Opinnäytetyö (AMK)

Tieto- ja viestintätekniikka

2021

Lauri Kattelus

OHJELMISTOKEHITTÄJÄN PERUSPALVELUT TARJOAVAN PALVELIMEN TOTEUTUS

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tieto-ja viestintätekniikka

2021 | 40 sivua

Lauri Kattelus

OHJELMISTOKEHITTÄJÄN PERUSPALVELUT TARJOAVAN PALVELIMEN TOTEUTUS

Ohjelmistokehittäjälle hakukone- versionhallinta- tiedosto- ja sähköpostipalvelut ovat erittäin tärkeitä. Ilman näitä työskentely monien teknologioiden, yhteydenpidon tärkeyden ja jatkuvan integraation aikana on hankalaa. Ohjelmistokehitystä tehdään monesta eri sijainnista, monen eri henkilön voimin. Tällöin tiedostojen ja ohjelmistokoodin helppo jakaminen on olennainen osa projektia.

Opinnäytetyön tavoitteena oli toteuttaa ohjelmistokehittäjälle tärkeät palvelut tarjoava palvelin. Opinnäytetyössä tutkittiin, miten korvataan kolmannen osapuolen palvelut itse ylläpidettävillä palveluilla ja toteutettiin palvelin, joka tarjoaa ohjelmistokehittäjän peruspalvelut. Opinnäytetyö toteutettiin kahdessa vaiheessa, jotka olivat tutkimus- ja toteutusvaihe.

Tutkimusvaiheessa tutkittiin, minkälaisia ohjelmistovaihtoehtoja on tarjolla kolmansien osapuolien tarjoamille palveluille, vertailtiin niiden eroja ja tehtiin valinta mitä ohjelmistoa minkäkin palvelun korvaamiseen käytetään. Ohjelmistot valittiin ennen toteutusvaiheen aloittamista tutkimusvaiheessa asetettujen kriteerien mukaan. Palvelut, jotka toteutettiin ovat korkealla tasolla määriteltyinä: sähköposti-, versionhallinta-, tiedosto-, kalenteri-, hakukone- ja yksityiset DNS-palvelut. Toteutusvaiheessa hankittiin yksityinen virtuaalinen palvelin ja asennettiin palvelut toteuttavat ohjelmistot palvelimelle. Ohjelmistot asennettiin hyödyntäen Docker-ohjelmistoa. Docker mahdollistaa ohjelmistojen eriyttämisen toisistaan hyödyntäen virtualisointia. Palveluiden ollessa eriytettynä toisistaan on yksittäisen palvelun ohjelmisto-ongelmien ratkonta helpompaa, koska ongelma on mahdollista rajata tiettyyn palvelimen virtualisoituun ympäristöön ja koska yksittäisten palveluiden sammuttaminen ja käynnistäminen on nopeaa ja helppoa mahdollisissa ongelmatilanteissa.

ASIASANAT:

SSH, VPS, Docker, Linux, Email, Versionhallinta

BACHELOR´S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information and Communications Technology

2021 | 40 pages

Lauri Kattelus

IMPLEMENTATION OF A SERVER CONTAINING BASIC SERVICES FOR SOFTWARE DEVELOPERS

For a software developer a search engine, version control, files, and email services are important. Developing software would be hard without these technologies. As multiple technologies are used, the importance of communication and continuous integration is increasing. When software development is carried out by multiple people in different locations, the sharing of code and files becomes an essential part of the project.

The objective of this thesis was to implement a server which provides important services for a software developer. The thesis studied how to replace third-party provided services with self-maintained alternatives and for this purpose a server that provides these services was implemented. The work in this thesis was carried out in two phases which were the research and the implementation phases.

The research phase consisted of researching what software there is to replace third-party services and deciding which ones to use. Before the start of the implementation phase, the software was selected according to criteria set in the research phase. The services implemented are defined at a high level as an email, a version control, a file, a calendar, a search engine, and a private DNS service. In the implementation phase, a virtual private server was acquired, and the selected software was installed on the server. The software was installed using the Docker software. Docker allows services to be separated from each other using virtualization. When services are separated from each other, it is easier to solve the problems of an individual service because it is easy to limit the problem into a specific virtualized environment. Additionally, Docker makes it quickly and easily facilitates the launch or shutdown of single services if problems arise.

KEYWORDS:

SSH, VPS, Docker, Linux, Email, Version control

SISÄLTÖ

KUVAT

TAULUKOT

SANASTO

CGI	Common Gateway Interface. Tekniikka palvelimella tuottaa selaimelle verkkosivu CGI-ohjelmiston avulla.
CSS	Cascading Style Sheets. Web-sivuston tyylimäärittelytapa.
DKIM	DomainKeys Identified Mail. Sähköpostien lähettäjän todennustapa.
DNS	Domain Name Server internetin nimipalvelujärjestelmä.
Docker	Ohjelmisto, joka mahdollistaa järjestelmätason virtualisoinnin.
Dockerfile	Docker-reseptin sisältävä tiedosto.
Docker-kontti	Docker-ohjelmistolla luotu virtualisoitu käyttöjärjestelmä.
Docker-kuva	Docker-kontin jakotiedosto.
Docker-resepti	Docker-kontin rakennusohje.
HTTPS	Hypertext Transfer Protocol Secure. Protokolla suojattuun tiedonsiirtoon webissä.
IMAP	Internet Message Access Protocol. Protokolla sähköpostien lukemiseen
POP3	Post Office Protocol version 3. Protokolla sähköpostien hakemiseen
SMTP	Simple Mail Transfer Protocol. Sähköpostien välitys protokolla
MTA	Mail Transfer Agent. Ohjelmisto sähköpostiviestienlähettämiseen käyttäen SMTP protokollaa.
SSH	Secure System Shell. Salatun tietoliikenteen protokolla, jonka avulla voidaan ottaa yhteys SSH-palvelimeen mahdollistaen mm. tietokoneen etäkäytön.
SSL/TLS	Transport Layer Security/Secure Sockets Layer. Salausprotokolla, joka mahdollistaa mm WWW-sivustojen siirron salattuna.
VPN	Virtual Private Network. Tapa verkkoliikenteen ohjaamiseen toisen laitteen verkon kautta.
SVN	Subversion-versionhallintajärjestelmä.
Git	Git-versionhallintajärjestelmä

1 JOHDANTO

Opinnäytetyön tarkoitus on tutkia sellaisia työkaluja ja ohjelmistoja, joilla ohjelmistokehittäjälle tärkeiden palveluiden hallinta voitaisiin siirtää kolmannen osapuolen hallusta itselle, sekä toteuttaa palvelin, joka tarjoaa nämä palvelut.

Itse ylläpidettävät palvelut mahdollistavat ohjelmistokehittäjälle tärkeiden palveluiden käytön ilman että käyttäjä on riippuvainen muista kuin palvelimen tarjoajastaan.

Opinnäytetyössä käytetään henkilökohtaista virtuaalista palvelinta, sillä se tarjoaa yleisesti ottaen korkeamman käytettävyysajan sekä paremmat kaistanleveydet kuin henkilökohtaisesti kotona ylläpidettävä palvelin.

Opinnäytetyöraportin ensimmäisessä osassa käsitellään työn tutkimusvaihetta, jossa määritettiin kriteerit ohjelmistoille, jotka palvelimelle asennettiin toteuttamaan eri palvelut ja vertaillaan eri työkaluja palveluiden toteuttamiseen. Lisäksi kerrotaan kunkin palvelun kohdalla, minkä vuoksi se on valittu mukaan työhön.

Opinnäytetyöraportin toisessa osassa käsitellään työn toteutusvaihetta sekä lopputulosta. Esitellään palvelimen pystytys, eri palveluiden asennusvaiheet, toiminnallisuuksien todentaminen ja lopputulokset.

2 OHJELMISTOJEN JA PALVELUIDEN VALINTA

Tässä luvussa määritellään kriteerit ohjelmistoille, raportoidaan valinnat palvelimesta, ohjelmistoista sekä vertaillaan niitä muihin vaihtoehtoisiin ohjelmistoihin. Palveluiden asentamiseen hyödynnetään Dockeria.

Docker on paketti ohjelmistoja, jotka mahdollistavat ohjelmiston ajamisen omassa eriytetyssä ympäristössä, joka sisältää mm. käyttöjärjestelmän sekä omat asennetut ohjelmistonsa. Tämänkaltaisia Dockerilla luotuja eriytettyjä ympäristöjä kutsutaan konteiksi. Ohjelmistojen käynnistäminen omissa konteissaan helpottaa palveluiden ylläpitoa erittymällä palvelut toisistaan sekä nopean palveluiden nopean käynnistämisen ja sammuttamisen mahdollisissa ongelmatilanteissa

Korkealla tasolla määriteltynä työssä toteutettavat palvelut ovat

- sähköpostipalvelu
- versionhallintapalvelu
- yksityinen DNS-palvelu
- henkilökohtainen verkkosivu
- tiedostopalvelu
- hakukonepalvelu
- kalenteripalvelu.

Ohjelmistojen valinnalle määritellyt kriteerit ovat

- Ohjelmistojen tulee olla vapaata ja avointa lähdekoodia.
- Ohjelmistojen tulee olla tietoturvallisia.
- Ohjelmistojen tulee olla laajennettavissa ja määriteltävissä käyttäjän tarpeisiin.
- Ohjelmistojen tule olla kevyitä, jotta yksi virtuaalipalvelin riittää ylläpitämään kaikki palvelut.

Ohjelmistojen eduiksi katsotaan että

- Ohjelmisto tekee yhden asian ja tekee sen hyvin.
- Ohjelmistolla on riittävän suuri yhteisö ongelmatilanteiden ratkomisen helpottamiseksi.

2.1 Palvelin

Palveluiden asentamisen ja käytön mahdollistamiseksi tarvitaan palvelin, jossa kyseisiä palveluita voidaan ylläpitää ja tarjota käyttäjille.

Työssä käytettiin virtuaalipalvelinta sen tarjoaman hyvän saatavuuden vuoksi. Virtuaalipalvelin tarjoaa kotiserveriin verrattaen usein paremmat käyttöoikeudet, koska useimmat internet-yhteyden palveluntarjoajat estävät käyttäjiltä tiettyjen porttien käytön niiden väärinkäyttöalttiuden vuoksi. Esimerkkinä tästä on roskapostin lähettäminen porttia 25 käyttäen.

Virtuaalipalvelin on palvelin, joka pyörii fyysisellä palvelimella muiden virtuaalipalvelimien kanssa samanaikaisesti. Virtuaalipalvelimien etuna dedikoituun palvelimeen on sen paljon huokeampi hinta. Virtuaalipalvelimen heikkouksiksi luetaan niiden yleisesti ottaen heikompi laskentateho verrattaessa dedikoituihin palvelimiin.

Virtuaalipalvelimen valinta tehtiin perehtymällä eri virtuaalipalvelimen tarjoajiin sekä heidän virtuaalipalvelin paketteihinsa.

Virtuaalipalvelimen valinnalle asetettiin seuraavat kriteerit

- Palvelimen ei tule sulkea oletuksena portteja.
- Palvelimen tulee sijaita Euroopassa mahdollistaen mahdollisimman pienen viiveen.
- Palvelimen tulee tarjota riittävästi kaistanleveyttä ulos- ja sisäänpäin.
- Palveluntarjoajan tulee olla luotettava ja tunnettu alalla.
- Palvelimen tulee mahdollistaa käänteisnimipalvelun asettaminen.
- Palvelimen hinnan tulee olla alle 60 €/vuosi.

Palveluntarjoajat, joiden välillä vertailu toteutettiin, olivat

- OVH
- Vultr
- Digital Ocean
- A2 Hosting
- Hetzner.

Kaikki palveluntarjoajat ovat tunnettuja ja toimineet vuosia markkinoilla.

Kaikkien palveluntarjoajien virtuaalipalvelimet täyttivät kaikki kriteerit, paitsi Vultr joka esti portin 25 käytön oletuksena [1]. Portin avaaminen on mahdollista myös Vultrilla.

Kaikki palveluntarjoavat tarjosivat varteenotettavia palvelinkokoonpanoja. Vertailu (Taulukko 1) palvelunkokoonpanojen sekä niiden kuukausihintojen perusteella työhön valittiin käytettäväksi Hetznerin palvelin sen tarjoaman parhaan hyötysuhteen vuoksi. Hetzner tarjosi kaikkiin muihin palveluntarjoajiin verrattaen kaksinkertaisesti prosessoriytimiä, lähes kaksinkertaisesti tallenustilaa ja kymmenkertaisesti verkkoliikennettä muihin paitsi OVH:n palvelimiin verrattaessa.

Taulukko 1 Virtuaalipalvelin hinta/kokoonpano vertailu.

Tarjoaja	CPU	Storage	RAM	Bandwidth	Hinta
Vultr	1	25GB	1GB	1TB	4,29€/kk
A2Hosting	1	20GB	512MB	2TB	4,27€/kk
OVH	1	20GB	2GB	~	3.00€/kk
Digital Ocean	1	25GB	1GB	1TB	4.29€/kk
Hetzner	2	40GB	2GB	20TB	4.33€/kk

2.2 Sähköpostipalvelu

Ohjelmistokehittäjät käyttävät työssään usein sähköpostia kommunikointiin joko muiden tiimissä työskentelevien kehittäjien tai asiakkaiden kanssa. On siis tärkeää, että kehittäjällä on käytössä sähköposti, jonka vuoksi työhön valittiin toteuttavaksi sähköpostipalvelu.

Huonosti konfiguroidut sähköpostipalvelimet ovat suuri heikkous internetissä. Yksinkertaisen sähköpostiprotokollan ongelmana on se, että botit pystyvät lähettämään protokollan yli todella paljon roskapostia. Vuonna 2019 lähetettiin 293 miljardia sähköpostia per päivä [2]. Tästä noin 55 % oli pelkkää roskapostia [3].

Tämän vuoksi työssä ei luoda itse Docker-reseptiä sähköpostipalvelimelle vaan käytämme valmista sähköpostipalvelimen luontiin tehtyä Docker-kuvaa.

Valmiiksi oikein konfiguroidun sähköpostipalvelimen käyttö lisää tietoturvaa palvelimella koska näin käyttäjältä ei unohdu asettaa kaikkein tärkeimpiä asetuksia sähköpostipalvelimella. Asetusten asetus väärin altistaa sähköpostipalvelimen erilaisille hyökkäyksille kuten sähköpostipalvelimen käytön roskapostin välittämisessä.

Valinta tehtiin kolmen hyvin suositun Docker kuvan välillä, jotka olivat

- Docker-mailserver
- Mailcow
- Mailu.

