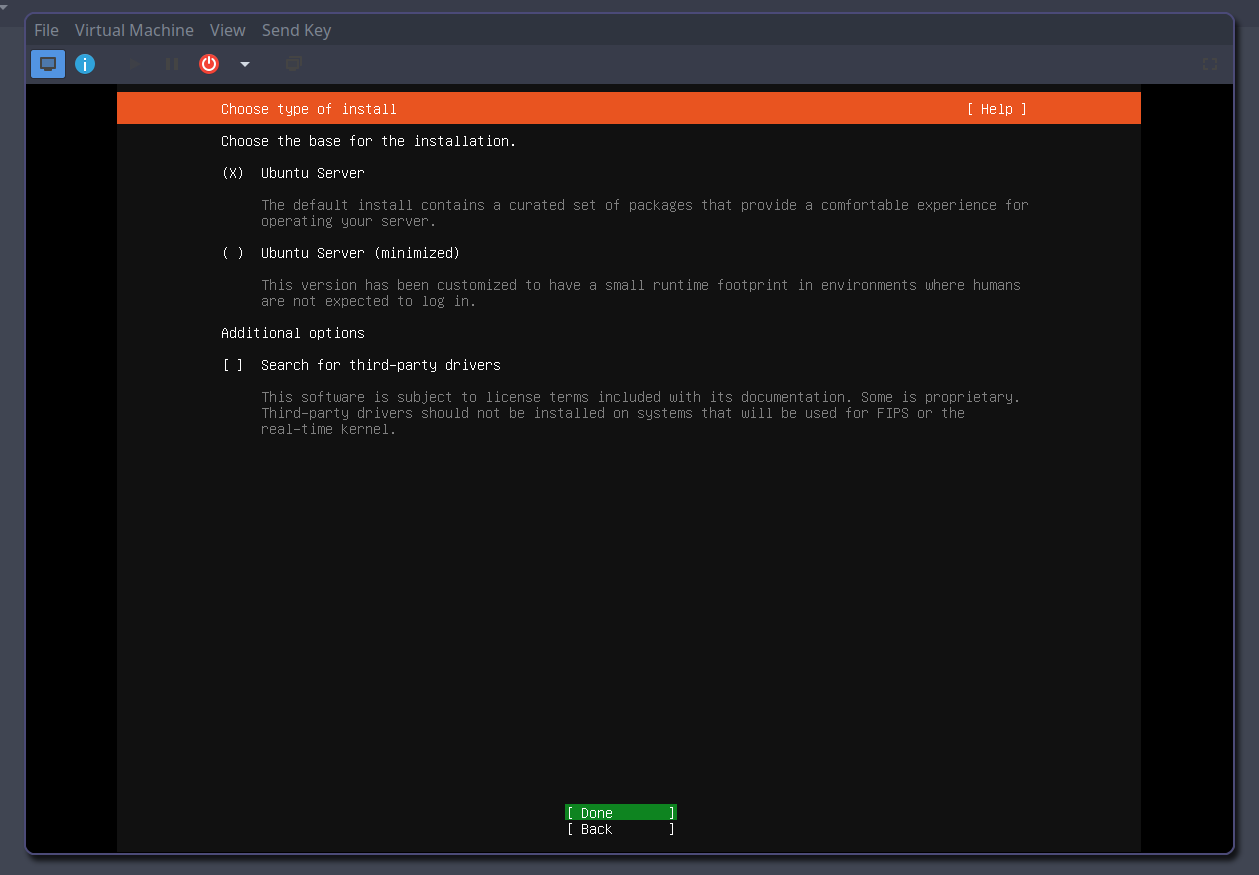


Slika 10. Odabir jezika



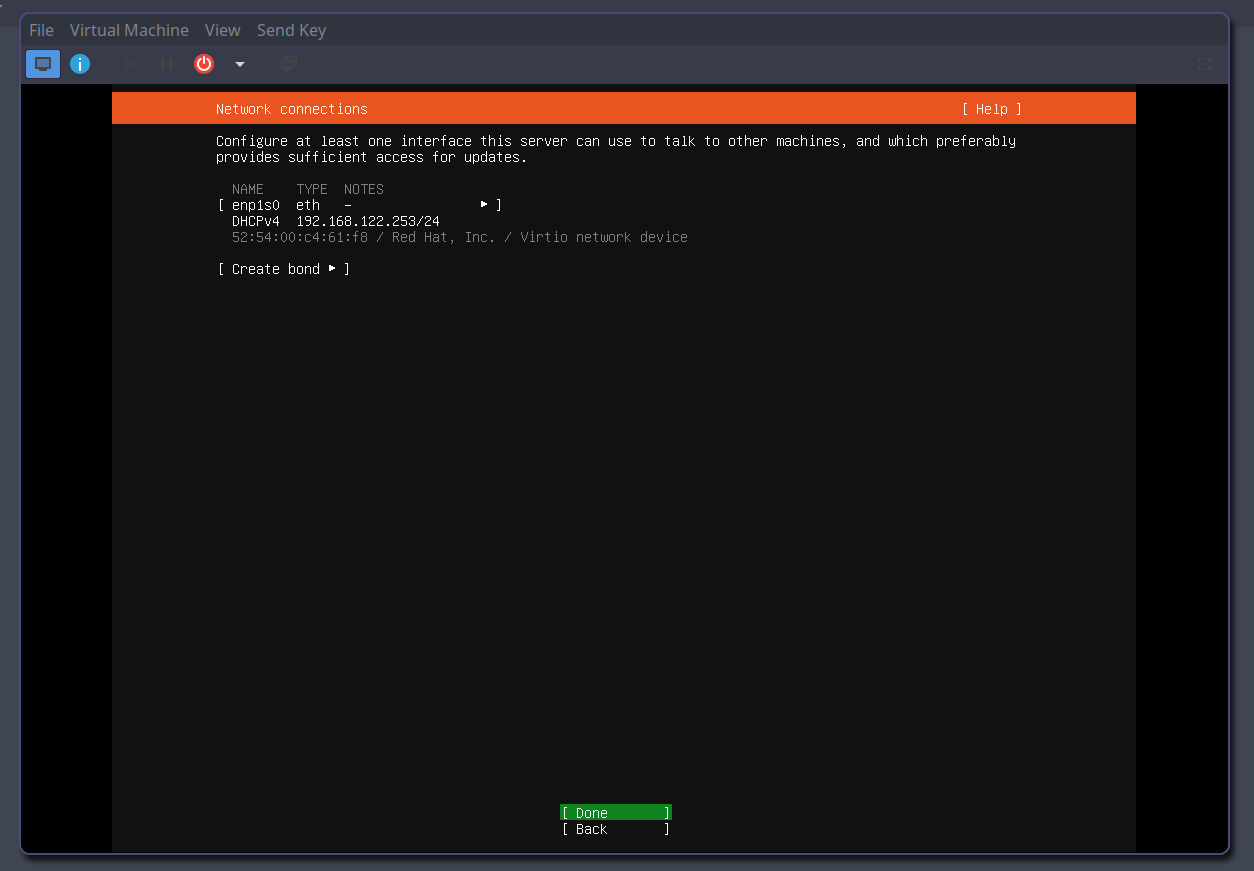
Slika 11. Odabir keyboard layout-a

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



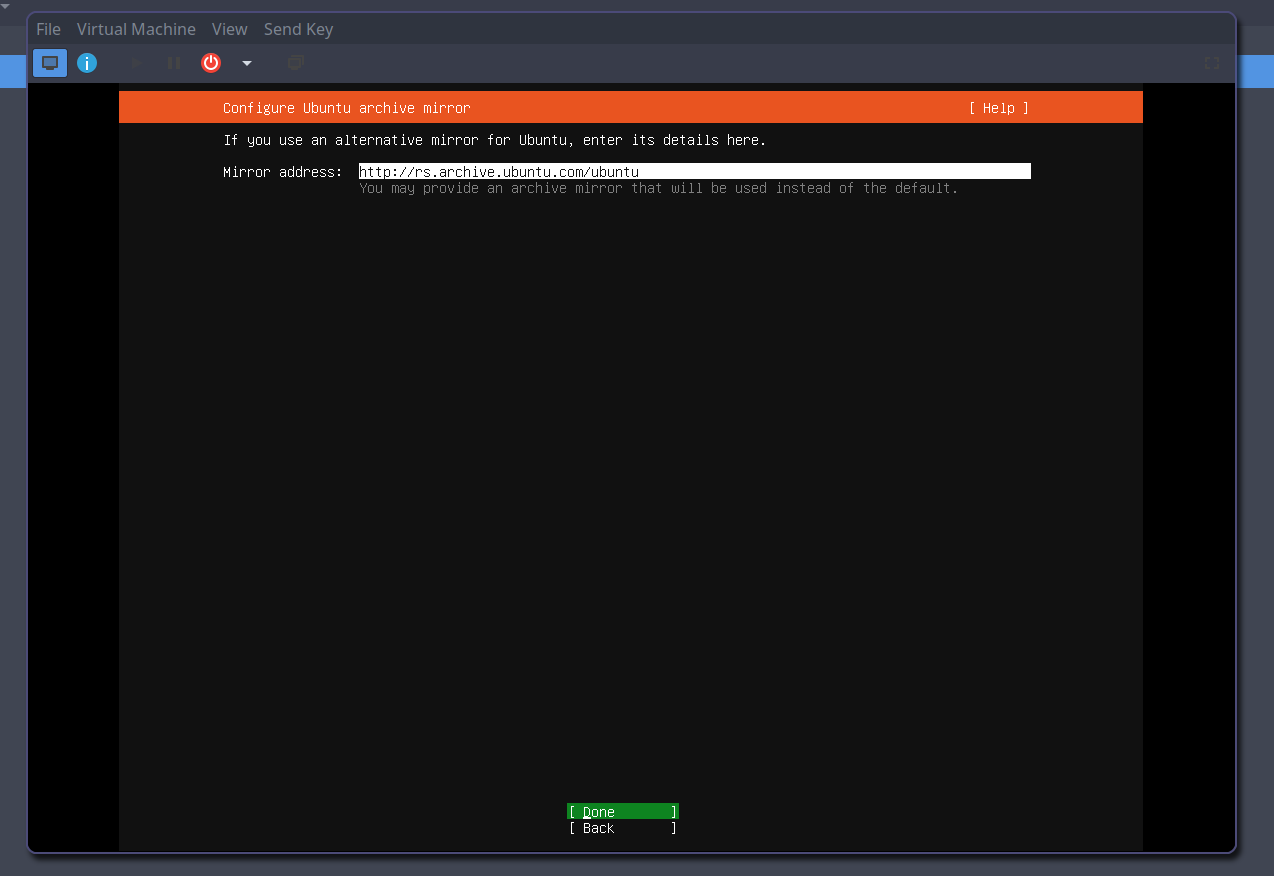
