



samk



Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

MATIAS MATTILA

Ympäristön rakentaminen verkkosivuston perustamiseen palvelimella

TIETOJENKÄSITTELYN TUTKINTO-OHJELMA
2023

TIIVISTELMÄ

Mattila, Matias:

Ympäristön rakentaminen verkkosivuston perustamiseen palvelimella

Opinnäytetyö, AMK

Tietojenkäsittelyn tutkinto-ohjelma

Huhtikuu 2023

Sivumäärä: 49

Tässä opinnäytetyössä käytiin läpi verkkosivuympäristön perustaminen vuokrattuun VPS eli Virtual Private Server -alustaan. Työ aloitettiin selvittämällä sopiva rekisterinpitäjä domainille, sekä Virtual Private Server palveluntarjoaja. Tutustuttiin niiden valintaprosessiin ja käsiteltiin asioita, joita kannattaa ottaa huomioon palvelujen valinnassa.

Työssä käytiin myös läpi, miten vuokrattuun VPS-palvelimeen otettiin etäyhteys eri käyttöjärjestelmillä. Palvelimelle toteutettiin helposti laajennettavissa oleva ympäristö verkkosivulle. Palvelimen, sekä VPS-hallintapaneeli konfiguroitiin valmiiksi verkkosivulle. Palvelimeen konfiguroitiin myös tiedostojen siirtomahdollisuus.

Työn loppuvaiheessa tehtiin oma verkkosivu, joka siirrettiin palvelimelle. Verkkosivu tehtiin käyttäen ReactJS-kirjastoa, joka mahdollisti komponenttipohjaisen kehittämisen. Ennen verkkosivun tekemistä tutkittiin Reactia ja sen kilpailijaa Vueta. Verkkosivun kehittäminen toteutettiin käyttäen Reactia ja Vite-työkalua. Kehittäminen tapahtui omalla laitteella ja valmiit tiedostot siirrettiin palvelimelle.

Tuloksena saatiin toimiva ja joustava alusta verkkosivulle, johon voidaan asentaa muitakin palveluja. Palvelimeen on mahdollista ottaa etäyhteys omalta laitteelta, sekä lähettää sinne tiedostoja, kuten verkkosivu. Työssä asiat käydään läpi lukijalle niin, että ne olisivat helposti ymmärrettävissä.

Avainsanat: domain, VPS-palvelin, DNS, ajoympäristö, SSL-sertifikaatti

Abstract

Mattila, Matias:

Building an environment to host a website in a server

Bachelor's thesis

Degree Programme in Business Information Systems Administration

April 2023

Number of pages: 49

The objective of this thesis was to create an environment for a website to be hosted. The thesis started from the evaluation of Virtual Private Server platforms and domain registrars. Also, the selection process of the services was explored.

Walked through the process of setting up a VPS, and how to make a secure remote connection to it from the user's machine. The server and VPS service were configured to meet the requirements for website hosting. Also, the server was configured for data transfer to give the user easy access for transferring files to the server.

A website was made for the thesis, by using a library called ReactJS, which allowed component-based website development. Before making the website, walked through React and its competitor Vue. Finally, the website was developed with React and a tool called Vite. The development occurred on a local machine, instead of the server.

As a result, this thesis provides a functional platform for a website that was created. It allows the installation of other services beside the website. The server is configured for remote connection and easy data transfer between the user and the server. All the steps for this thesis were walked through so, that the reader can understand them easily.

Keywords: domain, VPS, DNS, runtime environment, SSL certificate

SISÄLLYS

1 JOHDANTO	7
2 MITÄ TARVITAAN VERKKOSIVUN PERUSTAMISEEN	8
2.1 Yleistä.....	8
2.2 Verkkotunnus	8
2.3 DNS.....	9
2.4 Verkkosivun alusta	9
2.5 Virtual Private Server.....	10
3 VERKKOTUNNUKSEN REKISTERINPITÄJÄ	11
3.1 Yleistä rekisterinpitäjistä	11
3.2 Valinnassa vaikuttavat tekijät	11
3.3 Valittu rekisterinpitäjä	12
3.4 Domainin rekisteröinti.....	12
4 VPS-PALVELUNTARJOAJA.....	13
4.1 VPS-palveluntarjoajan arviointi.....	13
4.1.1 Palvelujen vertailu.....	14
4.2 Pilvipalveluntarjoajan valinta	15
4.3 Virtuaalikoneen luominen	15
4.3.1 Verkoasetukset	16
4.3.2 SSH-asetukset.....	17
4.3.3 Viimeistely	18
5 ETÄYHTEYDEN MUODOSTAMINEN	18
5.1 Etäyhteys Windows-käyttöjärjestelmällä.....	18
5.2 Etäyhteys unix-pohjaisella käyttöjärjestelmällä.....	19
5.3 Palvelimen hallinta etäyhteydellä	20
6 PALVELIMEN KONFIGUROINTI	22
6.1 Nginx-asennus palvelimelle.....	22
6.2 Porttien konfigurointi.....	22
6.2.1 Porttimäärytykset.....	23
6.2.2 Nginx-etusivu	24
6.3 Nginx-ohjelman konfigurointi	25
6.3.1 Verkkosivukansion oikeudet	25
6.3.2 Nginx-konfiguraatitiedosto	26
6.3.3 Konfiguraation käyttöönotto	27
7 DOMAININ OHJAUS PALVELIMEEN.....	28
7.1 Nimipalvelujen merkinnät	28

7.2 Verkkosivulle tarvittavat merkinnät.....	28
7.3 SSL-sertifikaatti	29
7.3.1 Yleistä.....	29
7.3.2 Tarvittavat työkalut.....	30
7.3.3 Certbotin asennus.....	30
7.3.4 SSL-sertifikaatin hankinta	31
8 VERKKOSIVUN RAKENTAMINEN.....	32
8.1 ReactJS.....	32
8.1.1 React-komponentit.....	32
8.1.2 Virtuaalinen DOM	33
8.1.3 React Native	33
8.2 Vue.js	34
8.2.1 Yhtenäisyydet ja erot Reactiin	34
8.2.2 JavaScript Framework	34
8.3 Lopputulos.....	35
8.4 Projektin aloitus	35
8.4.1 Node.js	35
8.4.2 Vite-projekti.....	36
8.5 Projektin muokkaaminen	37
8.5.1 Valmiin pohjan poistaminen.....	37
8.6 Oma verkkosivu.....	38
8.6.1 Navigaatio.....	39
8.6.2 Etusivu.....	40
8.6.3 Yhteydenotto-välilehti	41
8.6.4 Uusien komponenttien lisääminen "App"-komponenttiin.....	42
8.6.5 Verkkosivun CSS.....	43
8.6.6 Projektin viimeistely	43
8.7 Projektin lähettäminen palvelimelle SFTP:n avulla.....	44
8.7.1 Yleistä.....	44
8.7.2 SFTP:n edut ja ominaisuudet	44
8.7.3 SFTP:n toiminta.....	45
8.8 SFTP:n konfigurointi.....	45
8.8.1 SFTP käyttäjä ja oikeudet.....	45
8.9 SFTP-yhteyden muodostaminen	46
8.9.1 Yhdistäminen käyttäen FileZillaa	47
9 POHDINTA	49
LÄHTEET.....	50

SYMBOLI- JA LYHENNELUETTELO

DNS

Domain Name System, joka tarkoittaa netin nimipalvelujärjestelmää.

DOM

Document Object Model esittää verkkosivun elementtejä, sekä rakennetta.

Domain

Verkkotunnus eli domain, jonka avulla viitataan verkossa oleviin laitteisiin.

Framework

Rakenne, jota käytetään perustana ja tukena rakentamiselle.

SFTP

SSH File Transfer Protocol, tarkoittaa tiedonsiirtoprotokollaa, joka perustuu SSH-protokollaan.

SSH

Secure Shell, joka on protokolla tarkoitettu turvalliseen yhteyteen laitteiden välillä.

SSL

Secure Sockets Layer on salausprotokolla, joka on tarkoitettu verkkoliikenteen salaamiseen.

TLS

Transport Layer Security on vanhentuneen SSL protokollan uudempi versio.

VPS

Virtual Private Server tarkoittaa virtuaalista palvelinta, joka voidaan vuokrata omaan käyttöön.

1 JOHDANTO

Internetin käytön yleistyessä on tärkeää olla jonkinlainen läsnäolo verkossa. Jos halutaan vahvistaa brändiä tai omaa portfolioa, verkkosivu on oiva keino siihen. Verkkosivut mahdollistavat visuaalisen esittämisen itsestäsi tai yrityksestäsi. Monesti ihmiset tutkivat ensin verkosta yritystä, ennen päätöksiään. Yksityishenkilölle omat sivu voivat parantaa mahdollisuuksia työnhaussa. Kun käytetään omaa palvelinta, saadaan sen avulla parempi hallinta omasta sivusta.

Tässä opinnäytetyössä käydään läpi, miten palvelimelle asennetaan tarvittavat työkalut ja ohjelmat, jotta verkkosivu saadaan näkyviin julkisessa verkossa. Ensin tutustutaan mitä pitää tietää ennen asennusta, sekä miten verkkotunnukset eli domainit toimivat. Perehdytään myös domainin ja VPS-palveluntarjoajien valintaan. Lopuksi konfiguroidaan palvelin valmiiksi verkkosivua varten, sekä tehdään verkkosivu ja lähetetään sen tiedostot palvelimelle. Työllä on tarkoitus selvittää, kuinka vaikeaa on perustaa ympäristö verkkosivulle ilman webhotellia tai muuta valmista pohjaa.

2 MITÄ TARVITAAN VERKKOSIVUN PERUSTAMISEEN

2.1 Yleistä

Verkkosivun perustamiseen tarvitaan yleisesti kaksi asiaa: verkkotunnus eli domain, sekä palvelin tai palvelu, jossa verkkosivu sijaitsee. Verkkosivu voidaan myös tehdä ilman verkkotunnusta. Sellaisissa tapauksissa käyttäjä yhdistää palvelimen verkko-osoitteeseen eli IP-osoitteeseen eikä verkkotunnukseen kuten samk.fi. Näitä ovat yleensä erilaiset hallintapaneelit, joita ei ole tarkoitettu verkkosivuiksi. (Miller, 2021.) Tässä työssä verkkotunnus ja palvelin hankitaan eri palveluntarjoajilta, joista kerrotaan lisää myöhemmin.

2.2 Verkkotunnus

Verkkotunnus eli domain on tekstin pätkä, joka merkkaa verkko-osoitetta. Se koostuu nimestä kuten samk ja päätteestä fi, jotka erotetaan pisteellä. Yhdessä niistä saadaan verkkotunnus: samk.fi. Tällä verkkotunnuksella merkitään palvelinta, jossa verkkosivu sijaitsee. Käyttäjä kirjoittaa verkkotunnuksen selaimeensa kuten Google Chrome tai Mozilla Firefox, joka hakee oikean verkko-osoitteen käyttäen DNS:ää eli Domain Name Systemiä. (Internet Corporation for Assigned Names and Numbers, n.d.)

Verkkotunnuksia hallinnoidaan verkkotunnuksien rekistereillä. Näillä valtuutetaan verkkotunnuksien varaus niiden rekisteröijille. Jokainen verkkotunnuksen haluava voi ostaa sen rekisterinpitäjältä. Rekisterinpitäjiä ovat esimerkiksi Cloudflare, GoDaddy ja Namecheap. Verkkotunnuksen hinta voi vaihdella sen päätteiden mukaan. Rekisteröidessä verkkotunnusta kannattaa pitää myös mielessä, että sillä pitää olla uniikki nimi. Eli tunnuksia ei voi olla kahta samalla nimellä. (Internet Corporation for Assigned Names and Numbers, n.d.)

Verkkotunnuksen rekisterinpitäjä oli tässä työssä Cloudflare, josta kerrotaan lisää myöhemmin.

2.3 DNS

DNS mahdollistaa verkkotunnusten kuten samk.fi, google.com tai apple.com kääntämisen verkko-osoitteiksi kuten 192.168.1.100, joka voisi olla kyseisen palvelimen osoite, jossa esimerkiksi apple.com sijaitsee. Jokaisella laitteella verkossa on uniikki verkko-osoite. Kun käyttäjä kirjoittaa selaimensa osoitteen, tarvitaan käännoä löytääksemme oikean osoitteen. Internetin DNS-järjestelmä toimii kuin puhelinluettelo, hallitsemalla nimien ja numeroiden välistä kartoitusta. (Amazon Web Services, n.d.b)

2.4 Verkkosivun alusta

Verkkosivua varten tarvitaan oma palvelin tai vastaavasti jonkin palveluntarjoajan alusta, jossa verkkosivu sijaitsee. Palvelimeen tai alustaan pystytään otamaan etäyhteys, jonka avulla sitä hallitaan. Suurin osa verkkosivuista käyttää palveluntarjoajia kuten Google Cloud, Amazon Web Services tai jotain pienempää webhotelli-palvelua. (Schäferhoff, n.d.) Kun tehdään pienemmän kokuokan verkkosivua, on järkevämpää vuokrata resursseja esimerkiksi VPS-palveluista tai webhotelleista, jotta säästetään ylläpitokuluissa.

Webhotellit ovat palveluja, jotka ovat tehty verkkosivuja varten. Ne tarjoavat asiakkailleensa ympäristön, jolla he voivat luoda verkkosivun ja ylläpitää sitä. Webhotelleissa on yleensä jokin hallintapaneeli kuten cPanel, joka on suosittu valinta webhotelleissa. (Schäferhoff, n.d.) Koska verkkosivut eivät yleensä vaadi paljon resursseja, pystyvät webhotelli-palvelut jakamaan monta verkkosivua samalle palvelimelle. Koska webhotellit ovat tehty verkkosivuja varten, ne tarjoavat työkaluja kuten WordPress, jonka avulla voidaan luoda verkkosivu helposti käyttäen graafista käyttöliittymää. (Hartman, 2021.)

2.5 Virtual Private Server

VPS eli Virtual Private Server tarkoittaa virtuaalikonetta, joka sijaitsee jossain fyysisellä palvelimella. Sen avulla jokainen käyttäjä saa eristetyn ympäristön muista käyttäjistä. Se antaa käyttäjille paremmat hallintamahdollisuudet kuin esimerkiksi aikaisemmin mainittu webhotelli. Niitä voidaan käyttää esimerkiksi verkkosivuja varten tai ohjelmistokehitykseen. (IBM, n.d.)