Valinta tehtiin tarkastelemalla työhön määriteltyjä kriteerejä sekä vertailemalla niitä jakeluiden tarjoamiin ratkaisuihin. Mailcow ja Mailu sisältävät web-pohjaiset käyttöliittymät, jotka eivät ole tarpeellisia työn käyttötarkoituksen kannalta. Molemmat jakelut käyttävät sähköpostien ja käyttäjien tallettamiseen tietokantaa, jolloin palvelimelle tulisi asentaa myös tietokanta. Docker-mailserver ei sisällä web-pohjaista käyttöliittymää ja käyttää tietokoneen omaa tiedostojärjestelmää sähköpostien ja käyttäjien tallettamiseen. Docker-kuvien järjestelmävaatimuksista huomattiin Docker-mailserverin olevan jakeluista kevyin (Taulukko 2).

Taulukko 2 Sähköpostipalvelin järjestelmävaatimukset.

Jakelu	**RAM**	**Disk**	**CPU**
Docker-mailserver	512 MB	-	1 Core
Mailcow	800 MB	5 GB	1 GHz
Mailu	2 GB	-	1 Core

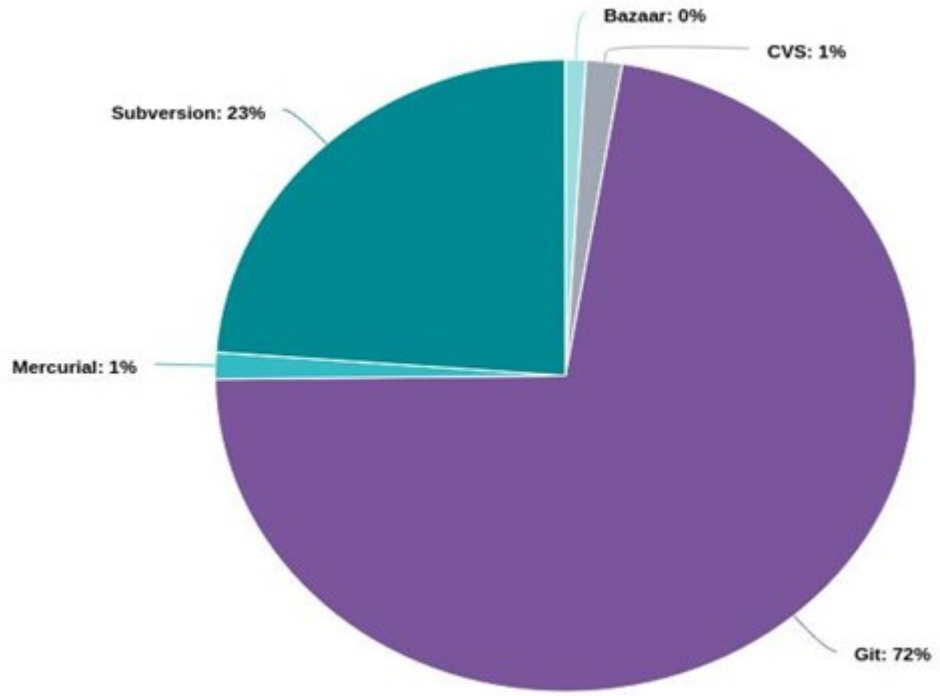
Palvelun loppukäyttäjät käyttävät sähköpostipalvelinta hyödyntäen sähköpostiohjelmistoja kuten Thunderbird ja Mutt. Nämä mahdollistavat helpon sähköpostien lukemisen sekä lähettämisen hyödyntäen SMTP- ja IMAP-protokollia. Tällöin web-pohjaista käyttöliittymää ei tarvita.

Työhön valittiin käytettäväksi Docker-mailserverin Docker-kuvaa sen pienten resurssivaatimusten ja yksinkertaisuutensa vuoksi.

2.3 Versionhallinnan backend

Versionhallintaohjelmistot kuten Git ja SVN ovat tärkeitä työkaluja työskennellä niin yksin kuin tiiminkin kanssa. Versionhallinta mahdollistaa lähdekoodien jakamisen ja tallentamisen siten, että lähdekoodi ei katoa, mikäli henkilökohtaisella tietokoneella tapahtuu virhe. Tämän vuoksi työhön valittiin toteutettavaksi myös versionhallinta.

Versionhallintajärjestelmistä suosituin on Git, toiseksi suosituin on SVN. [4] Nämä kaksi ovat selkeästi käytetyimmät versionhallintajärjestelmät verrattuna muihin järjestelmiin kuten huomataan kuvasta 1 jossa esitetään graafi Black Duck Open Hub -sivuston sisältämistä tietovarastoista. Google trends -vertailussa huomataan, että Gitin hakumäärät ovat reilusti korkeammat kuin SVN:n (Kuva 2).



Kuva 1. Black Duck Open Hub -sivuston tietovarasto-vertailu [5]

Kuva 2. Google trends -vertailu Git- ja SVN-versionhallintajärjestelmien välillä

Gitin ja SVN:n suurin ero on, että SVN toimii keskitetyllä toimintamallilla ja Git epäkeskitetyllä.

SVN:n toimintamallissa palvelimella on päätietovarasto, josta kehittäjät saavat kopion lähdekoodista ja tehtyään muutokset lähdekoodiin lähettävät he muutokset päätietovarastoon.

Gitin toimintamallissa ei ole tarkasti määritettyä päätietovarastoa. Kehittäjät kopioivat tietovaraston palvelimelta itselleen, tekevät muutokset tietovarastoon, jonka jälkeen he voivat lisätä muutoksensa toiseen tietovarastoon. Yleisesti ottaen työskentely tapahtuu palvelimella olevan tietovaraston kautta.

Molemmissa versionhallintajärjestelmissä on etunsa. Gitin eduiksi luetaan parempi lähdekoodin haaroittamismahdollisuus sekä yhdistettävyys. SVN:n etuja ovat mahdollisuus käsitellä suurempia tiedostokokoja.

Gitin suosion sekä hieman vapaamman työskentelyn mahdollistavan toimintomallin vuoksi työhön käytettäväksi valittiin Git-versionhallintajärjestelmä.

2.4 Versionhallinnan frontend

Jos versionhallinnassa olevien projektien lähdekoodeja haluaa esittää verkkosivulla helposti ja selkeästi tarvitaan palvelimelle Git-frontend ohjelmisto. Esimerkkejä palveluista, jotka tarjoavat Git-palvelimen ja Git-frontendin käyttäjilleen ovat mm. GitHub ja GitLab.

Suosituimpia ohjelmistoja, jotka helpottavat Git-frontendin tarjoamista ovat mm.

- GitWeb
- GitLab
- GitList
- Cgit.

GitLab on moderni versionhallintajärjestelmän tarjoava ohjelmisto. GitLab ohjelmisto sisältää valmiin asennuspaketin Gitille, web-frontendiksi sekä monelle muulle ominaisuudelle. Se on listatuista ohjelmistoista raskain vaihtoehto ja sisältää enemmän ominaisuuksia kuin mitä työmme tarvitsee. [6]

GitWeb on web-frontendin Gitille tarjoava CGI-ohjelmisto, joka on kirjoitettu Perlillä. Se sisältyy mukaan Git-palvelimen asennukseen ja sen käyttöönotto ohjeet löytyvät Git pro -kirjasta. GitWebin ulkoasu on muokattavissa CSS-tyylimäärittelyiden avuilla. [7]

GitList on moderni web-frontend Git-palvelimelle. Se on kirjoitettu PHP:llä ja JavaScriptillä. Gitlabin ohella GitList on näistä ohjelmistoista oletuksena moderneimman ulkoasun. Viimeisin versio GitLististä on julkaistu huhtikuussa 2019. [8]

Cgit on C:llä kirjoitettu web-frontend Git-palvelimelle. Cgit kuvailee itseään hypernopeaksi web-frontendiksi Git-tietovarastoille [9]. Cgitin ulkoasu on muokattavissa käyttäen CSS tyylimäärittelyitä ja se tarjoaa kattavat konfigurointimahdollisuudet asetustiedoston kautta.

Vertailu tehtiin lopuksi GitListin ja Cgitin välillä. Cgitin minimalistisuuden sekä tehokkuuden vuoksi työssä päädyttiin käyttämään sitä. Se tarjoaa frontendiltä toivotut toiminnallisuudet ja on kriteereiden määritellyn mukaisesti mahdollisimman minimalistinen sekä tehokas.

2.5 Yksityinen DNS- ja VPN-palvelu.

Yksityisellä DNS-palvelulla voidaan mahdollistaa verkonlaajuinen verkkoliikenteen suodatus. Tämä tarkoittaa mm. mainos-, seuranta ja haittaliikenteen estämistä. Tällaisiin DNS-palveluihin on tarjolla estolistoiksi kutsuttuja suuria listoja, jotka sisältävät tunnettuja epätoivotun liikenteen osoitteita. Tunnetuin ohjelmisto, joka tarjoaa tämänkaltaisen suodatuksen on Pi-hole. [10]

Pi-hole on ohjelmisto, jonka läpi tunneloidaan liikenne niille laitteille, joista halutaan estää epätoivottu verkkoliikenne. Yksityinen DNS- ja VPN-palvelu mahdollistavat tarkemman verkkoliikenteen tarkkailun ja määrittelyn, jonka vuoksi yksityinen DNS- ja VPN-palvelu valittiin työhön.

Pi-hole toimii tässä työssä DNS-sinkholena mutta Pi-holea on mahdollista käyttää kokonaisvaltaisena DNS-palvelimenakin. DNS-sinkhole tarkoittaa käytännössä sitä, että käyttäjä tekee kyselyn Pi-holen DNS-palveluun mistä IP-osoitteesta löytää verkkosivuston google.com. Pi-hole tarkistaa tietääkö se jo vastauksen välimuistista tälle, mikäli ei tiedä se tarkistaa kuuluuko kyseinen osoite estettyjen osoitteiden listalle. Mikäli se ei kuulu estolistalle eikä sitä löydy välimuistista Pi-hole tekee kyselyn ulkoiselle DNS-serverille ja välittää vastauksen käyttäjälle.

Jotta mahdollistetaan DNS-palvelun etäkäyttö, asennetaan palvelimelle myös OpenVPN -ohjelmisto. OpenVPN-ohjelmiston avulla mahdollistetaan palvelimen verkon käyttö esimerkiksi puhelimella, jolloin puhelimen verkkoliikenne kulkee palvelimen verkon kautta ja saapuva data suodattuu.

2.6 Henkilökohtainen verkkosivu

Henkilökohtainen verkkosivu on hyödyllinen mm. tarjoamaan linkit palvelimelta löytyviin palveluihin, kertomaan hieman palvelimen omistajasta tai palvelimen käyttötarkoituksista. Riippuen verkkosivuston tarpeista sekä määrittelyistä, joita verkkosivulta vaaditaan, valitaan teknologiat, joilla verkkosivusto toteutetaan. Ohjelmistokehittäjällä on hyvä mahdollisuus esimerkiksi esitellä hallitsemansa teknologiat omalla verkkosivustolla, jonka vuoksi henkilökohtainen verkkosivu valittiin mukaan työhön.

Tämän työn tarpeisiin riittää yksinkertainen HTML/CSS-sivusto. Verkkosivun tarkoituksena on tarjota tietoa palvelimen omistajasta ja linkit palvelimen palveluihin, jotka on tarkoitettu julkisesti näkyviksi kuten versionhallinnan frontend. Verkkosivulle ei toteuteta työn puitteissa toimintoja, joilla esimerkiksi lähetetään palvelimelle tietoja, jolloin ei tarvita JavaScriptiä tai PHP:tä. Yksinkertainen HTML/CSS-sivusto on yleisesti ottaen todella nopea ja vähän laskentatehoa käyttävä.

Verkkosivusto vaatii myös HTTP palvelin -ohjelmiston tarjoamaan sivuston internetiin. Tähän sivustoon HTTP-palvelin-ohjelmistoksi valittiin Lighttpd, joka on turvallinen, nopea ja aikakriittinen HTTP-palvelin. Se tarjoaa tärkeät ominaisuudet, joita HTTP-palvelimelta tarvitaan kuten TLS- ja SSL-tuen.

2.7 Tiedosto- ja kalenteripalvelut

Tiedostopalvelut kuten tiedostojen säilytys ja jakaminen laitteelta toiselle on tärkeää ohjelmistokehittäjän työssä. Jotta tiedostojen jakaminen olisi helppoa tarvitaan tiedostopalvelu. Myös kalenteripalvelut ovat tärkeitä ohjelmistokehittäjälle, ne auttavat muistamaan esimerkiksi tapaamiset. Kalenteripalvelujen käyttö mahdollistaa kalenterien jakamisen useille eri laitteille. Näiden mainittujen asioiden vuoksi työhön valittiin toteutettavaksi tiedosto- ja kalenteripalvelut.

Ohjelmistot, joita tutkittiin tarjoamaan tiedosto- ja kalenteripalvelut

- Syncthing
- Nextcloud
- Seafile

Syncthing on nimensä mukaan eri laitteiden välillä tiedostojen synkronoinnin tarjoava ohjelmisto. Syncthing ei tarjonnut kaikkea mitä tiedostojen jako -palvelulta haettiin, joka oli tiedostojen tarjoaminen minkä tahansa laitteen välillä, milloin vain web-palvelun kautta. Laitteet, joiden välillä synkronointi tehdään, täytyy tunnistaa toisensa ja se tekee tästä turvallisen ratkaisun eri laitteiden tiedostojen synkronoinnin kannalta.

Nextcloud on laajempi kokonaisuus palveluita. Sen päätoimintona on toimia tiedostopalveluna, joka on Dropboxin kaltainen. Nextcloud mahdollistaa myös palvelun laajentami-

sen Office tyyppiseksi ratkaisuksi sisältäen valmiita lisäosia kuten kalenteripalvelut, sähköposti- ja RSS-ohjelmistot. Nextcloud tarjoaa web-käyttöliittymän sekä ohjelmistot monille eri käyttöjärjestelmille.

Seafile on nopea tiedostojen synkronointi -ohjelmisto, joka tarjoaa myös web-käyttöliittymän. Seafile tarjoaa myös ylimääräisiä lisäosia kuten kalenteripalvelut. Seafile tarjoaa myös ohjelmistot sen käyttämiseksi monille eri käyttöjärjestelmille.

Valinta tapahtui Nextcloudin ja Seafilen välillä. Seafilen etuina olivat sen nopeus ja tehokkuus verrattaen Nextcloudiin. Molemmissa olivat kalenteripalvelut, jotka kuuluivat palveluihin, jotka palvelimelle asennetaan. Seafilessä koettiin huonoksi puoleksi sen tarjoamat eri versiot ohjelmistosta. Community- ja Professional-versiot. Community-versiosta puuttuu monia ominaisuuksia, joita Professional-versio tarjoaa. Nextcloud tarjoaa myös maksullista versiota, mutta se ei rajoita ohjelmiston ominaisuuksia, vaan Enterprise-versio tarjoaa mm. esiasetettuja paketteja sekä parempaa tukea Nextcloudin käyttöön. Tämä ero ohjelmistojen välillä teki lopulta päätöksen käyttää Nextcloud-ohjelmistoa Seafilen sijaan.