Slika 12. Odabir vrste instalacije

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)



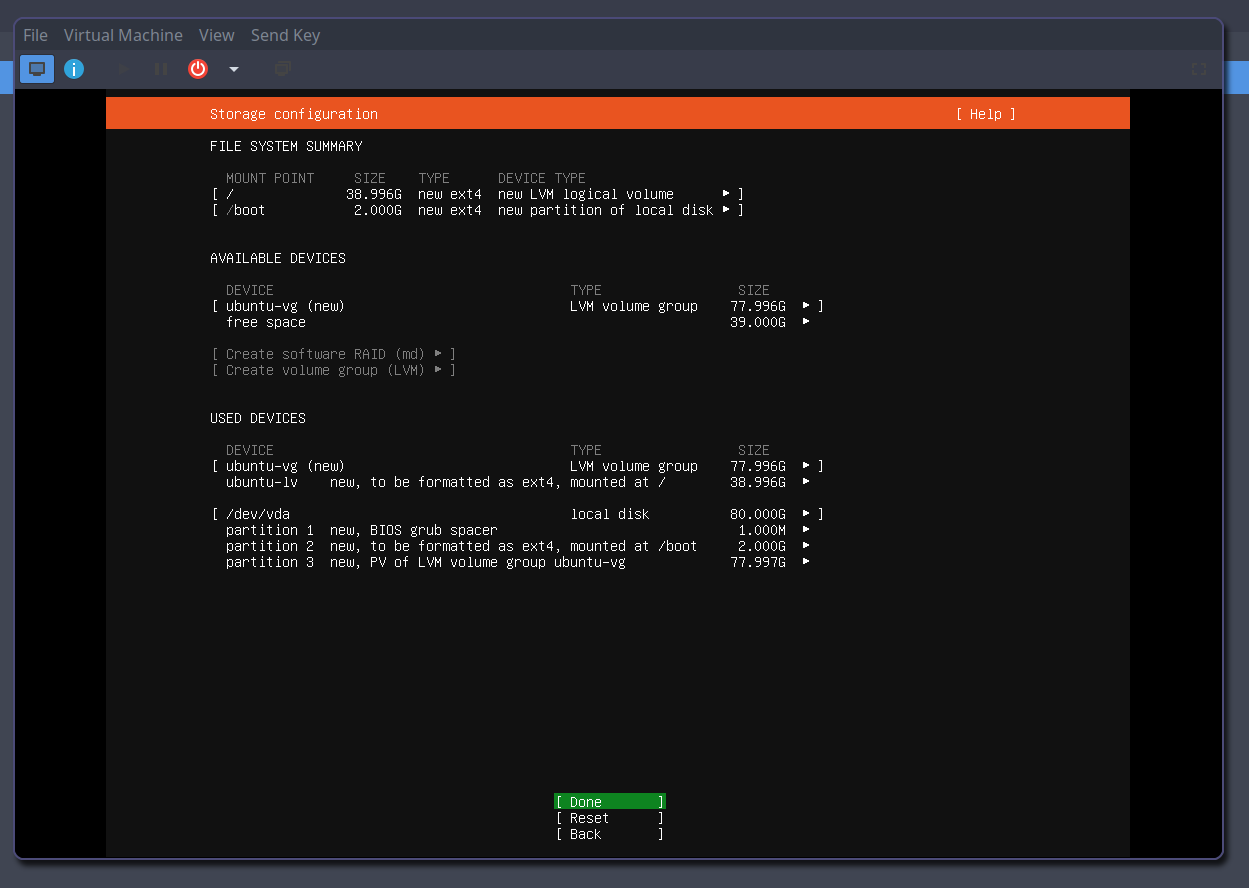
Slika 13. Podešavanje default network interface-a

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)



