# Poboljšanje digitalnog suvereniteta kroz konfiguraciju vlastitog poslužitelja datoteka i korištenja usluga otvorenog koda

**Kernjus, Deni**

Sveučilište u Rijeci, Fakultet informatike i digitalnih tehnologija

Sveučilišni prijediplomski studij Informatika

Deni Kernjus

# Poboljšanje digitalnog suvereniteta kroz konfiguraciju vlastitog poslužitelja datoteka i korištenja usluga otvorenog koda

Završni rad

Mentor: doc. dr. sc. Vedran Miletić

Rijeka, kolovoz 2023.

Sveučilište u Rijeci
FIDIT Fakultet informatike i digitalnih tehnologija
www.inf.uniri.hr

Rijeka, 18. svibnja 2023.

# Zadatak za završni rad

Pristupnik: Deni Kernjus

Naziv završnog rada: Poboljšanje digitalnog suvereniteta kroz konfiguraciju vlastitog poslužitelja datoteka i korištenja usluga otvorenog koda

Naziv završnog rada na engleskom jeziku: Enhancing digital sovereignty through own file server configuration and open-source services utilization

Sadržaj zadatka:

Poslužitelj datoteka je dio gotovo svakog suvremenog poslovnog sustava. Bio taj poslužitelj dio komecijalnih oblaka kao što su Google Drive i Microsoft OneDrive ili dio vlastitog oblaka kao što je Nextcloud, njegova uloga uloga u poslovnom sustavu je ista: spremanje i dijeljenje datoteka koje omogućuju poslovanje. Dakako, taj poslužitelj zahtijeva odgovarajući hardver da bi ispravno obavljao svoju zadaću. Zadatak rada je opisati kompletan postupak postavljanja poslužitelja datoteka, što uključuje hardversku osnovu poslužitelja: svojstva i vrste medija za pohranu podataka koji se koriste (tvrdi diskovi i diskovi čvrstog stanja) te sučelja i kontroleri koje mediji za pohranu podataka koriste, softversku osnova poslužitelja, sustavski dio: hardverske i softverske implementacije RAID-a (Redundant Array of Inexpensive/Independent Disks), specifično Linuxov softverski RAID te razine RAID-a koje se koriste, softversku osnova poslužitelja, aplikacijski dio: prikazati postupak instalacije poslužitelja datoteka (Samba i NFS).

Mentor

Doc. dr. sc. Vedran Miletić

Vedran Miletić

Voditelj za završne radove

Doc. dr. sc. Miran Pobar

Zadatak preuzet: 18. svibnja 2023.

Deni Kernjus

(potpis pristupnika)

Adresa: Radmile Matejčić 2, 51000 Rijeka, Tel: +385(0)51 584 700, E-mail: ured@inf.uniri.hr
OIB: 64218323816
IBAN: HR1524020061400006966
uniri

# Sažetak

Ovaj rad proučava način postizanja digitalnog suvereniteta kroz konfiguraciju vlastitog poslužitelja datoteka. Na temelju pregleda relevantnih istraživanja, dolazimo do zaključka da je želja i potreba za digitalnim suverenitetom, odnosno vlasništvom nad osobnim podatcima sve izraženija u suvremenom društvu. Na međunarodnoj razini, vidimo regulative poput Opće uredbe o zaštiti podataka (engl. *General dana protection regulation*, kraće GDPR), dok na osobnoj razini postoje alati koji omogućavaju i potiču osobni digitalni suverenitet. Kroz ovaj rad prikazani su takvi alati te cjelokupni proces od odabira hardverskih komponenti do postavljanja vlastitog oblaka na poslužitelju. Ovaj osobni oblak omogućava sve funkcionalnosti vlasničkih usluga u oblaku, ali istovremeno omogućava potpuno vlasništvo nad vlastitim podacima i datotekama. Kroz rad koriste se isključivo alati i softveri otvorenog koda.



**Ključne riječi**: poslužitelj datoteka; digitalni suverenitet: GDPR; otvoreni kod; TrueNAS Scale; ZFS; virtualni stroj; Nextcloud; Tailscale; Docker; osobni oblak;

# Sadržaj

# 1. Uvod

U današnjem suvremenom društvu, digitalni suverenitet postaje sve značajnije pitanje kako na međunarodnoj, tako i na individualnoj razini. Na međunarodnoj razini vidimo pojavu regulativa kojima je cilj zaštiti osobne podatke pojedinaca, u tom pogledu ističe se Opća uredba o zaštiti podataka (GDPR) koja je na snazi u Europskoj uniji. Također, kontinuirane kontroverze oko sigurnosti i privatnosti podataka koje prate neke od najvećih tehnoloških korporacija naglašavaju potrebu za očuvanjem digitalnog suvereniteta.

Važnost vlasništva nad vlastitim podacima i datotekama na individualnoj razini prepoznaje sve veći broj ljudi. U vrijeme gdje komercijalne usluge u oblaku često posjeduju korisničke podatke i kontroliraju pristup tim podacima, postoji rastuća potreba za alternativnim rješenjima koja omogućuju digitalni suverenitet.

Ovaj rad istražuje jedan od načina postizanja osobnog digitalnog suvereniteta putem konfiguracije vlastitog poslužitelja datoteka. Konfiguracije poslužitelja datoteka sastoji se od hardverskog i softverskog dijela, no fokus ovog rada usmjeren je na softverski dio, a hardverske mogućnosti predstavljene su u generalnom obliku.

U nastavku rada dan je pregled važnosti Digitalnog suvereniteta kako na međunarodnoj razini tako i na individualnoj razini. Zatim slijedi konfiguracija poslužitelja. Predstavljena je generalna uputa za hardversku konfiguraciju te je zatim prikazana softverska arhitektura i konfiguracija koja će biti objašnjena korak po korak. Softverska konfiguracija sastoji se od postavljanja operacijskog sustava TrueNAS Scale na kojemu se pokreće virtualni stroj. Virtualni stroj služi za pokretanje Docker kontejnera za Nextcloud i Tailscale. Nextcloud omogućava osobni oblak za pohranu datoteka, dok Tailscale pruža siguran pristup oblaku izvan lokalne mreže. Kontejneri unutar virtualnog stroja koristiti će trajnu pohranu putem mrežnog datotečnog sustava (engl. *Network File System*, kraće NFS) koji se dijeli s domaćina, čime osiguravamo cjelovitost i dostupnosti naših podataka.

Kroz ovaj rad, istražujemo proces koji omogućuje pojedincima potpuni nadzor nad vlastitim digitalnim prostorom. Također, rad potiče uporabu alata otvorenog koda koji podržavaju osobni digitalni suverenitet, iz tog razloga korišteni su isključivo takvi alati i softveri. Ovakav pristup omogućava potpunu i besplatnu rekreaciju koraka koji se tiču softverske konfiguracije.

# 2. Digitalni suverenitet

Suverenitet možemo definirati kao pojam koji označava vrhovnu i konačnu političku vlast, iznad koje, u pogledu donošenja i provođenja političkih odluka, nema više vlasti [1]. No, u suvremenoj digitalnoj eri potrebno je proširiti doseg ove definicije kako bi obuhvatili sve bitniji pojam digitalnog suvereniteta. Proteklih godina digitalni suverenitet zauzeo je središte pozornosti, naglašavajući nacionalnu sposobnost da zaštiti svoju digitalnu infrastrukturu, podatke i upravljanje internetom od vanjskih utjecaja. Međutim, važno je istaknuti da digitalni suverenitet ne ostaje ograničen samo na nacionalnu razinu, već proširuje svoj utjecaj i na individualnoj/osobnoj razni, odnosno odnosi se na sposobnost pojedinca da uspostavi kontrolu nad svojim osobnim podatcima, privatnošću i identitetu u digitalnoj sferi. U tom kontekstu, Opća uredba o zaštiti podataka (GDPR) koju je uvela Europska unija predstavlja regulatorni okvir koji naglašava važnost zaštite podataka i prava pojedinca u digitalnom okruženju.

Koncept digitalnog suvereniteta, prema definiciji predstavljenoj od strane Saveznog kancelara Europske unije ima sljedeće značenje: „Digitalni suverenitet opisuje sposobnost oblikovanja digitalne transformacije na samoodređeni način u vezi s hardverom, softverom, uslugama i vještinama. Biti digitalno suveren ne znači primjenjivati protekcionističke mjere ili sve raditi samostalno. Biti digitalno suveren znači, unutar okvira primjenjivog zakona, donositi suverene odluke o područjima na kojima se želi ili je potrebna neovisnost.“ [2]. Prepoznavanje snage podataka dovelo je do nastanka složene eko-sfere entiteta koji prikupljaju, analiziraju i trguju vrijednostima koje se iz njih mogu generirati. Taj proces također nazivamo digitalizacija ili digitalna transformacija. Ova promjena ugrožava privatnost ljudi jer prikuplja osobne informacije nad kojima nemaju vlasništvo i/ili kontrolu. Stoga je nužno imati kontrolu nad vlasništvom, čuvanjem i korištenjem tih podataka te zaštititi digitalnu imovinu i identitet putem odgovarajuće uprave podacima, kontrole kibernetičke sigurnosti i zaštite privatnosti [3].

Analitičko istraživanje proveden u kontekstu rada [3] obuhvatio je analizu publikacija u istraživačkoj bazi podataka ProQuest Central, koji sadrže ključni termin „digital sovereignty“. U prikupljenim radovima često se citira europska Opća uredba o zaštiti podataka (GDPR) koja daje i štiti prava na privatnost pojedinaca koji se nalaze u Europskoj uniji kada su njihovi podaci obrađeni od tvrtke izvan EU. GDPR možemo sagledati kao borbu za digitalni suverenitet na internacionalnoj i nacionalnoj razini.

## 2.1. Nacionalna razina

Na nacionalnoj razini koristi se pojam digitalni suverenitet kako bi se prenijela ideja da bi država trebala afirmirati svoju kontrolu nad internetom te braniti svoje građane i poslovne subjekte kako bi održala kontrolu nad digitalnom infrastrukturom na svojem teritoriju i podacima svojih građana. S obzirom na dominantnu poziciju gigantskih tehnoloških korporacija poznatih kao „GAFAM“ (Google, Apple, Facebook, Amazon i

Microsoft) u sektoru računalstva u oblaku i društvenih medija, podatci građana sada su u osnovi pohranjeni i korišteni u oblaku ovih tehnoloških giganta [3]. Stoga države koje nemaju vezu s navedenim tehnološkim kompanijama žele demonstrirati svoj autoritet i kontrolu nad podacima i njihovim korištenjem [4]. Države trebaju poboljšati svoju podatkovnu infrastrukturu i oblikovanje digitalnih politika koje mogu doprinijeti stvaranju i provođenju digitalnog upravljanja kako bi se postigao digitalni suverenitet na nacionalnoj razini i osigurala kontrola digitalne imovine koju nastoje zaštiti. Također moraju se istovremeno baviti geoekonomskim i geopolitičkim grananjem ako žele pomicati multilateralnu suradnju na globalnoj razini [4]. U slučaju Europske unije, prepoznajući važnost digitalnog suvereniteta i ograničavanja moći platformi, Europska unija usmjerila je svoje napore prema sveobuhvatnoj regulaciji tehnološkog sektora kako bi u potpunosti zaštitila digitalni suverenitet Europe, postavši vodeća „regulatorna supersila“ svijeta [5].

Prethodno navedeno analitičko istraživanje, koje je provođeno u kontekstu [3] također je otkrilo da se uz ključnu termin „digital sovereignty“ često spominju i izrazi „personal“, „control“, „ownership“. Ovaj nalaz ukazuje na potrebu pojedinaca da ostvari suverenitet nad vlastitim podacima. Dakle, koncept digitalnog suvereniteta nije ograničen na nacionalnu kontrolu nad uporabom i dizajnom kritičkih digitalnih sustava i podataka koji se u njima generiraju i pohranjuju, već je primjenjiv i na individualnoj razini, omogućavajući pojedincima da povrate nadzor i kontrolu nad sigurnošću i privatnošću vlastitih podataka [3].

## 2.2. Individualna razina

Umjesto da digitalni suverenitet promatramo kao preduvjet za ostvarivanje ovlasti na određenom teritoriju pod kontrolom vlasti i regulacije vlade, digitalni suverenitet možemo promatrati i kao sposobnost pojedinca da donese odluke i akcije na svjestan, namjeran i neovisan način u vezi s pristupom i rukovanjem vlastitim podacima – samostalni suverenitet nad vlastitim podacima [6]. Stoga, s perspektive građana kao pojedinca, digitalni suverenitet može se primijeniti na individualnoj osnovi s osnovnim razmatranjem kako se postupa s njihovim osobnim podacima i digitalnom imovinom te kao pojedinačni korisnik digitalnih tehnologija i usluga [7]. Digitalna transformacija i korištenje privatnih i osjetljivih podataka dovodi u opasnost privatnost i suverenitet pojedinca. Pošto se takvi podaci često pohranjuju u oblacima koji su uglavnom kontrolirani od strane „GAFAM“ korporacija, postoji značajna zabrinutost oko načina rukovanja podacima koji mogu ugroziti osobnu privatnost pojedinca. Osobni podaci se često prikupljaju i koriste bez znanja ili pristanka pojedinca na kojeg se odnose [6]. U kontekstu umjetne inteligencije i strojnog učenja također se često koriste osobni i osjetljivi podatci, ali kako bi se ublažila zabrinutost, postoje metodologije rudarenja podataka koje čuvaju privatnost (engl*. privacy preservation in dana mining*, kraće PPDM) i štite osjetljive podatke vlasnika podataka [8]. Unatoč metodama i tehnikama za očuvanje i poboljšanje privatnosti kako bi se zaštitila privatnost vlasnika podataka, razina privatnosti nije određena od strane njih, već od strane čuvara podataka, bez potrebe za suglasnošću vlasnika podataka [9]. Vlasnici podataka trebali bi biti u mogućnosti odlučiti koji osobni podaci, koliko i na koji način se njihovi osobni podaci koriste. Također, vlasnicima podatak treba biti omogućeno da slobodno odluče žele li

sudjelovati u dijeljenju podataka. Cilj digitalnog suvereniteta je omogućiti svakom pojedincu, kao pravom vlasniku podataka, slobodu upravljanja i donošenja odluka umjesto da se podaci svakog pojedinca zaključavaju u odvojenim organizacijskim silosima koji su teško dostupni i smanjuju dostupnost i korisnost [3].

# 3. Poslužitelj datoteka

U potrazi za ostvarivanjem digitalnog suvereniteta, konfiguracija vlastitog poslužitelja datoteka može biti korak u pravom smjeru. Posjedovanje vlastitog poslužitelja datoteka te upravljanje njime omogućava korisnicima pohranjivanje i upravljanje vlastitim podacima pod svojim uvjetima. Pojedinci imaju kompletnu kontrolu nad sigurnosti, enkripcijskim algoritmima i pravima pristupa, odnosno mogu biti sigurni u najvišu sigurnost svojih podataka. Uz manje oslanjanja na centralizirane usluge u oblaku, manje je opasnosti od povrede podataka i neovlaštenog pristupa, a manja je i šansa da velike korporacije koriste podatke za vlastitu korist.

Osobni poslužitelj datoteka je poslužitelj smješten u privatnoj rezidenciji koji pruža usluge drugim uređajima unutar ili izvan kućanstva putem kućne mreže ili interneta [10]. Često se koristi za reproduciranje medijskih datoteka za svojem televizoru ili kao centralni uređaj za pohranu za sve vaše uređaje. Stručnjaci u području informacijske tehnologije često koriste osobni poslužitelj kako bi unaprijedili svoje IT vještine. Osobni poslužitelj omogućava testiranje skripti ili programa u sigurnom okruženju koje nije u produkcijskoj uporabi. Također, pruža vam mogućnost implementacije virtualnih strojeva te stvaranje male mreže/domena za testiranje i usvajanje novih vještina [11].

Neki od najčešćih načina uporabe osobnog poslužitelja [11]:

1. Osobni oblak za pohranu
2. Kućni medijski poslužitelj
3. Središnje mjesto za sigurnosne kopije
4. Platforma za kućnu automatizaciju
5. Kućni sigurnosni sustav
6. Upravitelj lozinki
7. Hosting vlastite web stranice
8. Kućni laboratorij (engl. *homelab*)

Korištenje usluga otvorenog koda također daje veliki doprinos u jačanju digitalnog suvereniteta. Softver otvorenog koda daje korisnicima mogućnost pregledavanja, uređivanja i dijeljenja koda kako god smatraju prikladnim zbog njegove transparentne i suradničke prirode. Odabirom alternativa otvorenog koda u zamjenu za vlasničke softvere, moguće je smanjiti ovisnost o zatvorenim ekosustavima i povratiti kontrolu nad svojim digitalnim iskustvom. Također, rješenja otvorenog koda često stavljaju prioritet na sigurnost podataka i privatnost, što je u skladu s načelima digitalnog suvereniteta. Poticanjem korištenja usluga otvorenog koda ljudima se daje prilika da aktivno pridonesu razvoju digitalnih alata koje koriste [12].

Ukratko, osobni digitalni suverenitet je ključan za snalaženje u modernom digitalnom okruženju uz neovisnost, kontrolu i poštovanje privatnosti. Pojedinci mogu vratiti kontrolu nad svojim digitalnim životom jačanjem svoje digitalne suverenosti

postavljanjem osobnog poslužitelja datoteka i korištenjem usluga otvorenog koda. U nastavku rada istražiti ćemo jedan od načina na koji je moguće postaviti osobni poslužitelj podataka te kako koristiti usluge otvorenog koda u svrhu poboljšanja digitalnog suvereniteta. Implementiranjem mjera navedenih u nastavku, pojedinci mogu otvoriti put prema pravednijem i otpornijem digitalnom društvu.

# 4. Hardver

Prilikom konstrukcije osobnog poslužitelja, važno je uzeti u obzir niz ključnih čimbenika kako bi se postigla optimalna korist uz minimalne troškove. Poslužitelji su u većini slučajeva neprestano uključeni 24 sata dnevno, tijekom cijele godine, stoga je nužno pažljivo promišljati o potrošnji električne energije te o selekciji komponenata koje su specifično prilagođene kontinuiranom operativnom okruženju. Također, s obzirom na često kućno smještene osobne poslužitelje, nužno je također u obzir uzeti potencijalnu razinu buke koju generira sustav [11].

Unatoč prijašnjoj preporuci, osobni poslužitelji su često prenamijenjeni kompjuteri i/ili komponente koje su izvan uporabe i često zastarjeli. Uzimajući ovakav kontekst u obzir, u nastavku će biti navedene osnovne smjernice za ključne komponente.