2.8 Hakukonepalvelu

Tiedon hakemiseen internetistä käytetään usein hakukonetta. Näistä suosituimmat ovat Google, Bing ja DuckDuckGo. Hakukoneita käytetään ohjelmistokehittäjän työssä paljon eri dokumentaatioiden löytämiseen ja ongelmanratkontaan. Tämän vuoksi hakukonepalvelu valittiin mukaan työhön.

Avoimen ja vapaan lähdekoodin itse ylläpidettäviä hakukonepalveluita löytyy muutamia. Näistä suosituimmat ovat YaCy ja Searx. YaCy ja Searx eroavat toiminnallisuudeltaan toisistaan merkittävästi. YaCy tekee hakukyselyt omaan tietokantaansa, kun taas Searx tekee hakukyselyt määriteltävissä oleviin hakukonepalveluihin kuten Google- tai DuckDuckGo-hakukoneisiin.

YaCy on hajautettu hakukonejärjestelmä. Se on rakennettu vertaisverkkoteknologian perusteita käyttäen. Instanssi voi toimia joko hakurobottina, joka selaa ja indeksoi itsekseen verkon sivustoja muistiinsa tai tehdä indeksointia samalla kun laite, johon ohjelmisto on asennettuna, vierailee verkkosivustoilla. Mikäli hakukoneen käyttäjä on osana YaCy-vertaisverkkoa jakaa tämä hakukone nämä indeksoidut sivunsa muiden vertaistensa kanssa, jolloin näiden hakukoneiden tietokannat kasvavat.

Searx on yksityisyyttä arvostava metahakukone. Se hakee kyselylle usealta eri verkkosivustolta tulokset ja esittää ne käyttäjälle. Searx mahdollistaa monien erityyppisten hakujen suorittamisen esimerkiksi kuva-, kartta- ja tiedostohaut

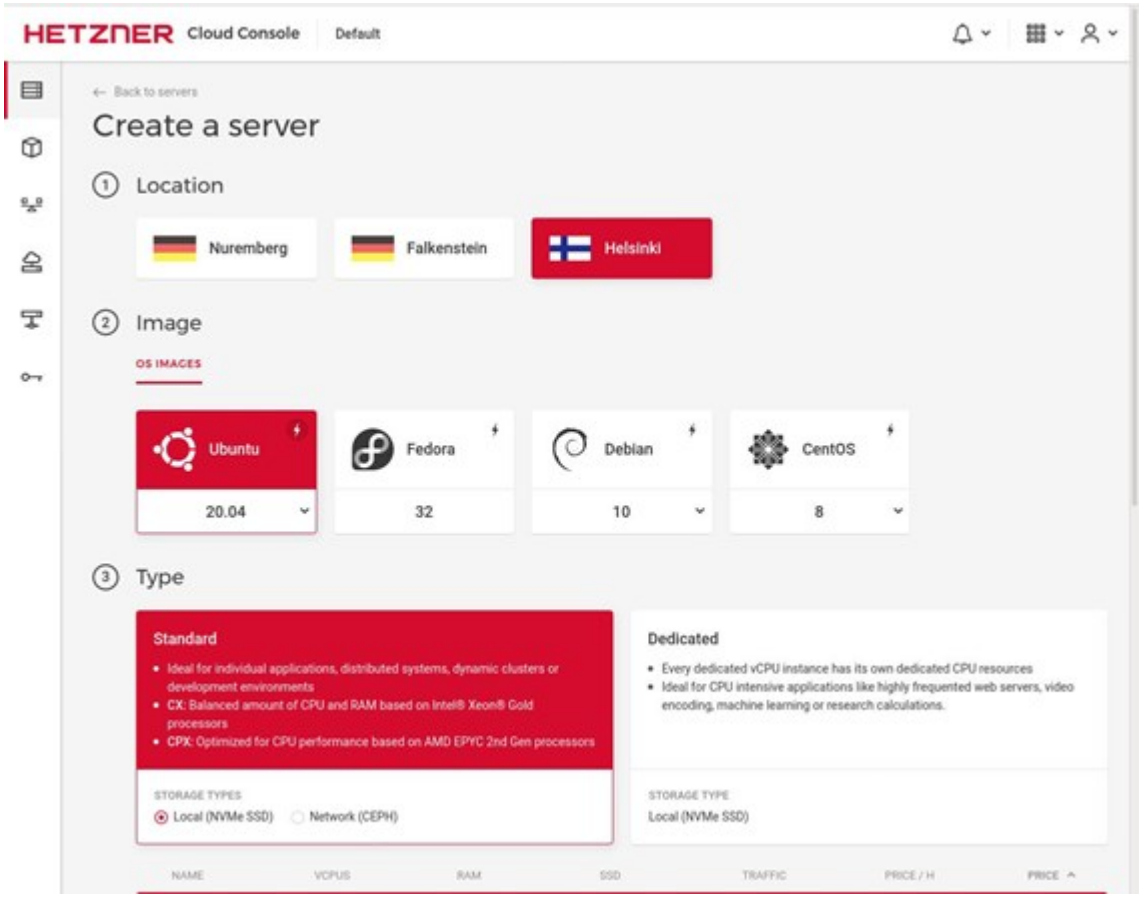
Searxia on mahdollista käyttää joko julkisten Searx-instanssien kautta tai asentaa se itse joko palvelimelle tai henkilökohtaiselle tietokoneelle. Searxin kotisivu sisältää linkkejä julkisiin Searx-instansseihin. Searxin asetuksista on mahdollista asettaa mm. miltä kaikilta verkkosivustoilta hakutuloksia haetaan, millä kielellä Searx hakee tulokset tai millaista teemaa sivusto käyttää.

Vertailu YaCyn ja Searxin välillä on hankalaa niiden eri toimintatapojen vuoksi. Valinta oli kuitenkin helppo, koska YaCyn toiminnallisuus tapahtuu oman tietokantansa kautta, on sen hakukattavuus heikompi kuin Searxin, joka mahdollistaa haut monilta eri sivustoilta kuitenkin tarjoten yksityisyydensuojaa käyttäjälleen. Searx on vapaata lähdekoodia, hyvin konfiguroitavissa ja omaa suhteellisen suuren yhteisön. Näistä syistä päädyttiin palvelimelle asentamaan Searx-hakukoneinstanssi täyttämään hakukonepalvelun asema.

3 TYÖVAIHE

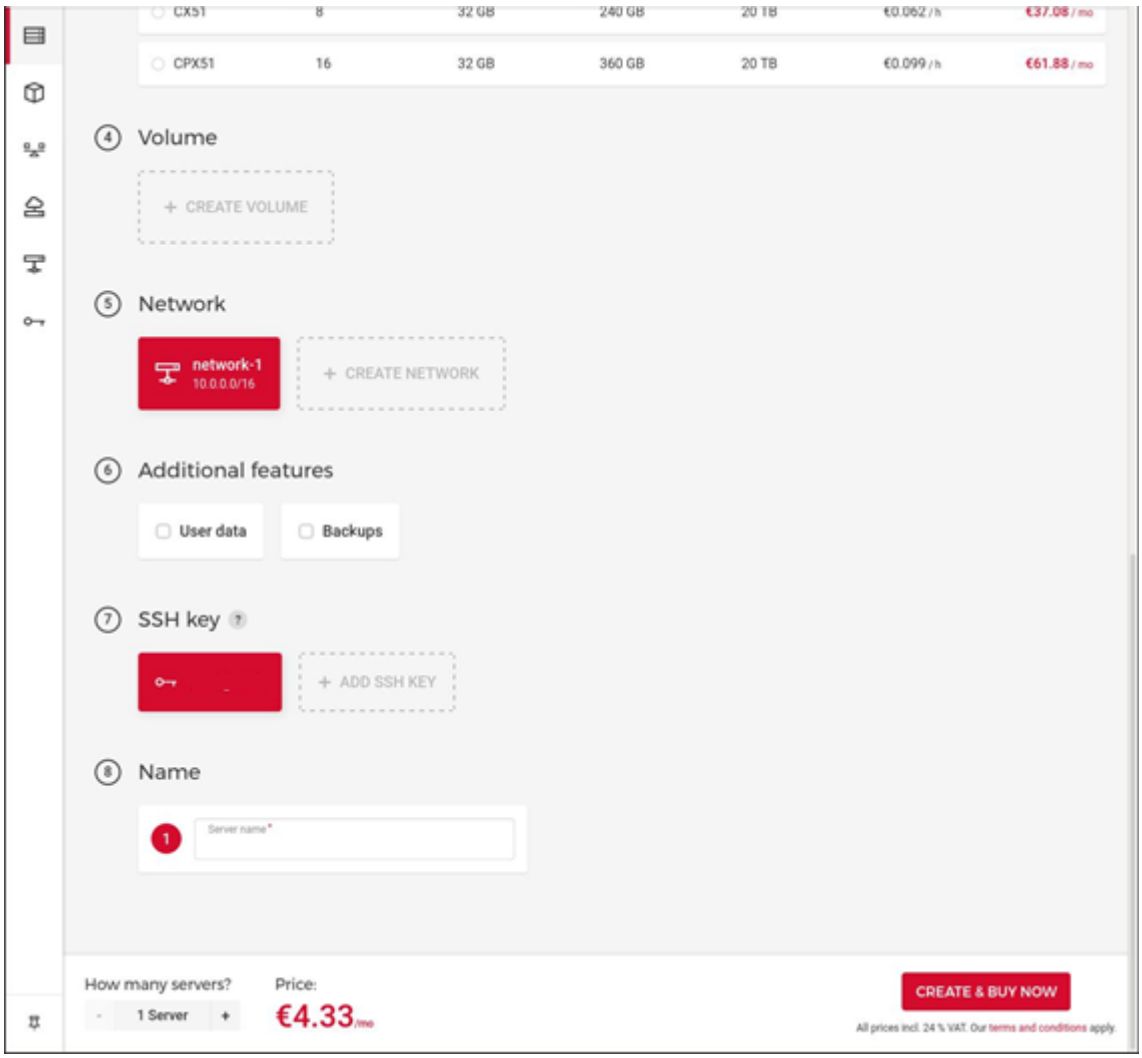
3.1 Palvelimen asennus

Hetzner mahdollistaa helpon palvelimen pystyttämisen. Palvelin on helposti konfiguroitavissa ja asennusvaiheessa on valittavissa mm. käytettävä käyttöjärjestelmä sekä palvelimen sijainti (Kuva 3). Työssä valittiin käytettäväksi serverin käyttöjärjestelmänä Ubuntu 20.04 Linux -jakelua koska sen suuri käyttäjäyhteisö mahdollistaa helpon ongelmanratkonnan. Muita vaihtoehtoja Hetznerillä olisivat olleet mm. Fedora, Debian ja CentOS. Palvelimen sijainti oli mahdollista valita Saksan ja Suomen väliltä. Työssä päädyttiin käyttämään Suomessa sijaitsevaa palvelinta latenssin minimoimiseksi.



Kuva 3. Palvelimen luonti asetukset sijainti ja käyttöjärjestelmä

Hetzner mahdollistaa palvelimelle asennettavan SSH-avaimen määrityksen web-asennuksessa (Kuva 4). SSH-avaimen lisääminen palvelimelle mahdollistaa public-private-key kirjautumisen, jolloin salasanaa ei tarvita. SSH-avain generoitiin paikallisella tietokoneella ja avaimen julkinen osa asetettiin palvelimelle. Asennusvaiheessa varmuuskopioita ei asetettu käytettäväksi, mutta pian työn aloituksen jälkeen ne asetettiin päälle. Varmuuskopiot otetaan joka päivä ja niitä säilytetään seitsemän päivää. Tämä mahdollistaa sen että, vahinkojen tapahtuessa on mahdollista palauttaa palvelin aikaisempaan tilaan.



Kuva 4. Palvelimen luonti -asetukset verkko ja SSH-avain

Palvelimen tietoturva vahvistettiin päivittämällä palvelimen ohjelmistot ja luomalla palvelimelle käyttäjä adduser komennolla. Käyttäjä asetettiin käyttäjäluokkaan sudo, joka mahdollistaa komentojen ajamisen korkeammilla käyttäjäoikeuksilla.

SSH-yhteyden turvallisuuden vahvistamiseksi pääkäyttäjänä kirjautuminen estettiin sekä salasanakirjautuminen poistettiin kokonaisuudessaan. SSH-yhteyden oletusportti muutettiin portista 22 toiseen porttiin.

Käyttäjä, joka yrittää kirjautua palvelimelle tulee omistaa paikallisella laitteellaan SSH-avaimen salaisen osuus. SSH-avain on salasanasuojattu, joten vaikka ulkopuolinen henkilö saisikin SSH-avaimen salaisen osuuden haltuunsa tulee hänen tietää oikea salasana ennen kuin kirjautuminen palvelimelle olisi mahdollista.

Palvelimelle asennettiin Fail2Ban-tietoturvaohjelmisto, joka mahdollistaa suojautumisen väsytyshyökkäyksiltä. Se asettaa porttikieltoon IP-osoitteet, jotka vaikuttavat epäilyttäviltä, kuten sellaiset, jotka yrittävät kirjautua liian useasti väärillä tunnuksilla palvelimelle tai etsivät palvelimelta tunnettuja haavoittavaisuuksia.

Palvelimelle asennettiin Docker ja Docker-compose ohjelmistot. Docker mahdollistaa mm. palvelujen ajamisen omissa eriytyteissä ympäristöissään, näitä ympäristöjä kutsutaan containereiksi eli konteiksi. Docker-compose on ohjelmisto, joka mahdollistaa useita Docker-kontteja sisältävien Docker-applikaatioiden luomisen ja mahdollistaa niiden automaattisen uudelleen käynnistämisen siinä tilanteessa, että Docker-applikaatiot ovat sammuneet. Jokaisessa palvelussa, jonka palvelin tarjoaa, käytetään hyödyksi Docker-ohjelmistoa ja mitään palvelua ei ajeta suoraan palvelimen omassa ympäristössä.

3.2 Verkkotunnuksen hankinta

Sähköpostipalvelimelle järkevän osoitteen asettamiseksi sekä palveluiden helpon löytämisen mahdollistamiseksi palvelimelle hankittiin verkkotunnus.

Verkkotunnus hankittiin Namecheap palvelun kautta, joka tarjoaa myös DNS-palvelun, jonka avulla asetetaan tarvittavat tietueet palveluita varten.

Verkkotunnuksen liittyvät käytännöt palvelimella ovat, että HTTP-palvelut löytyvät osoitteesta https://<palvelu>.<verkkotunnus>.net ja sähköpostiosoitteet ovat muotoa <käyttäjätunnus>@<verkkotunnus>.net

Palveluihin, jotka eivät näy internetselaimessa voidaan ottaa yhteys osoitteella <verkkottunnus>.net:<portti>

Namecheapin DNS-palveluun asetettiin kaksi A-tietuetta (Kuva 5). Toisen palvelimen juurta varten ja toisen sähköpostia varten. Jokaista web-palvelua varten luotiin CNAME-tietue aliverkkotunnuksen käytön mahdollistamiseksi.