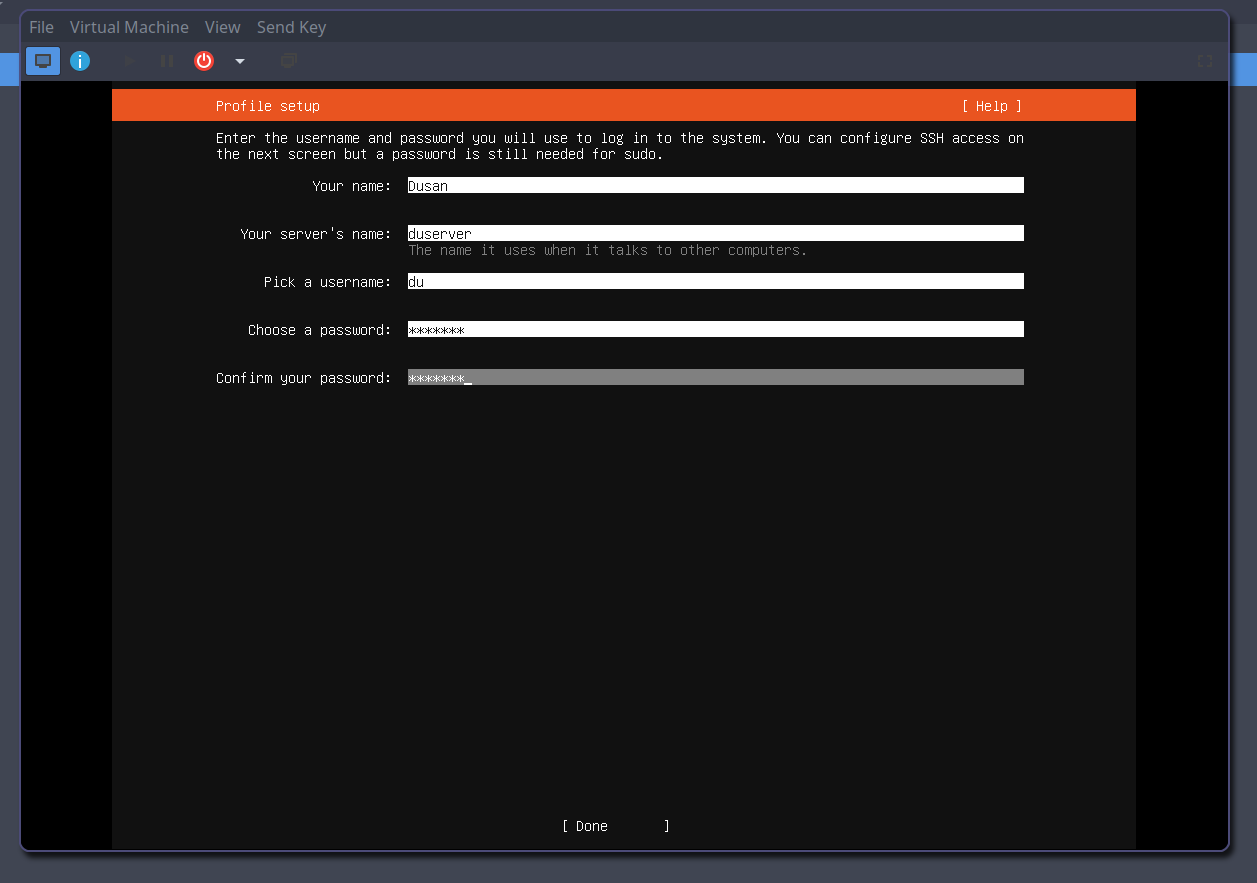
Slika 14. Podešavanje mirror adresa

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)



