Raspberry Pi 4 -palvelimen automatisoitu konfigurointi käyttäen Ansiblea

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Ammattikorkeakoulututkinnon opinnäytetyö

Tietojenkäsittelyn koulutus

kevät 2022

Sampsa Selin

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Tietojenkäsittelyn koulutus Tiivistelmä
Tekijä Sampsa Selin Vuosi 2022
Työn nimi Raspberry Pi 4 -palvelimen automatisoitu konfigurointi käyttäen Ansiblea
Ohjaaja Ismo Turve

Opinnäytetyön tarkoituksena oli selvittää, miten Ansible-ohjelman ydintoimintoja käyttäen voidaan automatisoida paikalliseen IT-infrastruktuuriin sisältyvän, yleisesti käytössä olevan Linux-palvelinkäyttöjärjestelmän yhteys- käyttäjä- ja koventamiskonfigurointeja. Työn tavoitteena oli luoda uudelleenajettavia Ansible-pelikirjoja, jotka toimivat samalla tehtyjen konfigurointien dokumentaatioina. Opinnäytetyön keskeinen tavoite oli opiskella tämän itselle uuden, ja yritysmaailmassa käytetyn Ansible-ohjelman peruskäyttöä, sekä luoda ohjeistus, jota seuraamalla myös muut järjestelmänhallintaa opiskelevat henkilöt voivat rakentaa paikallisen Ansible-harjoitusympäristön.

Opinnäytetyön tietopohja koostuu projektissa käytetyn Raspberry Pi 4 -minitietokoneen ominaisuuksien kuvauksesta, Raspberry Pi imager -ohjelman sisältävien palvelinkäyttöjärjestelmien vertailusta, sekä asennettavaksi valitun käyttöjärjestelmän tietoturvan tasoa parantavien koventamistoimien listaamisesta (Hardening). Tietopohja kuvaa myös Ansible-ohjelman ydintoiminnallisuuksia, ja ohjelman keskeistä osallisuutta modernissa DevOps -ohjelmistokehitysmetodissa, jossa sen avulla voidaan hallita ohjelmallisesti IT-infrastruktuuria. Tietopohja kuvaa lukijalle näitä IaC-työkalujen käytön hyötyjä (Infrastructure as Code). Opinnäytetyö on toiminnallinen, ja käytännön osuus kuvaa tehdyt asennukset, sekä luodut Ansible-pelikirjat, ja niiden toteutuneet ajoraportit.

Johtopäätöksenä voidaan todeta, että Raspberry Pi 4 soveltuu teknisiltä ominaisuuksiltaan hyvin yleisesti käytössä olevalle, ja pitkäaikaisesti tuetulle 64-bittiselle Ubuntu-palvelinkäyttöjärjestelmälle, jota voidaan hallita WSL2 ympäristöön asennetulla Ansiblella. Tässä paikallisessa Ansible-harjoitusympäristössä voidaan ajaa Ansiblen pelikirjoja, ja näin automatisoida sekä dokumentoida yhteys- käyttäjä- sekä koventamiskonfigurointeja.

Avainsanat Ansible, Raspberry Pi 4 B, Linux-palvelin, Automatisoitu konfigurointi
Sivut 42 sivua ja liitteitä 5 sivua

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Degree Programme in Business Information Technology — Abstract
Author Sampsa Selin — Year 2022
Subject Automated configuration of a Raspberry Pi 4 Server using Ansible
Supervisor Ismo Turve

The purpose of this thesis was to find out how the connection, user, and hardening configurations of a widely used local Linux server can be automated using the core functionalities of Ansible. The object of this project was to create rerunnable Ansible playbooks that also function as a documentation of the configurations. The main goal was to study the basic functionalities of Ansible that is used in enterprises, and to create a guide that other system administration students can follow to set up a local Ansible training environment.

The knowledge base of this thesis consists of a description of the specifications of the Raspberry Pi 4 minicomputer used in the project, a comparison of the server operating systems that are included in the Raspberry Pi Imager software, and a list of hardening measures to improve the security of the operating system selected for installation. The knowledge base also describes the core functionalities of Ansible, and its central part in the modern DevOps software development method where it can be used for programmatically manage the IT infrastructure. The benefits of using IaC (Infrastructure as Code) tools like Ansible are described to the user. The thesis is practical, and it describes the installations made, as well as the created Ansible Playbooks and the reports of their runs.

In conclusion, Raspberry Pi 4 is technically very compatible with the widely used and long term supported 64-bit Ubuntu server operating system that can be managed with Ansible installed in the WSL2 environment. In this local Ansible training environment, Ansible's Playbooks can be run to automate and document connection, user, and hardening configurations.

Keywords Ansible, Raspberry Pi 4 B, Linux server, Automated configuration
Pages 42 pages and appendices 5 pages

Sanasto

ARM	Advanced RISC Machines, prosessoriarkkitehtuuri, jota käytetään Raspberry Pi:ssä
Cloud-init	Työkalu, joka käynnistyy ensimmäisessä käynnistyksessä, ja jota voidaan käyttää lisäämään automaattisesti Ubuntu-pilvipalvelimen cloud-config.txt -tiedostoon käyttäjäasetuksia ja konfigurointeja, kuten yksityisen SSH-avaimen
CMDB	Configuration Management DataBase, järjestelmän konfigurointi-informaation sisältävä tietokanta
DevOps	Development and Operations, moderni ohjelmistokehitysmetodi
ESM	Extended Security Maintenance, Ubuntun laajennetun tuen ohjelma
GNU/Linux	UNIX-pohjainen avoimen lähdekoodin käyttöjärjestelmä, joka käyttää Linux-kerneliä
GPIO	General Purpose Input Output, RasPin pinnikisko, jota käytetään ulkoisiin laitekytkentöihin
IaC	Infrastructure as Code, ohjelmallisesti luotu ja hallittu IT-infrastruktuuri
IBM	International Business Machines Corporation, Amerikkalainen teknologiayritys
Imager	Ohjelma, kuten Raspberry Pi Imager tai Win32 Disk Imager, jolla käyttöjärjestelmä asennetaan microSD-kortille tai varmuuskopioidaan microSD-kortilta tietokoneelle
Industry 4.0	Teollinen esineiden internet -viitekehys
IoT	Internet of Things, esineiden internet
Kubernetes	Konttiteknologiaa käyttävien ohjelmien orkesterointiohjelma
LTS	Long-Term Support, Ubuntu-käyttöjärjestelmän pitkäaikainen tuki
microSD	Secure Digital -muistikorttityyppi
Playbook	Pelikirja (Ansible)
RasPi	Yleisesti käytetty lyhenne Raspberry Pi -minitietokoneille
RSA	Rivest-Shamir-Adleman, julkisen avaimen salausmenetelmä
SCP	Secure Copy Protocol, tiedonsiirtoprotokolla
SFTP	Secure File Transfer Protocol, tiedonsiirtoprotokolla
SHA-512	Secure Hash Algorithm, 512-bittinen salausalgoritmi
SSH	Secure Shell, protokolla salatun tunneliyhteyden luomiseen kahden laitteen välille avainparia käyttäen
Windows Terminal	Kustomoitava Cmd, PowerShell, ja bash -komentokehotyökalu
WSL2	Windows Subsystem for Linux, GNU/Linux-ympäristö, jota ajetaan Windows-koneella
YAML	Yet Another Markup Language, asetustiedostoissa usein käytetty kieli

Sisällys

Komennot, kuvat, ohjelmakoodit ja taulukot

Liitteet

1 Johdanto

Linux-palvelinkäyttöjärjestelmän yhteys- ja käyttäjäasetusten konfigurointi sekä palvelimen koventaminen sisältävät usein toistuvia vaiheita ja inhimillisen virheen mahdollisuus on suuri. Reaalimaailmassa on myös pystyttävä konfiguroimaan useita koneita yhtäaikaisesti. Tämän opinnäytetyön tarkoitus on oppia itselle uuden ja yrityksissä käytetyn Ansible-ohjelman peruskäyttöä, jonka avulla näitä konfigurointivaiheita voidaan automatisoida.

Työssä tullaan käyttämään Raspberry Pi 4 -minitietokonetta paikallisessa lähiverkossa. Laitteelle asennettava palvelinkäyttöjärjestelmä tullaan valitsemaan vertailun perusteella, ja valintakriteereinä tullaan käyttämään laite- ja Docker Engine -yhteensopivuutta, sekä päivitysten ja internetistä löytyvien ohjeiden saatavuutta. Käyttöjärjestelmän yleistä esiintyvyyttä pilvipalveluntarjoajien alustoilla tullaan myös pitämään valintaa puoltavana kriteerinä.

Ansible tullaan asentamaan Windows 10 -koneen WSL2-ympäristöön hallintakoneeksi, johon luodaan ohjelman ajoa varten oma käyttäjätili. Työssä tullaan kuvaamaan miten etäkoneet, kuten Raspberry Pi 4 ryhmitellään Ansiblen käyttämään inventaariotiedostoon, ja miten yhteysmuuttujia, sekä vault-salasanatiedostoa käyttäen voidaan luoda yhteys etäkoneille. Työssä tullaan luomaan Ansible-pelikirjoja, joita ajamalla yhdelle Raspberry Pi 4 -etäkoneelle tullaan tekemään konfigurointeja. Nämä pelikirjat tulevat olemaan uudelleen ajettavia, ja tulevat myös toimimaan työssä tehtyjen konfigurointien dokumentaatioina.

