



Verkkopalvelinten automatisoitu konfigurointi Linux-ympäristössä Ansiblella

Ammattikorkeakoulun opinnäytetyö

Tietojenkäsittelyn koulutus

Syksy 2024

Arttu Altio

Tietojenkäsittelyn koulutus

Tekijä Arttu Altio

Työn nimi Verkkopalvelinten automatisoitu konfigurointi Linux-ympäristössä Ansiblella

Ohjaaja Tero Keso

Tiivistelmä

Vuosi 2024

Opinnäytetyön aiheena oli tutkia automatisointityökalu Ansiblen käyttöä palvelinten konfiguroinnissa. Tutkimuksen tavoitteena oli vertailla manuaalisesti konfiguroitujen palvelinten eroavaisuuksia Ansiblella konfiguroituihin. Tutkimuksessa selvitettiin muun muassa kummallakin tavalla konfigurointeihin käytettyä aikaa sekä työkalun mahdollisia haasteita ja hyötyjä verrattuna perinteiseen manuaaliseen konfigurointitapaan. Tutkimuksessa selvitettiin manuaalisesti konfiguroimalla tapahtuvia inhimillisiä virheitä, jotka automatisointityökalua käyttäen pystyttäisiin välttämään. Opinnäytetyöllä ei ollut ulkoista toimeksiantajaa, työ tehtiin opinnäytetyön tekijän omasta mielenkiinnosta syventyä aiheeseen ja tutkia itselleen vierasta automatisointityökalua.

Opinnäytetyön teoriaosassa perehdyttiin ensin palvelimiin, automatisointiin ja Linuxiin, joka toimii käyttöjärjestelmänä tutkimuksessa konfiguroiduille palvelimille. Lisäksi perehdyttiin myös Ansible-työkaluun omassa luvussaan. Opinnäytetyön teoriaosassa saatiin perustietoa palvelimista sekä niiden konfiguroinnista, automatisoinnista yleisesti, sekä automatisoinnin hyötyjä ja mahdollisia haittoja. Työssä selvitettiin myös yleistietoa Linuxista, ja miksi se käyttöjärjestelmänä sopii juuri palvelintarkoitukseen. Ansible luvussa syvennyttiin työkaluun ja sen arkkitehtuuriin. Tietopohja tutkimukseen kerättiin erilaisista luotettavista IT-alan yritysten tai henkilöiden verkkojulkaisuista. Tutkimuksessa menetelmänä käytettiin vertailevaa tutkimusmenetelmää.

Tutkimuksen tuloksena saatiin selville Ansible-työkalun käyttöön liittyviä haasteita sekä ajallista eroavaisuutta manuaaliseen konfigurointitapaan. Tutkimuksessa selvitettiin myös, milloin työkalun käyttöönotto palvelinten konfiguroinnissa olisi kannattavaa tutkimuksen tapaisessa ympäristössä. Tutkimus myös kehitti opinnäytetyön tekijän ongelmanratkaisutaitoja ja antoi käytännön kokemusta työkalun käytöstä.

Avainsanat Automatisointi, Linux, Ansible, Palvelin, Konfigurointi

Sivut 40 sivua

The topic of the thesis was to study the use of the automation tool Ansible in server configuration. The aim of the research was to compare the differences between manually configured servers and those configured with Ansible. The study investigated, among other things, the time spent on configuration using both methods, as well as the potential challenges and benefits of the tool compared to the traditional manual configuration method. The study also examined human errors that occur with manual configuration, which could be avoided by using an automation tool. The thesis did not have an external commissioner; the work was done out of the author's own interest in delving into the topic and exploring an unfamiliar automation tool.

In the theoretical part of the thesis, the focus was first on servers, automation, and Linux, which was the operating system used for the configured servers in the study. Additionally, the Ansible tool was covered in its own chapter. The theoretical part of the thesis provided basic information about servers and their configuration, automation in general, and the benefits and potential drawbacks of automation. The study also provided general knowledge about Linux and why it is suitable as an operating system for servers. The Ansible chapter delved into the tool and its architecture. The knowledge base for the research was gathered from various reliable online publications by IT companies or individuals. The research method used was a comparative research method.

The results of the study revealed the challenges associated with using the Ansible tool and the time differences compared to the manual configuration method. The study also identified when it would be beneficial to adopt the tool for server configuration in an environment similar to the study. The research also developed the author's problem-solving skills and provided practical experience with the tool.

Keywords Automation, Linux, Ansible, Server, Configuration

Pages 40 pages

Sisällys

1	Johdanto	1
2	Palvelimista yleisesti	2
2.1	Palvelinten kehitys	4
2.2	Palvelimen konfigurointi	5
2.3	Linux	5
2.3.1	Linux ja palvelimet	7
2.3.2	Linux-palvelimen yleisiä konfiguroinnin vaiheita.....	9
2.4	Verkkopalvelin	11
2.5	Automatisointi	12
2.5.1	Automatisoinnin hyödyt.....	13
2.5.2	Automatisoinnin mahdollisia haittoja	14
3	Ansible	15
3.1	Ansiblen historiaa.....	16
3.2	Ansiblen arkkitehtuuri.....	17
3.2.1	Moduulit.....	17
3.2.2	Inventory (varasto).....	19
3.2.3	Playbookit (pelikirjat)	19
3.2.4	YAML	20
4	Menetelmät	21
5	Tutkimus	22
5.1	Manuaalisen konfiguroinnin tulokset	25
5.2	Ansible-työkalun konfiguroinnin tulokset	26
6	Tulosten yhteenveto.....	32
7	Johtopäätökset.....	35
8	Jatkotutkimus	36
	Lähteet	37

Kuvat, taulukot ja kaavat

Kuva 1 Ensimmäinen inhimillinen virhe	25
Kuva 2 Toinen inhimillinen virhe	25
Kuva 3 Yksinkertainen pelikirja	27
Kuva 4 Komento, joka suorittaa pelikirjan inventory-tiedoston kanssa	27
Kuva 5 Pelikirja osa 1	29
Kuva 6 Pelikirja osa 2, loput suoritetuista vaiheista	30
Kuva 7 Inventory-tiedosto	31
Kuva 8 Konfigurointitapojen ajallinen vertailu	33
Kuva 9 Ansiblen käyttöönoton kannattavuusraja	34

1 Johdanto

Opinnäytetyön tarkoituksena on tutkia automatisointityökalu Ansiblen käyttöä Linux-palvelinten automatisoidussa konfiguroinnissa ja hallinnassa. Opinnäytetyön tarkoituksena on myös syventää opinnäytetyön tekijän osaamista automatisointiin sekä Linux-palvelimiin liittyen. Tutkimuksen ideana on saada ymmärrystä siihen, miten Ansible mahdollistaa automatisoidun palvelinten konfiguroinnin, tarjoten samalla hyödyllisiä näkemyksiä automatisoinnin käytöstä tietojärjestelmien hallinnassa. Tutkimuksen tarkoituksena on myös saada selville mitä inhimillisiä virheitä automatisointityökalulla pystytään välttämään.

Opinnäytetyön teoriaosuudessa perehdytään Ansible-työkaluun ja sen käyttöön palvelinten konfiguroinnissa. Työ sisältää tietoa myös palvelimista, Linuxista sekä automatisoinnista. Käytännön osuudessa toteutetaan vertaileva tutkimus, jossa tutkitaan manuaalisesti konfiguroitujen palvelinten eroavaisuuksia Ansiblella konfiguroituihin palvelimiin.

Opinnäytetyön aihe valikoitui opinnäytetyön tekijän omasta mielenkiinnosta tutkia Ansible-työkalun käyttöä sekä syventää osaamista automatisointiin ja Linux-palvelimiin liittyen. Automatisointi ja sen käyttö ovat jatkuvasti kasvussa, joten aihe on ajankohtainen.

Tutkimuskysymykset ovat:

- Mitkä ovat automatisoidun konfiguroinnin hyödyt verrattuna perinteisiin menetelmiin?
- Millä tavoin Ansible-työkalun käyttö konfigurointiprosessissa vaikuttaa palvelinten käyttöönottoon?
- Millaisia haasteita liittyy Ansible-työkalun käyttöön palvelinten konfiguroinnissa?

2 Palvelimista yleisesti

Palvelin on joko fyysinen tai virtuaalinen tietokone tai ohjelmistojärjestelmä, joka on erikoistunut tarjoamaan muille tietokoneille eli asiakkaille palveluita, tietoa tai resursseja verkossa. Yleisimpiä esimerkkejä palvelimen tarjoamista palveluista ovat verkkosivujen ja sähköpostien toimittaminen, tiedostojen säilyttäminen ja hallinta, sekä erilaisten sovellusten suorittaminen, kuten erilaiset verkkosivustot ja verkkokaupat. (Monteclaro, 2023; Paessler, n.d.)

Palvelinten toiminta perustuu asiakas-palvelin-malliin, joka yksinkertaistettuna tarkoittaa sitä, että asiakas pyytää palvelimelta tiettyjä palveluita tai resursseja ja palvelin puolestaan täyttää nämä pyynnöt. Hyvänä esimerkkinä on verkkosivu, johon asiakas haluaa siirtyä selaimella. Tällaisessa tapauksessa asiakas pyytää palvelimelta tietyn palvelun tässä tapauksessa verkkosivun avaamisen ja palvelin vastaa pyyntöön avaamalla kyseisen sivun asiakkaalle näkyväksi. Palvelin myös usein suorittaa lisätehtäviä asiakaspyyntöjen osana, kuten henkilötietojen varmistamisen, jotta saadaan vahvistettua asiakkaan oikeus käyttää resursseja tai nähdä pyydettyjä tietoja. (Monteclaro, 2023; Paessler, n.d.)

Palvelimet eroavat peruskäytössä olevista tietokoneista muutamilla tavoilla. Ne ovat optimoituja suorituskyvyltään, luotettavuudeltaan ja laajennettavuudeltaan käsittelemään samanaikaisesti tapahtuvia useita asiakaspalvelupyntöjä, esimerkiksi nettisivut, joissa saattaa olla suuri määrä samanaikaisia asiakkaita. Palvelimilla on keskeinen rooli tiedon järjestämisessä ja säilyttämisessä, jotta tiedon saavutettavuus, luotettavuus ja turvallisuus saadaan varmistettua. (Monteclaro, 2023; Paessler, n.d.)

Palvelintyyppejä on olemassa erilaisia ja niitä hyödynnetään moniin eri käyttötarkoituksiin. Alla lueteltuna yleisiä palvelintyyppejä ja niiden käyttötarkoituksia:

Verkkopalvelimet ovat yksi yleisimmistä palvelintyypeistä nykypäivänä. Ne ovat erikoistuneet ylläpitämään ja tarjoamaan sovelluksia ja tietoja internetissä tai organisaation sisäisissä verkoissa, tarjoten verkkosivuja tai muita verkkokohtaisia palveluita laitteille. Tunnettuja avoimen lähdekoodin verkkopalvelimia ovat Apache ja Nginx, joita käytetään maailmanlaajuisesti. (Indeed Editorial Team, 2023; Paessler, n.d.)

Sovelluspalvelimet yhdistävät käyttäjät ohjelmistosovelluksiin virtuaalisten palvelinyhteyksien välityksellä, mahdollistaen käyttäjille palvelimen tarjoamien sovellusten käytön ilman, että käyttäjän tarvitsee ladata omalle laitteistolleen mitään. Sovelluspalvelimet suorittavat sovelluksia asiakastietokoneiden sijaan jakamalla resursseja käyttäjien kesken ja poistamalla tarpeen ohjelmistojen asennukselle käyttäjien omille laitteille. Sovelluspalvelimet ovat ihanteellisia yrityksille, koska ne kykenevät tehokkaasti tarjoamaan sovellusdataa useille käyttäjille samanaikaisesti. Yleisiä sovelluspalvelimia ovat esimerkiksi sähköposti- ja DNS-palvelimet. Sähköpostipalvelimet vastaanottavat ja säilyttävät käyttäjän lähettämiä sähköposteja. DNS-palvelimet muuttavat laitteiden ihmisen ymmärrettäviä nimiä tietokoneen luettavaksi IP-osoitteiksi. (Indeed Editorial Team, 2023; Paessler, n.d.)

Tiedostopalvelimet säilyttävät ja jakavat tiedostoja, jotta useat asiakkaat tai käyttäjät voivat jakaa ja ladata palvelimelle tallennettuja tiedostoja. Tätä peruspalvelimen käyttöä esiintyy yleisesti organisaatioissa, missä useilla käyttäjillä on yhteinen tarve käyttää ja päästä käsiksi samoihin tiedostoihin. Tiedostojen sijainti yhdellä palvelimella tekee varmuuskopiointista yksinkertaisempaa ja lisää tietoturvaluottuutta. Sen sijaan että yritettäisiin varmistaa turvallisuus ja eheys jokaisen henkilökohtaiselle laitteelle, tiedostoja on käytännöllistä säilyttää yhdellä palvelimella. (Indeed Editorial Team, 2023; Paessler, n.d.)

Tulostuspalvelimet mahdollistavat tehokkaan tulostusten hallinnan ja jakelun. Yksittäinen keskitetty tulostuspalvelin, joka käsittelee tulostuspyyntöjä useilta käyttäjiltä, sen sijaan, että jokaisen työasemalla olisi oma erillinen tulostin, on resurssien kannalta tehokkaampaa sekä tietoturvaluottuudesta näkökulmasta parempi ratkaisu. Näin voidaan esimerkiksi hallita palvelimen käyttöoikeuksia ja seurata tapahtuneita tulostustehtäviä. Tulostuspalvelimet toimivat yhdistymällä paikallisiin palvelimiin etänä tulostaakseen organisaation tietoverkoston kautta. Osa nykypäivän korkealaatuisista tulostuspalvelimista omaavat sisäänrakennetun palvelimen, ja ovat valmiita liittymään tietoverkkoon, kun ne ovat asennettu organisaation toimiston alueelle. (Indeed Editorial Team, 2023; Paessler, n.d.)

2.1 Palvelinten kehitys

Osa varhaisimmista palvelimista oli joko suurtietokoneita tai minitietokoneita. Nämä varhaiset palvelimet olivat yhteydessä ns. tyhmiin päätteisiin (dumb terminal). Nämä päätteet eivät omanneet lainkaan laskentatehoa ja kaikki laskennalliset tehtävät suoritettiin palvelimilla. Päätteet ainoastaan vastaanottivat syötettä näppäimistöiltä tai kortinlukijoilta ja näyttivät laskennan tulokset näytöillä tai tulostimilla. (Paessler, n.d.)