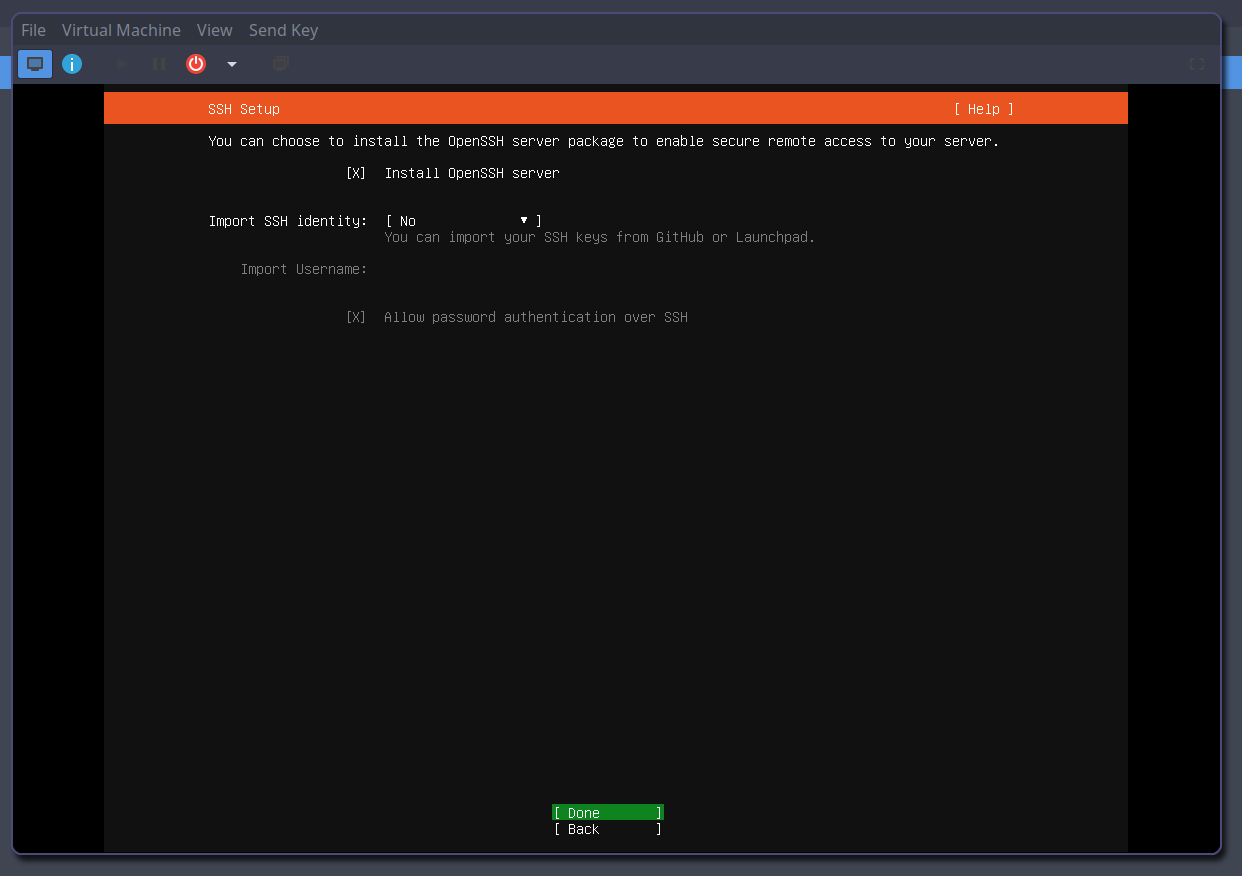
Slika 15. Podešavanje particija

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



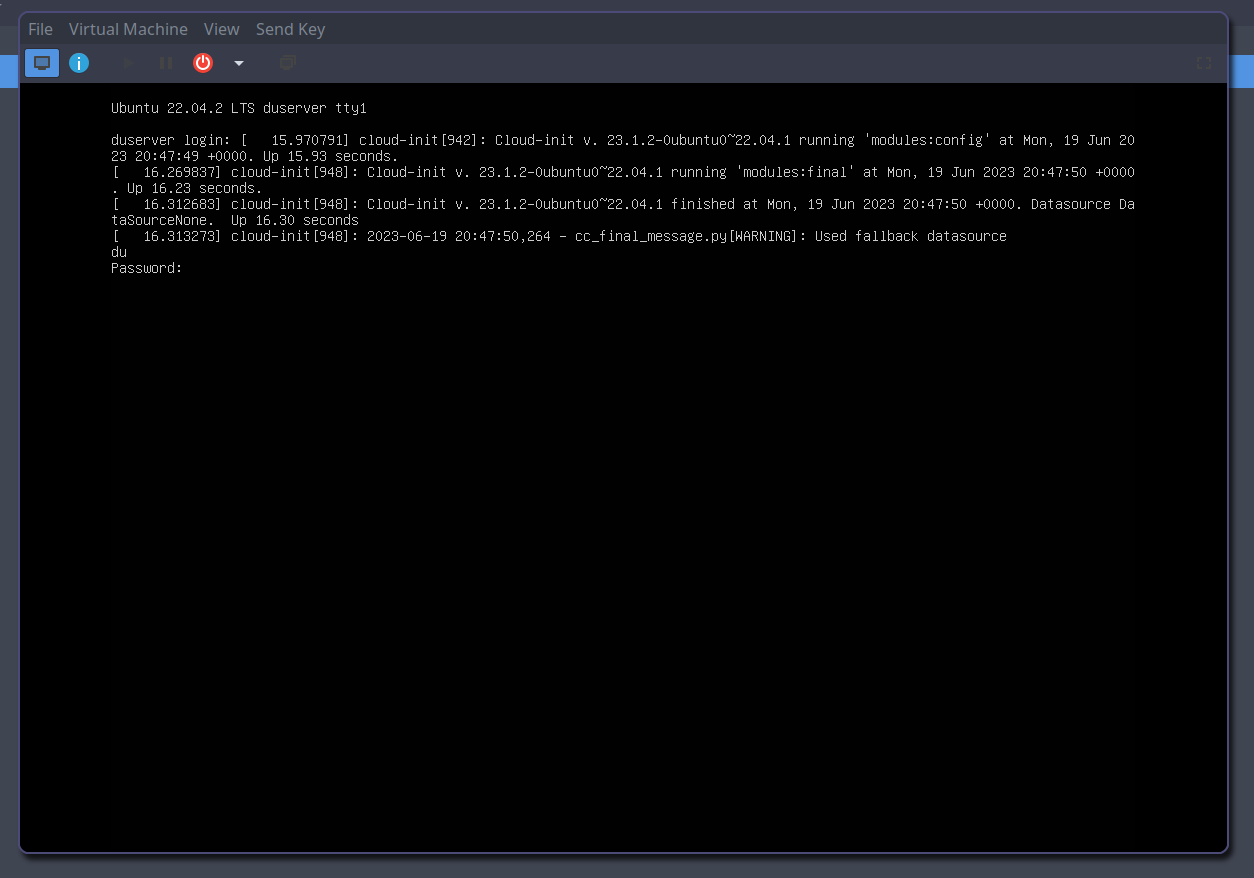
Slika 16. Podešavanje kredencijala

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

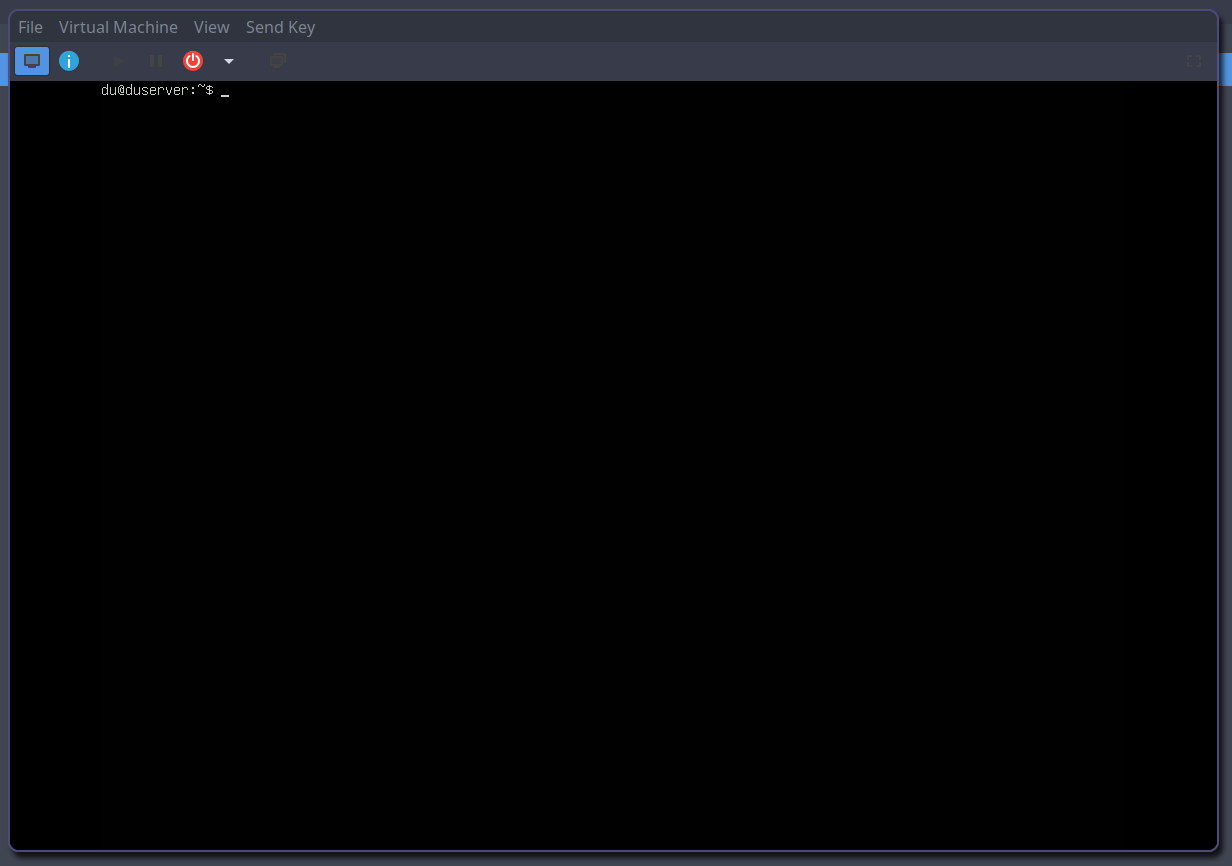


Slika 17. Instaliranje OpenSSH servera

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)