Kuva 5. DNS A-tietueet

3.3 Sähköpostipalvelimen asennus

Sähköpostipalveluksi valitsemamme vapaasti konfiguroitavissa oleva Docker-mailserver [11] sisältää tärkeimmät palvelut ja tietoturvatarkistukset mahdollistavat ohjelmistot:

- Postfix on sähköpostin välitysohjelmisto, joka mahdollistaa sähköpostin lähettämisen ja vastaanottamisen
- Dovecot on IMAP- ja POP3-palvelin. Se mahdollistaa sähköpostiohjelmiston kuten esimerkiksi Thunderbird, Gmail tai Mutt ohjelmistojen kautta sähköpostin lukemisen. Näin ollen käyttäjän ei tarvitse aina kirjautua palvelimelle lukemaan sähköposteja. Dovecot ohjelmiston on kehittänyt suomalainen Timo Siirainen
- Amavis on ohjelmisto, joka toimii MTA-ohjelmistojen kuten Postfix ja virus- ja roskapostitarkastaja ohjelmistojen välissä. Amavista voidaan hyödyntää virusten huomaamiseen, viestien karanteeniin asettamiseen, sähköpostin uudelleenohjaamiseen, kuten roskapostin ohjaamiseen roskapostikansioon. Amavis voidaan integroida toimimaan esimerkiksi Spamassassinin ja ClamAvin kanssa.
- Spamassassin on ohjelmisto, joka toimii roskapostitusta vastaan. Se mahdollistaa roskapostin tunnistamisen ja käsittelyn kuten sähköpostiosoitteiden automaattisen estämisen. Sen toiminta perustuu sähköpostiviestin eri osien kuten otsaketietojen testaamiseen.
- ClamAV on vapaan lähdekoodin virusturvaohjelmisto, joka mahdollistaa virusten, troijalaisten ja muiden uhkien huomaamisen. ClamAV on mahdollista integroida Amavikseen.
- OpenDKIM on vapaan lähdekoodin ohjelmisto, joka mahdollistaa DKIM autentikointi järjestelmän käytön. DKIM käyttää julkisen avaimen kryptografiaa tarkis-

taakseen, että sähköposti on lähetetty oikealta sähköpostipalvelimelta ja tunnistamaan sähköpostin sisällön muuttumattomuus ja oikeanmukaisuus. DKIM on yksi tavoista taistella roskapostia vastaan.

- OpenDMARC on ohjelmisto, joka mahdollistaa DMARC autentikointi protokollan käytön. Käytännössä se varmistaa, että sähköposti on todella tullut siitä osoitteesta, josta se väittää tulevansa, ja näin ollen tunnistaa väärennetyistä sähköpostiosoitteista lähetetyt sähköpostit.
- Fail2Ban ohjelmisto auttaa suojautumaan väsytyshyökkäyksiä vastaan. Se asettaa porttikieltoon IP-osoitteet, joiden toiminta palvelimella on epäilyttävää. Epäilyttävää toimintaa on esimerkiksi useat epäonnistuneet kirjautumisyritykset.
- Fetchmail on ohjelmisto, joka hakee sähköpostit IMAP- tai POP3-palvelimelta, tässä yhteydessä Dovecot-palvelimelta.
- Postscreen on ohjelmisto, joka suojaa Postfix SMTP -palvelinta ylikuormittumasta roskapostin takia. Yksi Postscreen prosessi käsittelee useita sisään saapuvia SMTP-yhteyksiä ja valitsee niistä ne, jotka saavat puhua SMTP-palvelimen SMTP-prosessille ja estää bottien yhteydenotot SMTP-prosessiin.
- Postgrey on ohjelmisto, jota käytetään Postfix-palvelimeen tehtävien yhteyksien harmaalistaamiseen. Palvelin estää sähköpostit lähettäjältä hetkeksi aikaa, lähettää sähköpostin takaisin, jossa ilmoitetaan, että se tulisi lähettää tietyn ajan päästä takaisin. Viestin saapuessa uudestaan palvelimelle se vastaanotetaan.
- Let’s Encrypt SSL-sertifikaattien asennukseen ja käyttöön.

Lisäksi paketin mukana tulee asennus-komentosarja, mm. uusien sähköpostikäyttäjien luomiseen ja DKIM avaimen luomiseen käytettävät komennot.

Sähköpostipalvelin käyttää ClamAV ohjelmistoa, joka käyttää paljon muistia ja jotta riittävä muistin käyttö voidaan varmistaa tarvitaan palvelimelle swap-tiedosto. Palvelimelle luotiin 2 gigatavun swap-tiedosto fallocate ohjelmistolla. Swap-tiedosto lisättiin fstab tiedostoon, joka aiheuttaa swap-tiedoston asettamisen swap-tilaksi aina käyttöjärjestelmän käynnistyksen yhteydessä.

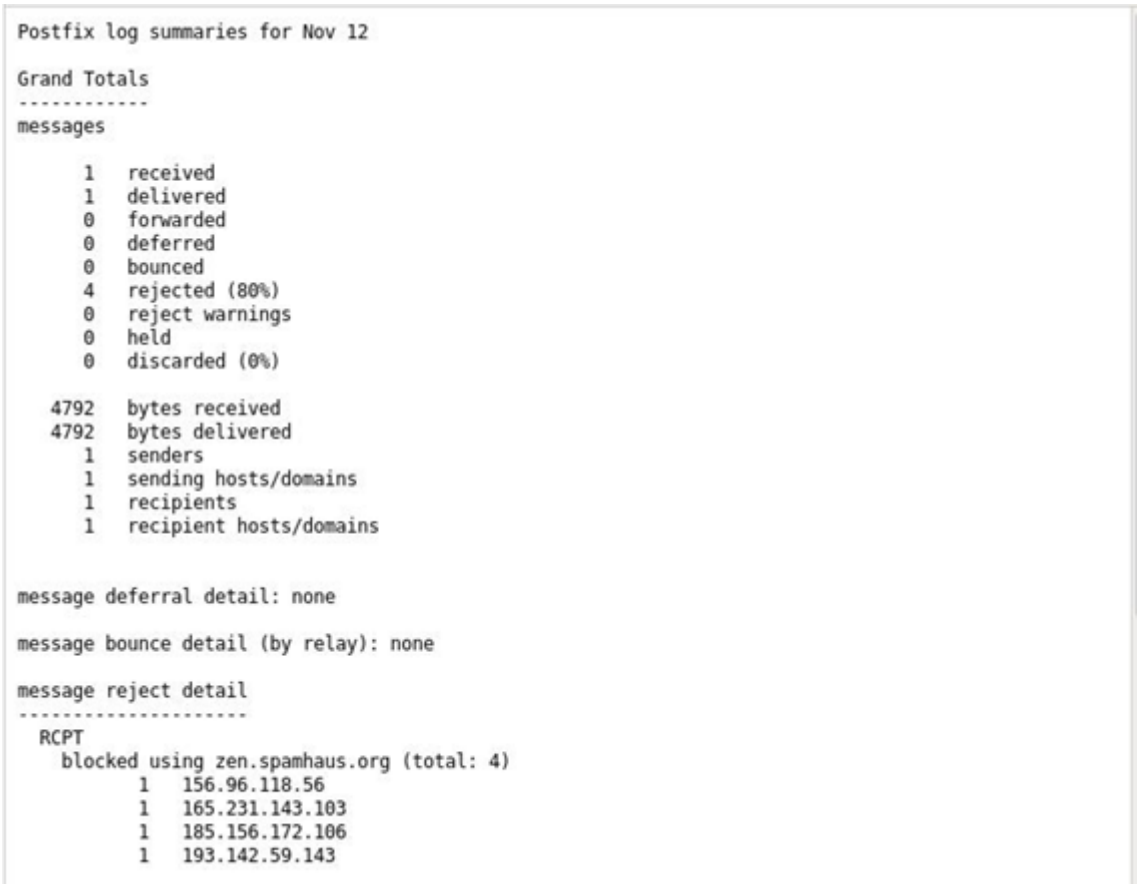
Sähköpostipalvelin tarvitsee lisäksi DNS-palveluun MX-tietueen. MX-tietueen arvoksi asetettiin mail.<verkkotunniste>.net. Tämä tietue määrittelee sähköpostipalvelimen, joka käsittelee sähköpostit tämän verkkotunnuksen alla.

Sähköpostipalvelin asennettiin käyttäen Docker-mailserverin GitHub-sivuilta löytyviä ohjeita. Asennus tehtiin muokkaamalla .env tiedostoa, johon asetettiin isäntänimi, verkkotunnus sekä Docker-kontin tuleva nimi.

Toinen Docker-mailserverin mukana tuleva asetustiedosto on nimeltään env-mailserver, joka sisältää asetukset itse sähköpostipalvelimen ohjelmistoille. Asetustiedoston avulla on mahdollista määritellä mitkä palvelut otetaan käyttöön ohjelmistolle sekä mitkä jätetään käyttämättä.

Asetustiedostossa tätä palvelinta varten asetettiin

- ClamAV lisäämään tietoturvaa sekä tarkistamaan sähköpostit viruksien varalta, harmaalistaus, joka mahdollistaa suojautumisen roskaposteja vastaan, Fail2Ban ohjelmiston estämään toistuvan epäilyttävän toiminnan samasta IP-osoitteesta, Spamassassin ohjelmisto estämään roskapostiliikennettä.
- Estettiin POP3-protokollan käyttö palvelimen kautta IMAP-protokollan ollessa riittävä. IPv6-protokollan käyttö estettiin koska tietyt sähköpostipalvelimet estävät liikenteen IPv6-osoitteista.
- Sähköpostipalvelin asetettiin lähettämään Postfixin ja Logwatchin raportit. Logwatch raportit lähetetään kahden viikon välein ja sisältävät mm. ClamAV- ja Fail2Ban-ohjelmistojen raportit. Postfix raportti lähetetään päivittäin ja se sisältää tiedot, kuinka monta sähköpostia on lähetetty, vastaanotettu sekä mitä Postfixin käyttöyrityksiä on estetty. (Kuva 6)
- Asetettiin SSL:n käyttö päälle, palvelu hyödyntää Let’s encryptin tarjoamia ilmaisia SSL-sertifikaatteja. SSL-sertifikaatit luodaan palvelimen pääympäristössä, jonka jälkeen ne siirretään sähköpostipalvelimen Docker-kontin käyttöön.
- Yhden sähköpostiviestin rajaksi asetettiin 20 MB sisältäen liitteet.
- Asetettiin Sähköpostin lähetys tapahtuvaksi vain sisäisen verkon kautta, tämä estää sen, että joku voisi käyttää palvelinta avoimena sähköpostivälittäjänä. Avoimena sähköpostivälittäjänä toimivat sähköpostipalvelimet saattavat joutua hyökkäyksen kohteeksi, jossa roskapostittaja lähettää roskapostia avoimen sähköpostipalvelimen kautta. Tämän tapahtuessa sähköpostipalvelin yleensä asetetaan estolistalle roskapostittajalistojen toimesta. Palvelimemme hyödyntää näitä roskapostittajalistoja Spamassassin ohjelmiston avulla, jolloin niistä tuleva liikenne osataan estää suoraan



```
Postfix log summaries for Nov 12

Grand Totals
------------
messages

      1   received
      1   delivered
      0   forwarded
      0   deferred
      0   bounced
      4   rejected (80%)
      0   reject warnings
      0   held
      0   discarded (0%)

   4792   bytes received
   4792   bytes delivered
      1   senders
      1   sending hosts/domains
      1   recipients
      1   recipient hosts/domains


message deferral detail: none

message bounce detail (by relay): none

message reject detail
---------------------
  RCPT
    blocked using zen.spamhaus.org (total: 4)
            1   156.96.118.56
            1   165.231.143.103
            1   185.156.172.106
            1   193.142.59.143
```

Kuva 6. Sähköpostiin saapuva Postfix-raportti

Palvelimen asetuksien määrittelemisen jälkeen palvelun käynnistäminen tapahtuu käyttäen Docker-compose-ohjelmistoa ja ajamalla komento Docker-compose up -d. Parametrilla -d ilmoitetaan, että applikaatio halutaan ajaa taustaohjelmistona. Komennon ajamisen jälkeen Docker luo kontin, jossa asetetut ohjelmistot alkavat pyörimään.

Käyttäjien luominen sähköpostipalveluun tapahtuu setup.sh-komentosarjan avulla. Sähköpostijärjestelmään luotiin käyttäjätili <käyttäjänimi>@<verkkotunnus>.net sekä postmaster-tili postmaster@<verkkotunnus>.net

Sähköpostiviestien lähettäjän tarkistaminen tapahtuu käyttäen DNS palvelimelta löytyviä TXT-tietueita. tällaisia ovat mm. DMARC, DKIM ja SPF-varmenteet.

DKIM-avaimen luonti tapahtuu automaattisesti käyttäen setup.sh-komentosarjan komentoa config dkim. Tämän jälkeen on mahdollista asettaa palvelimelle DKIM TXT -tietue.

DKIM TXT -tietue on varotoimi sähköpostiosoitteen väärentämistä varten. DKIM perustuu salattuun varmenteeseen, joka tulee palvelimelta. DKIM mahdollistaa lähettäjän varmistamisen tekemällä kyselyn nimipalvelimelta. DKIM:n avulla voidaan varmistaa myös, että viestiä ei ole muokattu sen jälkeen, kun viesti lähtenyt lähettäjän sähköpostipalvelimelta. Luotu DKIM-avain kopioitiin ja palvelimelle luotiin uusi TXT-tietue, jonka host-osioon asetettiin mail._domainkey ja jonka arvoksi asetetaan “v=DKIM1; h=sha256; k=rsa; p=<komento sarjan luoma avain>”

SPF TXT -tietue sisältää tiedon siitä ketkä saavat lähettää verkko-osoitteen kautta sähköpostia. Asetettiin DNS-palveluun uusi TXT-tietue, jonka host-osioon asetettiin @ ja arvoksi asetettiin mail.<verkkotunnus>.net arvo kenttään asetettiin ”v=spf1 mx ~all”, joka tarkoittaa, että kaikki MX-tietueessa määritellyt verkko-osoitteet voivat lähettää tämän verkko-osoitteen puolesta sähköpostia.