Tutkimuskysymykset

- Millainen laite Raspberry Pi 4 on?
- Minkä käyttöjärjestelmän asentaminen Raspberry Pi 4:lle olisi tuen saatavuuden ja internetistä löytyvän dokumentaation kannalta edullisinta, ja suunnitellut asennukset olisivat kokonaisuus huomioiden mahdollisia?
- Mitkä ovat Ansiblen ydintoiminnallisuudet?
- Miten yhteys- käyttäjä- sekä koventamiskonfiguroinnit ajetaan Raspberry Pi 4:lle käyttäen Ansible-pelikirjoja?

2 Kehittämistyön tietoperusta

Tietoperustassa tutustutaan ensin Raspberry Pi 4 B minitietokoneen teknisiin ominaisuuksiin, ja laitteeseen erikseen hankittavissa ja asennettavissa olevaan PoE+ HAT -lisäosaan, jonka jälkeen siirrytään vertailemaan Raspberry Pi Imagerilla asennettavia ja työhön soveltuvia palvelinkäyttöjärjestelmiä.

Tämän jälkeen kuvataan valitun käyttöjärjestelmän yleisiä ominaisuuksia ja tuki- sekä julkaisuaikataulua. Valitun käyttöjärjestelmän yhteydessä tutkitaan myös Linux-käyttöjärjestelmän koventamiseen tähtääviä toimia, joilla käyttöjärjestelmän tietoturvaa voidaan parantaa.

Automatisoinnin yhteydessä esiintyy usein termejä, kuten IaC ja DevOps, joiden ideaa ja hyötyjä kuvataan omassa kappaleessaan. Teoriaosuuden loppupuolella tutkitaan tarkemmin automatisointityökaluksi valitun Ansiblen ydintoimintoja, jotta käytännön osuudessa tehtävät asennukset voitaisiin toteuttaa Ansiblea käyttäen.

2.1 Raspberry Pi 4 B

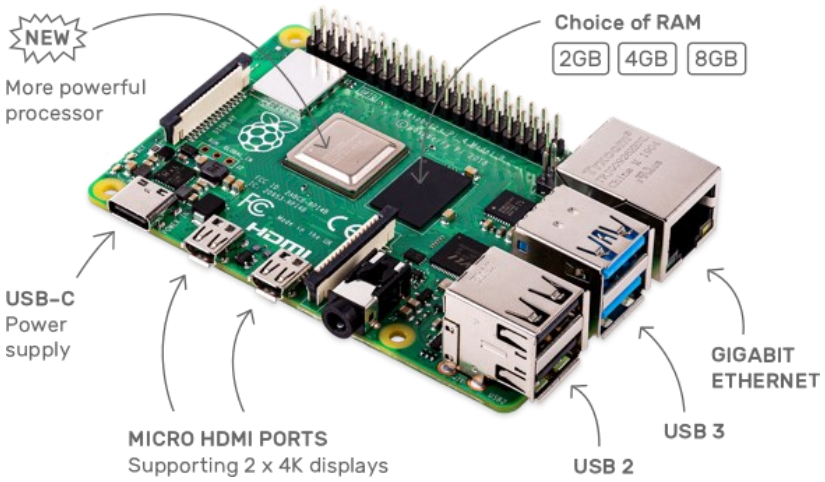
Raspberry Pi 4 B on valmistajan Raspberry Pi -tuoteperheen uusin ja tehokkain luottokortin kokoinen minitietokone. Aikaisempiin Raspberry Pi -malleihin verrattuna se on ensimmäinen, jonka käyttökokemus vastaa 32-bittisen työpöytä-PC:n suorituskykyä. Raspberry Pi 4 B tarjoaa aikaisempaan Raspberry Pi 3 B+ -malliin verrattuna huomattavasti nopeamman 64-bittisen ARM-pohjaisen neliydinprosessorin 1.5GHz kellotaajuudella, ja jopa 8 GB RAM-muistilla. (Datasheets.Raspberrypi, 2021)

Laitteen micro-HMDI -liittimiin voidaan kytkeä yhtä aikaa kaksi kappaletta näyttöjä, ja toistaa videoita jopa 4Kp60-resoluution tasolla. Raspberry Pi 4 Model B -tuote-esitteessä mainitaan muina keskeisinä ominaisuuksina myös USB 3.0, 2.4/5.0 GHz wlan, Bluetooth 5.0, ja gigabitin Ethernet-liitäntä PoE-tuella. (Datasheets.Raspberrypi, 2021)

RasPi 4 B on taakse päin yhteensopiva ja sen virrankulutus vastaa RasPi 3 B+ kulutuksen tasoa. Laitteesta löytyy edelleen standardimallinen 40-pinninen GPIO-liitäntä, microSD-

korttipaikka ja 5V DC USB-C virtaliitin. Kahden USB 3.0 liitännän lisäksi laitteessa on myös kaksi kappaletta USB 2.0 liitäntöjä, 2-kaistaiset kamera- ja näyttöportit, sekä audio portti.(Kuva 1) (Datasheets.Raspberrypi, 2021)

Kuva 1 Raspberry Pi 4 B -liitännät (Raspberrypi, n.d.-a)



Hankkimalla erillinen PoE HAT -lisäosa (Kuva 2), voidaan Ethernet-liitännän kautta dataliikenteen lisäksi tuoda RasPille myös virta. Raspberry Pi 3 B+ malleista lähtien minitietokoneen Ethernet-liitännän takana on 4-pinninen PoE-liitin, johon PoE HAT -lisäosa kytketään. Lisäosa pystyy muuntamaan PoE-liitännän kautta tulevan 37-57V DC -jännitteen RasPille sopivaksi 5V DC -jännitteeksi. Raspberry Pi Trading -yhtiön toimitusjohtaja Eben Upton kertoo Raspberry Pi-uutisissa heidän nähneensä Poe HATin harrastajien käytössä, sekä teollisissa automaatioratkaisuissa, joissa RasPi on sijoitettu haluttuun paikkaan kytkemällä vain Ethernet-kaapeli. (Upton, 2021)

Kuva 2 Raspberry Pi 4 B ja PoE+ HAT (Upton, 2021)

2.2 Käyttöjärjestelmävertailu ja Raspberry Pi Imager v1.6.2

Raspberry Pi Imager -ohjelma on ladattavissa Windows 10 -koneelle valmistajan virallisilta sivuilta. (Raspberrypi, n.d.-c)

Ohjelmaan tutustumisen tuloksena selvisi, että Raspberry Pi Imager -ohjelma on tarkoitettu Raspberry-tietokoneille asennettavien erilaisten ARM-prosessoriarkkitehtuurille soveltuvien 32- ja 64-bittisten käyttöjärjestelmien kirjoittamiseen microSD-kortille tai USB-tallennusvälineelle. Se ei tue käyttöjärjestelmän varmuuskopioimista microSD-kortilta tietokoneelle, joten siihen tarkoitukseen on käytettävä toista imager-ohjelmaa, kuten Win32Disk-Imageria. (Win32DiskImager, n.d.)

Raspberry Pi Imager tarjoaa ensisijaisesti asennettavaksi Debian-käyttöjärjestelmään pohjautuvaa Raspberry Pi OS (32-bit) -käyttöjärjestelmäänsä sekä työpöytäversiona että kevyenä versiona ilman työpöytää. Molemmat versiot ovat asennettavissa, joko uudemmalla Debian Bullseye, tai vanhemmalla Debian Buster käyttöjärjestelmällä. (Debian, n.d.)

Raspberry Pi OS (64-bit) oli tätä kirjoittaessa beta-testausvaiheessa, eikä siitä ollut vielä julkaistu vakaata versiota Imagerilla asennettavaksi. (Forums.Raspberrypi, 2021) Raspberry

PI ohjelmistojohtaja Gordon Hollingworth on jakanut virallisella Raspberry Pi -foorumilla linkin, jonka kautta uusin versio 64-bittisestä beta-testausvaiheessa olevasta käyttöjärjestelmästä on ladattavissa. Käyttöjärjestelmän kehitystyön ollessa tätä kirjoittaessa vielä kesken, se rajattiin tämän työn käyttöjärjestelmävaihtoehtojen ulkopuolelle. (Downloads.Raspberrypi, 2021)

Imagerista löytyy myös muita käyttöjärjestelmiä erilaisiin tarkoituksiin, kuten Kodi OS, joka on tarkoitettu viihdekeskuskäyttöön, ja RetroPie, joka tekee RasPista retropelikoneen. Vaihtoehtoina on myös käyttöjärjestelmiä esimerkiksi 3D-tulostimelle, kotiautomaatiolle ja infotaululle. (Raspberrypi, n.d.-c)

Imagerissa on myös versiosta 1.6 lähtien lisäominaisuus, jonka avulla käyttöjärjestelmäkuvaan voidaan kustomoida erilaisia asetuksia, kuten SSH-yhteyden salliminen, salasanan lisääminen pi-käyttäjälle, vain julkisen avaimen käyttäminen yhteyden ottamiseen, sekä wifi- ja sijaintiasetuksia. Asetusvalikon saa esiin painamalla näppäinyhdistelmää Ctrl+Shift+X. (Raspberrypi, n.d.-b) Todettiin että tämä lisäominaisuus on tarkoitettu Raspberry Pi OS käyttöjärjestelmille, eikä sen avulla onnistuttu kokeilun tuloksena lisätä asetuksia Ubuntu-kuvaan.