## 4.1. Matična Ploča

Matična ploča igra značajnu ulogu u kontekstu energetske efikasnosti sustava. Matične ploče namijenjene za igranje računalnih igara često su obogaćene dodatnim funkcionalnosti koje pridonose potrošnji energiju te su cjenovno skuplje, a nisu nužne za funkcionalnost poslužitelja. Stoga su osnovni modeli matičnih ploča cjenovno i energetski prihvatljivija opcija [11]. No, svakako je dobro pripaziti na broj SATA utora koje sadržava kako bi mogli povezati veći broj diskova za pohranu.

## 4.2. Procesor

Osobnom poslužitelju načelno nije potrebno puno računalne snage, stoga dvojezgreni ili četverojezgreni procesor je za većinu osobnih poslužitelja dostatan i energetski učinkovit [11].

## 4.3. Memorija

Opće mišljenje je da što je više memorije to bolje, te da manje od 8 GB nije dostatno u većini slučajeva. No, niti previše RAM memorije nije poželjno. Sigurnost podataka na poslužitelju moguće je unaprijediti s memorijom s kodom za ispravak grešaka (engl. *Error Correction Code*, kraće ECC). ECC memorija ima sposobnost automatskog otkrivanja i ispravljanja pogrešaka u memoriji, čime se suzbija korupcija podataka. Takva memorije je poželjna za uporabu s kritičnim podatcima, no zahtijeva matičnu ploču koja podržava takav tip memorije. [13]

## 4.4. Pohrana

Uglavnom se za pohranu podataka u velikim količinama koristi HDD-ove, dok se za pokretanje operacijskog sustava koristi brži SSD ili noviji NVMe SSD (potrebna komatibilna matična ploča). U većini slučajeva diskovi za pohranu unutar poslužitelja biti će povezani u softverski ili hardverski redundantni niz neovisnih diskova (engl. *Redundant Array of Independent Disks*, kraće RAID). RAID povezuje više neovisnih diskova u jednu ili više logičkih jedinica, njegova svrha je povećati pouzdanost u smislu otpornosti na kvar jednog ili više diskova u sustavu te poboljšava performanse [14]. Pošto će se koristiti više diskova koji će raditi 24 sata dnevno kroz cijelu godinu potrebno je razmišljati o posebnim diskovima namijenjenim za tu upotrebu, odnosno o diskovima koji imaju otpornost na vibracije, bolju garanciju te nisu bučni, itd..

## 4.5. Ostale i dodatne komponente

Poslužitelju je također neophodno napajanje. Ovisno o ostalim komponentama te njihovoj potrošnji energije potrebno je odabrati dostatno napajanje. U ovom segmentu također je moguće birati između napajanja specifične namjene za poslužitelje i „običnog“ napajanja. Kao i pohrana, napajanje koje je namijenjeno za poslužitelje ima prednosti u vidu garancije, buke, efikasnosti i sigurnosti te su dizajnirani za neprekidnu uporabu. Pri odabiru napajanja potrebno je uzeti navedena svojstva u obzir [15].

Komponente poslužitelja moguće je smjestiti u gotovo bilo koje kućište, no često je pitanje želi li korisnik smjestiti komponente unutar stalka (engl. *rack*) za poslužitelje ili ne. Stalak za poslužitelje je standard u industriji no za osobni poslužitelj dostatno je i kućište namijenjeno za osobna računala.

U industriji se koristi napredna tehnologija kako bi se osigurala radna temperatura poslužitelja, odnosno cijele poslužiteljske sobe pošto su industrijski poslužitelji značajno veći od osobnih poslužitelja. Neke od tih tehnologija su zračno hlađenje, vodeno hlađenje (krug hlađenja tekućinom ili uranjanje), termalni senzor, pametna kontrola i hlađenje ormarića poslužitelja [16]. Ovisno o namjeni osobnih poslužitelja moguće je također koristiti neku od tih tehnologija, no za osnovnu namjenu dovoljno je zračno hlađenje slično onome koje pronalazimo u osobnim računalima ili ukoliko postoji potreba za nešto tišom varijantom moguće je koristiti i vodeno hlađenje namijenjeno za osobna računala.

Kako bi dodatno poboljšali sigurnost osobnog poslužitelja možemo ga priključiti na besprekidni izvor napajanja (engl. *Uniterruptible Power Supply*, kraće UPS). UPS je uređaj koji pruža rezervno napajanje u slučaju nestanka električne energije ili pada napona na neprihvatljivu razinu. Manji UPS sustavi koji su za osobnu uporabu dovoljni, mogu osigurati napajanje od nekoliko minuta te imaju mogućnost poslati signal serveru da se isključi na uredan način (engl. *graceful shutdown*) [17].

## 4.6. Alternative

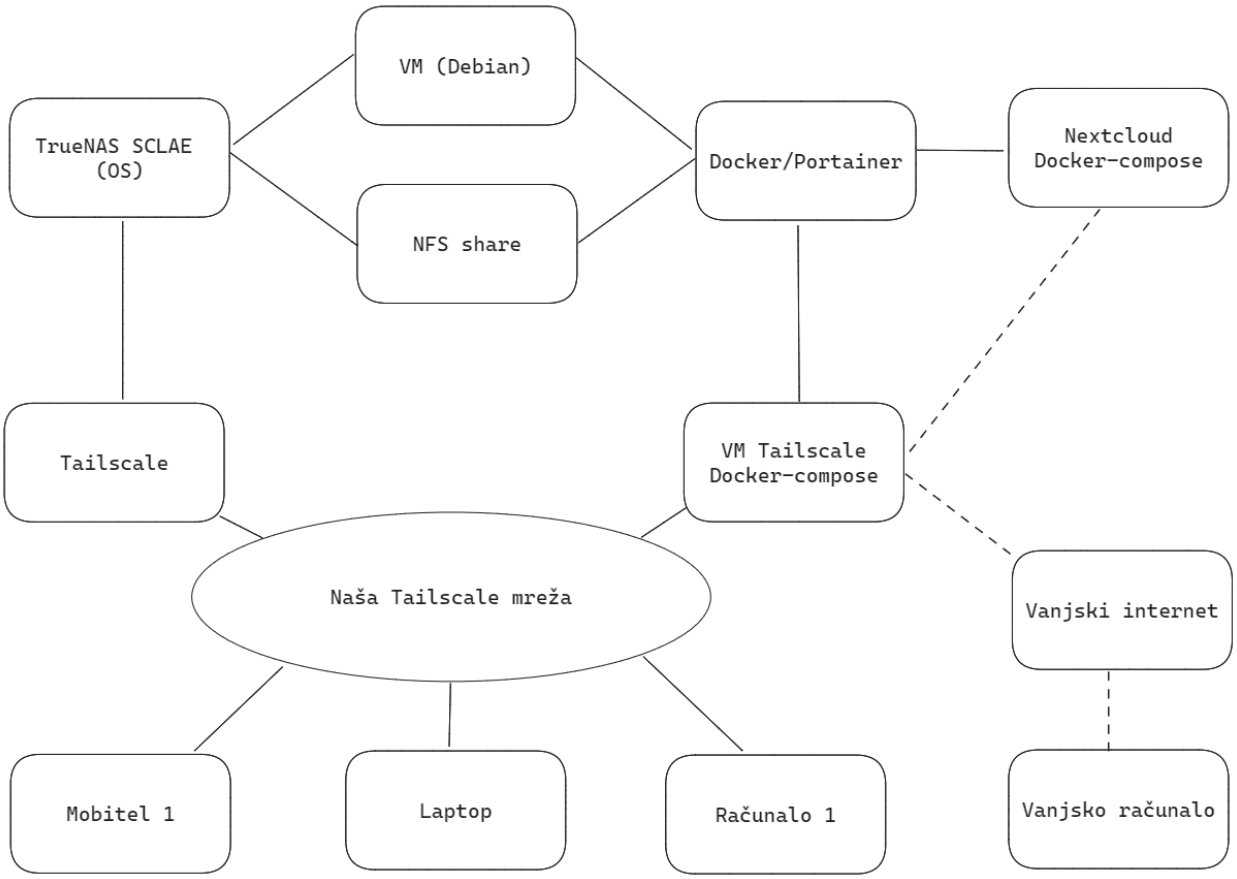
Ukoliko ne želimo konfigurirati i konstruirati vlastiti poslužitelj datoteka, na tržištu postoje već izgrađeni modeli poput Synology NAS i Intel NUC. Također, moguće je koristiti popularni Raspberry Pi za jednostavniju uporabu. No, uvijek postoji kompromis kada su u pitanju gotovi poslužitelji; mogu imati veću potrošnju energije ili ne nude dovoljno prostora za proširenje [11].

# 5. Softver

U nastavku, detaljno će biti razrađen niz koraka i postupaka koji se odnose na konfiguraciju osobnog poslužitelja s ciljem stvaranja funkcionalnog osobnog oblaka, što će omogućiti pohranu i dijeljenje podataka unutar opsega lokalne mreže. Također, istražiti ćemo način kako proširiti ovu sposobnost dijeljenja podataka prema vanjskim mrežama na siguran i jednostavan način. Osim stvaranja oblaka, konfiguracija ovog osobnog poslužitelja otvara niz mogućnosti za daljnje primjene. Korisnici će biti u mogućnosti instalirati i izvoditi raznovrsne aplikacije, čime će proširiti funkcionalnost poslužitelja prema potrebama korisnika. Također, poslužitelj je moguće koristiti kao uređaj za mrežno pohranjivanje pružajući fleksibilnu i skalabilnu infrastrukturu za sigurnu pohranu podataka.

Infrastruktura osobnog poslužitelja strukturirana je oko operacijskog sustava TrueNAS Scale temeljenog na Debian GNU/Linuxu, koji implementira ZFS sustav za pohranu podataka. Unutar ovog okruženja, konfigurirana je virtualan mašina koja će poslužiti kao platforma za pokretanje Docker kontejnera. Administracija Docker kontejnera na virtualnoj mašini izvodi se putem alata Portainer. Kako bi se izbjegla uporaba virtualne memorije virtualne mašine, Docker kontejneri imati će konfiguriranu trajnu pohranu putem NFS share-a.

Specifični Docker kontejneri koji će se pokretati na virtualnoj mašini vezani su za Nextcloud i Tailscale softvere. Nextcloud, kao softverski alat otvorenog koda, omogućava funkcionalnosti zajedničke pohrane datoteka, suradnje i komunikacije, osiguravajući interno posluživanje osobnog oblaka. S druge strane, Tailscale predstavlja vrstu virtualne privatne mreže (engl. *Virtual Private Network,* kraće VPN) rješenja koje omogućava stvaranje virtualne lokalne mreže, spajajući osobne uređaje. Na taj način, pristup podacima postaje moguć i izvan okvira lokalne mreže. Osim toga, Tailscale nudi mogućnost selektivnog izlaganja pojedinih čvorova mreže na javnom webu na siguran način, čime postižemo širu funkcionalnost osobnog oblaka.




*Slika 1: Arhitektura osobnog poslužitelja*

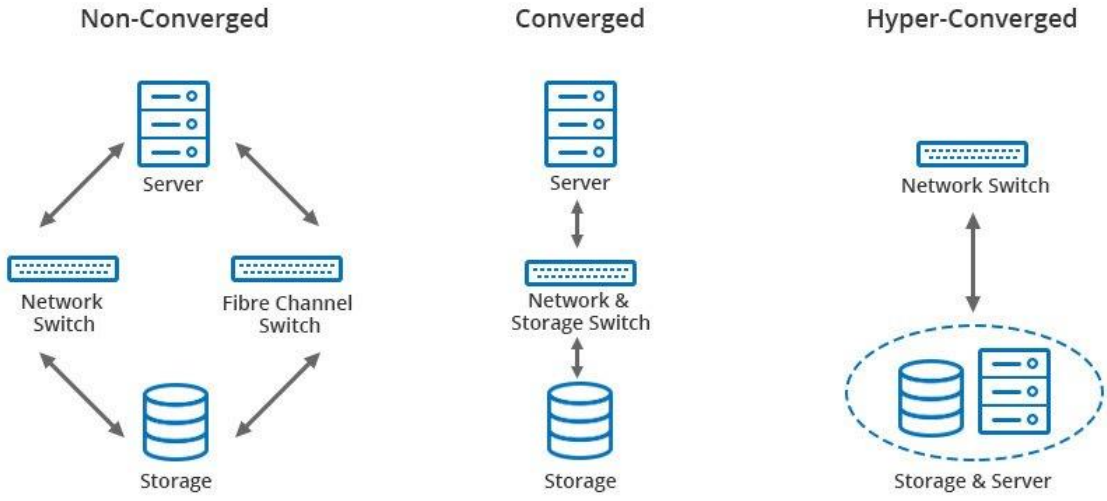
Tailscale će također biti konfiguriran na samom poslužitelju, pružajući mogućnost pristupa poslužitelju i svim njegovim značajkama. Na taj način se ostvaruje potpuna kontrola nad pristupom i uporabom poslužitelja.

# 6. OS (TrueNAS Scale)

TrueNAS predstavlja seriju besplatnih i otvorenog koda operacijskih sustava za umreženo spremište podataka (engl. *Network-Attached Storage*, kreće NAS), razvijenu od strane kompanije iXsystems. Ovi operacijski sustavi temelje se na platformama FreeBSD i Linux te koriste OpenZFS datotečni sustav, čime osiguravaju robusne mogućnosti za pohranu i upravljanje podacima.

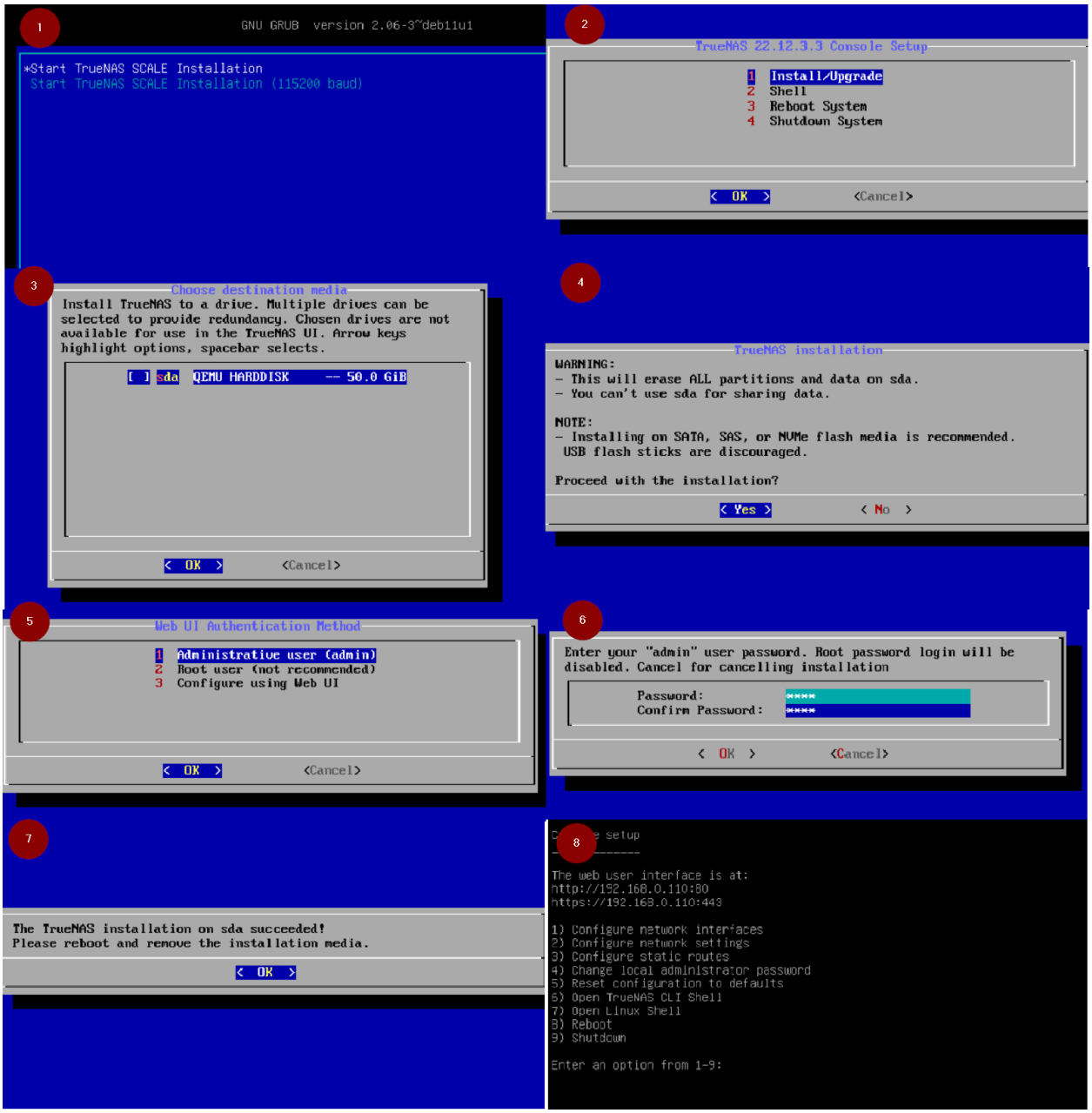
Starija ali stabilnija verzija TrueNAS-a je TrueNAS Core (prethodno poznat kao FreeNAS). Inicijalno stvoren oko 2005. godine, ovaj operacijski sustav temelji se na FreeBSD platformi i koristi OpenZFS datotečni sustav. TrueNAS Core je u osnovi koncipiran kao umreženo spremište podataka (NAS), čime osigurava pouzdano i učinkovito rješenje za pohranu podataka.

iXsystems 2022. godine predstavio je TrueNAS Scale nakon višegodišnjeg razvoja i testiranja. TrueNAS Scale je hiperkonvergirana infrastruktura (engl. *Hyperconverged Infrastructure*, kraće HCI) bazirana na Debian Linuxu te uz snažne sposobnosti za skaliranu pohranu, omogućuje potporu za Linux kontejnere i virtualne strojeve kako bi aplikacije radile bliže podacima [18]. Hiperkonvergirana infrastruktura kombinira pohranu, računalne resurse i mrežu u jedan sustav s ciljem smanjenja kompleksnosti podatkovnih centara i povećanje skalabilnosti [19].