Myöhemmässä vaiheessa palvelinten kehitystä ne muuttuivat yleensä yhdeksi tehokkaaksi tietokoneeksi, joka oli yhteydessä verkon kautta vähemmän tehokkaisiin asiakastietokoneisiin. Tässä palvelinmallissa asiakas sekä palvelin omistivat palvelimen laskentatehon, mutta tiettyjä vaativampia tehtäviä suoritettiin pelkästään palvelimilla. (Paessler, n.d.)

Teknologian edistyessä palvelinkäsite on laajentunut, sillä nykyajan palvelimet voivat olla joko ohjelmistoja fyysisillä tietokonelaitteilla tai virtuaalisia palvelimia, jotka toimivat pilvipalvelurakenteessa internetissä usein kolmansien osapuolten hallinnoimina. Tämän ansiosta palvelimet ovat nykyään entistä monipuolisempia ja skaalautuvampia. Nykyiset palvelimet voivat olla erikoistuneita tiettyihin tehtäviin, kuten sähköpostipalvelimet käsittelemään sähköpostin tallennusta ja jakelua, tai monipuolisia, kuten tiedosto- ja tulostinpalvelimet, jotka suorittavat erilaisia tehtäviä yhteydessä oleville asiakkaille. (Paessler, n.d.)

Virtualisoinnin tulo merkitsi merkittävää muutosta palvelinarkkitehtuurissa.

Virtuaalipalvelimet, jotka perustuvat hypervisoriohjelmistoon, mahdollistivat useiden virtuaali-instanssien luomisen yhdelle fyysiselle palvelimelle. Tämä lähestymistapa maksimoi laitteiston käytön ja joustavuuden. Laitteiston resurssit jaettiin useiden teräspalvelimien, verkkoon liitetyn tallennuksen ja virtalähteiden kesken. Yksittäisten palvelimien väliset rajat hämärtyivät tässä ympäristössä, kun alkoi olla vaikeampaa erottaa, missä yksi palvelin päättyi ja toinen alkoi. Tämä kehitys korostaa palvelinteknologian jatkuvaa hiomista ja sopeutumista tietotekniikan muuttuviin vaatimuksiin. (Paessler, n.d.)

2.2 Palvelimen konfigurointi

Palvelimen konfiguroinnilla tarkoitetaan prosessia, jossa palvelimen laitteisto, esimerkiksi tietokone sekä tarvittavat ohjelmistot asetetaan käyttökuuntoon. Tämä prosessi sisältää useita vaiheita, kuten käyttöjärjestelmän, käyttöäoikeuksien, verkko- ja turvallisuusasetusten, jaettujen resurssien sekä verkkopalveluiden määrittelyn. Palvelimen konfiguroinnissa on erittäin tärkeää olla huolellinen, jotta palvelimen ja koko verkon suorituskyky, vakaus ja turvallisuus saadaan varmistettua. Palvelimet ovat useiden nykypäivän liiketoimintojen ydin, ja ne muodostavat epäilemättä infrastruktuurien keskeisimmän osan. (Jones IT, 2023; Vaadata, 2022)

Palvelinten on toimittava jatkuvasti optimaalisella tasolla. Tämän saavuttamiseksi tarvitaan huolellista suunnittelua, asennusta ja etenkin konfiguraatiota sekä asetuksia. Palvelimelta ei saavuteta odotettua suorituskykyä, vaikka käytössä oleva laitteisto ja ohjelmisto olisivat kalliita, mikäli konfiguraatio ja vaadittavat asetukset eivät ole kunnossa. Jos palvelimen konfigurointia ei ole hoidettu asianmukaisesti, se voi mahdollistaa vakaville tietoturvariskeille, kuten hyökkäyksille, tietovuodoille tai palvelimen hallinnan menetykselle. Tällaiset virheet voivat johtaa palvelimen käyttökatkoihin, tehottomuuteen ja jopa liiketoiminnan lamaantumiseen. Puutteellinen konfiguraatio voi myös aiheuttaa tiedonsiirron ongelmia ja altistaa haavoittuvat sovellukset erilaisille hyökkäyksille. Oikein konfiguroitu palvelinympäristö on äärimmäisen tärkeä turvallisen ja kustannustehokkaan toiminnan varmistamiseksi. (Jones IT, 2023; Vaadata, 2022)

2.3 Linux

Linux on avoimen lähdekoodin käyttöjärjestelmä, joka sai alkunsa alun perin harrastuksena käyttöjärjestelmän kehittäjän Linus Torvaldsin toimesta vuonna 1991. Yliopistossa opiskeleva Torvalds halusi luoda ilmaisen avoimen lähdekoodin version maksullisesta MINIX-käyttöjärjestelmästä, joka perustui Unixin periaatteisiin ja suunnitteluun. Harrastuksena alkunsa saanut Linux on sittemmin tullut käyttäjämäärältään suurimmaksi käyttöjärjestelmäksi, ja se on eniten käytetty käyttöjärjestelmä julkisesti saatavilla olevista internet-palvelimista. Se on myös ainoa maailman 500 nopeimpien supertietokoneiden joukossa käytetty käyttöjärjestelmä. (Red Hat, 2023)

Linuxia julkaistaan GNU General Public License (GPL) -lisenssillä mikä tarkoittaa, että kuka tahansa halukas voi käyttää, tutkia, jakaa tai muokata ohjelmistoa. Muokattua ohjelmistoa voi myös halutessaan jakaa uudelleen tai jopa myydä, kunhan sen tekee saman GPL-lisenssin alaisena. Nämä tekijät erottavat Linuxin merkittävästi muista käyttöjärjestelmistä, esimerkiksi Windowsista, jonka lähdekoodi on suljettu ja lukittu, sekä toimitetaan sellaisenaan muuttumattomana. Monet muut käyttöjärjestelmävaihtoehdot ovat myös maksullisia, minkä takia Linuxin yksi parhaista puolista, eli avoin lähdekoodi korostuu. (Red Hat, 2023)

Koska Linux on vapaasti muokattavissa ja jaettavissa oleva avoimeen lähdekoodiin perustuva käyttöjärjestelmä, siitä on saatavilla useita erilaisia Linux-jakeluita. Linux-jakelut ovat käyttöjärjestelmien kokonaisuuksia, jotka perustuvat Linux-ytimeen ja sisältävät tarvittavat ohjelmistot, työkalut ja sovellukset. Nämä jakelut voivat vaihdella tarkoituksesta, käyttökohteista ja sisällöstä riippuen. Linux-jakeluihin kuuluvat tyypillisesti Linux-ydin, perustyökalut ja -ohjelmistot, käyttöjärjestelmän hallintaan liittyvät työkalut, sekä usein graafinen käyttöliittymä ja muita käyttäjäystävällisiä piirteitä. (Red Hat, 2023)

Linux-jakelut muodostavat käyttöjärjestelmiä, jotka koostuvat Linux-ytimestä ja usein myös paketinhallintajärjestelmästä. Jakeluita on olemassa vajaa tuhat erilaista, tarjoten monipuolisia vaihtoehtoja erilaisiin käyttötarpeisiin. Ne ovatkin teollisuudessa standardi verkkopalvelinten, sovellusten ja muiden vaativien työkuormien suorittamisessa. Linux-jakeluita on saatavilla laitteisiin laidasta laitaan, kuten tehokkaisiin supertietokoneisiin, kodinkoneisiin sekä henkilökohtaisiin tietokoneisiin ja muihin elektronisiin laitteisiin, joissa on sisäänrakennettu tietokoneohjelma. (Zenarmor, 2023)

Kaikki Linux-pohjaiset käyttöjärjestelmät sisältävät Linux-ytimen, joka on käyttöjärjestelmän peruselementti, jota ilman ei käyttöjärjestelmä toimi. Ytimen tehtävänä on hallita järjestelmän resursseja, kuten milloin, mitkä ja kuinka kauan prosessit voivat käyttää prosessoria sekä seurata mihin ja missä muistia käytetään. Ytimessä toimivat myös laitteistoajurit, jotka toimivat välittäjänä tai tulkkina laitteiston ja prosessien välillä. Ydin vastaa myös turvallisuudesta ja järjestelmäkutsuista vastaanottamalla palvelupyynnöt prosesseilta. Oikein toteutettu ydin on käyttäjälle näkymätön, kun se toimii omassa erillisessä ydintilaksi kutsutussa tilassaan. Tässä tilassa ydin esimerkiksi hallinnoi muistin varaamista pitäen kirjaa tallennussijainneista ja järjestelmän käyttäjä puolestaan havaitsee sovelluksia, kuten tiedostoja käyttäjätilassa. Tiedostot ja muut sovellukset puolestaan kommunikoivat järjestelmäkutsuliittymän avulla ytimen kanssa. (Red Hat, 2019; Zenarmor, 2023)

Linux-ytimen tehtävän selkeyttämiseksi Linux-konetta voidaan ajatella kolmena osana: laitteisto, Linux-ydin ja käyttäjäprosessit. Laitteisto muodostuu fyysisistä koneesta, muistista ja prosessorista. Ydin on muistissa sijaitseva ohjelmisto, joka on keskeisin osa järjestelmää. Se kertoo prosessorille mitä tehdä. Käyttäjäprosessit ovat ytimen hallinnoimia käynnissä olevia ohjelmia ja ne muodostavat käyttäjätilan. (Red Hat, 2019; Zenarmor, 2023)

2.3.1 Linux ja palvelimet

Linux-palvelimet ovat verkossa toimivia tietokonejärjestelmiä, jotka tarjoavat kustannustehokkaita ja monipuolisia alustoja useisiin eri käyttötarkoituksiin, kuten verkkopalvelimille, tietokantojenhallintaan sekä virtualisointiin. Linux-pohjaiset palvelimet ovat turvallinen ja luotettava vaihtoehto niin yrityksille, organisaatioille kuin yksityishenkilöillekin. Pilvipalvelujen yleistymisen myötä myös Linux-palvelimet ovat kasvattaneet suosiotaan verkkosivustojen, sovellusten ja datan isännöinnissä. Avoin lähdekoodi mahdollistaa käyttäjille ohjelmiston muokkaamisen tarpeidensa mukaan. Lähdekoodin avoimuus myös auttaa kustannuksia pysymään alhaisina tarjoten lisää joustavuutta palvelinten asennukseen, käyttöön sekä hallintaan. (MonoVM, 2024; LogicMonitor, 2023; Red Hat, 2022)

Linux-palvelimia on asennettu kaikista yleiskäyttöön tarkoitetuista käyttöjärjestelmistä eniten, ja ne ovat myös eniten käytetty käyttöjärjestelmä palvelimilla. Linux-palvelimet ovat yleensä tekstipohjaisia käyttöjärjestelmiä, eivätkä ne sisällä graafista käyttöliittymää. Eli palvelinten hallinta ja käyttö tapahtuu pääosin komentoriviltä. Tekstipohjaiset palvelinjärjestelmät ovat kevyempiä, ja niitä on mahdollista käyttää fyysisillä palvelimilla sekä pilvipalvelimilla. (MonoVM, 2024; LogicMonitor, 2023; Red Hat, 2022)

Linuxin yksi merkittävin hyöty palvelinkäytössä on sen turvallisuus. Se on suunniteltu turvallisuus ensisijaisena tavoitteenaan, ja siinä on useita tätä vahvistavia ominaisuuksia. Ylläpitäjillä on mahdollisuus hallita ja määritellä käyttäjätilien ja tiedostojen oikeuksia ja rajoituksia, esimerkiksi asettamalla vain luku- tai kirjoitusoikeuksia tietyille käyttäjille tiettyihin tiedostoihin ja hakemistoihin. Tällä pystytään estämään rajoitetuilta käyttäjiltä luvottomien ohjelmien suorittamista palvelimilla. Linux tukee myös useita todennusmenetelmiä turvallisuuden varmistamiseksi, kuten perinteistä käyttäjätunnus ja salasana yhdistelmää, sekä edistyneempiä menetelmiä, esimerkiksi älykortteja, sekä biometrisiä ja digitaalisia sertifikaattivarmenteita. Nämä menetelmät tuovat lisäsuojan varmistaakseen tarkasti jokaisen käyttäjän henkilöllisyyden, ennen kuin pääsy myönnetään palvelimelle, tietoihin tai resursseihin. (MonoVM, 2024; LogicMonitor, 2023; Red Hat, 2022)

Avoim lähdekoodi vaikuttaa merkittävästi myös Linuxin turvallisuuden kestävyteen. Suljetuissa ja omistajayhtiöiden hallinnoimissa järjestelmissä kuten Windowsissa, käyttäjät ovat riippuvaisia siitä, että valmistaja tunnistaa ja korjaa virheet. Linux taas mahdollistaa jokaiselle käyttäjälle mahdollisuuden löytää ja ilmoittaa havaittuja turvallisuusongelmia, ja jopa osaamisen puitteissa ratkaista nämä ongelmat. Lähdekoodin avoimuus vahvistaa Linuxin kykyä mahdollisten haavoittuvuuksien torjunnassa suljettuihin järjestelmiin verrattuna. (MonoVM, 2024; LogicMonitor, 2023; Red Hat, 2022)

Turvallisuus on Linux-palvelinten yksi tärkeimmistä tavoitteista, ja se näkyy erilaisissa järjestelmiin sisällytetyissä turvallisuusominaisuuksissa, kuten palomuurin asetusten hallinnassa. Turvallisuusominaisuuksien ansiosta ylläpitäjät voivat tehostaa turvallisuutta ja arkaluontoisten tietojen suojaamista tärkeillä toimenpiteillä, kuten salauksilla, vahvoilla todennusmenetelmillä sekä käyttöoikeuksien hallinnalla. Linux-palvelinten turvallisuuden kannalta on myös oleellista säännölliset järjestelmäpäivitykset ja korjaukset. Osassa Linux-jakeluissa on myös turvallisuutta parantava SELinux, jonka ansiosta ylläpitäjät voivat laajemmin hallita järjestelmän käyttöoikeuksia. (MonoVM, 2024; LogicMonitor, 2023; Red Hat, 2022)

Linux on suosittu palvelinympäristöissä myös laajan sovellus- ja työkaluvalikoimansa ansiosta, mikä mahdollistaa käyttäjille järjestelmien mukauttamisen ja konfiguroinnin lähes mihin tahansa tarkoitukseen. Linux tarjoaa tämän lisäksi käyttäjäystävällisiä ominaisuuksia. Nämä tekevät siitä sopivan valinnan ylläpitäjille, jotka haluavat hallita palvelinympäristöään helposti. Avoimen lähdekoodin ansiosta ylläpitäjät voivat hyödyntää automatisointityökaluja hallitakseen useampaa Linux-palvelinta tai virtuaalikonetta yhtäaikaaisesti, poistaen tarpeen kirjautua erikseen jokaiseen järjestelmään konfiguroimaan ne manuaalisesti. Tämä mahdollistaa ympäristöjen helpon ja toistettavan käyttöönoton automatisoiduilla toimenpiteillä. (MonoVM, 2024; LogicMonitor, 2023; Red Hat, 2022)

2.3.2 Linux-palvelimen yleisiä konfiguroinnin vaiheita

Ennen palvelimen käyttöönottoa on tärkeää varmistaa, että se on konfiguroitu turvallisesti. Konfigurointivaiheet voivat vaihdella jakelun mukaan, koska osa jakeluista voi olla jo valmiiksi pidemmälle konfiguroituja turvallisuuden kannalta. Seuraavat vaiheet kattavat Linux-palvelimen perustason suojauksen tunnetuimpia uhkia vastaan. (Sen. K, 2024)

Vaiheet ovat: salasana, verkkoasetukset, sovellukset, NTP- ja palomuuriasetukset, sekä SSH-etäyhteyden suojaaminen. Nämä ovat yleispäteviä vaiheita suurimpaan osaan Linux-jakeluista.