Raspberry Pi Imagerissa on Ubuntu-käyttöjärjestelmävaihtoehtoina sekä 32- että 64-bittiset käyttöjärjestelmät kolmena eri versiona. Työpöytä- ja palvelinversioiden lisäksi asennettavina vaihtoehtoina ovat myös esimerkiksi erilaisiin robotti- Industry 4.0- ja IoT-käyttöön suunnitellut Ubuntu Core20-versiot. (Ubuntu, n.d.-f)

Ubuntu tarjoaa RasPille myös uutta Ubuntu 20.10 versiota. Ubuntun julkaisulistauksesta kuitenkin ilmenee, että tätä 14.10.2021 julkaistua Impish Indri -nimistä käyttöjärjestelmää tarjotaan tällä hetkellä vain heinäkuuhun 2022 asti ulottuvalla tuella, jolloin sen elinkaari tulee myös loppumaan. (Ubuntu, n.d.-e) Työn tavoitteena oli asentaa pitkäaikainen toimiva järjestelmä käyttäen apuna internetistä löytyviä lähteitä ja dokumentaatioita, joten Ubuntu Server 20.10 rajattiin edellä mainituista syistä käyttöjärjestelmävaihtoehtojen ulkopuolelle.

RasPille on tulevaisuudessa tarkoitus asentaa Ansiblea käyttäen myös Docker Engine, jonka dokumentaatiosta selviää minimivaatimukset käyttöjärjestelmälle. Sen mukaan Docker Engine voidaan asentaa neljälle erilaiselle 64-bittiselle Ubuntu-käyttöjärjestelmäversiolle.

Dokumentaation mukaan Docker Engine tukee projektissa käytettävän RasPin ARM64-prosessoriarkkitehtuuria. Tieto Docker Enginen vaatimuksesta käyttöjärjestelmän 64-bittisyydelle rajaa Raspberry Pin 32-bittisen käyttöjärjestelmän vaihtoehtojen ulkopuolelle. (Docker, n.d.)

Käyttöjärjestelmävaihtoehtojen tutkimisen jälkeen kokonaisuuteen parhaiten sopiva käyttöjärjestelmä on Ubuntu Server 20.04.3 LTS (64-bit). Valintaa tukee myös tieto siitä, että käyttöjärjestelmälle on luvattu pitkäaikainen tuki huhtikuuhun 2025 asti. (Taulukko 1)

Valintaa perustelee myös Ubuntun aktiivinen tuki RasPin Docker- ja Kubernetes -yhteensopivuudelle. Ubuntu on nostanut Raspberry Pi -sivullaan esiin myös Amazon Graviton2 yhteensopivuuden, jossa ajatuksena on Raspberry Pi:llä ARM64-prosessoripohjaiselle käyttöjärjestelmälle kehitetyn, ja konttiteknologiaa käyttävän sovelluksen nostaminen seuraavan sukupolven kustannustehokkaaseen ja suorituskykyiseen Graviton2 -pilveen. (Ubuntu, n.d.-d)

Ubuntu Server 20.04 -palvelinkäyttöjärjestelmät ovat kokemuksen mukaan käytössä hyvin laajasti myös kaikilla suurilla pilvipalveluntarjoajilla, joten luotavat Ansible-pelikirjat ovat sellaisenaan mahdollisimman yhteensopivia myös pilveen asennettavan vastaavan palvelimen kanssa. Ubuntu Server 20.04.3 LTS on avoimen lähdekoodin käyttöjärjestelmä ja se on asennettavissa microSD-kortille Raspberry Pi Imager -ohjelman käyttöliittymästä Ubuntu-valikon alta.

Taulukko 1 Palvelinkäyttöjärjestelmävertailu

Palvelinkäyttöjärjestelmä Raspberry Pi 4B -tietokoneelle	**Pitkäaikainen tuki**	**Vakaa / Beta**	**Julkaistu**	**Standardi tuki loppuu**	**Elinkaari loppuu**	**Docker Engine - yhteensopiva**
RasPi OS Lite (32-bit) Debian Bullseye ✗	kyllä	vakaa	14.08.2021 (Debian Bullseye)			Ei
RasPi OS Lite (32-bit) Debian Buster ✗	Ei	vakaa	06.07.2019 (Debian Buster)		noin. elokuu 2022 (Debian Buster)	Ei
Raspberry Pi OS Lite (64-bit) ✗	beta	beta	08.11.2021	beta	beta	beta
Ubuntu Server 21.10 (64-bit) Impish Indri ✗	Ei	vakaa	14.10.2021	heinäkuu 2022	heinäkuu 2022	kyllä
Ubuntu Server 20.04.3 LTS (64-bit) Focal Fossa ✓	kyllä	vakaa	26.08.2021	huhtikuu 2025	huhtikuu 2030	kyllä

2.3 Ubuntu Server 20.04.3 LTS käyttöjärjestelmä (64-bit)

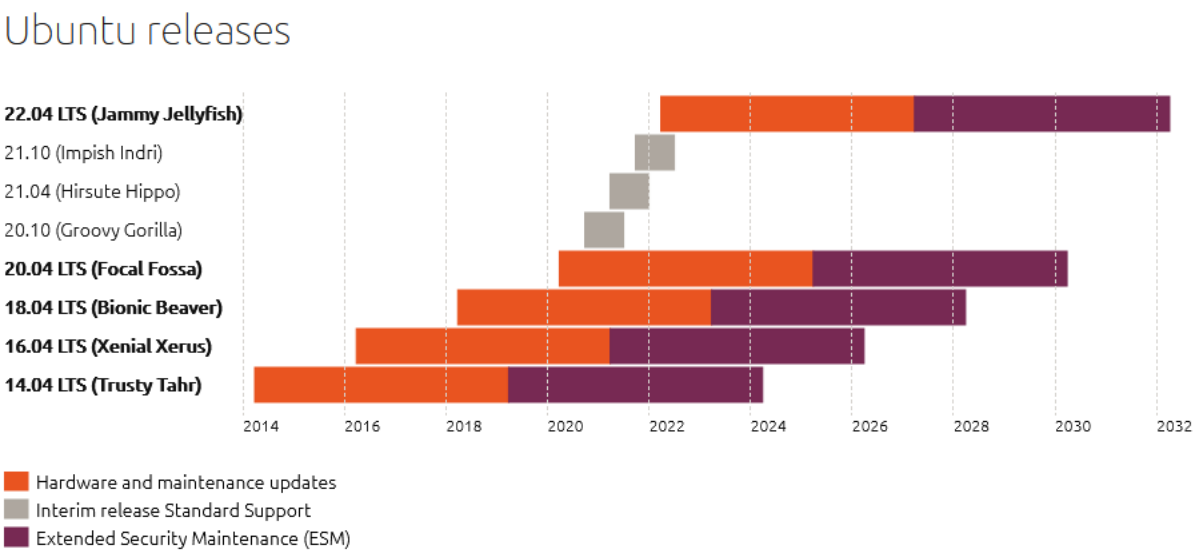
Ubuntu Server 20.04.3 LTS on Canonical Ltd:n kehittämä, Debian GNU/Linux -pohjainen avoimen lähdekoodin palvelinkäyttöjärjestelmä. Ubuntun mission mukaan sen kehittämät käyttöjärjestelmät ovat ilmaisia kaikille käyttäjilleen, ja niiden kehittäminen rahoitetaan tarjoamalla Canonicalin muita palveluja ammattilaisille, jotka käyttävät Ubuntu-käyttöjärjestelmiä isossa mittakaavassa. Ubuntu-palvelinkäyttöjärjestelmiä asennetaan hyvin paljon pilvipalveluntarjoajien alustoille, kuten AWS, Azure ja Google Cloud. (Ubuntu, n.d.-a)

Ubuntu-palvelinkäyttöjärjestelmä on nimensä mukaisesti tarkoitettu esimerkiksi tiedosto- web- ja postipalvelin käyttöön. Asennus ei sisällä graafista käyttöliittymää eikä mitään työpöytäversion mukana tulevia ohjelmia. Ubuntu-julkaisun palvelin- ja työpöytäversiot käyttävät molemmat samaa tiedostorakennetta, joten palvelinversio on päivitettävissä työpöytäversioksi ja työpöytäversio palvelinversioksi. Käyttöjärjestelmä voidaan asentaa neljälle erilaiselle 64-bittiselle prosessoriarkkitehtuurille, kuten Raspberry Pi 4:n ARM64:lle. Suositeltuja vaatimuksia järjestelmältä ovat vähintään 1 GHz prosessori, ja 1GB RAM-muistia, sekä minimissään 2.5GB vapaata tallennustilaa. (Ubuntu, n.d.-c)

2.3.1 Ubuntun LTS- ja välijulkaisut

Ubuntusta julkaistaan uusi pitkäaikaisesti tuettu LTS-versio aina kahden vuoden välein huhtikuussa (Kuva 3). Versionumero ilmaisee myös, että 20.04 LTS ensimmäinen versio on julkaistu huhtikuussa 2020. Seuraava 22.04 LTS-julkaisu on tulossa suunnitellusti huhtikuussa 2022. LTS-julkaisujen tuki on kaikille ilmaista ensimmäiset viisi vuotta, jolloin käyttöjärjestelmää tuetaan huolto- ja tietoturvapäivityksin. Viiden vuoden jälkeen käyttäjät, jotka ovat hankkineet Ubuntu Advantage -tilauksen sisältämän laajennetun tietoturvahuollon, tai maksimissaan kolmen koneen ilmaisen yksityiskäyttäjä-tilauksen, voivat vastaanottaa tietoturvapäivityksiä käyttöjärjestelmään vielä seuraavatkin viisi vuotta. Ensimmäisen viiden vuoden ilmaisen jakson lähestyessä loppuaan, Canonical suosittaa käyttäjiä päivittämään uudempaan LTS-versioon tai hankkimaan laajennettu tuki seuraavalle viiden vuoden jaksolle. Arviolta noin 95-prosenttia kaikista Ubuntu-asennuksista on LTS-versioita, joten ne ovat eniten käytettyjä Ubuntu-käyttöjärjestelmiä. LTS-versiot ovat yrityskäyttötasoisia Ubuntu-julkaisuja. (Ubuntu, n.d.-g)

Kuva 3 Ubuntu LTS- ja välijulkaisu- sekä päivitysaikataulu (Ubuntu, n.d.-g)



