Implementacija mrežnih servisa u lokalnim računalnim mrežama

Dujak, Ena

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:665317

Rights / Prava: Attribution-NonCommercial-NoDerivs 3.0 Unported/Imenovanje-Nekomercijalno-Bez prerada 3.0

Download date / Datum preuzimanja: **2024-03-29**

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository

zir.nsk.hr

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Ena Dujak

IMPLEMENTACIJA MREŽNIH SERVISA U LOKALNIM RAČUNALNIM MREŽAMA

ZAVRŠNI RAD

Varaždin, 2020.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Ena Dujak

Matični broj: 0016129773

Studij: Informacijski sustavi

IMPLEMENTACIJA MREŽNIH SERVISA U LOKALNIM RAČUNALNIM MREŽAMA

ZAVRŠNI RAD

Mentor:

Izv. prof. dr. sc. Ivan Magdalenić

Varaždin, kolovoz 2020.

Ena Dujak

Izjava o izvornosti

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristila drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autorica potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Tema ovog završnog rada jest implementacija često korištenih mrežnih servisa kao što su DNS, DHCP, Proxy server i dr. na Linux server poslužitelja i klijentska računala u mreži. Na poslužiteljskom računalu korišten je operacijski sustav Ubuntu server 18.04, a na dva od tri klijentska računala instaliran je Ubuntu 18.04. Bionic Beaver. Treće računalo u mreži pokreće Windows 10 kako bi se prikazale mogućnosti Ubuntu Servera i na komercijalnim rješenjima operacijskih sustava. Kroz rad je tako prikazana implementacija prethodno navedenih servisa na poslužiteljsko računalo te uz svakoga od njih kroz jedno poglavlje prikazana je njihova teorijska osnova i uloga u samoj mreži. Nakon toga opisana je instalacija LAMP servisa te komunikacija klijenata s poslužiteljem kroz ulogu web servera i *file sharing* servera čija je svrha učitavanje web stranica s klijentskog na poslužiteljsko računalo. Osim dijeljenja datoteka klijent-server, u radu je prikazano i dijeljenje klijent-klijent realizirano kroz Samba protokol koji uvelike olakšava razmjenu podataka među računalima u mreži. Na samome kraju i prije zaključka prikazano je nekoliko skripti za automatizaciju procesa za Ubuntu i Windows operacijske sustave.

Ključne riječi: Linux server; Ubuntu; DHCP; DNS; Proxy server; LAMP; Mrežni servisi;

Sadržaj

1. Uvod 1
2. Metode i tehnike rada 2
2.1. VirtualBox okruženje za virtualizaciju 3
2.2. Ubuntu Server operacijski sustav 3
2.2.1. Prikaz radnog sučelja 4
3. Virtualno okruženje i virtualna računala 6
3.1. Mreža virtualnih računala 7
3.2. Statička IP adresa na serveru 9
4. Pristup internetu 10
4.1. DHCP 12
4.1.1. Implementacija DHCP servera 12
4.1.2. Dodjeljivanje statičke IP adrese putem DHCP servera i testiranje 14
4.2. DNS i port forwarding 15
4.2.1. Implementacija DNS servera 16
4.2.2. Port forwarding 21
4.3. Proxy server i web cache 23
4.3.1. Implementacija Proxy servera 23
4.4. Skriptiranje – automatizacija postavki 27
4.4.1. Novo računalo u mreži 28
5. Web server i File sharing 30
5.1. LAMP stack 31
5.1.1. Apache server 31
5.1.2. MySQL server i phpMyAdmin 32
5.1.3. PHP 32
5.1.4. FTP server i pristup putem Filezille 33
5.2. File sharing među klijentskim računalima – Samba 35
6. Zaključak 37
Popis literature 38
Popis slika 40

1. Uvod

Današnji svijet počiva na internetu – dio je naše svakodnevice i utkan je u gotovo sve sfere našega života. Iz tog razloga teško je zamisliti budućnost bez umreženosti koju nam internet pruža. Osim privatnih korisnika koji su umreženiji nego ikada, poduzeća i poslovni svijet općenito sve se više oslanjaju na internet i sve njegove mogućnosti kako bi obavljali dio svojih poslovnih procesa. Ipak, kako bi osigurali sigurnost, privatnost i neovisnu internu komunikaciju unutar poduzeća potrebno je kreirati internu mrežu računala sa zasebnim izlazom na internet. Na ovaj način moguće je kontrolirati ulazni i izlazni promet računala u mreži i ujedno omogućiti neometanu internu komunikaciju koja ne ovisi o internetu i vanjskim poslužiteljima.

Ideja interne mreže u poduzeću, školi ili drugdje premisa je ovoga rada stoga će na sljedećim stranicama biti prikazana konfiguracija interne mreže koristeći virtualna računala, implementacija osnovnih mrežnih servisa kao što su DHCP i DNS na interni poslužitelj kako bi računala u mreži mogla komunicirati s vanjskim poslužiteljima te prikaz komunikacije računala u mreži na razini klijent-server i klijent-klijent.

Kako bi se što bolje simulirali uvjeti iz stvarnoga života kao poslužitelj je korišteno virtualno računalo s instaliranim operacijskim sustavom Ubuntu Server 18.04. budući da se radi o *open source* rješenju koje koristi većina današnjih web servera[1]. Dva od tri klijentska računala također pokreće Ubuntu, ali njegova *desktop* verzija, zbog jednostavnosti korištenja i *open source* prirode. Na treće klijentsko računalo instaliran je Windows 10 operacijski sustav kako bi se testirale funkcionalnosti i na komercijalnim rješenjima, ali i zato što veliki broj poduzeća koristi oba operacijska sustava u svojim internim mrežama.

Nakon konfiguracije interne mreže u aplikaciji za virtualizaciju, potrebno je instalirati sve potrebne servise na poslužitelja kako bi klijentska računala imala izlaz na internet. Osim mrežnih servisa u radu će biti prikazana i instalacija web servisa koji se mogu koristiti za posluživanje web stranice poduzeća i slično. Poglavlje automatizacije procesa nakon toga bit će posvećeno skriptiranju i dvjema skriptama koje uvelike mogu olakšati konfiguraciju budućih računala u mreži, nakon čega slijedi zaključak i sinteza mojega rada.

Tvrdnja navedena na početku uvoda o neizostavnosti interneta u današnjem svijetu potkrijepljena je i ovim radom kojemu je glavni i jedini izvor literature upravo internet te brojni članci, forumi i e-knjige na istome.

2. Metode i tehnike rada

U tijeku pisanja završnoga rada korištene su metode i tehnike isključivo tehničke prirode. Za pokretanje svih virtualnih mašina – i klijentskih i poslužiteljskih – korištena je aplikacija *VirtualBox* unutar koje se konfigurirala interna mreža računala. Poslužiteljskom računalu dodijeljeno je 4 GB, a svakom od klijentskih računala po 2 GB radne memorije kako bi se osiguralo nesmetano pokretanje svih operacijskih sustava. Sve virtualne mašine instalirane su na *host* računalo sa 16 GB radne memorije stoga je sva računala bilo moguće pokrenuti i koristiti u isto vrijeme.

Kao što je već navedeno u uvodnome poglavlju, na poslužiteljskom računalu instaliran je operacijski sustav Ubuntu server 18.04. što će biti baza na kojemu će se temeljiti cijeli završni rad. Ovaj OS odabran je zbog svoje jednostavnosti i *open source* prirode koja se koristi i u komercijalne svrhe te ga je moguće vidjeti na velikom broju poslužitelja koji danas pokreću internet. Dva klijentska računala također pokreće Ubuntu, a na trećem je instaliran OS Windows 10. Kako bi se ostvarila željena mrežna konfiguracija na poslužiteljsko računalo instalirane su sljedeće aplikacije i servisi:

- DHCP server
- Bind9 DNS server
- Squid proxy server
- Apache server
- MySQL server
- PHP
- phpMyAdmin
- Vsftpd FTP server

Na klijentska računala instalirane su sljedeće aplikacije i servisi:

- Samba
- FileZilla

U sljedećem poglavlju bit će detaljnije opisana konfiguracija svih računala u mreži i operacijskih sustava koji ih pokreću.

2.1. VirtualBox okruženje za virtualizaciju

Virtualizacija je pojam koji se odnosi na instaliranje i pokretanje operacijskih sustava unutar virtualnog okoliša, u obliku aplikacije na već instaliranom operacijskom sustavu. Lako se stoga može zaključiti kako je virtualizacija savršena za proučavanje, testiranje i korištenje više virtualnih računala istovremeno. Za svrhe pisanja ovog završnog rada virtualizacija je korištena kao simlulator, odnosno kako bi se istovremeno moglo koristiti tri klijentska i jedno poslužiteljsko računalo, bez ikakvih fizičkih komponenti. U slučaju kreiranja stvarne mreže na temelju ovdje prikazane testne mreže, svi postupci mogu se replicirati koristeći fizička računala uz jednake rezultate. U sljedećim poglavljima bit će također opisano postavljanje simulirane lokalne mreže unutar postavki aplikacije korištene za virtualizaciju – VirtualBoxa.

VirtualBox se na operacijski sustav korisnika instalira kao aplikacija, a prije pokretanja prve virtualne mašine u BIOS postavkama računala potrebno je omogućiti virtualizaciju. Nakon toga kao jedine predispozicije za virtualizaciju, moguće je instalirati više operacijskih sustava na proizvoljan broj računala želji. Pri instalaciji je potrebno paziti na dodjeljenje radne memorije kako zajedno pokrenuta računala ne bi prešla granicu radne memorije na *host* računalu. Ipak, o tome i svim drugim postavkama bit će više riječi kasnije.

Novije verzije VirtualBoxa mogu pokretati veliki broj virtualnih mašina s gotovo svim poznatim operacijskim sustavima koji uključuju Windows XP, Vista, 7, 8, 10, razne Linux distribucije, Solaris i dr. [3]. Osim operacijskih sustava koje može pokrenuti, važno je napomenuti i kako VirtualBox podržavaju Windows, Mac i Linux operacijski sustavi.

Glavna premisa virtualizacije je korištenje *host* računala i memorije na istome kao fizičku memoriju na kojoj su spremljene sve datoteke i postavke instaliranog operacijskog sustava zbog čega je moguće kreirati i svojevrsne snimke ili zapise trenutačnog stanja operacijskog sustava na koje se računalo uvijek može vratiti, što se u obliku datoteke sprema na *host* računalo [3]. Iz tog je razloga virtualizacija savršena za testiranje i slične svrhe.

2.2. Ubuntu Server operacijski sustav

Ubuntu Server operacijski sustav jedan je od najkorištenijih operacijskih sustava za poslužiteljska računala na svijetu. Jedan od razloga za to jest jednostavnost operacijskog sustava kao takvog, mala iskorištenost radne memorije i minimalistička priroda grafičkoga sučelja odnosno nedostatak istog. Osim prethodno navedenih prednosti, Ubuntu Server je *open source* rješenje što znači da je besplatan za korištenje u komercijalne i nekomercijalne svrhe, za razliku od popularnog komercijalnog rješenja Windows Servera čija se cijena za mnoga manja poduzeća predstavlja kao nenadmašiva prepreka. Windows Server također ne stavlja naglasak na jednostavnost i potpuno korištenje svih kapaciteta računala na kojemu je

instalirano, budući da dolazi uz grafičko sučelje koje podsjeća na ono koje koriste milijuni privatnih korisnika na klijentskoj verziji Microsoftova operacijskog sustava.

Kao što je već navedeno, Ubuntu Server minimalistički je operacijski sustav te se kao takav može koristiti u najjednostavnije svrhe što dokazuje i svojim izrazito niskim tehničkim specifikacijama, a koje su kako slijedi [2]:

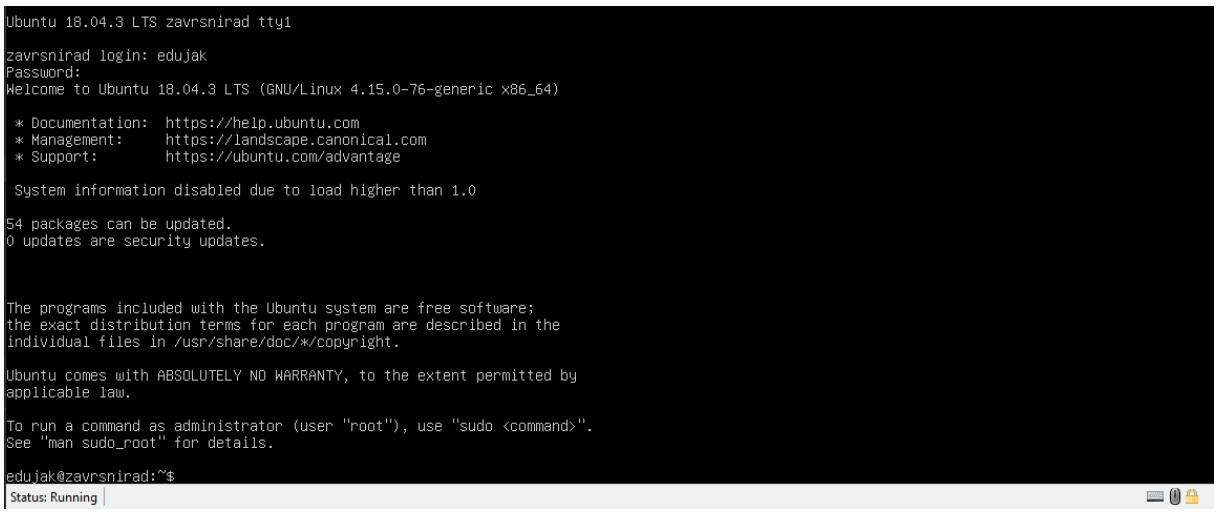
- Jedna CPU jezgra
- 512 MB radne memorije
- 10 GB fizičke memorije

Iz ovoga se može zaključiti kako i običan korisnik ovaj operacijski sustav može koristiti za svrhe kućne multimedije i slično. Windows Server s druge strane namijenjen je komercijalnim rješenjima stoga se podrazumijeva da će računalo koje ga pokreće imati nešto više tehničke mogućnosti i vrlo često nije odabir običnih korisnika ili za jednostavnije operacije.

Kopiji operacijskog sustava na *host* računalu korištenog za potrebe pisanja završnoga rada dodijeljeno je 4 GB radne memorije i 50 GB fizičke memorije, što se pokazalo kao više nego dovoljno za potrebe pokretanja svih instaliranih servisa i usluživanja svih klijentskih računala.

2.2.1. Prikaz radnog sučelja

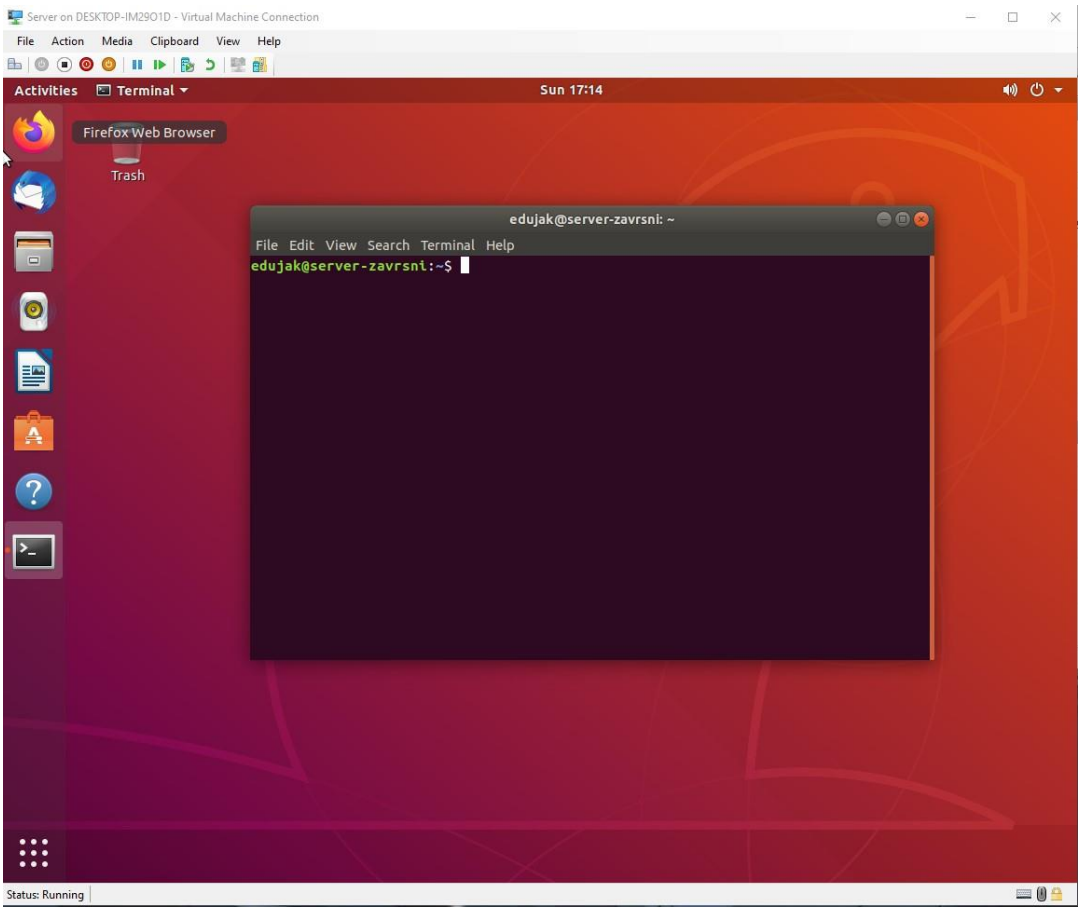
Na slici 1 je prikazano jednostavno sučelje s kojim dolazi Ubuntu Server koje izgledom podsjeća na računala iz prošlosti. Izgled je jedina poveznica koju Ubuntu dijeli s prošlosti, jer u pozadini se krije moćan operacijski sustav sposoban za stotine i tisuće zahtjeva u ulozi web servera ili pružanje mrežnih servisa velikom broju računala u ulozi DHCP, DNS i Proxy servera.