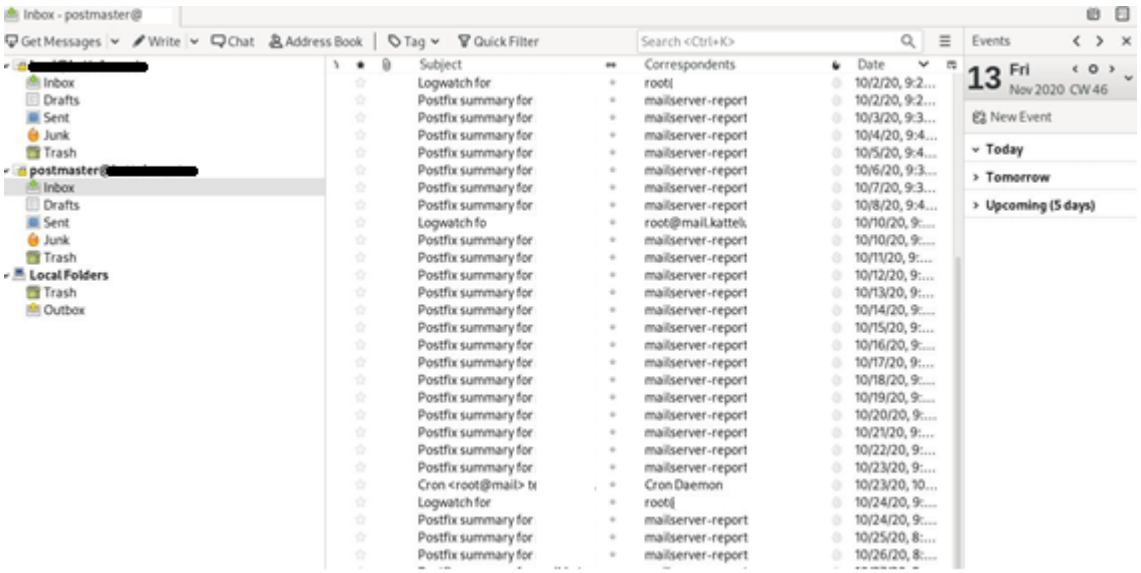
DMARC TXT -tietue liittyy suoraan aiemmin asetettuihin tietueisiin. Se kertoo palvelimelle, jolla lähetetään sähköposti viestin käyttävän SPF- ja DKIM-varmenteita, sekä mitä viesteille tehdään, mikäli varmennus epäonnistuu. DMARC-tietue määriteltiin palvelimelle näin

"v=DMARC1;p=quarantine;rua=mailto:dmarc.report@<verkkotunnus>.net;ruf=mailto:dmarc.report@<verkkotunnus>.net; fo=0; adkim=r; aspf=r; pct=100; rf=afrf; ri=86400; sp=quarantine"

- p: Määrittää että mikäli DMARC-testi epäonnistuu, asetetaan viesti karanteeniin. rua määrittää osoitteen johon palaute DMARC:sta lähetetään.
- rua: Määrittää sähköpostiosoitteen, johon lähetetään korkean tason kootut DMARC-raportit.
- ruf: Määrittää osoitteen, minne ilmoitetaan viestikohtaiset DMARC-epäonnistumisraportit.
- fo 0: määrittää että epäonnistumisraportti luodaan, mikäli molemmat SPF- ja DKIM-tarkistukset epäonnistuvat.
- adkim: Määrittää DKIM-tietueen ja viestin otsakkeesta löytyvän osoitteiden tarkistus tarkkuuden. Adkim määritettiin käyttämään relaxed-asetusta, jolloin mikäli DKIM-tietue määrittää mail.<verkkotunnus>.net mutta sähköposti tulee <verkkotunnus>.net verkko-osoitteesta, menee tarkistus silti läpi. Mikäli käytettäisiin strict-asetusta tarkistus epäonnistuisi.

- aspf: Määrittää SPF:n tarkastuksen tarkkuuden. Aspf asetettiin relaxed-asetukselle. Tarkistus on periaatteessa sama kuin adkim määrittelyssä otsakkeessa määritellyn verkko-osoitteen ja SPF-tietueessa määriteltyjen verkko-osoitteiden vertailussa.
- pct: Määrittää prosentuaalisesti viestien määrän joka vastaanottajien kuuluu suodattaa DMARC-tarkastuksesta epäonnistuneista viesteistä. Käytettäessä arvoa 100 kaikki viestit suodatetaan. Mikäli arvo olisi 50 vain puolet viesteistä suodatettaisiin.
- rf: Määrittää verkko-osoitteen omistajalle lähetettävän raportin formaatin. Tällä hetkellä atrf on ainoa asetus, joka on mahdollista asettaa rf kenttään. Atrf tulee sanoista Authentication Failure Reporting Format.
- ri: Määrittää kuinka usein voidaan raportti lähettää rua:ssa määriteltyyn osoitteeseen. Se määritellään sekunteina, jolloin arvo 86400 vastaa 24 tuntia. DMARC ei tällöin lähetä koottua raporttia kuin kerran vuorokaudessa.
- sp : Määrittää mitä kaikille viesteille, jotka saapuvat tämän verkko-osoitteen aliverkko-osoitteesta, jota ei ole määritetty DMARC:ssa tapahtuu. Quarantine-asetuksella kaikki määrittämättömistä aliverkko-osoitteista saapuvat sähköpostit asetetaan karanteeniin.

Sähköpostipalvelun käyttö tapahtuu yhdistämällä palveluun jonkin sähköpostiohjelmiston kautta esim. Thunderbird tai Mutt. Yhteyden muodostus testattiin käyttäjätilille käyttäen Thunderbird-sähköpostiohjelmistoa (Kuva 7).



Kuva 7. Thunderbird-sähköpostiohjelmisto

Yhteyden testaamisen ja viestin lähettämisen ja vastaanottamisen jälkeen sähköpostipalvelinta testattiin palveluilla, jotka varmistavat että

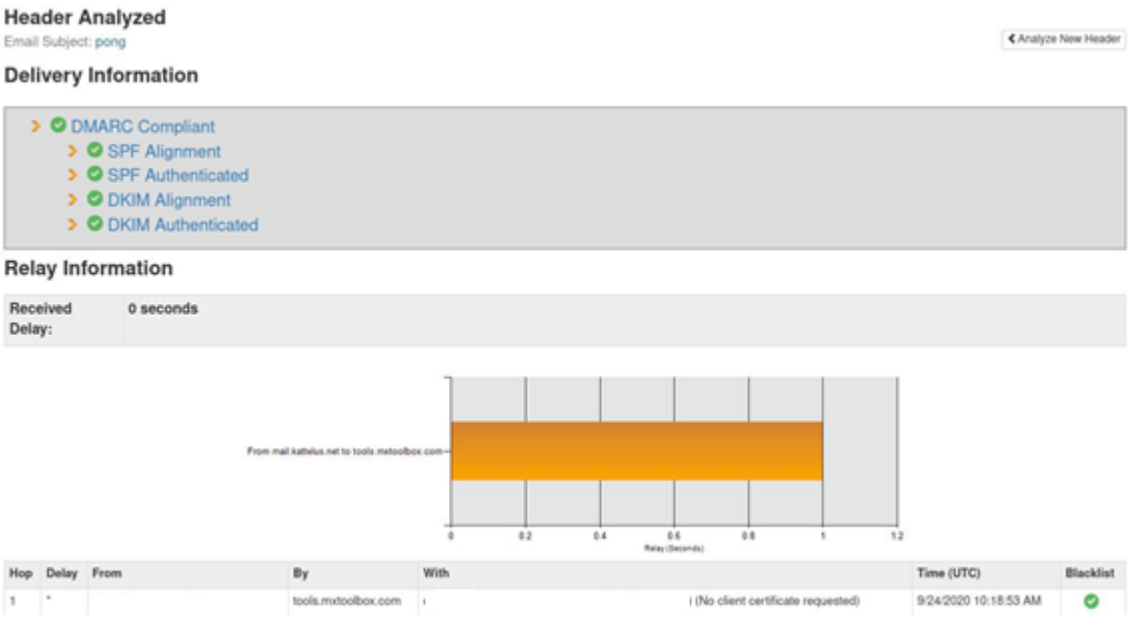
- Palvelin ei toimi avoimena välityspalvelimena.
- Tietueiden olevan oikeat DNS palvelimella.
- Sähköpostipalvelin pystyy lähettämään sähköpostin oikein.

Avoimen välityspalvelimen testaus toteutettiin käyttäen MXToolBox-sivuston tarjoamalla Test Email Server SMTP -työkalulla, jonka raportista nähdään testien tulokset (Kuva 8). [12]

	Test	Result
✓	SMTP Reverse DNS Mismatch	OK - resolves to mail. .net
✓	SMTP Valid Hostname	OK - Reverse DNS is a valid Hostname
✓	SMTP Banner Check	OK - Reverse DNS matches SMTP Banner
✓	SMTP TLS	OK - Supports TLS.
✓	SMTP Connection Time	0.639 seconds - Good on Connection time
✓	SMTP Open Relay	OK - Not an open relay.
✓	SMTP Transaction Time	1.414 seconds - Good on Transaction Time

Kuva 8. MXToolBox Test Email Server raportti.

Sähköpostin lähetyksen ja tietueiden tarkistus toteutettiin MXToolBox-sivuston tarjoamalla Email Deliverability työkalulla, jonka raportista nähdään DMARC-sopivuus sekä lähetyksen onnistuminen (Kuva 9). [12]



Kuva 9. MXToolBox Email Deliverability raportti

3.4 Versionhallintapalvelin

Versionhallinta-palvelimeksi valittiin Git. Toteutuksen aikana tutkittiin valmiita Docker-pohjaisia Git-ratkaisuja mutta valmiiden ratkaisujen testaamisen jälkeen päädyttiin toteuttamaan itse Docker-resepti Git-palvelimen asentamiseksi.

Työhön määriteltiin Git-palvelimen toiminta

- Palvelimella on mahdollisuus säilöä yksityisiä ja julkisia tietovarastoja.
- Käyttäjät varmennetaan palvelimelle SSH-avainta hyödyntämällä.
- Palvelimella voidaan käyttää Git- ja SSH-protokollaa tietovarastojen kopiointiin ja päivittämiseen.

Palvelimen Dockerfileä kirjoittaessa seurattiin Pro Git -kirjan ohjeita. [13]

Git-palvelimen asennuskansio sisältää tiedostot

- docker-compose.yaml Docker-compose -asetustiedoston.
- Dockerfile -Docker-reseptin
- init.sh-komentosarjan, joka ajetaan, kun palvelu käynnistetään
- sshd_config-asetustiedoston.

Ja kansiot

- private-repos
- public-ids
- public-repos.

Kun Dockerfile ajetaan, se:

- Asentaa tarvittavat ohjelmistot uuteen Docker-ympäristöön.
- Alustaa SSH-palvelun, estää SSH-käyttäjiltä interaktiivisen komentotulkin käytön.
- Asettaa kansiosta löytyvän SSH-asetustiedoston käytettäväksi Docker-kontissa.
- Paljastaa Docker-sovellukselta portin 22 jota käytetään SSH-yhteyden luontiin.
- Paljastaa Docker-sovellukselta portin 9418 jota Git-protokolla käyttää.
- Suorittaa init.sh komentosarjan.

Komentosarja init.sh

- Kopioi kaikki public-ids kansiossa olevat julkiset SSH-avaimet sallittujen avainten tiedostoon Docker-kontissa.
- Käynnistää Git daemon -ohjelmiston, joka mahdollistaa Git-protokollan käytön.
- Käynnistää SSH-palvelun, joka mahdollistaa SSH-yhteyden luomisen palvelimeen.

Docker-compose tiedostossa määritellään palvelimen portit, joista liikenne ohjataan Docker-konttiin portteihin ja, että kansiot private-repos, public-ids ja public-repos ovat sidottuna Docker-konttiin siten että muutokset, jotka tehdään esimerkiksi public-repos kansioon näkyvät Docker-kontissa sekä toisinpäin.

Git-palvelimen käynnistys tapahtuu komennolla docker-compose -d.

Palvelimen toimintaa testattiin ensin yrittämällä kirjautua palvelimelle SSH-yhteyden kautta komennolla ssh -i <SSH-avain> git@<verkkotunniste>.net:<portti>. Testin yhteydessä saatiin varmistus siitä, että komentotulkin käyttö on estetty.

Git-versionhallinnan toiminnallisuuden testaamiseksi paikallisella laitteella luotiin bare Git -tietovarasto eli tietovarasto, joka on .git päätteisen tiedoston sisällä. Tietovaraston kopioimisen mahdollistamiseksi Git-protokollan avulla luotiin tietovarastoon git-daemon-export-ok niminen tiedosto. Tietovarasto siirrettiin palvelimelle kansioon public-repos. Tämän jälkeen kloonattiin tietovarasto palvelimelta käyttäen SSH-protokollaa ja Git-protokollaa (Kuva 10). Git-tietovaraston muuttamista ja päivitystä testattiin muokkaamalla kloonatun Git-tietovaraston tiedostoja ja puskemalla se takaisin palvelimelle. Testeistä todettiin palvelimen toimivan toivotusti.

Kuva 10. Git-tietovaraston kopiointi Git- ja SSH-protokollalla

3.5 Käänteinen HTTP-välityspalvelin ja SSL-sertifikaatit

Palvelin esittää palveluitaan verkkosivustoilla käyttäjille HTTP-palvelimien kautta. Koska HTTP-palvelimien oletus portit ovat 80 ja 443 joita vain yksi Docker-sovellus voi käyttää kerrallaan ja palvelimella pyörii useita Docker-sovelluksia yhdenaikaisesti, tarvitaan jotain, joka ohjaa liikenteen porttien 80 ja 443 kautta oikeaan palvelimella pyörivään Docker-konttiin. Tarvitsemme väliin käänteisen välityspalvelimen. Käänteinen välityspalvelin ottaa yhteyden vastaan käyttäjältä ja ohjaa yhteyden oikeaan Docker-konttiin palvelimella.

Tämän käänteisen välityspalvelimen lisäksi tarvitaan HTTPS-liikennettä varten SSL-sertifikaatit jokaiselle HTTP-palvelulle.

Käänteisen HTTP-välityspalvelimeen ja SSL-sertifikaattien käyttöönottoon hyödynnettiin kahta suosittua valmista Docker-kuvaa nginx-proxya ja letsencrypt-nginx-proxy-companionia. Nginx-proxy mahdollistaa automaattisen käänteisen välityspalvelimen käytön niille Docker-konteille, jotka ovat yhdistetty samaan verkkoon kuin nginx-proxyn Docker-

kontti. Näiden yhdistäminen tapahtuu asettamalla ennalta määriteltyjä ympäristömuuttujia sen Docker-konttiin docker-compose.yaml -asetustiedostoon, joka halutaan yhdistää välityspalvelimeen (Kuva 11).