LTS-julkaisujen välillä julkaistaan puolen vuoden välein oma välijulkaisunsa, kuten Impish Indri-julkaisu (Kuva 3), joka tarjoaa Canonicalin kehittämät uusimmat ominaisuudet ja avoimeen lähdekoodiin perustuvat projektit. Välijulkaisuja tuetaan yhdeksän kuukautta ja ne ovat suosittuja kehittäjien keskuudessa. Julkaisemalla sekä LTS- että välijulkaisuja, Canonical mahdollistaa kehittäjäyhteisönsä, yritysten ja sovelluskehittäjien paremman suunnitelmallisen siirtymisen käyttämään uudempia yksilöllisiin tarkoitusperiinsä soveltuvia käyttöjärjestelmäversioita. (Ubuntu, n.d.-g)

2.3.2 Hardening eli käyttöjärjestelmän tietoturvan koventaminen

UNIX-käyttöjärjestelmään pohjautuvat GNU/Linux-käyttöjärjestelmät ovat pääsääntöisesti turvallisia, koska niiden käyttämät prosessit toimivat erillisinä ja käyttäjien käyttöoikeuksia järjestelmän sisällä rajoitetaan. Nämä GNU/Linux-jakelut eivät kuitenkaan ole lähtökohtaisesti täysin tietoturvallisia, koska niiden täytyy tasapainoilla ja tehdä kompromisseja käytettävyyden, suorituskyvyn, ja tietoturvallisuuden välillä, mikä tyypillisesti johtaa heikompaan tietoturvan tasoon. Siksi on tärkeää teknisten ohjeiden noudattamisen ohella tehdä tietoturvan tasoa parantava järjestelmän koventaminen. (Boelen, 2018)

Järjestelmän koventaminen tarkoittaa prosessia, jonka aikana tehdään tietoturvan tasoa parantavia toimia. (Boelen, 2018) Michael Boelen listaa kymmenen askelta järjestelmän kovettamiseen, jotka ovat:

1. Asenna tietoturvapäivitykset ja korjaukset
2. Käytä vahvoja salasanoja
3. Jos mahdollista, käytä prosesseja vain paikallisesti
4. Ota käyttöön palomuuri
5. Pidä järjestelmä puhtaana poistamalla käyttämättömät paketit ja käyttäjät kotikansioineen
6. Tee tarvittavat tietoturvakonfiguroinnit
7. Rajoita pääsyä järjestelmään
8. Monitoroi järjestelmää
9. Ota varmuuskopioita ja testaa niiden toiminta
10. Suorita järjestelmätarkastuksia säännöllisesti

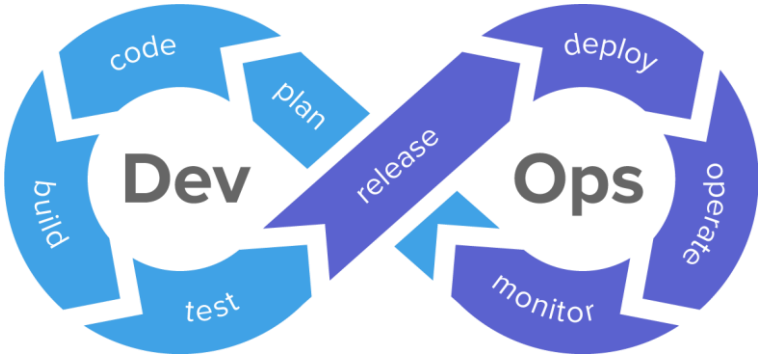
George Wilder on puolestaan laatinut Ubuntu-palvelinkäyttöjärjestelmän koventamiseen tähtääviä toimia Ansible-pelikirjan muotoon. Wilderin laatima ohje näyttää miten Mac-hallintakoneelle asennetulla Ansible-ohjelmalla automatisoidaan Ubuntu 20.04 tai sitä uudemman palvelimen ensimmäinen konfigurointi. (Wilder, 2021)

2.4 DevOps-viitekehys ja IaC-työkalujen hyödyt

Infrastructure as Code (IaC) on osa DevOps-viitekehyksen mukaista ohjelmistokehitystä, jossa kehitetyn ohjelman käyttämä IT-infrastruktuuri voidaan skripti suorittamalla asentaa erilaisiin ympäristöihin, kuten julkiseen tai paikalliseen pilveen. Esimerkiksi Terraform on tähän tarkoitukseen kehitetty ohjelma. Sen avulla IT-infrastruktuuri voidaan asentaa paikalliseen pilveen, sekä rinnakkain usean pilvipalvelutarjoajan alustoille mihin päin maailmaa tahansa. Toinen IaC-työkalu on Ansible, jonka toiminnallisuudet keskittyvät DevOps-viitekehyksessä esimerkiksi Terraformin avulla asennetun IT-infrastruktuurin automatisoituun konfiguroimiseen, ohjelmien asentamiseen ja järjestelmien päivittämiseen. (IBM, 2019)

IaC-työkalut sijoittuvat DevOps-kaavion deploy ja operate-osiin. (Kuva 4)

Kuva 4 DevOps-kaavio (Spoclearn, n.d.)



Käyttämällä IaC-työkaluja ohjelmistokehityksessä ja systeemityössä saavutetaan merkittäviä hyötyjä. Kehitys- testaus- ja tuotantoympäristöjä voidaan ottaa nopeasti käyttöön helposti ymmärrettävillä skripteillä, jolloin käyttöönotot tulevat samalla dokumentoiduiksi. Hyötynä on myös ympäristöjen vakioiminen, jolloin vältytään manuaalisesti konfiguroitujen ympäristöjen yhteensopivuus ja tietoturvaongelmilta. IaC-työkaluja käyttämällä voidaan myös aina tarvittaessa toistaa sama IT-infrastruktuuri samoilla konfiguroinneilla ja samoilla asennetuilla ohjelmilla. Yrityksen jatkuvuuden kannalta toiminta on paremmin turvattua tilanteissa, joissa yrityksen IT-infrastruktuurista vastaava henkilöstö vaihtuu, koska tietotaito on hyvin dokumentoitu ja muidenkin henkilöiden toistettavissa. Merkittävinä hyötyinä pidetään myös ajansäästöä toistuvien työvaiheiden tekemisestä, sekä mahdollisuudesta käyttää skaalautuvasti pilvipalveluntarjoajien kulutusperusteiseen laskutukseen perustuvaa laskentatehoa yrityksen todellisen ja vaihtelevan tarpeen mukaan. (IBM, 2019)

2.5 Ansible

Ansible on avoimen lähdekoodin projekti. Ohjelma on suunniteltu kaiken kokoisten IT-ympäristöjen helppokäyttöiseksi, tietoturvalliseksi ja luotettavaksi hallintatyökaluksi, jonka avulla voidaan automatisoidusti ajaa monimutkaisiakin päivityksiä IT-infrastruktuuriin ajamatta järjestelmiä alas. Ohjelmaa käytetään IT-infrastruktuurin automatisoituun konfiguroimiseen ja ohjelmien asennuksiin. Sen avulla voidaan myös tehdä määrityksiä

Ansible-yhteensopiville verkkolaitteille. Sen toiminta on agentitonta, mikä tarkoittaa, ettei etäkoneille asenneta erikseen ylläpidettävää taustaprosessia keskustelemaan Ansiblen kanssa. Näin vältytään mahdollisilta yhteysongelmilta hallinta- ja etäkoneen välillä. Ansible käyttää pääasiassa avoimen lähdekoodin OpenSSH-tunneliyhteyttä koneiden välillä, sekä helposti ymmärrettävää komentokieltä, joka ei vaadi käyttäjältään ohjelmointikielien ymmärtämistä. Red Hat ylläpitää Ansiblesta yhtä aikaa useita eri versioita ja niiden dokumentaatioita, ja julkaisee ohjelmasta uuden version noin kaksi kertaa vuodessa. Ansible-dokumentaatiota lukiessa on aina tarkistettava verkkosivun vasemmasta yläkulmasta dokumentaation versio. Valitsemalla versiovalikosta latest, varmistetaan dokumentaation olevan viimeisimmän julkaistun community-version mukainen. (Ansible, n.d.-l)

2.5.1 Ansiblen IBM ja Red Hat omistajuus

Ansible on amerikkalaisen teknologiayritys IBM:n heinäkuussa 2019 hankkiman Red Hat -ohjelmistoyhtiön sponsoroima avoimen lähdekoodin projekti. Koko Red Hat -yhtiön hankintahinnaksi ilmoitettiin 34 miljardia dollaria, joka on helmikuun 2022 arvossa noin 30 miljardia euroa. Kaupan yhteydessä ilmoitettiin IBM:n ja Red Hatin yhteistyön syventyvän kehittämään ja tarjoamaan asiakkailleen avoimeen lähdekoodiin pohjautuvaa seuraavan sukupolven monipilvistä hybridialustaa. Kaupasta annetun tiedotteen mukaan alusta mahdollistaa tietojen ja sovellusten turvallisen käyttöönoton ja hallinnan sekä paikallisilla koneilla, että yksityisissä ja useissa julkisissa pilvissä. Samalla IBM ilmoitti sitoutuvansa säilyttämään Red Hatin avoimen lähdekoodin perinnön itsenäisyyden ja neutraalisuuden. Osa mainittua Red Hat -perintöä on yhtiön tuotevalikoima, johon Ansible sisältyy yritysten paljon käyttämänä teknologiana. (RedHat, 2019)