Slika 1: Prikaz sučelja Ubuntu Servera (slika autora)

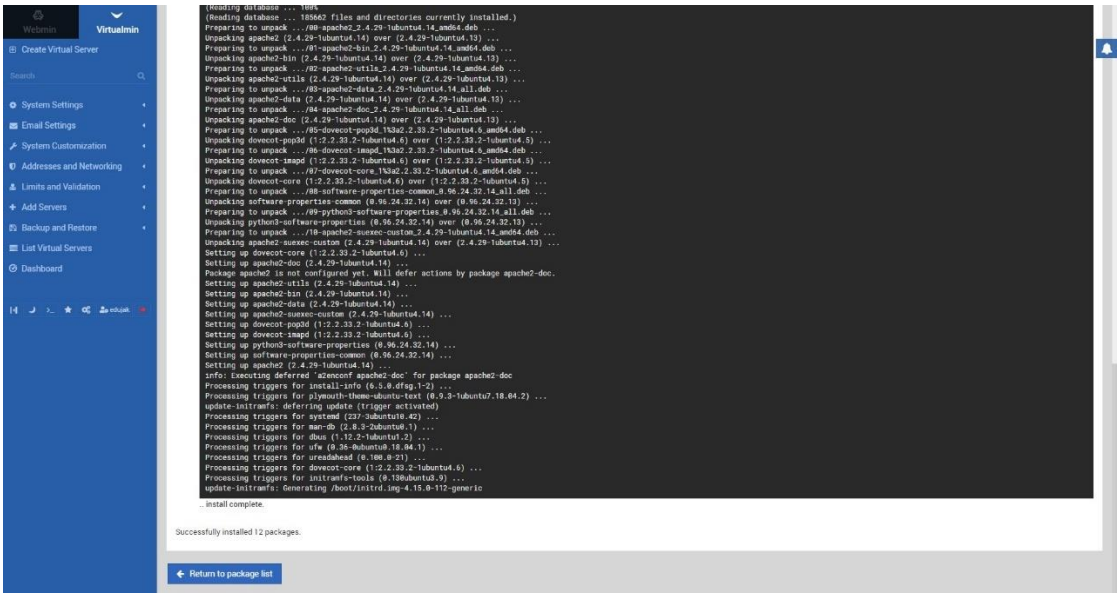
Još jedna od prednosti ovog operacijskog sustava jest komandna linija čiji je jezik identičan onome na klijentskoj verziji Ubuntua. Iz tog razloga korisniku, pogotovo ako je već

otprije upoznat s bilo kojom od Ubuntu-baziranih distribucija, nije teško naviknuti se na korištenje terminala. Važno je za napomenuti također da korisnik nije obvezan koristiti osnovnu verziju operacijskog sustava bez grafičkoga sučelja, jer se ono u nekoliko koraka može instalirati kako bi se dobilo vizualno ljepše rješenje od osnovnoga.



Slika 2: Prikaz grafičkoga sučelja na Ubuntu serveru (slika autora)

Na instanci Ubuntu servera na slici iznad instalirano je Gnome grafičko sučelje, gdje se svim funkcionalnostima servera pristupa putem terminala koji se kao i sve drugo može personalizirati. Iz slike se također može uočiti kako je Ubuntu server sada, osim komandnom linijom, izgledom identičan *desktop* verziji ovog operacijskog sustava. Zbog toga je važno napomenuti kako se svi servisi korišteni na Ubuntu Serveru mogu instalirati i na njegov *desktop* pandan, no time se eliminira osnovna ideja jednostavnosti, minimalizma i optimizacije korištenja resursa za pozadinske servise, a ne za grafičko sučelje. Pogotovo zbog toga što poslužitelji po svojoj prirodi za svakodnevnu upotrebu niti ne trebaju monitore, već je moguće koristiti razna rješenja poput Webmina koji funkcionira kao servis kojemu se može pristupiti s računala na istoj mreži, a kojim se dobiva potpuni pristup funkcionalnostima servera uz jednostavno grafičko sučelje i pristup terminalu. Na sljedećoj slici tako se može vidjeti funkcionalnost Webmina i korištenje komandne linije na poslužitelju, s klijentskog računala na istoj mreži.

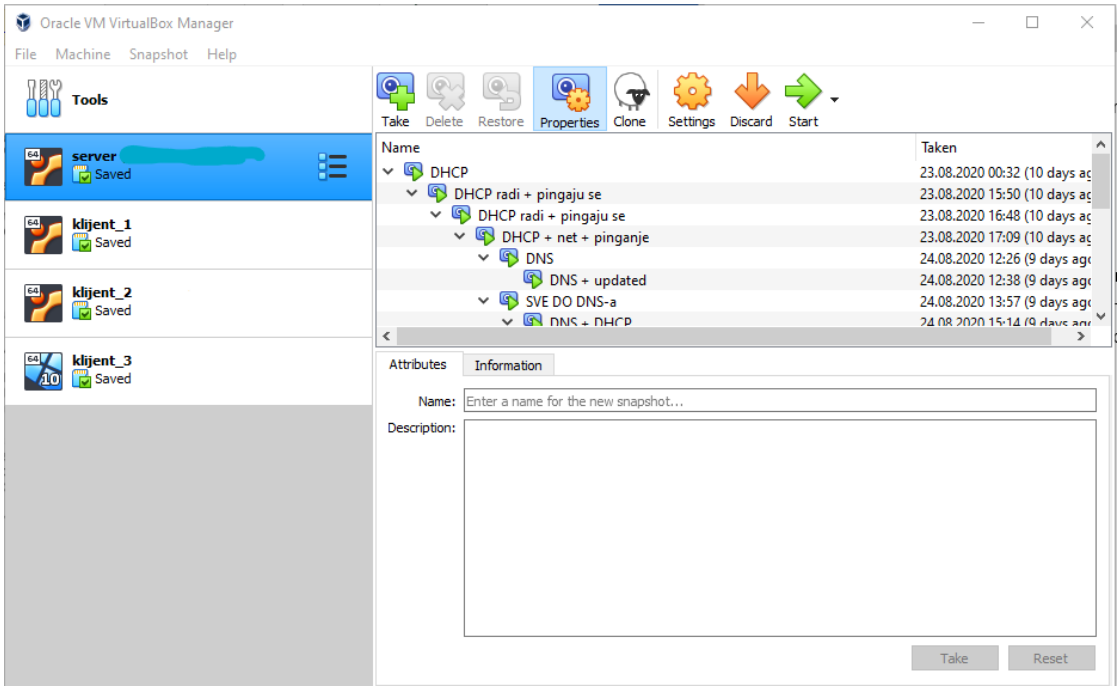


Slika 3: Webmin (slika autora)

Ipak, budući da je u ovome slučaju poslužitelj bio lokalno računalo s pristupom monitoru, nije bilo potrebno koristiti ovakav servis.

3. Virtualno okruženje i virtualna računala

Nakon prolaska kroz teorijsku osnove, potrebno je kreirati internu mrežu unutar aplikacije VirtualBox, a prije toga instalirati sama računala i potrebne operacijske sustave. Na sljedećoj slici moguće je vidjeti uspostavljene virtualne mašine i njihove nazive.



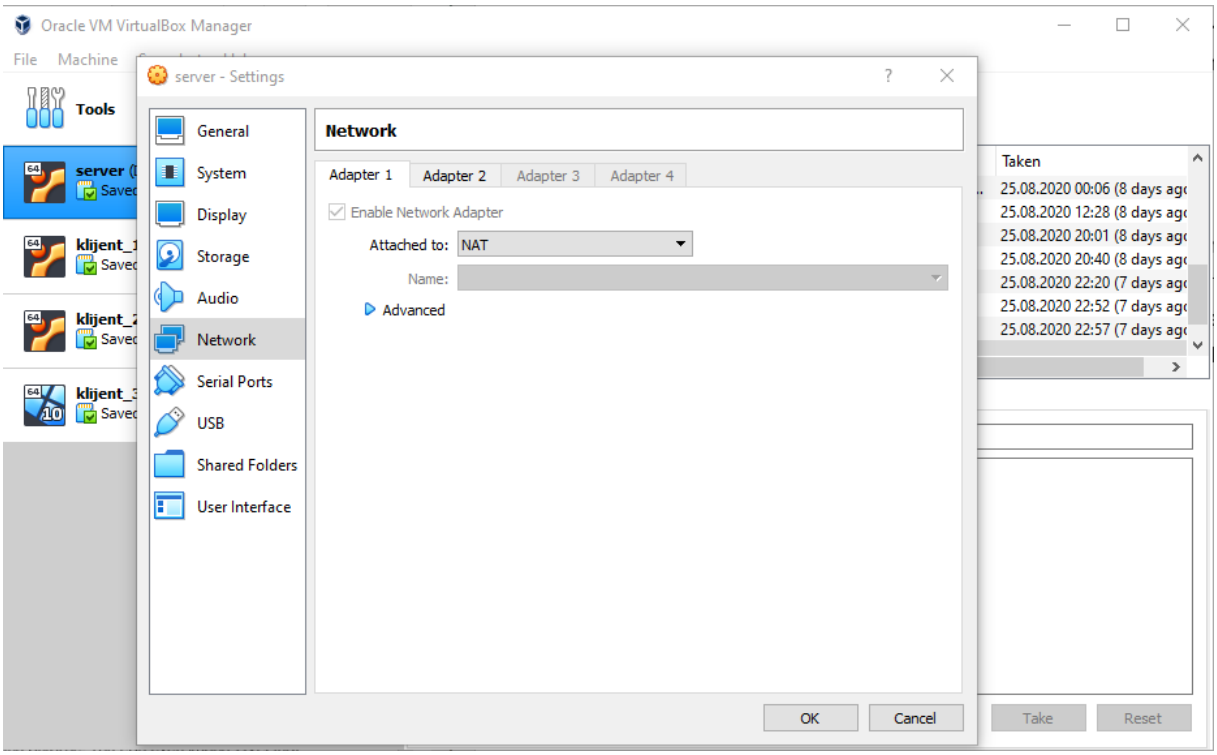
Slika 4: Prikaz instaliranih virtualnih mašina (slika autora)

Prvo virtualno računalo pod nazivom „server“ na jednostavnom korisničkom sučelju VirtualBox aplikacije poslužiteljsko je računalo s Ubuntu Server operacijskim sustavom, nakon čega slijede dva Ubuntu klijentska računala „klijent_1“ i „klijent_2“ te na koncu Windows 10 klijentsko računalo pod nazivom „klijent_3“. „Klijent_1“ i „klijent_3“ zamišljena su kao obična klijentska računala s dinamički dodijeljenim IP adresama, dok je „klijent_2“ zamišljen kao računalo za jednostavni *hosting* datoteka budući da je običnim korisnicima puno lakše koristiti računalo s jednostavnim grafičkim sučeljem i intuitivnim naredbama, ali i kako bi se demonstriralo klijent-klijent dijeljenje datoteka u mreži. Iz tog razloga ovome je računalu putem DHCP-a dodijeljena statička IP adresa kako bi druga računala u mreži mogla pronaći datoteke koje ovo računalo dijeli uz jednostavno mrežno dijeljenje.

Na desnoj strani sučelja aplikacije vidljive su sve snimke stanja operacijskoga sustava koji su rađeni tijekom instalacije svih potrebnih servisa. Na svaki od njih moguće se vratiti i nastaviti raditi od te točke u vremenu. Isto je i za klijentska računala, no njihov je broj snimki nešto manji nego za poslužiteljsko računalo.

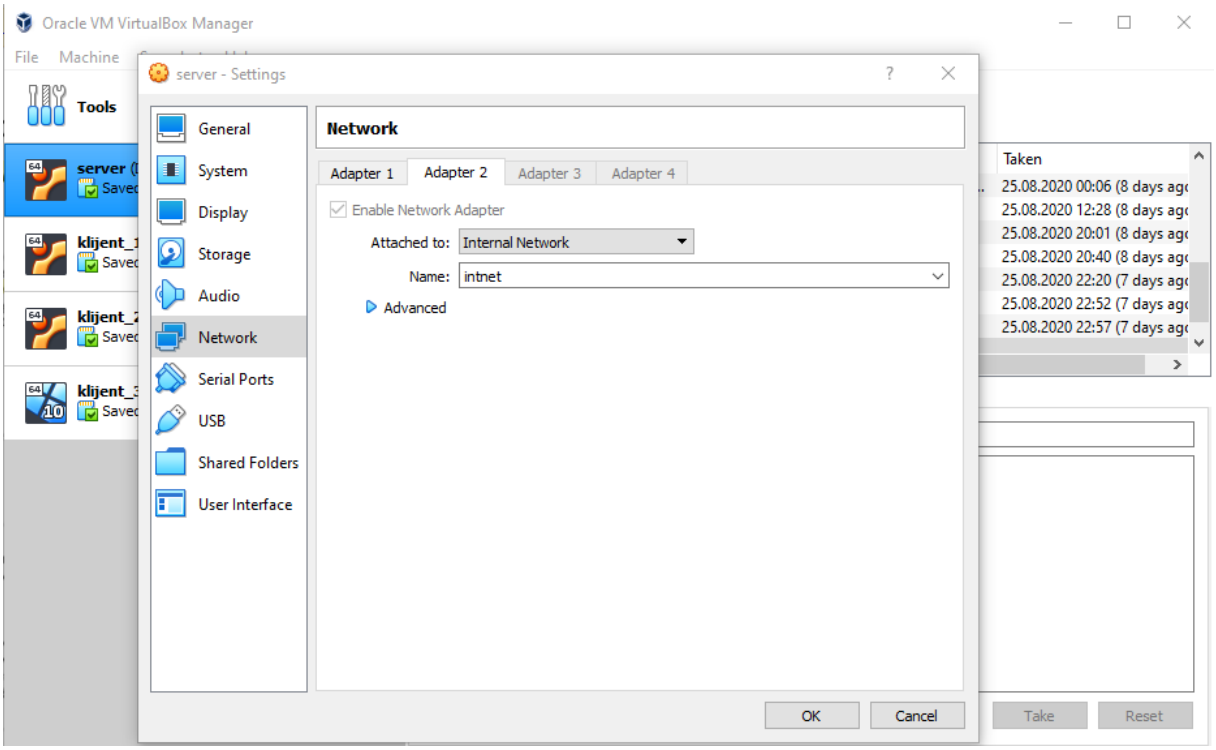
3.1. Mreža virtualnih računala

Nakon instalacije svih računala bilo je potrebno kreirati virtualnu mrežu kako bi se sva računala u mreži mogla prepoznati, bez utjecaja vanjskih čimbenika kao što su DHCP i DNS server. Ipak, računalima u mreži potreban je izlaz na vanjski internet, a kako je ideja da sav promet ide preko poslužiteljskog računala, na istome će mrežna konfiguracija biti kako slijedi:



Slika 5: Mrežna konfiguracija 1 – poslužitelj (slika autora)

Naime, na poslužiteljskom je računalu potrebno konfigurirati dva mrežna adaptera. *Default* konfiguracija na svakom novo-instaliranom računalu je postojanje jednog mrežnog adaptera, vidljiv na slici 5, koji virtualnom računalu pruža pristup lokalnom DHCP i DNS serveru odnosno pristup internetu. Budući da je ideja kreiranja interne mreže ta da sav promet prolazi isključivo kroz poslužitelja, na klijentskim računalima potrebno je promijeniti mrežni adapter, a na poslužiteljskom potrebno je jedan dodati. Ovaj adapter može se ostaviti budući da pruža izlaz na internet, a na sljedećoj slici prikazan je konfigurirani drugi mrežni adapter za poslužitelja koji služi internoj komunikaciji računala.



Slika 6: Mrežna konfiguracija 2 – poslužitelj (slika autora)

Preko ovog mrežnog adaptera poslužitelj se povezuje sa svim ostalim računalima u mreži stoga će mrežna konfiguracija na klijentskim računalima sadržavati samo ovaj mrežni adapter kao „Adapter 1“.

U slučaju fizičkih računala ove se postavke konfiguriraju na usmjerniku no budući da se radi o virtualnim računalima te je postavke potrebno konfigurirati unutar same aplikacije za virtualizaciju. Za potrebe testiranja iskušana je i konfiguracija sa samo jednim mrežnim adapterom na svim računalima, onim koji po *default* postavkama pruža izlaz na internet, a konsenzus je bio da je i dalje moguće klijentskim računalima promet usmjeriti na unutarnjeg poslužitelja. Ipak, pri svakom *rebootu* bilo je potrebno otpustiti dobivene IP adrese i klijente natjerati da ih ponovno pozovu od DHCP servera gdje su tek tada dobili adrese on unutarnjeg poslužitelja. Problemi su se ipak pokazali pri konfiguraciji drugih mrežnih servisa zbog čestih konflikata u mreži stoga je kreirana nova mreža s dva mrežna adaptera što se pokazalo kao logično i efikasno rješenje.