*Slika 2: Usporedba infrastruktura [20]*

## 6.1. Instalacija

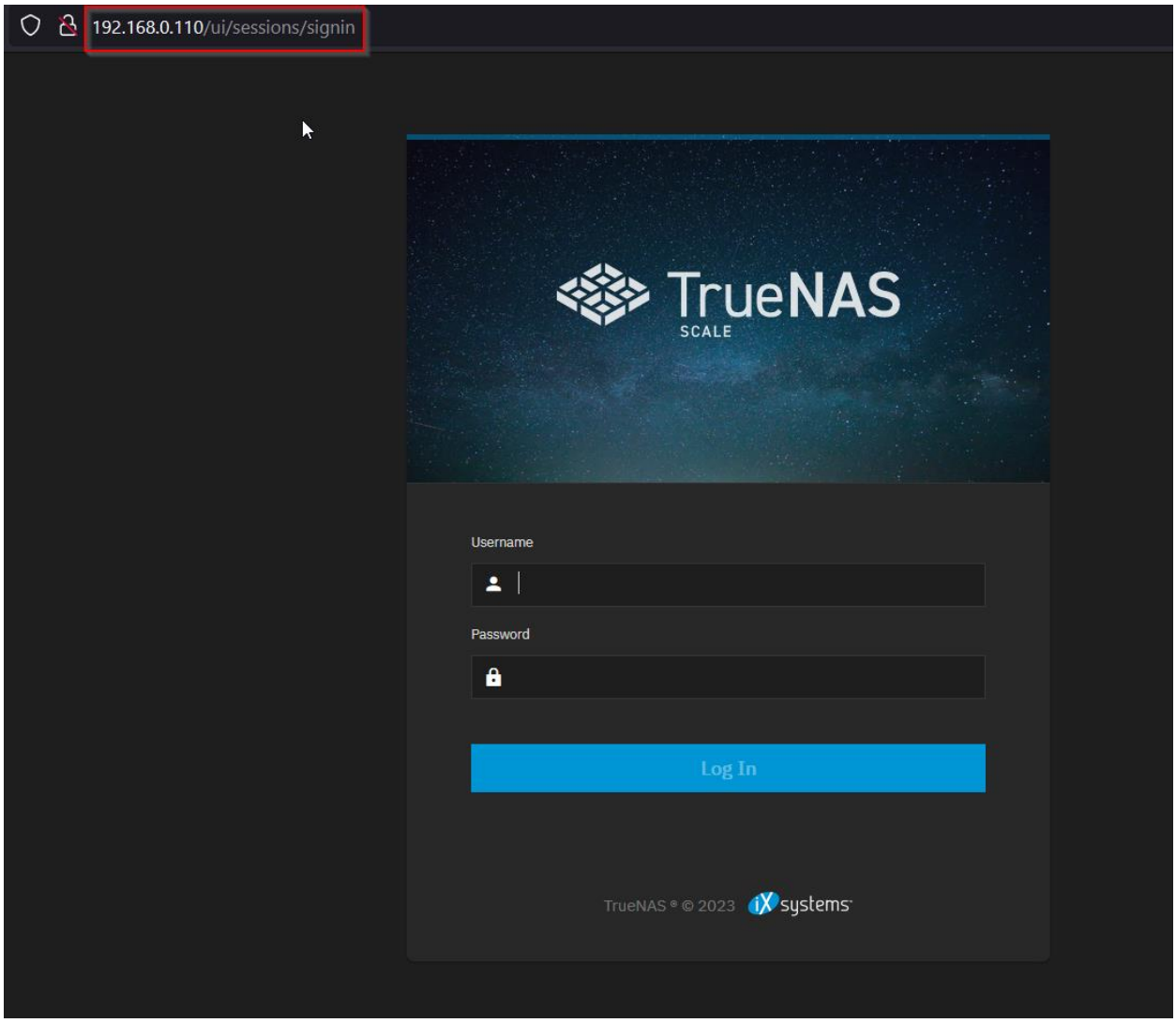



*Slika 3: Instalacija TrueNAS Scale-a*

Za instalaciju TrueNAS Scale prvo je potrebno preuzeti sliku optičkog diska (engl. *optical disc image,* kraće ISO) s TrueNAS web stranice. ISO datoteka, koju često nazivamo ISO slika, je točna kopija cijelog optičkog diska kao što je CD, DVD ili Blu-ray arhivirana u jednu datoteku, odnosno ISO slika je duplikat velikog skupa podataka u manjem formatu. Zbog svoje kompaktnosti često se upotrebljava za prenošenje operacijskih sustava. No, ISO datoteka sama po sebi nije od koristi ukoliko se ne može otvoriti, sastaviti i koristiti [21]. S ciljem omogućavanja upotrebe, ISO datoteka se prenosi na medij poput USB diska,

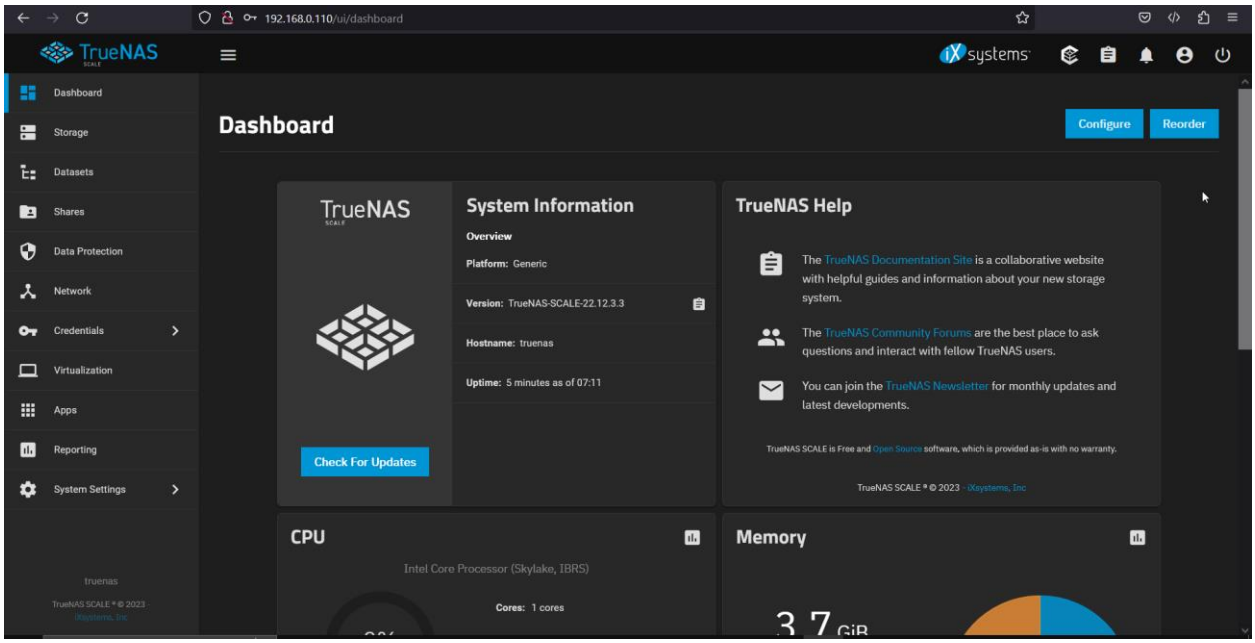
stvarajući tako osnovu za pokretanje operacijskog sustava. U svrhu ovog prijenosa, koriste se jednostavni i besplatni alati poput Rufus-a ili Etcher-a, omogućujući korisnicima uspješno postavljanje ISO datoteke na odabrani medij.

Zatim USB disk možemo koristiti kako bi pokrenuli instalaciju TrueNAS Scale-a na poslužitelju. Priključimo USB disk i zatim pokrenemo poslužitelj. Sada možemo pratiti korake instalacije prikazane na slici 3. U ovom postupku moguće je koristiti isključivo tipkovnicu za odabira opcija. Prvo odaberemo prvu opciju instalacije te pritiskom tipke enter prikazuje se izbornik iz drugog koraka te na njemu odabiremo opciju pod broj 1. Zatim moramo odabrati na koji disk želimo instalirati TrueNAS Scale. Bitno je napomenuti da odabrani disk biti će u potpunosti iskorišten od stane operacijskog sustava te se neće moči koristiti za pohranu. Prema preporuci u poglavlju „Hardver“ to bi trebao biti SSD disk. Odabir vršimo tipkom space. U sljedećem koraku vidimo napomenu koja nam dodatno pojašnjava postupak te nastavljamo dalje s instalacijom. U koraku 5 i 6 postavljamo lozinku admin korisnika. Nakon toga slijedi instalacija koja traje oko desetak minuta te pri završetku dobijemo poruku iz koraka 7 te zatim gasimo poslužitelj i uklonimo USB disk te ponovno pokrenemo poslužitelj. Ovog puta dobivamo prikaz iz koraka 8 u kojem nam je bitna informacija IP adresi na kojoj se nalazi grafičko sučelje TrueNAS Scale-a koji je upravo instaliran. Sada možemo u tražilici preglednika upisati IP adresu te zatim dolazimo na prijavu u naš TrueNAS Scale poslužitelj. Za prijavu je potrebno iskoristiti podatke koji su ranije definirani u postupku instalacije.




*Slika 4: Okvir za prijavu u TrueNAS Scale*

Nakon uspješne prijave, korisniku se pruža pristup korisničkom sučelju TrueNAS Scale-a. U nastavku ćemo, koristeći se ovim sučeljem, definirati konfiguraciju pohrane, NFS share, postaviti virtualni stroj te prikazati način konfiguriranja vlastitog oblaka.




*Slika 5: Nadzorna ploča TrueNAS Scale-a*

# 7. ZFS

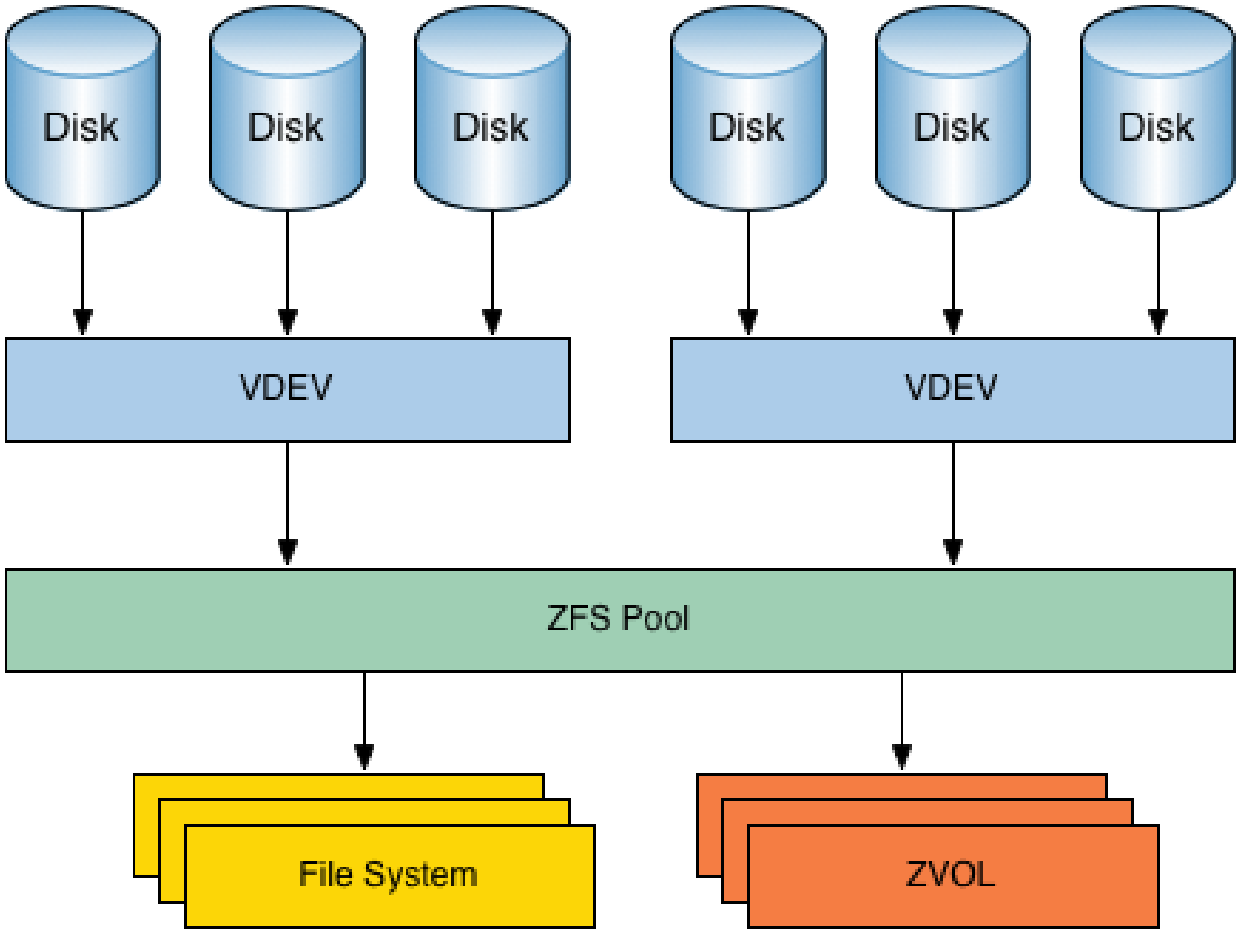
Zettabyte File System, poznat kao ZFS, predstavlja datotečni sustav koji je razvijen od strane kompanije Sun Microsystems u 2006. godini. Glavna svrha razvoja ovog sustava bila je prevladavanje ključnih izazova s kojima se suočavaju konvencionalni datotečni sustavi, a to uključuje osiguranje pouzdanosti integriteta podataka, olakšano administriranje te osiguravanje znatno prostranijih kapaciteta za pohranu [22].

Godine 2001., Matt Ahrens i Jeff Bonwick započinju rad na ZFS projektu u Sun Microsystems s ciljem olakšavanja posla sistemskih administratora koji su trebali upravljati kompleksnim i često pogrešivim sustavima za pohranu podataka. U tu svrhu su zajedno s desetak inženjera razvili temelje ZFS-a kao što su grupirana pohrana (engl. *pooled storage*), pisanje pri kopiranju (engl. *copy-on-write*[1], kraće COW), RAID-Z, snimke (engl. *snapshots*) i slanje/prijem podataka. Jednostavan administrativni model temeljen na hijerarhijskom nasljeđivanju svojstava omogućio je sistemskim administratorima da lako izraze svoju namjeru i učinio je napredne mogućnosti pohrane poput kontrole zbroja (engl. *checksum*), snimki, RAID-a i transparentnih kompresija dostupne i neiskusnim korisnicima [24].

Sun Microsystems je kao dio OpenSolaris projekta objavio izvorni kod ZFS-a kao softver otvorenog koda. No, nakon što je Oracle preuzeo Sun Microsystems 2010. godine kontribucije izvornom kodu su prestale te je nekolicina originalnih ZFS inženjera osnovala OpenZFS kako bi osigurali kontinuirani kolaborativni razvoj verzije otvorenog koda [25].

TrueNAS Scale koristi upravo OpenZFS datotečni sustav. U nastavku će biti pojašnjena njegova arhitektura unutar TrueNAS Scale poslužitelja. Za vizualizaciju potrebno je pratiti sliku 6 u nastavku.

[1] Copy-on-write (COW), koji se ponekad naziva implicitnim dijeljenjem ili sjenčanjem, tehnika je upravljanja resursima koja se koristi u računalnom programiranju za učinkovitu implementaciju operacije "dupliciranja" ili "kopiranja" na resursima koji se mogu mijenjati [23].




*Slika 6: Arhitektura ZFS-a [26]*

Diskovi su uređaji za pohranu, odnosno HDD ili SSD priključeni putem PCIe, SATA ili SAS konektora na matičnu ploču poslužitelja. U ZFS datotečnom sustavu nije preporučeno diskove postaviti u hardverski RAID. Iako će ZFS vjerojatno biti pouzdaniji od drugih datotečnih sustava na hardverskom RAID-u, neće biti pouzdan kao što bi bio bez hardverskog RAID-a [27]. Svi diskovi moraju biti fizički uređaji kako bi TrueNAS mogao provesti S.M.A.R.T. testove[2], provjeravati temperaturu diskova te pokrenuti ih ili ugasiti ih ispravno. Locirani su na dnu arhitekture pohrane te se podatci pohranjuju na njima, ali osim prilikom kreiranja bazena (engl. *pool*) i održavanja, nije potrebno direktno komunicirati s njima [26].

[2] S.M.A.R.T (Self-Monitoring, Analysis and Reporting Technology) je tehnologija samonadzora, analize i izvješćivanja te je industrijski standarda za nadzor i testiranje diska [28].

Sljedeći sloj predstavljaju virtualni uređaju (engl. *virtual devices*, kraće VDEV). To su logički uređaji koji čine bazen pohrane (engl. *storage pool*) i stvoreni su od jednog ili obično više diskova. Postoje različite vrste VDEV-ova za različite potrebe, a to su [26]:

- Data VDEV: koristi se za pohranu podataka koji su pohranjeni u bazenima za pohranu i njihovim skupovima podataka (engl. *datasets*)
- Cache VDEV: koristi se za L2ARC [3]predmemoriju
- Log VDEV: koristi se za ZFS logove
- Hot Spare VDEV: za rezervne diskove koji mogu automatski zamijeniti pokvarene diskove u Data VDEV
- Metadata VDEV: namijenjen za pohranu meta podataka
- Dedup VDEV: namijenjen za pohranu deduplikaciju podataka

U većini osobnih poslužitelja postojati će samo Data VDEV. Ukoliko je broj diskova veći od 10, poželjno je kreirati više Data VDEV-ova. VDEV-ovi mogu se konfigurirati u RAID, odnosno u RAIDZ koji predstavlja ZFS implementaciju RAID-a. Slično kao konvencionalni RAID, RAIDZ je shema podataka/pariteta, osim što koristi dinamičku širinu trake (engl. *stripe*). Bez obzira na veličinu bloka, svaki blok predstavlja svoju vlastitu RAIDZ traku, to implicira da svaki zapis izveden od strane RAIDZ-a predstavlja zapis cijele trake. Ovo u potpunosti eliminira RAID zapisnu rupu[4] kada se kombinira sa semantikom transakcija pisanja pri kopiranju (engl. *copy-on-write*) u okviru ZFS-a. S obzirom na to da nikada ne zahtijeva operacije čitanja-modifikacije-zapisivanja, RAIDZ je također brži od konvencionalnog RAID-a [31]. Moguće RAIDZ opcije su [26]:

- Stripe: blokovi pohrane raspodijeljeni su preko jednog ili više diskova te nije moguće ispraviti greške i pohrana je bez redundancije (nalik RAID0)
- Mirror: Dva diska su zrcaljena te imaju potpuno iste blokove pohrane, odnosno sadrže iste podatke (nalik RAID1)
- RAIDZ1,2,3: podatci o paritetu rasprostranjeni su preko svih diskova te do gubitka podataka dolazi nakon što zakažu jedan, dva ili tri diska (nalik RAID5,6)

ZFS bazen je kombinacija jednog ili više VDEV-a, ali barem jednog Data VDEV-a. Unutar sebe pohranjuje skupove podataka. Skupovi podataka, odnosno datotečni sustav je sloj s kojim korisnik ili program komunicira kako bi pohranio podatke. Skupove podataka možemo usporediti s particijama u konvencionalnim datotečnim sustavima, ali imaju i neke dodatne mogućnosti [26]:

- ZFS enkripcija: skupovi podatak mogu se kriptirati s ključem ili lozinkom

[3] L2ARC je adaptivna zamjenska predmemorija druge razine, koja koristi pogone predmemorije dodane ZFS spremištima [29].