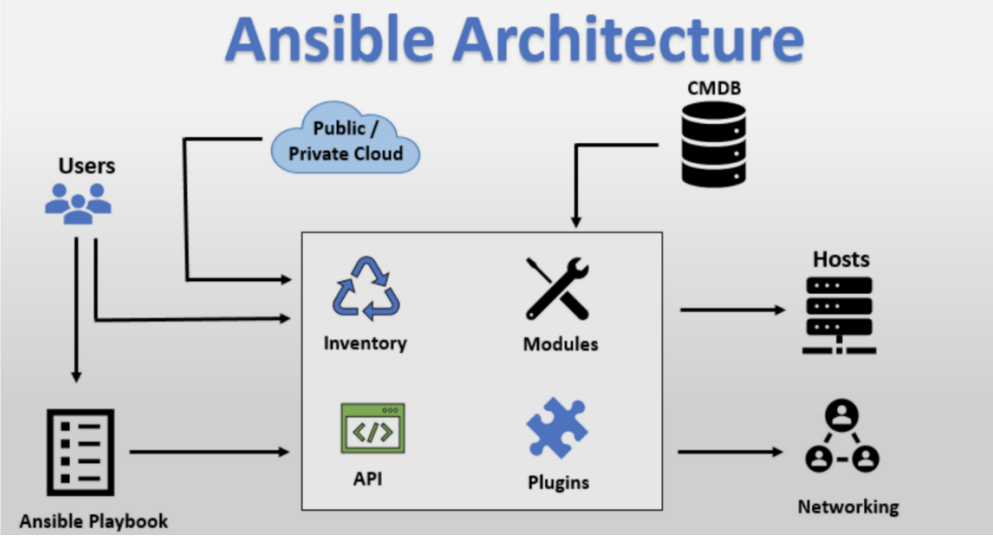
Red Hat on kehittänyt Ansible-projektin lähdekoodista myös Red Hat Ansible Automation Platform -nimisen maksullisen palvelun, jonka hinnoittelu perustuu hallittavien etäkoneiden, kuten järjestelmien, koneiden, pilvi-instanssien, virtuaalikoneiden, konttien ja laitteiden määrään. Palvelua tarjotaan kahtena ominaisuuksiltaan erilaisena versiona. Standard-paketti on tarkoitettu yrityksen IT-järjestelmien hallintaan arkipäiviin sijoittuvalla 8x5 tuella. Premium-paketti on puolestaan tarkoitettu vaativiin DevOps toimintoihin jatkuvalla 24x7 tuella. Molemmissa paketeissa on lisäksi mahdollista ostaa neljää eritasoista tuen

vasteaikaan perustuvaa palvelua. Tuotteiden tuen palveluselosteen mukaan aiemmin kolmantena vaihtoehtona ollutta itsepalvelupakettia web-perustuella ei ole enää saatavilla. (Ansible, n.d.-j)

2.5.2 Ansiblen arkkitehtuuri

Ohjelman arkkitehtuurikaaviossa (Kuva 5) on esitelty Ansiblen toiminnallisuuden kannalta tärkeimmät osat, ja taulukossa (Taulukko 2) on niiden lyhyt kuvaus.

Kuva 5 Ansible-arkkitehtuurikaavio (Dineshbaburam, 2020)



Taulukko 2 Ansible-arkkitehtuurikaavion osien kuvaus

Osa	**Kuvaus**
Users	Ansible-käyttäjät hallintakoneilla
Ansible Playbook	Pelikirja, joka sisältää pelejä, tehtäviä ja etäkoneilla suoritettavia moduuleja
Inventory	Ryhmitelty hosts-inventaariolista etäkoneista
Public / Private Cloud	Hallittava IT-infrastruktuuri. Inventaariotiedosto voi olla myös pilvessä.
API	Ansiblen ohjelmointirajapinta
CMDB	Ottaa vastaan Ansiblen keräämät faktat etäkoneista. Tietokanta sisältää järjestelmien konfigurointi-informaation. (Ansible-cmdb, n.d.)
Modules	Ohjelmakoodeja, jotka suorittavat tehtäviä etäkoneilla
Plugins	Ohjelmakoodeja, joita suoritetaan hallintakoneella lisäämään toiminnallisuuksia Ansiblen ydintoimintoihin
Hosts	Hallittavat etäkoneet
Networking	Hallittavat Ansible-yhteensopivat verkkolaitteet

2.5.3 Hallintakone ja etäkoneet

Ansible asennetaan Linux-pohjaiselle hallintakoneelle, ja se vaatii versiosta 2.12 lähtien toimiakseen hallintakoneelle asennetun Python 3.8 tai sitä uudemman version. Windows käyttöjärjestelmä ei sovellu hallintakoneeksi, mutta Ansible voidaan asentaa esimerkiksi Ubuntulle Windows 10 -koneen WSL2-ympäristöön. Paremman toimivuuden saavuttamiseksi Ansible-dokumentaatio suosittaa hallintakoneen sijoittamista samaan pilvi- tai paikalliseen ympäristöön etäkoneiden kanssa. (Ansible, n.d.-q)

Ansible käyttää SSH-yhteyttä koneiden välillä ja lähettää moduulit oletuksena SFTP-protokollalla. Mikäli joillakin etäkoneilla ei ole SFTP käytettävissä, voidaan ansible.cfg konfigurointitiedostossa ottaa vaihtoehtoisesti SCP käyttöön. Hallittaville etäkoneille ei ole tarpeen asentaa erillisiä ohjelmia Ansiblea varten, riittää kun etäkoneilla on Python 2.6 tai uudempi, tai Python 3.5 tai uudempi asennettuna. (Ansible, n.d.-q)

2.5.4 Hosts-inventaariolista

Ansible käyttää hosts-inventaariolistaa tai listoja useiden palvelimien, työasematietokoneiden ja verkkolaitteiden, kuten reitittimien ja kytkimien automatisoituun etähallintaan valitun IT-infrastruktuurin sisällä. Hosts-tiedostossa hallittavat palvelimet ovat ryhmiteltynä esimerkiksi web-palvelin- ja tietokantapalvelinryhmiinsä, perustuen niiden yksilölliseen käyttötarkoitukseen. Oletuksena tämä hosts-tiedosto on hallintakoneella sijainnissa /etc/ansible/, mutta inventaariotiedosto voidaan sijoittaa myös pilveen. Jos hosts-tiedosto on muussa kuin oletussijainnissa, tulee käyttää i-parametria osoittamaan polkuun. Ansible-dokumentaation mukaan hosts-inventaariotiedostot ovat useimmiten ihmiselle helposti ymmärrettävässä ja luettavassa YAML- ja INI-tiedostomuodoissa. (Ansible, n.d.-p)

DigitalOcean-kirjoittaja Erika Heidin mukaan on suositeltavaa käyttää projektikohtaisia inventaariolistoja oletustiedoston sijaan, jotta vältyttäisiin vahingossa ajamasta asetuksia väärille palvelimille. Näin toimiessa tulee muistaa käyttää -i -parametriä osoittamaan inventaariotiedoston sijaintiin, Ansible-komentoja ajettaessa. (Heidi, 2020)

2.5.5 Ansible Playbook

Ansible Playbookissa, suomalaisittain ilmaistuna pelikirjassa, määritetään tehtävien suoritusjärjestys etäkoneilla. Se on suunniteltu uudelleenajettavaksi, sillä tehtäviä, jotka ovat etäkoneella jo pelikirjassa määritetyllä tasolla, ei suoriteta uudelleen. Kun pelikirjaa halutaan käyttää esimerkiksi jonkin ohjelman päivittämiseen viimeisimpään julkaistuun versioon, voidaan sitä käyttää sellaisenaan yhä uudelleen. Pelikirja (playbook) koostuu yhdestä tai useammasta pelistä (play), ja pelien sisältämät tehtävät (tasks) suorittavat yksittäisiä tai useampia moduuleja valituilla etäkoneilla. Pelikirjan pelit ja tehtävät suoritetaan vaiheittain ylhäältä alas järjestyksessä. Pelikirjassa olevan yksittäisen pelin pitää vähintäänkin määrittää ne etäkoneet, joita peli koskee, ja yksi suoritettava tehtävä. Pelikirjan syntaksi on helposti ymmärrettävässä YAML-muodossa. (Ansible, n.d.-s)

2.5.6 Ansiblen kokoelmat, moduulit ja liitännäiset

Moduulit ovat tiettyyn tehtävään suunniteltuja ohjelmakoodeja, joita kutsutaan ad-hoc -komentoina ja pelikirjojen tehtävinä suorittamaan haluttuja tehtäviä etäkoneilla, kuten konfigurointeja ja ohjelmien asennuksia. Moduulit lähetetään SSH-yhteyden yli, ja suoritettuaan tehtävänsä ne poistetaan etäkoneelta. Moduuli voi olla Python-kielinen, PowerShell-skripti, tai millä tahansa kielellä kirjoitettua koodia, jolla on tietty tehtävä. (Red Hat Ansible, n.d.)