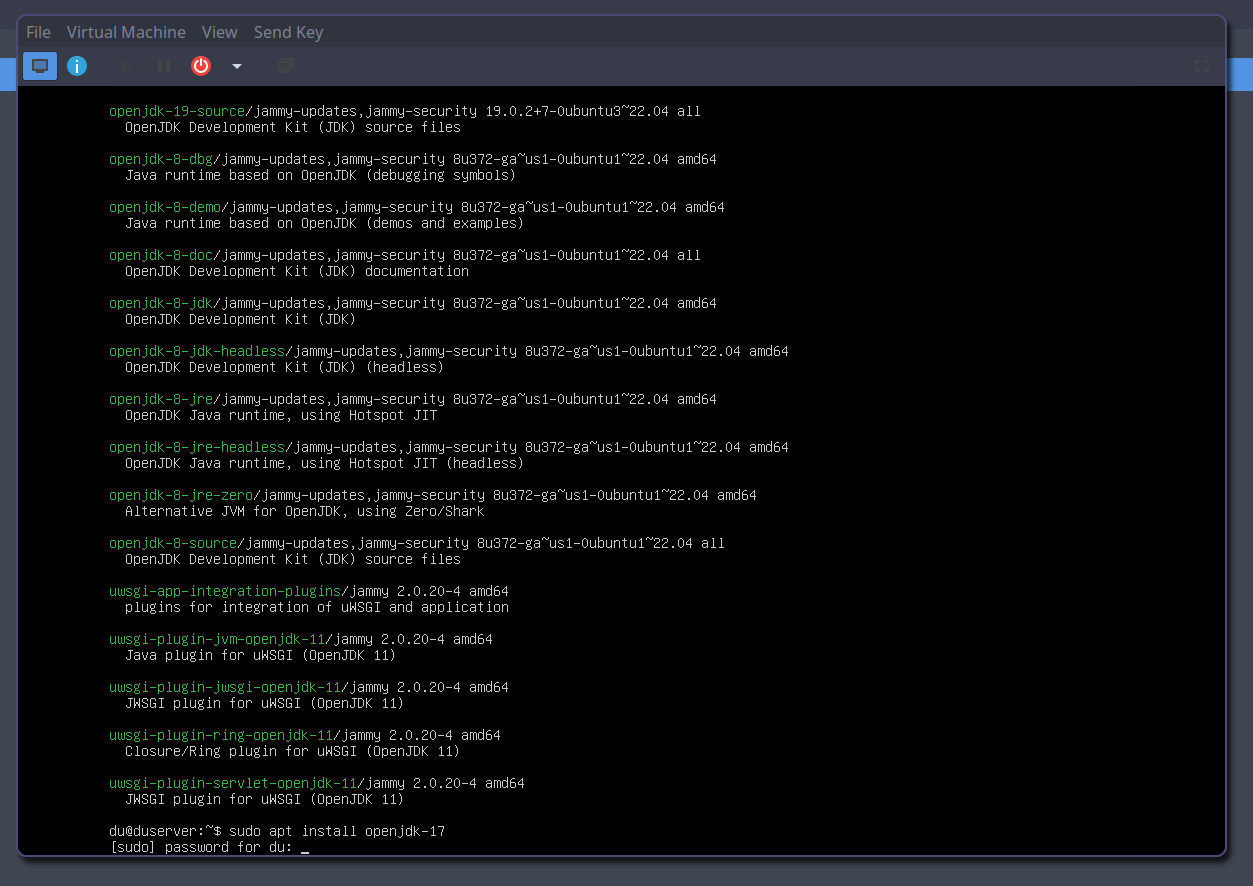
Slika 18. Logovanje na sistem



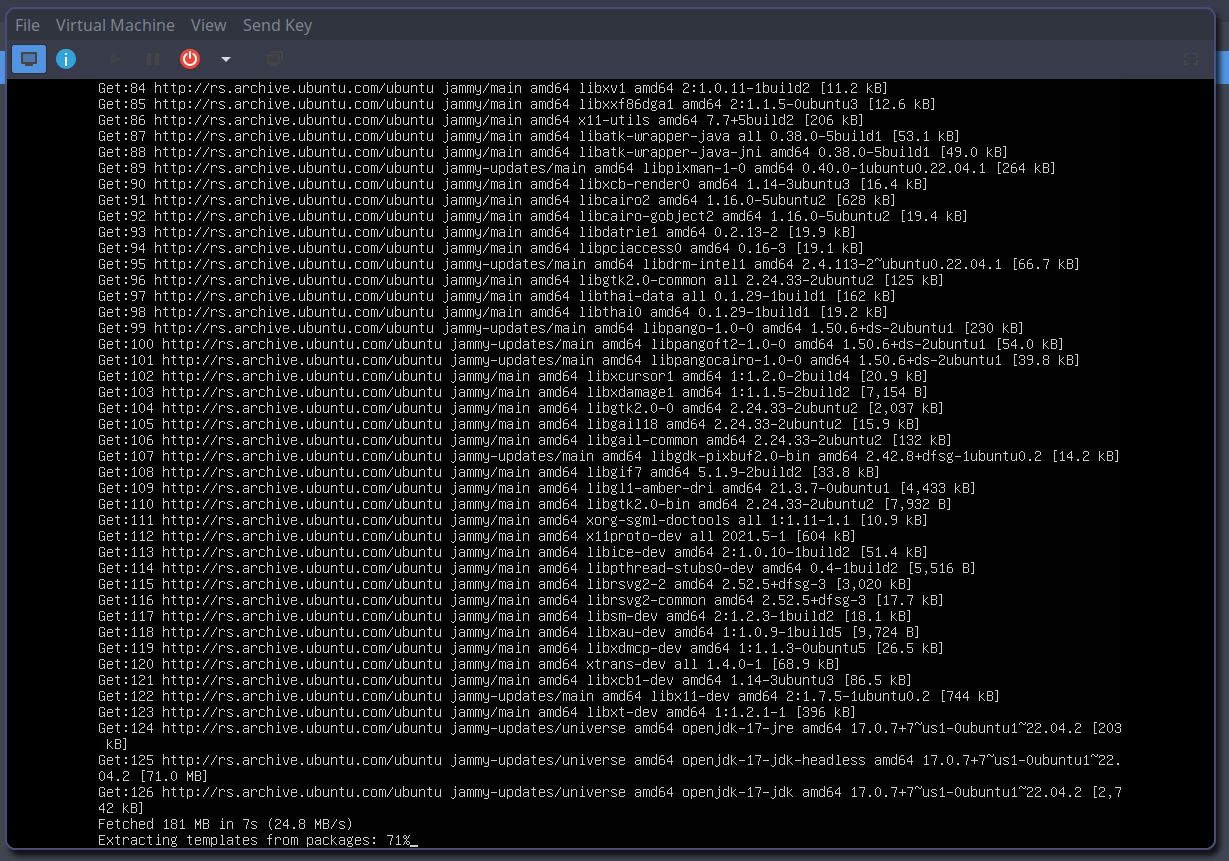
Slika 19. CLI nakon logovanja na sistem

## Instalacija i podešavanje softvera za cloud

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

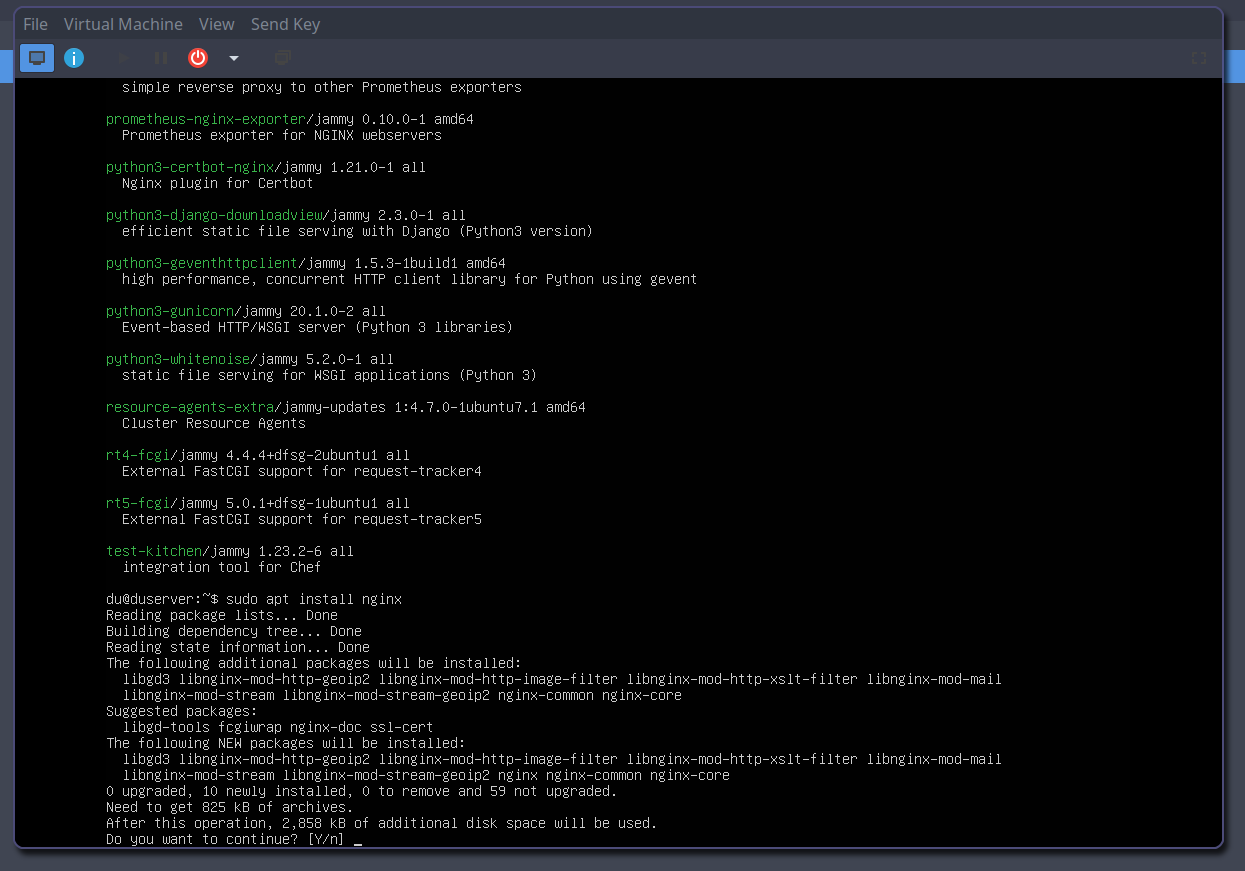


Slika 20. Pretraga openjdk paketa