[4] RAID Write Hole (RWH) je scenarij greške povezan s RAID sustavima koji koriste paritet. Pojavljuje se kada se nestanak struje/sudar i kvar diska na traci ili potpuni pad pogona dogode istodobno ili vrlo blizu jedan drugome [30].

- Snimke (engl. *snapshots*): referenciraju stare blokove umjesto kloniranja cijele particije za kreiranje sigurnosne kopije
- Kvote: restrikcija dostupne količine za pohranu unutar skupa podataka za korisnika

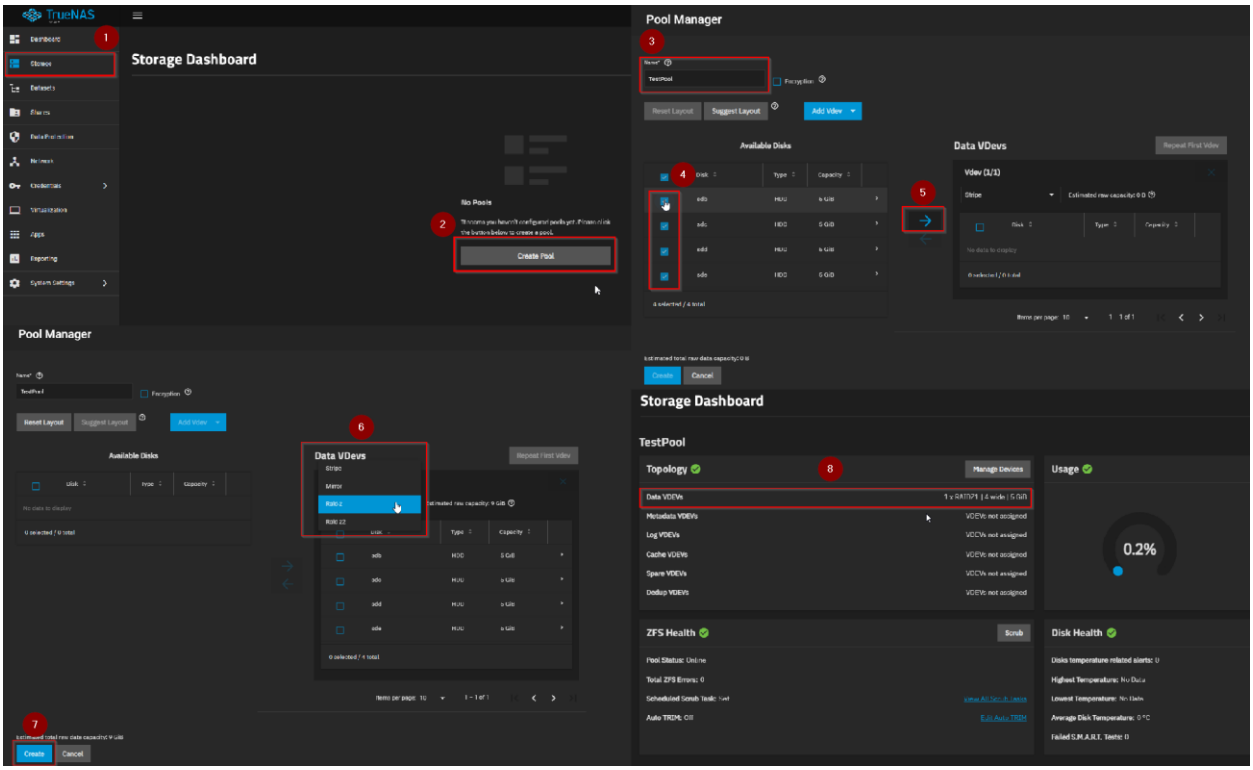
Zvols su blok uređaji unutar ZFS bazena koji se mogu koristiti kao Swap[5] ili kao disk za virtualnu mašinu.

Česta zabluda vezana uz ZFS datotečni sustav je ta da je integritet podataka lošiji u usporedbi s drugim datotečnim sustavima ukoliko se ne koristi ECC memorija. No to je temeljito opovrgnuto [33]. Sva programska oprema zahtjeva ECC memoriju za poboljšanu pouzdanost i integritet podataka te se ZFS ne razlikuje u tom pogledu od drugih datotečnih sustava.

## 7.1. Konfiguracija

U nastavku je prikazana konfiguracija ZFS datotečnog sustava, odnosno konfiguracija jednog bazena i jednog skupa podataka unutar tog bazena na ranije postavljenom TrueNAS Scale poslužitelju. Pratimo sliku 7. Prvo se moramo pozicionirati na „Storage“ karticu (1) te zatim odabrati „Create Pool“ (2) što sugerira da kreiramo bazen.

[5] Swap prostor, poznat i kao virtualna memorija ili stranični prostor, značajka je operativnog sustava koja mu omogućuje privremeno premještanje neaktivnih ili rjeđe korištenih stranica memorije iz RAM-a u određeno područje na tvrdom disku [32].



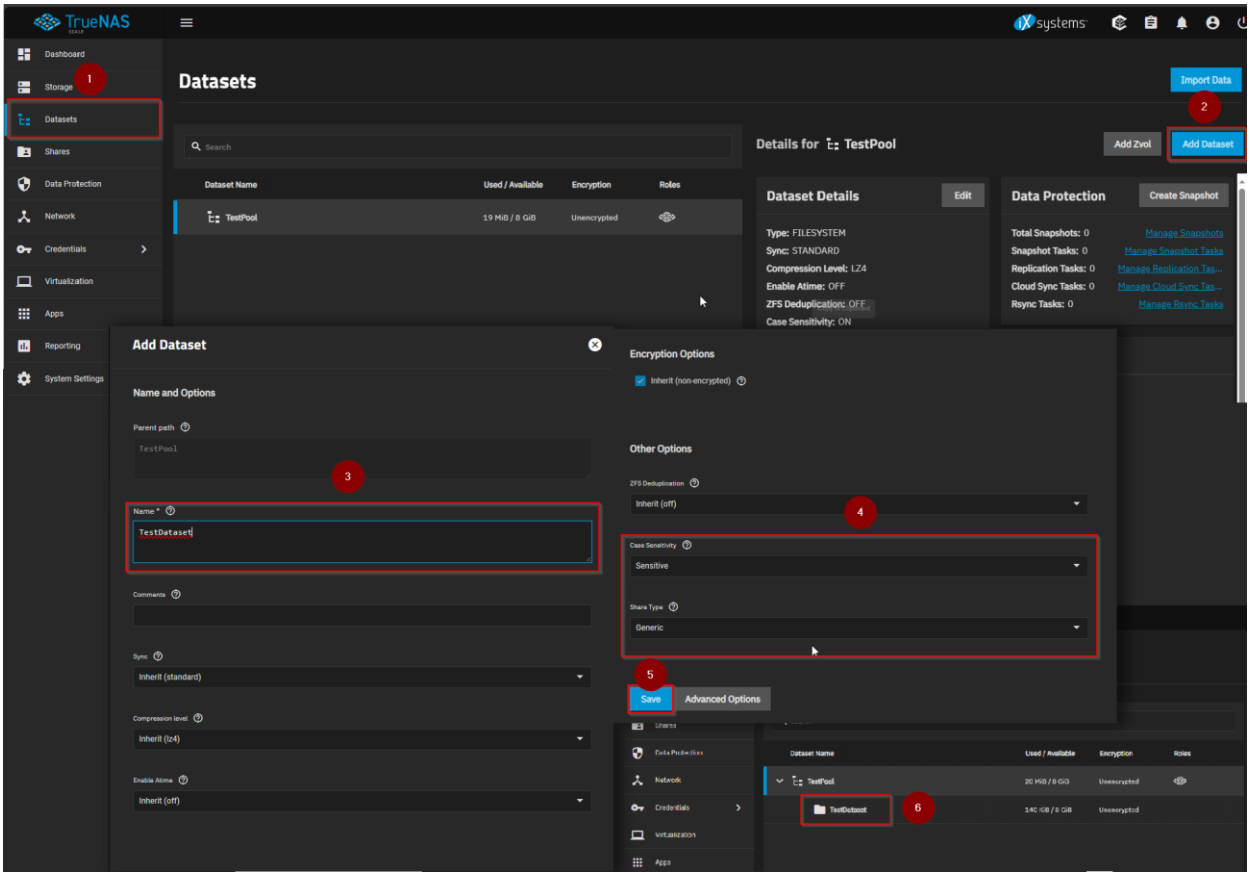

*Slika 7: Konfiguracija ZFS-a*

U Novo otvorenom prozoru prvo dodjeljujemo ime bazenu (3). U okviru „Available Disks“ prikazani su dostupni uređaji za pohranu na poslužitelju. Odabirom kvačice (4) pored pojedinog disk i klikom na strelicu (5) prebacujemo disk u okvir za konfiguraciju RAIDZ-a, odnosno VDEV-a.

Vidimo da ime iznad okvira sugerira da kreiramo VDEV tipa Data. Moramo odabrati jednu od mogućih RAIDZ opcija (6) te zatim klikom na gumb „Create“ (7) kreiramo bazen. Sada na kartici „Storage“ vidimo da je kreiran jedan Data VDEV koji sadrži 4 diska u RAIDZ1 konfiguraciji (8).

Za sada je kreiran Data VDEV i bazen TestPool, no potrebno je kreirati i skup podataka kako bi mogli pohranjivati podatke. Pratimo sliku 8. Potrebno je odbrati karticu „Datasets“ (1) te zatim kliknuti na gumb „Add Dataset“ (2). U novom otvorenom okviru konfigurira se skup podataka. Potrebno mu je dodijeliti ime (3), od ostalih opcija bitne su „Case Sensitivity“ i „Share Type“ (4), a preostale opcije pustimo na zadanim vrijednostima. Ukoliko će datoteke unutar skupa podataka biti pristupane od strane Linux klijenta ona pod „Share Type“ biramo „Generic“, a pod „Case Sensitivity“ biramo „Sensitive“. Ukoliko

planiramo posluživati datoteke za Windows klijente, onda pod „Share Type“ biramo „SMB[6]“, a pod „Case Sensitivity“ biramo „Insensitive“.



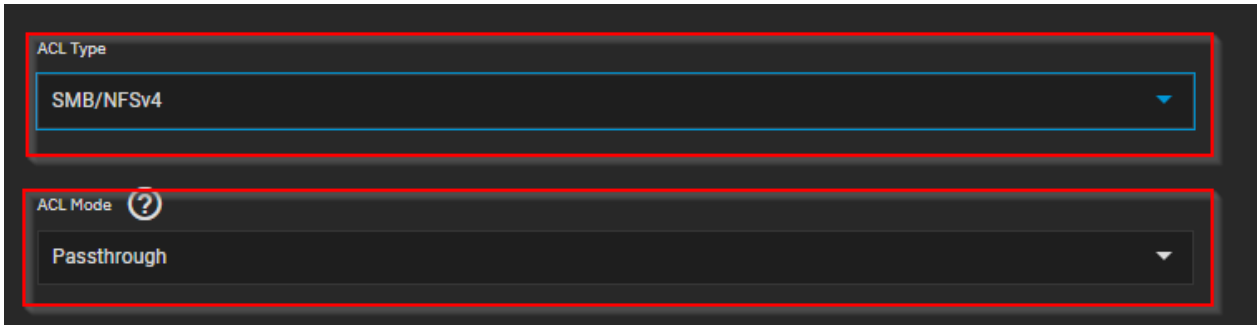

*Slika 8: Stvaranje novog skupa podataka*

## 7.2. NFS share

Na isti način možemo stvoriti skup podataka koji će se koristiti kao NFS share. NFS razvio je Sun Microsystems 1984. godine. Omogućava pregledavanje, ažuriranje i dijeljenje datoteka na udaljenom računalu ili poslužitelju kao da se nalaze na lokalnom računalu. Osmišljen je da bude neovisan o operacijskom sustavu. To postiže tako što je relativno jednostavnog dizajna i ne oslanja se previše na određeni model datotečnog sustava. Izgrađen je povrh ONC protokola udaljene procedure (engl. *Remote Procedure Protocol*, kraće RPC) [35]. Remote Procedure Call (RPC) definira proceduralni model za distribuirane aplikacije i temeljna je arhitektura svih NFS implementacija [36].

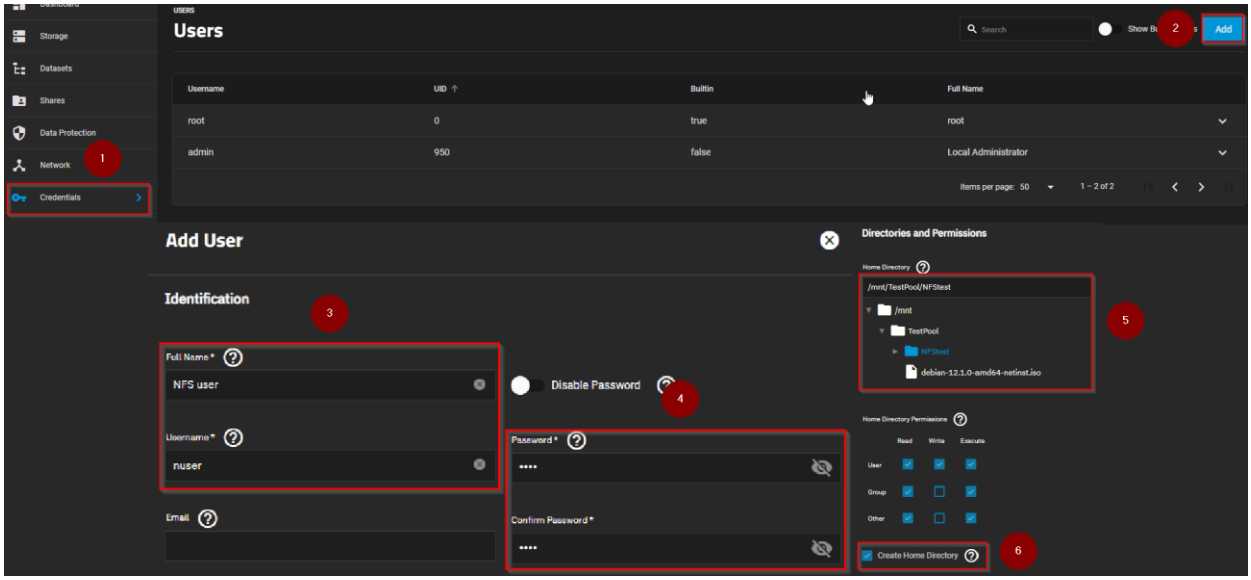
[6] Server Message Block (SMB) omogućuje dijeljenje datoteka, dijeljenje pisača, pregledavanje mreže i međuprocesnu komunikaciju (putem imenovanih cijevi) preko računalne mreže [34].

Pri konfiguraciji skupa podataka koji će biti korišten kao NFS potrebno je pratiti isti postupak kao za prethodni skup podataka te se dodatno pod naprednim opcijama treba postaviti „ACL Type“ na „SMB/NFSv4“ i „ACL Mode“ na „Passthrough“.




*Slika 9: Skup podataka za NFS*

Dobra praksa je kreirati novog korisnika kojemu će biti kućni direktoriji unutar skupa podataka koji će biti korišten kao NFS share te će u njemu imati ovlasti kao korijenski korisnik. Kasnije u virtualnom stroju koristiti ćemo kućni direktoriji kreiranog korisnika kao njegov vlasnik.

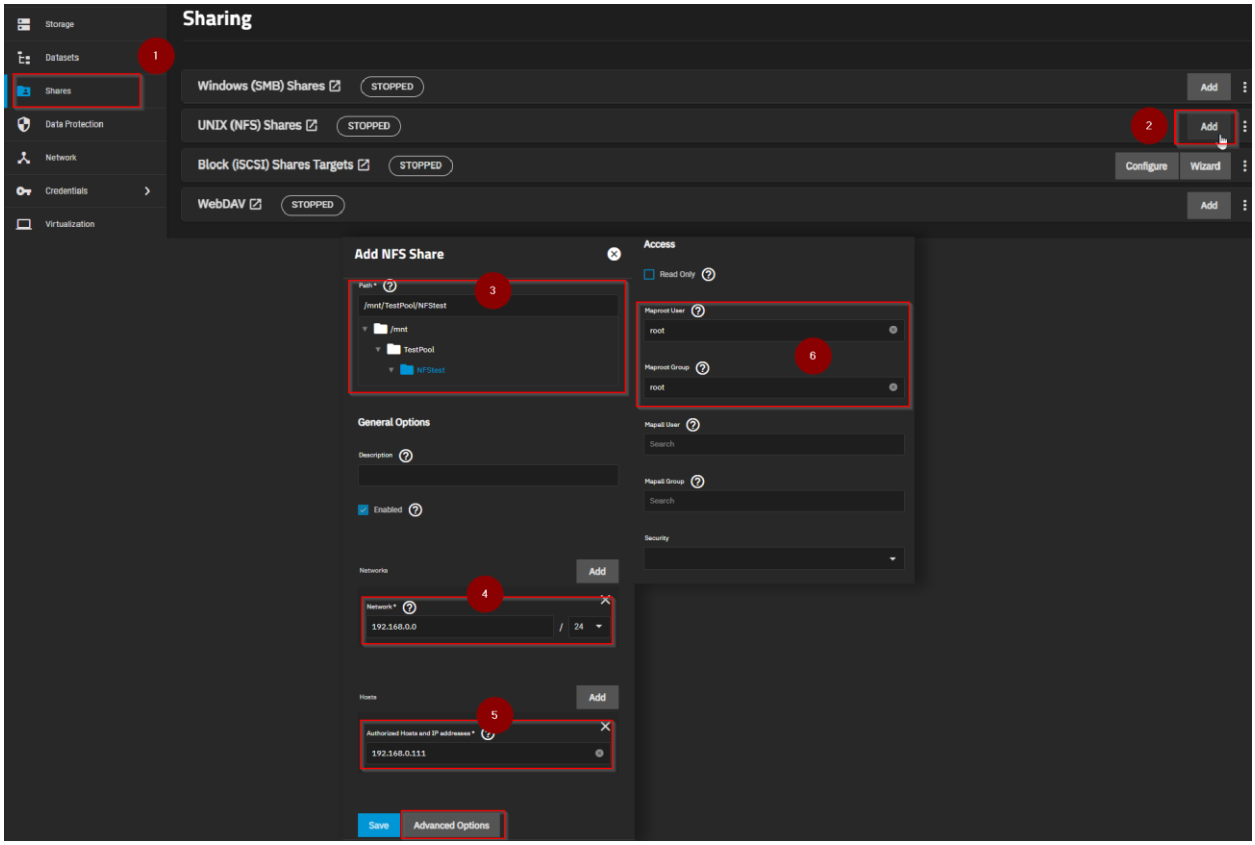



*Slika 10: Stvaranje novog korisnika*

U kartici „Credentials“ (1) možemo kreirati novog korisnika klikom na gumb „Add“ (2). U konfiguracijskom okviru potrebno je dodijeliti naziv novom korisniku (3) te lozinku (4). Pod opcijama za direktorije i dopuštenja biramo kućni direktoriji (5), a to će biti ranije kreirani skup podatka namijenjen za NFS. Također, moramo odabrati opciju „Create Home Directory“ (6) kako bi se kreirao korisnikov direktoriji unutar odabranog skupa podataka. Sada možemo kreirati NFS share.