Salasanan vaihtaminen pääkäyttäjälle on yksi tärkeimmistä ensimmäisistä vaiheista. Tämä yleensä tehdään jo palvelimena toimivan käyttöjärjestelmän asennuksen aikana. Vaikka vaihe on oleellinen, se saattaa usein toistuvissa konfiguraatioprosesseissa unohtua. Suositeltavaa on käyttää vähintään 8 merkkistä salasanaa, mitä enemmän merkkejä sen turvallisempi. Salasanan tulisi sisältää yhdistelmiä isoista- ja pienistä kirjaimista sekä symboleita ja numeroita. Kannattavaa on myös määritellä salasanoille niiden vanhenemis- ja lukitusajankohdat, monimutkaisuusvaatimukset sekä salasanojen historia, ettei samaa salasanaa käytettäisi uudelleen turvallisuussyistä. Yleensä pääkäyttäjän poistaminen kokonaan käytöstä ja peruskäyttäjien luonti tarvittavilla ja rajatuilla pääkäyttäjäoikeuksilla on myös suositeltavaa. (Sen. K, 2024)

Verkkoasetusten määrittäminen on oleellinen osa palvelimen toimivaksi ja turvallisemmaksi saamista. Verkkoyhteyden muodostaminen palvelimelle on oleellista. Palvelimelle tulisi määritellä laitteiden kommunikointia varten IP-osoite sekä isäntänimi, joka on palvelimen verkko-osoite ihmiselle luettavassa muodossa, kuten 'esimerkkiosoite.com'. On järkevää käyttää pysyvää IP-osoitetta, jotta palvelin ja sen resurssit löytyvät käyttäjille samasta osoitteesta. Jos ei esimerkiksi käytä ipv6-protokollaa, on kannattavaa kytkeä se pois päältä kokonaan. DNS-palvelimen tiedot tulisi asettaa sekä käyttää vähintään kahta eri DNS-palvelinta varmuuskopiointia varten. Verkkoasetusten määrittämisen jälkeen on hyvä varmistaa, että palvelimen nimi ja osoite ovat oikeat. Tämä tapahtuu esimerkiksi syöttämällä komento "nslookup" komentoriville. (Sen. K, 2024; one.com, n.d.)

Sovellusten asentaminen, joita palvelimen käyttötarkoitus vaatii, on oleellinen vaihe konfigurointiprosessia. Turhien sovellusten poistaminen käytöstä, joille ei ole käyttöä on myös hyvä käytäntö. On tärkeää myös päivittää koko järjestelmä sekä kaikki järjestelmän sovellukset. Uusimman järjestelmäversion käyttäminen on suositeltavaa järjestelmän turvallisuuden kannalta, ellei palvelimen käyttö jostain syystä vaadi vanhemman version

käyttöä. Automaattisten päivitysten käyttöönotto palvelimen sovelluksille on myös hyvä käytäntö, mikäli se sopii palvelimen tarjoamille palveluille. Oikeiden ohjelmien automaattinen käynnistys palvelimen käynnistyessä ja turhien sovellusten sekä taustaprosessien poistaminen käytöstä on myös hyvä käytäntö (Sen. K, 2024)

Palvelimen kellon synkronoiminen NTP-palvelimiin (engl. Network Time Protocol) on myös oleellista kellonaikojen eroavaisuuksien välttämisen kannalta. Palvelimen kellonajan virheellisyys saattaa tuoda esiin esimerkiksi todennukseen liittyviä ongelmia, joissa aikaero palvelimen ja todennuksesta vastaavan järjestelmän välillä tarkistetaan ennen käyttöoikeuden tai pääsyn myöntämistä. NTP-palvelimet voivat olla järjestelmän sisäisiä, jos ne ovat ympäristössä käytössä tai kaikkien saatavilla olevia ulkoisia kolmannen osapuolen aikapalvelimia. (Sen. K, 2024)

Palomuuriasetukset ovat myös oleellinen vaihe palvelinkonfiguraatiota, jonka avulla määritellään mikä verkkoliikenne on sallittu ja mikä estetty. Iptables-ohjelmisto vastaa Linux-palvelimissa palomuuriasetuksista, ja tietyissä jakeluissa iptables on jo valmiiksi täysin lukittu ja käyttäjän täytyy itse avata tarvittavat portit. Vaikka käytetyssä jakelussa olisi valmiiksi määritetyt palomuuriasetukset, on silti aina kannattavaa varmistaa, että ne ovat oikein määritetty. Palomuuriasetuksia määrittäessä on tärkeää avata ainoastaan ne portit, joita palvelin tarvitsee. (Sen. K, 2024)

SSH-protokollaa käytetään yleisesti etäyhteyden muodostamiseen Linux-jakeluissa, jonka tulisi myös olla oikein suojattu. On suositeltavaa poistaa pääkäyttäjältä mahdollisuus SSH-etäyhteyksiin varmuuden vuoksi, vaikka pääkäyttäjä olisikin poistettu kokonaan käytöstä. Tämä vähentää pääkäyttäjän haavoittuvaisuutta etäyhteyden kautta tapahtuville mahdollisille hyökkäyksille, jos pääkäyttäjä jostain syystä otettaisiin palvelimella uudelleen käyttöön. SSH-yhteyttä on myös mahdollista rajata valittuihin IP-alueisiin, esimerkiksi tilanteissa, jossa yhteyden muodostavien laitteiden IP-osoitteet ovat ennalta tiedossa. Salasanatodennuksen poistaminen kokonaan käytöstä ja sen korvaaminen sertifikaattitodennuksella, kuten SSH-avainparilla tekee palvelimelle tunnistautumisesta turvallisempaa. SSH-avainpari koostuu julkisesta- ja yksityisestä avaimesta, joista yksityisavain tulisi säilyttää täysin suojattuna. Julkista avainta voi jakaa vapaasti ja se ladataan palvelimelle, mikä mahdollistaa tunnistautumisen ilman lisätodennusta. Kun SSH-avaimilla yritetään tunnistautua, palvelin varmistaa, omistaako yhteyttä ottava laite yksityisavaimen. Mikäli näin on, pääsy palvelimelle sallitaan. (Sen. K, 2024; DigitalOcean, 2021)

2.4 Verkkopalvelin

Verkkopalvelin on sekä laitteisto- että ohjelmistoyhdistelmä, joka on suunniteltu käsittelemään käyttäjiltä internetin yli tulevia pyyntöjä erilaisia protokollia yleisimmin, HTTP (Hypertext Transfer Protocol) käyttäen. HTTP-protokolla määrittelee, miten linkitettyjä verkkodokumentteja siirretään kahden laitteen välillä. Verkkopalvelin voi viitata joko laitteistoon, ohjelmistoon tai niihin yhdessä. Laitteiston näkökulmasta se on tietokone, joka säilöö verkkopalvelinohjelmiston sekä verkkosivuston osakomponentteja, kuten kuvia ja tiedostoja. Laitteisto on kytketty verkkoon, ja se tukee fyysistä tiedonsiirtoa muiden verkkoon kytkettyjen laitteiden kanssa. Ohjelmistopuolella verkkopalvelin koostuu erilaisista osista, jotka hallitsevat käyttäjien pääsyä säilöttyihin tiedostoihin. (SolarWinds, n.d.; MDN, 2023)

Verkkopalvelimen tehtävä on tallentaa, käsitellä ja toimittaa verkkosivuja ja niiden sisältöä käyttäjille. Tähän tehtävään käytetään fyysistä tallennustilaa (laitteistoa) sekä verkkoselainta (ohjelmistoa). Verkkosivuston tiedot tallennetaan fyysiseen verkkopalvelimeen niiden turvallisuuden varmistamiseksi. Käyttäjän syöttäessä verkkosivuston osoitteen tai avainsanalla etsiessä, selain luo ja lähettää pyynnön verkkopalvelimelle, jotta se voi käsitellä tietoja. Verkkoselaimen tehtävänä on löytää verkkopalvelin, jossa verkkosivuston tiedot sijaitsevat. Selaimen löytäessä palvelimen, se lukee pyynnön ja käsittelee siinä olevat tiedot. Vaikka verkkopalvelimet käsittelevät ja hallinnoivat pääasiassa HTTP/HTTPS-pyyntöjä ja vastauksia asiakasjärjestelmiltä, ne voivat myös muun muassa säilyttää ja suojata verkkosivuston dataa, hallita kaistanleveyttä sekä toimia virtuaalipalvelimina monille sovelluksille ja verkkosivustoille. (SolarWinds, n.d.; MDN, 2023)

Verkkopalvelin on joko staattinen tai dynaaminen. Staattinen verkkopalvelin koostuu laitteistosta, kuten tietokoneesta sekä ohjelmistosta, esimerkiksi HTTP-palvelimesta. Tämän tyyppisiä palvelimia kutsutaan staattisiksi, koska ne toimittavat palvelimella säilötyn sisällön selaimelle sellaisenaan. Esimerkiksi NGINX on tunnettu ja tehokas staattinen verkkopalvelinratkaisu. Dynaaminen verkkopalvelin sisältää staattisen verkkopalvelimen sekä sovelluspalvelimen ja tietokannan. Sitä kutsutaan dynaamiseksi, koska sen sovelluspalvelin päivittää säilöttyt tiedostot tai luo sisältöä jopa reaaliaikaisesti ennen sisällön lähettämistä selaimelle HTTP-palvelimen kautta. Dynaaminen verkkosivusto voi näyttää erilaista sisältöä käyttäjille, kuten kuvia ja videoita. Apache-verkkopalvelin on tunnettu esimerkki dynaamisesta verkkopalvelimesta. (SolarWinds, n.d.; MDN, 2023)

Tunnetuimpia verkkopalvelinratkaisuja ovat muun muassa NGINX ja Apache. NGINX-verkkopalvelinohjelmisto on kevyt ja tehokas ratkaisu, joka käsittelee suuria liikennemääriä tarjoten muun muassa käänteisen välityspalvelimen ja kuormantasauksen ominaisuuksia, mikä tekee siitä ihanteellisen staattisen sisällön tarjoamiseen. Apache-verkkopalvelinohjelmisto puolestaan käsittelee käyttäjän pyynnöt ja toimittaa verkkosisällön HTTP:n välityksellä, tukien dynaamisen sisällön luomista PHP:n ja MySQL-tietokantaintegraation avulla. (SolarWinds, n.d.; MDN, 2023)

2.5 Automatisointi

Automatisoinnilla tarkoitetaan teknologian käyttöä toistuvien tehtävien ja prosessien korvaamiseen vähentämällä ihmisen manuaalisen työn tarvetta. Tämän avulla saadaan tehostettua esimerkiksi IT-infrastruktuuria sekä palvelujen toimittamista korvaamalla aiemmin manuaalisesti suoritettut toistuvat prosessit. Automaatiossa käytetään ohjelmistoa tai automatisointityökalua asettamaan ja toistamaan erilaisia ohjeita, prosesseja sekä käytäntöjä. Automaation käytön ansiosta saadaan säästettyä aikaa ja voidaan vapauttaa työntekijöitä tärkeämpiin tehtäviin, jotka vaativat nopeaa ja monimutkaisempaa palvelujen tarjoamista. (VMware, n.d.)

Automaatio on tehokas työkalu, jonka avulla voidaan skaalata yritysten liiketoimintaa vähentämällä sen kustannuksia, sekä mahdollistamalla työntekijöille keskittyminen strategisimpiin tehtäviin hallinnollisten työtehtävien sijaan. Suuri osa datakeskusten ja pilvipalvelujen toiminnoista on automatisoitavissa, mikä johtaa nopeampiin toimiin. Automaation ansiosta IT-ympäristöt kykenevät skaalautumaan nopeammin yritysten tarpeisiin pienemmällä määrällä virheitä sekä nopeammalla reaktioajalla. Täysin automatisoiduissa ympäristöissä voi tuotantovalmiiden resurssien toimitusaika lyhentyä viikoista jopa alle vuorokauteen. (VMware, n.d.)