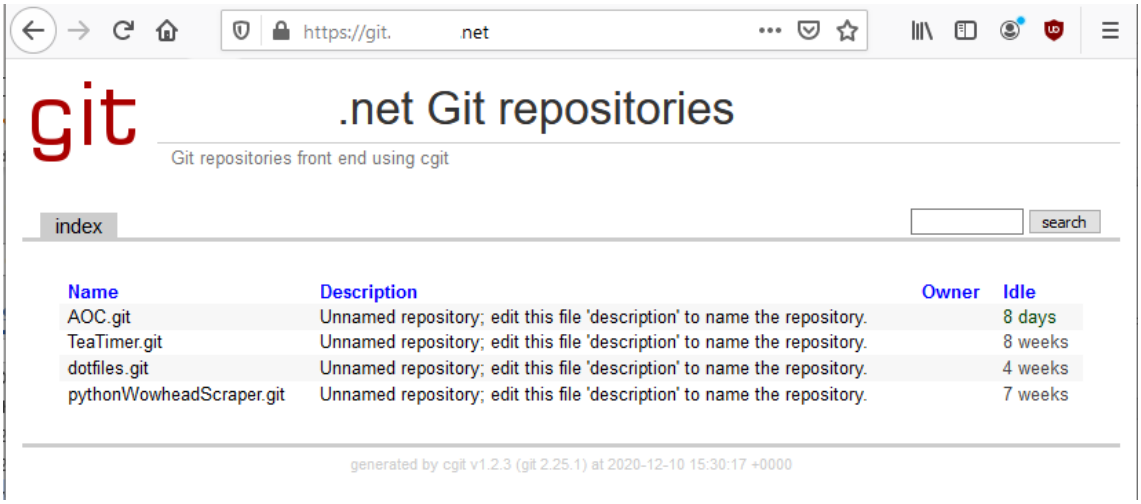
Letsencrypt-nginx-proxy-companion mahdollistaa automaattisten SSL-sertifikaattien luomisen Let's Encrypt -palvelua hyödyntäen. Se asettaa Docker-kontille SSL-sertifikaatin sekä tarvittaessa uusii sertifikaatit. Docker-kontille sertifikaattien luonti tapahtuu asettamalla Docker-sovelluksen docker-compose-tiedostoon ennalta määriteltyjä ympäristömuuttujia (Kuva 11).

Kuva 11. Nginx-proxya ja letsencrypt-nginx-proxy-companionia käyttävän Docker-kontin ympäristömuuttujia

3.6 Git-palvelimen frontend

Git-palvelimen julkisten tietovarastojen esittäminen tapahtuu Git-frontendin kautta. Tämän mahdollistamiseksi palvelimelle asennettiin Cgit-ohjelmisto, joka muuttaa sille annetut Git-tietovarastot verkkosivulla esitettävään muotoon (Kuva 12 ja Kuva 13).

Kuva 12. Cgit-ohjelmiston esitys Git-tietovaraston tiedostorakenteesta



Kuva 13. Git-palvelun cgit-frontend

Cgit ohjelmistolle löytyi valmiita Docker-kuvia, mutta nämä olivat vanhoja sekä heikosti optimoituja, joten työssä luotiin Docker-resepti itse.

Ennen Dockerfilen luomista valmisteltiin asetustiedostot Cgitille ja Apache HTTP -palvelimelle. Cgit:n asetustiedostossa määriteltiin mm. Git-kloonausosoite sekä polku, josta löytyvät Git-tietovarastot Docker-kontissa. Apachen asetustiedostossa määriteltiin mm. polku johon Cgit on asennettu Docker-kontin sisällä.

Dockerfile perustuu minimalistiseen Alpine Linux -jakeluun. Kun Dockerfile ajetaan se

- Asentaa Cgit ja Apache2-ohjelmistot sekä niiden riippuvuudet.
- Kopioi valmiiksi luodut asetustiedostot oikeisiin tiedostopolkuihinsa.
- Ajaa init.sh komentosarjan, joka käynnistää Apache2 HTTP-palvelimen.

Docker-Compose tiedostoon määriteltiin, että Cgitin käyttämät Git-tietovarastot haetaan palvelimelle asennetun Git-palvelimen public-repos kansiosta sekä määriteltiin palvelimen kuuluvan saamaan verkkoon nginx-proxyn Docker-kontin kanssa sekä ympäristömuuttujat, joiden avulla nginx-proxy ja letsencrpyt-nginx-proxy-companion osaavat osoittaa tälle oikean aliverkko-osoitteen ja SSL-sertifikaatit.

Palvelun osoitteeksi asetettiin git.<verkkotunnus>.net, osoitteelle asetettiin oma CNAME-tietue Namecheapin DNS-palveluun.

3.7 Hakukonepalvelun asennus

Työhön valittu metahakukoneohjelmisto Searx tarjoaa virallisen Docker-kuvan hakukoneinstanssin asentamiseen. Tämän Docker-kuvan hyödyntämiseksi luotiin docker-compose.yaml tiedosto Searxille.

Searxin asetuksia varten luotiin searx-settings niminen kansio. Kansiosta luotiin yhteys Docker-kontin /etc/searx kansioon, joka mahdollistaa helpon globaalien asetuksien muuttamisen sekä säilyttämisen.

Asetukset mahdollistavat mm.

- Turvahaun asettamisen päälle tai pois
- Hakukoneiden määrittelyn
- Uusien hakukoneiden lisäämisen
- Automaattisen täydennyksen oletuskielen.

Searx asetetaan samaan sisäiseen verkkoon kuin nginx-proxy-välityspalvelin. Searxille asetettiin ympäristömuuttujiksi nginx-proxyn sekä letsencrypt-nginx-proxy-companionin vaatima virtuaalinen portti, verkko-osoite sekä sähköpostiosoite.

Docker-kontti käynnistettiin komennolla docker-compose up -d

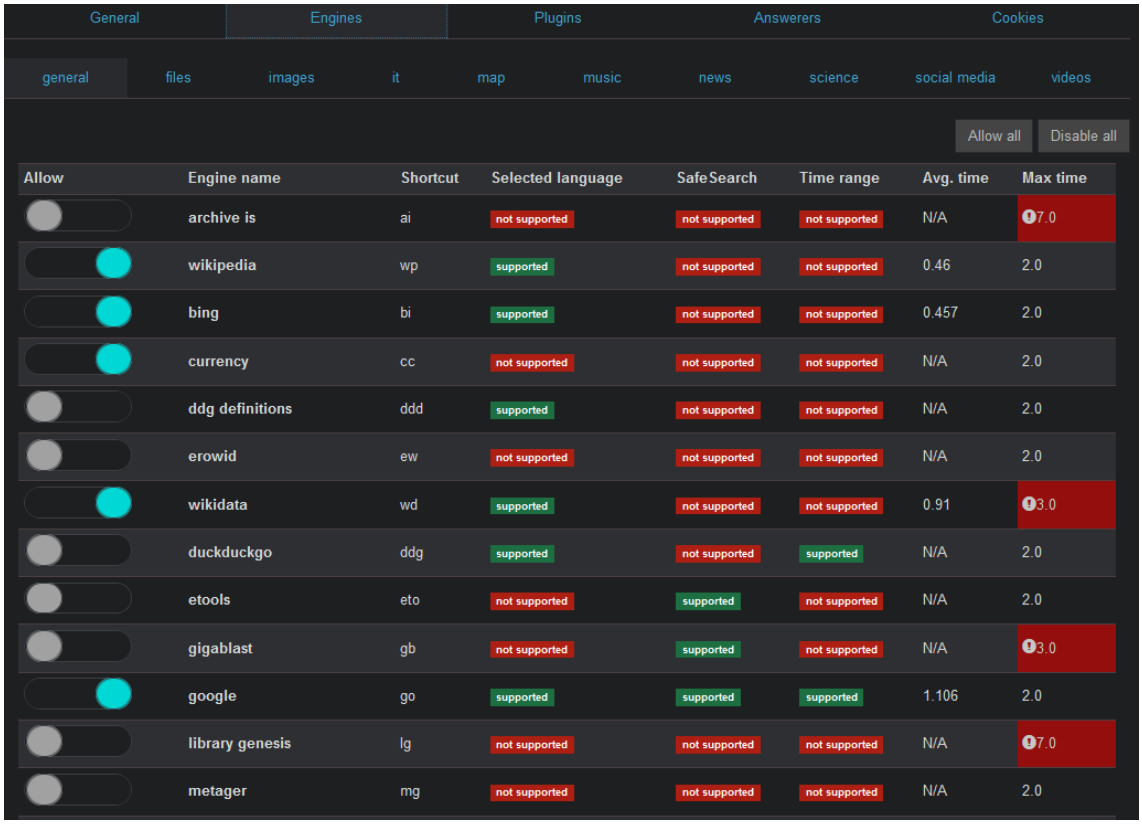
Docker-kontin käynnistämisen jälkeen searx-settings kansioon ilmestyvät settings.yml ja uwsgi.ini asetustiedostot.

Globaalista settings.yml-asetustiedostosta asetettiin

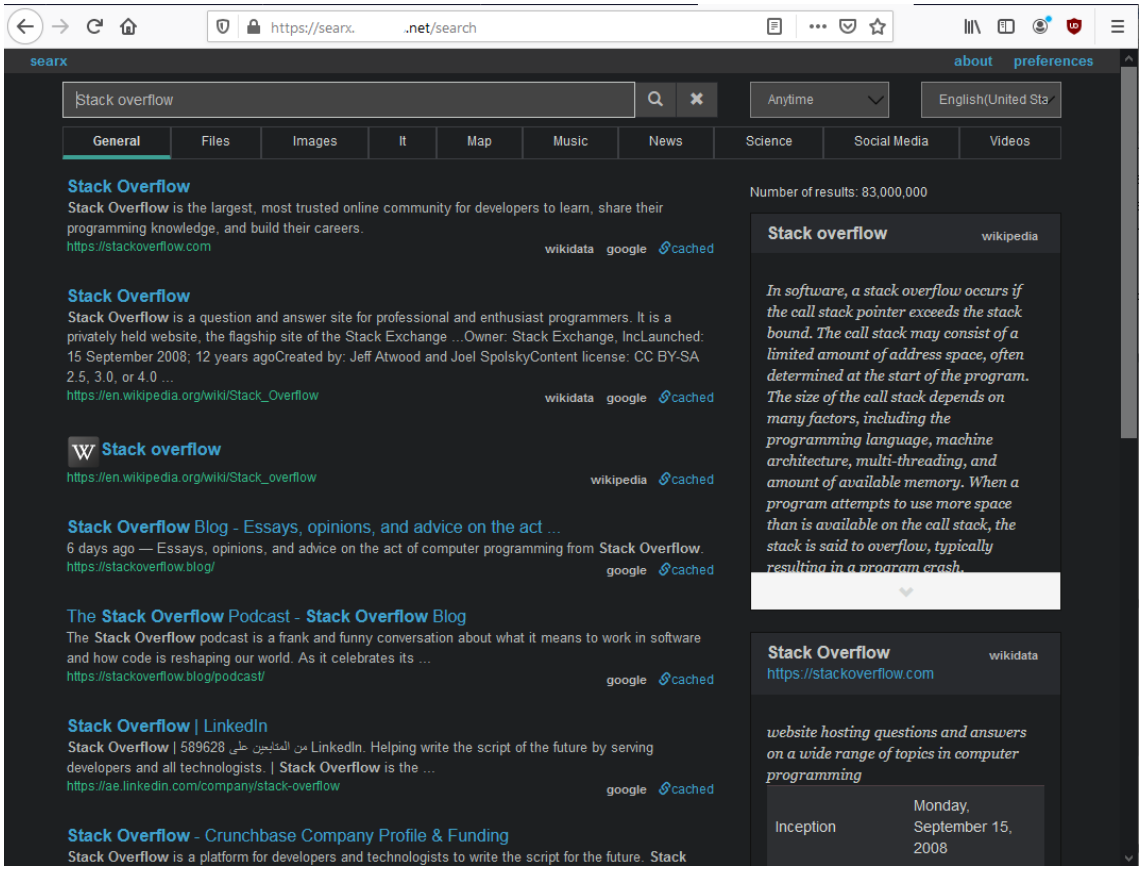
- Sivusto käyttämään tummaa Oscar teemaa.
- Sivuston oletus hakukieli.

Searx mahdollistaa useiden eri hakukonepalveluiden käytön hakutulosten luomiseen. Nämä hakukoneet ovat asetettavissa Searxin web-käyttöliittymästä (Kuva 14).

Hakukoneinstanssin toiminnallisuus testattiin käymällä verkkosivustolla sekä tekemällä haku (Kuva 15).



Kuva 14. Searxin hakukone vaihtoehtoja



Kuva 15. Searxin testihaku

3.8 Pi-holen ja OpenVPN:n asennus

Pi-hole asennettiin hyödyntäen kahta valmista Docker-kuvaa. Käytetyt Docker-kuvat olivat PiHolen virallinen Docker-kuva sekä Kyle Mannan OpenVPN-palvelimen Docker-kuva [14].

Pi-holea varten luotiin oma tietoverkko nimeltä Pi-hole, jotta OpenVPN-palvelin voidaan asettaa pyörimään samaan verkkoon Pi-holen kanssa. Pi-holelle asetettiin staattinen IP-osoite, jotta OpenVPN osaa käyttää Pi-holea DNS-palvelimenaan.

OpenVPN-palvelin toteutettiin palvelimella siten että yhteen OpenVPN-instanssiin voi olla vain yksi laite kerrallaan VPN-yhteydessä. Työssä luotiin puhelimelle VPN-yhtey-

den tarjoava Docker-kontti. Docker-kontin nimeksi annettiin openvpn-phone. Nimeämistavalla tavoiteltiin helppoa Docker-konttien hallintaa useiden eri laitteiden VPN-yhteyksiä varten.

OpenVPN-palvelimen asetuksien luontiin käytettiin OpenVPN:n Docker-kuvan tähän tarkoitukseen tarjoamia työkaluja. OpenVPN:n asetuksissa asetettiin yhteysosoitteeksi vpn.<verkkotunnus>.net ja tälle luotiin uusi CNAME-tietue. OpenVPN-yhteyden DNS-palvelimeksi asetettiin Docker-kontissa pyörivän Pi-holen staattinen IP-osoite.