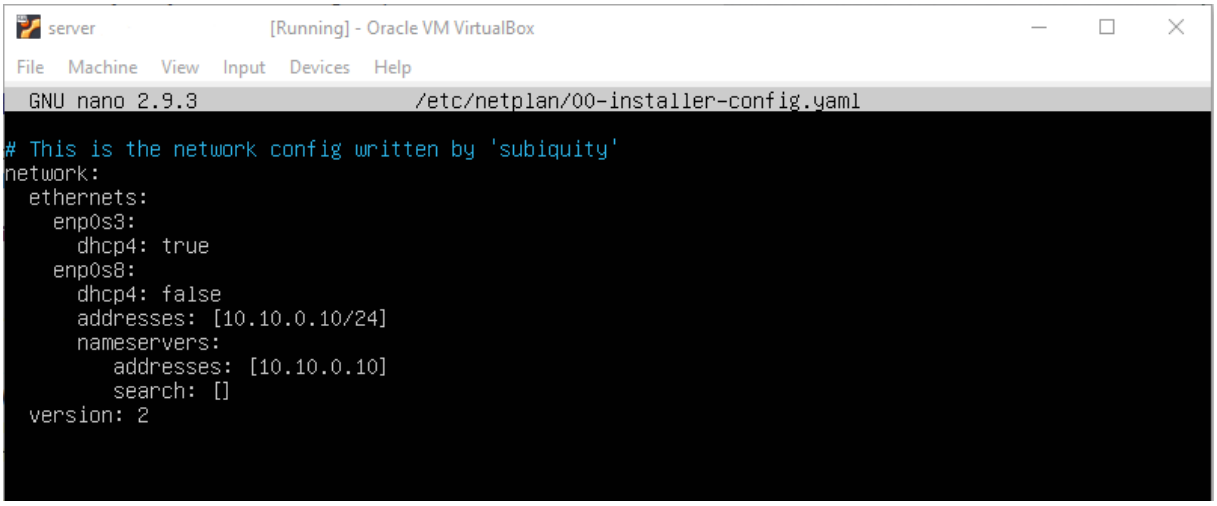
3.2. Statička IP adresa na serveru

Prvi korak u uspostavljanju poslužiteljskog računala jest postavljanje statičke IP adrese. Razlog za to je jednostavan – IP adresa jedinstveni je identifikator po kojemu će klijentska računala u mreži prepoznati poslužitelja kao pružatelja željenih servisa. U prethodnome koraku na poslužitelja su dodana dva mrežna adaptera, prvi ima izlaz na internet, a drugi je usmjeren ka internoj mreži. Iz toga se može zaključiti koji od adaptera mora imati statičku IP adresu, a koji može imati i od DHCP-a dinamički dodijeljenu adresu.

Adapter 1 ima izlaz na internet te je kao takav podložan DHCP serveru svojeg prespojnika; na njemu se iz toga razloga IP adresa ne mora dirati. Adapter 2 s druge strane služit će kao *gateway* za izlaz na internet klijentskim računalima stoga, kao što je prethodno navedeno, mora imati statičku IP adresu. Unutar sučelja upaljenog poslužitelja potrebno je upisati sljedeću komandu:

```
sudo nano /etc/netplan/00-installer-config.yaml
```

Odnosno, potrebno je otvoriti .yaml datoteku koja je dio novijeg Ubuntu rješenja za mrežnu konfiguraciju, Netplan. Ova datoteka sadrži informacije o mrežnim adapterima koji su prethodno postavljeni te se bilo kakva nepravilna promjena u ovoj datoteci negativno odražava na sve mrežne servise na poslužitelju. Budući da je potrebno na jedan od adaptera postaviti mrežnu adresu, nakon promjena nova .yaml datoteka izgledat će ovako:



Slika 7: Konfiguracija mrežnih adaptera (slika autora)

Prethodno spomenute nepravilne promjene u datotekama uvelike se odnose na uvlačenje odnosno *indentaciju* unutar datoteke. Potrebno je paziti da je svaki red jednake uvlake te da su sve ključne riječi točno napisane.

Adapter 1, u ovome slučaju automatski nazvan „enp0s3“ vanjski je mrežni adapter i kod njega je dovoljno ostaviti redak „dhcp4: true“ koji označava da se IP adresa na tom adapteru dobiva od DHCP servera zadanog rutera. Adapter 2 odnosno „enp0s8“ ima nešto drukčije postavke gdje je „dhcp4“ redak postavljen na false što označava da se IP adresa ne dobiva od DHCP servera već se ona postavlja u narednim retcima. Redak „addresses“ označava raspon adresa te zadani *mask* koji je u ovome slučaju postavljen na 255.255.255.0 odnosno /24. Polje „search“ ostavlja se prazan budući da nije potreban za ovu konfiguraciju, ali u njega se upisuje vrijednost naziva domena za pretragu. [4]

Ovim postavkama tako je postignuto da mrežni adapter 2 ima statičku IP adresu 10.10.0.10, sa *maskom* 255.255.255.0. Nakon postavljanja željenih postavki za dane mrežne adaptere potrebno je primijeniti postavke što će se postići sljedećim komandama:

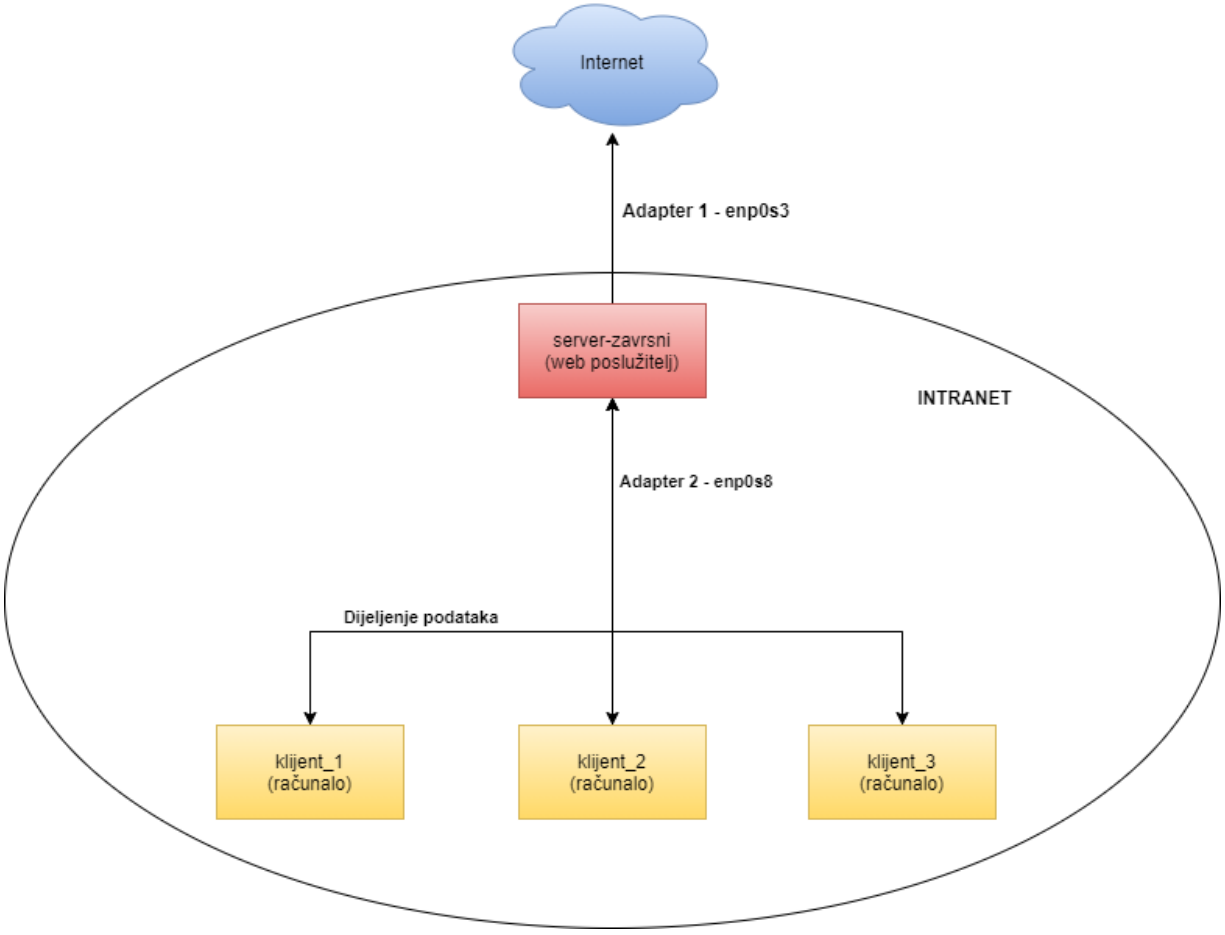
```
Sudo netplan generate | sudo netplan apply
```

čime se generiraju nove „Netplan“ postavke odmah primjenjuju. Moguće je i ponovno upaliti poslužiteljsko računalo no i ovo će biti dovoljno za primjenu novih postavki.

4. Pristup internetu

Nakon postavljanja statičke IP adrese na poslužitelja sljedeći korak je instalacija samih mrežnih servisa i pružanje izlaska na internet klijentskim računalima. Poslužitelj će zapravo klijentskim računalima služiti kao prespojnik stoga je potrebno definirati sve predispozicije kako bi usmjeravanje bilo uspješno. Potrebno je prvo uspostaviti DHCP server kako bi svako klijentsko računalo imalo identifikator na internetu – IP adresu, ali i identifikator prema kojemu će poslužitelj raspoznavati svako računalo. Kao što je navedeno u uvodnome poglavlju dva od tri klijentska računala imat će dinamički dodijeljene IP adrese, dok će za treće računalo biti definirana statička adresa unutar DHCP-a. Nakon dodijeljene IP adrese potrebno je definirati DNS kako bi računala unutar mreže mogla komunicirati međusobno putem dodijeljenih imena, a ne samo adresa. Prema tome može se zaključiti kako je DNS servis koji prevodi IP adresu u ime odnosno domenu danu od strane korisnika [5]. Bez DNS-a ne bi bilo moguće do stranice facebook.com doći preko njenog naziva, već samo putem IP adrese. Obrnuto, za svaki naziv potrebno je zadati IP adresu kako bi se i putem upisane adrese znao naziv domene stranice. U ovome slučaju domena interne mreže bit će „zavrsni.com“, a onda će naziv poslužitelja biti „server.zavrsni.com“ te klijenta za dijeljenje dokumenata sa statičkom IP „klijent.zavrsni.com“. Za upsanu IP adresu moguće je navesti nekoliko imena kako bi se do računala moglo doći na više načina, ali o tome će više riječi biti u poglavlju za DNS.

Uz DNS se veže i *forwarding*, budući da su na poslužitelju sada omogućene sve predispozicije za izlazak računala na internet. Kreirana je domena, računala imaju svoje identifikatore, a mogu se i međusobno „pingati“ koristeći nazive kreirane unutar DNS servisa. Jedino je sada potrebno kreirati put kojim će računala od internog mrežnog adaptera 2 doći do eksternog adaptera 1 koji ima izlaz na internet. Dakle, potrebno je kreirati svojevrsni most između ova dva adaptera kako bi se iz interne mreže moglo doći do eksterne, što pruža dodatan sloj sigurnosti internoj mreži koja u slučaju velikog broja poduzeća može sadržavati osjetljive informacije. Na sljedećoj slici moguće je vidjeti skicu topologije dosada kreirane mreže:



Slika 8: Topologija mreže (slika autora)

U trenutku kada je dobivena željena topologija mreže i sva računala međusobno mogu komunicirati, kao i s udaljenim računalima, može se reći da je uspješno kreirana privatna interna mreža s izlazom na internet. Iako je ovakva mreža relativno sigurna, implementacija proxy servera uvelike će povećati sigurnost i privatnost klijentskih računala na internetu.

Proxy server poslužit će kako bi sva računala u mreži imala prividno jednaku IP adresu te kako bi se sav izlazni i ulazni promet mogao kontrolirati. Moguće je blokirati određene stranice ili ključne riječi, a koristeći *web cache*, ubrzati učitavanje stranica kojima su računala u mreži već pristupila. Vrlo korisno u slučaju poduzeća s potrebama visokog stupnja sigurnosti.

Konfiguriranjem proxy servera dovršena je mrežna konfiguracija na ovome poslužitelju čiji klijenti sada nesmetano pristupaju internetu putem poslužitelja. U sljedećim poglavljima detaljno će biti opisani svi od implementiranih servisa, zajedno s koracima potrebnim za instalaciju.

4.1. DHCP

DHCP ili „Dynamic Host Configuration Protocol“ jedan je od najvažnijih protokola na internetu. Glavna zadaća DHCP-a odnosno DHCP servera je dinamičko dodjeljivanje IP adresa računalima u mreži ne temelju zadanih parametara ili automatski. Ipak, osim pukog dodjeljivanja IP adresa DHCP se bavi dodjeljivanjem svih drugih mrežnih parametara za dani uređaj kao što su *gateway*, DNS server i drugi [19]. Upravo iz tog razloga DHCP je prvi implementirani servis, jer na temelju njega počivaju svi ostali.

Prilikom dijeljenja IP adrese svakom računalu zasebno, DHCP se brine o vremenu posudbe adrese za svako računalo. Vrijeme posudbe postoji zbog prebrzog trošenja IPv4 adresa stoga je potrebno dinamički dodjeljivati IP adrese samo onim računalima koji su trenutačno na internetu. Osim dinamičkoga DHCP-a ovaj se problem rješava i postupnim uvođenjem IPv6 adresa kojima se DHCP također bavi.

Unutar konfiguracija za DHCP server moguće je promijeniti i po želji postaviti gotovo sve parametre, a osim dinamičkoga dijeljenja adresa također je moguće na temelju MAC adrese računalima u mreži svaki puta dodijeliti istu, statičku IP adresu što će i biti prikazano u narednim poglavljima.

4.1.1. Implementacija DHCP servera

Nakon teorijske osnove, nije teško shvatiti postupak konfiguracije DHCP servera. Potrebno ga je prvo instalirati koristeći sljedeću naredbu:

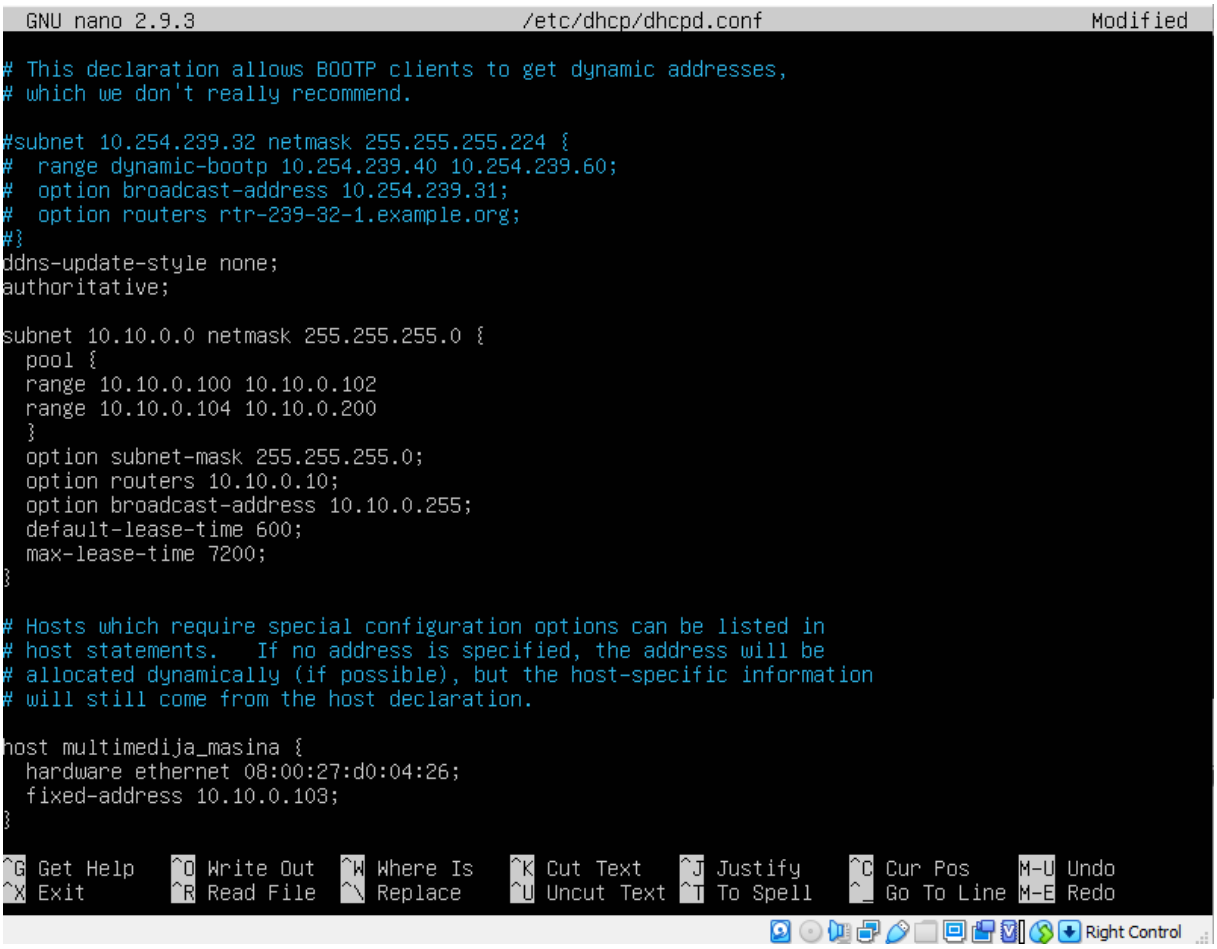
```
sudo apt-get install isc-dhcp-server
```

Sljedeći korak je konfiguracije mrežnog adaptera odnosno zadavanje onog koji će biti DHCP server. To se može napraviti izvođenjem sljedeće naredbe:

```
sudo nano /etc/default/isc-dhcp-server
```

gdje će se pod polje „INTERFACES“ staviti vrijednost koja označava naziv željenog mrežnog adaptera. U ovome slučaju to će biti enp0s8, interni mrežni adapter, budući da će se klijentska računala putem njega povezivati na DHCP server. Sada je potrebno konfigurirati sam DHCP

server biranjem imena domene, *subneta* i dr. Na sljedećoj slici prikazane su kreirane postavke koje se pristupaju putem putanje „/etc/dhcp/dhcpd.conf“.