Tässä työssä käytettiin vuokrattua VPS-palvelinta Oracle Cloud Infrastructure -palvelusta. Tämä asennus voidaan myös suorittaa omalle palvelimelle, sen ei tarvitse olla vuokrattu. Jos asennukseen käytetään omaa palvelinta, osa vaiheista voi olla erilaisia, kuten VPS-hallintapaneeli, jota ei omalla palvelimella ole. Myös vaiheet voivat olla erilaisia riippuen VPS-palveluntarjoajasta ja sen hallintaympäristöstä.

Vuokrattu palvelin valittiin työhön, koska se tarjoaa verkkosivun tekijälle laajemman hallinnan alustasta, jossa verkkosivu sijaitsee. Kun käytetään omaa tai vuokrattua palvelinta, tekijä pystyy myös laajentamaan toimintaansa. On mahdollista asentaa palvelimelle muitakin palveluja, kuten Nextcloud, joka on palvelimella toimiva Microsoft Officen tyyppinen avoimenlähdekoodin palvelu.

Vuokratulle palvelimelle valittiin saatavilla olevista vaihtoehdoista Linux-pohjainen käyttöjärjestelmä, johon myöhemmin otettiin etäyhteys. Valitsin palvelimeen resurssit tarpeideni mukaan. En kuitenkaan tarvinnut kovin tehokkaita resursseja pelkästään verkkosivua varten. Jos asennus tehdään omalle palvelimelle, olisi suositeltua tehdä uusi virtuaalikone verkkosivua varten, jotta se olisi eristetty oikeasta palvelimesta ja helpommin hallittavissa ongelmatilanteiden sattuessa.

3 VERKKOTUNNUKSEN REKISTERINPITÄJÄ

On tärkeää valita hyvä ja luotettu domainin rekisterinpitäjä. Domainin rekisteröinti ei ole vaikeaa, mutta on tärkeää tietää ominaisuudet, joita palveluntarjoajalla on saatavilla ennen kuin sitoudutaan mihinkään yhteen yritykseen. Koska verkkosivu tullaan löytämään sen nimen avulla, siksi kannattaa valita domainin nimeksi jokin helposti muistettava.

3.1 Yleistä rekisterinpitäjistä

Rekisterinpitäjä on yksikkö, joka rekisteröi ostamasi verkkotunnuksen ICANN nimisen kansainvälisen hallinnointiorganisaation kanssa. ICANN on lyhenne, joka tulee nimestä: International Corporation for Assigned Names and Numbers. ICANN hallinnoi verkkotunnuksia kuten, .com ja .org. Se toimii välikätenä ja varmistaa, että se on juuri sinun verkkotunnuksesi. (Cloudflare, n.d.a.)

3.2 Valinnassa vaikuttavat tekijät

Eri domainin päätteillä kuten .com, .org tai .fi on eri hinnat. Kallein päätte ei aina ole paras. Päätteillä on eri tarkoituksia, esimerkiksi .fi tarkoittaa maakoodia ja se on uniikki Suomelle. Samalla logiikalla maakoodi Saksalle olisi .de. (Organ, 2023.) Suomessa yritykset yleensä käyttävät .fi-päätettä ja kansainväliset yritykset käyttävät yleensä .com-päätettä, koska ei ole pakollista käyttää tiettyä päätettä domainit voivat olla melkein mitä tahansa.

Kun ostat domainin, se on sinulla vain rajatun ajan, tämän ajan jälkeen se pitää uusida. Yleisin uusimisaika domainille on vuosi, mutta on mahdollista uusida kymmenenkin vuoden välein. Ei ole ideaalista valita domainin nimeä ja sitten menettää se vuoden päästä. Tämän takia rekisterinpitäjät uusivat domainin automaattisesti puolestasi. On tärkeää pitää mielessä, että jotkin palveluntarjoajat antavat alennuksia ensimmäisen domainin ostamisesta, mutta uusimismaksu onkin seuraavana vuonna huomattavasti suurempi. (Organ, 2023.)

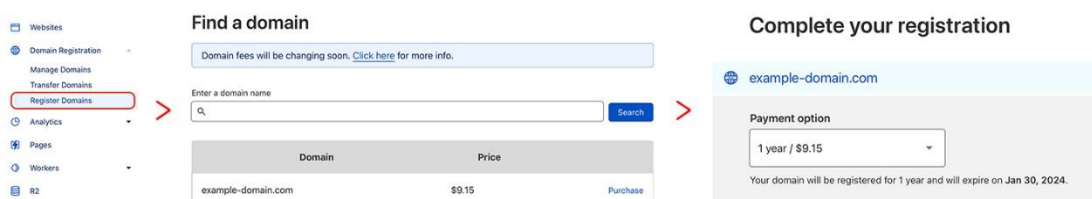
Hyvä paikka aloittaa tutkiminen on dokumentaatio. Ovatko tärkeät tiedot helposti saatavilla, kuten ohjeet uusimisprosessiin, tuleeko verkkosivun mukana SSL-sertifikaatti, miten verkkotunnuksen saa siirrettyä toiseen palveluun, sekä kuinka domainin tilaus voidaan perua. Kannattaa myös lukea arvosteluita verkosta ja kokemuksia, miten yritys on hoitanut muiden asiakkaiden ongelmatilanteet.

3.3 Valittu rekisterinpitäjä

Käyttäen edellä mainittuja keinoja, valitsin työhön rekisterinpitäjäksi Cloudflaren. Päädyin tähän valintaan koska Cloudflare on yleisesti luotettu ja moni verkkosivu käyttää jo sen tarjoamaa suojauspalvelua. Se tarjoaa myös ilmaisen SSL-sertifikaatin ja tietojen suojauksen domainin mukana. Tietojensuojaus tarkoittaa sitä, että nimesi ja osoitteesi eivät ole näkyvissä, kun haetaan tietoja domainista. Kun tietojensuojaus on käytössä domainista tietoja haettaessa näkyvissä on Cloudflaren tiedot, jonka avulla saadaan yhteys sinuun. (Cloudflare, n.d.d) Cloudflaren dokumentaatio on myös selvää ja sieltä löytyy tarvittavat tiedot domainin hallintaan. Koska minulla oli aikaisempaa kokemusta Cloudflaren palveluista, se helpotti käyttöä.

3.4 Domainin rekisteröinti

Tässä vaiheessa olin tehnyt tilin Cloudflareen ja suoritin domainin rekisteröinnin. Suuntasin valikosta kohtaan ”Domain Registration” ja sieltä valitsin ”Register Domains” (Kuva 1). Noudatin ruudulla näkyviä ohjeita ja suoritin maksutapahtuman, kun olin valinnut nimen omalle domainille. Rekisteröinnin jälkeen voitiin siirtyä valitsemaan VPS-palvelua, jossa verkkosivun tiedostot tulevat sijaitsemaan.



Kuva 1. Domainin rekisteröinti Cloudflaren hallintapaneelissa.

4 VPS-PALVELUNTARJOAJA

On tärkeää tietää VPS-palvelun turvallisuusominaisuudet, kun vertaillaan eri palveluja. Monet VPS-palvelut tarjoavat mahdollisuuden kokeilla niitä ilmaiseksi ja ne ovat usein helppokäyttöisiä. VPS-palveluita voidaan arvioida esimerkiksi tekemällä niihin testiasennuksia ilmaisella kokeiluversiolla. Testaamisen avulla saadaan selville, täyttääkö VPS-palvelu kaikki odotukset, ennen kuin siitä maksetaan. (Rowland, 2021)

4.1 VPS-palveluntarjoajan arviointi

Yleisiä toimenpiteitä VPS-palvelun ja sen turvallisuuden arviointiin ovat esimerkiksi dokumentaation sekä sen verkkoympäristön palomuurien käytäntöjen tutkiminen, jotta virheellisiä määrittämiä ei tulisi tehtyä vahingossa. On myös tärkeää selvittää, miten virtualisoidut palvelimet on suojattu palvelussa.

Kuten verkkotunnuksien rekisterinpitäjien valinnassa, on suositeltavaa valita tunnettu ja luotettu VPS-palveluntarjoaja, ennen kuin luovutamme omia tietoja. Huomioon kuuluu myös ottaa, miten pilvipalvelu käsittelee sinne tallentamaasi tietoa. Ovatko esimerkiksi virtuaalikoneesi lukittu kyseiseen palveluun vai pystytkö mahdollisesti siirtämään ne tulevaisuudessa toiseen palveluun. Tulevaisuuden kannalta on myös tärkeää, että siirtyminen on helppoa ja nopeaa.

Palvelua valittaessa on tärkeää miettiä sen virhemarginaalia. Onko palvelussa paljon katkoja? Saatavuus on tärkeä osa palveluntarjoajan valitsemisessa. Jos palvelussa on paljon katkoja, se tarkoittaa yleisesti sitä, ettei verkkosivu olisi katkon aikana saatavilla vierailijoille. Katkoksia voi seurata yleensä palveluntarjoajan tai jonkin kolmannen osapuolen sivuilta. On silti ymmärrettävää, että

katkoksia voi sattua. Tämän takia kannattaa myös tietää pilvipalvelun mahdollisuudet varmuuskopiointiin. (Rowland, 2021)

4.1.1 Palvelujen vertailu

Palveluja vertaillessa on arvioitava, kuinka paljon resursseja tarvitaan. Monet palvelut tarjoavat pienitehoisia virtuaalikoneita, jotka on tarkoitettu pienimuotoisille sovelluksille tai esimerkiksi verkkosivulle. Ne yleensä sisältävät yhden ytimen ja yhden gigatavun tai vähemmän järjestelmämuistia.

Resurssien käyttöä voidaan arvioida miettimällä, kuinka monta vierailijaa luolet, että sivu saa. Jos arvelet sivulla vierailevan paljon käyttäjiä voit varata enemmän resursseja. Jos sivu onkin verkkokauppa ja tarvitsee resursseja maksutapahtumien käsittelyyn, sekin voi nostaa resurssitarpeita. (Hostman, 2021) Koska verkkosivuni on staattinen, eli se ei muutu, niin sille ei ole pakko myöskään varata tehokkaita resursseja.

Vertasin neljää palvelua valitessani käytettävää VPS-palveluntarjoajaa. Verratut palvelut olivat: Google Cloud, Amazon Web Services, Digital Ocean ja Oracle Cloud Infrastructure. Kaikki palvelut tarjoavat runsaan dokumentaation ja ovat tunnettuja, sekä luotettuja VPS-palveluntarjoajia. Koska näiden palvelujen takana ovat suuret yritykset, yleensä katkoja ei palveluissa esiinny. Voidaan siis olettaa, että verkkosivussa ei tule esiintymään katkoja palveluntarjoajan takia.

Google Cloud, Amazon Web Services ja Oracle Cloud tarjoavat "pay as you go" -mallia. Se tarkoittaa käytännössä sitä, että maksamme varaamastamme resurssista sen verran kuin käytämme sitä. Palvelut tarjoavat työkaluja suunnata antavien hintojen laskemiseen. Testasin Amazonin palvelusta hintojen laskuria yhdelle virtuaalipalvelimelle, jossa on Linux käyttöjärjestelmä, 20 gigatavua tilaa, yksi teratavu datansiirtoa ja 500 megatavua järjestelmämuistia, sen hinnaksi tulisi noin. 3,16 euroa kuukauteen. Hinta on laskettu 1.2.2023 Amazon Web Services -palvelussa.

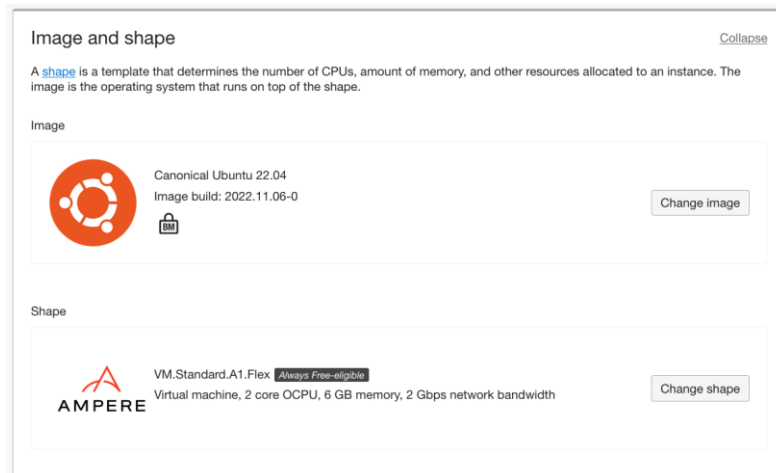
4.2 Pilvipalveluntarjoajan valinta

Palveluita verratessa päädyin valitsemaan Oracle Cloudin, koska olen käyttänyt sitä aikaisemminkin ja tiedän, että se tulee sopimaan tähän asennukseen mainiosti. Oracle Cloud tarjoaa hyvät mahdollisuudet testaamiseen, sekä on helposti laajennettavissa, jos sitä tulevaisuudessa tarvitsee. Oracle Cloud on muita palveluja kalliimpi, mutta lupaa parempaa toimintakykyä. Oracle mittaa virtuaalikoneidensa prosessorit käyttäen OCPU:ta, kun muut palvelut mittaavat ne käyttäen vCPU:ta. Oracle lupaa, että OCPU on ainakin kahden vCPU:n arvoinen. (Reger, 2021.)

CPU tarkoittaa tietokoneen prosessoria. OCPU on Oraclen oma määritelmä ja tarkoittaa Oraclen prosessoria, kun taas vCPU tarkoittaa virtuaalista prosessoria. (Reger, 2021.) Virtuaalisia prosessoreita ei lasketa millään kaavalla, vaan se riippuu käyttöjärjestelmästä ja fyysisestä prosessorista. Virtuaaliset prosessorit merkkavat ajan määrää, jonka virtuaalikone käyttää oikealla prosessorilla. (Intel, n.d.)