Manuaalisen työn haittapuolet tulevat esiin useissa liiketoiminta- ja it-prosesseissa, etenkin käsin tehdyissä rutiinimaisissa työtehtävissä, kuten esimerkiksi tietojen syöttämisessä sekä lomakkeiden käsittelyssä. Näiden manuaalisten prosessien heikkouksia ovat alttius inhimillisille virheille ja niihin kulutettava aika. Nämä puolestaan hidastavat tuottavuutta ja vähentävät organisaatioiden kustannustehokkuutta. Virheiden lisäksi manuaalinen työ lisää alttiutta riskeille, kuten virheellisille laskelmille sekä tietoturvaongelmille. Manuaalisesti toteutetut prosessit ovat vaikeasti skaalautuvia ja hidastempoisia erityisesti kiireaikoina, jolloin vaaditaan enemmän resursseja esimerkiksi lisää työntekijöitä tai aikaa. Nämä haasteet korostavat tarvetta automaation käyttöönotolle. Automatisoinnilla tehokkuutta voidaan parantaa ja vähentää virheiden määrää, samalla varmistuen liiketoiminnan sujuvuus, esimerkiksi sovellusten ja palvelinten hallinnassa IT-ympäristöissä. (Vijh, 2022)

2.5.1 Automatisoinnin hyödyt

Automatisoinnin tarkoituksena ei ole korvata ihmisiä, vaan vahvistaa inhimillisiä kykyjä. Mitä enemmän automaatiota käytetään eri tehtäviin, ihmisen osallistumisesta prosesseihin tulee vähäisempää, mutta tärkeämpää. Jotkut saattavat myös kokea automatisoinnin uhkana työpaikkansa puolesta, vaikka se todellisuudessa vapauttaa työntekijöitä rutiininomaisista työtehtävistä suurempiin ja tärkeämpiin haasteisiin. Prosessien yksinkertaistamisella sekä manuaalisten tehtävien vähentämisellä, IT-tiimit voivat parantaa tehokkuuttaan ja tuottavuuttaan tinkimättä resursseistaan strategisten tavoitteiden, kuten paremman asiakastytyväisyyden saavuttamiseksi. (Red Hat, 2023.)

Automatisoinnilla on monenlaisia hyötyjä. Se tehostaa merkittävästi tuottavuutta automatisoimalla toistuvat rutiinitehtävät. Tämä mahdollistaa tehtävien nopeamman ja tehokkaamman suorittamisen vähäisellä ihmisen valvonnalla. IT-työntekijät voivat näin käyttää enemmän aikaa tärkeämpiin ja tuottavampiin projekteihin ja työtehtäviin. Automaatio myös parantaa luotettavuutta vähentämällä ihmisen osallistumista prosesseihin, joka puolestaan vähentää virheriskiä ja inhimillisten virheiden määrää, koska prosessit ovat varmasti johdonmukaisia. Vakiintuneen suorituksen myötä tiimit pysyvät hyvin perillä prosessien, testien, päivitysten ja työnkulkujen ajoituksesta ja tarkkuudesta. (Red Hat, 2023.)

Automaation käyttö yksinkertaistaa hallintaa poistamalla tarpeen suurille ihmistiimeille monimutkaisten prosessien suorittamiseen. Tämä vähentää tiedon puutteen ja väärinkäsitysten riskiä organisaation sisällä ja eri ryhmien välillä. Tiedon koodaaminen automaatioon edistää hallinnan yhtenäisyyttä erilaisten tiimien ja tehtävien kesken. Lisäksi automaatio mahdollistaa ympäri vuorokauden toiminnan, koska se kykenee suorittamaan tehtäviä yritystoiminnan ulkopuolisina tunteina. Tämä joustavuus mahdollistaa automatisoitujen korjaustoimenpiteiden suorittamisen milloin tahansa, vähentäen keskimääräistä vikojen korjausaikaa ja minimoiden käyttökatkosten syntymistä. (Red Hat, 2023.)

Automaation hyödyt myös tietoturvassa ovat merkittävät. Se vahvistaa tietoturvatyökaluita nopeasti tunnistamalla, estämällä ja reagoimalla potentiaalisiin tietoturvauhkisiin IT-ympäristöissä vakiintuneella tavalla. Tällä ennakoivalla lähestymistavalla yritykset voivat parantaa yleistä tietoturvasaantoaan. (Red Hat, 2023.)

2.5.2 Automatisoinnin mahdollisia haittoja

Automaation käyttöönoton alkuvaiheen käyttökustannukset ja alkuinvestoinnit voivat olla merkittäviä, vaikka automaatio voikin pitkällä aikavälillä tuoda huomattavia kustannussäästöjä. Automatisoinnin käyttöönottoon liittyy paljon kuluja, kuten laitteiston hankinta, ohjelmistojen hankkiminen, henkilöstön koulutus sekä automaatiolla korvattavien prosessien uudelleenjärjestely. Näiden alkuinvestointien vuoksi pienemmillä yrityksillä tai toimialoilla missä resurssit ovat rajallisia, voi olla haasteita ottaa automaatiota käyttöön. (Badkar, 2023)

Automatisoinnilla on myös omat tekniset haasteensa ja rajoituksensa, koska se perustuu vahvasti teknologiaan ja monimutkaisiin järjestelmiin. Tekniset ongelmat, kuten ohjelmistovirheet, laiteviat tai tietoturvaheikkouudet, voivat haitata toimintaa ja aiheuttaa käyttökatkoja. Tiettyjä tehtäviä voi olla myös haastavaa automatisoida niiden monimutkaisuuden tai ihmisen tarpeellisen prosessiin osallistumisen takia. (Badkar, 2023)

Työpaikkojen siirtyminen ja työttömyys voivat olla haittavaikutuksena, kun käytetään automaatiota toistuvien tehtävien ja prosessien suorittamiseen. Esimerkiksi tilanteissa, joissa toimistot ja teollisuus ottavat käyttöönsä koneita ja robotteja tuottavuutensa tehostamiseksi, voi työttömyysaste nousta erityisesti toistuvien tehtävien osalta. Tämän takia voi aiheutua taloudellisia ja sosiaalisia haasteita muun muassa tuloerojen kasvua ja korkeampia työttömyysasteita. (Badkar, 2023)

Ihmiskontaktin väheneminen ja negatiivinen vaikutus asiakaskokemukseen ja asiakastyytyväisyyteen ovat myös mahdollisia automaation haittavaikutuksia. Tiettyjen alojen asiakkaat saattavat suosia henkilökohtaisempaa tai suoraan ihmiseltä saatua palvelua. Automatisoiduilla järjestelmillä ei myöskään ole aina kykyä tarjota samaa empatiaa, ymmärrystä tai ongelmanratkaisutaitoja kuin oikealla ihmisellä. Automaatiosta riippuvaisuus voi myös johtaa taitojen ja asiantuntemuksen menetykseen. Mikäli työntekijät tulevat liian riippuvaisiksi automaattisista järjestelmistä, voi heidän taitonsa automaation korvaamisessa tehtävissä heikentyä. Tämä taas vaikeuttaa toimimista ilman automaatiota, joka voi puolestaan johtaa osaamisvajeeseen. (Badkar, 2023)

3 Ansible

Ansible on Python-ohjelmointikielillä kirjoitettu automatisointiin tarkoitettu helposti lähestyttävä, mutta tehokas ohjelmistotyökalu, joka tarjoaa monialustaista tukea tietokoneille. Ohjelma on avointa lähdekoodia eli se on julkisesti saatavilla käyttäjille, ja käyttäjillä on vapaus tarkastella, muokata ja jakaa sitä oman tarpeen tai halunsa mukaan. Sen yleisimpiä käyttötarkoituksia ovat: sovellusten-, palvelinten- ja pilvipalvelujen käyttöönotto ja konfiguraatiohallinta, sekä monimutkaistenkin työnkulkujen suoritusajavälien järjestäminen. (BasuMallick, 2022; Opensource.com., n.d.)

Ansiblesta on saatavilla kahta eri vaihtoehtoa. Ilmainen kaikille saatavilla oleva avoimen lähdekoodin versio sekä maksullinen Red Hatin Ansible Automation Platform. Avoimen lähdekoodin versio on tuettuna useimmissa käyttäjärjestelmissä edellyttäen, että Python on asennettuna. Maksullinen Red Hatin tarjoama Ansible Automation Platform on tilauspohjainen tuote, joka sisältää useita erityisesti yrityskäyttöön soveltuvia ominaisuuksia, kuten tapahtumapohjaisen automaation ja generatiivisen tekoälyn. Lisäksi siinä on enemmän hallintamahdollisuuksia turvallisuus- ja raportointityökalujen avulla, sekä mahdollisuus saada teknistä tukea. (BasuMallick, 2022; ansible.com., n.d.)

Työkalun keskeisiä vahvuuksia ovat sen helppokäyttöisyys ja yksinkertaisuus. Ohjelma painottaa myös vahvasti turvallisuuttaan ja luotettavuuttaan, koska se on suunniteltu mahdollisimman yksinkertaiseksi, välttämällä tarpeettomia monimutkaisuuksia ja liikkuvia osia. Ansiblen käyttöönotto sekä käyttäminen on helppoa, koska ohjelman käyttäminen ei vaadi erillistä agenttiohjelmistoa tai ylimääräistä turvallisuusinfrastruktuuria toimiakseen. (BasuMallick, 2022)

3.1 Ansiblen historiaa

Ansiblen kehitti vuonna 2012 ohjelmistokehittäjä Michael Dehaan. Dehaan kehitti Ansiblen halustaan luoda uuden työkalun, jonka avulla voitaisiin tehostaa ja yksinkertaistaa useiden palvelinten automatisoitua konfigurointia ja hallintaa. Dehaan päätti luoda avoimen lähdekoodin ratkaisun automaattisiin asennuksiin turhautuneena siitä, että asennusten saaminen toimiviksi vei yhä useita päiviä tai jopa pidempään erilaisten haasteiden, kuten DNS- ja NTP-ongelmien vuoksi. Ansible sai alkunsa tämän suunnittelutavoitteen perusteella tarjoten yksinkertaisen ja nopean ratkaisun verrattuna muihin sen aikaisiin samankaltaisiin työkaluihin, kuten Chef- ja Puppet-työkalujen käyttämiseen, joiden saaminen toimiviksi vei runsaasti aikaa. (Kadima, 2023; Rometsch, 2021)

Vuoden 2012 helmikuussa, kun Ansible julkaistiin avoimen lähdekoodin projektina, oli automaation kysyntä suuremmissa kasvussa kuin nykyaikana. Se keräsi nopeasti suosiota sen helppokäyttöisyyden ja yksinkertaisuutensa ansiosta. Suosionsa takia projekti houkutteli oman yhteisönsä jäseniä, joiden avulla Ansiblen ominaisuuksia laajennettiin ja ohjelmaan kehitettiin lisämoduuleja monenlaisten palveluiden ja järjestelmien hallintaa varten. Ansible tarjosi ratkaisun, joka mahdollisti yhden henkilön käyttöönottaa sen alle päivässä, mikä oli huomattava parannus monimutkaisempiin työkaluihin verrattuna. Nimensä Ansible on saanut kirjailija Orson Scott Cardin kirjoittamasta tieteiskirjasta nimeltä ”Ender’s Game”. Kyseisessä kirjassa Ansible on kuvitteellinen laite, jonka ansiosta voidaan kommunikoida suurien etäisyyksien yli välittömästi. (Kadima, 2023; Rometsch, 2021)

Viime vuosien aikana Ansiblesta on tullut nopeaa tahtia yksi IT-alan käytetyimmistä ja suosituimmista automatisointityökaluista ympäri maailman. Työkalun avoimen lähdekoodin yhteisö on nykypäivään mennessä laajentunut jo yli 1200 yksittäiseen osallistajaan projektin alkua ajoista lähtien. Ansiblen modulaarinen rakenne sekä laaja soveltuvuus erilaisiin automaatio- ja orkestrointihaasteisiin toivat erittäin suuren vaikutuksen ja suosion satoihin tuhansiin ihmisiin ympäri maailmaa. (Geerling, 2022, s.1)

3.2 Ansiblen arkkitehtuuri

Ansiblen käyttö perustuu solmujen (node) käsitteisiin, jotka jakautuvat ohjauspalvelimeen (control node) ja kohdepalvelimeen (managed node). Ansiblen toiminnot, kuten Ansible-pelikirjojen (ansible-playbook) suorittaminen, tapahtuvat ohjauspalvelimelta. Kohdepalvelimet ovat niitä laitteita tai järjestelmiä, joita automatisoidaan, kuten Windows- ja Linux-palvelimet. Erityisesti verkkolaitteiden ja muiden IT-laitteiden automatisoinnissa, joissa moduuleja ei voi suorittaa, Ansiblen käyttö tapahtuu ohjauspalvelimelta. Ansible käyttää oletuksena OpenSSH-protokollaa tiedonsiirtoon järjestelmien välillä, ja sen käyttämä ihmiselle luettava YAML-kieli tekee ohjelman käytön opettelusta helpompaa ja käyttäjäystävällisempää. (How Ansible Works, n.d.; Opensource.com., n.d.)

Linuxin ja Windowsin automatisoinnissa, Ansible ottaa yhteyden hallittuihin solmuihin ja lähettää niihin pieniä ohjelmia. Näitä ohjelmia kutsutaan Ansible-moduuleiksi. Kyseiset moduulit on kirjoitettu siten, että ne kuvastavat järjestelmän toivottua tilaa resurssimalleja hyödyntäen. Ansible suorittaa nämä moduulit oletuksena SSH:n kautta ja poistaa ne suoritettuaan. Moduulit on suunniteltu idempotentteiksi aina kun mahdollista. Tämä siksi, jotta järjestelmässä tapahtuisi muutoksia vain tarvittaessa. (How Ansible Works, n.d.)