Preteći sliku 11, otvorimo karticu „Shares“ (1) te pod NFS Shares gumbom „Add“ otvara se konfiguracijski okvir novog NFS Share-a. Prvo je potrebno definirati putanju do

skupa podataka koji želimo dijeliti (2), u ovom slučaju to će biti ranije kreirani skup podataka. Zatim u „Network“ odjeljku (3) definiramo za koju mrežu i podmreže će NFS biti dostupan. Pod „Hosts“ dodajemo IP adresu stroja s kojim se želimo spojiti na NFS, u ovom slučaju to će biti IP adresa virtualnog stroja koji ćemo kreirati u nastavku.




*Slika 11: Stvaranja NFS share-a*

# 8. Virtualni stroj

Virtualni stroj (engl. *virtual machine*, kraće VM) je virtualno okruženje koje djeluje kao virtualni računalni sustav sa svojim vlastitim procesorom, memorijom, mrežnim sučeljem i pohranom. Takvo virtualno okruženje stvoreno je na fizičkom hardverskom sustavu te softver zvan hipervizor odvaja resurse stroja od hardvera i pravilno ih dodjeljuje kako bi ih virtualni stroj mogao koristiti [37].

Hipervizor je softver koji stvara i pokreće virtualne strojeve, ponekad se naziva monitor virtualnog stroja (engl. *virtual machine monitor*, kraće VMM). Njegova svrha je izoliranje operacijskog sustava na kojemu se nalazi i njegovih resursa od virtualnih strojeva te omogućava kreiranje i upravljanje virtualnim strojevima [38].

Fizički stroj na kojem se nalazi hipervizor naziva se stroj domaćin (engl. *host machine*), a virtualni strojevi koji koriste njegove resurse nazivaju se gost strojevi (engl. *guest machines*). Hipervizor tretira resurse domaćina kao što su procesor, memorija i pohrana kao bazen dostupnih resursa koje može realocirati između više virtualnih gost strojeva. Nadzire raspored resursa virtualnih strojeva u odnosu na fizičke resurse te raspodjeljuje resurse koji su dodijeljeni svakom virtualnom stroju. Dok hipervizor upravlja rasporedom, fizički hardver nastavlja izvršavati radnje, pa tako procesor i dalje obavlja instrukcije kako ih zahtijevaju virtualni strojevi [38].

Postoje dva tipa hipervizora koji se mogu koristiti za virtualizaciju a to su hipervizor tip 1 i hipervizor tip 2.

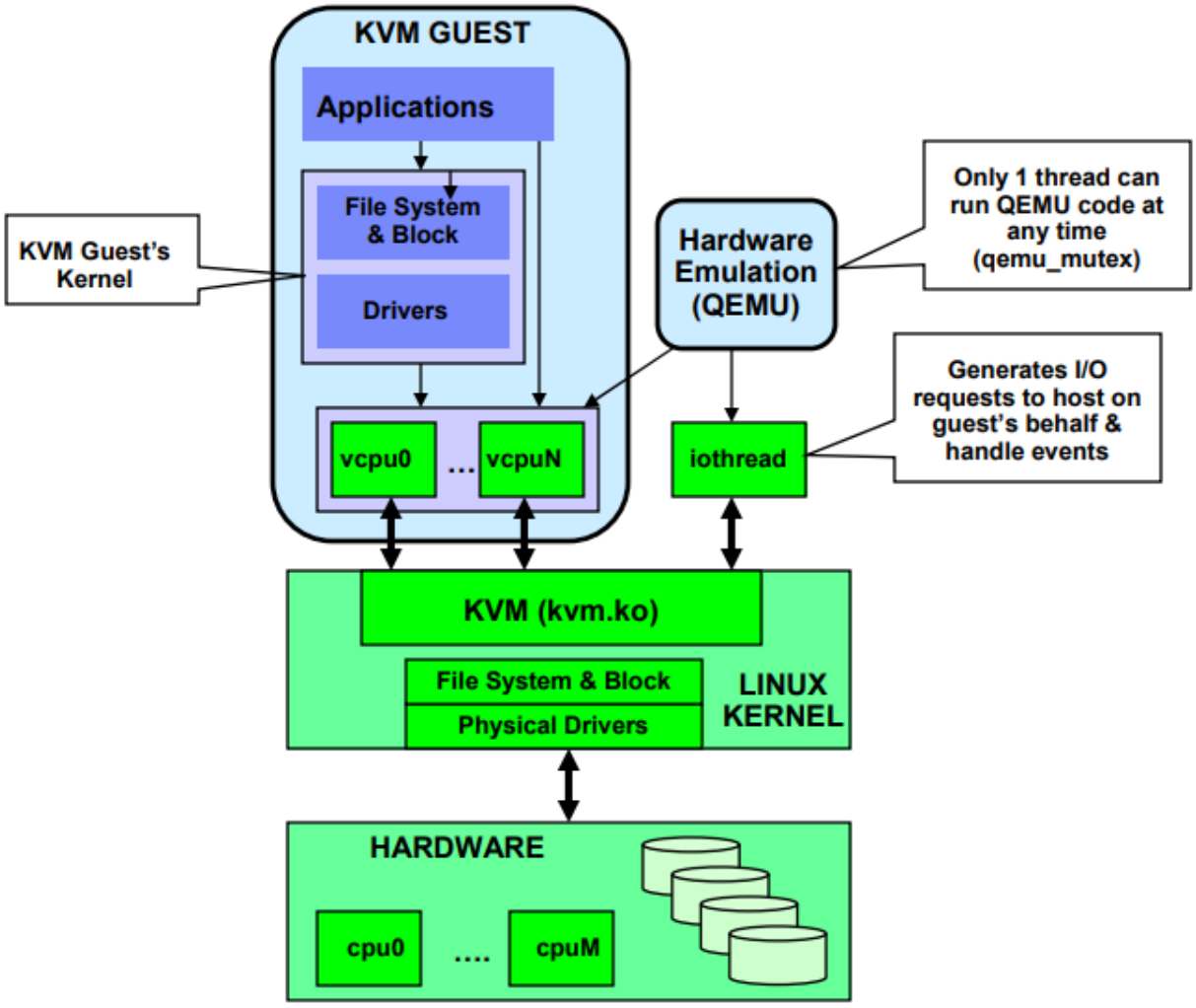
Tip 1 je nativni hipervizor, odnosno pokreće se direktno na hardveru domaćina kako bi upravljao operacijskim sustavima gostiju. Raspoređuje resurse virtualnih strojeva direktno na hardver. To su hipervizori poput KVM i Microsoft Hyper-V [37].

Tip 2 se izvodi na uobičajenom operacijskom sustavu kao sloj softvera ili aplikacija. Djeluje tako da apstrahira gostujuće operacijske sustave od operacijskog sustava domaćina. Resursi virtualnih strojeva raspoređuju se prema domaćem operacijskom sustavu, koji se postom izvodi na hardver-u. To su hipervizori poput Oracle VirtualBox i VMware Workstation [37].

Virtualni strojevi pružaju izolirano okruženje od ostatka sustava koje je pogodno za testiranje novih aplikacija ili za specifičnu upotrebu kao pokretanje vlastitog oblaka.

## 8.1. Virtualni strojevi i TrueNAS Scale

TrueNAS Scale koristi KVM (engl. *Kernel-based Virtual Machine*) hipervizor. To je virtualizacijski softver, odnosno hipervizor tipa 1 otvorenog koda. KVM je dio Linux-a te sve što Linux ima, ima i KVM [39]. Ovaj tip virtualizacije zahtjeva hardver x86 arhitekture, odnosno Intel procesor s virtualizacijskom tehnologijom (engl. virtualization technology, kraće VT) ekstenzijom ili AMD procesor s SVM (AMD-V) ekstenzijom [40].

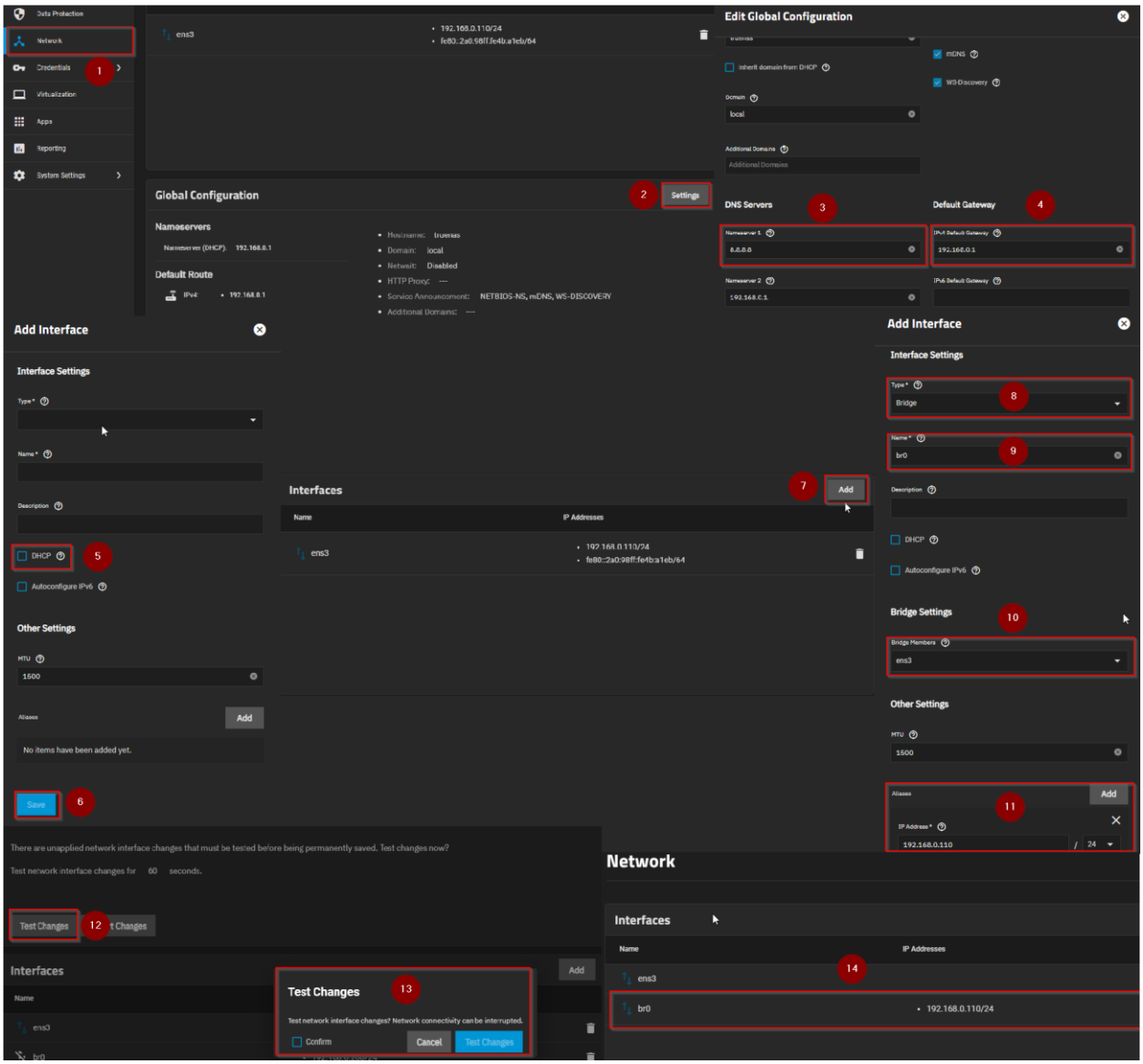



Slika 12: Arhitektura KVM-a [41]

## 8.2. Postavljanje virtualnog stroja

U nastavku biti će postavljen virtualni stroj koji će pokretati Debian GNU/Linux, odabrana je ova distribucija Linux-a pošto je TrueNAS Scale baziran upravo na Debian distribuciji Linux-a te bi se virtualni stroj trebao pokretati bez komplikacija. Virtualni stroj koji ćemo kreirati pokretati će Docker kontejnere za osobni oblak.

Prije postavljanja virtualnog stroja potrebno je kreirati mrežni premosnik (engl. *network bridge*) na domaćinu kako bi virtualni stroj mogao pristupiti skupu podataka, odnosno NFS share-u na domaćinu [42]. Također, u ovom koraku možemo se pobrinuti da poslužitelju postavimo statičnu IP adresu što je svakako preporuka za svaki poslužitelj.



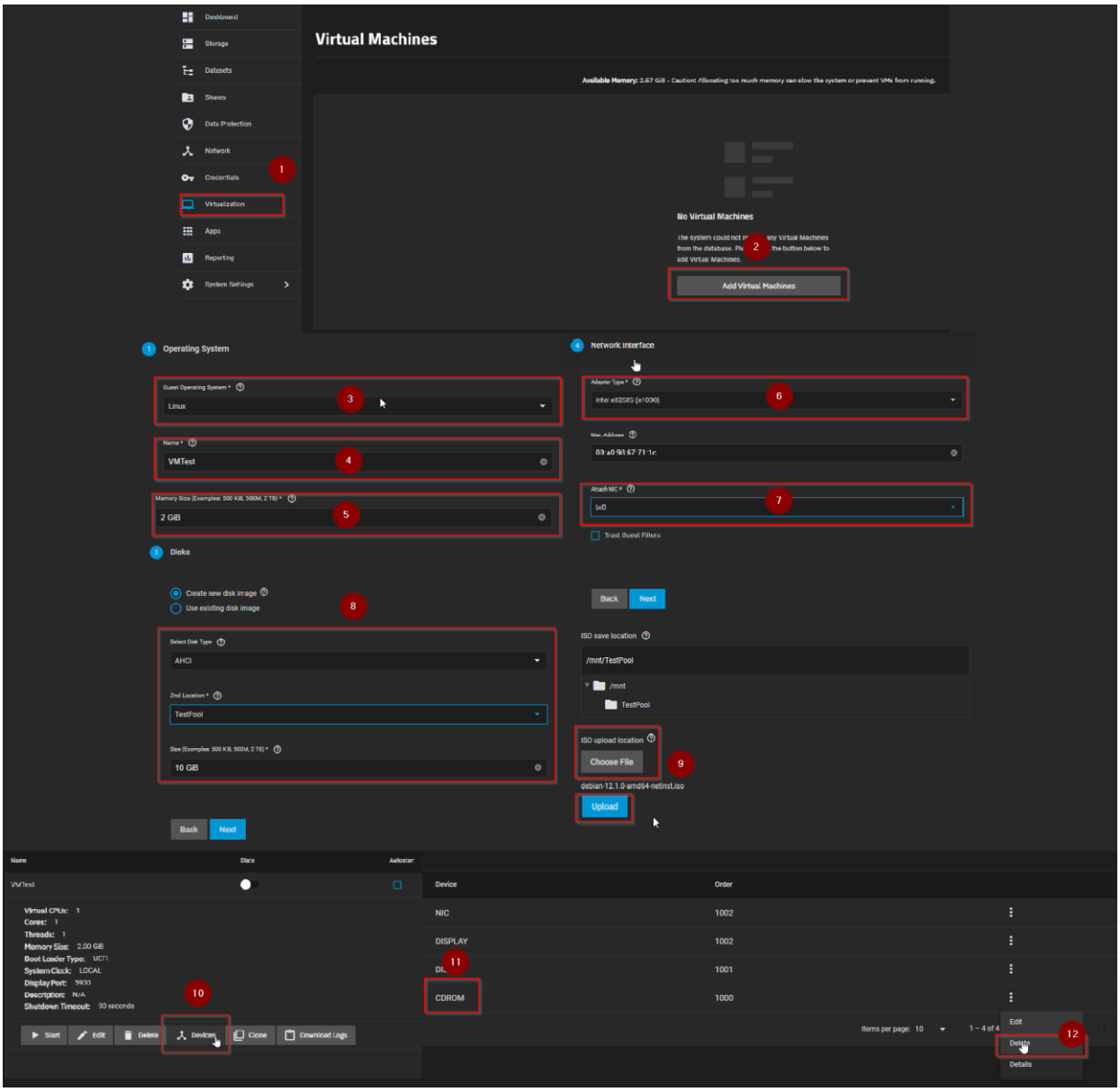

*Slika 13: Stvaranje mrežnog premosnika*

Na kartici „Network“ (1) prvo moramo postaviti globalne konfiguracije (2), konkretno adresu za sustav domenskih imena (engl. *Domain Name Server,* kraće DNS) poslužitelj (3) koji će nam omogućiti da pronađemo naš poslužitelj putem preglednika koristeći domensko ime umjesto IP adrese. U tu svrhu možemo koristiti Google-ov javni DNS poslužitelj koji se nalazi na adresi 8.8.8.8. Također, potrebno je postaviti zadani pristupnik (engl. *default gateway*) (4). Zadani pristupnik je usmjerivač (engl. *router*) koji prosljeđuje pakete prema drugim mrežama kad niti jedna druga specifikacija rute ne odgovara odredišnoj IP adresi paketa. Njegova IP adresa uglavnom odgovara IP adresi usmjerivača.