4.3 Virtuaalikoneen luominen

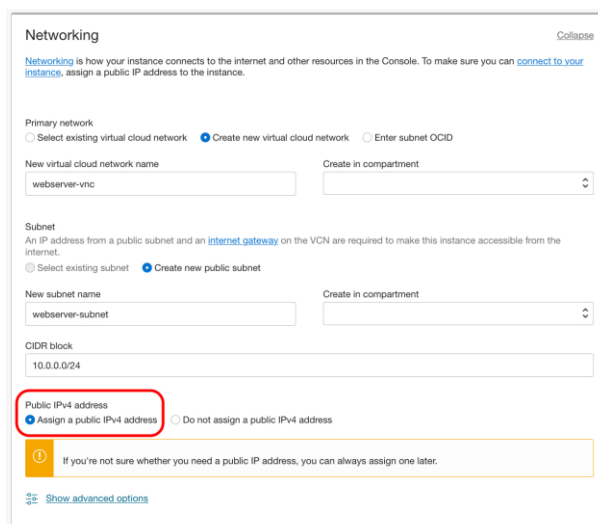
Tässä työn vaiheessa luotiin uusi virtuaalikone käyttäen Oracle Cloudin käyttöliittymää. Luominen tapahtui ”Instances”-välilehdeltä. Tässä huomattiin nappi, jossa lukee ”Create Instance”. Nimesin sekä konfiguroin virtuaalikoneen resurssitarpeideni eli verkkosivun mukaan. Valitsemalla ”Image” pystyttiin muuttamaan käyttöjärjestelmä. Valitsin käyttöjärjestelmäksi Ubuntu 22.04, sekä ”Shape” eli virtuaalikoneen resursseiksi valitsin VM.Standard.A1.Flex-konfiguraation kahdella ytimellä ja kuuden gigatavun järjestelmämuistilla, koska se ei tuonut lisäkuluja ja mahdollistaa joustavuuden tulevaisuudessa (Kuva 2).



Kuva 2. Virtuaalikoneen käyttöjärjestelmä ja sen muoto

4.3.1 Verkoasetukset

Seuraavaksi luotiin verkko virtuaalikoneelle valitsemalla “Networking” ja sen alavalikoista “Create new virtual cloud network”, sekä “Create new public subnet”. Tämä avasi uuden valikon, josta pääsin konfiguroimaan uuden verkon. Annoin verkolle ja sen aliverkolle nimet webserver-vnc ja webserver-subnet. Koska verkkosivu haluttiin julkiseen verkkoon, se tarvitsee IP-osoitteen. Se saatiin valitsemalla “Assign a public IPv4 address”. (Kuva 3).

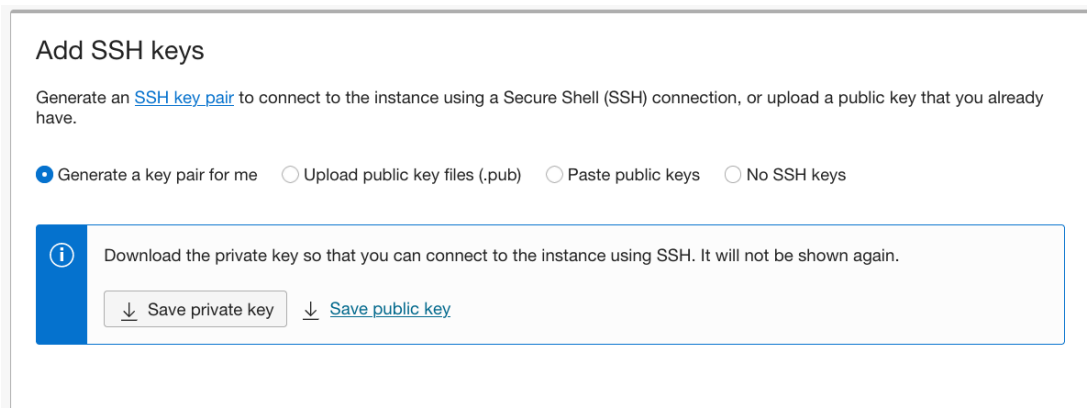


Kuva 3. Verkon konfigurointi Oracle Cloudissa.

4.3.2 SSH-asetukset

SSH tulee sanoista Secure Shell ja se on protokolla, joka mahdollistaa turvallisen kommunikaation oman laitteemme ja palvelimen välillä. Siinä on kolme kerrosta: kuljetus-, todennus- ja yhteyskerros. Kuljetuskerros varmistaa turvallisen kommunikaation, se ylläpitää myös välimuistia. Todennuskerros suorittaa todennusmenettelyn ja varmistaa, että yhdistävällä käyttäjällä on käyttöoikeus. Yhteyskerros ylläpitää yhteyttä oman laitteemme ja palvelimen välillä todennuksen jälkeen. (SSH, n.d.)

Yhteyttä varten tarvitaan SSH-avain, joka mahdollistaa etäyhteyden palvelimeen sen hallintaa varten. On tärkeää luoda SSH-avain turvallisuuden vuoksi. Avainta luodessa valittiin kohta "Generate a key pair for me", joka tekee automaattisesti avaimen, sekä antaa mahdollisuuden ladata sen. Latasin luodun "private key" -avaimen omalle tietokoneelleni (Kuva 4). Siirsin avaimen paikkaan, josta se on helposti löydettävissä.



Kuva 4. SSH-avaimien luominen Oracle Cloud -palvelussa

Kun avain oli luotu ja tallennettu laitteelle, jätettiin loput asetukset suositeltuihin vaihtoehtoihin, eli ei muokattu niitä. Asetukset usein menevät valmiiksi hankitun virtuaalikoneen mukaan. Kannattaa pitää mielessä, ettei muokkaa tässä vaiheessa virtuaalikonetta tehokkaammaksi, sillä se voi johtaa lisäkuluihin.

4.3.3 Viimeistely

Kun konfigurointi oli valmis, ei tarvinnut enää kuin viimeistellä virtuaalikone painamalla nappia "Create". Koneen luomisessa kesti muutama minuutti, joten odotin niin kauan, että virtuaalikone oli tilassa "running". Tila "running", tarkoittaa, että virtuaalikone on päällä. Kun virtuaalikone oli päällä, nähtiin hallintapaneeli, joka sisältää kaikki sen tiedot. Tärkeät tiedot tässä vaiheessa olivat "Public IP address" ja "Username". Näitä tietoja käytetään, kun muodostetaan etäyhteyttä.

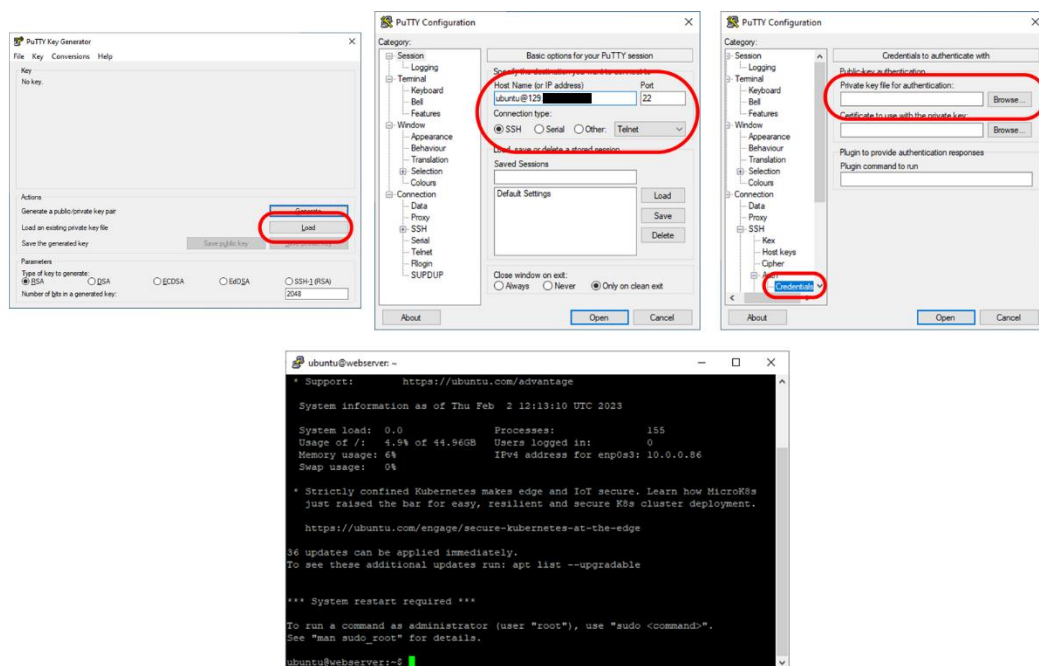
5 ETÄYHTEYDEN MUODOSTAMINEN

Ennen kuin etäyhteyttä muodostetaan, pitää valmistella oma laite siihen. Jos laitteessa on käytössä Windows-käyttöjärjestelmä, voidaan etäyhteys muodostaa käyttämällä PuTTY-nimistä ohjelmaa. PuTTY tarjoaa käyttöliittymän, jolla voidaan yhdistää palvelimeen käyttämällä ladattua SSH-avainta. Jos käytössä on Unix-pohjainen käyttöjärjestelmä kuten macOS, Linux tai FreeBSD, SSH-ohjelma on usein asennettu valmiiksi ja sitä voidaan käyttää terminaalikomennolla "ssh". PuTTY voidaan kuitenkin asentaa sen verkkosivuilta: www.chiark.greenend.org.uk/~sgtatham/putty/

5.1 Etäyhteys Windows-käyttöjärjestelmällä

Kun muodostetaan etäyhteyttä Windows-käyttöjärjestelmällä, ladattu avain pitää ensin muuntaa sopivaan muotoon, jotta sitä voidaan käyttää PuTTY-ohjelmassa. Avaimen muunto tapahtui PuTTYgen-ohjelmalla, joka tulee PuTTY-ohjelman asennuksen mukana. Avain muunnettiin oikeaan muotoon valitsemalla kohta "Load", johon syötettiin Oracle Cloudista saatu avain. Tämän jälkeen valittiin "Save private key", joka tallensi avaimen "ppk"-muotoon (Kuva 5).

Kun avain oli tallennettu, pystyttiin yhdistämään palvelimeen PuTTY:n avulla. Yhdistettäessä valittiin SSH-kohdasta "Connection type" ja syötettiin aikaisemmin Oracle Cloudista saatu käyttäjänimi ja IP-osoite, sekä annettiin portille numero 22. Vielä avasin PuTTYgeniä käyttäen tehdyn avaimen valitsemalla PuTTY:n vasemmanpuoleisesta valikosta "Connection"-kohdan alta "SSH", jonka jälkeen vielä valitsin "Auth" ja lopuksi "Credentials" (Kuva 5). Nyt tehty "ppk"-avain pystyttiin syöttämään kohtaan "Private key file for authentication". Kun nämä vaiheet oli suoritettu, avasin etäyhteyden painamalla "Open". (Oracle, 2023b.)



Kuva 5. Etäyhteys Windows käyttöjärjestelmästä PuTTY:n avulla

5.2 Etäyhteys unix-pohjaisella käyttöjärjestelmällä

Kun käytetään etäyhteyden muodostamiseen Unix-pohjaista käyttöjärjestelmää, ennen yhdistämistä pitää muokata avaimen oikeudet, jotta vain omistaja voi lukea sen. Avaimelle kuuluu antaa tason 400 oikeudet. Se tarkoittaa, että vain omistaja voi lukea tiedoston, eikä se ole suoritettavissa (Oracle, 2023b.). Komento suoritettiin käyttäen terminaalia. Koska tallensin aikaisemmin ladatun avaimen kansioon nimeltä "sshx" ja annoin sille nimeksi "ock.key", komentoni oli siis "chmod 400 sshx/ock.key".

On hyvä silti testata ennen komennon suorittamista, onko avain oikeasti kansiossa. Kansion tiedostot nähtiin käyttämällä "ls"-komentoa, joka listaa kansion tiedostot. Siihen käytin komentoa "ls sshx/". Koska tiedosto oli kansiossa, pystyttiin aikaisempi "chmod"-komento suorittamaan onnistuneesti. Komennosta ei tullut palautetta terminaaliin.

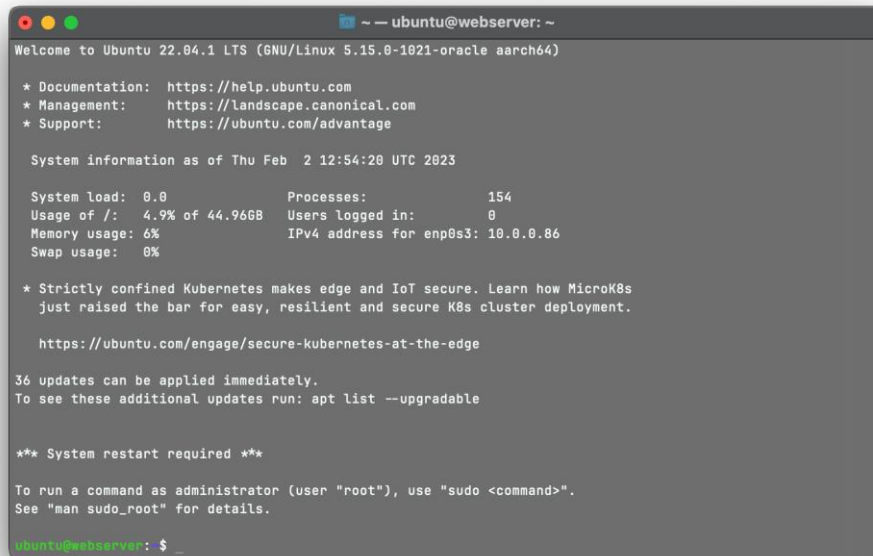
Kun komento oli suoritettu, oltiin valmiita muodostamaan etäyhteys palvelimeen. Yhdistämiseen käytettiin komentoa "ssh -i". (Oracle, 2023b.) Siihen tarvittiin aikaisemmin saadut, käyttäjänimi ja IP-osoite. Käytän työssä esimerkkinä kuvitteellista IP-osoitetta: 123.456.789.123. Ohjeiden mukaisesti komentoni oli "ssh -i /Users/matias/sshx/ock.key ubuntu@123.456.789.123".

Tämä komento muodostaa yhteyden palvelimeen, jonka IP-osoite on 123.456.789.123, sekä kirjautuu sisään avaimella "ock.key" ja käyttäjänimellä ubuntu. Kun yhteys oli avattu, saatiin hallinta palvelimesta. Se huomattiin terminaalissa olevasta tunnuksesta "ubuntu@webserver", ubuntu tarkoittaa käyttäjänimeä ja webserver palvelimelle annettua nimeä (Kuva 6).

5.3 Palvelimen hallinta etäyhteydellä

Nyt hallinta tapahtuu samalla tavalla riippumatta oman laitteen käyttöjärjestelmästä. Ennen palvelimen konfigurointia päivitin palvelimen ohjelmat. Se tapahtui syöttämällä komento "sudo apt update" etäyhteysterminaaliin. Päivityksen jälkeen järjestelmä käynnistettiin uudelleen komennolla "sudo reboot". Etäyhteys piti muodostaa käynnistyksen jälkeen uudelleen.