Slika 21. Instaliranje openjdk-až

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



Slika 22. Instaliranje nginx

## deploy backend i frontend aplikacija

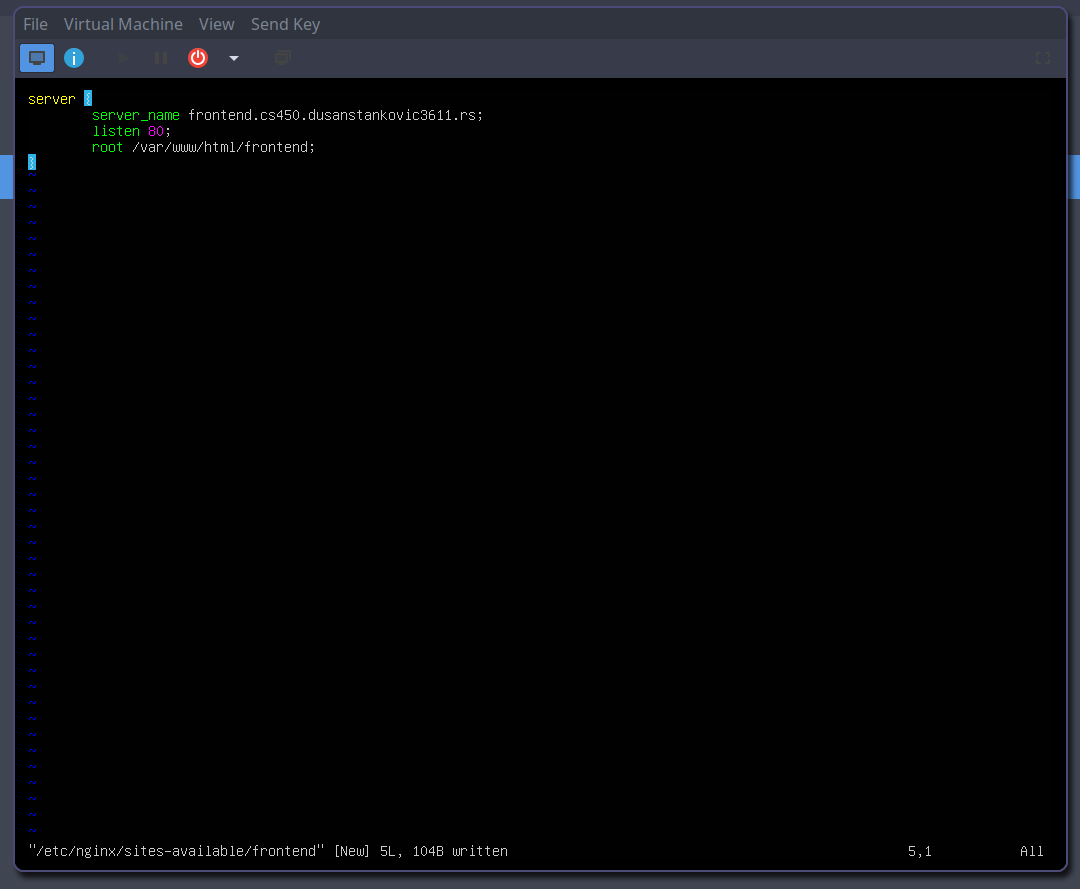
Kreiramo Nginx konfiguraciju za usmeravanje zahteva ka frontend aplikaciji sa sledećim postavkama:

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)