Pi-holen asennuksessa hyödynnettiin nginx-proxy:n Docker-kuvaa, jonka avulla asetettiin Pi-holen verkkosivusto osoitteeseen pihole.<verkkotunnus>.net. Pi-holelle asetettiin ylemmän hierarkian DNS-palvelimiksi 1.1.1.1 ja 1.0.0.1. Nämä ovat Cloudflaren tarjoamia SSL-suojattuja DNS-palvelimia, jotka mahdollistavat paremman yksityisyyden DNS-kyselyitä tehtäessä. [15]

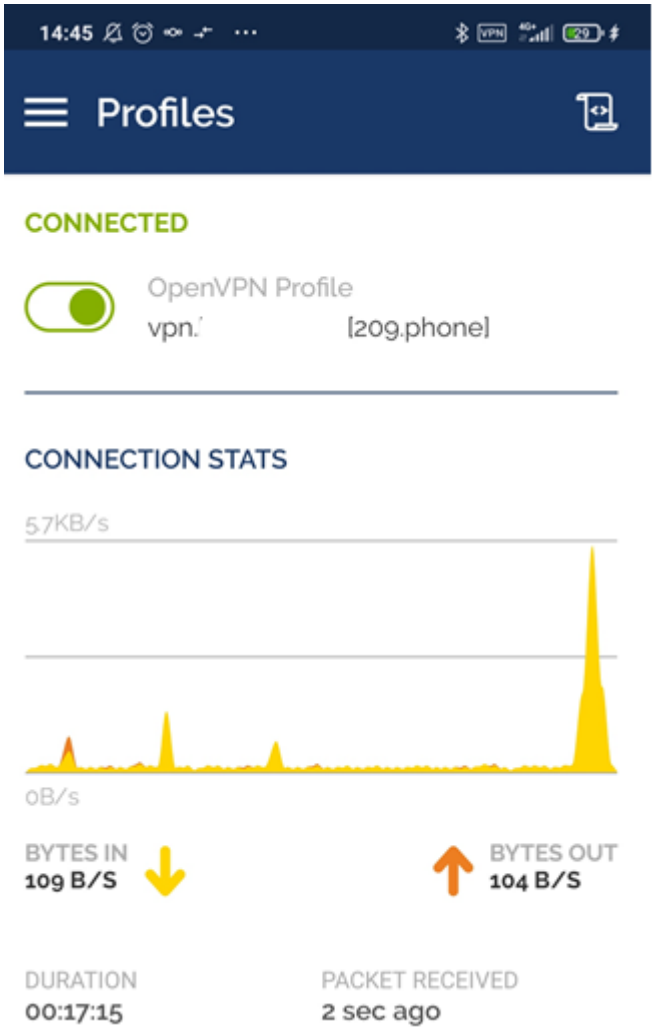
Pi-holen verkkosivulle asetettiin vahva pääkäyttäjän salasana, jonka avulla on mahdollista kirjautua pääkäyttäjä-paneeliin mm. monitoroimaan yhdistettyjen laitteiden sekä tehtyjen DNS-hakujen määrää.

Pi-hole Docker-kontti yhdistettiin kahteen eri tietoverkkoon: nginx-proxy:n verkkoon, jolla mahdollistettiin nginx-proxy:n käyttö sekä Pi-holen verkkoon mahdollistamaan DNS-kyselyiden tarjoaminen VPN-yhteyden muodostaneille laitteille.

Pi-holen verkkosivustoa varten luotiin pitkä 50 merkin pituinen salasana, jotta estetään ulkopuolisten pääsy sivustolle. Salasana asetettiin hyödyntäen nginx-proxy:n tarjoamaa tapaa käyttää autorisointiin htpasswd tiedostoja.

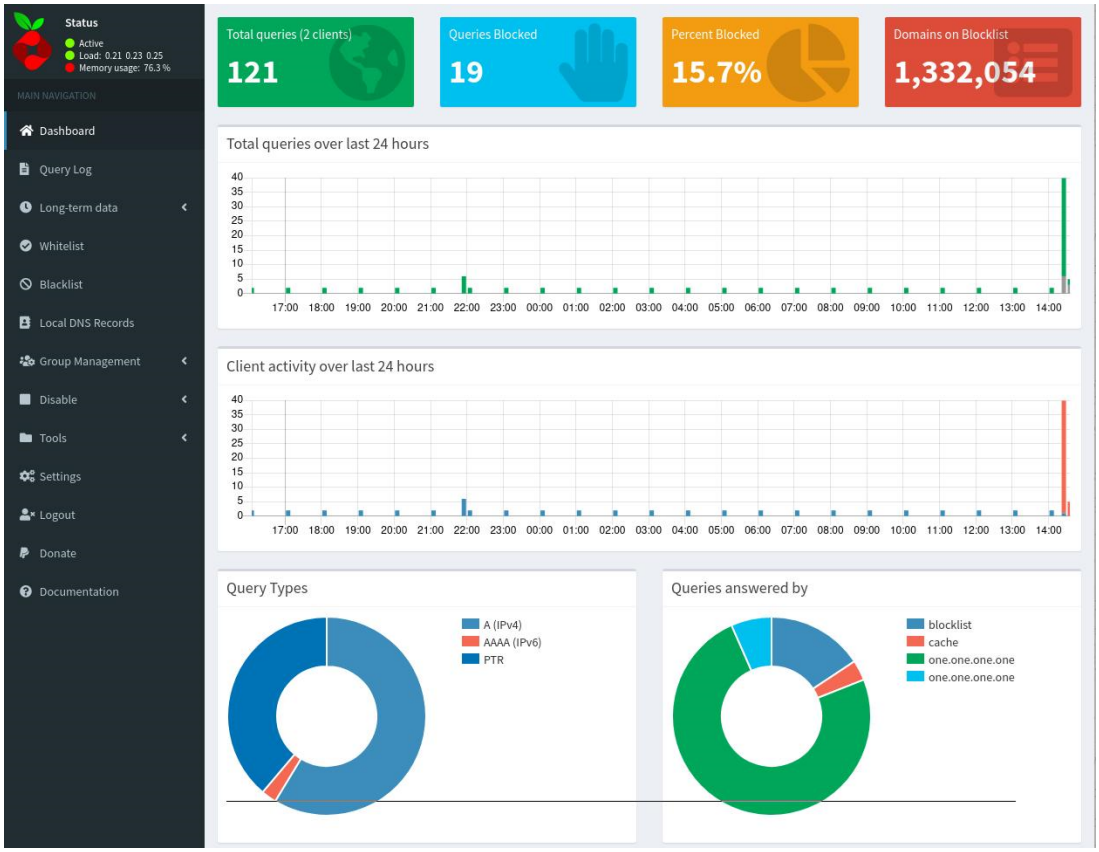
Pi-holen käynnistämisen jälkeen kirjauduttiin verkkosivustolle ja asetettiin estolistat käyttöön. Tämän jälkeen VPN-yhteyttä hyödyntävän laitteet ovat turvassa estolistan tunnistamilta haitallisilta yhteyksiltä.

Jotta VPN-yhteyden käyttö olisi mahdollista tuotiin käynnistetyltä OpenVPN:n Docker-kontilta ohjelmiston luoma OpenVPN-profiili. Yhdistäminen palvelimelle on mahdollista vain käyttämällä tätä tuotua profiilia. Profiilista vaihdettiin oletusportti 1194:n tilalle Docker-compose-tiedostossa asetettu portti. Profiili siirrettiin puhelimelle ja yhteyttä testattiin OpenVPN Client -sovelluksen avulla (Kuva 16).



Kuva 16. Puhelin yhdistettynä OpenVPN-palvelimeen

Puhelimen yhdistämisen jälkeen kirjauduttiin palvelimelle varmistamaan, että Pi-hole tunnistaa laitteen. Pi-holen web-käyttöliittymästä (Kuva 17) pystyttiin todentamaan puhelimen yhteys palvelimeen.



Kuva 17. Pi-holen web-käyttöliittymä

3.9 Tiedosto- ja kalenteripalvelun asentaminen

Tiedosto- ja kalenteripalvelut tarjoava Nextcloud-ohjelmistolla on virallinen Docker-kuva asennusta varten. Tämän Docker-kuvan lisäksi Nextcloud tarvitsee tietokannan mm. käyttäjätilien säilytystä varten. Työssä käytettiin tietokantana MariaDB:tä, josta on tarjolla virallinen Docker-kuva.

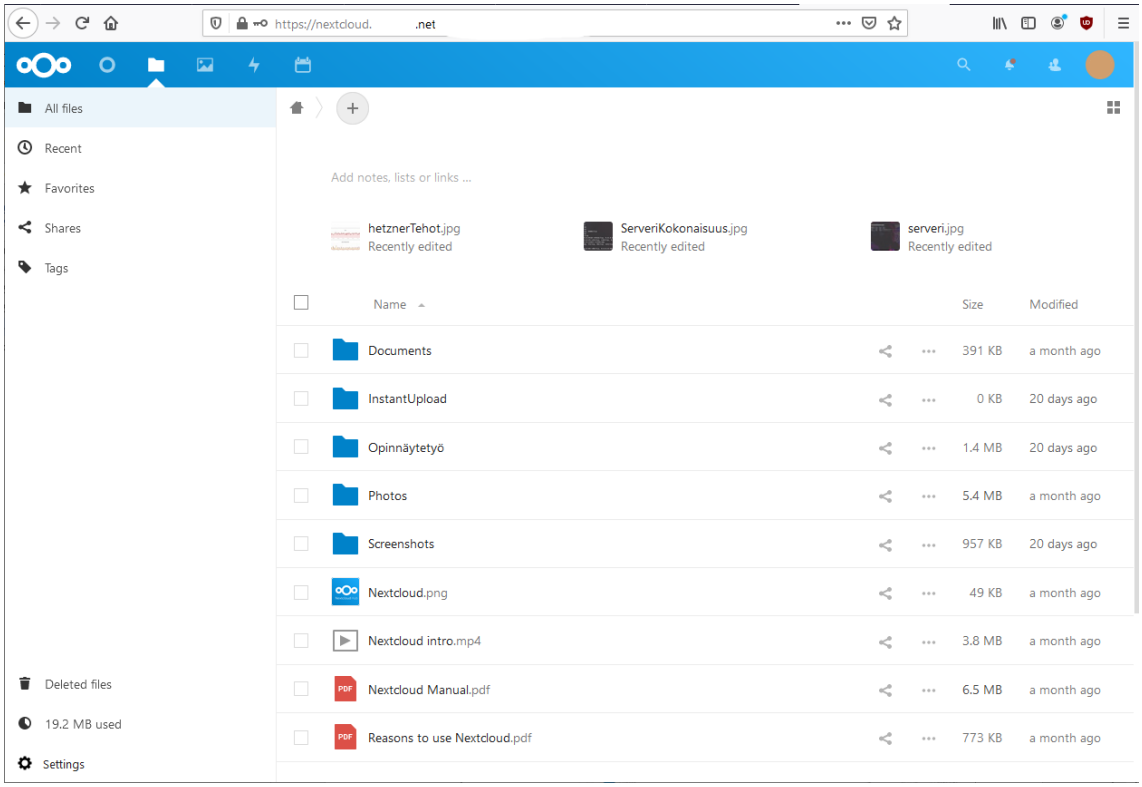
Jotta tietokannan tiedot eivät katoaisi luotiin Docker-kontille yksi Docker-volume. Erilliseen db.env-asetustiedostoon asetettiin tietokannan salasana, nimi sekä tietokannan pääkäyttäjän nimi.

Nextcloudin Docker-kontille luotiin Docker-volume, jotta käyttäjien tiedostot säilyvät vaikka Docker-kontti hajoaisi. Docker-compose-tiedostoon asetettiin, että Nextcloudin Docker-kontin käynnistäminen vaatii, että MariaDB:n Docker-kontti on käynnissä. Nginx-Proxyn avulla asetettiin Nextcloud toimimaan verkko-osoitteessa nextcloud.<verkkotunniste>.net

Docker-kontit käynnistettiin ja tämän jälkeen siirryttiin Nextcloudin verkkosivustolle, jossa asennusprosessi jatkui. Asennusprosessissa luotiin Nextcloudille pääkäyttäjä sekä määritettiin yhteys tietokantaan. Asennuksen jälkeen luotiin yksi normaalin tason käyttäjä, jotta tiedostojen jakaminen onnistuu laitteiden välillä ilman että on tarvetta käyttää pääkäyttäjätiliä.

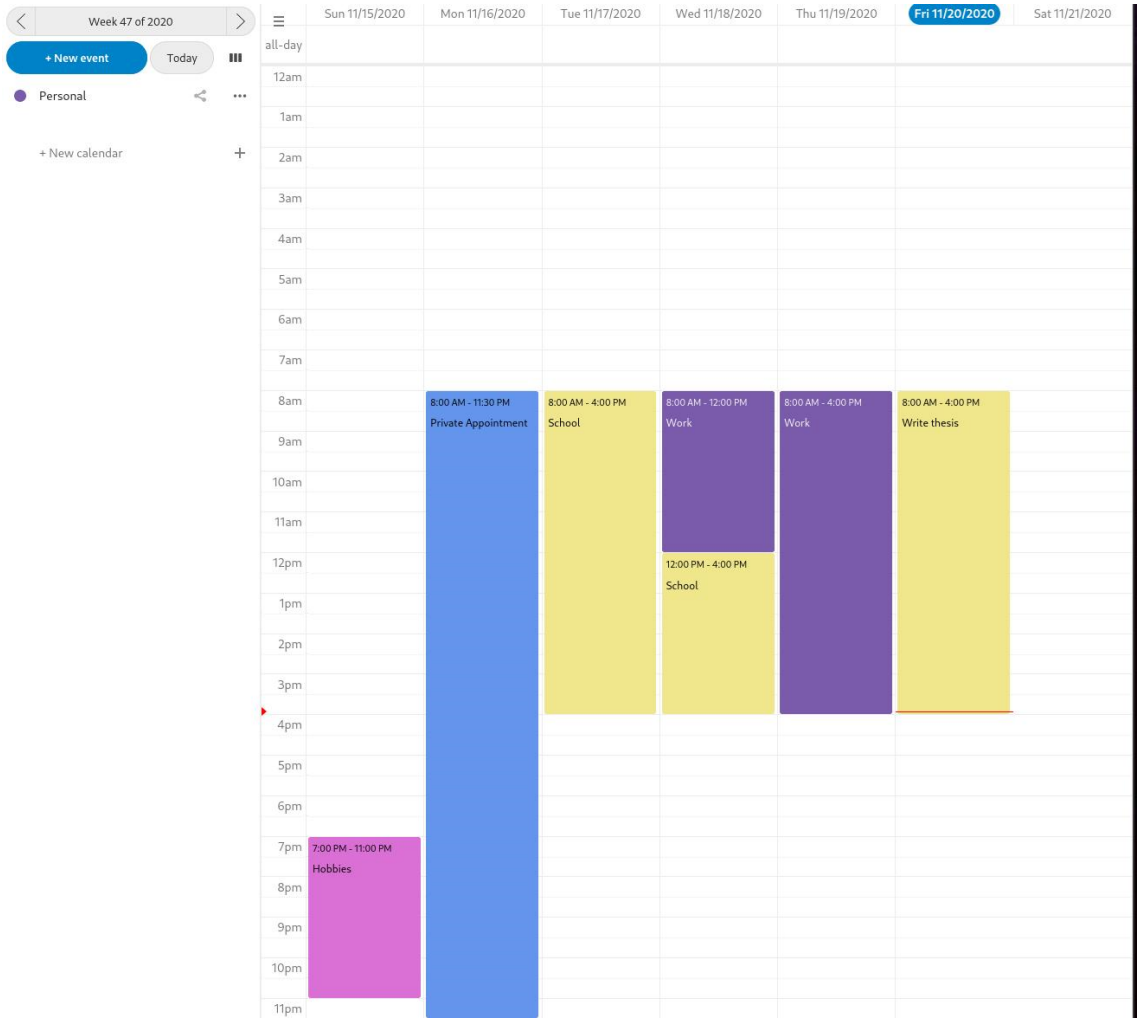
Nextcloudin asennuksessa huomattiin, että yhteyden kulkiessa välityspalvelimen kautta asiakasohjelmistot eivät pystyneet yhdistämään Nextcloud palveluun. Tämä ongelma korjattiin asettamalla Nextcloudin ympäristömuuttujiin OVERWRITEPROTOCOL=https ympäristömuuttuja, joka ohjaa liikenteen kulkemaan HTTPS-protokollan kautta.