Monesti hallitessa Linux-käyttöjärjestelmää terminaalikomentojen eteen pitää laittaa termi "sudo". Se on lyhenne sanoista "superuser do" ja antaa käyttäjälle laajemmat oikeudet hallintaan. Tämän avulla pystytään muokkaaman tiedostoja ja suorittamaan komentoja, jotka tarvitsevat laajemmat käyttöoikeudet käyttöjärjestelmässä. Laajemmat oikeudet voidaan myös hankkia käyttämällä "root"-käyttäjää, mutta se ei ole usein suositeltua. (Carrigan, 2020)



```
ubuntu@webserv: ~
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1021-oracle aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Thu Feb  2 12:54:20 UTC 2023

System load:  0.0          Processes:      154
Usage of /:   4.9% of 44.96GB  Users logged in:  0
Memory usage: 6%          IPv4 address for enp0s3: 10.0.0.86
Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

36 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

*** System restart required ***

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@webserv: $
```

Kuva 6. Etäyhteys on muodostettu käyttäen "ssh"-komentoa.

6 PALVELIMEN KONFIGUROINTI

6.1 Nginx-asennus palvelimelle

Kun palvelin oli päivitetty, pystyttiin aloittamaan sen konfigurointi verkkosivua varten. Palvelimelle asennettiin Nginx, jonka avulla voidaan ylläpitää verkkosivua. Asennus tapahtui käyttäen komentoa "sudo apt install nginx". Asennus alkoi toimenpiteen hyväksymisen jälkeen. (Canonical, n.d.) Asennuksen jälkeen tarkistin onko Nginx-prosessi päällä. Tarkistus tapahtui komennolla "sudo systemctl status nginx". Saatiin vastaukseksi prosessin tiedot (Kuva 7).

```
ubuntu@webserver:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-02-02 14:03:37 UTC; 26min ago
     Docs: man:nginx(8)
   Process: 854 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Process: 964 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 965 (nginx)
    Tasks: 3 (limit: 7015)
   Memory: 16.8M
     CPU: 113ms
   CGroup: /system.slice/nginx.service
           └─965 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─966 "nginx: worker process"
               └─967 "nginx: worker process"

Feb 02 14:03:36 webserver systemd[1]: Starting A high performance web server and a reverse proxy server...
Feb 02 14:03:37 webserver systemd[1]: Started A high performance web server and a reverse proxy server.
lines 1-17/17 (END)
```

Kuva 7. Aktiivinen Nginx-prosessi

6.2 Porttien konfigurointi

Koska Nginx oli aktiivinen, pystyttiin sille antamaan oikeus kommunikointiin portin 80 kautta. Oikeus annettiin käyttäen komentoa "sudo iptables -I INPUT 6 -m state --state NEW -p tcp --dport 80 -j ACCEPT". Komento päivitti palomuurin ja antoi Nginx-ohjelman käyttää HTTP-porttia, eli porttia 80 kommunikointiin. Muutos tallennettiin komennolla "sudo netfilter-persistent save". (Oracle, 2023c)

Kun muutokset oli tehty, siirryttiin takaisin Oracle Cloud -palveluun. Ennen kuin Nginx testisivu tulee näkyviin julkisessa verkossa, piti portti 80 avata vielä Oracle Cloudista. Voidaan myös avata portti 443, joka on tarkoitettu HTTPS-

yhteydelle. Sitä tullaan tarvitsemaan myöhemmin, kun hankitaan SSL-sertifiikaattia.

HTTP ja HTTPS ovat lyhenteet sanoista "Hypertext Transfer Protocol", jossa HTTPS:än tapauksessa lopussa on vielä lisättynä sana "Secure". HTTP on protokolla, jota käytetään tiedon välittämiseen internetissä. HTTPS on versio HTTP-protokollasta, joka tarjoaa tiedon turvallisen välityksen, sillä se salaa tiedon, jotta vältetään luvatonta pääsyä tietoon. (Cloudflare, n.d.c)

Porttien avaaminen tapahtui Oracle Cloud -hallintapaneelin "Virtual Cloud Networks"-välilehdeltä. Valittiin aikaisemmin luotu verkko ja sen aliverkko. Valinnan jälkeen päädyttiin "Security Lists"-valikkoon, josta valittiin "Default Security List" (Kuva 8). Sieltä pystyttiin lisäämään sääntöjä, joiden avulla määriteltiin porttien avoimuus. (Oracle, 2023a.)

6.2.1 Porttimääritykset

Uudet määritykset asennukselle tehtiin antamalla oikeus kaikille IP-osoitteille kohtaan "Source CIDR", sekä antamalla portin numero. Määritettiin IP-verkko "0.0.0.0/0", sekä lisättiin "Destination Port Range" porttiin 80. Toistin saman määrityksen uudelleen, mutta vaihdoin "Destination Port Range"-kohdan portille 443. Oli mahdollista myös antaa määrityksille kuvaus, jotta se on helppo muistaa. Määritin kuvauksen http portille 80 ja portille 443 kuvauksen https (Kuva 8).

The image shows a composite screenshot of the Oracle Cloud console. On the left, the 'Virtual Cloud Networks' section shows a list with 'Virtual Cloud Networks' highlighted. Below it, 'Resource search results' shows 'webservice-vnc' highlighted. The 'Subnets' section shows a 'Create Subnet' button and a list with 'webservice-subnet' highlighted. The 'Ingress Rules' section shows an 'Add Ingress Rules' button and a table with columns 'Stateless' and 'Source'. On the right, the 'Add Ingress Rules' dialog is open, showing two rules. Rule 1 is for port 80 (http) and Rule 2 is for port 443 (https). Both rules have 'Source CIDR' set to '0.0.0.0/0' and 'Destination Port Range' set to '80' and '443' respectively. Red circles highlight these key fields.

Kuva 8. Porttien avaaminen Oracle Cloud -palvelussa

6.2.2 Nginx-etusivu

Porttien määrittelyn jälkeen, oli mahdollista hakea verkkoselaimen avulla palvelimen julkista IP-osoitetta. Osoite on sama, jolla muodostettiin etäyhteys palvelimeen. Kun syötin selaimen palvelimen osoitteen, sain tulokseksi Nginx-ohjelman tervetulotoivotuksen. Nyt tiesin, että Nginx toimii ja pystyn jatkamaan konfigurointia (Kuva 9).

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Kuva 9. Nginx-oletussivu

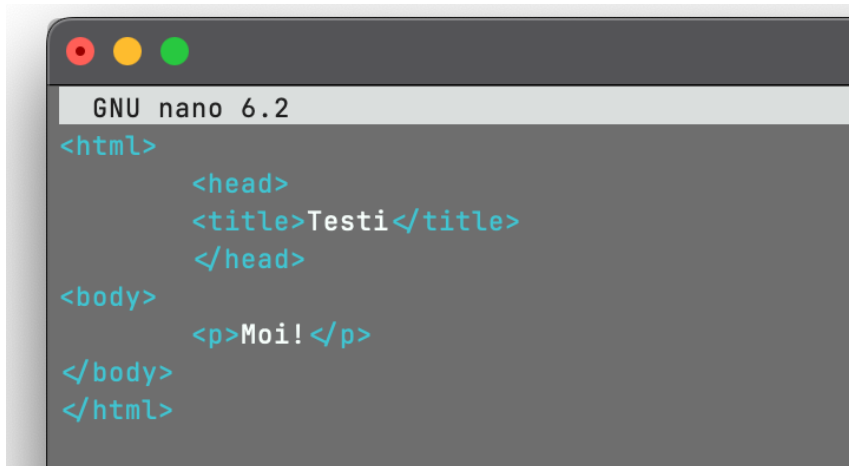
6.3 Nginx-ohjelman konfigurointi

Aloitin Nginx-konfiguroinnin omaa verkkosivua varten. Ensin loin uuden kansion, jossa verkkosivun tiedostot sijaitsevat. Kansio luotiin komennolla `"sudo mkdir -p /var/www/verkkotunnus.com/html"`. Komennossa tulee esille termi `"verkkotunnus.com"`, jota käytän tästä lähtien merkkamaan aikaisemmin hankkimaani domainia. Pitää kuitenkin huomata, että en omista domainia `"verkkotunnus.com"`, käytän sitä vain esimerkkinä tässä työssä. Mainittu komento teki kaksi uutta kansiota kohteeseen `"/var/www/"`, uudet kansiot olivat `"verkkotunnus.com"` ja sen sisällä sijaitseva `"html"`.

6.3.1 Verkkosivukansion oikeudet

Seuraavaksi asetin nykyisen käyttäjän eli `"ubuntu"`-käyttäjän kansion omistajaksi. Kansion omistaja voidaan määrittää komennolla `"sudo chown -R $USER:$USER /var/www/verkkotunnus.com"`. Kansion omistajaksi tulee käyttäjä, jolla komento suoritetaan. Muokkasin käyttäjälle kansioon luku- ja kirjoitusoikeuden. Muokkaus tapahtui käyttäen komentoa `"sudo chmod -R 755 /var/www/verkkotunnus.com"`. Se määritti kansion `"verkkotunnus.com"` oikeudet. (Glass, 2020)

Kun oikeudet oli määritelty, pystyttiin luomaan testisivu. Testisivu kuitenkin korvattiin omalla verkkosivulla myöhemmin. Käytin testisivun luomiseen nano-tekstineditoria, se on valmiiksi asennettu monessa Linux-käyttöjärjestelmässä. Nanao päästiin käyttämään komennolla `"nano"`, sekä tiedoston nimellä. Tein nanon avulla uuden HTML-tiedoston, käyttäen komentoa `"sudo nano /var/www/verkkotunnus.com/html/index.html"`, joka avaa tekstieditorin ja luo tiedoston `"index.html"` kansioon `"html"`. (Canonical, n.d.) Lisäsin tekstieditorin avulla kuvan mukaisen HTML-sivun ja tallensin sen (Kuva 10).



```
GNU nano 6.2
<html>
    <head>
        <title>Testi</title>
    </head>
    <body>
        <p>Moi! </p>
    </body>
</html>
```

Kuva 10. Testi HTML-sivu

6.3.2 Nginx-konfiguraatitiedosto

Konfiguraatitiedosto määrittää Nginx-ohjelmalle juuren eli "root". Se on kansio, jossa sivuni tiedostot tulevat olemaan. Se pystyttiin luomaan nano-tekstieditorin avulla, käyttämällä komentoa "sudo nano /etc/sites-available/verkkotunnus.com", jossa "verkkotunnus.com" oli oman domainin nimi.

Tiedosto "index" on verkkosivun etusivu. Määrittelin sen niin, että se voi olla muotoa index.html tai index.htm. Konfiguraatiossa asetin oman domainin muuttujalla "server_name". Lopuksi oli määritelty "location", joka ohjaa kansion tiedostoihin. Esimerkiksi jos käyttäjä hakee selaimella "verkkotunnus.com/about.html", hänet ohjataan "about.html"-tiedostoon, jos tiedostoa ei löydy Nginx antaa käyttäjälle virhekoodin 404 (Kuva 11).



```
GNU nano 6.2
server {

    root /var/www/verkkotunnus.com/html;
    index index.html index.htm;

    server_name verkkotunnus.com www.verkkotunnus.com;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

Kuva 11. Esimerkki Nginx-konfiguraatiosta

6.3.3 Konfiguraation käyttöönotto

Käyttöönottoa varten konfiguraatio piti linkittää. Uusi konfiguraatio pystyttiin aktivoimaan käyttämällä komentoa "sudo ln -s /etc/nginx/sites-available/verkkotunnus.com /etc/nginx/sites-enabled/", jolloin tapahtui linkitys (Megida, 2020). Komento teki linkin luotuun "verkkotunnus.com"-konfiguraatioon ja lisäsi sen Nginx-aktiivisten sivujen listaan. Muutin vielä nginx.conf-tiedostoa ja aktivoisin sieltä "server_names_hash_bucket_size"-muuttujan. Tämä muuttuja on jo valmiiksi tiedostossa, mutta on asetettu kommentiksi. (Glass, 2020)

Muokkasin tiedostoa nanon avulla ja poistin kommentoinnin muuttujasta. Kommentti on merkattu silloin kun rivi alkaa "#" -merkillä, joten poistin sen muuttujan edestä. Komento muokkaamiseen oli "sudo nano /etc/nginx/nginx.conf". Sen jälkeen pystyin varmistamaan, onko konfiguraatio todenmukainen. Nginx testaa konfiguraatiot, kun sille annetaan komento "sudo nginx -t". Komennon käytön jälkeen sain viestin "syntax is ok" ja "test is successful" (Kuva 12). Käynnistin vielä Nginx-prosessin uudelleen komennolla "sudo systemctl restart nginx". (Nginx, n.d.)