3.2.1 Moduulit

Ansiblen moduulit ovat pieniä ohjelmia, joita suoritetaan paikallisella tietokoneella, sovellusohjelmointirajapinnoilla (API) tai etäpalvelimillä. Moduulit ovat yleensä Python-ohjelmointikielillä kirjoitettuja koodeja. Nämä koodit sisältävät metatietoja, jotka määrittelevät missä ja milloin tietty automaatiotehtävä suoritetaan sekä ketkä kaikki käyttäjät voivat tehtävän suorittaa. Ansiblen moduuleja voi halutessaan luoda omia tai ladata saatavilla olevista GitHub repositoryistä tai Red Hatin tarjoaman maksullisen Ansible-automatisointityökalu alustan kautta. (Red Hat, 2024)

Useita tietotekniikan, kuten verkkotoimintojen, resurssien käyttöönoton ja valmistelun, tietoturvan, pilven-, käyttäjän- ja konfiguraatiohallinnan sekä viestinnän tehtäviä voidaan suorittaa moduulien avulla. On tärkeää saada käsitys moduulien suhteista muihin keskeisiin Ansiblen-automatisoinnin osiin, jotta voidaan ymmärtää moduulien hyöty ja tarkoitus käyttötapauksien automatisoinnissa. (Red Hat, 2024)

Näitä muita keskeisiä Ansiblen osia ovat muun muassa:

Tehtävät (task), jotka määrittelevät kohdepalvelimella (managed node) suoritettavan toimenpiteen, määrittelemättä kuitenkaan tarkempaa kohdetta johon toimenpide vaikuttaa. Tehtävän sisällä oleva moduuli voidaan suorittaa kohdepalvelimella, mutta kyseisen kohdepalvelimen määrittely tapahtuu tehtävän ulkopuolella toistotasolla. (Red Hat, 2024)

Toisto (play) on Ansiblen suorituksen keskeinen osa, joka koostuu tehtäväluettelosta ja näiden tehtävien soveltamisesta tiettyihin kohdepalvelimiin. Toistossa määritellään myös muuttujia, ja monimutkaisemmissa tilanteissa se voi sisältää myös rooleja. (Red Hat, 2024)

Pelikirjat (Ansible playbook) ovat YAML-kielisiä tiedostoja, joihin sisältyy aina yksi tai useampi toisto (play). (Red Hat, 2024)

Roolit (Ansible role) pakkaavat toistossa käytettävää sisältöä, kuten tehtäviä, käsittelijöitä, muuttujia, lisäosia, malleja sekä tiedostoja. Rooleja on mahdollista tuoda toistoon sekä käyttää uudelleen muissa toistoissa tai pelikirjoissa. (Red Hat, 2024)

Lisäosat (Plugins) ovat Ansiblen ydintoiminnallisuuteen perustuvia koodinpätkiä, joita yleensä suoritetaan ohjauspalvelimelta (control node). Moduulitkin ovat yksi lisäosatyyppi, mutta niitä yleensä suoritetaan kohdepalvelimella ohjauspalvelimen sijaan. (Red Hat, 2024)

Ansiblen moduuleja voi ajatella yksinkertaisesti työkaluina, jotka suorittavat tehtäviä. Tehtävät määrittelevät suoritettavan toimenpiteen ja moduulit suoritetaan halutuissa kohdepalvelimissa tämän toimenpiteen saavuttamiseksi. Moduulit sijaitsevat yleensä pelikirjojen kirjastossa ja niitä suoritetaan silloin kun pelikirjat suorittavat niihin liittyviä tehtäviä. Moduuleja voidaan myös sisällyttää rooleihin tai kokoelmiin. Kun Ansible-rooli otetaan käyttöön pelikirjassa, roolihakemiston moduulit suorittavat roolille määritetyt tehtävät. Mikäli moduuli on toistokohtainen, se suoritetaan vain kerran, kun taas roolin sisäiset toistot suoritetaan aina kun roolia kutsutaan pelikirjassa. (Red Hat, 2024)

3.2.2 Inventory (varasto)

Ansiblen varastot ovat luetteloja kohdepalvelimista ja niiden ryhmistä infrastruktuurissa. Niiden tehtävänä on määrittellä automatisoitavat kohdepalvelimet ja niiden ryhmät. Varastot määritellään tyypillisesti INI- tai YAML-tiedostomuodoissa ja ne sisältävät kohdepalvelinten nimet tai IP-osoitteet sekä ryhmät, joka mahdollistaa tehtävien suorittamisen usealle palvelimelle samanaikaisesti. Ansible luo oletuksena kaksi ryhmää varastotiedostoon All-ryhmän ja ungrouped, jos ryhmiä ei ole itse määriteltä. All-ryhmä sisältää kaikki kohdepalvelimet ja ryhmä 'ungrouped' sisältää kaikki kohdepalvelimet, jotka kuuluvat ainoastaan All-ryhmään. (Ansible Community Documentation, n.d.)

3.2.3 Playbookit (pelikirjat)

Ansiblen pelikirjat ovat automaattisesti suoritettavia tehtäväluetteloja, jotka suoritetaan määritetyille inventoreille tai palvelinryhmille. Ne tarjoavat toistettavan, uudelleenkäytettävän sekä yksinkertaisen konfiguraationhallinnan ja monipalvelinjärjestelmän. Jos on jokin tehtävä mikä pitää suorittaa useammin kuin kerran, on järkevää kirjoittaa pelikirja ja laittaa se itselleen talteen. Näin pelikirjaa voi käyttää uudelleen palvelinten konfigurointiin ja etäjärjestelmien konfiguraation vahvistamiseen. (Red Hat, 2023.; Ansible Community Documentation, n.d.)

Ansible-tehtäviä voidaan yhdistää yksi tai useampi kerrallaan muodostamaan toistoja (play), jotka ovat tiettyihin palvelimiin kartoitettuja järjestettyjä tehtäväryhmiä. Pelikirjan tehtävät suoritetaan niiden kirjoitusjärjestyksessä ja ne voivat sisältää yhden tai useamman toiston, Ansible-rooleja sekä tehtäväpaketteja ja niihin liittyviä automaatioresursseja, joita voidaan suorittaa monissa toistoissa tai uudelleen käyttää muissa pelikirjoissa. (Red Hat, 2023.; Ansible Community Documentation, n.d.)

Pelikirjoja käytetään toistuvasti eri IT-infrastruktuurin osa-alueiden automatisoinnissa, kuten käyttäjärjestelmissä, kubernetes-alustoissa, verkkojen, turvajärjestelmien sekä erilaisten repositoryjen esimerkiksi GitHubin hallinnassa. Pelikirjojen avulla voidaan ohjelmoida sovelluksia, palveluita, ohjaus- ja hallintapalvelimia sekä muita laitteita, vähentämällä manuaalisen työn tarpeen määrää. Pelikirjat ovat YAML-kielisiä tiedostoja ja ovat tämän ansiosta helppolukuisia, yksinkertaisia sekä mahdollistavat pelikirjojen luomisen ilman monimutkaisen ohjelmointikielen opettelua. (Red Hat, 2023.; Ansible Community Documentation, n.d.)

3.2.4 YAML

YAML on merkintäkieli, joka tarjoaa vakiintuneen tavan esittää selkeästi tietoja, jotka ovat ihmiselle sekä koneelle ymmärrettävissä. Nimi "YAML" on lyhenne ja tulee sanoista "YAML Ain't Markup Language", jolla korostetaan kielen tarkoitusta dataan eikä dokumentointiin tai "Yet Another Markup Language". YAML on myös suunniteltu yksinkertaiseksi ja luotettavaksi. Sen syntaksi on selkeä ja minimalistinen, joka mahdollistaa monimutkaistenkin tietorakenteiden ilmaisemisen luonnollista kieltä muistuttavassa muodossa. Selkeän syntaksin vuoksi sen tietorakenteet ovat myös helppoja hahmottaa pikaisella silmäyksellä. (IBM, 2023)

Koska YAML käyttää ihmiselle luettavaa syntaksia, se sopii erityisesti esimerkiksi konfiguraatitiedostoihin ja eri järjestelmien väliseen tietojenvaihtoon. Sen selkeä ja helppokäyttöinen rakenne tekee siitä soveltuvan eri aloille, koska käyttäjät voivat järjestellä ja määrittellä dataa selkeällä tavalla. Koska YAML tukee Unicode-merkkejä, se mahdollistaa ison määrän erilaisia merkkejä ja symboleita käytettäväksi syntaksissa. (IBM, 2023)

YAML:n yksinkertaisuutta pidetään merkintäkielen parhaana puolena, mikä tekee siitä helposti ymmärrettävän sekä kehittäjille että ei-teknisille käyttäjille. Sen rakenteellinen formaatti yksinkertaistaa konfiguraatitiedostojen luomista ja varmistaa sovellusten monipuolisen soveltuvuuden. Lisäksi YAML:n yhteensopivuus eri järjestelmien ja ohjelmointikielten kanssa mahdollistaa saumattoman datan vaihdon ja yhteen toimivuuden. Tämä tekee siitä monipuolisen vaihtoehdon nykyaikaisille kehitysympäristöille. (IBM, 2023)

YAML on kuitenkin herkkä virheille muutamista syistä. Se on kirjainkoosta riippuvainen, joten isot ja pienet kirjaimet käsitellään eri asioina, mikä voi aiheuttaa sekaannusta. YAML on myös välilyönneistä riippuvainen, ja sisennykset määrittelevät tiedostojen rakenteen. Vain välilyönnit hyväksytään tiedostojen sisennyksessä, joten sarkaimet eivät kelpaa. (platform.sh, n.d.)

4 Menetelmät

Opinnäytetyön käytännön osuus suoritetaan vertailevalla tutkimusmenetelmällä. Tutkimukseen sisältyy verkkopalvelinten konfigurointi manuaalisesti sekä automatisointityökalu Ansiblen avulla. Kummallakin tavalla konfiguroidaan sama määrä palvelimia, samassa Linux-ympäristössä sekä samoilla konfigurointivaiheilla, jotta saadaan mahdollisimman tarkka vertailu aikaiseksi. Työssä tutkitaan Ansiblen käyttöä verkkopalvelinten konfiguroinnissa ja sitä, miten se eroaa manuaalisesta konfigurointitavasta.

Aineistoa työtä varten kerätään Ansiblen dokumentaationsivuilta, Githubista sekä muista luotettavista verkkolähteistä. Aineistopohjaa Linux-palvelinten konfiguraatiosta kerätään luotettavista verkkolähteistä, muun muassa DigitalOceanin sivuilta. Tutkimuksessa käytetyt kaikki palvelinten konfiguroinnin vaiheet niin manuaalisesti kuin Ansiblella tehdyt kuvaillaan tai dokumentoidaan. Tämä sisältää tutkimuksessa käytetyt Ansible-pelikirjat, inventory-tiedostot ja muut oleelliset osat. Manuaalisessa konfiguroinnissa käytetyt ohjeet merkitään. Kaikista pelikirjoista ja muista käytetyistä tai muokatuista ohjemateriaaleista merkitään myös lähteet.

Tutkimus toteutetaan käyttäen VirtualBox-virtualisointiympäristöä, joka mahdollistaa useiden virtuaalisten Linux-palvelinten luomisen ja tarjoaa kontrolloidun turvallisen ympäristön. Vertailuksi tutkimuksesta kerätään tuloksia palvelinten johdonmukaisuudesta, eli onko palvelinten konfiguroinnissa jotakin eroavaisuuksia manuaalisen ja automatisoidun konfiguroinnin välillä. Kerätään tietoa palvelinten valmiudesta ja vertaillaan palvelimen käyttöönottoaikoja. Empiirisiä havaintoja ja kokemuksia myös otetaan talteen tutkimusprosessin ajalta. Tutkimuksessa konfiguroidaan ainoastaan Linux-palvelimia kummallakin konfigurointitavalla tutkimukseen varatun rajallisen ajankäytön takia.

5 Tutkimus

Tutkimuksen tarkoituksena on tutkia palvelinten konfigurointia automatisoidusti ja manuaalisesti sekä saada selville näiden tapojen välisiä eroja. Tutkimuksella selvitetään myös mitä inhimillisiä virheitä palvelinten manuaalisessa konfiguroinnissa tapahtuu, jotka mahdollisesti automatisoinnilla voidaan välttää. Tutkimuksessa selvitetään myös Ansible-työkalun käytön hyviä puolia sekä analysoidaan työkalun mahdollisia haasteita.

Opinnäytetyön käytännön osuus toteutettiin Hamkin vCommander-virtuaalitietokoneella, johon oli valmiiksi asennettuna VirtualBox-ohjelmisto. VirtualBox-ympäristöön voidaan asentaa tarvittava määrä palvelimia. Tämä toimintatapa ei täysin vastaa tuotannossa käytettyä tapaa asentaa Linux-palvelimia, mutta se tarjoaa tutkimukseen vakaan ja samanlaisen toimintaympäristön kumpaankin konfigurointitapaan. VirtualBox valikoitui myös sen avoimen lähdekoodin, laajojen ominaisuuksien sekä resurssien hallinnan mahdollisuuden takia. Työssä käytettiin samaa Ubuntu-palvelinversiota kaikkiin manuaalisesti sekä Ansiblella konfiguroituihin palvelimiin.

Opinnäytetyön tutkimuksessa tavoitteena oli LAMP-pinon asennus, jota yleisesti käytetään verkkopalvelinten isännöintiin. LAMP-pino viittaa avoimeen lähdekoodin perustuviin ohjelmistoihin. Nimitys tulee Linux-käyttäjärjestelmästä, Apache-verkkopalvelinratkaisusta, MySQL-tietokannasta sekä PHP-ohjelmointikielestä. Tutkimuksen tavoitteena oli myös rajata palomuuriasetuksia siten, että palvelimelle ja sen isännöimille sivuille olisi pääsy vain sisäiseltä hallinnassa olevalta ip-osoitteelta. (DigitalOcean, 2024-a; DigitalOcean, 2024-b)

Manuaalisessa konfigurointiosuudessa käytetyt konfigurointivaiheet saatiin DigitalOceanin verkkosivuilta. Saman yrityksen mukaisia ohjeita käytettiin myös palvelinten palomuurien asetuksiin, jotta Ubuntu-palvelimille ja niille asennetuille verkkopalvelimille olisi pääsy ainoastaan sisäisestä hallinnassa olevasta ip-osoitteesta. (DigitalOcean, 2024-a; DigitalOcean, 2024-b) Ansiblen konfigurointivaiheisiin aineistopohjaa haettiin DigitalOceanin sekä Atlantic.Net:in verkkosivuilta. (Heidi, 2024; Jethva, 2024)

LAMP-pinon asentaminen ja palomuuriasetusten muokkaaminen vastaa luvussa 2.3.2 esiteltyjä yleisiä Linuxin konfigurointivaiheita. Käyttäjälle vahvan salasanan asettaminen tehtiin jo osana Ubuntu-palvelinten asennusta. SSH-avainparien luonti tehtiin myös ennen palvelinten konfigurointia, joten nämä kaksi vaihetta on rajattu pois tutkimuksen konfigurointivaiheista.

Vaiheet, jotka tutkimuksessa oli tavoitteena kummallakin konfigurointitavalla saavuttaa, olivat seuraavat:

Järjestelmän päivitys, jolla varmistetaan kaikkien järjestelmän osien ajantasaisuus ja turvallisuus.

Apache-verkkopalvelimen asennus, joka on keskeinen osa LAMP-pinoa ja mahdollistaa verkkosivujen vastaamisen HTTP-pyyntöihin sekä niiden toimittamisen.

Varmistetaan, että Apache-verkkopalvelin sallii verkkoliikenteen vain sisäisestä hallinnassa olevasta IP-osoitteesta määrittämällä turvallisuussäännöt palomuurissa. Tämä lisää järjestelmän turvallisuutta ja hallintaa sekä varmistaa, että vain hallitulla tietokoneella on pääsy tarkastelemaan isännöityjä sivuja.

MySQL-tietokantapalvelimen asennus sekä tarvittavat tietoturva-asetukset, mukaan lukien vahvan salasanan asettamisen MySQL:n pääkäyttäjälle, jolle ei vakiona ole asetettu salasanaa.