Slika 9: Postavke za DHCP (slika autora)

Prvo polje u postavkama odnosi se na dinamičko ažuriranje DNS-a koje se može ostaviti na *default* vrijednosti none, dok se drugo polje „authoritative“ odnosi na postavljanje ovog DHCP servera kao autoritativnog servera za danu domenu [6]. Nakon toga za zadani *subnet* od 10.10.0.0 (ostavljen je ovakav *subnet* zbog interne statičke IP adrese poslužitelja koja je 10.10.0.10) dan je *netmask* od 255.255.255.0 odnosno /24. Ovaj će se *subnet* odnositi na sva računala u internoj mreži stoga je pod vrijednosti „pool“ dan izbor IP adresa koje poslužitelj može koristiti za dodjeljivanje klijentskim računalima. U ovome slučaju taj je odabir podijeljen na dva jer je trećem računalu sa statičkom IP adresom dana adresa 10.10.0.103 koja je dio tog subneta, ali onda ne smije biti dio dinamičkoga „poola“ kako ne bi dolazilo do konflikata.

Sljedeće polje „Option subnet-mask“ IP adresi klijentskoga računala dodjeljuje već prethodno definiranu *subnet masku*, a polje nakon toga „option routers“ računalima govori adresu poslužitelja kojim će se koristiti za izlaz na internet, ovdje se naravno radi o samom poslužitelju [6]. Nakon toga definirana je *broadcast* adresa, te redom vrijeme posuđivanja adresa i maksimalno vrijeme posuđivanja adresa od 7200 sekundi.

4.1.2. Dodjeljivanje statičke IP adrese putem DHCP servera i testiranje

Nakon što su definirane sve adrese koje se mogu dinamički dodjeljivati klijentskim računalima potrebno je za treće računalo rezervirati, također putem DHCP-a, statičku IP adresu. Na slici 9 funkcija „host multimedija_masina“ trećem računalu koje se definira putem jedinstvene MAC adrese dodjeljuje statičku IP adresu 10.10.0.103 te je iz tog razloga, kao što je prethodno navedeno, ova adresa uklonjena iz bazena mogućih IP adresa.

Prije nego li se konfiguracija testira potrebno je DHCP server ponovno pokrenuti, omogućiti automatsko paljenje pri uključivanju poslužitelja te provjeriti njegov status [7]. Naredbe za to upisuju se kako slijedi:

```
sudo systemctl restart isc-dhcp-server
sudo systemctl enable isc-dhcp-server
sudo systemctl status isc-dhcp-server
```

Ako je DHCP server dobro konfiguriran status će biti postavljen na „Active“ te je sada spreman za dijeljenje adresa računalima u mreži. Ovime se zaključuje konfiguracija IP adresa, a testiranjem klijenata koristeći sljedeću komandu:

```
hostname -I
```

može se provjeriti je li IP adresa odgovarajuća, odnosno pripada li zadanom bazenu adresa i jesu li sve druge postavke u redu.

```
edujak@klijent1-VirtualBox:~$ hostname -I
10.10.0.101
klijent2@klijent2-VirtualBox:~$ hostname -I
10.10.0.103
```

Slika 10: Prikaz IP adresa – Ubuntu (slika autora)

Na Windows operacijskom sustavu komanda je nešto drukčija („ipconfig“), ali rezultati isti:

```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::4839:9ff0:745a:2099%10
   IPv4 Address. . . . . . . . . . . : 10.10.0.100
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.0.10
```

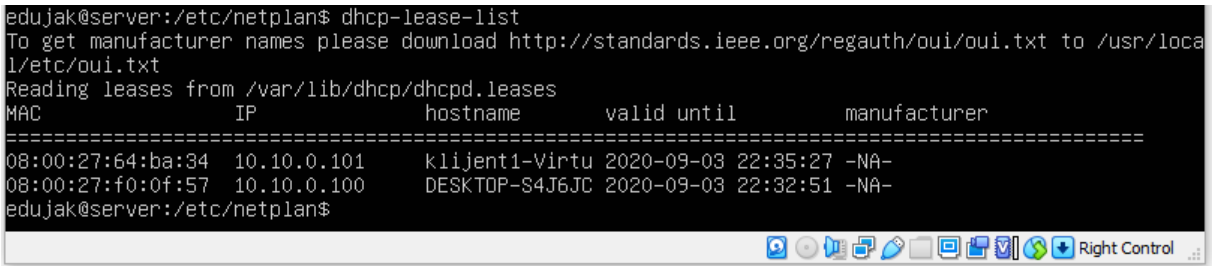
Slika 11: Prikaz IP adresa – Windows (slika autora)

Ovdje se također može vidjeti i kako je *Default Gateway* pravilno postavljen i pokazuje na poslužitelja. Čak i nakon otpuštanja i ponovnog dohvaćanja adresa koristeći „dhclient -r “ na Ubuntu računalima i „ipconfig /release“ na Windows računalu, rezultati su isti i IP adrese ostaju jednake budući da nije istekao rok za dane adrese.

Osim na samim klijentskim računalima posuđene IP adrese mogu se provjeriti na poslužiteljskome računalu pokretanjem komande

```
dhcp-lease-list
```

čiji je rezultat kako slijedi:



Slika 12: Lease-list naredba (slika autora)

Na slici se tako mogu vidjeti sva klijentska računala zajedno sa svojim MAC i IP adresama te vrijeme do kada njihove IP adrese vrijede. Na popisu se ne nalazi treće računalo sa statičkom adresom budući da njemu adresa nije posuđena već dana i neće se promijeniti.

Nakon uspješnog dodjeljivanja IP adresa, konfiguriranja *gateway* računala i svih ostalih DHCP postavki, potrebno je konfigurirati još nekoliko servisa kako bi mreža bila potpuno funkcionalna.

4.2. DNS i port forwarding

„Domain Name System“ odnosno DNS servis je koji je dozvolio komercijalno širenje interneta i svih njegovih blagodati. Prije DNS-a za svaku se stranicu morala znati njena točna adresa bez koje joj se nije moglo pristupiti. DNS ovdje dolazi kako bi korisnicima uvelike olakšao pa služi kao svojevrstan imenik unutar svakog prespojnika jer povezuje IP adrese s njezinim nazivom. U prijevodu, umjesto adrese 8.8.8.8 dovoljno je samo upisati google.com kako bi se došlo do iste stranice, ali bez znanja točne IP adrese.

DNS je protokol u kojemu je vrlo važna hijerarhija pa je tako u ovom slučaju „zavrsni.com“ *root* domena, dok je „server.zavrsni.com“ po hijerarhiji stepenicu iznad budući da se radi o poslužiteljskom računalu unutar domene „zavrsni.com“ [19]. Zona je pojam koji se bavi prethodno opisanom hijerarhijom pa će u poglavlju ispod biti prikazana konfiguracija

forward i *reverse* zona kako bi rezolucija bila uspješna u oba smjera – bez obzira je li se upisao naziv pa je potrebno dobiti adresu, ili se upisala adresa pa je potrebno dobiti naziv.

4.2.1. Implementacija DNS servera

Sljedeći korak u implementaciji mrežnih servisa konfiguracija je DNS servera koji se ujedno pokazao kao najizazovniji dio projekta. DNS će unutar lokalne mreže poslužiti za više namjena, najvažnija od njih bit će web hosting i pristupanje drugim sličnim servisima unutar mreže i izvan nje. Web stranici koja će biti spremljena na poslužitelja moći će se pristupati putem njenog imena, a ne IP adrese. Nadalje, ovime se definira svojevrsna domena unutar koje su sva računala u mreži i putem koje se može definirati servis za slanje mailova što nije opisano u ovome radu.

Prvo što je potrebno napraviti je preuzimanje i instalacija DNS servisa pod nazivom „bind9“ i njemu potrebnih alata za konfiguraciju „bind9utils“ putem naredbe:

```
sudo apt-get install bind9 bind9utils
```

Nakon instalacije DNS-a na poslužitelja sljedeći korak je kreiranje zona čiji će naziv biti „zavrsni.com“. Odnosno, kreira se zona za prevođenje naziva u IP adresu – *forward* zona i zona za prevođenje IP adrese u naziv – *reverse* zona. Kreiranje zona provodi se unutar mape „bind“ pod putanjom „/etc/bind/“, a nakon pozicioniranja u mapu otvara se datoteka „named.conf.local“ koja će nakon konfiguracije izgledati ovako:

```
  GNU nano 2.9.3                        /etc/bind/named.conf.local

//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "zavrsni.com" IN {
type master;
file "/etc/bind/forward.zavrsni.com";
};

zone "0.10.10.in-addr.arpa" IN {
type master;
file "/etc/bind/reverse.zavrsni.com";
};
```

Slika 13: Kreiranje zona (slika autora)

Zona „zavrsni.com“ odnosi se dakle na identificiranje IP adrese na temelju naziva računala, a vrijednost polja „type“ postavljena je na „master“ budući da je ovaj DNS server glavni za dano računalo poslužitelja. Putanja pod poljem „file“ pokazuje na datoteku koja još nije kreirana, ali će biti u sljedećem koraku. Slično je i za zonu „0.10.10.in-addr.arpa“ čiji naziv sadrži IP adresu

poslužiteljskog računala bez posljednjeg znaka i obrnutim redoslijedom. Razlog ovakvome zapisu je čitanje domene od najniže do najviše hijerarhije kako bi se lakše odgonetnuo naziv kojemu adresa pripada. [8]

Prije nego što se pozovu datoteke referencirane u konfiguraciji sa slike 13 potrebno ih je kreirati koristeći naredbe:

```
sudo nano /etc/bind/forward.zavrsni.com
sudo nano /etc/bind/reverse.zavrsni.com
```

U prvu od navedenih datoteka zapisuje se konfiguracija u obliku naziva domene i računala koji joj pripadaju. Za jedno računalo odnosno jednu IP adresu moguće je zadati više imena. U slučaju ovoga projekta u DNS je uključeno poslužiteljsko računalo i klijentsko računalo sa statičkom IP adresom budući da su to jedina dva računala u mreži kojima će pristupati druga računala. Dovršena konfiguracija za *forward* DNS vidljiva je na sljedećoj slici:

```
 GNU nano 2.9.3                    /etc/bind/forward.zavrsni.com

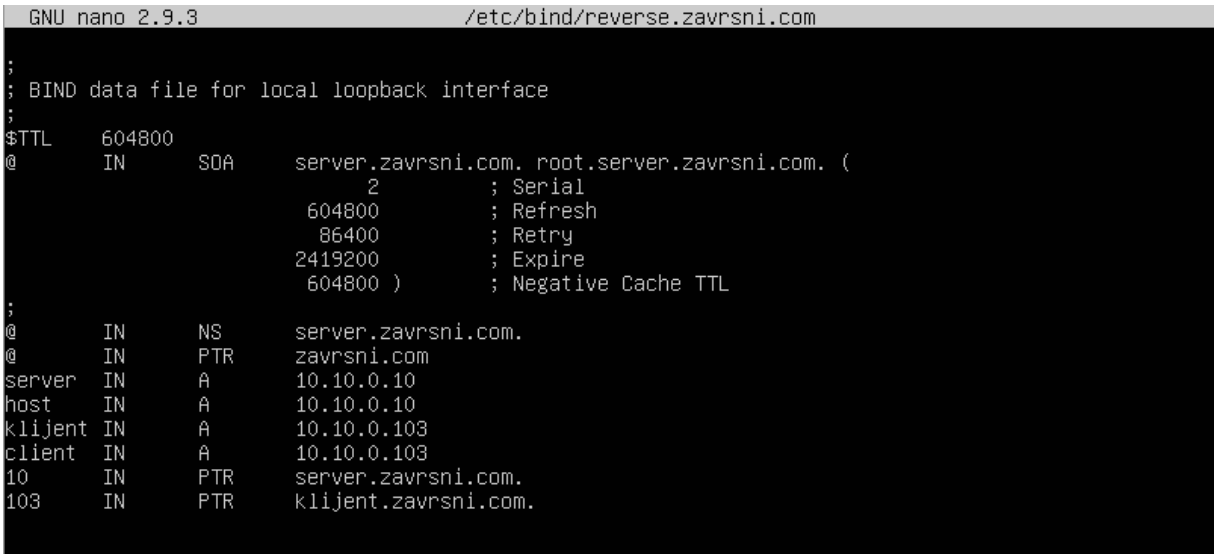
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     server.zavrsni.com. root.server.zavrsni.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      server.zavrsni.com.
@       IN      A       10.10.0.10
server  IN      A       10.10.0.10
host    IN      A       10.10.0.10
klijent IN      A       10.10.0.103
client  IN      A       10.10.0.103
```

Slika 14: forward DNS konfiguracija (slika autora)

Naziv domene je zapisana pod poljem SOA odnosno „Start of authority“, a glasi „server.zavrsni.com“ čime se poslužitelj označava kao autoritativan DNS server za danu domenu. „NS“ polje odnosi se na „Name server“ vrijednost gdje se umjesto *default* vrijednosti „localhost“ upisuje novi naziv poslužiteljskoga računala. Polje „A“ označava IPv4 adrese stoga se na tom mjestu zapisuju vrijednosti IP adresa zajedno s njihovim nazivima prije toga. Polje „IN“ znači „internet“. Kao što je prethodno navedeno, za jedno računalo (jednu IP adresu) moguće je navesti više naziva, pa je ovdje za poslužitelja upisan naziv „server“ i „host“, a za klijentsko računalo sa statičkom IP adresom dani su nazivi „klijent“ i „client“. [9]

Slična konfiguracija može se vidjeti i u *reverse* datoteci gdje je jedina razlika u upisivanju prvo IP adresa, a nakon toga u polju „A“ njima dodijeljene nazive. Još jedna od razlika je postojanje pokazivača gdje se u polju „PTR“ upisuje naziv računala do kojega se

dolazi putem hijerarhijski posljednjega broja u IP adresi. Konfiguracija je detaljnije prikazana na sljedećoj slici:



Slika 15: reverse DNS konfiguracija (slika autora)

Ovako postavljena konfiguracija može se provjeriti putem sljedećih komandi [9]:

```
sudo named-checkconf -z /etc/bind/named.conf
sudo named-checkconf -z /etc/bind/named.conf.local
sudo named-checkzone forward /etc/bind/forward.zavrsni.com
sudo named-checkzone reverse /etc/bind/reverse.zavrsni.com
```

Prvim dvjema komandama provjerava se konfiguracija općenito, a s posljednje dvije komande provjerava se da su *forward* i *reverse* DNS zone dobro postavljene. U slučaju da nema sintaksnih ili drugih grešaka na ekranu će se za svaku zonu ispisati „loaded“ što znači da su sve zone pravilno konfigurirane. Nakon toga je potrebno promijeniti prava na datoteci „/etc/bind/“ kako bi DNS server imao prava pristupa datoteci, koristeći sljedeće komande [9]:

```
sudo chown -R bind:bind /etc/bind
sudo chmod -R 755 /etc/bind
```

Na kraju je potrebno pokrenuti servis, omogućiti ga pri paljenju poslužitelja i provjeriti status naredbama:

```
sudo systemctl start bind9
sudo systemctl enable
sudo systemctl status bind9
```

gdje će se u slučaju dobre konfiguracije ispisati da je servis pokrenut i da je aktivan. Nakon što je servis pokrenut i aktivan sljedeći je korak konfiguracija vatrozida kako bi promet nesmetano prolazio kroz njega. Naredbom ispod u ugrađenom vatrozidu omogućen je „bind9“ DNS servis:

```
sudo ufw allow bind9
```

Prije finalnog testiranja DNS servera treba se konfigurirati sadržaj datoteke na putanji „etc/network/interfaces“, u datoteci gdje se prethodno namjestila statička IP adresa za poslužitelja. U toj će se datoteci sada ispod internog mrežnog adaptera „enp0s8“ dodati dva retka koja se odnose na DNS server, a od kojih je prvi polje naziva „dns-search“ s vrijednosti naziva domene, a drugi „dns-nameserver“ čija se vrijednost postavlja na IP adresu poslužitelja. Gotova konfiguracija vidljiva je na slici ispod.

```
  GNU nano 2.9.3                        /etc/network/interfaces

auto lo
iface lo inet loopback
auto enp0s8
iface enp0s8 inet static
address 10.10.0.10
netmask 255.255.255.0
dns-search zavrsni.com
dns-nameserver 10.10.0.10
auto enp0s3
iface enp0s3 inet dhcp
```

Slika 16: interfaces datoteka – DNS (slika autora)