Liitännäiset ovat hallintakoneella suoritettavia Ansiblen ydinominaisuuksia laajentavia toimintoja, joiden tehtävänä on esimerkiksi kytkeytyä inventaariotiedostoihin paikallisesti ja pilvessä. Liitännäiset ovat Python-kielisiä. (Ansible, n.d.-k)

Yksinkertaisin esimerkki moduulista on Ansibleen sisäänrakennettuun ansible.builtin-kokoelmaan sisältyvä ping-moduuli. Sitä käytetään pelikirjojen ulkopuolella ad-hoc -komentona, kun halutaan testata hallintakoneen Ansible-käyttäjän kyky kirjautua hallittaville ei-Windows -etäkoneille ja todeta asennettu ja yhteensopiva Python-versio. Ping-moduulin onnistuessa tehtävässään, se palauttaa tuloksen pong. Windows koneille yhteystestiä tehdessä käytetään ansible.windows-kokoelmaan sisältyvää erillistä ja vastaavaa ping-moduulia. Mikäli halutaan testata kirjautuminen Ansible-yhteensopiviin verkkolaitteisiin,

käytetään vastaavasti ansible.netcommon -kokoelmaan kuuluvaa ping-moduulia. (Ansible, n.d.-m)

Ansible 2.10 versiosta lähtien moduulit ja liitännäiset sisältyvät kokoelmiin. Dokumentaatio kokoelmista, moduuleista ja liitännäisistä, sekä niissä käytössä olevista parametreista on kattava ja sisältää havainnollisia esimerkkejä. (Ohjelmakoodi 1) on Ansible-dokumentaatiosta löytyvä esimerkki, jossa käytetään ansible.posix -kokoelmaan kuuluvaa authorized_key -liitännäistä kopioimaan hallintakoneen käyttäjän julkinen avain etäkoneen ubuntu-käyttäjän authorized_key -tiedostoon. (Ansible, n.d.-i)

Ohjelmakoodi 1 Esimerkki Authorized_key -liitännäisen käytöstä (Ansible, n.d.-i)

```
- name: Set authorized key for user ubuntu copying it from current user
  ansible.posix.authorized_key:
    user: ubuntu
    state: present
    key: "{{ lookup('file', lookup('env','HOME') + '/.ssh/id_rsa.pub') }}"
```

Ansible-community version mukana tulevia moduuleja ja niiden dokumentaatioita on mahdollista tarkastella myös komentokehotteessa käyttämällä ansible-doc -komentoja. Esimerkiksi Komento 1 komentoja soveltaen voidaan etsiä Ansible-community versioon sisältyviä muitakin kokoelmia, moduuleja, sekä liitännäisiä. Ohjelmaan voidaan myös asentaa lisää kokoelmia esimerkiksi Ansible-galaxy -komentokehotetyökaluja apuna käyttäen. Kaikki ohjelman sisäiset ad-hoc -komennot ovat listattuna Ansible-dokumentaatio -verkkosivuilla. (Ansible, n.d.-u)

Komento 1 Komennot authorized_key liitännäisen etsimiseen ja tarkasteluun

```
$ ansible-doc -l | grep ansible.posix.authorized_key
$ ansible-doc ansible.posix.authorized_key
```

2.5.7 Ansible Vault

Ansible Vault on tarkoitettu arkaluontoisten tietojen, kuten etäkoneiden käyttäjätunnusten salasanojen ja SSH-avaimien salaamiseen, jotta ne eivät olisi selväkielisinä luettavissa inventaariotiedostossa ja pelikirjoista. Salaus voidaan tehdä yksittäisille muuttujille tai

kokonaisille tiedostoille käyttämällä ansible-vault -komentokehotetyökalua. Työkalua käyttäen voidaan esimerkiksi luoda uusi salasanasuojattu ja salattu tiedosto, avata tiedoston sisältö editoriin muokattavaksi, antaa tiedostolle uusi salasana ja purkaa tiedoston salaus. Kun etäkoneiden salasananoja sisältävä pelikirja ajetaan komentokehotteessa, voidaan komennon yhteydessä määrittää Ansible kysymään tämän salatun tiedoston salasana, jonka syöttämisen jälkeen tiedot ovat pelikirjan ajon käytettävissä. (Ansible, n.d.-o)

Ansiblen sisällä toimiva Vault-salasana voidaan tallentaa tiedostoon johon Ansiblella on käyttöoikeudet. Salasana voidaan myös tallentaa kolmannen osapuolen salasanatyökaluun, ja luoda skripti sen noutamiseksi. Yksinkertaisin tapa on opetella salasana ulkoa, tai kopioida ja liittää se manuaalisesti pelikirjan suorittamiskomentoon mistä valitusta lähteestä tahansa. Käsiteltäessä salattuja tiedostoja tulisi myös huomioida käytetyn editorin sisäiset tietoturvaan vaikuttavat asetukset. Ansible Vault -dokumentaation mukaan salasanoja ei tulisi testauskäyttöä lukuun ottamatta antaa selkokielisenä komentojen yhteydessä, koska ne jäävät luettavassa muodossa shell-historiaan. Käyttämällä kehotetta kysyä vault-salasana pelikirjan ajon yhteydessä, salasana ei tallennu shell-historiaan. (Ansible, n.d.-o)

3 Kehittämistyön tavoite ja tarkoitus

Työn tavoitteena on automatisoida Raspberry Pi 4 B -minitietokoneelle asennetun, yleisesti myös pilvipalveluntarjoajien alustoilla käytössä olevan Linux-palvelinkäyttöjärjestelmän yhteys- ja käyttäjäasetusten sekä järjestelmän koventamiseen tähtäävät peruskonfiguroinnit.

Tavoitteena on myös dokumentoida nämä tehdyt konfiguroinnit ja asennukset Ansiblen pelikirjoiksi, joita voidaan käyttää uudelleen tulevissa palvelinkäyttöjärjestelmien käyttöönotoissa, ja näin välttää inhimillisiä virheitä. Luodut pelikirjat lisätään työn liitteiksi, ja pelikirjojen tehtäviin merkitään lähdeviittaukset käytettyihin ohjeisiin ja dokumentaatioihin, jotta ne olisivat jatkossa helposti löydettävissä ja muokattavissa.

Työssä luodut pelikirjat, inventaariotiedosto, ja vault-tiedosto ovat perusesimerkki yhden etäkoneen hallintaan, mutta ne ovat muokattavissa ja laajennettavissa erilaisiin konfigurointitarpeisiin.

Työ on toiminnallinen kehitysprojekti, jonka teoriaosuudessa kuvataan työssä käytetyt laitteet, ohjelmistot, ja teoreettiset käytännöt. Työn käytännönosuudessa sovelletaan tätä tietoa, internetistä löytyviä virallisia dokumentaatioita, sekä toteutettujen projektien ohjeita.

Asennukset tehdään testaten ensin niiden toimivuus pitäen samalla päiväkirjaa. Lopuksi toteutetut toimivat ratkaisut ja työn vaiheet dokumentoidaan päiväkirjaa apuna käyttäen työn käytännön osuuteen. Liitteen 2 pelikirjan toteutuksessa sovelletaan Sarav AK:n luomaa pelikirjaa yhdessä Ansible dokumentaation kanssa. Liitteen 3 -pelikirjan toteutuksessa sovelletaan puolestaan George Wilderin luomaa pelikirjaa yhdessä Ansible dokumentaation ja Michael Boelen järjestelmän koventamiseen tähtäävän listan kanssa.

Kehitysprojekti on oman tarpeen lisäksi tarkoitettu muillekin systeemityöhön suuntautuneille opiskelijoille yhdeksi tavaksi hyödyntää ja omaksua työelämässä käytössä olevien työkalujen yhteiskäyttöä, ja oppia hallitsemaan etäkoneita esimerkkitasolla automatisointia apuna käyttäen.

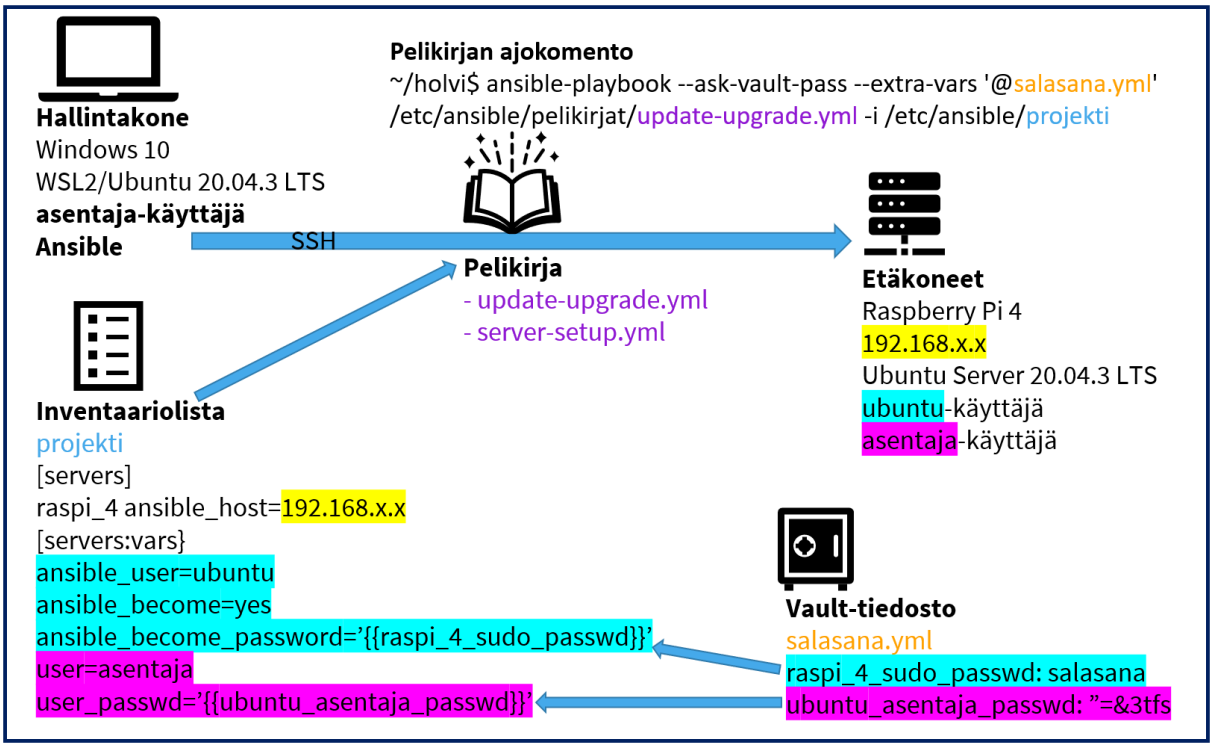
4 Kehittämistyön toteutus

Ensin listataan työssä käytetty laitteisto käyttöjärjestelmät ja ohjelmat, jonka jälkeen kuvataan Ubuntu-palvelinkäyttöjärjestelmän asentaminen Raspberry Pi 4:lle ilman hiiren, näppäimistön tai näytön kytkemistä RasPiin. Sitten käyttöjärjestelmään tehdään ensimmäinen järjestelmä- ja tietoturvapäivitys, jonka jälkeen se varmuuskopioidaan Windows 10 -koneelle.