Odabirom trenutno aktivnog mrežnog sučelja otvara se okvir s opcijama konfiguracije mrežnog sučelja. U ovom okviru potrebno je osigurati da je protokol za dinamičke

konfiguracije glavnog računala (engl. *Dynamic Host Configuration Protocol*, kraće DHCP) za trenutno aktivno mrežno sučelje isključen (5, 6). DHCP dinamički dodjeljuje poslužitelju IP adresu. Sada možemo dodati novo mrežno sučelje (7). U okviru za konfiguraciju odabiremo „Bridge“ vrstu sučelja (8) te nazivamo ga „br0“ (9). Kao člana premošćivanja odabiremo trenutno aktivno mrežno sučelje (10). Pod opcijom „Aliases“ postavljamo IP adresu i masku podmreža (11). Potrebno je odabrati statičnu IP adresu. Statična IP adresa može biti IP adresa koja je izvan DHCP bazena usmjerivača ili možemo unutar usmjerivača konfigurirati IP/MAC vezanje kako bi DHCP dodijelio uvijek istu IP adresu za određenu MAC[7] (engl. *Media Access Control*) adresu. Po završetku konfiguracije mrežnog premosnika nudi nam se mogućnost testiranja mrežnog sučelja (12, 13) te ukoliko testiranje uspješno prođe završili smo postavljanje mrežnog premosnika (14). Sada možemo postaviti virtualni stroj.

[7] MAC adresa jedinstveni je identifikator dodijeljen kontroleru mrežnog sučelja (engl. network interface controller, kraće NIC) za korištenje kao mrežna adresa u komunikaciji unutar mrežnog segmenta [43].




*Slika 14: Stvaranje virtualnog stroja*

U kartici „Virtualization“ (1) potrebno je odabrati opciju za stvaranje virtualnog stroja (2). Okvir za konfiguraciju virtualnog stroja ima popriličan broj opcija no, u ovom slučaju nije potrebno iskoristiti sve opcije, zato su prikazane samo opcije koje su izmijenjene. Tako je odabran operacijski sustav virtualnog stroja (3) i naziv (4), zatim odabiremo koliko memorije želimo dodijeliti stroju (5). Kod odabira mrežnog sučelja bitno je odabrati ranije kreirani mrežni premosnik (7), zatim biramo vrstu i količinu pohrane (8) te učitavamo ISO sliku operacijskog sustava (9). Sada možemo pokrenuti virtualni stroj te proći kroz instalaciju operacijskog sustava, u ovom dijelu bitno je odabrati „SSH server“ i „standard

system utilities“ prilikom odabira softvera. Nakon što je instalacija završila, moramo isključiti virtualni stroj te pod opcijom „Devices“ (10) uklanjamo CDROM (11, 12). Sada možemo pokrenuti virtualni stroj. No, vidimo da se pokreće UEFI[8] ljuska. Kako bi to izbjegli potrebno je konfigurirati startup.nsh datoteku. To učinimo sljedećim naredbama koje upisujemo u ljusku:

1. fs0: (ulaz u datotečni sustav)
2. edit startup.nsh (mogućnost uređivanja i stvaranje datoteke startup.nsh)
3. U startup.nsh unosimo \EFI\debian\grubx64.efi, ovo je putanja pokretača (engl. *bootloader path*). Pokretač je zaslužen za pokretanje operacijskog sustava
4. Kako bi pohranili sadržaj startup.nsh datoteke koristimo sekvencu kombinacija tipki na tipkovnici: ctrl + s (spremanje) -> enter (potvrđivanje lokacije spremanja) -> ctrl + q (izlaz)

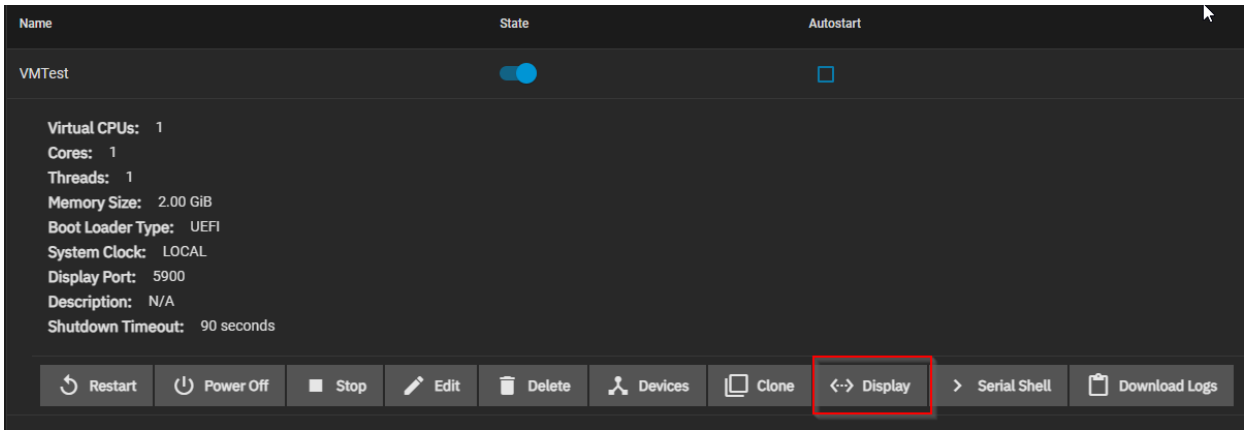
## 8.3. Pridruživanje NFS-a

Unutar virtualnog stroja, potrebno je pridružiti NFS share koji je prethodno uspostavljen na domaćinu. Docker kontejneri koji će biti stvoreni u narednom koraku koristit će NFS kao svoj trajni sustav za pohranu. Ovaj korak posebno je značajan za Docker kontejner Nextcloud koji će služiti kao osobni oblak za pohranu, omogućavajući pohranu svih datoteka koje se prenose u oblak na NFS zajedničkom resursu. Ovaj pristup nam omogućava izbjegavanje pohrane na relativno osjetljivom disku virtualnog stroja, čime se umanjuje potencijalni rizik od gubitka podataka ili degradacije performansi [45].

Kako bi to napravili potrebno se povezati s virtualnim strojem putem protokola sigurne ljuske (engl. *Secure Shell Protocol*, kraće SSH) ili pomoću gumba „Display“ (slika 15). Secure Shell protokol je protokol za siguran udaljeni pristup i druge sigurne mrežne usluge putem nesigurne mreže [46]. Kako bi se putem SSH-a spojili na virtualni stroj moramo unutar naredbenog retka upisati (engl. *Command-line interface*, kraće CLI) naredbu:

```
$ ssh {naziv korisnika}@{ip adresa virtualnog stroja}
```

[8] Unified Extensible Firmware Interface (UEFI, nasljednik EFI-ja) je sučelje između operativnih sustava i firmware-a [44].




*Slika 15: Prikaz zaslona virtualnog stroja*

Kada samo uspješno pristupili virtualnom stroju možemo izvršit naredbe za povezivanje s NFS-om. Prvo je potrebno instalirati paket koji pruža alate za pridruživanje NFS-a. U Debian distribuciji Linuxa naredba je:

```
$ apt install nfs-common
```

Nakon uspješne instalacije paketa možemo pridružiti NFS skup podataka na domaćinu nekom od direktorija na virtualnom stroju. U tu svrhu prvo je potrebno kreirati novi direktoriji, nazvati ćemo ga „nfs“:

```
$ mkdir /nfs
```

Sada možemo pridružiti NFS skup podataka na domaćinu novo kreiranom direktoriju na virtualnom stroju:

```
$ mount 192.168.0.110:/mnt/TestPool/NFStest/nuser /nfs
```

Uočimo kako smo pridružili kućni direktoriji od ranije kreiranog korisnika.

Kako ne bi ovaj postupak ponavljali pri svakom pokretanju virtualnog stroja možemo definirati automatsko pridruživanje NFS-a. To radimo unutar /etc/fstab konfiguracijske datoteke. U navedenoj datoteci nalazi se definicija načina pridruživanja datotečnih sustava prilikom pokretanja operacijskog sustava [47]. Na kraj datoteke moramo upisati definiciju u obliku:

```
# <file system>        <dir>         <type>    <options>             <dump>
<pass>
192.168.0.110:/mnt/TestPool/NFStest/nuser /nfs   nfs  x-
systemd.automount,rw,async,noatime,hard  0  0
```

Više mogućih opcija (engl. *options*) pridruživanja i njihovih značenja moguće je pronaći u službenoj Debian dokumentaciji [47].

Kako bi promjene u /etc/fstab konfiguracijskoj datoteci imale učinak potrebno je ponovno pokrenuti systemd[9]. U tu svrhu možemo koristiti naredbu:

```
$ systemctl daemon-reload
```

Kako prilikom pokretanja virtualnog stroja ili prilikom gubitka internetske veze može doći do pogreške pri pridruživanju NFS-a zbog zakašnjele inicijalizacije internetske veze ili potpunog nedostatka iste, možemo koristiti naredbu koja odgađa aktivaciju servisa koji ovise o internetskoj vezi dok internetska veza nije uspostavljena:

```
$ systemctl enable systemd-networkd-wait-online.service
```

## 8.4. Postavljanje statične IP adrese

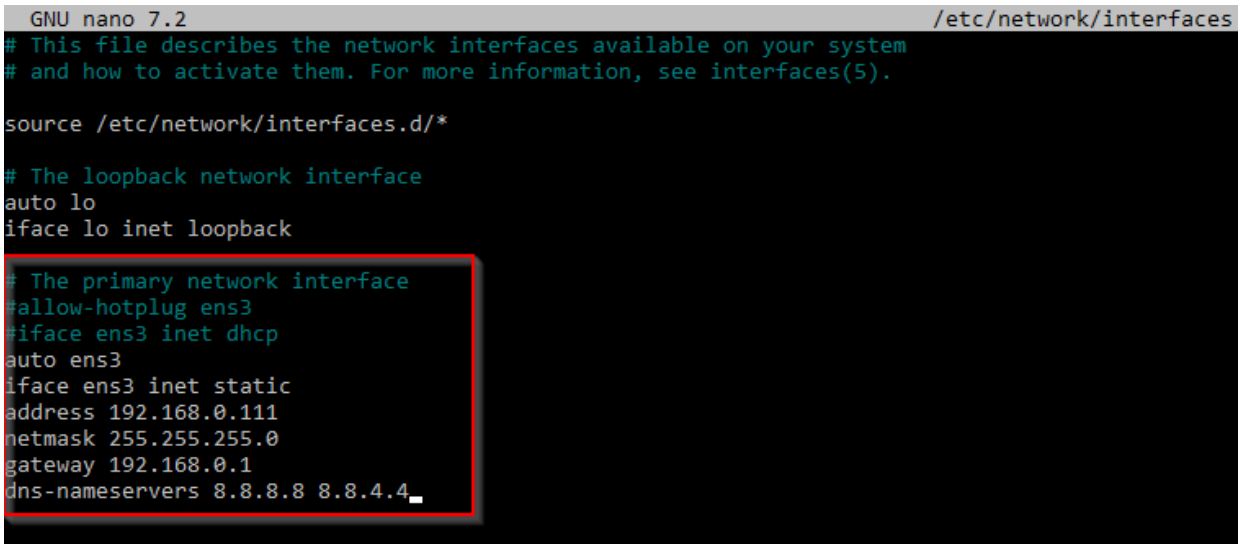
Dobra praksa je postaviti virtualnom stroju statičnu IP adresu kao što smo i poslužitelju domaćinu postavili. Kako bi to napravili potrebno je navigirati do datoteke /etc/network/interfaces:

```
$ nano /etc/network/interfaces
```

Unutar datoteke, pod primarnim mrežnim sučeljem, potrebno je za zakomentirati linije koje se odnose na DHCP pomoću znaka #. Zatim moramo dodati linije kao što je prikazano u slici 16. Dodane linije automatski postavljaju primarno mrežno sučelje (ens3) tako da mu konfigurirana statička dodjela IP adrese, te je ona 192.168.0.11 i ima masku 255.255.255.0. Postavke zadanog pristupnika (engl. *default gateway*) i DNS-a možemo postaviti na iste vrijednosti kao na domaćinu. Kako bi promjene imale utjecaj potrebno je ponovno pokrenuti systemd servis systemd-networkd [49].

```
$ systemctl restart systemd-networkd.services
```

[9] systemd je skup osnovnih građevnih blokova za Linux sustav. Omogućuje upravitelja sustava i usluga koji radi kao PID 1 i pokreće ostatak sustava [48].



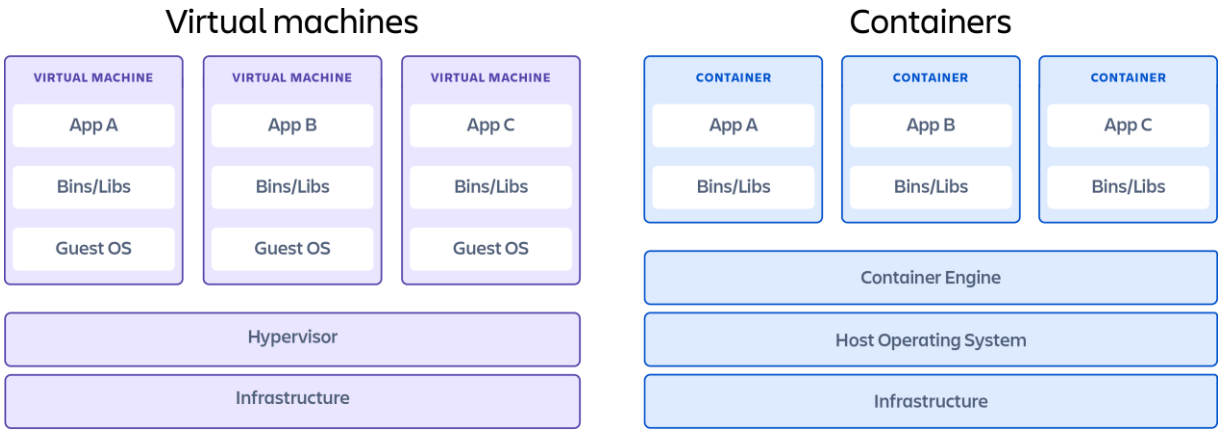

*Slika 16: Sadržaj interfaces datoteke*

# 9. Docker

Docker je platforma otvorenog koda koja pokreće aplikacije i olakšava proces razvoja i distribucije istih. Aplikacije stvorene pomoću Dockera zapakirane su zajedno sa svim svojim pomoćnim ovisnostima u zajedničku strukturu poznatu kao kontejner. Kontejneri se pokreću u izoliranoj virtualnoj okolini povrh jezgre operacijskog sustava [50]. Docker kontejner možemo usporediti s vrlo laganim virtualnim strojem.

Kreacija kontejnera se provodi na temelju Docker slike. Ova slika predstavlja paket softverske aplikacije, zajedno s potrebnim okruženjem i ovisnostima, što omogućava konzistentno i ponovljivo izvršavanje aplikacije neovisno o okruženju na kojem se izvodi. Za izradu Docker slike, koristi se Dockerfile. Dockerfile je tekstualna datoteka koja sadrži niz instrukcija za definiranje izgradnje slike. Ove instrukcije uključuju specificiranje osnovne slike koja služi kao temelj, te niz dodatnih naredbi za konfiguraciju, instalaciju potrebnih paketa i konstrukciju okruženja. Osnovne slike najčešće su slike operacijskih sustava. Takve, ali i ostale slike dostupne su u privatnim ili javnim repozitorijima kao što je Docker Hub.

Možemo zaključiti da je koncept kontejnera vrlo sličan konceptu virtualnog stroja. Glavna razlika tiče se dosega virtualizacije. Dok virtualni stroj obuhvaća cijelu virtualizaciju od softverskog do hardverskog sloja, kontejner se ograničava na softversku komponentu iznad operacijskog sloja. Ovakav pristup donosi prednosti u pogledu brzine, portabilnosti, skalabilnosti, brzini isporuke aplikacija i gustoći, odnosno mogućnost većeg broja kontejnera na jednom domaćinu u odnosu na virtualne strojeve [50].




*Slika 17: Usporedba virtualnog stroja i kontejnera [51]*

S obzirom da je TrueNAS Scale baziran na Linux-u, moguće je instalirati Docker direktno na domaćina. No, trenutno je ova mogućnost ograničena te postoji mogućnost da se stvoreni kontejneri unište prilikom ažuriranja sustava [45]. Iz tog je razloga ranije

kreiran virtualni stroj kako bi imali stabilnu okolinu nad kojom imamo potpunu kontrolu te će se kontejneri pokretati unutar njega. Kako bi to bilo moguće potrebno je postaviti Docker repozitoriji te zatim instalirati Docker Engine, to radimo na sljedeći način [52]:

1. Kako bi izbijeli potrebu za korištenjem naredbe „sudo“ možemo se prebaciti na korijenskog korisnika koji ima sva prava:

   ```
   $ su
   ```

2. Kako bi instalirali najnovije pakete potrebno je ažurirati repozitoriji paketa te također omogućiti da naredba „apt“ koristi repozitoriji putem HTTPS-a:

   ```
   $ apt-get update
   $ apt-get install ca-certificates curl gnupg
   ```

3. Potrebno je dodati službeni Docker GPG ključ:

   ```
   $ install -m 0755 -d /etc/apt/keyrings
   $ curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --
   dearmor -o /etc/apt/keyrings/docker.gpg
   $ chmod a+r /etc/apt/keyrings/docker.gpg
   ```

4. Sada možemo postaviti repozitoriji:

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

5. Ponovno je potrebno ažurirati apt repozitorij:

   ```
   $ apt-get update
   ```

6. Kad je repozitoriji postavljen možemo instalirati potrebne pakete, a to su Docker Engine, containerd i Docker Compose

   ```
   $ apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-
   plugin docker-compose-plugin
   ```

7. Kako bi potvrdili uspješnu instalaciju možemo pokrenuti naredbu:

   ```
   $ docker run hello-world
   ...
   Hello from Docker!

   This message shows that your installation appears to be working
   correctly
   ...
   ```

   Ako je instalacija bila uspješna dobiti ćemo potvrdnu poruku.