Posljednja datoteka koju je potrebno izmijeniti ona je na putanji „/etc/resolv.conf“ u kojoj se nalazi IP adresa DNS servera koji poslužuju računalo. Ipak, pri otvaranju te datoteke na njenom vrhu piše kako se sve ručno upisane izmjene automatski brišu. Jedan način kako se to može razriješiti je instalacija novog servisa čija će jedina zadaća biti redovito ažuriranje ove datoteke kako bi izmjene ostale stalne [10]. Paket „resolvconf“ nakon isntalacije kreira datoteku na putanji „/etc/resolvconf/resolv.conf.d/head“ u kojoj se upisuju željene vrijednosti. U ovom slučaju to su adresa i naziv DNS servera, dakle vrijednosti 10.10.0.10 i „zavrsni.com“, nakon čega se servis ponovno pokreće i promjene ostaju stalne. Sadržaj „/etc/resolv.conf“ datoteke tada će biti kako slijedi:

```
  GNU nano 2.9.3                          /etc/resolv.conf

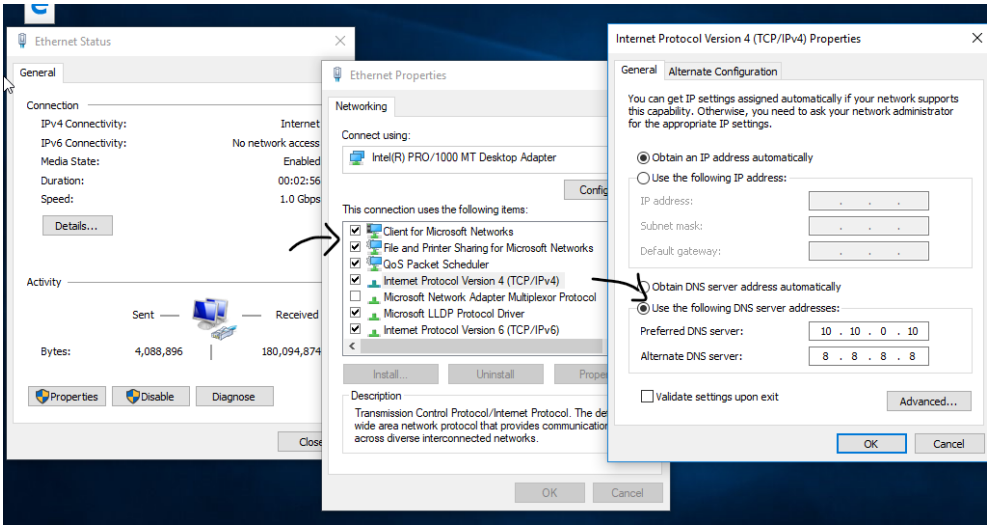
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
# 127.0.0.53 is the systemd-resolved stub resolver.
# run "systemd-resolve --status" to see details about the actual nameservers.

nameserver 10.0.2.15
search zavrsni.com
nameserver 127.0.0.53
```

Slika 17: resolv.conf datoteka (slika autora)

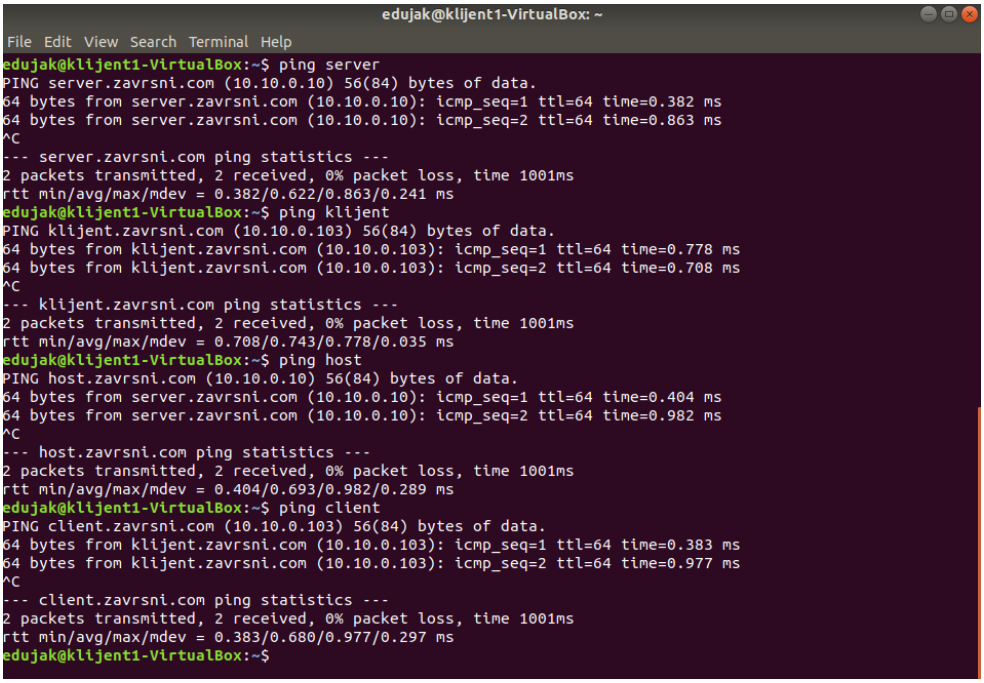
Drugi DNS server u datoteci *default* je poslužitelj koji je dodijeljen eksternom mrežnom adapteru i nije ga potrebno mijenjati. Ponovnim pokretanjem računala ili DNS servisa sve promjene se spremaju i moguće ga je testirati.

Na Ubuntu klijentima nije potrebna posebna konfiguracija budući da su DNS podatke povukli direktno od svojeg *gateway* poslužitelja, no Windows operacijski sustav nešto je drukčiji pa je ondje bilo potrebno ručno upisati adresu DNS servera. Na sljedećoj slici prikazan je postupak dolaska do postavki i prikaz upisane DNS adrese.



Slika 18: Konfiguriranje DNS servera na Windows OS-u (slika autora)

Nakon ovakve konfiguracije moguće je da će pri prvom preuzimanju podataka od DNS servera Windows upozoriti na grešku u rezoluciji, no prilikom testiranja očigledno je da je sve u redu. Na sljedećoj slici prikazan je primjer *pinganja* s jednog klijenta na statički postavljenog klijenta i poslužitelja.



Slika 19: Rezolucija u internoj mreži (slika autora)

Ovime se završava konfiguracija DNS servera, a u idućem poglavlju kao posljednjem koraku prije izlaska klijenata na internet bit će prikazana implementacija skripte za efikasan *port forwarding*.

4.2.2. Port forwarding

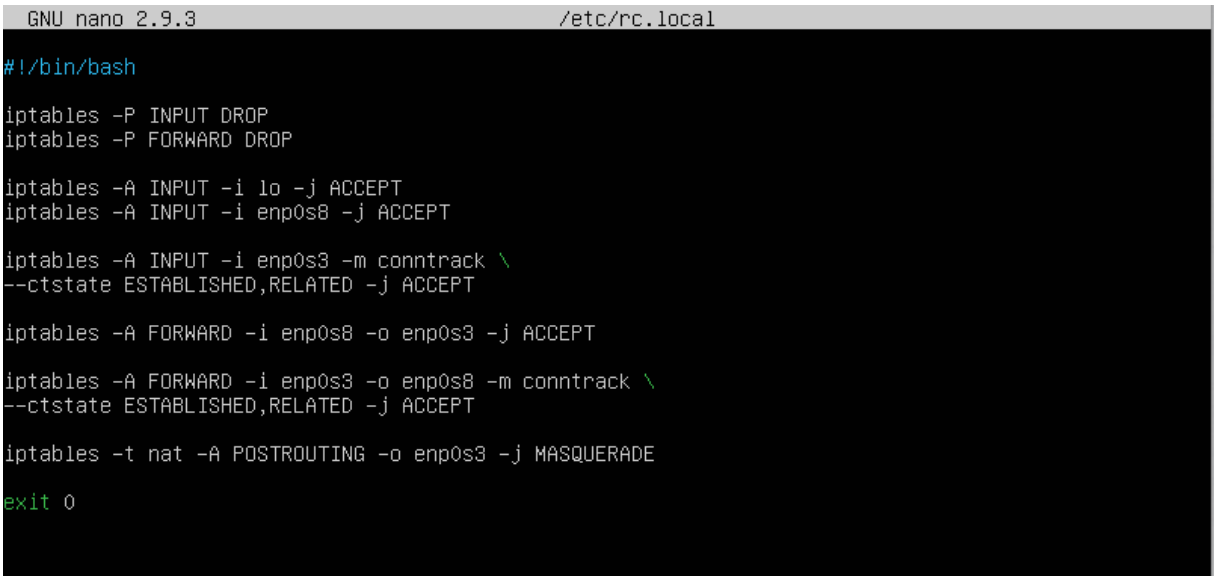
U poglavlju o DHCP-u konfiguriran je servis čija je glavna zadaća dodjeljivanje privatnih IP adresa i usmjeravanje klijenata na poslužitelja kao prespojnika. Ipak, dodijeljene adrese su privatne (zbog čega je korišten bazen adresa 10.10.0.x, a ne neki drugi), te s njima računala ne mogu pristupati internetu. Zadaća prespojnika je da privatne adrese zamijeni s javnom IP adresom, zbog čega će sva računala na mreži imati jednaku izlaznu IP adresu [11]. Važan dio prevođenja privatnih u javne IP adrese je određivanje porta na kojem će vanjski poslužitelji čekati odgovor od klijenata unutar mreže, zbog čega je važno ovu funkciju omogućiti na poslužitelju. Nadalje, konfiguracija kreirana na ovome poslužitelju koristi dva mrežna adaptera čija je zadaća objašnjena u prethodnim poglavljima. Kako bi se promet s unutarnjeg proslijedio na vanjski mrežni adapter koji ima pristup internetu kreirat će se svojevrsni most koristeći skriptu čime će klijenti u internoj mreži moći komunicirati s poslužiteljima na internetu. U sljedećih nekoliko koraka prvo će se omogućiti vatrozid ako on već nije bio omogućen te je u datoteci na putanji „/etc/ufw/sysclt.conf“ potrebno omogućiti *port forwarding.* Naredbe za to su sljedeće [7]:

```
sudo ufw enable
sudo nano /etc/ufw/sysctl.conf
```

Redak koji je potrebno otkomentirati je „net/ipv4/ip_forward=1“, a skriptu je potrebno napisati u datoteku na putanji „sudo nano /etc/rc.local“. Datoteka se treba moći pokretati zbog čega se njena svojstva treba postaviti putem sljedeće naredbe [7]:

```
sudo chmod 755 /etc/rc.local
```

Na sljedećoj slici prikazan je sadržaj datoteke s gotovom skriptom [7].

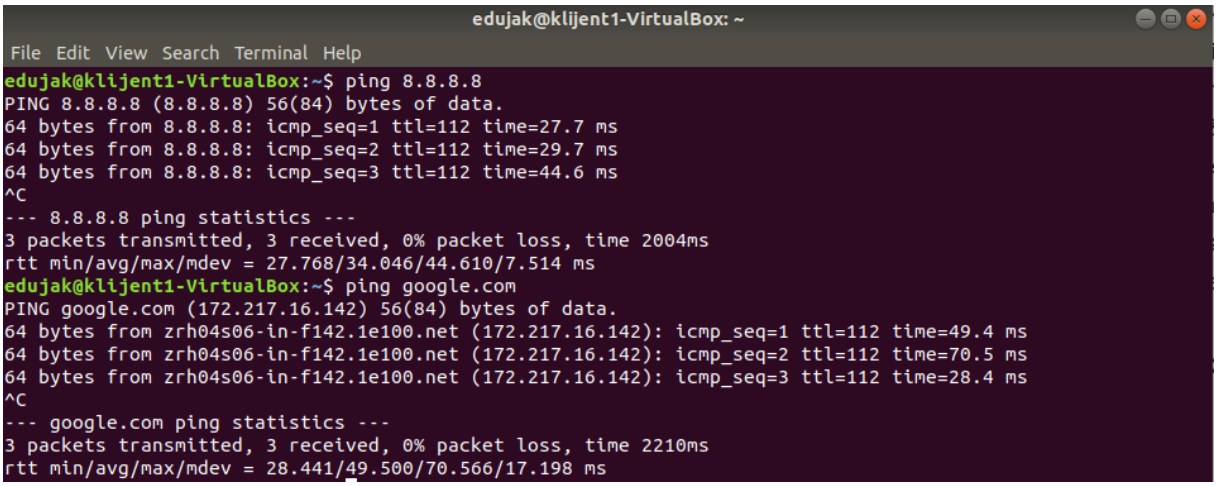


Slika 20: rc.local skripta (slika autora)

Prije pojašnjavanja skripte i njezine funkcije, važno je definirati još jedan pojam – *iptables.* To je pojam koji se odnosi na *routing* u kontekstu prespojnika, a uloga mu je konfiguracija prespojnika kao takvog i mrežnih adaptera u njem. Polja INPUT, FORWARD, OUTPUT i POSTROUTING dio su *iptables* mogućnosti kojima se specificira procesuiranje mrežnih paketa te se kao takvi koriste i u ovoj skripti. INPUT omogućava procesuiranje paketa netom prije nego su dostavljeni lokalnome procesu, a OUTPUT nakon. FORWARD se odnosi na pakete koji dolaze od *gateway* računala putem jednog mrežnog adaptera i izlaze iz drugog, zbog čega je posebno važan u ovome slučaju, a POSTROUTING omogućava procesuiranje paketa koji su tek došli od jednog mrežnog adaptera u računalo. [12]

Definiranjem osnovnih pojmova i naredbi moguće je shvatiti namjenu skripte i svakog njenog retka odnosno linije. Prva i zakomentirana linija koda računalu govori kako se radi o *bash* skripti stoga je vrlo važno da se ne briše iz nje. Redovi nakon toga omogućavaju procesuiranje paketa prije i nakon procesuiranja, nakon čega se na mrežnom adapteru „enp0s8“ omogućava ulazni promet paketa. Isto se definira i na eksternom mrežnom adapteru „enp0s3“. Dakle, nakon što su paketi došli na vanjski mrežni adapter, oni se funkcijama FORWARD i ACCEPT prenose na unutarnji mrežni adapter kako bi se mogli podijeliti na željenu destinaciju unutar lokalne mreže. Isto se omogućava i sa internog na eksterni mrežni adapter kako bi računala iz interne mreže mogla slati podatke na internet. Posljednjom linijom definira se POSTROUTING funkcija kako bi se svi paketi mogli nesmetano procesuirati.

Ponovnim pokretanjem računala skripta se pokreće i sva računala u mreži sada mogu neometano slati i primati podatke s interneta. Dokaz tome vidljiv je na slici ispod.



Slika 21: Pinganje udaljenih računala (slika autora)

Računalo „klijent_1“ može neometano *pingati* udaljena računala. Izvršavanje „traceroute“ naredbe također je vidljivo kako klijent do adrese 8.8.8.8 dolazi putem poslužitelja i adrese 10.10.0.10.

4.3. Proxy server i web cache

Mnoga poduzeća danas žele robusnu internu mrežu, ali ujedno i izlaz na internet na svakome računalu. Uz porast broja korisnika interneta raste i broj opasnosti na njemu stoga je posebno važno paziti na sigurnost informacija u internoj mreži – pogotovo u vremenu kad su informacije nerijetko skuplje od fizičkih tvorevina. Sve veća opasnost također su društvene mreže koje sakupljaju podatke o svojim korisnicima i njihovoj aktivnosti na internetu, a poslodavcima je tada u interesu takve stranice blokirati na radnome mjestu i osigurati svoje podatke i na toj fronti. Sve to i još više pruža proxy server, koji uz dodatan sloj sigurnosti omogućava praćenje internetske aktivnosti zaposlenika u poduzeću, ali i donekle ubrzava surfanje po internetu. Naime, *web cache* je funkcija koja prilikom prvog učitavanja stranice informacije o njoj pohranjuje na poslužitelja na kojemu je instalirana kako bi svako sljedeće računalo koje traži pristup toj stranici rezultat dobilo brže nego da se svaki puta stranica preuzima ispočetka. U sljedećem poglavlju tako će biti opisana instalacija ovog web servisa na poslužitelju, prikazana konfiguracija i funkcionalnost na klijentima.

4.3.1. Implementacija Proxy servera

Kao proxy server na poslužitelja je instaliran servis „Squid proxy“ koji će se pokazati kao najbolje rješenje što zbog jednostavnosti konfiguracije, što zbog efikasnosti koju pruža svim klijentima u mreži. Nakon instalacije paketa „squid“, servis je potrebno pokrenuti, omogućiti i provjeriti njegov status. Status bi trebao biti postavljen na Active, ako nije bilo nikakvih grešaka u instalaciji.

Prvi korak u personaliziranoj konfiguraciji proxy servera je omogućavanje pristupa proxy serveru što se može napraviti uređivanjem datoteke na putanji „/etc/squid/squid.conf“. Potrebno je promijeniti redak „http_access“ s vrijednosti „deny all“ u „allow all“ [13]. U ovoj je datoteci također moguće promijeniti port servera no za potrebe ovog projekta to nije bilo potrebno stoga je port ostavljen na *default* vrijednost od 3128. Nakon svake promjene servis je potrebno ponovno pokrenuti. Još jedna postavka koja se može konfigurirati unutar datoteke na istoj putanji je definiranje IP adresa koje imaju pristup proxy serveru kako bi samo određena klijentska računala mogla koristiti isti. Ova se postavka omogućava upisivanjem retka „acl localnet src“ s vrijednosti IP adrese kojoj se pruža pristup.