Seuraavaksi asennetaan Ansible Windows 10 -koneen WSL2 ympäristöön Ubuntu 20.04:lle, joka toimii Ansible-hallintakoneena. Asennus tarkistetaan ja siirrytään luomaan Raspberry Pi 4 -etäkoneen IP-osoitteen sisältävää inventaariolistaa Ansiblen käyttöön. Hallintakoneelle luodaan uusi asentaja-niminen käyttäjä, ja käyttäjälle SSH-yhteys Raspiin, jonka jälkeen yhteys tarkistetaan Ansiblen omaa työkalua käyttäen. Ansiblea tullaan ajamaan kirjautumalla hallintakoneelle asentaja-käyttäjänä.

Käyttöön otetaan Ansible Vault -salasanatiedosto. Edellä luotuun inventaariotiedostoon lisätään yhteysmuuttujat RasPi-käyttäjille, joiden salasanatieto tullaan hakemaan Vault-tiedostosta Ansible-pelikirjojen ajon yhteydessä (Kuva 6). Ansible käyttää oletuksena vi-editoria, ja se vaihdetaan tässä yhteydessä nano-editoriin.

Kuva 6 Ansible-harjoitusympäristön kuvaus



Seuraavaksi luodaan ensimmäinen pelikirja nimeltä update-upgrade.yml, jota ajamalla testataan inventaariolistassa olevien yhteysmuuttujien kyky noutaa Vault-salasanatiedostossa olevat salasanat, ja käyttää niitä Raspberry Pi 4:lle kirjautumiseen sudo-käyttäjänä suorittamaan pelikirjan sisältämät järjestelmän päivittämisen tehtävät.

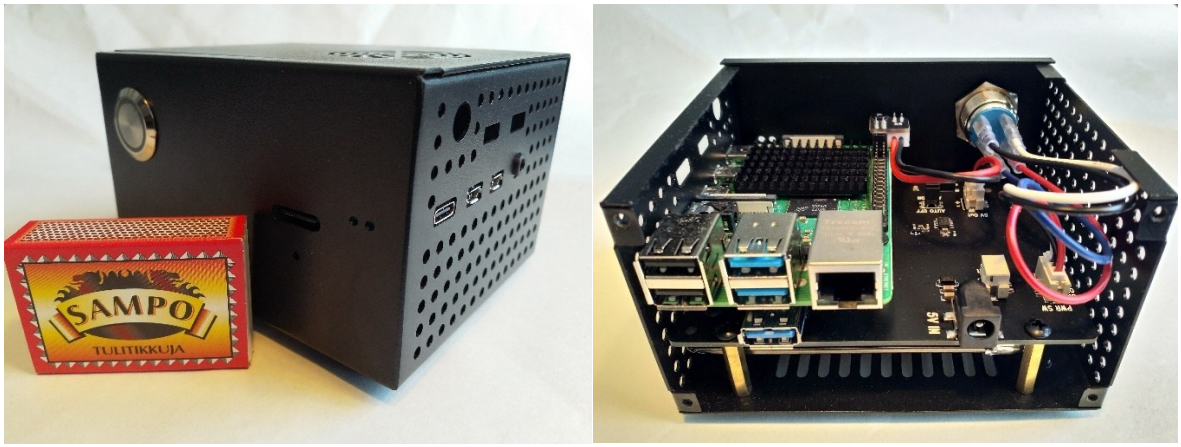
Sitten luodaan toinen pelikirja nimeltä server-setup.yml, jota ajamalla myös Raspberry Pi 4:lle lisätään uusi asentaja-niminen sudo-käyttäjä, jonka nimi- ja salasanatiedot tuodaan yhteysmuuttujia käyttäen. Hallinta- ja etäkoneiden asentaja-käyttäjien välille luodaan SSH-yhteys kopioimalla käyttäjälle aiemmin luotu julkinen avain. SSH-yhteyttä kovennetaan sallimalla yhteys vain avainparia käyttäen. Myös palomuuri otetaan käyttöön sallien vain SSH-yhteydet. Pelikirjan tehtäviin lisätään myös järjestelmän päivittämisen, uudelleenkäynnistyksen tarpeen tarkastamisen, ja uudelleenkäynnistämisen toiminnot. Lopuksi tarkistetaan pelikirjalla ajetut konfiguroinnit Raspberry Pi 4:llä.

4.1 Asennukset hallinta- ja etäkoneelle

4.1.1 Työssä käytetyt laitteet, käyttöjärjestelmät, ja ohjelmat

- Windows 10, versio 21H2 -tietokone
  - WSL2 / Ubuntu 20.04.3 LTS **(Hallintakone)**
  - Windows Terminal
  - Ansible 5.2.0 [core 2.12.1]
- Raspberry Pi 4 B -minitietokone 8GB RAM
  - Ubuntu Server 20.04.3 LTS **(Etäkone)**
  - 5V/4.0A -virtalähde (DC-liitin 5. 5 mm/2. 5 mm)
  - Geekworm X825 -metallikotelo virtapainikkeella Raspberry Pi 4:lle (Kuva 7)
  - Geekworm X825 2.5” Sata HDD/SSD lisäosa Raspberry Pi 4:lle (Kuva 7)
  - 32 GB microSD-kortti
  - Ethernet-kaapeli
- Huawei WiFi AX3 -reititin
- SD-kortin lukija
  - Raspberry Pi Imager v1.6.2
  - Win 32 Disk Imager

Kuva 7 Raspberry Pi 4 Geekworm-kotelossa



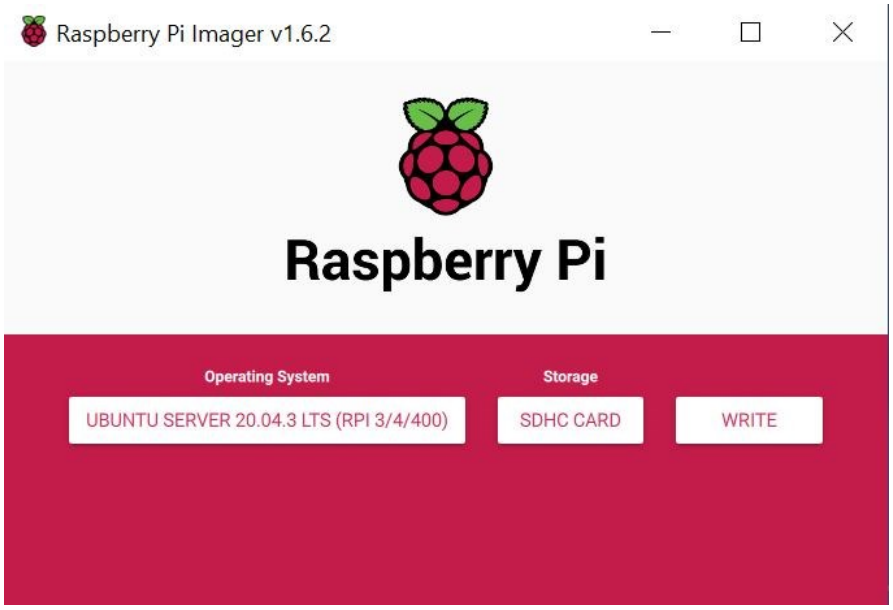
4.1.2 Ubuntu Server 20.04.3 LTS (64-bit) asennus etäkoneelle

Valittu Ubuntu-palvelinkäyttöjärjestelmä asennettiin microSD-kortille käyttäen Raspberry Pi Imager v1.6.2-ohjelmaa.

Käyttöjärjestelmän kirjoittaminen Windows-tietokoneeseen kytkettyyn microSD-korttiin aloitettiin painamalla ensin CHOOSE OS -painiketta. Avautuvasta valikosta valittiin Other general purpose OS, ja sitten Ubuntu valinnan takaa Ubuntu Server 20.04.3 LTS (64-bit) -käyttöjärjestelmä.

Seuraavaksi valittiin CHOOSE STORAGE -painikkeen takaa Windows-tietokoneeseen kytketty microSD-kortti. Kun molemmat valinnat olivat kuvan (Kuva 8) mukaiset, painettiin WRITE-painiketta, ja käyttöjärjestelmä alkoi kirjoittumaan microSD-kortille.

Kuva 8 Raspberry Pi Imager -valinnat.



Kun kirjoittaminen oli valmis, kytkettiin microSD-kortti Raspberry Pi 4 -tietokoneeseen sille varattuun korttipaikkaan. Seuraavaksi kytkettiin internet-yhteyden mahdollistava Ethernet-kaapeli. Todettiin että ennen virtajohdon kytkemistä RasPiin, on hyvä avata ensin esiin reitittimen hallintapaneelin laitehallinta-välilehti, jotta RasPin IP-osoite olisi heti

käytettävissä seuraavassa vaiheessa. Se tapahtuu yleisesti avaamalla reitittimen osoite 192.168.x.x selaimessa ja antamalla kirjautumistiedot.