Mysql_secure_installation komento, joka parantaa MySQL:n tietoturvaa. Komennon suorittamisella poistetaan anonyymit käyttäjät ja estetään pääkäyttäjältä mahdollisuus etäkirjautua tietokantaan, sallien vain paikallisen kirjautumisen. Komennolla poistetaan myös 'test' oletustietokanta, joka on oletuksena kaikkien käyttäjien saatavilla. Prosessi viimeistellään päivittämällä asetetut käyttöoikeudet ja muutokset, jotta ne tulevat heti voimaan.

PHP-ohjelmointikieli ja sen vaadittavat lisäosat asennetaan tuomaan dynaamista sisällöntuotantoa palvelimelle.

PHP:n asentamisen jälkeen luodaan virtuaalinen isäntäpalvelin, jolle määritetään omistajuus tärkeille hakemistoille, jotta verkkosivuston tiedostot olisivat hallinnassa.

Apache-verkkopalvelimen määrittämiseksi luodaan uusi konfiguraatitiedosto, johon syötetään palvelimen nimi, toissijaiset nimet (aliakset) sekä muut tarpeelliset asetukset. Tämän jälkeen aktivoidaan virtuaalipalvelin (VirtualHost), joka mahdollistaa useiden sivustojen isännöinnin samasta palvelininfrastruktuurista. Palvelimen oletusasetukset poistetaan käytöstä, jotta uudet määrittelyt tulisivat voimaan. Lopuksi tarkastetaan, että kaikki asetukset ovat kunnossa ja varmistetaan, että palvelin toimii odotetusti sen jälkeen, kun Apache käynnistetään uudelleen.

Apachen-verkkopalvelimen hakemiston juureen järjestetään HTML-sivu, joka tarjoaa perustietoja sivustosta.

Lisätään PHP-skripti, joka näyttää yksityiskohtaisia tietoja PHP-konfiguraatiosta, jotta voidaan varmistaa PHP-ympäristön toimivuus. Sivun toimivuus tarkastetaan siirtymällä osoitteeseen "palvelimen-ip-osoite/info.php" selaimella, jonka jälkeen kyseinen PHP-skripti poistetaan turvallisuussyistä.

Tietokantaan luodaan uusi tietokanta ja käyttäjä, jossa luodulle käyttäjälle annetaan täydet oikeudet kyseiseen tietokantaan. Tämän jälkeen varmistetaan oikeuksien toimivuus kirjautumalla juuri luodulla käyttäjällä ja luodaan tietokantataulu, johon lisätään tietoja. Luotu PHP-skripti hyödyntää luotua tietokantaa näyttämällä taulukon tiedot dynaamisesti verkkosivulla.

Lopuksi tarkastetaan kaikkien järjestelmän osien toimivuus, siirtymällä verkkosivustolle, joka näyttää tietokantataulua hyödyntävän PHP-skriptin. Tällä varmistetaan, että kaikki toimii yhteen suunnitellulla tavalla, ja että sivusto on käytettävissä määrättyssä verkkosivuston osoitteessa.

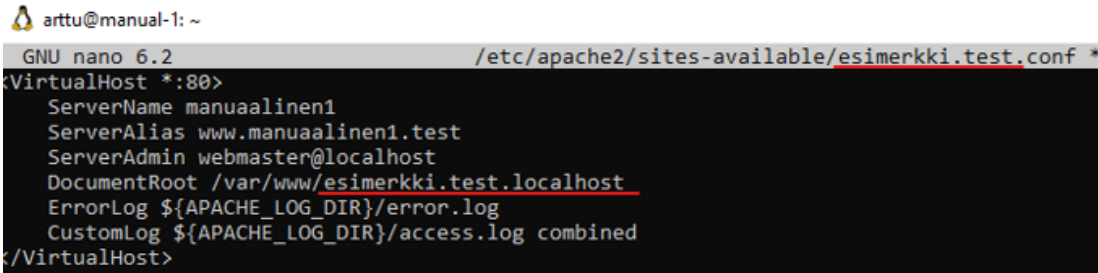
Tutkimuksessa konfiguroitiin yhteensä kuusi palvelinta, joista kolme manuaalisesti ja kolme Ansiblella. Käytännön osuuden toteutusta varten asennettiin kuusi kappaletta identtisiä Ubuntu-palvelimia VirtualBox ympäristöön, jossa kaikilla oli sama määrä prosessoreita ja muistia käytössä. Oli tärkeää, että palvelimet olivat identtisiä, jotta vertailu olisi mahdollisimman tarkka ja puolueeton palvelimille määritettyjen resurssien takia.

Palvelimiin asennettiin kaikkiin OpenSSH-palvelin Ubuntu-serveriä VirtualBox-ympäristöön asentaessa, jotta niitä voitiin konfiguroida etäyhteyden kautta manuaalisesti ja Ansiblella tehdyissä konfiguroinneissa. Etäyhteyden kautta hallinnoitiin palvelimia käyttäen virtuaalitietokoneella valmiiksi asennettua WSL eli Windows Subsystem for Linux ohjelmaa. Etäyhteyden kautta palvelimien konfigurointi olisi sujuvampaa ja Ansiblen käyttöä varten tarvitsee etäyhteyden palvelimiin, joten etäyhteyttä käytettiin myös manuaalisessa konfiguroinnissa.

5.1 Manuaalisen konfiguroinnin tulokset

Ensimmäisen manuaalisen palvelimen konfigurointi alkoi kello 18.58 ja viimeinen vaihe suoritettiin kello 19.21. Kellonajat merkittiin talteen palvelimia konfiguroidessa. Tähän aikaväliin sisältyi kaksi taukoa kellonaikoina 19.09–19.13 sekä 19.19–19.20. Ensimmäinen tauko johtui virheellisestä "DocumentRoot" polun merkitsemisestä Apachen konfiguraatitiedostoon. Virhe huomattiin, kun siirryttiin verkkosivulle tarkistamaan, toimiiko kaikki odotetusti, mutta sivuun ei saatu yhteyttä. Virhe löydettiin, kun viimeisimpiä vaiheita tarkasteltiin ja hakemistopolussa virhe korjattiin. Toinen tauko johtui hetkellisestä ajatuskatkoksesta, kun unohdettiin mikä konfigurointivaihe oli viimeisimpänä suoritettu. Ensimmäisen manuaalisen palvelimen konfigurointi kesti 23 minuuttia.

Kuva 1 Ensimmäinen inhimillinen virhe

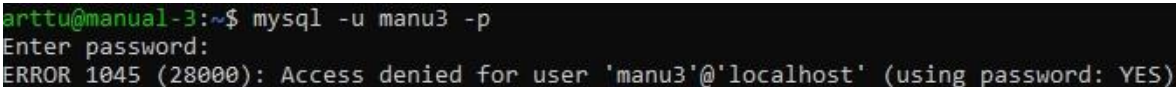


```
arttu@manual-1: ~  
GNU nano 6.2 /etc/apache2/sites-available/esimerkki.test.conf *  
VirtualHost *:80>  
  ServerName manuaalinen1  
  ServerAlias www.manuaalinen1.test  
  ServerAdmin webmaster@localhost  
  DocumentRoot /var/www/esimerkki.test.localhost  
  ErrorLog ${APACHE_LOG_DIR}/error.log  
  CustomLog ${APACHE_LOG_DIR}/access.log combined  
/VirtualHost>
```

Toisen manuaalisen palvelimen konfigurointi alkoi kello 19.24, ja viimeinen vaihe saatiin valmiiksi kello 19.39. Tähän ei sisältynyt taukoja eikä tapahtunut sekaannuksia. Konfiguraation kesto aika alkuvaiheesta loppuun kesti n.15 minuuttia.

Kolmannen manuaalisen palvelimen konfigurointi alkoi kello 19.43, ja viimeinen vaihe saatiin suoritettua kello 20.03. Tähän sisältyi myös ajatuskatkoksesta johtuva tauko kello 19.57–20.00. Tässä konfiguraatiossa tapahtui myös yksi inhimillinen virhe, salasana syötettiin virheellisesti MySQL-tietokantaan uudella luodulla käyttäjällä kirjautuessa. Kolmannen palvelimen manuaalinen konfigurointi kesti n.21 minuuttia.

Kuva 2 Toinen inhimillinen virhe



```
arttu@manual-3:~$ mysql -u manu3 -p  
Enter password:  
ERROR 1045 (28000): Access denied for user 'manu3'@'localhost' (using password: YES)
```

Manuaalisissa konfiguroinneissa aikaa vievintä oli komentojen ja salasanojen kirjoittaminen komentoriville. Tämä vaatii tarkkuutta ja komentojen lukemista ennen suoritusta, ettei vahingossa suoriteta väärinä komentoja. Väärän salasanan kirjoitus varsinkin, jos se sisältää paljon erilaisia merkkejä sekä isoja- ja pieniä kirjaimia voi olla turhauttavaa ja aikaa vievää. Salasanoja kirjoittaessa komentoriviltä ei myöskään näe mitä on kirjoittanut tai montako merkkiä syöttänyt. Salasanan suojaus on tietoturvan kannalta hyvä, mutta aikaa kuluttavaa, mikäli virheitä tapahtuu. Tämä todennäköisesti korostuisi tilanteissa, joissa tarvitsisi konfiguroida tai hallita useita palvelimia nopeasti.

Manuaalinen konfiguraatio on selkeää ja yksinkertaista eikä vaadi ohjelmiston käyttöä. Se vaatii kuitenkin keskittymistä sekä ohjeiden ja komentojen järjestyksen tarkkaa seuraamista, etenkin jos ei usein käytä komentoriviä tai käytä tekstipohjaisia käyttöjärjestelmiä.

Manuaalisia palvelimia konfiguroitaessa jouduttiin välillä katsomaan ohjeita ja pohtimaan mitä kohtaa oltiin suorittamassa. Useiden komentojen peräkkäin syöttäminen samalla ohjeita seuraten, voi olla suuressa määrässä palvelinten konfigurointia työlästä. Kaikista aikaa vievintä oli järjestelmän päivitysten hakeminen ja niiden asentaminen sekä LAMP-pinon pakettien asennus. Virheellisiä komentoja ei manuaalisia palvelimia konfiguroidessa juurikaan tapahtunut, paitsi virheellinen salasanan kirjoittaminen sekä väärän hakemistopolun tiedoston merkitseminen Apachen asetustiedostoon.

5.2 Ansible-työkalun konfiguroinnin tulokset

Ansiblen konfigurointivaiheessa suoritettiin ensin yksinkertainen pelikirja, joka asentaa vain LAMP-pinon kuuluvat paketit ja varmistaa, että Apache ja MySQL ovat päällä, muuttamatta niiden sisältöä tai muita muokkauksia tekemättä. Tehtiin myös yksinkertainen inventory-tiedosto, jossa määriteltiin kohdepalvelimen IP-osoite, kohdekoneella operoiva ja muokkauksia tekevä käyttäjä sekä hakemistopolku ohjauspalvelimella toimivan käyttäjän yksityiseen SSH-avaimeen. Pelikirjoja suorittamalla normaalisti, lisäämällä '—check' komennon loppuun, Ansible suoritti pelikirjan tekemättä kuitenkaan mitään muokkauksia tai asennuksia kohdepalvelimeen.

Kuvassa 3 on yksinkertainen pelikirja, joka asentaa LAMP-pinon paketit ja varmistaa Apachen sekä MySQL:n tilan.

Kuva 3 Yksinkertainen pelikirja

```
GNU nano 4.8      testi-playbook.yml
---
- hosts: webservers
  become: yes
  tasks:
    - name: Update and upgrade apt packages
      apt:
        update_cache: yes
        upgrade: 'dist'

    - name: Install Apache2
      apt:
        name: apache2
        state: present

    - name: Install MySQL Server
      apt:
        name: mysql-server
        state: present

    - name: Install PHP and modules
      apt:
        name:
          - php
          - libapache2-mod-php
          - php-mysql
        state: present

    - name: Ensure Apache is running and enabled
      systemd:
        name: apache2
        state: started
        enabled: yes

    - name: Ensure MySQL is running and enabled
      systemd:
        name: mysql
        state: started
        enabled: yes
```

Pelikirjojen ja halutessa inventory-tiedostojen suoritus tapahtui komentoriviltä yksinkertaisella komennolla. Jos pelikirjan lisäksi haluaa suorittaa tietyn inventory-tiedoston, tapahtuu se lisäämällä pelikirjaa ennen "-i" ja halutun inventory-tiedoston nimi.

Kuva 4 Komento, joka suorittaa pelikirjan inventory-tiedoston kanssa

```
student@DESKTOP-P336IFS:~/ansible$ ansible-playbook -i varasto.ini pelikirja.yml
```

Tutkimuksessa oikeiden sisältöjen etsimiseen pelikirjaa varten käytettiin useampi päivä aikaa. Vaadittujen vaiheiden, jotka manuaalisesti pystyttiin suorittamaan, ei löytynyt vastaavia ohjeita Ansiblella konfiguroitaviin palvelimiin. Vaiheet, joita ei saatu pelikirjalla suoritettua liittyivät MySQL pääkäyttäjän salasanan asettamiseen, `mysql_secure_installation` vaiheiden suorittamiseen, MySQL:n uuden käyttäjän ja taulujen lisäämiseen. Myöskään viimeistä vaihetta, jossa tietokantaan uudella käyttäjällä syötetyt tiedot lisätään PHP-skriptiin, joka näyttää verkkosivuilla tietokantataulun sisällön ei saatu pelikirjalla suoritettua. Vaihteita, jotka tarkastavat vastaako verkkosivu pyyntöihin ei myöskään saatu toimimaan.

Ansiblen asentaminen WSL:ään eli ohjauspalvelimeen oli helppoa. Ansiblen optimaalisen toimivuuden kannalta luotiin ohjauspalvelimella ssh-avaimet yksinkertaisella `'ssh-keygen'`-komennolla. Komento luo julkisen- ja yksityisen avaimen, joista julkinen SSH-avain lisättiin manuaalisesti kolmeen Ubuntu-kohdepalvelimeen komennolla `'ssh-copy-id'`, joka kopioi avaimen valitulle palvelimelle.