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

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

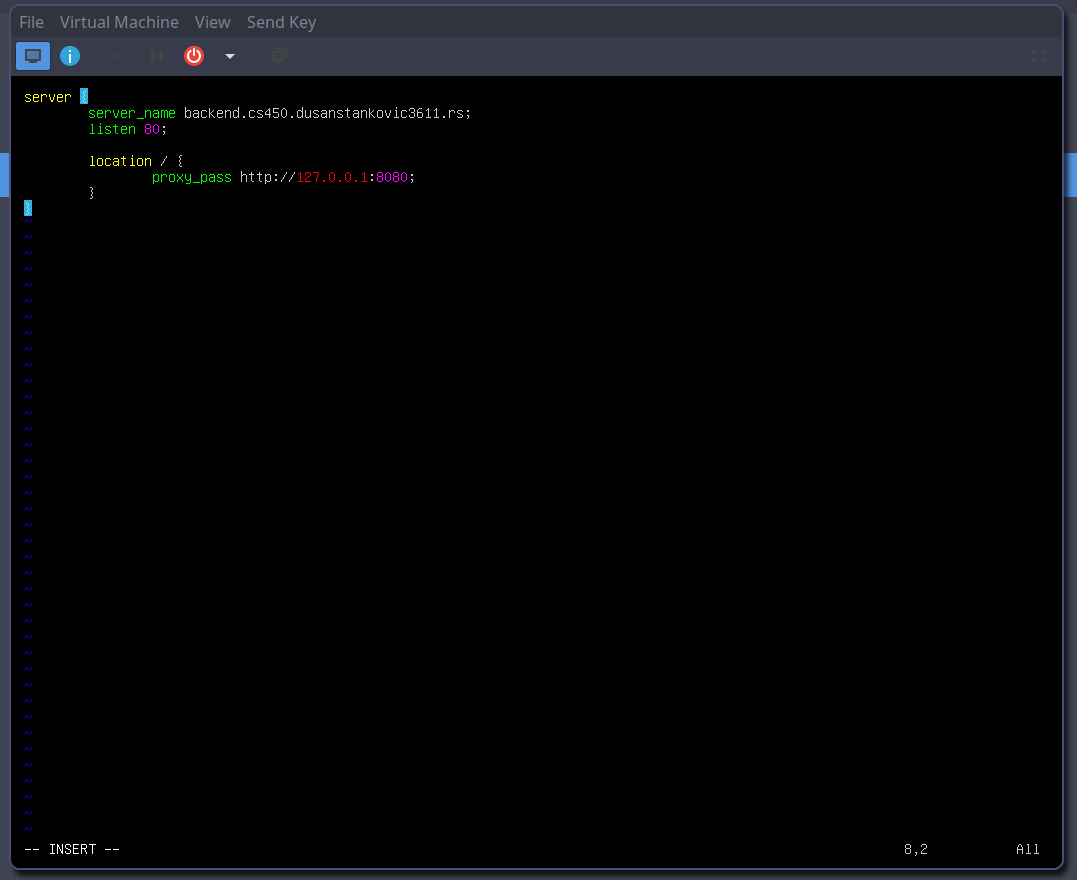
U prvoj liniji konfiguracije servera, definišemo naziv servera koji je "backend.cs450.dusanstankovic3611.rs". Ovo omogućava Nginx serveru da prepozna koji zahtevi treba da budu usmereni ka ovom serveru.

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.

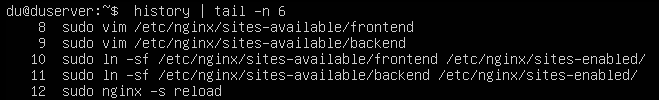
Unutar bloka "location /", koristimo "proxy\_pass http://127.0.0.1:8080;" kako bismo defisnali adresu backend servera na koji će Nginx usmeriti zahteve koji dolaze na ovu putanju. U ovom primeru, backend server je dostupan na adresi "http://127.0.0.1:8080".

Ove konfiguracijske 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".



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

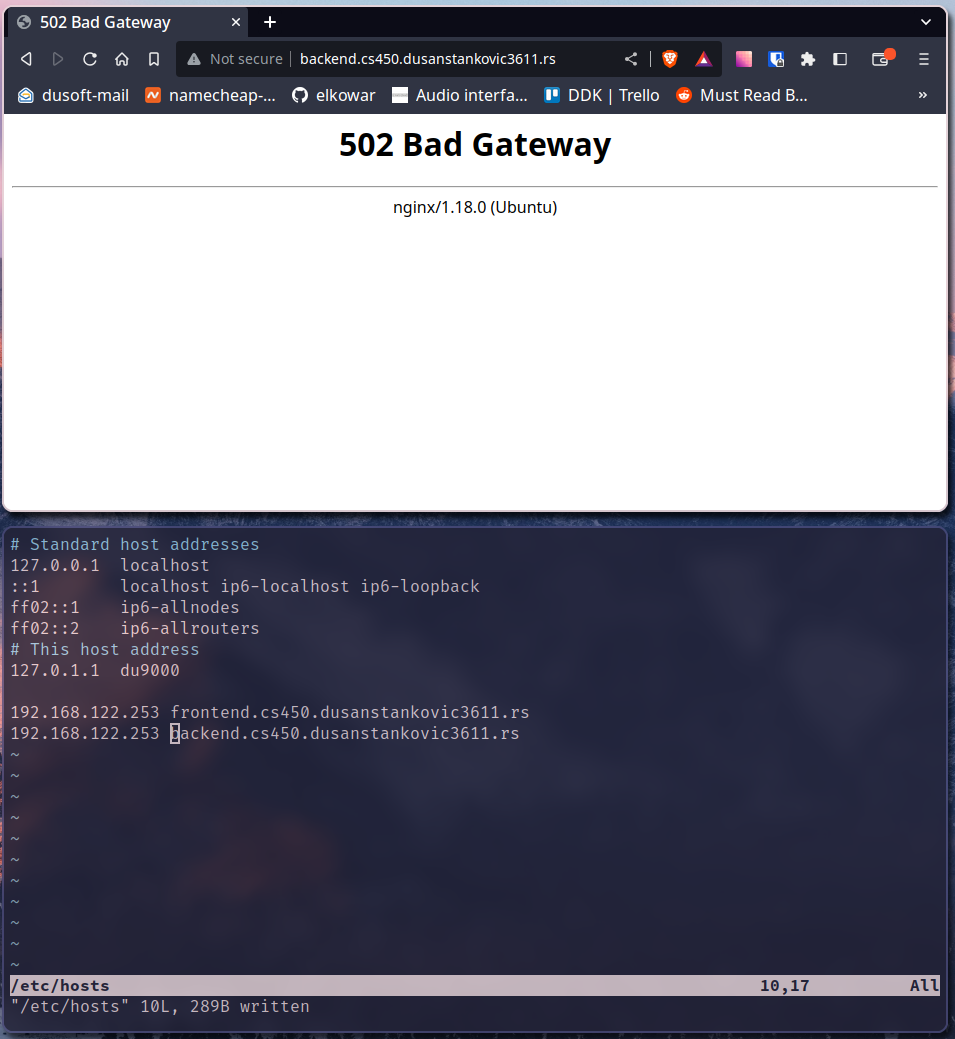
Pravimo prečice obe konfiguracije u sites-enabled folder nginx-a kako bi uključili podizanje front i back end projekata, prečice se korišćenjem ln komande, gde flag -sf predstavlja soft link. (slika 25)



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

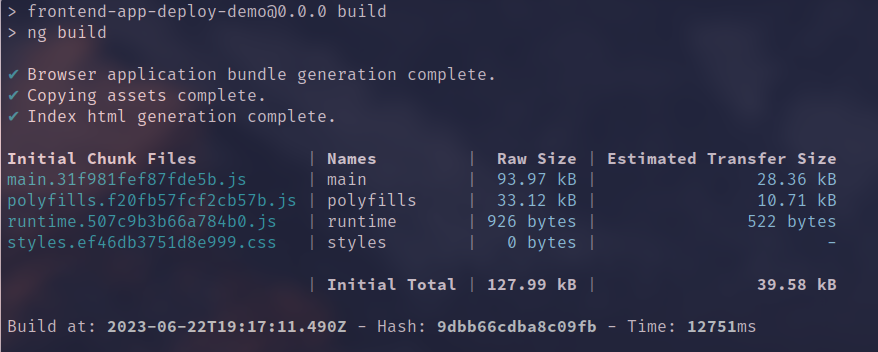
Ovde je prikazana mock konfiguracija koja nam omogućava da vežemo različite domene za IP adresu servera koji je podignut kao virtuelna mašina, bez potrebe da kupujemo domen i prolazimo kroz komplikovan proces njegove validacije.