Proxy server već je sada moguće koristiti, no zbog pojačane sigurnosti u datoteku „/etc/squid/squid.conf“ dodani su sljedeći redovi:

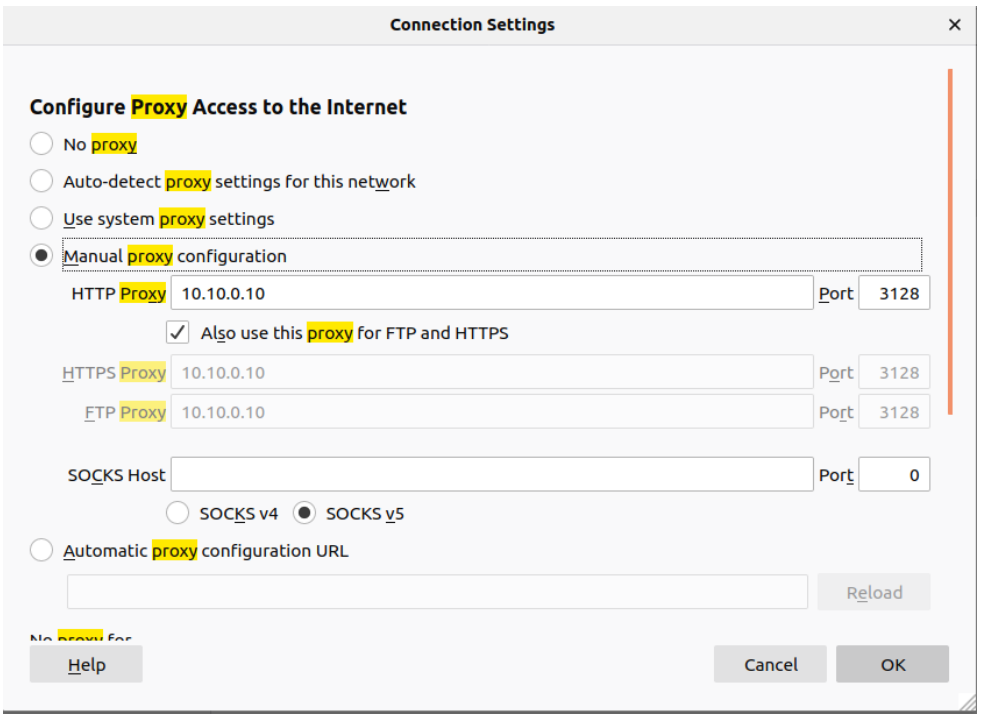
```
acl bad_urls dstdomain "/etc/squid/blacklisted_sites.acl"
http_access deny bad_urls

acl blockkeywordlist url_regex "/etc/squid/blockkeywords.lst"
http_access deny blockkeywordlist
```

Slika 22: Squid config (slika autora)

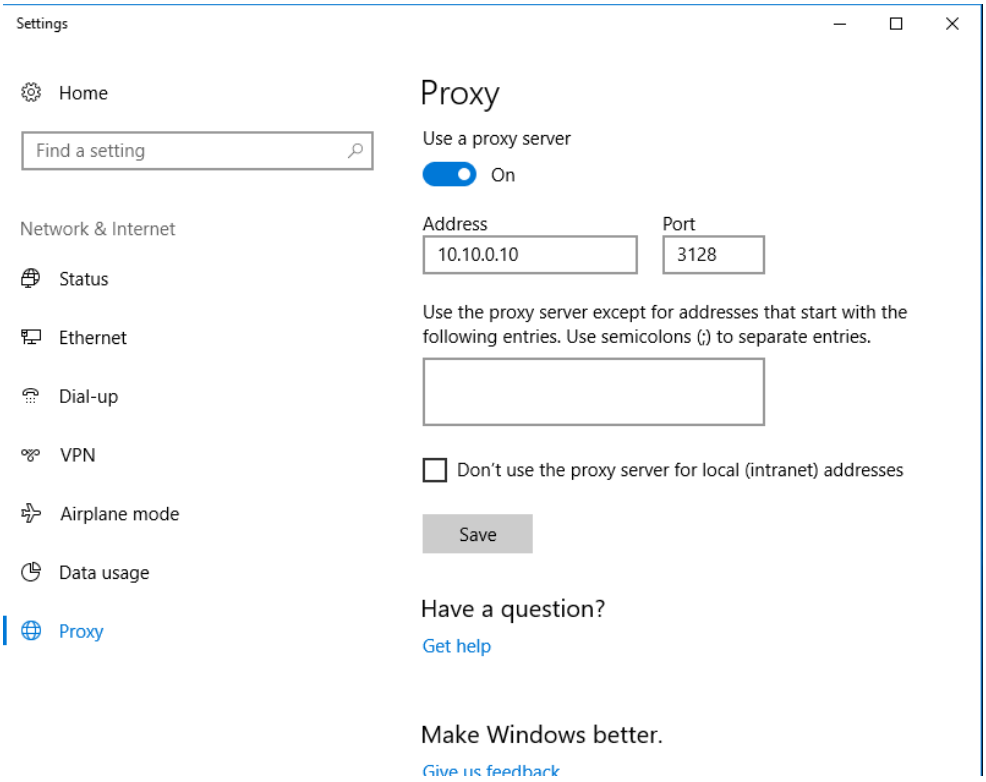
Prva postavka odnosi se na blokiranje neželjenih URL adresa, a druga na blokiranje ključnih riječi. Obje postavke vode na putanje vanjskih datoteka čiji su nazivi proizvoljni. Sadržaj datoteke „/etc/squid/blacklisted_sites.acl“ jest „.facebook.com“ zbog toga što se klijentskim računalima ne želi dozvoliti pristup društvenoj mreži facebook, kao niti njenim pod-domenama što se naznačilo točkom prije URL-a. Sadržaj sljedeće datoteke, „/etc/squid/blockkeywords.lst“ je u obliku liste „foi, instagram, gmail“ zbog toga što korisnici ne smiju posjećivati navedene stranice. Prije testiranja konfiguracije na klijentskim računalima je potrebno promijeniti određene postavke. I za Ubuntu i za Windows operacijski sustav postavke su vrlo slične – nalaze se u postavkama, a moguće ih je promijeniti i direktno na pretraživaču interneta. Iz tog razloga na Ubuntu klijentskim računalima proxy je postavljen koristeći web pretraživač, a na Windows računalu direktno u mrežnim postavkama.

Na Ubuntu računalima u Firefox pretraživaču proxy se konfigurira u postavkama, a postavljena je na sljedeći način:



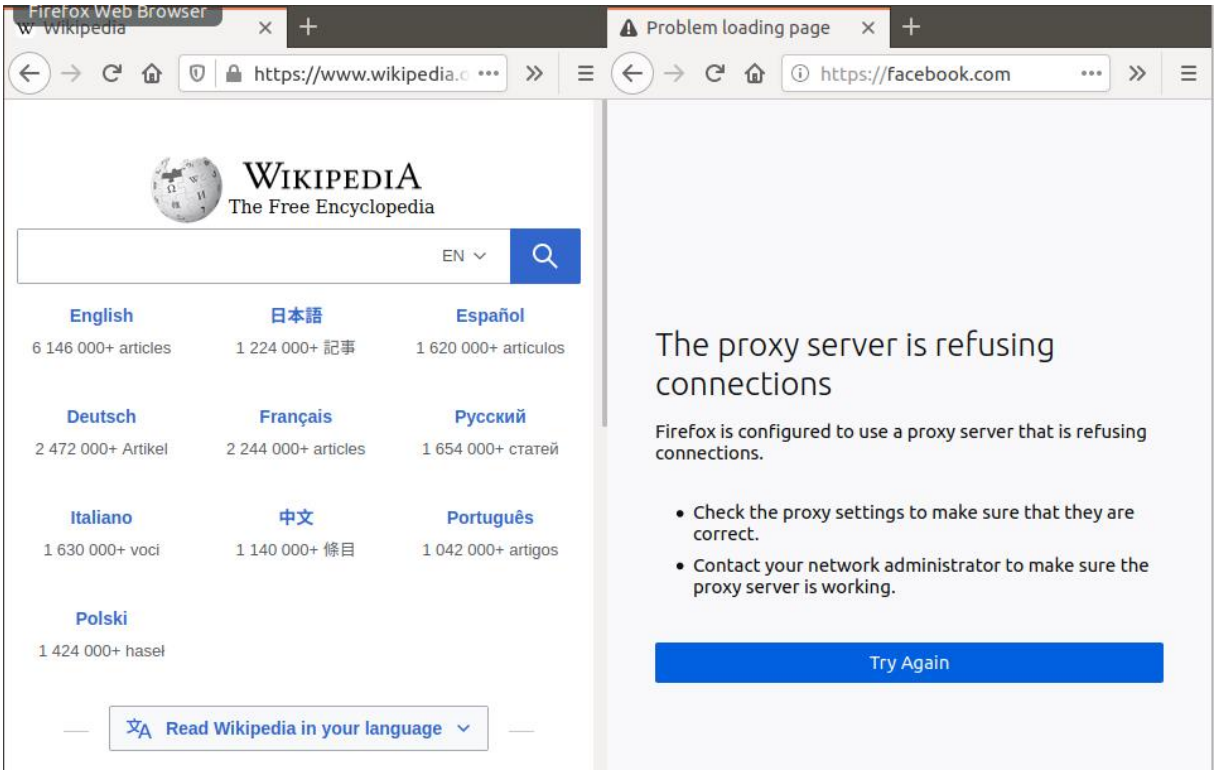
Slika 23: Proxy: Firefox (slika autora)

Dok se na Windows 10 operacijskome sustavu postavke za proxy nalaze unutar sustavskih postavki, a postavljene izgledaju ovako:



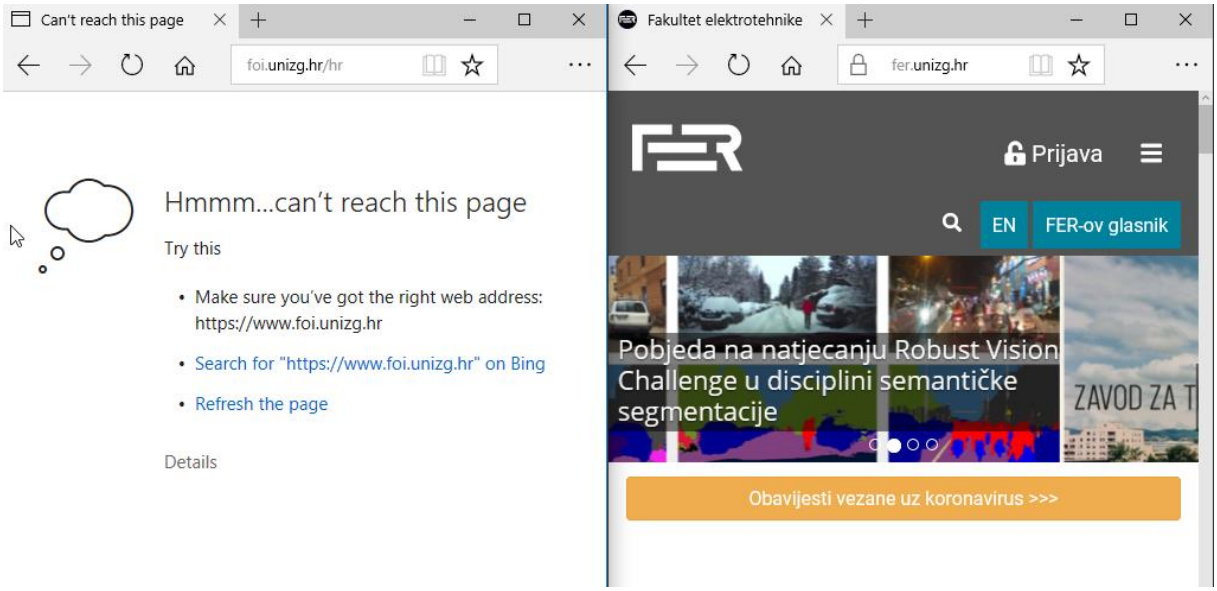
Slika 24: Windows 10: Proxy (slika autora)

Prateći teoretsku osnovu proxyja, sva bi računala povezana na proxy sada trebala imati jednaku IP adresu, sadržaj pristupljenih stranica sprema se u web cache, a sav se promet može pratiti direktno na poslužitelju. Osim toga, nijedno od povezanih računala ne smije imati pristup društvenoj mreži facebook.com, kao ni ključnim riječima „foi“, „instagram“ i „gmail“.



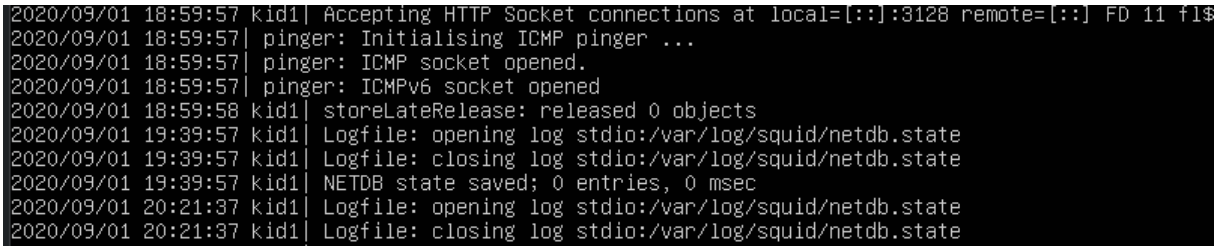
Slika 25: Provjera proxy konfiguracije 1 (slika autora)

Na Ubuntu klijentima proxy ne dopušta povezivanje na društvenu stranicu facebook.com, ali se Wikipedia učitava bez problema. Slično je i na Windows klijentu gdje se zbog nedozvoljene ključne riječi stranica FOI-ja ne može učitati, no stranica FER-a normalno se učitala.



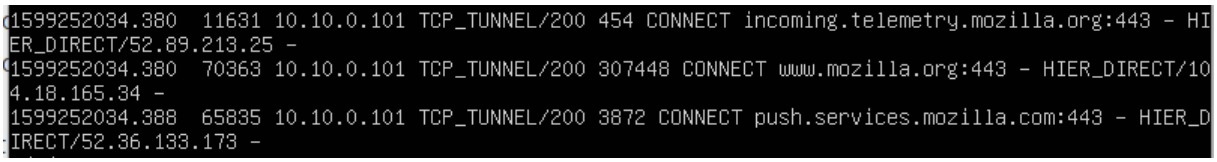
Slika 26: Provjera proxy konfiguracije 2 (slika autora)

Proxy server radi kako je i konfiguriran, no potrebno je još pregledati sprema li se web cache kako bi trebao te zapisuju li se stranice u *log* na poslužitelju. Web cache podaci nalaze se na putanji „/var/log/squid/cache.log“, dok se *access log* nalazi na putanji „/var/log/squid/access.log“ [13].



Slika 27: Web cache – isječak (slika autora)

Na slici iznad vidljiv je isječak datoteke „cache.log“ i njenog sadržaja koji potvrđuje spremanje keša u ovu datoteku. Osim isječka datoteke, keš je očigledan i u vremenu učitavanja stranice gdje se vrijeme učitavanja prvog posjeta stranici razlikuje od drugog i svakog sljedećeg posjeta.



Slika 28: Proxy log – isječak (slika autora)

Isječak iznad zanimljiviji je budući da se radi o razumljivim informacijama koje se lako mogu iščitati – posljednje računalo internetu je pristupilo s adrese 10.10.0.101, a preuzeti su podaci sa stranice mozilla.org, što se poklapa s onime što se zaista i dogodilo.

Ovime se zaključuje instalacija i konfiguracija Proxy i web cache servera budući da je servis pravilno instaliran i funkcionira kako bi i trebao.

4.4. Skriptiranje – automatizacija postavki

Svaki korisnik Linuxa barem se jednom susreo sa skriptiranjem i ogromnom lepezom mogućnosti koje ono pruža. Od automatizacije procesa, do masovnih instalacija aplikacija, kreiranja datoteka i slično. Tematika koja se provlači kroz ovaj rad jest implementacija lokalne, interne mreže unutar poduzeća stoga se postavlja pitanje - što učiniti sa svakim novim računalom koje se priključuje u mrežu? Ponovna instalacija i konfiguracija svih prethodno opisanih servisa na svako zasebno računalo gubljenje je vremena te se upravo ovdje skriptiranje pokazalo kao najbolji odabir. Skripta koja se pokreće jednom na računalu treba moći konfigurirati proxy (budući da se DHCP i DNS konfiguriraju automatski, putem *gateway* poslužitelja), instalirati sve aplikacije koje su potrebne korisniku za svakodnevni rad i na pozadini kreirati kratke upute za korištenje istih aplikacija. No, to je samo u slučaju Ubuntu

klijenata, za Windows klijente treba postojati druga skripta, ali s nešto proširenim mogućnostima. Na Windows klijentu naime potrebno je ručno konfigurirati DNS server stoga se i to nameće kao dodatni zadatak skripte.

Za potrebe ove lokalne mreže aplikacija koja se treba instalirati svakome korisniku može biti FileZilla (kako bi se demonstrirale mogućnosti File sharing i web servera u sljedećim poglavljima), a na pozadini će se kreirati tekstualna datoteka s uputama za povezivanje na poslužitelja koristeći također FileZillu. U sljedećem poglavlju bit će opisane dvije skripte – jedna za Ubuntu, a jedna za Windows klijente, njihove mogućnosti i na kraju rezultat.