```
[ubuntu@webserv:~]$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ubuntu@webserv:~]$ sudo systemctl restart nginx
```

Kuva 12. Nginx-konfiguraation varmistaminen ja uudelleenkäynnistys

7 DOMAININ OHJAUS PALVELIMEEN

Ennen kuin oli mahdollista nähdä luotu testisivu, domain eli ”verkkotunnus.com” piti ohjata palvelimen julkiseen IP-osoitteeseen. Siirryin muokkaamaan Cloudflare-palvelussa oman domainin DNS-tietoja eli nimipalvelun tietoja. Nimipalvelujen ohjeita on erityyppisiä kuten, A record, CNAME, MX Entry, TXT Record ja AAAA Record. A record on ohjeista yksinkertaisin. Kun käyttäjä syöttää domainin nimen selaimeensa, vastaukseksi saadaan sille osoitettu IP-osoite.

7.1 Nimipalvelujen merkinnät

CNAME, joka on lyhenne sanoista ”canonical name”, ohjaa domainin toiseen domainiin. Kun haetaan selaimella esimerkiksi ”verkkotunnus.com”, CNAME voi muuttaa sen ”www.verkkotunnus.com”-osoitteeksi. MX Entry on tarkoitettu sähköpostia varten ja se tulee sanoista ”mail exchanger entry”. Se ohjaa sähköpostin toiseen palvelimeen, vaikka sähköpostia lähetettäisiin kyseiseen domainiin. Se mahdollistaa sähköpostin hallinnan toisella palvelimella. (Cloudflare, n.d.b)

TXT Record eli ”text record” ei ohjaa yhtään liikennettä, mutta sillä voidaan käyttää tiedon jakamiseen ulkoisille palveluille. Sille on monia eri käyttötarkoituksia. AAAA Recordilla on sama tarkoitus kuin A Recordilla, eli se ohjaa domainin IP-osoitteeseen. AAAA eroaa A:sta siten, että se ei ohjaa IPv4 osoitteeseen vaan IPv6 osoitteeseen. (Cloudflare, n.d.b)

7.2 Verkkosivulle tarvittavat merkinnät

Verkkosivua varten minun piti asettaa A record, koska palvelimen IP-osoite on muodoltaan IPv4-tyyppiä, eikä IPv6-tyyppiä. Koska IPv4-osoite sisältää vain numeroita ja pisteitä, kun taas IPv6 voikin sisältää kirjaimia, ne voidaan erottaa helposti toisistaan. Siirryttyäni Cloudflaren hallintapaneeliin valitsin oman domainin ja kohdasta ”Overview” laitoin päälle ”Development Mode”-vaihtoehdon

ja poistin käytöstä "Under Attack Mode"-valinnan, jotta pystyin näkemään muutokset verkkosivulla reaaliajassa. Nämä asetukset palautin opinnäytetyön jälkeen oletuksiin. Muutoksien jälkeen valittiin "DNS Settings", joka avasi hallintapaneelin domainin DNS asetuksille.

Lisäsin domainille kolme A recordia, jotka ohjaavat palvelimen IP-osoitteeseen. Ensimmäiselle annoin arvoksi www, toiselle oman domainin eli "verkkotunnus.com" ja viimeiselle tähtimerkin eli "*". Kaikkiin laitettiin sama IP-osoite, joka oli saatu Oracle Cloud -palvelusta. Poistin myös käytöstä Cloudflare Proxyn, koska sitä ei tarvita tässä työssä. TTL-asetus säätää kuinka pitkään yksi merkintä on voimassa. Se jätettiin automaattiseksi, koska minulla ei ollut syytä muuttaa sitä. Kun muutokset oli tallennettu, oman domainin pitäisi tulla näkyviin, kun sitä haetaan selaimesta. Muutoksissa kesti hetki, mutta pienen ajan kuluttua ne olivat tulleet voimaan.

7.3 SSL-sertifikaatti

Testasin sivun toiminnan hakemalla omaa domainia verkkoselaimella. Koska testisivu oli näkyvässä, pystyttiin aloittamaan SSL-sertifikaatin hankkiminen. SSL tulee sanoista Secure Sockets Layer, mutta nykyisin se on kehittynyt uuteen TLS eli Transport Layer Security muotoon. SSL/TLS on protokolla, jonka avulla voidaan salata liikenne oman laitteen ja palvelimen välillä. (Digicert, n.d)

7.3.1 Yleistä

SSL-sertifikaatti sisältää domainin, esimerkiksi "verkkotunnus.com", kenelle sertifikaatti on annettu, kuka on antanut sertifikaatin, antajan digitaalinen allekirjoitus, alidomainit kuten "demo.verkkotunnus.com", sertifikaatin voimassaoloaika ja julkisen avaimen, jota käytetään liikenteen salaamiseen.

On tärkeää, että verkkosivulla on SSL-sertifikaatti, jotta voidaan pitää käyttäjän tiedot turvassa, sekä varmistaa sivun omistaja. Varmistamalla sivun omistaja pahantekijöiden on vaikeampi luoda valheellista versiota sivusta. SSL-

sertifikaatilla varmistetaan, että sivu käyttää HTTPS yhteyttä. HTTPS on HTTP:n suojattu muoto. Useimmat selaimet ilmoittavatkin käyttäjälle sivuston olevan ”Ei turvallinen”, kun avataan sivu, joka ei käytä HTTPS yhteyttä.

HTTPS yhteys ja SSL-sertifikaatti voidaan varmistaa lukon kuvasta, joka yleensä sijaitsee selaimen vasemmassa yläkulmassa. Sertifikaattia voidaan tutkia painamalla lukon kuvaa. SSL-sertifikaatti pitää hankkia luotetulta ”Certificate Authority” eli CA:lta, joka myöntää ja varmistaa sertifikaatit. Palvelut kuten Cloudflare, DigiCert ja Let’s Encrypt ovat sertifikaatin myöntäjiä. Monien sertifikaatin myöntäjien palvelut ovat maksullisia, mutta esimerkiksi Let’s Encrypt myöntää sertifikaatteja ilmaiseksi. (DigiCert, n.d)

7.3.2 Tarvittavat työkalut

Käytin verkkosivun SSL-sertifikaatin hankkimiseen Let’s Encrypt palvelua ja työkalua nimeltä Certbot. Vaikka Cloudflare tarjoaa ilmaisen sertifikaatin, sitä ei käytetä, koska halusin selvittää, miten sertifikaatti hankitaan, jos sitä ei ole valmiiksi annettu. Certbot hankkii sivulle automaattisesti SSL-sertifikaatin, sekä päivittää sen 60 päivän väliajoin. Se turvaa, ettei sivun sertifikaatti pääse umpeutumaan. Työkalu rekisteröi sivun Let’s Encrypt palveluun ja pyytää sieltä sertifikaattia. Se myös päivittää tehdyn Nginx-konfiguraation HTTPS-yhteyttä varten. (Certbot, n.d)

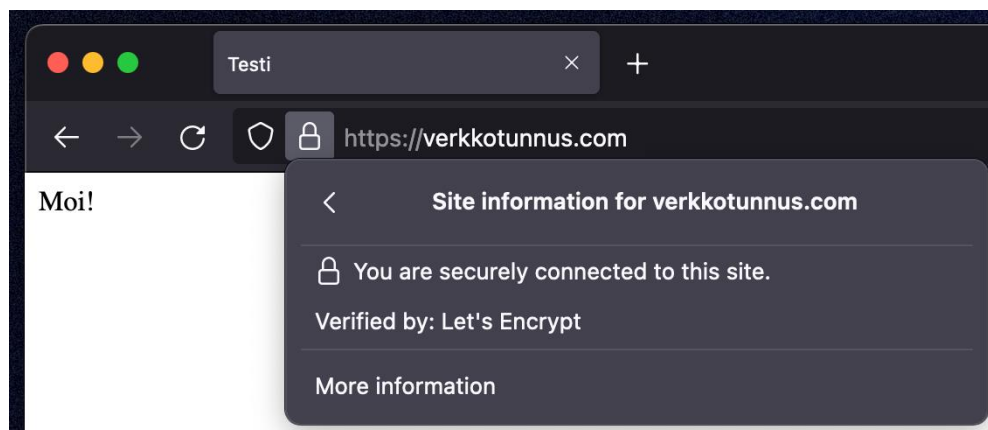
7.3.3 Certbotin asennus

Asensin Certbotin palvelimeen komenolla ”sudo apt install certbot python3-certbot-nginx”. (Boucheron, 2020) Sallittiin vielä HTTPS-yhteys palomuurista komennolla ”sudo iptables -I INPUT 6 -m state –state NEW -p tcp –dport 443 -j ACCEPT”. Komento oli sama, jonka avulla avasin portin 80 aikaisemmin. Nyt vain muokkasin komentoa niin, että se avaa portin 443. Sen jälkeen vielä tallennettiin muutokset komennolla ”sudo netfilter-persistent save”.

7.3.4 SSL-sertifikaatin hankinta

Nyt pystyin hankkimaan SSL-sertifikaatti Certbotin avulla. Ensimmäisellä kerralla Certbot kysyi sähköpostia, sekä pyysi hyväksymään käyttöehdot. Tämän prosessin jälkeen työkalu pyysi Let's Encrypt palvelusta SSL-sertifikaattia. Kun Certbot oli suorittanut prosessin onnistuneesti, sain ilmoituksen "Congratulations!". Testasin vielä Certbotin ajastustoiminnon olevan päällä komennolla "sudo systemctl status certbot.timer".

Koska portit oli jo avattu aikaisemmin palvelimesta ja Oracle Cloudista, sivulla pitäisi olla SSL-sertifikaatti. Testasin toimivuuden hakemalla oman domainin verkkoselaimella (Kuva 13). Nyt minulla oli valmis pohja tulevaa sivua varten. Uuden verkkosivun tiedostot siirretään palvelimelle myöhemmin verkkosivukansioon. Siellä sijaitsee vielä aikaisemmin tehty index.html, jossa on otsikko "Testi" ja elementti, jonka sisällä lukee teksti "Moi!". Seuraavaksi aloitin tekemään oikeaa sivua omalla koneellani, käyttämällä React JavaScript kirjastoa ja Viteä.



Kuva 13. Esimerkki sertifikaattista verkkosivulla.

8 VERKKOSIVUN RAKENTAMINEN

Kun tein päätöksen, mitä käytän verkkosivun tekemiseen, vertasin kahta vaihtoehtoa. React ja Vue. Vaikka ne jakavatkin ominaisuuksia keskenään, on niiden välillä kuitenkin eroja, jotka voivat olla tärkeämpiä johonkin tiettyyn tarkoitukseen. On tärkeää valita oikea työkalu tarkoitukseen. Valintaan vaikuttavat myös aiemmat kokemukset. On helpompi vain lähteä suunnittelemaan ja toteuttamaan, kun työkalu on jo ennestään tuttu.

8.1 ReactJS

ReactJS on yksi suosituimmista JavaScript kirjastoista, sen avulla voidaan tehdä verkko- ja mobiili sovelluksia. Se tarjoaa kehittäjille JSX-syntaksin, jota käytetään React sovellusten tekemiseen. JSX helpottaa vaikeidenkin elementtien rakentamista.

React on tehokas ja joustava kirjasto tehty käyttöliittymien rakentamiseen. Se perustuu komponenttipohjaan, sen avulla voidaan luoda modulaarisia käyttöliittymiä. Komponentteja voidaan uudelleen käyttää eri paikoissa käyttöliittymää. Näissä käyttöliittymissä voidaan näyttää dynaamisesti muuttuvaa tietoa. (Jordana, 2022)

React on kehitetty Facebookissa ja se julkistettiin maailmalle avoimen lähdekoodin kirjastona vuonna 2013. Sen käyttö mahdollistaa suurienkin verkkosovellusten luomisen. Sovellukset voivat muuttaa tietojaan ilman sivun latautumista uudelleen. Reactin tarkoitus on olla nopea, helposti laajennettavissa ja yksinkertainen. Sitä on myös mahdollista käyttää muiden JavaScript kirjastojen kanssa. (Arancio, 2021)

8.1.1 React-komponentit

Koska React koostuu komponenteista, voidaan komponentteja sisällyttää toisten komponenttien sisään. Se mahdollistaa monimutkaisten sovelluksien

tekemisen pienemmistä yksinkertaisista komponenteista. React komponentit voivat ylläpitää sisäistä tilaa kuten numeroa, joka kertoo kuinka monta kertaa, esimerkiksi nappia on painettu (Kuva 14). Numero-muuttujaa voidaan muokata ilman sivun uudelleenpäivittämistä. (Jordana, 2022)

The image shows a code editor window with the following code:

```

1  import { useState } from 'react';
2
3  export default function MyApp() {
4    return (
5      <div>
6        <h1>Tila</h1>
7        <Nappi />
8      </div>
9    );
10 }
11
12 function Nappi() {
13   const [count, setCount] = useState(0);
14
15   function handleClick() {
16     setCount(count + 1);
17   }
18
19   return (
20     <button onClick={handleClick}>
21       Painettu {count} kertaa
22     </button>
23   );
24 }

```

The browser window on the right shows the rendered application at <https://1ghtss.csb.app/>. The page title is "Tila" and it displays a button that says "Painettu 1 kertaa".

Kuva 14. React komponentin sisäinen tila

8.1.2 Virtuaalinen DOM

React toteuttaa virtuaalisen DOM:in, joka on JavaScript versio tavallisesta DOM:ista. Kun sitä pitää lukea tai sinne pitää kirjoittaa, React käyttää sen virtuaalista DOM:ia. Virtuaalinen DOM yrittää päivittää selaimen oikean DOM:in mahdollisimman tehokkaasti. Reactin DOM pitää huolen tavallisten DOM-elementtien päivittämisestä, jotta ne vastaavat niiden React versiota. (Jordana, 2022)

8.1.3 React Native

React tarjoaa myös natiivisia kirjastoja mobiilikäyttöjärjestelmille kuten Android ja iOS. Sitä kutsutaan React Nativeksi. Se eroaa hiukan syntaksiltaan tavallisesta Reactista, mutta perusteet ovat samat. React Native käyttää

verkkoelementtien sijasta käyttöliittymän omia elementtejä kuten muut sovellukset, jotka on tehty suoraan laitteelle. (Jordana, 2022)

8.2 Vue.js

Vue.js on Reactin tapaan yleisimmin käytetty käyttöliittymien rakentamiseen. Se voidaan myös lisätä jo olemassa oleviin projekteihin. Vue on yksi helpommista aloittaa, sillä se ei vaadi paljon oppimista tai muuten vaikeaa uutta syntaksia. Vuen on kehittänyt Evan You, joka halusi tehdä kevyemmän Frameworkin verrattuna jo saatavilla oleviin vaihtoehtoihin, kuten React. Vue esiintyi ensimmäisen kerran vuonna 2014. (Corbo, 2022)

8.2.1 Yhtenäisyydet ja erot Reactiin

Vuessa myös käytetään komponenttipohjaista tyyliä ja virtuaalista DOM:ia käyttöliittymien ja responsiivisten sovellusten rakentamiseen, samoin kuten Reactissa. Vue myös tukee JSX-syntaksia, jos sitä halutaan käyttää. Normaalisti kuitenkin Vue kuitenkin erottaa CCS:n, HTML ja JavaScriptin. Vuen komponentit ovat Reactin komponentteihin verrattuna yksinkertaisempia. Vuella voidaan myös rakentaa komponentteja, joita voidaan käyttää muualla kuten Reactissa tai jopa ilman mitään kirjastoa tai JavaScript frameworkia.

8.2.2 JavaScript Framework

JavaScript Framework on työkalu, joka helpottaa ja parantaa työskentelyä JavaScriptin kanssa. Koska kaikki Vue-komponentit eivät vaadi laajaa tietämystä eri kirjastoista tai esimerkiksi JSX-syntaksista, se helpottaa niiden jakamista muille kehittäjille, sillä komponenttien ymmärtäminen ei vaadi suurta työtä (Corbo, 2022). Toisin kuin React, Vuen takana ei ole suurta organisaatiota. Se ei kuitenkaan tarkoita, sitä että se olisi huonompi. Vueta tukee iso yhteisö kehittäjiä ja sen sponsorit kuten Storyblok ja Chrome Frameworks Fund. (Vue, n.d)

Vue on oikea JavaScript framework toisin kuin React. Vue tavoittelee nopeutta ja keveyttä. Vaikka React ei olekaan oikea framework vaan kirjasto, se tarjoaa samankaltaisia ominaisuuksia ja siinä mielessä sitä pidetäänkin frameworkinä. Vue tekee myös siirtymisen Reactista helppoa, sillä kuten aiemmin mainittiin, se tukee JSX-syntaksia.

8.3 Lopputulos

React ja Vue eivät ole ainoat vaihtoehdot, kun aletaan rakentamaan verkkosovellusta. Sen kanssa kilpailevat muun muassa Svelte ja Next.js. (Technostacks, 2023) Kuitenkin React ja Vue ovat yhdet suosituimmista vaihtoehdoista, kun aletaan luomaan uusia sovelluksia. Päädyin lopulta valitsemaan verkkosivun tekemiseen React JavaScript -kirjaston, sillä se on hyvin suosittu ja monet yrityksetkin käyttävät sitä omissa sovelluksissa ja verkkosivuissaan.

8.4 Projektin aloitus

Katkaisin palvelimen etäyhteyden ja siirryin omalle koneelleni valmistelemaan verkkosivu-projektia, koska suoritan verkkosivun kehittämisen omalla koneellani. Käytin verkkosivun projektin ylläpitoon Vite.js-työkalua, joka helpottaa sen ylläpitoa ja aloitusta. Viten on tehnyt sama henkilö, joka on Vuen takana. Vite tarjoaa nopean kehitysalustan ja tukee normaalia JavaScriptiä, Vueta, Reactia, Preactia, Litiä ja Svelteä. Ennen projektin aloitusta piti kuitenkin asentaa Node.js. (Vite, n.d.)

8.4.1 Node.js

Node.js on avoimenlähdekoodin JavaScript ajoympäristö, jonka avulla voidaan suorittaa JavaScript-koodia. Node.js mahdollistaa JavaScriptin käyttämisen muuallakin kuin selaimessa. Se perustuu Chromen V8 JavaScript Engineen. Noden mukana tulee pakettienhallintaohjelma npm. Sen avulla voidaan asentaa ja hallinnoida tarvittavia paketteja kuten Viteä. (Semah, 2022.)

Node.js asennettiin sen verkkosivulta: <https://nodejs.org/en>. Latasin omalle käyttöjärjestelmälleni tarkoitetun paketin. Node.js ympäristön asennuksen jälkeen voidaan sen versio tarkistaa terminaalista, jos käyttöjärjestelmänä on macOS, Linux tai Windows 11. Jos Windows-terminaalia ei ole asennettu, voidaan tarkistaa versio PowerShell tai komentokehote-työkaluista. Versio tarkistettiin komennolla "node --version". Koska Node.js oli asennettuna, sain vastauksena asennetun version numeron.

8.4.2 Vite-projekti

Uusi Vite-projekti luotiin komennolla "npm create vite@latest". Lisäsin komentoon vielä kansion, joka tulee olemaan projektin kansio. Tapauksessani komento oli "npm create vite@latest /Desktop/website", joka loi projektini tietokoneen työpöydälle ja sinne uuden kansion nimeltä "website". Vite-työkalu kysyi, minkälainen projekti halutaan tehdä. Valitsin projektiksi "desktop-website", frameworkiksi "React" ja lopuksi valitsin "variant" JavaScript (Kuva 15).

```
-> % npm create vite@latest /Desktop/website
✔ Package name: ... -desktop-website
✔ Select a framework: > React
✔ Select a variant: > JavaScript

Scaffolding project in /Users/matias/Desktop/website...

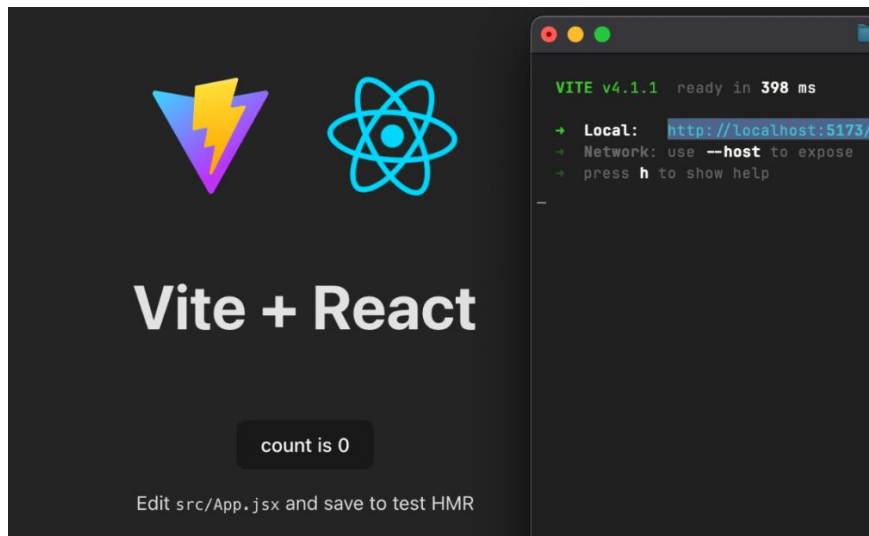
Done. Now run:

  cd Desktop/website
  npm install
  npm run dev
```

Kuva 15. Vite-projektin aloitus

Koska tein uuden projektin kansioon työpöydälle, piti vielä siirtyä sinne käyttämällä "cd"-komentoa. Myös Vite antoi ohjeet siihen. Ennen kuin pystyin aloittamaan verkkosivun tekemisen, piti asentaa projektin tarvitsemat moduulit. Moduulit asennettiin komennolla "npm install". Asennuksen jälkeen pystyin käynnistämään kehitysympäristön komennolla "npm run dev". Tässä vaiheessa verkkosivu oli valmis kehitykseen. Komento antoi vastauksena "Local"-osoitteen, jonka syötin verkkoselaimeen saadakseni live-näkymän

kehitettävästä verkkosivusta (Kuva 16). Muutokset sivuun tulivat näkyviin reaaliajassa, kun muokkasin projektin tiedostoja.



Kuva 16. Vite oletussivu

8.5 Projektin muokkaaminen

Projektikansion tiedostot voitiin avata koodineditointiohjelmalla tai tekstineditorilla. Tiedostoja oli mahdollisuus myös muokata yksitellen. Esimerkiksi nanoa käyttäen on mahdollista muokata projektin tiedostoja. Itse käytän tiedostojen muokkaamiseen Vimiä. Vim toimii terminaalista käsin ilman hiirtä. Sen avulla pystyn muokkaamaan tiedostoja kätevästi. Toinen vaihtoehto olisi kuitenkin Visual Studio Code, joka on Microsoftin tekemä avoimenlähdekoodin koodineditori.

8.5.1 Valmiin pohjan poistaminen

Aloitin poistamalla projektista valmiin pohjan, joka sisältää Viten ja Reactin logot. Poistin pohjan koska sille ei ole tarvetta. Poistin kuvat "React.svg" ja "Vite.svg", sekä tiedoston "App.css". Muokkasin myös tiedostoja "index.css", "App.jsx" ja "index.html". Tiedosto "index.css" tyhjätettiin ja lisäsin sinne merkinän, joka muuttaa "body"-elementin marginaalin nolnaan. Muokkasin vielä

tiedostoa "App.jsx" siten, että se ei sisällä "App.css"-tiedostoa vaan sisältääkin tiedoston "index.css".

Tämän avulla sain kaikki CSS-tyylit yhteen tiedostoon. Poistettiin myös turhat linjat, jotka ovat "div"-elementin "App" sisällä. Myöskään muuttujia "count" ja "setCount" ei tarvittu. Tiedosto "index.html":stä saatiin myös poistaa linja, joka sisältää linkin kuvaan. Sekä muutin verkkosivun "title"-elementin sisältämään oman otsikon. Näiden muutoksien jälkeen sivu oli tyhjä ja valmiina oman verkkosivun kehitykseen (Kuva 17).

The image shows three side-by-side screenshots of code editors. The first screenshot on the left shows the content of 'index.html', which is a basic HTML5 boilerplate with a title 'Website' and a script tag for 'main.jsx'. The middle screenshot shows 'src/App.jsx', a React component that imports 'useState' from 'react' and 'index.css', and renders a 'div' with 'className="App"' containing the text 'Hello!'. The third screenshot on the right shows 'src/index.css', which contains a single rule for 'body' with 'margin: 0;'. The file paths at the bottom of each editor are: '/Desktop/website/index.html', '/Desktop/website/src/App.jsx', and 'src/index.css'.

Kuva 17. Viten valmis pohja poistettuna

8.6 Oma verkkosivu

Tein sivusta kuvitteellisen valokuvaajan esittelysivun, jossa näkyvillä on valokuva ja yhteystiedot. Asensin projektiin uuden moduulin "react-router-dom". Sen avulla oli mahdollista määrittää sivun välilehdet ja niiden välillä siirtyminen. Ennen asennusta pysäytin Vite-prosessin, jonka jälkeen annoin terminaaliiin komennon "npm i react-router-dom", joka asensi moduulin.

Asennuksen jälkeen käynnistin Viten uudelleen komennolla "npm run dev". Otin asennetun moduulin käyttöön lisäämällä sen "main.jsx"-tiedostoon määrittämällä "import"-muuttujan, joka tuo "BrowserRouter"-komponentin. Tiedostossa "main.jsx", asetin "BrowserRouter"-komponentin niin, että se ympäröi "App"-komponentin.

```

1 import React from 'react'
2 import ReactDOM from 'react-dom/client'
3 import App from './App'
4 import './index.css'
5 import { BrowserRouter } from 'react-router-dom'
6
7 ReactDOM.createRoot(document.getElementById('root')).render(
8   <React.StrictMode>
9     <BrowserRouter>
10       <App />
11     </BrowserRouter>
12   </React.StrictMode>,
13 )
~
~/Desktop/website/src/main.jsx

```

Kuva 18. Tiedosto "main.jsx", jossa lisättyä "BrowserRouter"-komponentti

8.6.1 Navigaatio

Tein sivulle uuden komponentin, joka hoitaa navigoimisen. Sille luotiin uusi tiedosto "Navigation.jsx". Se sisältää linkit eri välilehdille verkkosivulla, sekä verkkosivun nimen. Komponentti on koko sivun mittainen ja pysyy paikallaan sivun yläreunassa, kun sivua selataan. Se on jaettu kahteen osaan, vasen- ja oikea puoli. Vasemmalla puolella on sivun otsikko, kun taas oikealla ovat linkit, joiden avulla suunnistetaan muihin välilehtiin.

Komponentille annetaan muuttujat "title", joka määrittää näytettävän otsikon ja "links", jolle annetaan kaikki linkit erotettuna pilkulla. Navigaatio-komponentti ottaa tekstin muuttujasta "links" ja jakaa sen listaan "navItems". Jokainen linkki sitten esitetään "button"-elementissä. Määritin myös elementeille "className" tiedot, että niitä voidaan hallita CSS-tiedostossa. Asetin CSS-tiedostossa tyylit navigaatiolle, että se näyttää paremmalta.

Lisäsin myös "Link"-komponentin asennetusta "react-router-dom"-moduulista. Sen avulla voitiin määrittää linkkejä sivun eri välilehdille. Otsikolle on määritetty linkki "/", se tarkoittaa etusivua. Muut linkit määrittyvät syötetyn "links" muuttujan mukaan. Rivillä 16 tarkistetaan ettei linkki ole "home". Jos linkki onkin

”home” se palauttaa tyhjän tekstin, jotta ”react-router-dom” osaa ohjata sen oikein etusivulle (Kuva 19).

```

1 import { Link } from "react-router-dom";
2
3 function Navigation(props) {
4
5   let navItems = props.links.split(',');
6
7   return (
8     <nav className="nav-bar">
9       <div className='nav-wrapper nav-left'>
10        <Link to='/'>
11          {props.title}
12        </Link>
13      </div>
14      <div className='nav-wrapper nav-right'>
15        { navItems.map((item, index) =>
16          <Link to={`/${item !== 'home' ? item : ''}`}>
17            <button key={index}>{item}</button>
18          </Link>
19        )}
20      </div>
21    </nav>
22  )
23 }
24
25 export default Navigation
~/Desktop/website/src/Navigation.jsx

```

Kuva 19. Valmis ”Navigation”-komponentti

8.6.2 Etusivu

Seuraavaksi tein komponentin, joka tulee olemaan verkkosivun etusivu. Etusivu-komponentille luotiin myös uusi tiedosto ”Home.jsx”, ja se ottaa muuttujat ”title”, ”sub” ja ”link”. Etusivu täyttää ruudun kuvalla ja se sisältää otsikon, alaotsikon, sekä linkin. Teksti näkyy ruudun vasemmalla puolella ja sen alapuolella on nappi, joka sisältää linkin.

Tein komponentille ”div”-elementin ja sen alle ”h1”, ”p” ja ”button”-elementit, jotka sisältävät annetut muuttujat. Määritin ne myös CSS-tyylittelyä varten. Tyylitiedostossa asetin tekstit ja napin ruudun vasemmalle puolelle ja annoin

niille eri fontit. Lisäsin projektiin ottamani kuvatiedoston, jota käytin etusivun taustakuvana (Kuva 20).

```
1 import { Link } from "react-router-dom";
2
3 function Home(props) {
4
5   return (
6     <div className="home-view">
7       <h1>{props.title}</h1>
8       <p className='home-sub'>{props.sub}</p>
9       <Link to={'/' + props.link}>
10        <button>{props.alt}</button>
11        </Link>
12      </div>
13    )
14  }
15
16 export default Home
~
~/Desktop/website/src/Home.jsx
```

Kuva 20. Valmis "Home"-komponentti

8.6.3 Yhteydenotto-välilehti

Tein vielä uuden komponentin "Contact", jolle loin tiedoston "Contact.jsx". Se sisältää "div"-elementin, jonka sisällä on "img", "a" ja "p"-elementit. Elementillä "img" määriteltiin kuva, joka annetaan komponentille muuttujana "img". Elementit "p" sisältävät muuttujat "name" ja "email". Elementtiä "a" käytetään linkkeihin. Määritin sen avaamaan sähköpostin, kun sitä painetaan. Se tapahtui antamalla "a"-elementille "href"-arvo: "mailto:", sekä sähköpostiosoite. Elementti "p", joka sisältää sähköpostin on laitettu "a"-elementin sisään, jotta linkki aktivoituu sähköpostia painettaessa. Sähköpostille on myös määritelty tyyli, jotta se erottuu nimestä (Kuva 21).