# 10. Portainer

Kako bi stvorili, ažurirali i upravljali kontejnerima potrebno je koristiti naredbeni redak, no takav način može uzrokovati kompleksnost. Rješenje je koristiti softver poput Portainer-a. Portainer je softver otvorenog koda koji skriva kompleksnost upravljanja kontejnerima iza korisničkog sučelja jednostavnog za uporabu. Eliminira potrebu za korištenjem naredbenog retka, pisanja YAML datoteka ili razumijevanje manifesta. [53]

Portainer se sastoji od dva element, Portainer poslužitelj i Portainer agent. Oba elementa pokreću se kao Docker kontejneri [54]. Prilikom inicijalnog postavljanja Portainer-a, odnosno gotovo bilo kojeg Docker kontejnera potrebno je definirati mjesto trajne pohrane. U tu svrhu koristiti ćemo ranije povezan NFS. Prvo kreirajmo direktoriji unutar direktorija povezanog s NFS-om:

```
$ mkdir /nfs/portainer_data
```

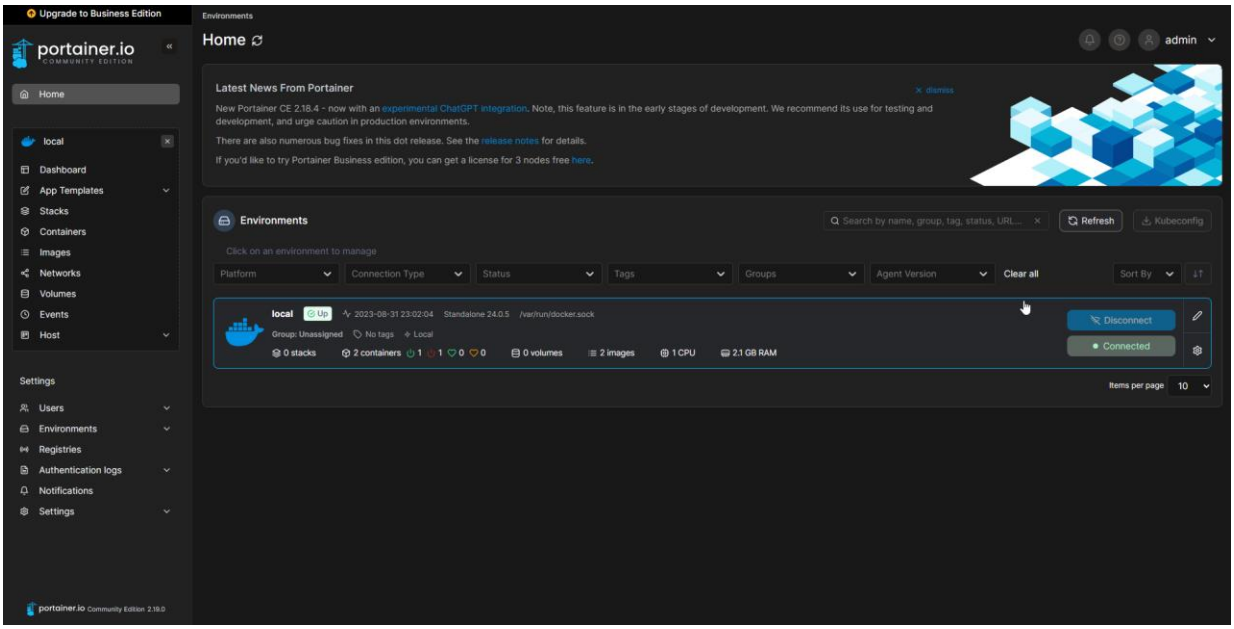
Sada možemo preuzeti, instalirati i pokrenuti Portainer poslužitelj unutar kontejnera:

```
$ docker run -d -p 8000:8000 -p 9443:9443 --name portainer --
restart=always -v /var/run/docker.sock:/var/run/docker.sock -v
/nfs/portainer_data:/data portainer/portainer-ce:latest
```

S naredbom:

```
$ docker ps
```

Možemo provjeriti ako je pokrenut Portainer kontejner. Ukoliko je kontejner uspješno kreiran možemo mu pristupiti putem https://{ip adresa virtualnog stroja}:9443/. Nakon inicijalnog kreiranja administratorskog računa, prikazuje nam se nadzorna ploča.




*Slika 18: Nadzorna ploča Portainer-a*

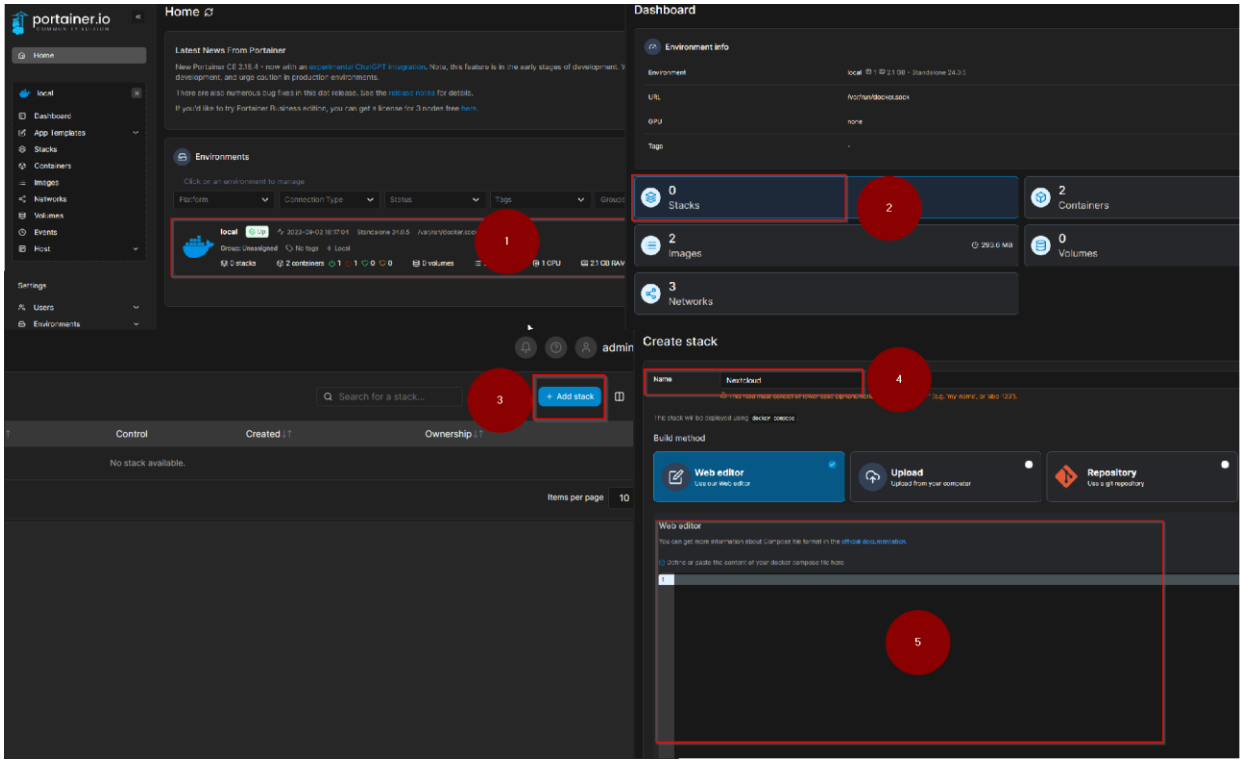
# 11. Nextcloud

Nextcloud je paket klijent-poslužitelj softvera otvorenog koda koji omogućava različite oblike dijeljenja, suradnje i komunikacije putem aplikacija, na primjer [55]:

- Dijeljenje datoteka
- Osobni upravitelj informacija (kontakti, kalendar, zadaci)
- Komunikacija (e-pošta, chat, video konferencije)
- Suradničko uređivanje dokumenata (tekst, integracija s Microsoft Office)

Temelji se na otvorenim principima i pruža suverenitet nad osobnim podacima, tj. uz vlastitu instancu Nextcloud-a, korisnici se oslobađaju od usluga zatvorenog koda kao što su Dropbox, Office 365 ili Google Drive. Može se na implementirati na računalima s jednom pločom kao što je Raspberry Pi pa sve do skaliranih podatkovnih centara s više milijuna korisnika [55].

Konfigurirati ćemo Nextcloud kao Docker kontejner putem Docker Compose-a pomoću Portainer grafičkog sučelja. Docker Compose je alat koji omogućava definiranje i pokretanje više kontejnerskih aplikacija. Unutar YAML datoteke konfiguriraju se svi potrebni servisi aplikacije te se zajedno pokrenu pomoću jedne naredbe [56]. U slučaju Nextclouda potrebna će nam biti baza podataka te sama aplikacija, iz tog razloga pogodno je korisitit Docker Compose. Kako bi kreirali novu Docker Compose konfiguraciju u Portainer-u potrebno je stvoriti Stack kao što je prikazano u slici 19.




*Slika 19: Stvaranje novog Stack-a*

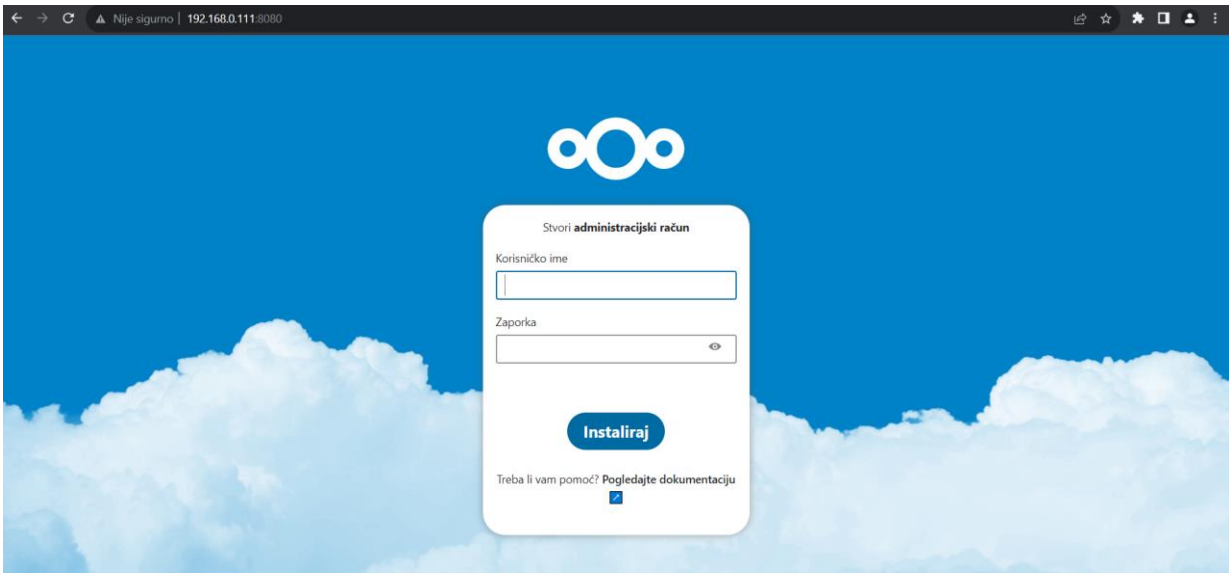
U koraku 5, slike 19, možemo upisati sadržaj Docker Compose datoteke u yaml formatu. A u slučaju Nextcloud-a izgledati će nešto poput:

```
version: '2'

# Definiramo koji servisi se pokreću
services:
  # Baza podataka
  db:
    # Koristimo MariaDB kao bazu podataka
    image: mariadb:10.5
    # Kontejener se treba uvijek ponovno pokrenuti ukoliko prestane raditi
    restart: always
    # Naredba koja će se izvršiti pri pokretanju kontejnera baze podataka
    # Definiramo izolaciju transakcija koja određuje kako se obrađuju
konkurentske transakcije i kako mogu pristupiti i mijenjati podatke
    # binlog definira format binary log-a
    # Ovo je tipična konfiguracija za baze podataka
    command: --transaction-isolation=READ-COMMITTED --binlog-format=ROW
    # Povežemo mjesto pohrane na domaćinu s mjestom pohrane unutar kontejnera
    volumes:
      - /nfs/nextcloud/database:/var/lib/mysql
    # Varijable okruženja potrebne za konfiguraciju MariaDB-a
    environment:
      - MYSQL_ROOT_PASSWORD=rootchangeme123
      - MYSQL_PASSWORD=changeme123
      - MYSQL_DATABASE=nextcloud
      - MYSQL_USER=nextcloud

  # Aplikacija
  app:
    image: nextcloud
    restart: always
    # Mapiranje porta 8080 na domaćinu s portom 80 unutar kontejnera
    # Moći ćemo pristupiti Nextcloudu na portu 8080 na domaćinu
    ports:
      - 8080:80
    # Mrežna veza između ovog servisa i servisa db
    links:
      - db
    volumes:
      - /nfs/nextcloud/data:/var/www/html
    environment:
      - MYSQL_PASSWORD=changeme123
      - MYSQL_DATABASE=nextcloud
      - MYSQL_USER=nextcloud
      - MYSQL_HOST=db
```

Sada možemo postaviti ovaj Stack te nakon dovršetka postavljanja moguće je pristupiti Nextcloud-u na adresi http://{ip adresa domaćina}:8080.




*Slika 20: Početni okvir za prijavu u Nexcloud*

Nakon početne konfiguracije Nextcloud računa, korisnik će imati potpunu funkcionalnost osobnog oblaka, uključujući pohranu datoteka. Važno je napomenuti da se oblak poslužuje s vlastitog računala/poslužitelja, što doprinosi postizanju cilja poboljšanja digitalnog suvereniteta. Sljedeći korak je omogućiti pristup oblaku sa svih osobnih uređaja na jednostavan i siguran način.

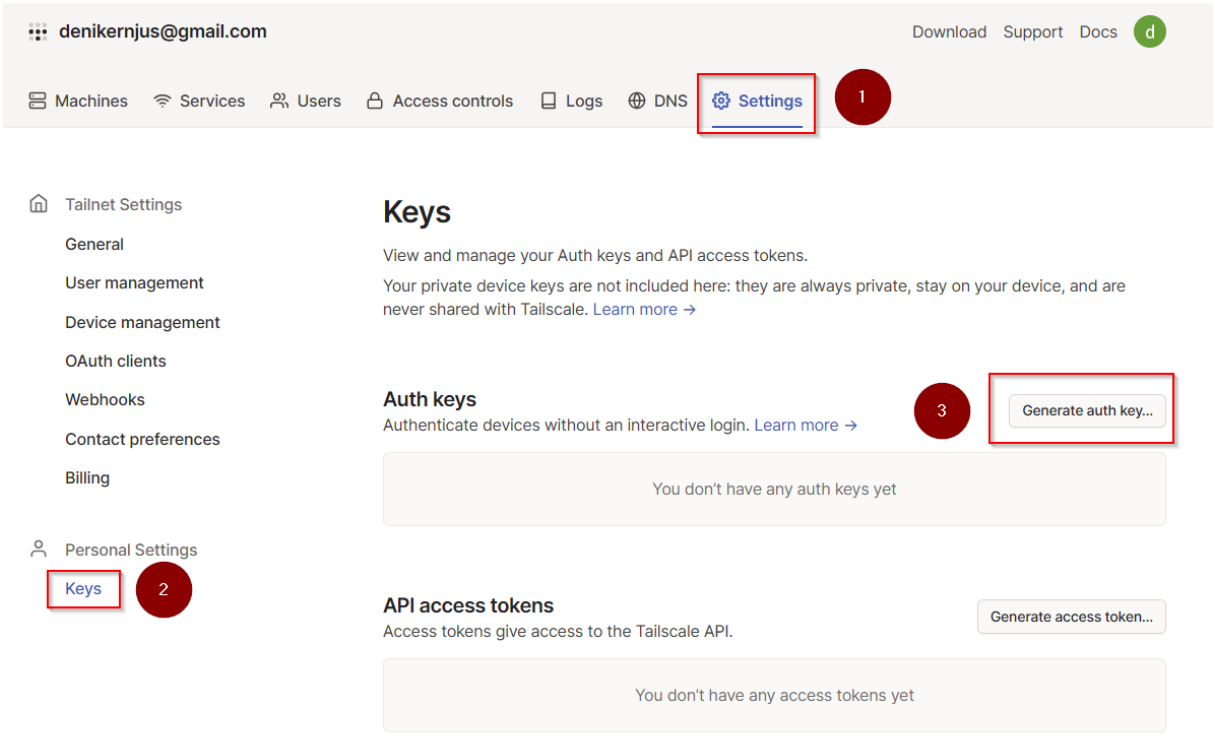
# 12. Tailscale

Tailscale je virtualna privatna mreža (engl. *Vritual Private Network*, kraće VPN) otvorenog koda koja omogućava dostupnost vaših uređaja i aplikacija bilo gdje u svijetu na siguran i jednostavan način. Koristi protokol otvorenog koda WireGuard[10] koji omogućava šifriranu vezu od točke do točke, što znači da samo uređaji unutar privatne mreže mogu komunicirati međusobno [58].

Tradicionalne virtualne privatne mreže usmjeravaju sav mrežni promet kroz središnji pristupni poslužitelj, no Tailscale kreira peer-to-peer mesh mrežu koju naziva tailnet. Na taj način se izbjegava centraliziranost što omogućuje veću propusnost i nižu latenciju jer mrežni promet može teći izravno između strojeva. Također, decentralizacija poboljšava stabilnost i pouzdanost smanjenjem pojedinačnih točaka kvara. Za korištenje Tailscale-a nije potrebno manualno prosljeđivati portove, pruža svoje usluge DNS-a te je moguće izložiti jedan ili više čvorova unutar tailnet-a vanjskom internetu ukoliko želimo da bude dostupan i uređajima koji nisu unutar tailnet-a [58].

Za početak potrebno je napraviti račun na Tailscale web stranici https://login.tailscale.com/. Zatim je potrebno unutar administratorske konzole generirati ključ putem kojeg će se pojedini uređaj moći spojiti u tailnet (slika 21). Ovaj postupak potrebno je ponoviti za svaki novi uređaj.

[10] WireGuard je komunikacijski protokol i besplatni softver otvorenog koda koji implementira šifrirane virtualne privatne mreže (VPN), a dizajniran je s ciljem jednostavnosti korištenja, velike brzine i niske površine napada [57].