4.4.1. Novo računalo u mreži

Prva od dviju skripti napisana je za Ubuntu operacijske sustave a sastoji se od tri dijela. U prvome se dijelu instaliraju sve potrebne aplikacije (FileZilla, Samba i cfs-utils) nakon čega se kreira mapa i povezuje s mrežnim dijeljenjem koje će se odvijati na na adresi 10.10.0.103 gdje će klijent 2 dijeliti datoteke. Drugi dio posvećen je kreiranju tekstualne datoteke na pozadini s uputama za korištenje FileZille, a u trećem se dijelu konfigurira proxy server na razini cijelog računala.

```
sudo apt-get update
sudo apt-get install filezilla -y
sudo apt-get install smbclient -y
sudo apt-get install cifs-utils -y
sudo mkdir /media/dijeljenje
sudo mount -t cifs //10.10.0.103/test /media/dijeljenje

echo "UPUTE ZA KORIŠTENJE FILEZILLE
      server: 10.10.0.10
      korisnicko ime: klijent
      lozinka: pustime
      port: 21 ili 22" > ./Desktop/upute.txt

sudo printf
"http_proxy=\"10.10.0.10:3128\"\nhttps_proxy=\"10.10.0.10:3128\
"\nftp_proxy=\"10.10.0.10:3128\"\nno_proxy=\"localhost,127.0.0.
1,::1\"" >> /etc/environment
sudo printf "pref(\"browser.startup.homepage\",
\"server.zavrsni.com\");" >> /etc/firefox/syspref.js
```

Dakle, nakon ažuriranja repozitorija i instaliranja potrebnih aplikacija i servisa, pomoću naredbe „echo“ u datoteku se upisuje zadani tekst. Posljednji dio za konfiguraciju proxy-ja ujedno je bio i najteži za konfigurirati budući da je bilo potrebno osmisliti način kako uređivati *root* datoteke i spremati izmjene bez upisivanja lozinke što je inače potrebno. Osim toga, naredba „echo“ koja je korištena par redaka prije pokazala se kao neefikasna pri upisivanju teksta u *root* datoteke zbog neprepoznavanja simbola „/n“ koji označava novi red stoga je korišten „printf“. Nakon što je sintaksa bila dobra, novi je problem bio u navodnim znakovima nad kojima je napravljen *break* koristeći kose crte kako bi se normalno prepoznali u tekstualnoj datoteci. Također je važno napomenuti da se za pokretanje ove skripte potrebno ulogirati kao root korisnik koristeći naredbu „sudo -s“ upravo zbog uređivanja *root* dokumenata.

Jezik za skriptiranje kakav koristi Windows operacijski sustav nešto je kompleksniji i teži za korištenje budući da za razliku od Linux distribucija običnome korisniku nije potrebno koristiti komandnu liniju niti skriptiranje. Iz tog je razloga ova skripta nešto drukčija, a postavljanje proxy servera realizirano je na drukčiji način – sve drugo ima jednaku svrhu i rezultate.

```
[System.Net.ServicePointManager]::SecurityProtocol=[System.Net.
SecurityProtocolType]'Tls,Tls11,Tls12'
Invoke-WebRequest -Uri 'https://download.filezilla-
project.org/client/FileZilla_3.50.0_win64_sponsored-setup.exe'
-OutFile FileZilla_3.50.0_win64_sponsored-setup.exe
.\FileZilla_3.50.0_win64_sponsored-setup.exe

$tekst = "UPUTE ZA KORIŠTENJE FILEZILLE
          server: 10.10.0.10
          korisnicko ime: klijent
          lozinka: pustime
          port: 21 ili 22"
Set-Content -Path "C:\Users\klijent3\Desktop\UPUTE.txt" -Value
$tekst
$varijabla = Get-NetAdapter | Select-Object InterfaceIndex
$varijabla_2 = $varijabla.InterfaceIndex
set-DnsClientServerAddress -InterfaceIndex $varijabla_2 -
ServerAddresses ("10.10.0.10","8.8.8.8")

New-PSDrive -Name "Z" -Root "\\10.10.0.103\test" -Persist -
PSProvider FileSystem -Scope Global
```

```
Start-Process "ms-settings:network-proxy"
[System.Windows.MessageBox]::Show('Use a proxy server -> On
Address: 10.10.0.10
Port: 3128')
```

Prije objašnjavanja same skripte važno je napomenuti kako se za skriptiranje u Windows operacijskim sustavima može koristiti Windows PowerShell ISE sučelje koje svojom jednostavnošću stoji uz bok popularnom Linuxovom terminalu. Skripte se mogu spremiti u *default* PowerShell tipu podataka ili kao „.exe“ datoteka. Za ovaj slučaj dovoljno je datoteku spremiti kao PowerShell datoteku.

Na samome početku skripte koristeći „InvokeWebRequest“ preuzima se .exe datoteka koja se u retku nakon otvara kako bi se pokrenula instalacija. Nakon toga, baš kao i u Linux skripti, kreira se kratka tekstualna datoteka s uputama za korištenje FileZille i svim potrebnim podacima za pristup serveru koja se naredbom „Set-Content“ sprema na pozadinu korisnika. Sljedećih nekoliko redaka odnosi se na konfiguraciju DNS servera stoga se prvo otkriva broj mrežnog adaptera, nakon čega se na temelju tog broja pospremljenog u varijablu kreiraju postavke za DNS server [20]. Prva IP adresa je ona poslužitelja kako bi računalo moglo komunicirati s drugim računalima u mreži, a druga je osnovna DNS IP adresa za svako računalo – 8.8.8.8 (google.com). Naredbom „New-PSDrive“ započinju dvije linije koje se odnose na mapiranje mrežne lokacije koje dijeli računalo na adresi 10.10.0.103. Na Windows operacijskome sustavu tako je automatski kreirana mrežna datoteka kojoj se može pristupati i na ovom računalu u mreži. Na samome kraju, ponovno se najtežim pokazala konfiguracija proxy servera. Prvotna je ideja bila konfiguracija registara za zadano računalo no takav se način nije pokazao ispravnim, a internetski izvori suglasni su u tome kako se postavke proxy servera trebaju konfigurirati koristeći *Group policy* kojim upravlja *Domain controller* odnosno poslužiteljsko računalo nadređeno danom klijentskom računalu. Naposlijetku je ovaj problem riješen automatskim otvaranjem postavki na mjestu za konfiguraciju Proxy servera te izbacivanje *pop-up* poruke s kratkim uputama o postavljanju istoga.

5. Web server i File sharing

Razmjena podataka između klijenta i poslužitelja osnovna je premisa svih poslužiteljskih servisa kao i samog pojma poslužitelja. Reprezentativni primjer jedne takve razmjene je uloga poslužitelja kao web servera; odnosno, *hosting* stranice na poslužitelju i

upload podataka ili koda stranice na poslužitelja. U sljedećim poglavljima iz tog će razloga biti opisana instalacija „LAMP stack“ aplikacija, njihova konfiguracija i korištenje. Na poslužitelja će tada biti uploadana *dummy* stranica koja će se u stvarnome vremenu moći uređivati i pregledavati sa svih klijentskih računala u mreži koristeći FTP server i FileZilla aplikaciju na klijentima.

5.1. LAMP stack

LAMP stack je pojam poznat u *web development* području koji označava servise esencijalne za pokretanje i održavanje svake web stranice. L stoji za Linux, A za Apache, M za MySQL, a P za PHP. Linux označava server na kojemu je stranica pohranjena, Apache servis koji služi za *hosting* stranice, MySQL sustav za upravljanje bazama podataka, a PHP za programski jezik neophodan u razvoju svake web stranice. Ovi se servisi mogu izmjenjivati pa se tako za stranicu pohranjenu na Windows operacijskome sustavu kaže da koristi WAMP stack, ali LAMP je prvi pojam od kojega su proizašli svi drugi. MySQL također može biti MariaDB, a PHP – Perl. Nakon postavljanja svi servisa stranica će biti dostupna na kljentskim računalima putem adrese „server.zavrsni.com“ zbog DNS rezolucije koja je postavljena u jednom od prehodnih poglavlja, a phpMyAdmin koji će također biti instaliran bit će dostupan putem adrese „server.zavrsni.com/phpmyadmin“.

5.1.1. Apache server

Apache server je open source rješenje kojim se pruža brzo i jednostavno održavanje web stranica na velikom broju platformi, komercijalno ili ne [14]. Instalacija na Ubuntu server prilično je izravna i jednostavna, izvediva jednom naredbom:

```
sudo apt-get install apache2
```

Nakon instalacije i provjere statusa instaliranoga servisa serveru se već može pristupiti na poslužitelju, što je vidljivo i na slici ispod:



Slika 29: Pristup Apache serveru (slika autora)

U slučaju da se stranici ne može pristupiti iz prve, potrebno je Apache server dozvoliti u postavkama vatrozida koristeći naredbu [16]:

```
sudo ufw allow 'Apache'
```

Nakon ponovnog pokretanja Apache servisa putem linka „server.zavrsni.com“ se dolazi do stranice koja pokazuje na putanju na serveru „/var/www/html“ što je *default* putanja za spremanje stranice na Apache serveru. U poglavlju 5.1.4. bit će detaljnije opisano kako korisnici mogu pristupati toj datoteci i u njoj mijenjati datoteke. Nakon Apache servera potrebno je instalirati MySQL server.

5.1.2. MySQL server i phpMyAdmin

Glavna uloga MySQL servera je opsluživanje nove web stranice sa servisom za upravljanje bazama podataka. Jedan od jednostavnijih načina pristupa bazama na serveru je servis phpMyAdmin koji ide ruku pod ruku s MySQL serverom i kojem se pristupa, u slučaju ovoga projekta, putem adrese „server.zavrsni.com/phpmyadmin“. Baš kao i za Apache, instalacija MySQL servera jednostavna je no uključuje dodatan korak kojim se omogućava dodatna sigurnost za bazu gdje će se root korisniku dati lozinka. Koraci su zbog toga kako slijedi [15]:

```
sudo apt-get install mysql-server
sudo mysql_secure_installation
sudo apt install phpmyadmin php-mbstring php-gettext
sudo phpenmod mbstring
```

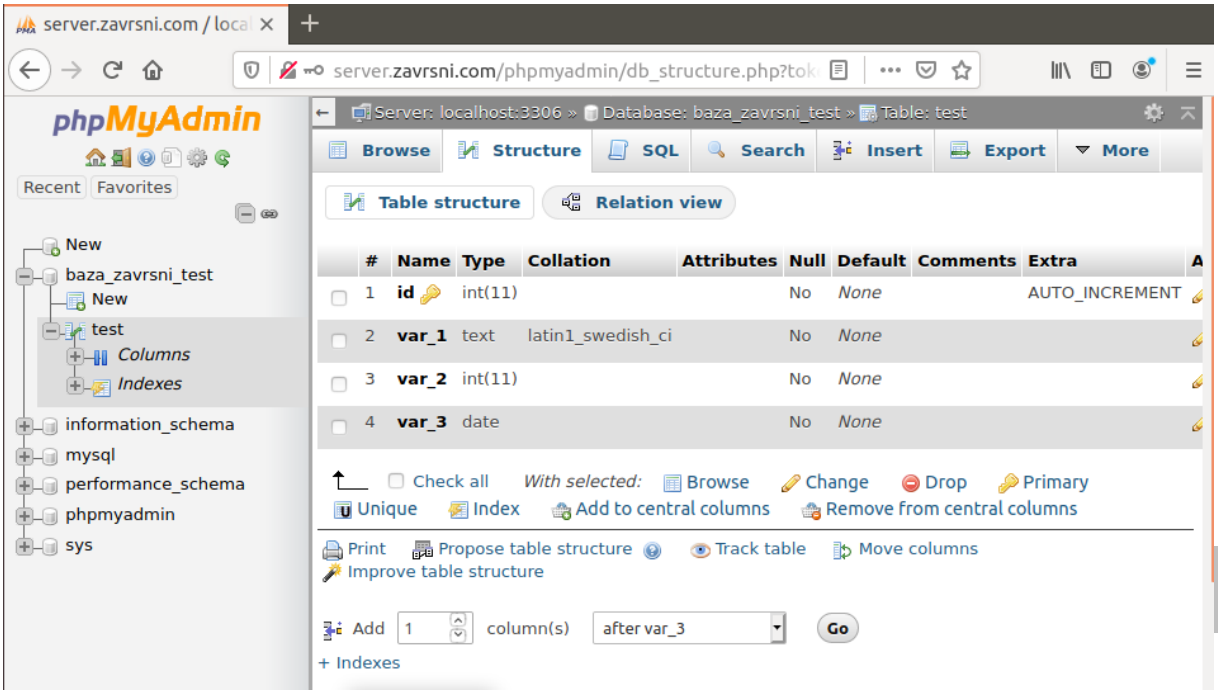
Nakon ovoga je dovoljno ponovno pokrenuti Apache server kako bi svi servisi bili pokrenuti. Problem koji ovdje nastaje je zbog lozinke koja se kod *root* korisnika autenticira pomoću „auth_socket“ konfiguracije, umjesto klasične „mysql_native_password“ [16]. Iz tog je razloga potrebno kreirati novoga korisnika koji će pristupati bazi, ili izmijeniti način autentikacije *root* korisnika. Oba načina mogu se napraviti unutar mysql servisa pozivanjem „sudo mysql“ naredbe u terminalu.

5.1.3. PHP

Posljednji korak u LAMP Stack instalaciji i konfiguraciji je instalacija PHP-a koristeći sljedeću naredbu:

```
sudo apt-get install php libapache2-mod-php
```

Nakon čega je konfiguracija gotova i može joj se pristupiti putem prethodno navede adrese. Na sljedećoj slici prikazana je instalirana instanca phpMyAdmin s kreiranom testnom bazom i testnom tablicom, čime se zaključuje konfiguracija LAMP stack-a.



Slika 30: Testna baza – phpMyAdmin (slika autora)

5.1.4. FTP server i pristup putem Filezille

Sada kada je konfigurirano sve za *hosting* web stranice na poslužitelju, slijedi još samo konfigurirati pristup datotekama stranice što će se ostvariti kombinacijom FTP servera i FileZilla aplikacije. Paket koji je potrebno instalirati je „vsftpd“, nakon čega se unutar vatrozida treba omogućiti promet na portovima na kojima FTP radi. Koraci za to su sljedeći [17]:

```
sudo ufw allow OpenSSH
sudo ufw allow 20/tcp
sudo ufw allow 21/tcp
sudo ufw allow 40000:50000/tcp
```

Budući da FTP radi na portovima 20 i 21 treba ih otvoriti, a portovi 40000 i 50000 pasivni su FTP portovi stoga se i njih treba ostaviti otvorene. Nakon konfiguriranja vatrozida potrebno je kreirati korisnika i dodijeliti mu lozinku te potrebna prava [17]:

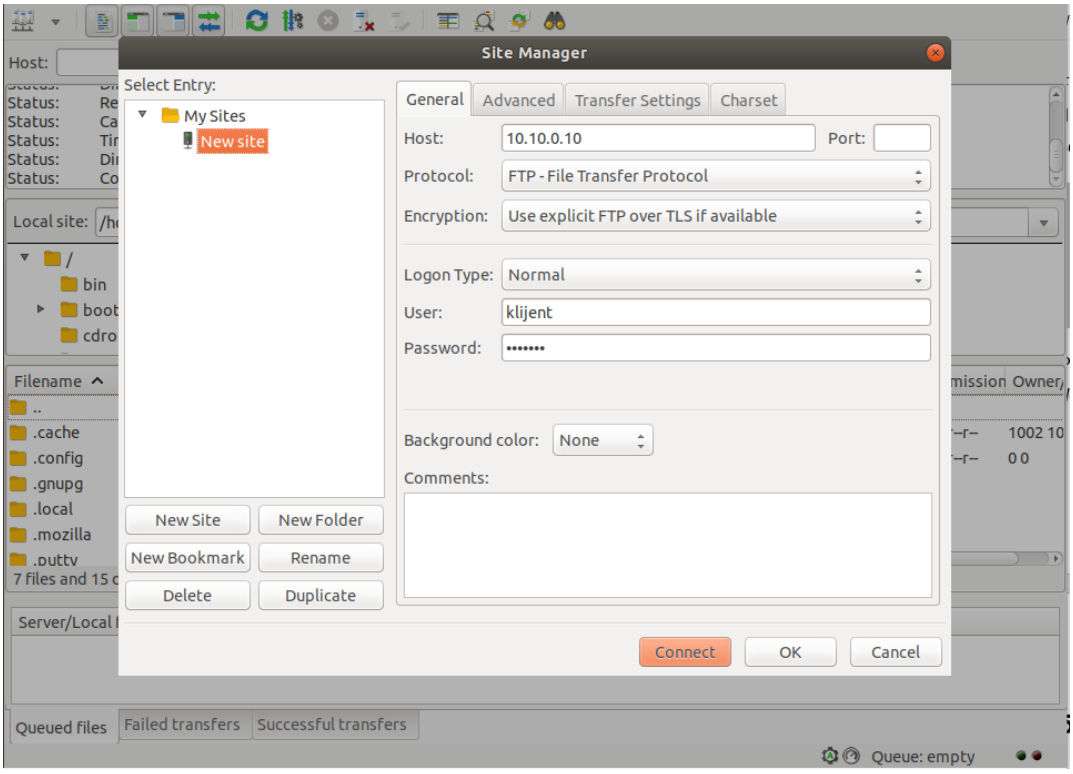
```
sudo adduser klijent
sudo usermod -d /var/www klijent
sudo chown klijent:klijent /var/www/html
```

Prvom linijom korisnik se kreira zajedno s lozinkom, a drugom i trećom linijom definira se direktorij kojemu će korisnik imati pristup. Budući da se radi o web serveru i korisnik treba imati prava mijenjanja samo direktorija u kojemu je web stranica, putanja je navedena kao „/var/www“ što je direktorij u koji je Apache usmjeren po *default* postavkama.