```

1
2 function Contact(props) {
3
4   return (
5     <div className="contact-view">
6       <img src={props.img}/>
7       <p className='contact-sub'>{props.name}</p>
8       <a href={'mailto:' + props.email}>
9         <p className='contact-email' style={{ color: 'grey', fontSize: '14px' }}>{props.email}</p></a>
10    </div>
11  )
12 }
13
14 export default Contact
~
~
~/Desktop/website/src/Contact.jsx 7,47

```

Kuva 21. Valmis "Contact"-komponentti

8.6.4 Uusien komponenttien lisääminen "App"-komponenttiin

Viimeisenä muokkasin "App"-komponenttia, jotta se sisältäisi aikaisemmin tehdyt "Contact", "Home" ja "Navigation"-komponentit. Määritin navigaatio komponentin ensimmäiseksi koska se pysyy ruudun yläreunassa paikallaan. Navigaation alle lisättiin "react-router-domia" käyttäen "Routes", jolla saatiin asetettua polut, joita loput komponentit käyttävät. Jokainen komponentti laitettiin "Route"-komponentin sisälle ja niille määrättiin oma polku. Komponentille "Home" määritin polun "/" ja komponentille "Contact" taas "/contact". Lisäsin myös aikaisemmin tehtyihin komponentteihin muuttujat kuten "title" ja "links", joita ne odottavat. Määritin myös kuvatiedoston "user", joka tapahtui määrittämällä sille "import" (Kuva 22).

```

6 import user from './assets/user.png';
7
8 function App() {
9
10  return (
11    <div className="App">
12      <Navigation title='Photography' links='home,contact' />
13      <Routes>
14        <Route exact path="/" element={ <Home title='Beautiful Photos' sub='From the surrounding nature and beyond' alt='Get in contact' link='contact' /> } />
15        <Route exact path="/contact" element={ <Contact img={user} name='Matti Meikäläinen' email='matti@example.com' /> } />
16      </Routes>
17    </div>
18  )
19 }
20
21 export default App
~/Desktop/website/src/App.jsx 15,65

```

Kuva 22. Valmis "App"-komponentti

8.6.5 Verkkosivun CSS

Muokkasin vielä CSS-tiedostoa, joka sisältää sivun tyyli. Tyyli määriteltiin HTML-elementtien luokkien eli ”className” tai elementin tyyppiin, kuten ”a” tai ”h1” mukaan. CSS-tiedostossa luokat merkitään käyttäen ”.”-merkkiä, esimerkiksi ”.nav-bar” määrittää komponentissani luokan navigaatiolle (Kuva 23). Luokkien nimet voivat olla melkein mitä tahansa. Jos CSS-tiedostossa merkitään pelkkä elementin tyyppi, kuten ”p”, se muokkaa kaikkien ”p”-elementtien tyyliä. Jos elementin edessä onkin luokan nimi ja väli, silloin muokataan ”p”-elementtejä, jotka ovat sen edessä mainitun luokan sisällä.

```

1  body {
2    margin: 0;
3    width: 100%;
4    height: 100vh;
5    font-family: 'Open Sans', sans-serif;
6    color: #fff;
7  }
8
9  a {
10   text-decoration: none;
11   color: #000;
12 }
13
14 .nav-bar {
15   margin: 0;
16   top: 0;
17   z-index: 999;
18   position: fixed;
19   background-color: transparent;
20   display: flex;
21   flex-direction: row;
22   height: 50px;
23   width: 100%;
24   box-sizing: border-box;
25   padding: 10px;
26   padding-right: 20px;
27   padding-left: 20px;
28   transition: all .5s ease;
29 }
30
31 .nav-bar:hover {
32   -webkit-box-shadow: 0px 1px 5px 0px rgba(0,0,0,0.72);
33   -moz-box-shadow: 0px 1px 5px 0px rgba(0,0,0,0.72);
34   box-shadow: 0px 1px 5px 0px rgba(0,0,0,0.72);
35 }
36
37 .nav-app-div {
38   height: 100%;
39   width: 50%;
40 }
41
42 .nav-right {
43   display: flex;
44   flex-direction: row;
45   justify-content: flex-end;
46 }
47
48 .nav-left {
49   font-weight: bold;
50   line-height: 30px;
51   margin: 0;
52   vertical-align: middle;
53   text-align: left;
54   font-family: 'Poppins', sans-serif;
55   text-transform: uppercase;
56 }
57
58 .nav-bar button {
59   outline: none;
60   border: none;
61   background-color: transparent;
62   height: 100%;
63   text-align: center;
64   vertical-align: middle;
65   cursor: pointer;
66   box-sizing: border-box;
67   margin-left: 20px;
68   color: #000;
69   transition: all .5s ease;
70 }
71
72 .nav-bar button:hover {
73   text-decoration: underline;
74 }
75
76 .home-view {
77   width: 100%;
78   height: 100vh;
79   position: absolute;
80   z-index: 1;
81   background-image: url('../src/assets/bg.jpg');
82   background-position: center;
83   background-repeat: no-repeat;
84   background-size: cover;
85 }
86
87 .home-bg {
88   width: 100%;
89   height: 100vh;
90   position: fixed;
91   z-index: -1;
92   object-fit: cover;
93 }
94
95 .home-view h1 {
96   position: absolute;
97   top: 35%;
98   left: 5%;
99   z-index: 0;
100  font-family: 'Poppins', sans-serif;
101  text-transform: uppercase;
102  font-size: 50px;
103 }
104
105 .home-view p {
106   position: absolute;
107   top: 45%;
108   left: 5%;
109   color: lightblue;
110   font-style: italic;
111 }
112
113 .home-view button {
114   position: absolute;
115   top: 52%;
116   left: 5%;
117   outline: none;
118   border: none;
119   background-color: #fff;
120   color: #000;
121   padding: 10px;
122   padding-left: 15px;
123   padding-right: 15px;
124   cursor: pointer;
125   border-radius: 10px;
126   opacity: 0.6;
127   transition: all .5s ease;
128 }
129
130 .home-view button:hover {
131   opacity: 1;
132 }
133
134 .contact-view {
135   width: 100%;
136   height: 100vh;
137   display: flex;
138   flex-direction: column;
139   align-items: center;
140   justify-content: center;
141   color: #000;
142 }
143
144 .contact-view img {
145   height: 100px;
146   width: 100px;
147   border-radius: 50%;
148   margin-bottom: 25px;
149 }
150
151 .contact-view p {
152   margin: 0;
153   margin-bottom: 5px;
154 }

```

Kuva 23. Valmis CSS-tiedosto

8.6.6 Projektin viimeistely

Verkkosivu on nyt valmiina ja toiminta oli mahdollista testata Viten ylläpitämästä kehitysympäristöstä. Verkkosivu näkyi selaimesta, kun syötettiin sinne Viten antama osoite. Minulla osoite oli ”http://localhost:5173/”. Koska sivu toimii niin kuin halusin, voin käyttää Viten ”build”-työkalua verkkosivun viimeistelyyn.

Sammutin Viten kehitystilän ja käytin komentoa ”npm run build”. Tämä komento suoritti Viten ”build”-toiminnon, joka rakentaa verkkosivun. Vite loi uuden kansion ”dist”, joka sisältää verkkosivun tiedostot. Nämä tiedostot lähetän

palvelimelle oman domainin "html"-kansioon, jossa on vielä aikaisemmin tehty testisivu. Uuteen "dist"-kansioon tulleet tiedostot pystyin lähettämään palvelimelle käyttäen SFTP eli Secure File Transfer-protokollaa.

8.7 Projektin lähettäminen palvelimelle SFTP:n avulla

SFTP on turvallinen tapa siirtää tiedostoja oman ja palvelimen välillä. Se käyttää hyödyksi IP-osoitteita yhdistettäessä palvelimeen. Toisin kuin vanhempi FTP eli File Transfer Protokolla, SFTP tarjoaa salatun ja turvallisemman tavan siirtää arkaluontoisempiakin tiedostoja.

8.7.1 Yleistä

SFTP toimii portissa 22, joka on sama portti kuin SSH-protokollalla. SFTP on integroitu SSH-protokollaan, joka tarjoaa todennuksen ja salauksen tiedostojen siirroille. Tämä tarkoittaa sitä, että SFTP käyttää samoja todennusmenetelmiä kuin SSH. Esimerkiksi käytössä on todistus avaimilla tai salasanoilla. Se varmistaa, että vain valtuutetut käyttäjät saavat oikeuden tiedostoihin. (Singh, n.d)

8.7.2 SFTP:n edut ja ominaisuudet

SFTP tarjoaa monia suojausominaisuuksia kuten siirrettävien tiedostojen salauksen, sekä se estää tiedostojen luvattoman käytön. Se tarkistaa myös tiedostojen eheyden tarkistuksen. Tiedostojen eheys tarkistetaan, jotta ne eivät olisi vioittuneet siirron aikana. SFTP voi myös pakata tiedostoja pienempään tilaan siirron ajaksi.

Etuna on myös se, että SFTP tarjoaa helpon hallinnan tiedostoille. Tämä helpottaa järjestelmänvalvojen valvontaa tiedostoista ja käytäntöjen määrittämisestä. SFTP tarjoaa myös useita käteviä ominaisuuksia, kuten mahdollisuuden jatkaa keskeytyneitä tiedonsiirtoja ja siirtää monta tiedostoa kerralla. Se säästää aikaa ja parantaa tuottavuutta. (Singh, n.d)

8.7.3 SFTP:n toiminta

SFTP on yleensä valmiina asennettuna Linux ja macOS käyttöjärjestelmissä, siksi palvelimelle ei tarvitse asentaa sitä. Portteja ei myöskään tarvitse avata Oracle Cloud palvelusta, sillä kuten aiemmin mainittiin, SFTP käyttää samaa porttia kuin SSH. SSH-portti on valmiiksi avattu, koska sitä tarvitaan etäyhteyteen. Ennen tietojen siirtoa pitää kuitenkin konfiguroida SFTP-käyttäjä, jolle annetaan oikeus muokata verkkosivun tiedostoja.

8.8 SFTP:n konfigurointi

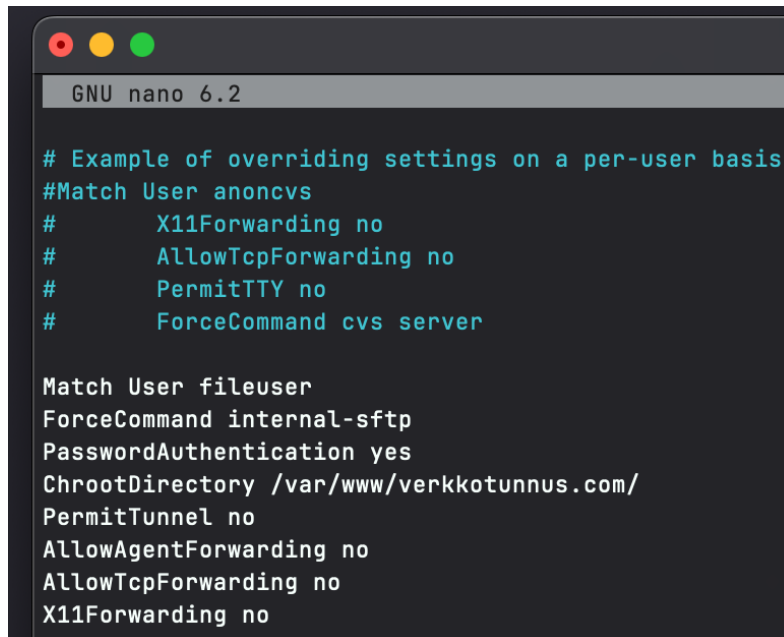
Avasin uudelleen etäyhteyden palvelimeen, ja käytin palvelimessa komentoa "sudo adduser", jotta saatiin luotua uusi käyttäjä. Tein uuden käyttäjän nimeltä "fileuser", joten käytin komentoa "sudo adduser fileuser". On tärkeää antaa käyttäjälle salasana, jotta yhdistäminen tapahtuu turvallisesti SFTP:n avulla.

Seuraavaksi uusi käyttäjä tarvitsi oikeuden verkkosivun kansioon. SFTP vaatii, että yhdistettävän kansion juuren omistaja on "root"-käyttäjä. (Horcasitas, 2021) Vaihdoin siis kansion "/var/www/" omistajaksi "root"-käyttäjän. Kyseinen toimenpide tapahtui komennolla "sudo chown -R root:root /var/www/". Lisäsin myös kirjoitus- ja lukuoikeudet komennolla "chmod 755 /var/www/". (Horcasitas, 2021)

8.8.1 SFTP käyttäjä ja oikeudet

Vaihdoin "fileuser"-käyttäjän verkkosivun kansion omistajaksi komennolla "sudo chown fileuser:fileuser /var/www/verkkotunnus.com/html/". Muokkasin vielä "sshd_config"-tiedostoa, jotta käyttäjällä on vain oikeus SFTP-yhteyteen, eikä esimerkiksi SSH-yhteyteen. Konfiguraatio-tiedostoa muokataan esimerkiksi aikaisemmin mainitulla nano-tekstineditorilla. Avasin tiedoston komennolla "sudo nano /etc/ssh/sshd_config". (Horcasitas, 2021)

Tein muutoksia tiedoston pohjalle, jossa määrittelin arvoja muuttujille: "Match User", "ForceCommand", "PasswordAuthentication", "ChrootDirectory", "PermitTunnel", "AllowAgentForwarding", "AllowTcpForwarding" ja "X11Forwarding" (Kuva 24). Tiedosto sisälsi jo valmiiksi ohjeita kommentteina, siitä miten muuttujien määrittely tapahtuu. Muokkauksen jälkeen tiedosto tallennettiin ja käynnistin sshd-prosessin uudelleen, jotta muokattu konfiguraatio latautuisi.