Nextcloudin toimintaa testattiin lähettämällä web-käyttöliittymän (Kuva 18) kautta palvelimelle tiedosto ja lataamalla se.



Kuva 18. Nextcloudin web-käyttöliittymä

Nextcloud-palveluun asennettiin Calendar-lisäosa. Tämä mahdollistaa kalenteripalvelut Nextcloudin web-käyttöliittymässä. Calendar-lisäosan avulla luodut kalenterit on mahdollista ladata ICS-tiedostona, joka mahdollistaa kalenterien käyttämisen muissakin sovelluksissa kuin pelkästään Nextcloudissa. Kalenteripalveluita testattiin luomalla web-käyttöliittymässä viikkokalenteri (Kuva 19)

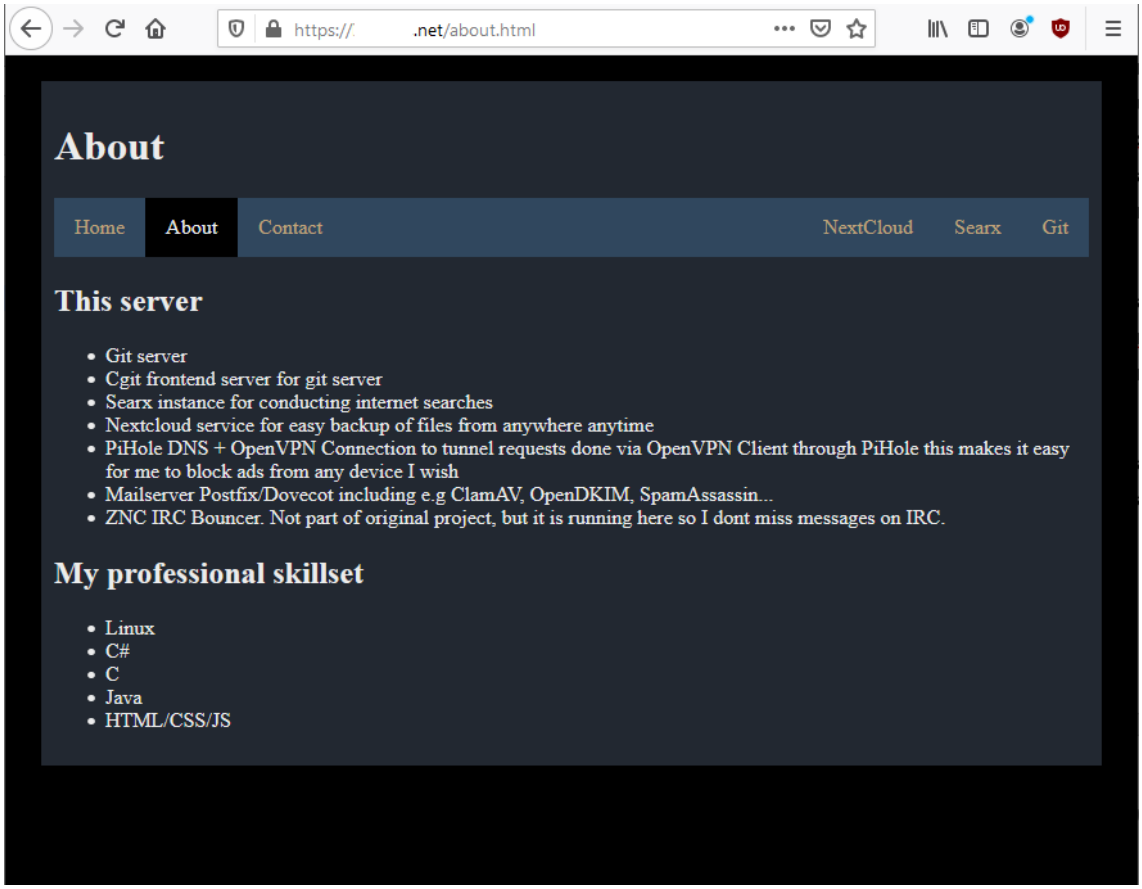


Kuva 19. Nextcloudin Calendar-lisäosa

3.10 Henkilökohtainen verkkosivu

Henkilökohtainen verkkosivu toteutettiin HTML- ja CSS-teknologioilla. Verkkosivulle toteutettiin etusivu-, yhteystiedot- sekä lisätietoja-sivu. Etusivu sisältää tietoa itse verkkosivustosta ja sen sisällöstä. Yhteystiedot sivulle listattiin tavat, joilla on mahdollista ottaa yhteys sivuston omistajaan. Lisätietoja sivustolle listattiin palvelut, jotka palvelimelle on asennettu sekä palvelimen omistajan tietotekniset taidot. Navigointipalkkiin lisättiin näiden sivulinkkien lisäksi linkit Nextcloud-, Searx- ja Git-palveluihin.

Sivustolle määriteltiin yksinkertainen CSS-tyylitiedosto, joka sisältää sivuston asettelun sekä värimaailman (Kuva 20.). Eri sivustot toteutettiin puhtaina HTML-tiedostoina. Tämä mahdollistaa nopeat verkkosivut tarjoten kaiken tarpeellisen tämän tason verkkosivulle.



Kuva 20. Henkilökohtainen verkkosivu

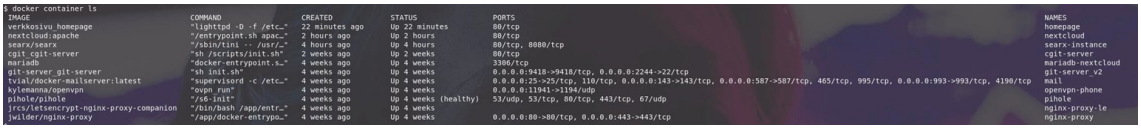
Docker-resepti pohjautuu Alpine Linux -jakeluun. Reseptissä konttiin asennetaan ja käynnistetään Lighttpd HTTP-palvelinohjelmisto.

Docker-compose-tiedostossa määriteltiin tarvittavat ympäristömuuttujat nginx-proxy:n käyttämiseksi sekä luodaan yhteys HTML- ja CSS-tiedostot sisältävän kansion sekä Docker-kontin /var/www/localhost/htdocs kansion välille. Tällä mahdollistetaan sivuston nopea muokkaus ja päivitys palvelimelle.

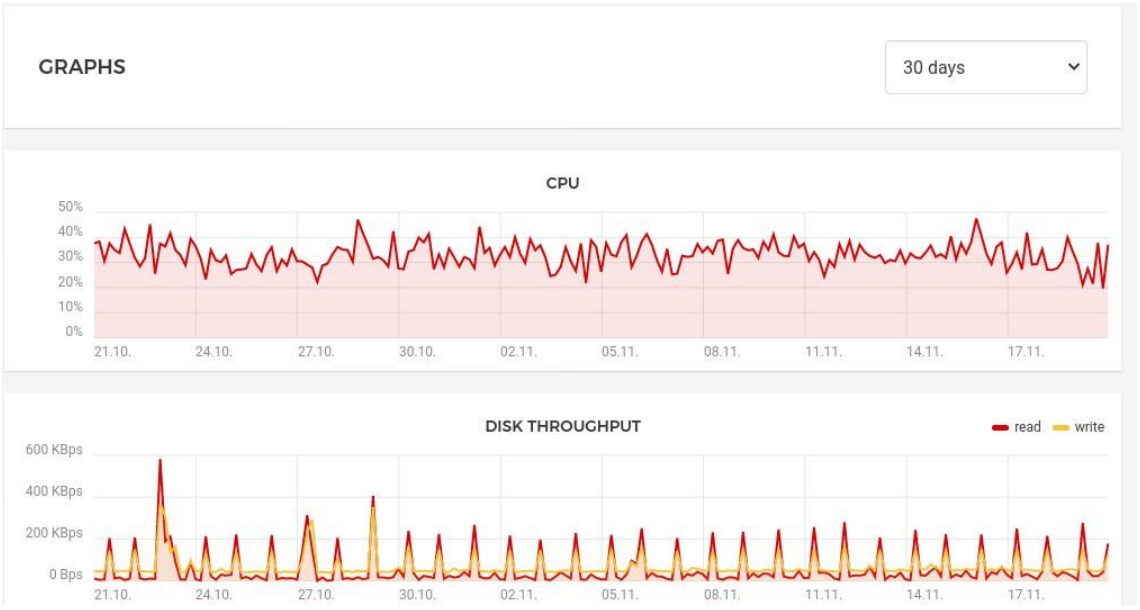
3.11 Palvelimen toiminnan seuranta

Palveluiden oltua palvelimella kuukauden verran (Kuva 21) tarkistettiin palvelimen prosessorin käyttöaste ja muistinkäyttö viimeisen kuukauden ajalta. Tähän hyödynnettiin

Hetznerin tarjoamia palvelimen seuranta työkaluja (Kuva 22). Huomattiin että palvelimen korkein prosessorin käyttöaste on ollut 50 % ja pääsääntöisesti noin 30 %. Todettiin, että palvelin on ollut riittävä tähän käyttötarkoitukseen ja kestää hyvin usean palvelun pyörittämisen. Palvelin ei ole kaatunut pystyttämisen jälkeen kertaakaan ja palvelut ovat toimineet odotetusti ja stabiilisti.



Kuva 21. Docker-kontit palvelimella



Kuva 22. Palvelimen käyttöasteet 30 päivän ajalta

Käytön aikana olen ollut erittäin tyytyväinen Git- ja Cgit-ohjelmistojen toimintaan. Git-tietovarastot ovat toimineet odotetusti ja niiden käyttäminen on ollut sujuvaa. Sähköpostipalvelin on toiminut hyvin ja luotettavasti. Searx-hakukoneen toiminta on ollut hakutulosten kannalta oikein sujuvaa, ainoastaan haun suorituksen kesto on häirinnyt käyttökokemusta. Searx-hakukoneella saattaa mennä 3–4 sekuntia hakupyynnöstä hakutulosten palauttamiseen. Puhelimeni on ollut kytkettynä pitkiä aikoja VPN-yhteydellä palvelimeen ja näin ollen DNS-palveluun. Puhelimen verkkoyhteys on toiminut luotettavasti, normaalisti ja verkkosisältö on suodatettu Pi-holen kautta odotetusti. Tiedostopalvelu on

toiminut luotettavasti ja sitä on käytetty tiedostojen synkronointiin tietokoneen ja puhelimen välillä. Tiedostojen synkronointi on tapahtunut odotetusti ja tiedostojen siirto laitteiden välillä on ollut vaivatonta.

4 LOPUKSI

Opinnäytetyön tavoitteena oli tutkia ja toteuttaa ohjelmistokehittäjälle tärkeät internetin peruspalvelut tarjoava palvelin. Valitut palvelut ovat ohjelmistokehittäjälle tarpeellisia ja hyödyllisiä. Toteutettavat palvelut olivat sähköposti-, versionhallinta-, tiedosto-, kalenteri-, hakukone-, VPN- ja yksityiset DNS-palvelut, jonka lisäksi toteutettiin henkilökohtaiset verkkosivut. Ennen toteutusta tutkittiin ohjelmistot, joilla nämä palvelut voitaisiin toteuttaa, sekä valittiin ne määriteltyjen kriteereiden mukaisesti. Kaikki palvelut eriytettiin toisistaan hyödyntämällä Docker-ohjelmiston tarjoamaa säiliöintiä.

Ennen opinnäytetyön aloittamista kokemusta palveluiden tarjoamisesta palvelimella oli vain vähän. Dockerin käyttökokemusta palvelinympäristössä ei ollut lainkaan. Asennettujen ohjelmistoiden ylläpidosta palvelimella ei ollut lainkaan kokemusta ja käyttökokemusta oli vain Git-palvelimen osalta.

Opinnäytetyön aikana opin palvelimen tietoturvasta ja suojaamisesta paljon. Dockerin käyttö on nykyään luontevaa ja ymmärrän paremmin, miten Docker toimii, sekä erilaisia tapoja käyttää Dockeria. Opin lisää, miten sähköposti toimii, mitä kaikkia protokollia sähköpostiliikenteeseen liittyy ja minkälaisia tietoturvaratkaisuja sähköpostiliikennettä varten on luotu.

Opinnäytetyön aikana luotu palvelin sekä sen palvelut ovat olleet aktiivisessa käytössä työn jälkeen. Palvelin on toiminut luotettavasti kuukauden ajan eikä palveluissa ole ilmennyt ongelmia.

LÄHTEET

[1] What Ports Are Blocked? Vultr, 2015. Saatavissa (Viitattu 27.11.2020) https://www.vultr.com/docs/what-ports-are-blocked

[2] Number of sent and received e-mails per day worldwide from 2017 to 2024? Statista, 2020. Saatavissa (Viitattu 27.11.2020) https://www.statista.com/statistics/456500/daily-number-of-e-mails-worldwide

[3] Global spam volume as percentage of total e-mail traffic from January 2014 to March 2020, by month, Statista, 2020. Saatavissa (Viitattu 27.11.2020) https://www.statista.com/statistics/420391/spam-email-traffic-share

[4] Version Control Systems Popularity in 2016? RhodeCode, 2017. Saatavissa (Viitattu 4.12.2020) https://rhodecode.com/insights/version-control-systems-2016

[5] Compare Repositories, 2020. Saatavissa (Viitattu 4.12.2020) https://www.openhub.net/repositories/compare

[6] Devops Platform Delivered As A Single Application, GitLab 2020. Saatavissa (Viitattu 4.12.2020) https://about.gitlab.com

[7] Gitweb Documentation, Git, 2020. Saatavissa (Viitattu 4.12.2020) https://git-scm.com/docs/gitweb

[8] Gitlist - An Elegant And Modern Git Repository Viewer, GitList, 2020. Saatavissa (Viitattu 4.12.2020) https://gitlist.org

[9] Cgit - A Hyperfast Web Frontend For Git Repositories Written In C, Cgit, 2020. Saatavissa (Viitattu 4.12.2020) https://git.zx2c4.com/cgit/about/

[10] Pi-hole – Network-wide protection, Pi-hole, 2020. Saatavissa (Viitattu 4.12.2020) https://pi-hole.net/

[11] Docker Mailserver, Docker Mailserver, 2020. Saatavissa (Viitattu 4.12.2020) https://github.com/tomav/docker-mailserver

[12] MX Lookup Tool - Check Your DNS MX Records Online, MxToolbox, 2020. Saatavissa (Viitattu 4.12.2020) https://mxtoolbox.com

[13] Scott C, Ben S. Pro Git: APress: 2014

[14] OpenVPN for Docker, Docker Hub, 2020. Saatavissa (Viitattu 10.12.2020) https://hub.docker.com/r/kylemanna/openvpn/

[15] What is 1.1.1.1?, CloudFlare, 2020. Saatavissa (Viitattu 10.12.2020) https://www.cloudflare.com/learning/dns/what-is-1.1.1.1