*Slika 21: Stvaranje Tailscale ključa*

Na isti način kao u slici 19 moramo kreirati novi Stack unutar Portainer-a. On će imati sljedeću konfiguraciju:

```
version: '3.8'
services:
  tailscaled:
    container_name: tailscaled
    # Kontejner se pokreće kao korisnik 0, odnosno kao root korisnik
    user: "0:0"
    # Ima root ovlasti
    privileged: true
    # Dovoljavamo kontejneru da obavlja mrežne poslove
    cap_add:
        - NET_ADMIN
    volumes:
        - '/nfs/tailscale/SettingsFolder:/var/lib'
        - '/dev/net/tun:/dev/net/tun'
    # Kontejner koristi mrežni stog domaćina
    network_mode: "host"
    image: tailscale/tailscale
    command:
        - tailscaled
    restart: unless-stopped
    environment:
```

```
      - PUID=1000
      - PGID=1000
      - TS_USERSPACE=true
      - TS_AUTH_KEY={stvoreni Tailscale ključ}
      # Podmreže koje želimo da ovaj Tailscale čvor oglašava
      #- TS_ROUTES=192.168.0.0/24
```

Parametar „TS_ROUTES“ označava da se ovaj čvor u tailnetu može koristiti kao usmjerivač podmreža, odnosno da kroz ovaj čvor možemo pristupiti uređajima koji se nalaze na definiranoj podmreži. To je od velike koristi ukoliko želimo pristupiti uređaju kao što je pisač, ali nije moguće instalirati Tailscale direktno na njega. Kako bi ova funkcionalnost radila potrebno je na stroju domaćinu Docker kontejnera omogućiti IP prosljeđivanje sa sljedećim naredbama:

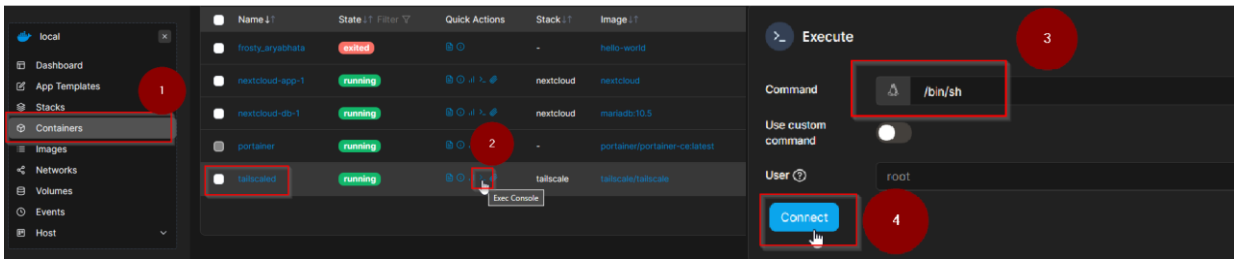
```
$ echo 'net.ipv4.ip_forward = 1' | sudo tee -a /etc/sysctl.conf
$ echo 'net.ipv6.conf.all.forwarding = 1' | sudo tee -a
/etc/sysctl.conf
$ sudo sysctl -p /etc/sysctl.con
```

No, mi nećemo iskoristiti ovu opciju pošto trenutno nemamo potrebe za njom i preporuka Tailscale je da se na svaki čvor na koji je to moguće postavi Tailscale zasebno [59].

Nakon postavljanja Tailscale-a postoji mogućnost da se čvor ne pridruži tailnetu, ukoliko je to slučaj, potrebno je unutar Docker kontejnera dodatno pridružiti čvor tako što upišemo naredbu:

```
$ tailscale up --authkey={stvoreni Tailscale ključ}
```

Konzoli Docker kontejnera možemo pristupiti putem Portainer sučelja prateći korake u slici 22.




*Slika 22: Pristupanje konzoli Docker kontejnera*

Sada možemo uređaje poput mobilnog telefona ili laptopa pridružiti istoj Tailscale mreži, odnosno našem tailnet-u te na taj način pristupiti Nextcloud-u i našim podacima izvan lokalne mreže. Za Windows, Android i iOS uređaje dostupna je aplikacija.

Ako želimo podijeliti sliku unutar našeg oblaka s osobom koja nema uređaj unutar našeg tailnet-a, možemo postaviti Tailscale Funnel. Tailscale Funnel omogućuje usmjeravanje prometa sa šireg interneta prema jednom ili više naših Tailscale čvorova. Prilikom omogućavanja ove funkcionalnosti, postavlja se javni DNS zapis za odabrani čvor

koji upućuje na Funnel poslužitelj kojim upravlja Tailscale. Kad korisnik pristupi dodijeljenoj adresi čvora, Funnel poslužitelj prihvaća dolazne zahtjeve i šalje TCP proxy putem Tailscale-a do našeg čvora. Čvor zatim završava TLS[11], što znači da Funnel ulazni čvorovi ne vide nikakve informacije o tom prometu ili o tome što se poslužuje. Vidljiva je samo IP adresa izvora i port, SNI [12]ime i količina bajtova koji prolaze kroz promet [62].

[11] Transport Layer Security (TLS) je kriptografski protokol dizajniran za pružanje sigurnosti komunikacije preko računalne mreže [60].

[12] SNI je proširenje protokola TLS kojim klijent označava na koji se naziv glavnog računala pokušava spojiti na početku procesa rukovanja [61].

# 13. Zaključak

Koncept digitalnog suvereniteta, kako na međunarodnoj tako i na individualnoj razini, stekao je značajnu važnost kao odgovor na razvoj privatnosti podataka i kontrole nad istima. Međunarodne regulative poput GDPR-a odraz su potrebe za zaštitom prava na osobne podatke na globalnoj razini, dok pojedinci tragaju za načinom na koji mogu ostvariti suverenitet nad vlastitom digitalnom imovinom.

Ovaj rad prikazuje način na koji se takvo što može ostvariti putem konfiguracije vlastitog poslužitelja datoteka. Fokusirajući se na softverski aspekt konfiguracije i korištenje alata otvorenog koda prikazan je opsežni vodič koji može poslužiti kao generalna uputa za ostvarivanje osobnog suvereniteta na gotovo besplatan način. Ovaj pristup omogućuje korisnicima da sigurno upravljaju svojim datotekama i pristupaju im s bilo kojeg mjesta, a sve to uz zadržavanje potpune kontrole nad svojim podacima.

U suvremenom društvu gdje je vlasništvo i kontrola nad osobnim podacima od sve veće važnosti, ovaj rad pruža alternativu za često nesigurne centralizirane komercijalne usluge. Sposobnost konfiguriranja vlastitog poslužitelja datoteka i oblaka omogućava povrat vlasništva i kontrole nad osobni podacima. Također, korištenje alata otvorenog koda pridonosi širem pokretu koji zagovara osobni digitalni suverenitet u digitalnom dobu.

# 14. Popis slika

# 15. Literatura


[1] Hrvatska enciklopedija, "suverenitet | Hrvatska enciklopedija." https://www.enciklopedija.hr/natuknica.aspx?id=58947 (accessed Aug. 10, 2023).

[2] German Presidency programme, "German EU council presidency- Digital sovereignty", Accessed: Aug. 08, 2023. [Online]. Available: https://erstelesung.de/wp-content/uploads/2020/10/20-10-14_Germany_EU_Digital-Sovereignty.pdf

[3] K.-L. Tan, C.-H. Chi, and K.-Y. Lam, "Analysis of Digital Sovereignty and Identity: From Digitization to Digitalization." arXiv, Feb. 21, 2022. doi: 10.48550/arXiv.2202.10069.

[4] A. Cattaruzza, D. Danet, S. Taillat, and A. Laudrain, "Sovereignty in cyberspace: Balkanization or democratization," in *2016 International Conference on Cyber Conflict (CyCon U.S.)*, Oct. 2016, pp. 1–9. doi: 10.1109/CYCONUS.2016.7836628.

[5] L. Floridi, "The Fight for Digital Sovereignty: What It Is, and Why It Matters, Especially for the EU," *Philos. Technol.*, vol. 33, no. 3, pp. 369–378, Sep. 2020, doi: 10.1007/s13347-020-00423-6.

[6] E. Digital, "New report on European Digital Infrastructure and Data Sovereignty - Archive // EIT Digital," Jun. 05, 2020. https://www.eitdigital.eu/newsroom/news/archive/new-report-on-european-digital-infrastructure-and-data-sovereignty/ (accessed Aug. 10, 2023).

[7] J. Pohle and T. Thiel, "Digital sovereignty," *Internet Policy Rev.*, vol. 9, no. 4, Dec. 2020, doi: 10.14763/2020.4.1532.

[8] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, in SIGMOD '00. New York, NY, USA: Association for Computing Machinery, Svibanj 2000, pp. 439–450. doi: 10.1145/342009.335438.

[9] E. Celeste and F. Fabbrini, "Competing Jurisdictions: Data Privacy Across the Borders," in *Data Privacy and Trust in Cloud Computing: Building trust in the cloud through assurance and accountability*, T. Lynn, J. G. Mooney, L. van der Werff, and G. Fox, Eds., in Palgrave Studies in Digital Business & Enabling Technologies. Cham: Springer International Publishing, 2021, pp. 43–58. doi: 10.1007/978-3-030-54660-1_3.

[10] Wikipedia, "Home server," *Wikipedia*. Jun. 11, 2023. Accessed: Aug. 13, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Home_server&oldid=1159612515

[11] R. Mens, "Home Server - Everything you want to Know!," *LazyAdmin*, May 27, 2021. https://lazyadmin.nl/home-network/home-server/ (accessed Aug. 13, 2023).

[12] Anwesha Ray, “Data Sovereignty, Protection and Future of Open Source,” Aug. 29, 2022. https://owncloud.com/news/data-sovereignty/ (accessed Aug. 12, 2023).

[13] S. H. last updated, “What Is ECC Memory in RAM? A Basic Definition,” *Tom’s Hardware*, Mar. 10, 2019. https://www.tomshardware.com/reviews/ecc-memory-ram-glossary-definition,6013.html (accessed Aug. 15, 2023).

[14] Oracle, “Sun StorEdge 3000 Family Configuration Service 2.5 User’s Guide.” https://docs.oracle.com/cd/E19236-01/817-3337-18/appa_raid_basic.html (accessed Aug. 15, 2023).

[15] R. Parry, “Tech Help - What wattage PSU do I need for my server? - Server Case Blog.” https://www.servercase.co.uk/blog/article/tech-help---what-wattage-psu-do-i-need-for-my-server (accessed Aug. 18, 2023).

[16] Howard, “What Are Server Cooling Technologies? | FS Community,” *Knowledge*, May 24, 2022. https://community.fs.com:7003/blog/what-are-server-cooling-technologies.html (accessed Aug. 18, 2023).

[17] PCMAG, “Definition of UPS,” *PCMAG*. https://www.pcmag.com/encyclopedia/term/ups (accessed Aug. 19, 2023).

[18] iXsystems, “TrueNAS SCALE - Hyperconverged Storage Scales Up & Out,” *TrueNAS - Welcome to the Open Storage Era*. https://www.truenas.com/truenas-scale/ (accessed Aug. 20, 2023).

[19] J. Fruhlinger, “What is hyperconvergence?,” *Network World*, Mar. 16, 2022. https://www.networkworld.com/article/3207567/what-is-hyperconvergence.html (accessed Aug. 20, 2023).

[20] Fishezz, “Hyper-converged infrastructure,” *Wikipedia*. May 08, 2023. Accessed: Aug. 20, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Hyper-converged_infrastructure&oldid=1153859882

[21] Tim Fisher, “What Is an ISO File?,” *Lifewire*, Sep. 08, 2021. https://www.lifewire.com/iso-file-2625923 (accessed Aug. 20, 2023).

[22] J. Bonwick, M. Ahrens, V. Henson, M. Maybee, and M. Shellenbaum, “The Zettabyte File System,” *2003*, Accessed: Aug. 23, 2023. [Online]. Available: https://www.cs.hmc.edu/~rhodes/courses/cs134/fa20/readings/The%20Zettabyte%20File%20System.pdf

[23] Wikipedia, “Copy-on-write,” *Wikipedia*. Mar. 25, 2023. Accessed: Sep. 06, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Copy-on-write&oldid=1146594085

[24] M. Ahrens, “OpenZFS: a Community of Open Source ZFS Developers.",” *AsiaBSDCon 2014*, p. 27, 2014.

[25] iXsystems, “ZFS: Enterprise-Grade File System - OpenZFS with TrueNAS,” *TrueNAS - Welcome to the Open Storage Era*. https://www.truenas.com/zfs/ (accessed Aug. 24, 2023).

[26] LevelOneTechs user, “Getting Started with TrueNAS Scale | Part 2 | Learning ZFS Storage in TrueNAS; Creating a Pool, Dataset and Snapshot Task,” *Level1Techs Forums*, Mar. 30, 2022. https://forum.level1techs.com/t/getting-started-with-truenas-scale-part-2-learning-zfs-storage-in-truenas-creating-a-pool-dataset-and-snapshot-task/182481 (accessed Aug. 24, 2023).

[27] OpenZFS, “Hardware — OpenZFS documentation.” https://openzfs.github.io/openzfs-docs/Performance%20and%20Tuning/Hardware.html#drive-interfaces (accessed Aug. 24, 2023).

[28] iXsystems, “Running S.M.A.R.T. Tests.” https://www.truenas.com/core/coretutorials/tasks/runningsmarttests/ (accessed Sep. 06, 2023).

[29] iXsystems, “L2ARC.” https://www.truenas.com/references/l2arc/ (accessed Sep. 06, 2023).

[30] Intel, “What Is RAID 5 Write Hole (RWH) Protection in Intel® Virtual RAID...,” *Intel*. https://www.intel.com/content/www/us/en/support/articles/000057368/memory-and-storage/datacenter-storage-solutions.html (accessed Sep. 06, 2023).

[31] Jeff Bonwick, “RAID-Z (Jeff Bonwick’s Blog),” Nov. 17, 2005. https://web.archive.org/web/20141216015058/https://blogs.oracle.com/bonwick/en_US/entry/raid_z (accessed Aug. 25, 2023).

[32] ShrutiBahatla, “Swap Space in Operating System,” *GeeksforGeeks*, Mar. 28, 2018. https://www.geeksforgeeks.org/swap-space-in-operating-system/ (accessed Sep. 06, 2023).

[33] Jim Salter, “Will ZFS and non-ECC RAM kill your data? – JRS Systems: the blog,” Feb. 03, 2015. https://jrs-s.net/2015/02/03/will-zfs-and-non-ecc-ram-kill-your-data/ (accessed Aug. 25, 2023).

[34] Wikipedia, “Server Message Block,” *Wikipedia*. Jul. 25, 2023. Accessed: Sep. 06, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Server_Message_Block&oldid=1167038718

[35] R. Thurlow, “RPC: Remote Procedure Call Protocol Specification Version 2,” Internet Engineering Task Force, Request for Comments RFC 5531, May 2009. doi: 10.17487/RFC5531.

[36] B. Pawlowski, D. Noveck, D. Robinson, and R. Thurlow, "The NFS version 4 protocol," in *In Proceedings of the 2nd International System Administration and Networking Conference (SANE 2000*, Citeseer, 2000.

[37] Red Hat, "What is a virtual machine (VM)?" https://www.redhat.com/en/topics/virtualization/what-is-a-virtual-machine (accessed Aug. 26, 2023).

[38] Red Hat, "What is a hypervisor?" https://www.redhat.com/en/topics/virtualization/what-is-a-hypervisor (accessed Aug. 26, 2023).

[39] Red Hat, "What is KVM?" https://www.redhat.com/en/topics/virtualization/what-is-KVM (accessed Aug. 26, 2023).

[40] iXsystems, "Adding and Managing VMs." https://www.truenas.com/docs/scale/scaletutorials/virtualization/creatingmanagingvms scale/ (accessed Aug. 26, 2023).

[41] K. Huynh and S. Hajnoczi, "KVM/QEMU storage stack performance discussion," in *Linux Plumbers Conference*, 2010.

[42] iXsystems, "Accessing NAS From a VM." https://www.truenas.com/scale/scaletutorials/virtualization/accessingnasfromvm/ (accessed Aug. 27, 2023).

[43] Wikipedia, "MAC address," *Wikipedia*. Aug. 25, 2023. Accessed: Sep. 06, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=MAC_address&oldid=1172141446

[44] ArchWiki, "Unified Extensible Firmware Interface - ArchWiki." https://wiki.archlinux.org/title/Unified_Extensible_Firmware_Interface (accessed Sep. 06, 2023).

[45] W. Wilson, "TrueNAS Scale: Ultimate Home Setup incl. Tailscale," *Level1Techs Forums*, Aug. 01, 2022. https://forum.level1techs.com/t/truenas-scale-ultimate-home-setup-incl-tailscale/186444 (accessed Jul. 17, 2023).

[46] C. M. Lonvick and T. Ylonen, "The Secure Shell (SSH) Protocol Architecture," Internet Engineering Task Force, Request for Comments RFC 4251, Jan. 2006. doi: 10.17487/RFC4251.

[47] Debian, "fstab - Debian Wiki." https://wiki.debian.org/fstab (accessed Aug. 28, 2023).

[48] systemd, "System and Service Manager." https://systemd.io/ (accessed Sep. 06, 2023).

[49] S. Hameed, "How to set up a static IP address on Debian 11." https://linuxhint.com/debian-static-ip-configuration/ (accessed Aug. 28, 2023).

[50] B. Bashari Rad, H. Bhatti, and M. Ahmadi, “An Introduction to Docker and Analysis of its Performance,” *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 173, p. 8, Mar. 2017.

[51] I. Buchanan, “Containers vs Virtual Machines,” *Atlassian*. https://www.atlassian.com/microservices/cloud-computing/containers-vs-vms (accessed Sep. 04, 2023).

[52] Docker Inc., “Install Docker Engine on Debian,” *Docker Documentation*, 44:37 --0500 500. https://docs.docker.com/engine/install/debian/ (accessed Aug. 30, 2023).

[53] Portainer, “Release Notes.” https://docs.portainer.io/ (accessed Aug. 30, 2023).

[54] Portainer, “Portainer Documentation.” https://docs.portainer.io/ (accessed Aug. 31, 2023).

[55] ArchWiki, “Nextcloud - ArchWiki.” https://wiki.archlinux.org/title/Nextcloud (accessed Sep. 04, 2023).

[56] Docker Inc., “Docker Compose overview,” *Docker Documentation*, 46:06 + +0100 100AD. https://docs.docker.com/compose/ (accessed Sep. 02, 2023).

[57] Wikipedia, “WireGuard,” *Wikipedia*. Aug. 27, 2023. Accessed: Sep. 06, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=WireGuard&oldid=1172475650#cite_note-wireguard-site-5

[58] Tailscale, “What is Tailscale?,” *Tailscale*, Aug. 08, 2022. https://tailscale.com/kb/1151/what-is-tailscale/ (accessed Sep. 02, 2023).

[59] Tailscale, “Subnet routers and traffic relay nodes,” *Tailscale*, Aug. 17, 2023. https://tailscale.com/kb/1019/subnets/ (accessed Sep. 03, 2023).

[60] Wikipedia, “Transport Layer Security,” *Wikipedia*. Sep. 04, 2023. Accessed: Sep. 06, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Transport_Layer_Security&oldid=1173869666

[61] Wikipedia, “Server Name Indication,” *Wikipedia*. Sep. 02, 2023. Accessed: Sep. 06, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Server_Name_Indication&oldid=1173420385#cite_note-rfc3546-1

[62] Tailscale, “Tailscale Funnel,” *Tailscale*, Aug. 21, 2023. https://tailscale.com/kb/1223/tailscale-funnel/ (accessed Sep. 03, 2023).