```
GNU nano 6.2
# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server

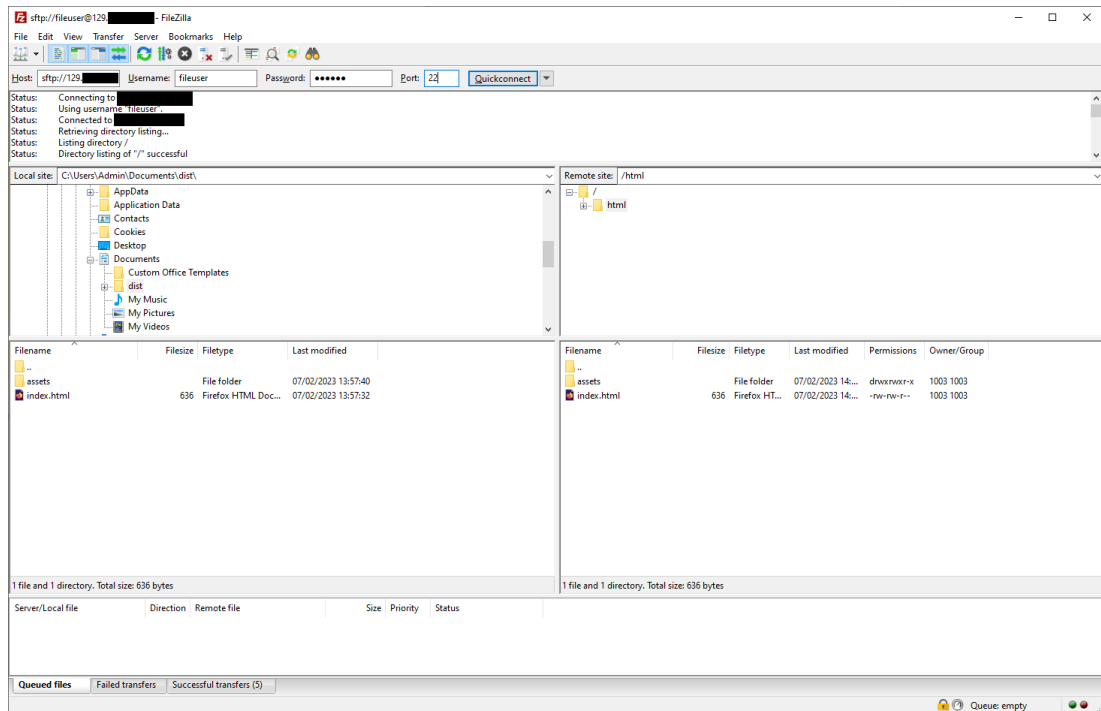
Match User fileuser
ForceCommand internal-sftp
PasswordAuthentication yes
ChrootDirectory /var/www/verkkotunnus.com/
PermitTunnel no
AllowAgentForwarding no
AllowTcpForwarding no
X11Forwarding no
```

Kuva 24. Muokattu "sshd_config"-tiedosto

8.9 SFTP-yhteyden muodostaminen

Nyt olin valmis siirtämään aikaisemmin Vitellä rakentamani verkkosivun tiedostot palvelimelle. Tiedostot pystytään siirtämään käyttämällä sftp komentoa terminaalista tai lataamalla ohjelma, joka tukee SFTP:tä. Käytin tiedonsiirtoon FileZilla-nimistä ohjelmaa (Kuva 25).

Se toimii Windows-, macOS- ja Linux -käyttöjärjestelmillä. FileZilla tukee eri tiedonsiirtoprotokollia, kuten SFTP ja FTP. Se tarjoaa käyttöliittymän tiedonsiirtoon ja siten helpottaa tiedonsiirtoa. Ohjelma voidaan ladata sen verkkosivuilta omalle käyttöjärjestelmälle: <https://filezilla-project.org/>

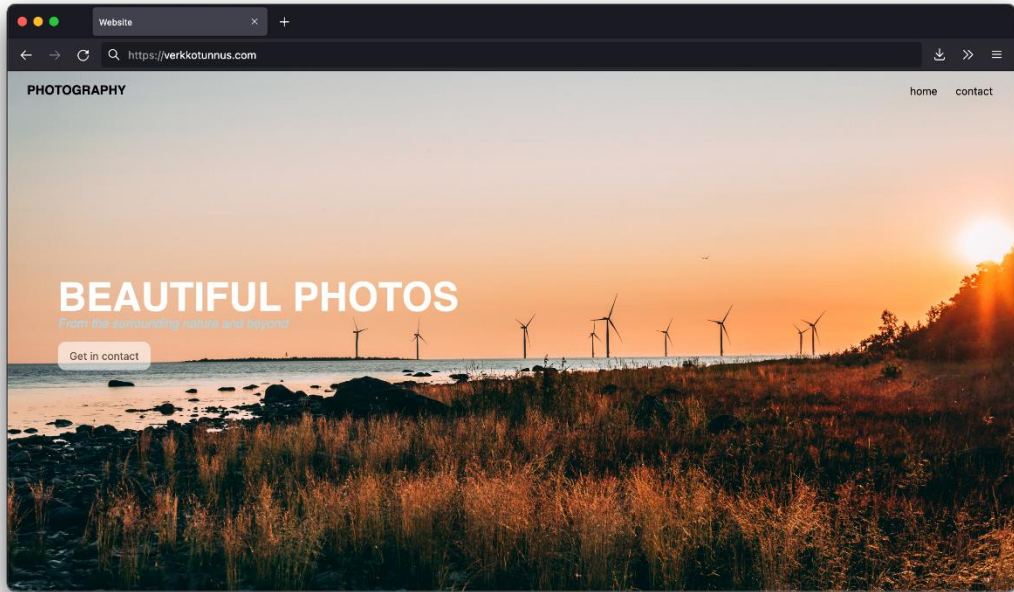


Kuva 25. FileZillan käyttöliittymä Windows-käyttöjärjestelmässä

8.9.1 Yhdistäminen käyttäen FileZillaa

FileZilla kysyy ennen yhdistämistä tietoja kuten "Host", "Username", "Password" ja "Port". Kohtaan "Host" syötettiin palvelimen IP-osoite, kun taas "Username" ja "Password" ovat aikaisemmin luodun käyttäjän tiedot. Itselläni käyttäjä oli "fileuser". Kohta "Port" taas määrittää portin, johon yhdistetään. Koska SFTP käyttää samaa porttia kuin SSH, asetin portin numeroksi 22.

Yhdistäminen tapahtui kohdasta "Quickconnect". Hallintapaneelin vasemmalla olivat oman tietokoneen tiedostot ja oikealla palvelimen tiedostot. Poistin palvelimelta vanhan testisivun eli "index.html"-tiedoston. Poistettuani tiedoston, siirsin omalta koneeltani Viten luoman kansion "dist" sisällön palvelimen "html"-kansioon raahaamalla ne vasemmalta puolelta oikealle. Kun siirto oli suoritettu, testasin verkkosivun toimivuutta hakemalla omaa domainia selaimella (Kuva 26).



Kuva 26. Valmis verkkosivu

9 POHDINTA

Opinnäytetyön tavoitteena oli tutustua ja selvittää, kuinka helppoa on tehdä alusta verkkosivulle, sekä mitä siihen tarvitaan. Tutustuttiin myös oman domainin hankintaan ja VPS-palveluihin. Työtä tehdessä, sitä helpottivat jo ennestään tutut alustat kuten, Cloudflare ja Oracle Cloud.

Työtä varten tehtiin myös verkkosivu, jossa käytettiin hyödyksi Viteä ja Reactia. Ne helpottivat verkkosivun rakentamista ja tarjosivat ominaisuuksia kuten "react-router-dom", jonka avulla pystyttiin hallitsemaan verkko-osoitteiden polkuja. Vite tarjosi myös hyvän kehitysympäristön ja työkalut verkkosivun rakentamiseen.

Opinnäytetyön tulokseksi saatiin toimiva verkkosivu, jossa on käytössä itsestään uusiutuva SSL-sertifikaatti ja toimiva SFTP yhteys, jonka avulla voidaan siirtää tiedostoja omalta laitteelta palvelimelle helposti. Työn tavoite toteutui mielestäni hyvin, sillä alusta on mainio staattiselle verkkosivulle ja siitä on helppo tulevaisuudessa laajentaa.

Asentamalla palvelimelle lisää palveluja, kuten työn alkuvaiheessa mainittu Nextcloud, pystytään saamaan palvelimelle lisää käyttötarkoituksia. Uusia palveluja voidaan sitten hallita Nginx-ohjelman avulla. Nginx mahdollistaa myös uusien domainien ja verkkosivujen lisäämisen palvelimelle.

LÄHTEET

Amazon Web Services (n.d.a) AWS Pricing Calculator Estimate the cost for your architecture solution. Haettu 1.2.2023 osoitteesta <https://calculator.aws/#/?nc2=pr>

Amazon Web Services (n.d.b) What is DNS? Haettu 7.12.2022 osoitteesta <https://aws.amazon.com/route53/what-is-dns/>

Arancio, S. (6.8.2021) ReactJS: A brief history <https://medium.com/@sjarancio/reactjs-a-brief-history-3c1e969a477f>

Boucheron, B. (20.5.2020) How To Secure Nginx with Let's Encrypt on Ubuntu 20.04 <https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-20-04>

Canonical (n.d.) Install and configure Nginx. Haettu 2.2.2023 osoitteesta <https://ubuntu.com/tutorials/install-and-configure-nginx#2-installing-nginx>

Carrigan, T. (11.2.2020) Linux command line basics: sudo <https://www.redhat.com/sysadmin/sudo>

Certbot (n.d.) About Certbot Haettu 15.2.2023 osoitteesta <https://certbot.eff.org/pages/about>

Cloudflare (n.d.a) What is a domain name registrar? <https://www.cloudflare.com/learning/dns/glossary/what-is-a-domain-name-registrar/>

Cloudflare (n.d.b) What is a DNS record? Haettu 17.2.2023 osoitteesta <https://www.cloudflare.com/learning/dns/dns-records/>

Cloudflare (n.d.c) What is HTTPS? Haettu 16.2.2023 osoitteesta <https://www.cloudflare.com/learning/ssl/what-is-https/>

Cloudflare (n.d.d) WHOIS redaction. Haettu 1.2.2023 osoitteesta <https://developers.cloudflare.com/registrar/get-started/whois-redaction/>

Corbo, A. (29.12.2022) What is Vue JS? <https://builtin.com/software-engineering-perspectives/vue-js>

Digicert (n.d) What is SSL, TLS and HTTPS? Haettu 18.2.2023 osoitteesta <https://www.websecurity.digicert.com/security-topics/what-is-ssl-tls-https>

Glass, E. (25.4.2020) How To Install Nginx on Ubuntu 20.04 <https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-20-04>

Hartman, S. (1.2.2023) How to start a website from A to Z: A 5-step guide. <https://www.godaddy.com/garage/how-to-start-a-website-from-a-to-z/>

Horcasitas, J. (1.10.2021) How To Enable SFTP Without Shell Access on Ubuntu 20.04 <https://www.digitalocean.com/community/tutorials/how-to-enable-sftp-without-shell-access-on-ubuntu-20-04>

Hostman (18.10.2021) What is a VPS? 4 tips on how to choose the best VPS <https://hostman.com/blog/what-is-vps/>

IBM. (n.d.) What is a VPS? Haettu 15.2.2023 osoitteesta <https://www.ibm.com/topics/vps>

Intel (n.d.) How Many Virtual Processors (vCPUs) are Within an Intel® Xeon® Processor? Haettu 2.2.2023 osoitteesta <https://www.intel.com/content/www/us/en/support/articles/000036898/processors/intel-xeon-processors.html>

Internet Corporation for Assigned Names and Numbers. (n.d) What Does ICANN Do? Haettu 31.1.2023 osoitteesta <https://www.icann.org/resources/pages/what-2012-02-25-en>

Jordana, A. (7.12.2022) What Is React & How Does It Actually Work? <https://www.hostinger.com/tutorials/what-is-react>

Rowland, C. (14.4.2021). How to Choose Your VPS Hosting Provider <https://www.hostgator.com/blog/choose-vps-hosting-provider/>

Reger, D. (25.8.2021). vCPU and OCPU pricing information. <https://blogs.oracle.com/cloud-infrastructure/post/vcpu-and-ocpu-pricing-information>

Megida, D (2.4.2020) Symlink Tutorial in Linux – How to Create and Remove a Symbolic Link <https://www.freecodecamp.org/news/symlink-tutorial-in-linux-how-to-create-and-remove-a-symbolic-link/>

Miller, D. (2021). What is a domain name? Everything you need to know about domain names. Haettu 7.12.2022 osoitteesta <https://www.godaddy.com/garage/what-is-a-domain-name/>

Nginx (n.d.) CommandLine Haettu 4.2.2023 osoitteesta <https://www.nginx.com/resources/wiki/start/topics/tutorials/commandline/>

Oracle (5.1.2023a). Adding Ingress Rules <https://docs.oracle.com/en-us/iaas/mysql-database/doc/adding-ingress-rules.html>

Oracle (4.1.2023b). Connecting to Your Instance. <https://docs.oracle.com/en-us/iaas/Content/GSG/Tasks/testingconnection.htm>

Oracle (18.1.2023c) Free Tier: Install Apache and PHP on an Ubuntu Instance. <https://docs.oracle.com/en-us/iaas/developer-tutorials/tutorials/apache-on-ubuntu/01oci-ubuntu-apache-summary.htm>

Organ, C. (21.1.2023) How To Register A Domain Name (2023 Guide). <https://www.forbes.com/advisor/business/how-register-domain-name/>

Schäferhoff, N. (n.d.) Web Hosting Statistics Market Share, Trends and Biggest Companies Haettu (1.2.2023) osoitteesta <https://websitesetup.org/news/web-hosting-statistics/>

Semah, B. (5.12.2022) What Exactly is Node.js? Explained for Beginners <https://www.freecodecamp.org/news/what-is-node-js/>

SSH (n.d.) SSH Protocol – Secure Remote Login and File Transfer. Haettu 3.2.2023 osoitteesta <https://www.ssh.com/academy/ssh/protocol>

Singh, A. (n.d) What is SFTP? Haettu 21.2.2023 osoitteesta <https://www.educba.com/what-is-sftp/>

Technostacks (12.1.2023) 17 Best JavaScript Frameworks For Robust Web Development <https://technostacks.com/blog/best-javascript-frameworks/>

Vite (n.d) Getting Started Haettu (20.2.2023) osoitteesta <https://vitejs.dev/guide/>

Vue (n.d) Become a Vue.js Sponsor Haettu 20.2.2023 osoitteesta <https://vuejs.org/sponsor/>