Pelikirjaa, joka sisälsi MySQL:n muokkaustoimintoja suorittaessa saatiin virheilmoituksia, jotka kertoivat puuttuvista moduuleista tai syntaksivirheistä. Näistä molemmat oli moneen kertaan varmistettu, että ovat kunnossa ja tarvittavat paketit sekä moduulit on asennettu. Siitä huolimatta näitä tietokannan ja järjestelmän turvallisuuden kannalta tärkeitä vaihteita ei saatu toimimaan odotetulla tavalla.

Toimiva pelikirja, joka kattaa suuren osan manuaalisessa konfiguroinnissa suoritetuista vaiheista saatiin kuitenkin tehtyä. Vaiheet, jotka saatiin toimimaan, tarjosivat suuntaa antavan tuloksen vertailua varten.

Kuvassa 5 pelikirja suorittaa vaiheet: Järjestelmän päivitysten hakeminen ja asennus, Apache2:n asennus, Palomuurisäännön asettaminen Apachea varten, PHP:n ja tarvittavien moduulien asennus, MySQL-tietokantapalvelimen asennus sekä Virtuaalipalvelimen hakemiston luominen.

Kuva 5 Pelikirja osa 1

```
GNU nano 4.8 pelikirja.yml
--
- name: Complete LAMP Setup with VirtualHost and PHP Testing
  hosts: webservers
  become: true

  tasks:
    - name: Update and upgrade apt packages
      apt:
        update_cache: yes
        upgrade: dist

    - name: Install Apache2
      apt:
        name: apache2
        state: present

    - name: Allow Apache through the firewall
      ufw:
        rule: allow
        port: 80
        src: 172.18.138.66

    - name: Install PHP and required modules
      apt:
        name:
          - php
          - libapache2-mod-php
          - php-mysql
        state: present

    - name: Install MySQL Server
      apt:
        name: mysql-server
        state: present

    - name: Create directory for virtualhost
      ansible.builtin.file:
        path: /var/www/ansible.test
        state: directory
        owner: "{{ ansible_ssh_user }}"
        group: "{{ ansible_ssh_user }}"
        mode: "0755"
```

Kuvassa 6 pelikirja suorittaa seuraavat vaiheet: Apache-virtuaalipalvelimen asetusten määrittäminen, Apache pääsivun lisääminen, PHP-tietojen tiedoston luominen sekä PHP-tietoja sisältävän tiedoston poistaminen.

Kuva 6 Pelikirja osa 2, loput suoritetuista vaiheista

```
- name: Set up Apache virtual host
blockinfile:
  path: /etc/apache2/sites-available/ansible.test.conf
  create: yes
  block: |
    <VirtualHost *:80>
      ServerName ansible
      ServerAlias www.ansible.test
      ServerAdmin webmaster@localhost
      DocumentRoot /var/www/ansible.test
      ErrorLog ${APACHE_LOG_DIR}/error.log
      CustomLog ${APACHE_LOG_DIR}/access.log combined
    </VirtualHost>

- name: Enable new site
  shell: a2ensite ansible.test && a2dissite 000-default && systemctl reload apache2

- name: Add index.html
  ansible.builtin.copy:
    dest: /var/www/ansible.test/index.html
    content: |
      <html>
      <head>
        <title>Ansible</title>
      </head>
      <body>
        <h1>Toimii</h1>
        <p>This is the landing page of <strong>ansible.test</strong>.</p>
      </body>
      </html>

- name: Create PHP info file
  ansible.builtin.copy:
    dest: /var/www/ansible.test/info.php
    content: |
      <?php
      phpinfo();
      ?>

- name: Remove PHP info file
  ansible.builtin.file:
    path: /var/www/ansible.test/info.php
    state: absent
```

Kuvassa 7 on tutkimuksessa pelikirjan kanssa käytetty INI-tyyppinen inventory-tiedosto, jossa on määritelty kohdepalvelinten IP-osoitteet, kohdepalvelimella operoiva käyttäjä sekä yksityisen SSH-avaimen hakemistopolku.

Kuva 7 Inventory-tiedosto

```
GNU nano 4.8                                varasto.ini
[webservers]
ansible1 ansible_host=172.18.137.21
ansible2 ansible_host=172.18.138.141
ansible3 ansible_host=172.18.138.145

[webservers:vars]
ansible_ssh_user=arttu
ansible_ssh_private_key_file=/home/student/.ssh/id_rsa
```

Ansiblen arkkitehtuurin osat, joita tutkimuksessa käytettiin, eli inventory-tiedostojen, pelikirjojen ja konfiguraatitiedostojen ymmärtäminen oli selkeää. YAML- syntaksi mitä pelikirjat käyttävät aiheutti tutkimuksen tekijälle turhautumista. Syntaksi on todella tarkka, esimerkiksi kaikki pelikirjassa määritettyjen tehtävien täytyy alkaa samalta riviltä ja tiedosto ei saa sisältää tabulaattorimerkkejä, vaan ainoastaan välilyöntejä.

Ansiblen käyttäminen oli yksinkertaista, mutta oikeiden vaiheiden löytäminen pelikirjaan ja niiden muokkaaminen tuotti tutkimuksen tekijälle haasteita. Tutkimuksen tekijällä ei myöskään ollut kuin hieman kokemusta työkalun käytöstä sekä YAML-syntaksista, mikä lisäsi tutkimuksen haasteellisuutta. Pelikirjojen syntaksin korjailuun, tarkastamiseen ja testisuoritukseen kului paljon aikaa.

6 Tulosten yhteenveto

Tuloksena syntyvä vertailu epäonnistui osittain, koska kaikkia vaadittavia manuaalisessa konfiguroinnissa suoritettuja vaiheita ei saatu suoritettua odotetulla tavalla. Tuloksena syntyi kuitenkin molemmilla tavoilla suoritettuihin konfigurointeihin käytetyt ajat sekä havaintoja automaation käytöstä tapauksessa, jossa konfiguroidaan pieni määrä palvelimia. Vaikka muutama konfigurointivaihe jäi Ansiblen pelikirjassa uupumaan, on tulos, jossa vertaillaan molempiin konfiguraatiotapoihin käytettyjä aikoja silti lähellä totuutta. MySQL vaiheet ja verkkosivujen tilan tarkastaminen ovat prosesseja, jotka eivät vie paljon aikaa, koska niissä ei tarvitse ladata tai asentaa mitään ja muutokset tulevat heti voimaan. Manuaalisesti ainoa aikaa vievä osuus näissä vaiheissa oli tietojen manuaalisesti syöttämiseen kuluva aika.

Ansible-työkalulla kolmen palvelimen onnistuneesti suoritettuihin konfigurointivaiheisiin kului aikaa 12 minuuttia ja 38 sekuntia. Ansiblella suoritettiin pelikirja kolmeen palvelimeen yhtäaikaaisesti, mikä oli yksi ohjelman hyvistä puolista. Manuaalisesti kolmen palvelimen konfigurointiin kului yhteensä n. 59 minuuttia. Tulos osoittaa, että manuaalisen ja automatisoidun konfigurointimenetelmien välillä on merkittäviä eroja suoritusaikojen osalta jopa pientä määrää palvelimia konfiguroidessa.

Manuaalisessa konfiguroinnissa virheiden vähyyys johtui pääosin siitä, että vaiheet suoritettiin rauhallisesti yrittäen nimenomaan välttää inhimillisiä virheitä. Tämä puolestaan jonkin verran lisäsi suorituksiin käytettyä aikaa. Palvelimia myös konfiguroitiin pieni määrä, mikä vaikutti inhimillisten virheiden vähyyteen. Todennäköisesti esimerkiksi kymmenen peräkkäisen palvelimen manuaalisessa konfiguraatiossa virheitä ilmenisi enemmän.

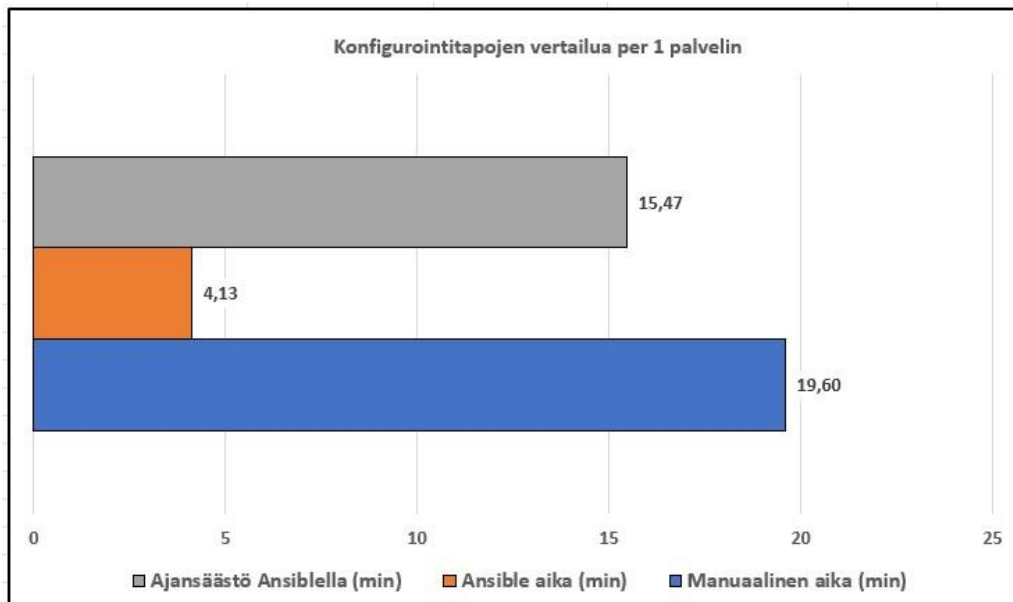
Ottaen kuitenkin huomioon kuinka kauan aikaa kului Ansiblen pelikirjojen testaukseen, syntaksin opetteluun sekä pelikirjojen vaiheiden muokkaamiseen, ei ole tutkimuksen kaltaisessa tapauksessa kannattavaa käyttää automaatiota. Ansiblen käyttöönotto sisälsi paljon opettelua, testausta ja vei paljon enemmän aikaa kuin manuaalinen konfigurointi. On hyvä miettiä konfiguroitavien palvelinten määrää missä automatisoitu ratkaisu olisi järkevää ottaa käyttöön, manuaalisen konfigurointitavan sijaan. Ansiblella pelikirjojen testaukseen ja toimivien lähteiden löytämiseen kulutettiin moninkertainen määrä aikaa verrattuna manuaalisen osuuden suorittamiseen. Tilanteessa, jossa tarvitsisi konfiguroida suuri määrä palvelimia, Ansiblen käytön tarve korostuisi enemmän. Etenkin tapauksissa, joissa useita palvelimia pitäisi jatkuvasti hallita ja suorittaa muokkauksia, työkalu olisi erittäin tarpeellinen.

Jos lasketaan yksittäisen manuaalisesti konfiguroituun palvelimeen käytetty aika tutkimuksesta saatujen tulosten keskiarvon perusteella, ajaksi tulisi 19,6 minuuttia. Ansiblella kolmen palvelimen konfigurointiin käytetystä ajasta yhteen palvelimeen käytetty aika olisi 4,13 minuuttia. Tuloksesta voidaan päätellä, että Ansible suorittaa tutkimuksen tapaisen konfigurointiprosessin 4,7 kertaa nopeammin. Jos kuvitellaan tilanne, missä konfiguroitaisiin esimerkiksi sata tutkimuksen kaltaista palvelinta, Ansiblen käytön hyödyt olisivat ajallisesti ja työmäärällisesti merkittävät manuaaliseen konfigurointitapaan verrattuna. Esimerkiksi sataa palvelinta manuaalisesti konfiguroidessa myös inhimillisten virheiden määrä todennäköisesti kasvaisi, kuten virheellisten konfiguraatioiden ja virheellisten syötettyjen salasanojen määrä.

Alapuolella esitetyt pylväskaaviot kuva 8 ja kuva 9, eivät täytä tilastollisesti hyvää tapaa laskea, koska mittausaineisto ja konfiguroitujen palvelinten määrä on pieni, mutta ne ovat suuntaa antavia. Pylväskaavioissa tulokset on laskettu ainoastaan tutkimuksessa käytetyn ympäristön mukaan.

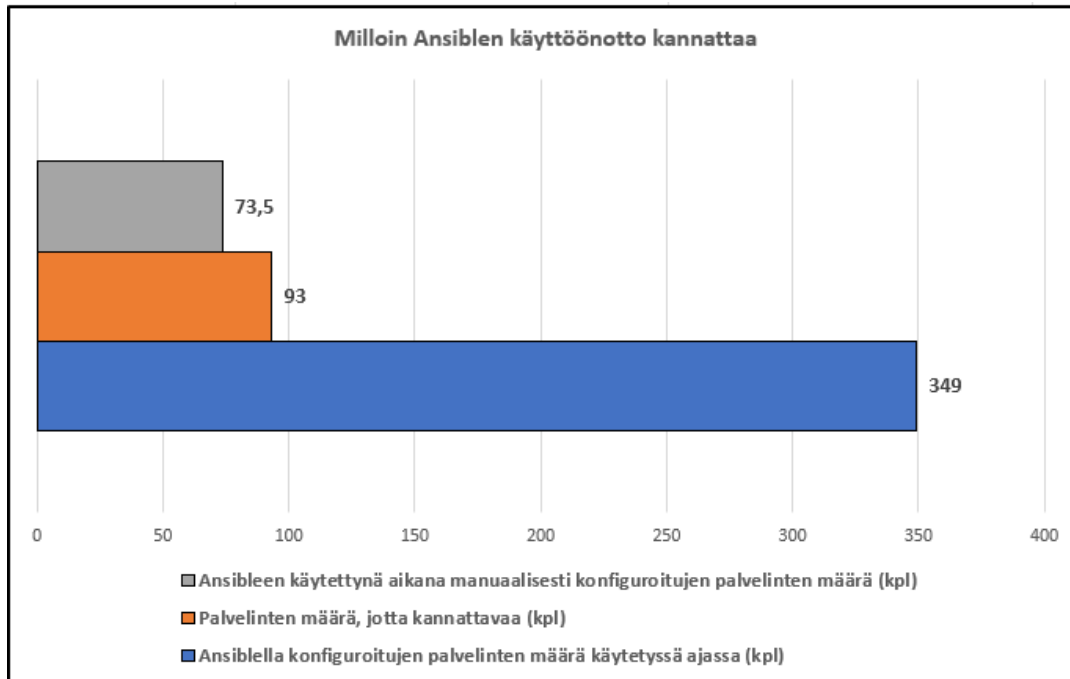
Kuvassa 8 on verrattuna konfigurointitapojen ajallista eroavaisuutta. Vertailussa on laskettu molempien konfigurointitapojen yhden palvelimen konfigurointiin käytetty keskimääräinen aika saaduista tutkimuksen tuloksista.