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).



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.

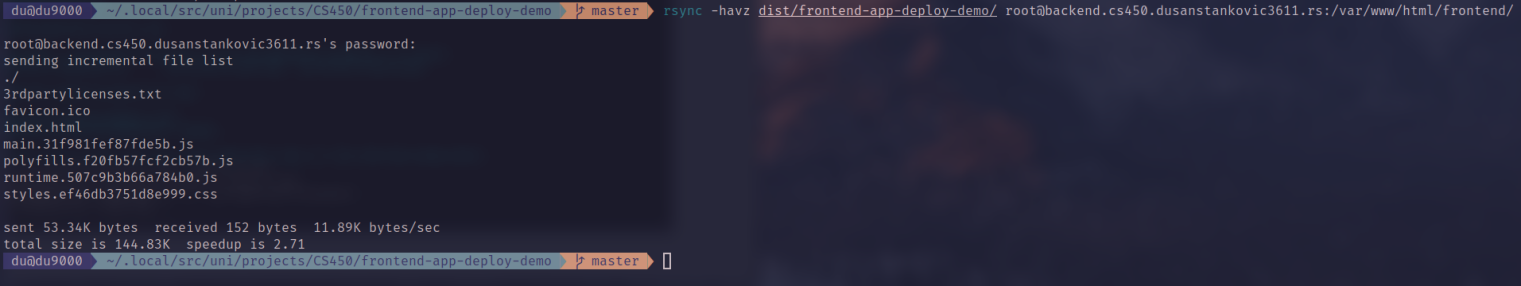
"-a" flag predstavlja "archive" (arhiva) mod i koristi se za očuvanje sveobuhvatnih atributa fajlova tokom sinhronizacije. To uključuje informacije o vlasniku, grupi, dozvolama, vremenskim oznakama i drugim atributima fajlova. Ovaj flag osigurava da se svi relevantni atributi fajlova očuvaju tokom kopiranja.

"-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.

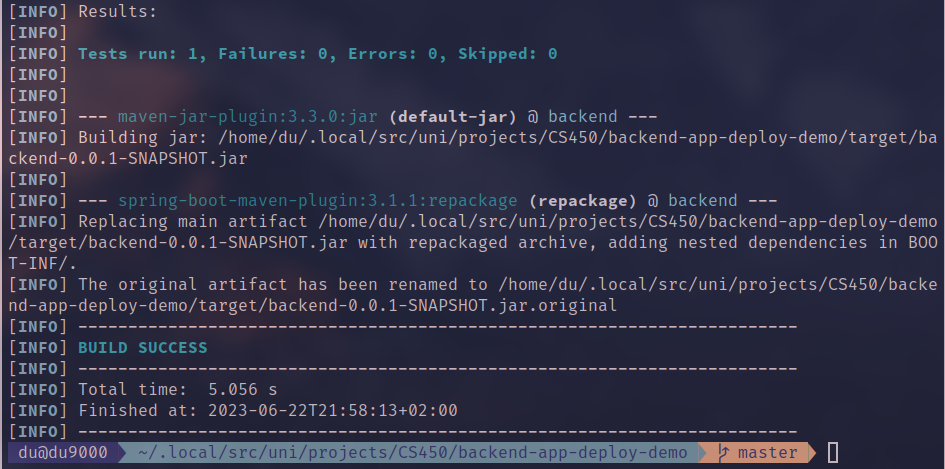
Kombinacija ovih flagova ("-havz") omogućava čitljiv ispis veličina fajlova, očuvanje atributa fajlova, detaljno praćenje napretka i kompresiju podataka tokom prenosa pomoću rsync-a.

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)



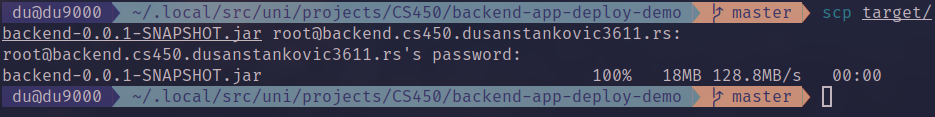
Slika 28. rsync komanda za prenos image-a na server

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



Slika 29. Build backend-a

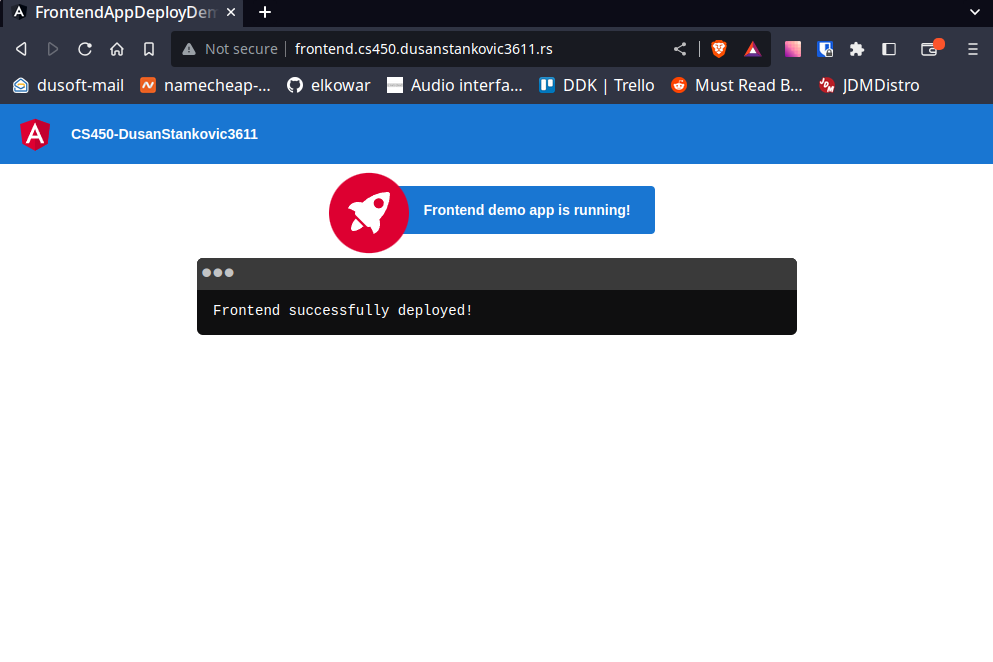
S obzirom na to da je artifakt buildovanja backend projekta samo jedan .jar fajl ne moramo da koristimo rsync (rsync znatno ubrzava proces kopiranja više fajlova), već možemo jednostavno koristiti secure copy (scp). (slika 30)



Slika 30. Kopiranje backend-a na server

# Analiza i prikaz postignutih rezultata

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



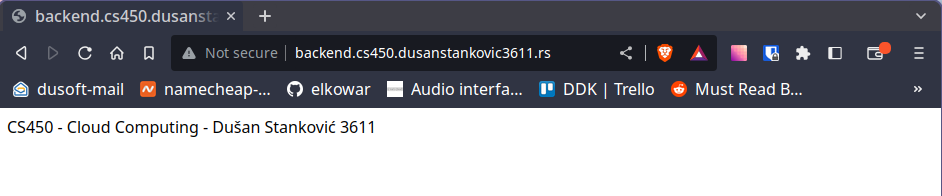
Slika 31. Rezultat pristupanja podignutom frontendu na serveru sa lokalne mašine

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).



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

# 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 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.

# 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>