Posljednji koraci odnose se na mijenjenje datoteke „vsftpd.conf“ odnosno postavki za FTP server kako bi se korisniku omogućila prava mijenjanja te onemogućilo izlaženje iz danog direktorija [17]. Nakon navigacije do datoteke na putanji „/etc/vsftpd.conf“ sljedeći redci trebaju se nadodati u datoteku (neki od redaka već postoje, ali su zakomentirani.):

```
write_enable=YES
chroot_local_user=YES
local_umask=022
force_dot_files=YES
```

Posljednja linija obvezat će klijenta da pročita datoteke koje počinju s točkom koje se prema *default* vrijednosti ignoriraju. Nakon upisivanja podataka za novog korisnika u FileZillu, klijentsko računalo sada može vidjeti datoteke na danoj putanji i mijenjati ih.



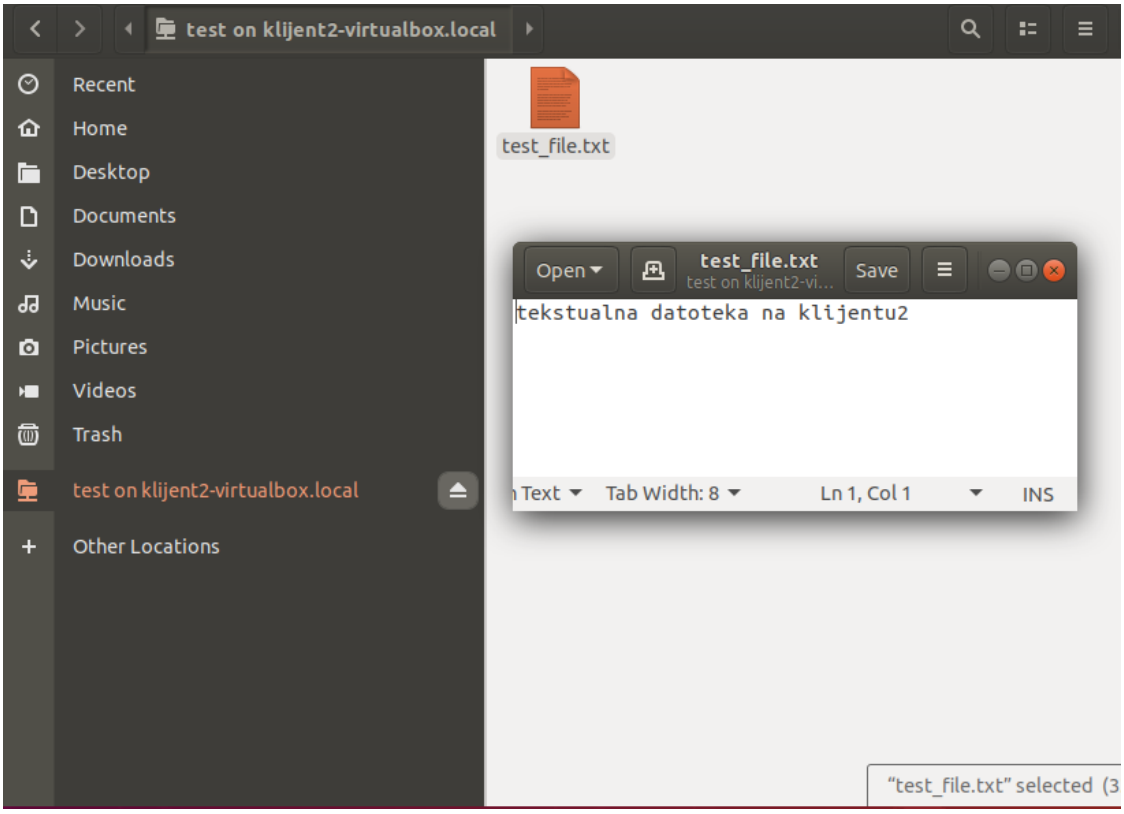
Slika 31: FileZilla podaci (slika autora)

5.2. File sharing među klijentskim računalima – Samba

Klijent – poslužitelj komunikacija u dijeljenju podataka najvažniji je oblik komunikacije u internoj mreži, no vrlo je važna komunikacija klijent – klijent. Demonstracija takve komunikacije ostvarit će se kroz jednostavno mrežno dijeljenje datoteke s klijentskog računala sa statički definiranom IP adresom koju će moći vidjeti i uređivati druga dva klijenta u mreži.

Nakon kreiranja nove datoteke bilo gdje unutar Klijenta 2, desnim klikom na nju otvara se dijalog s ponuđenim odabirom „Local Network Share“. Pritiskom na taj odabir otvara se prozor gdje se treba odabrati „Share this folder“ opcija. U slučaju da servis Samba nije prethodno instalirana na računalu, Ubuntu će ponuditi instalaciju iste stoga je potrebno samo prihvatiti instalaciju servisa. Moguće je također odabrati dopuštenje uređivanja datoteka u mapi, a ako se ta opcija ne odabere korisnici će moći samo pregledavati datoteke bez brisanja ili uređivanja. Također je moguće odabrati opciju „Guest access“ koja omogućava anoniman pristup datoteci. Pritiskom na „create share“ datoteka će sada moći biti vidljiva svim drugim računalima u mreži, budući da nisu konfigurirane nikakve zabrane. Unutar mape kreirana je obična tekstualna datoteka kako bi se mogle testirati mogućnosti na ostalim klijentskim računalima. Na drugom Ubuntu klijentskom računalu otvaranjem bilo koje mape s lijeve strane moguće je vidjeti općenitu hijerarhiju datoteka na tom računalu.

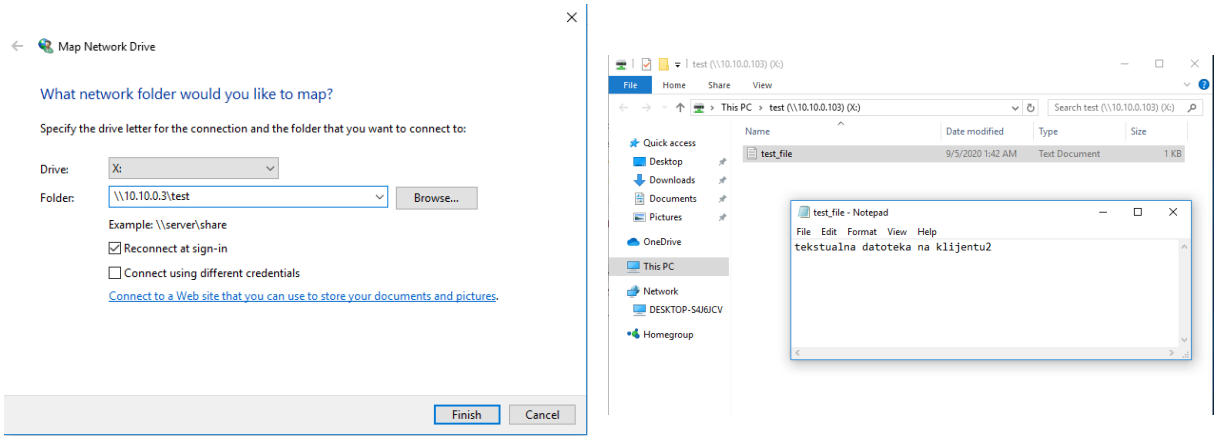
Pritiskom na „Other Locations“ zbog upaljenog mrežnog dijeljenja sada će biti vildljivo računalo Klijent 2, a otvaranjem tog računala datoteka kreirana za mrežno dijeljenje.



Slika 32: Prikaz mrežnog dijeljenja (slika autora)

Klijent sada može otvoriti mapu i kreiranu testnu tekstualnu datoteku.

Konfiguracija je nešto drukčija na Windows 10 klijentu no rezultati su jednaki. Otvaranjem bilo kakve mape ponovno se s lijeve strane može vidjeti hijerarhija datoteka, ali i opcija „Network“. Desnim klikom na tu opciju otvara se dijalog gdje je potrebno odabrati „Map Network Drive“, upisati IP adresu klijentskog računala zajedno s nazivom mape. Pritiskom na tipku „Finish“ otvara se željena datoteka.



Slika 33 i 34: Prikaz datoteke na Windows računalu (slika autora)

6. Zaključak

Internet je mnogim ljudima neshvatljiv i toliko ogroman da niti nema smisla pokušati uvesti smisao u njega. Ipak, ideja interneta vrlo je jednostavna i tek kad se replicira na vrlo maloj razini može se vidjeti kako svaki njegov servis ima smisao i svrhu bez obzira pokreće li se na malom kućnom poslužitelju kao virtualna mašina ili u nekoj velikoj farmi servera. Cilj ovoga rada zbog toga je bio demistificirati internet odnosno mrežu kao takvu i razlučiti sve servise i postupke koji su potrebni kako bi se jedna takva mreža kreirala – bez obzira koliko umanjena bila.

Tijekom izrade ovog završnog rada uspješno su odrađene svi zamišljeni ciljevi te je uspješno simulirana lokalna mreža s izlazom na internet putem poslužitelja koji je ujedno služio i kao web server za posluživanje web stranice. Budući da sam imala prethodno znanje o Windows serveru i njegovim servisima kao i o naredbenom retku Ubuntu distribucija najveći je izazov bio primijeniti to znanje na Ubuntu server, operacijski sustav bez grafičkoga sučelja. Na koncu se taj izazov pokazao vrlo zanimljivim jer me natjerao da razmišljam izvan okvira i svoje računalo vidim isključivo kroz naredbe, a ne šarene gumbiće na kakve sam dosada navikla.
Osim izazova kakav je bio Ubuntu server, implementacija DNS-a i port forwarding se pokazao kao najteži dio rada budući da mi je do tada koncept toga bio prilično apstraktan. Nakon DNS-a i port forwardinga na server su uspješno implementirani i drugi mrežni servisi kao što su proxy i web cache, a nakon toga i servisi rezervirani za web posluživanje stranice te FTP server.

Smatram kako će ovaj rad pomoći ljudima koji se sami žele okušati u umrežavanju vlastitoga poduzeća no za to dosada nisu imali vremena ili financijske sposobnosti. Iz tog razloga sam pokušala sve procese objasniti što je jednostavnije moguće i tako da budu razumljivi svima. Na kraju, kroz sve implementirane servise, aplikacije i skripte uspješno je uspostavljen funkcionalni servis za internu mrežu spreman za sve izazove kakve rad u poduzeću može ponuditi.

Popis literature

[1] K. Finley, „Linux Took Over the Web. Now, It's Taking Over the World“, 2016. [Na internetu]. Dostupno na: https://www.wired.com/2016/08/linux-took-web-now-taking-world/. [Pristupljeno: 15-kol-2020].

[2] J. LaCroix, Mastering Ubuntu Server: Master the art of deploying, configuring, managing, and troubleshooting Ubuntu Server 18.04, 2. izd. Birmingham: Packt Publishing Ltd., 2018.

[3] AMC The School Of Business, VM VirtualBox, 1. izd. Sabah: AMC The School Of Business, 2020.

[4] Netplan, „Netplan reference“, 2020. [Na internetu]. Dostupno na: https://netplan.io/reference/ . [Pristupljeno: 15-kol-2020].

[5] D. Medhi, K. Ramasamy, Network routing: Algorithms, protocols, and architectures, 2. izd. Cambridge: Morgan Kaufmann publishers, 2018.

[6] Debian, „dhcpd.conf - dhcpd configuration file“, 2020. [Na internetu]. Dostupno na: https://manpages.debian.org/unstable/isc-dhcp-server/dhcpd.conf.5.en.html . [Pristupljeno: 15-kol-2020].

[7] B. Valentinuzzi, „How To: Build a Simple Router with Ubuntu Server 18.04.1 LTS (Bionic Beaver)“, 2018. [Na internetu]. Dostupno na: https://www.ascinc.com/blog/linux/how-to-build-a-simple-router-with-ubuntu-server-18-04-1-lts-bionic-beaver/ . [Pristupljeno: 15-kol-2020].

[8] Oracle, „The in-addr.arpa Domain“, 2010. [Na internetu]. Dostupno na: https://docs.oracle.com/cd/E19683-01/816-4856/dnsintro-84/index.html . [Pristupljeno: 21-kol-2020].

[9] G. Ubnare, „How to configure DNS Name Server in Ubuntu Linux 18.04“, 2018. [Na internetu]. Dostupno na: https://www.youtube.com/watch?v=P1Kf3rDuhJE . [Pristupljeno: 21-kol-2020].

[10] A. Kili, „How To Set Permanent DNS Nameservers in Ubuntu and Debian“, 2020. [Na internetu]. Dostupno na: https://www.tecmint.com/set-permanent-dns-nameservers-in-ubuntu-debian/ . [Pristupljeno: 22-kol-2020].

[11] Educative, „What is port forwarding?“, 2020. [Na internetu]. Dostupno na: https://www.educative.io/edpresso/what-is-port-forwarding . [Pristupljeno: 22-kol-2020].

[12] G. N. Purdy, LINUX iptables Pocket Reference: Firewalls, NAT & Accounting, 1. izd. Sebastopol: O'Reilly Media, 2004.

[13] R. Saive, „How to Setup “Squid Proxy” Server on Ubuntu and Debian“, 2018. [Na internetu]. Dostupno na: https://www.tecmint.com/install-squid-in-ubuntu/ . [Pristupljeno: 30-kol-2020].

[14] Apache, „Apache – HTTP server project“, 2020. [Na internetu], Dostupno na: https://httpd.apache.org/ . [Pristupljeno: 1-ruj-2020].

[15] I. Cohen, „How To Install Apache, MySQL & PHP on an Ubuntu 18.04 VPS or Dedicated Server“, 2018. [Na internetu]. Dostupno na: https://hostadvice.com/how-to/how-to-install-apache-mysql-php-on-an-ubuntu-18-04-vps/ . [Pristupljeno: 1-ruj-2020].

[16] K. Juell, „How To Install the Apache Web Server on Ubuntu 18.04 [Quickstart]“, 2018. [Na internetu]. Dostupno na: https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-18-04-quickstart . [Pristupljeno: 1-ruj-2020].

[17] DevAnswers, „Installing an FTP server (vsftpd) on Ubuntu 18.04 / 19.10“, 2020. [Na internetu]. Dostupno na: https://devanswers.co/installing-ftp-server-vsftpd-ubuntu-18-04/ . [Pristupljeno: 1-ruj-2020].

[18] Efficient iP, „What is DHCP?“, 2020. [Na internetu]. Dostupno na: https://www.efficientip.com/what-is-dhcp-and-why-is-it-important/ . [Pristupljeno: 1-ruj-2020].

[19] NS1, „What is a DNS Zone? DNS Zones Explained“, 2020. [Na internetu]. Dostupno na: https://ns1.com/resources/dns-zones-explained . [Pristupljeno: 1-ruj-2020].

[20] J. Bernhardt, „How to set primary and secondary DNS server addresses with PowerShell“, 2020. [Na internetu], Dostupno na: https://www.jorgebernhardt.com/how-to-set-dns-server-addresses-with-set-dnsclientserveraddress/ . [Pristupljeno: 12-ruj-2020].

Popis slika

Slika 1: Prikaz sučelja Ubuntu Servera (slika autora)..... 4
Slika 2: Prikaz grafičkoga sučelja na Ubuntu serveru (slika autora)..... 5
Slika 3: Webmin (slika autora)..... 6
Slika 4: Prikaz instaliranih virtualnih mašina (slika autora)..... 6
Slika 5: Mrežna konfiguracija 1 – poslužitelj (slika autora)..... 7
Slika 6: Mrežna konfiguracija 2 – poslužitelj (slika autora)..... 8
Slika 7: Konfiguracija mrežnih adaptera (slika autora)..... 9
Slika 8: Topologija mreže (slika autora).....11
Slika 9: Postavke za DHCP (slika autora).....13
Slika 10: Prikaz IP adresa – Ubuntu (slika autora).....14
Slika 11: Prikaz IP adresa – Windows (slika autora).....14
Slika 12: Lease-list naredba (slika autora).....15
Slika 13: Kreiranje zona (slika autora).....16
Slika 14: forward DNS konfiguracija (slika autora).....17
Slika 15: reverse DNS konfiguracija (slika autora).....18
Slika 16: interfaces datoteka – DNS (slika autora).....19
Slika 17: resolv.conf datoteka (slika autora).....19
Slika 18: Konfiguriranje DNS servera na Windows OS-u (slika autora).....20
Slika 19: Rezolucija u internoj mreži (slika autora).....20
Slika 20: rc.local skripta (slika autora).....22
Slika 21: Pinganje udaljenih računala (slika autora).....23
Slika 22: Squid config (slika autora).....24
Slika 23: Proxy: Firefox (slika autora).....25
Slika 24: Windows 10: Proxy (slika autora).....25
Slika 25: Provjera proxy konfiguracije 1 (slika autora).....26
Slika 26: Provjera proxy konfiguracije 2 (slika autora).....26
Slika 27: Web cache – isječak (slika autora).....27
Slika 28: Proxy log – isječak (slika autora).....27
Slika 29: Pristup Apache serveru (slika autora).....31
Slika 30: Testna baza – phpMyAdmin (slika autora).....33
Slika 31: FileZilla podaci (slika autora).....34
Slika 32: Prikaz mrežnog dijeljenja (slika autora).....35
Slika 33 i 34: Prikaz datoteke na Windows računalu (slika autora).....36