Kuva 8 Konfigurointitapojen ajallinen vertailu



Kuvassa 9 kaavio, jossa on laskettu ns. kannattavuusraja, milloin Ansiblen käyttöönotto olisi tutkimuksen kaltaisessa tapauksessa kannattavaa. Kaaviosta selviää, että 93 kpl konfiguroitavia palvelimia on määrä, jolloin Ansiblen käyttöönotto olisi kannattavaa tutkimuksen tapaisessa tilanteessa.

Kuva 9 Ansiblen käyttöönoton kannattavuusraja



7 Johtopäätökset

Opinnäytetyön tarkoituksena oli tutkia Ansible-työkalun käyttöä palvelinten konfiguroinnissa ja syventää omaa osaamistani aiheeseen liittyen. Opinnäytetyössä myös haastoin itseäni tutkimalla tätä itselleni vierasta automatisointityökalua. Teoriaosuudessa syvensin omaa osaamistani palvelimiin ja automatisointiin liittyen, sekä hain pohjatietoa Ansiblesta. Teoriaosuus mielestäni tuki suurimmaksi osaksi käytännönsuudessa toteutettua tutkimusta. Vaikka tutkimuksessa ei Ansiblella täysin onnistuttu suorittamaan kaikkia samoja konfigurointivaiheita kuin manuaalisella tavalla, sain silti hyvää kokemusta ohjelman käytöstä sekä omakohtaisia näkemyksiä sen haasteista ja hyödyistä. Sain myös tietoa, siitä milloin automatisointia olisi hyödyllistä ottaa käyttöön ja milloin manuaalinen konfigurointitapa voisi olla parempi vaihtoehto.

Tutkimuskysymyksiin saatiin vastaus jo osittain teoriaosuudesta ja Ansible-työkalun käyttöön liittyviin kysymyksiin tutkimuksesta. Työssä kaikkein haastavinta itselleni oli Ansiblen pelikirjojen syntaksi, jonka korjailuun ja tarkistukseen kului paljon aikaa, sekä toimivien suoritettavien vaiheiden löytämiseen pelikirjaan. Ansiblen käyttöönotto eli kaikkien vaiheiden saaminen pelikirjaan ja pelikirjan toimivuuden testaus oli myös haastavaa ja ennen kaikkea aikaa vievää.

Ohjelman käyttö ja arkkitehtuurin ymmärtäminen ei tuottanut haasteita, koska ohjelman käyttö ei itsessään ollut haastavaa. Käytettyäni tutkimukseen enemmän aikaa, kuin olin suunnitellut, en saanut muutamia vaiheita Ansiblella toimimaan. Ohjelman käyttöönottoon olisi pitänyt varata enemmän aikaa, jotta nämä vaiheet olisi saatu suoritettua. Tutkimuksessa olisi voitu myös konfiguroida useampi palvelin laajemman vertailun saavuttamiseksi, mutta valitsemani testiympäristön rajallisten resurssien vuoksi se ei tässä tapauksessa ollut mahdollista.

Työssä teoriaosuudessa löysin hyvin tietoa aiheisiin luotettavista lähteistä ja niistä työhön kirjoittaminen ei tuottanut haasteita. Tämä oli kuitenkin myös aikaa vievää, koska kaikki käyttämäni lähteet olivat englanninkielisiä ja niiden sisäistäminen oli luonnollisesti työläämpää kuin suomenkielisten lähteiden.

8 Jatkotutkimus

Jatkotutkimuksessa palvelimia voitaisiin konfiguroida molemmilla tavoilla suurempi määrä, esimerkiksi 10–100 kappaletta. Tämä lisäisi aineiston laajuutta ja antaisi paremman käsityksen automatisoinnin ja manuaalisen konfiguroinnin eroista laajemmassa mittakaavassa. Konfigurointivaiheita voisi myös lisätä useampia, jotta inhimillisten virheiden riskien määrää saataisiin kasvatettua ja näin voitaisiin paremmin arvioida Ansible-työkalun tuomia etuja inhimillisten virheiden vähentämisessä. Olisi tärkeää pyrkiä ratkaisemaan tässä tutkimuksessa esiin tulleet ongelmat ja saada ne vaiheet jatkotutkimuksessa toimiviksi, joita ei tässä tutkimuksessa saatu toimimaan. Tämä voisi antaa vielä tarkempaa tietoa Ansible-työkalun käytöstä ja sen tuomista hyödyistä verrattuna manuaaliseen konfigurointiin.

Jatkotutkimuksessa voisi olla mukana useampi henkilö suorittamassa konfigurointeja. Tällä tavalla voitaisiin vertailla useamman kuin yhden tutkijan virheiden määrää sekä ajallisia eroavaisuuksia. Useamman osallistujan ansiosta saataisiin myös erilaisia mielipiteitä Ansiblen käytöstä ja käyttöönotosta, mikä rikastaisi tutkimustuloksia ja tarjoaisi monipuolisemman näkökulman työkalun käyttöön. Jatkotutkimuksessa olisi myös hyvä varata ajallisesti enemmän aikaa tutkimusta varten. Laajempi aikataulu mahdollistaisi syvällisemmän analyysin ja paremmin suunnitellut kokeet, mikä lisäisi tutkimuksen luotettavuutta ja kattavuutta.

Konfigurointia voitaisiin jatkotutkimuksessa suorittaa myös muissa kuin Linux-ympäristöissä, esimerkiksi Windows- ja pilviympäristöissä. Tämä antaisi arvokasta tietoa eri ympäristöjen eroavaisuuksista ja voisi auttaa selvittämään, missä ympäristössä Ansible toimii parhaiten. Samalla saataisiin käsitys siitä, kuinka hyvin Ansible soveltuu monialustaisiin ympäristöihin ja missä määrin sen käyttöönottoaminen on kannattavaa eri käyttötilanteissa.

Lisäksi jatkotutkimuksessa voitaisiin tarkastella Ansible-työkalun yhteensopivuutta muiden vastaavien IT-automatisointityökalujen kanssa, kuten Puppet tai Chef ja tutkia, kuinka hyvin Ansible toimii osana laajempaa IT-infrastruktuuria. Tämä voisi auttaa tunnistamaan mahdollisia synergiaetuja ja haasteita, jotka liittyvät monimutkaisempiin automatisointikokonaisuuksiin.

Lähteet

Ansible Community Documentation, (n.d.). *Ansible playbooks*. Noudettu 10. helmikuuta 2024, osoitteesta

https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_intro.html

Ansible Community Documentation, (n.d.). *How to build your inventory*. Noudettu 11. helmikuuta 2024, osoitteesta

https://docs.ansible.com/ansible/latest/inventory_guide/intro_inventory.html

Ansible.com (n.d.). *How Ansible works*. Noudettu 21. tammikuuta 2024, osoitteesta

<https://www.ansible.com/overview/how-ansible-works>

Badkar, A. (30 kesäkuuta, 2023). *Advantages and disadvantages of automation*.

Simplilearn.Com. Noudettu 20. tammikuuta 2024, osoitteesta

<https://www.simplilearn.com/advantages-and-disadvantages-of-automation-article>

BasuMallick, C. (15 heinäkuuta, 2022). *Ansible and how does it work*. Noudettu 15.

helmikuuta 2024, osoitteesta <https://www.spiceworks.com/tech/devops/articles/what-is-ansible/>

DigitalOcean (27 helmikuuta, 2024.-a). *How to install linux, apache, mysql, php (Lamp) stack on ubuntu*. Noudettu 16. huhtikuuta 2024, osoitteesta

<https://www.digitalocean.com/community/tutorials/how-to-install-lamp-stack-on-ubuntu>

DigitalOcean. (16 kesäkuuta, 2021). *How to configure ssh key-based authentication on a linux server*. Noudettu 3. maaliskuuta 2024, osoitteesta

<https://www.digitalocean.com/community/tutorials/how-to-configure-ssh-key-based-authentication-on-a-linux-server>

DigitalOcean. (27 helmikuuta, 2024.-b). *How to set up a firewall with ufw on ubuntu*.

Noudettu 16. huhtikuuta 2024, osoitteesta

<https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-ubuntu>

Geerling, J. (2022). *Ansible for DevOps: Server and configuration management for humans* (2nd edition). Leanpub.

- Heidi, E. (17 joulukuuta, 2019.). *How to use ansible to install and set up lamp on ubuntu 18.04*. DigitalOcean. Noudettu 18. huhtikuuta 2024, osoitteesta <https://www.digitalocean.com/community/tutorials/how-to-use-ansible-to-install-and-set-up-lamp-on-ubuntu-18-04>
- IBM. (11 joulukuuta, 2023). *What is yam!?* Noudettu 11. huhtikuuta 2024, osoitteesta <https://www.ibm.com/topics/yam!>
- Indeed Editorial Team. (31 heinäkuuta, 2023). *Types of Computer Servers and How They Function*. Noudettu 12. helmikuuta, osoitteesta <https://www.indeed.com/career-advice/career-development/types-of-servers>
- Jethva, H. (26 tammikuuta, 2024). *Installing and configuring lamp on ubuntu 22.04 with ansible*. Atlantic.Net. Noudettu 19. huhtikuuta 2024, osoitteesta <https://www.atlantic.net/dedicated-server-hosting/installing-and-configuring-lamp-on-ubuntu-22-04-with-ansible/>
- Jones IT. (1 huhtikuuta, 2023). *How to configure a network server*. Noudettu 25. tammikuuta 2024, osoitteesta <https://www.itjones.com/blogs/how-to-configure-a-network-server>
- Kadima, S. (9 heinäkuuta, 2023). *Ansible: History of ansible*. Noudettu 17. tammikuuta 2024, osoitteesta Medium. <https://medium.com/@kadimasam/ansible-history-of-ansible-4b16208c9188>
- LogicMonitor. (28 helmikuuta, 2023). *9 reasons Linux is a popular choice for servers*. LogicMonitor. Noudettu 18. helmikuuta 2024, osoitteesta <https://www.logicmonitor.com/blog/9-reasons-linux-is-a-popular-choice-for-servers>
- MDN. (3 heinäkuuta, 2023). *What is a web server? - Learn web development*. Noudettu 15. maaliskuuta 2024, osoitteesta https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Web_mechanics/What_is_a_web_server
- MonoVM (2 helmikuuta, 2024). *Everything you must know about a linux server*. Noudettu 18. helmikuuta 2024, osoitteesta <https://monovm.com/blog/what-is-a-linux-server/>

Monteclaro, A.J. (15 syyskuuta, 2023). *What is a server? | definition, types, and features.*

Noudettu 5. helmikuuta 2024, osoitteesta <https://www.serverwatch.com/guides/what-is-a-server/>

one.com. (n.d.). *Mikä on IPv6?* Noudettu 22. helmikuuta 2024, osoitteesta

<https://www.one.com/fi/webhotelli/mika-on-ipv6>

Opensource.com. (n.d.). *What is Ansible?* Noudettu 17. tammikuuta 2024, osoitteesta

<https://opensource.com/resources/what-ansible>

Paessler. (n.d.). *Server—Definition and details.* Noudettu 25. tammikuuta 2024, osoitteesta

<https://www.paessler.com/it-explained/server>

platform.sh (n.d.). *What yaml is.* Noudettu 29. toukokuuta 2024, osoitteesta

<https://docs.platform.sh/learn/overview/yaml/what-is-yaml.html>

Red Hat. (1 elokuuta, 2023.). *What is an ansible playbook?* Noudettu 9. helmikuuta 2024,

osoitteesta <https://www.redhat.com/en/topics/automation/what-is-an-ansible-playbook>

Red Hat. (11 tammikuuta, 2024.). *What is an Ansible module—And how does it work?*

Noudettu 2. helmikuuta 2024, osoitteesta

<https://www.redhat.com/en/topics/automation/what-is-an-ansible-module>

Red Hat. (16 joulukuuta, 2022). *What is a Linux server?* Noudettu 18. helmikuuta 2024,

osoitteesta <https://www.redhat.com/en/topics/linux/linux-server>

Red Hat. (20 syyskuuta, 2023.). *Understanding automation.* Noudettu 18. tammikuuta 2024,

osoitteesta <https://www.redhat.com/en/topics/automation>

Red Hat. (27 helmikuuta, 2019). *What is the Linux kernel?* Noudettu 17. helmikuuta 2024,

osoitteesta <https://www.redhat.com/en/topics/linux/what-is-the-linux-kernel>

Red Hat. (3 tammikuuta, 2023). *Understanding linux.* Noudettu 16. helmikuuta 2024,

osoitteesta <https://www.redhat.com/en/topics/linux>

Rometsch, B. (16 helmikuuta, 2021). *How Ansible got started and grew.* Noudettu 16.

helmikuuta 2024, osoitteesta <https://opensource.com/article/21/2/ansible-origin-story>

Scribe. (n.d.). *Manual vs. Automation process: Which one is better?* Noudettu 11. helmikuuta 2024, osoitteesta <https://scribehov.com/library/manual-vs-automation>

Sen, K. (22 tammikuuta, 2024). *10 Essential Steps for Configuring a New Server*. Noudettu 20. helmikuuta 2024, osoitteesta <https://www.upguard.com/blog/10-essential-steps-for-configuring-a-new-server>

SolarWinds. (n.d.). *What is a web server and how does it work?* Noudettu 14. maaliskuuta 2024, osoitteesta <https://www.solarwinds.com/resources/it-glossary/web-server>

Vaadata. (19 heinäkuuta, 2022). *Securing a server: Risks, challenges and best practices*. Noudettu 5. helmikuuta 2024, osoitteesta <https://www.vaadata.com/blog/how-to-secure-a-server/>

Vijh, H. (30 kesäkuuta, 2022). *Top 7 benefits of it automation you should know—Hexaware*. Noudettu 11. helmikuuta 2024, osoitteesta <https://hexaware.com/blog/benefits-of-it-automation/>

VMware. (n.d.). *What is it automation?* Noudettu 18. tammikuuta 2024, osoitteesta <https://www.vmware.com/topics/glossary/content/it-automation.html>

Zenarmor. (12 lokakuuta, 2023). *Linux explained. Distributions, differences, benefits, security*. Noudettu 18. helmikuuta 2024, osoitteesta <https://www.zenarmor.com/docs/linux-tutorials/what-is-linux>