Kun Ethernet-kaapeli ja virtajohto oli kytketty RasPiin, se käynnistyi ja kytkeytyi automaattisesti reitittimeen. Työssä käytetyn reitittimen päälle kytketty DHCP-palvelin jakoi RasPille oman IP-osoitteen. Ensimmäisen käynnistyminen kestää yleensä noin kaksi minuuttia, jona aikana Ubuntun cloud-init konfiguroi järjestelmän käyttäen kirjautumiseen tunnusta sekä salasanaa ”ubuntu”.

Ubuntu-asennusohjeen mukaan konfiguroinnin on mentävä keskeytyksettä loppuun asti, ja mikäli tässä vaiheessa ilmenee ongelmia, tulee laite käynnistää uudelleen. (Ubuntu, n.d.-b) Kun cloud-init oli valmis, näkyi RasPi reitittimen laitehallinnassa nimellä Ubuntu. Tämä Ubuntun IP-osoite lisättiin reitittimellä staattisten IP-osoitteiden luetteloon (Kuva 9), jolloin sen IP-osoite pysyy myös jatkossa samana.

Kuva 9 Reitittimen staattisten IP-osoitteiden luettelo

Ubuntun IP-osoite otettiin talteen reitittimeltä ja käytettiin sitä ensimmäisen SSH-yhteyden luomiseen avaamalla ensin Windows-Terminal -ohjelma pääkäyttäjänä, ja sitten antamalla PowerShell-komentokehotteessa komento Komento 2. Käyttäjältä kysyttiin seuraavaksi varmistusta yhteyden luomiseen, johon vastattiin kirjoittamalla ”yes”. Tämän jälkeen annettiin oletussalasana ”ubuntu” jonka jälkeen järjestelmä pyysi käyttäjää vaihtamaan salasanan turvalliseksi. Seuraavaksi otettiin uusi SSH-yhteys RasPiin, annettiin salasana, ja ajettiin tarvittavat komennot Komento 3 root-salasanan asettamiseksi ja käyttöjärjestelmän päivittämiseksi.

Komento 2 Komento SSH-yhteyden ottamiseen RasPille

```
$ ssh ubuntu@192.168.x.x
```

Komento 3 Käyttöjärjestelmän päivitys- ja root-salasanan asetuskomennot

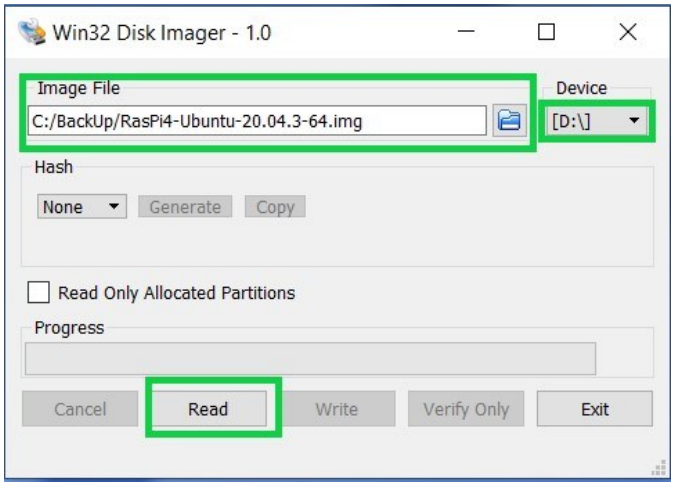
```
$ sudo passwd root
$ sudo apt update && sudo apt upgrade -y
```

Päivitysten asentuminen kesti ensimmäisellä kerralla melko kauan. Kun päivitys oli valmis, sammutettiin RasPi komennolla Komento 4. Seuraavaksi microSD-kortti siirrettiin RasPista takaisin Windows 10 -koneelle, ja siitä otettiin varmuuskopio C-asemalle käyttämällä Win32Disk-Imager ohjelmaa. (Kuva 10) Varmuuskopion ottamisen jälkeen microSD-kortti kytkettiin takaisin RasPiin.

Komento 4 Välitön sammutuskomento

```
$ sudo shutdown -h now
```

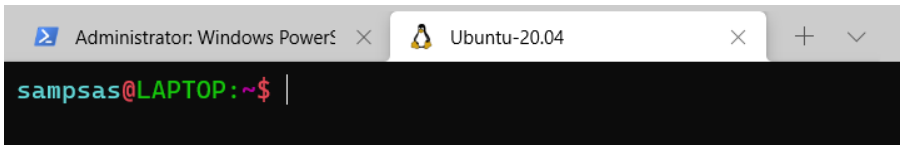
Kuva 10 Win32Disk-Imagerin valinnat microSD-kortin kopioimiseksi C-asemalle



4.1.3 Ansiblen asennus hallintakoneelle

Lähtötilanteessa Windows 10 -koneelle on asennettu Windows Terminaali, sekä WSL2. WSL2-käyttöjärjestelmäksi oli asennettu Ubuntu 20.04, joka toimii työssä Ansible-hallintakoneena. Ansiblen asennus aloitettiin avaamalla Windows Terminaali järjestelmänvalvojana ja avaamalla Ubuntu-20.04 uuteen välilehteen PowerShell-välilehden rinnalle. (Kuva 11)

Kuva 11 PowerShell ja Ubuntu-20.04 avoinna Windows Terminalissa



Ansible asennettiin antamalla Ansible-dokumentaation mukaiset komennot Komento 5. (Ansible, n.d.-r)

Seuraavaksi tarkastettiin Ansiblen versionumero antamalla komento Komento 6, joka listasi Ansiblen ja Pythonin versionumeroiden lisäksi tietoa moduulien ja konfigurointitiedostojen sijainnista.

Lopuksi annettiin Komento 7, joka tuotti lisäinformaationa asennetun Ansiblen versionumeron olevan 5.2.0. Tämä ilmaisi, että asennettuna oli uusin vakaa Ansible-community -versio. Ansible-community GitHub-dokumentaation mukaan Ansible 5.2.0 sisältää Ansible-core version 2.12.1, mikä selittää näiden kahden komennon tuloksena saadut erilaiset versionumerot. (Ansible-community, 2022)

Komento 5 Ansiblen asennuskomennot Ubuntu-käyttöjärjestelmään.

```
$ sudo apt update
$ sudo apt install software-properties-common
$ sudo add-apt-repository --yes --update ppa:ansible/ansible
$ sudo apt install ansible
```

Komento 6 Ansiblen version tarkistuskomento

```
sampsas@LAPTOP:~$ ansible --version
ansible [core 2.12.1]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/sampsas/.ansible/plugins/modules', '
/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/sampsas/.ansible/collections:/usr/share/
ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.8.10 (default, Nov 26 2021, 20:14:08) [GCC 9.3.0]
  jinja version = 2.10.1
  libyaml = True
```

Komento 7 Ansiblen asennuksen listauskomento

```
sampsas@LAPTOP:~$ apt list installed ansible -a
Listing... Done
ansible/focal,now 5.2.0-1ppa~focal all [installed]
```

Ansible-dokumentaation Ubuntu-asennusohjeen mukaan suoritettu asennus lisäsi järjestelmään uuden ppa.launchpad.net -lähteen (Personal Package Archive), jonka kautta Ansiblen-asennusta jatkossa päivitetään normaalin järjestelmäpäivityksen yhteydessä. Lisätty lähde löytyi hallintakoneelta komennoilla Komento 8.

Komento 8 Komennot Ansiblen asennuslähteen paikallistamiseen

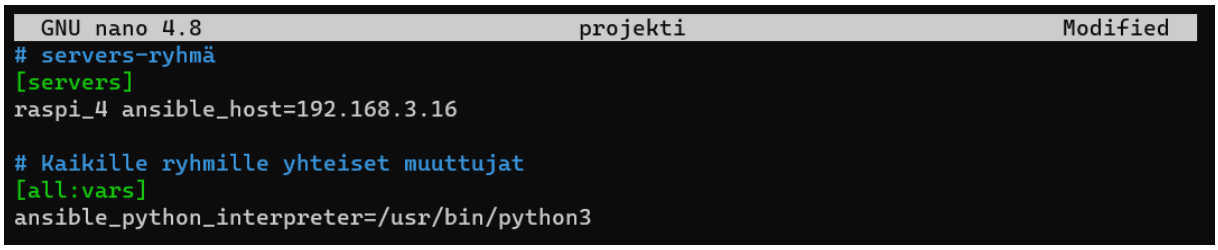
```
sampsas@LAPTOP:~$ ls -al /etc/apt/sources.list.d/
total 12
drwxr-xr-x 2 root root 4096 Jan 30 16:07 .
drwxr-xr-x 7 root root 4096 Jan 30 16:07 ..
-rw-r--r-- 1 root root  132 Jan 30 16:07 ansible-ubuntu-ansible-focal.list
sampsas@LAPTOP:~$ cat /etc/apt/sources.list.d/ansible-ubuntu-ansible-focal.list
deb http://ppa.launchpad.net/ansible/ansible/ubuntu focal main
# deb-src http://ppa.launchpad.net/ansible/ansible/ubuntu focal main
```

4.2 Ansiblen valmisteleminen pelikirjojen ajoa varten

4.2.1 Ansiblen inventaariotiedoston luominen

Seuraavaksi luotiin uusi inventaariotiedosto nimeltä projekti ja kopioitiin siihen /etc/ansible/hosts -oletusinventaariotiedoston sisältö. Projektitiedosto avattiin nanoeditoriin polusta /etc/ansible/projekti ja lisättiin uusi servers-ryhmä ja sen alle Ansiblen tarvitsema RasPi-etäpalvelimen IP-osoite. All:vars -alaryhmään lisättiin asetus, joka määrittää kaikki projektitiedostossa listatut etäpalvelimet käyttämään vain python3:a asioidessaan Ansible-moduulin kanssa. (Kuva 12)

Kuva 12 Ansiblen projekti-inventaariotiedosto

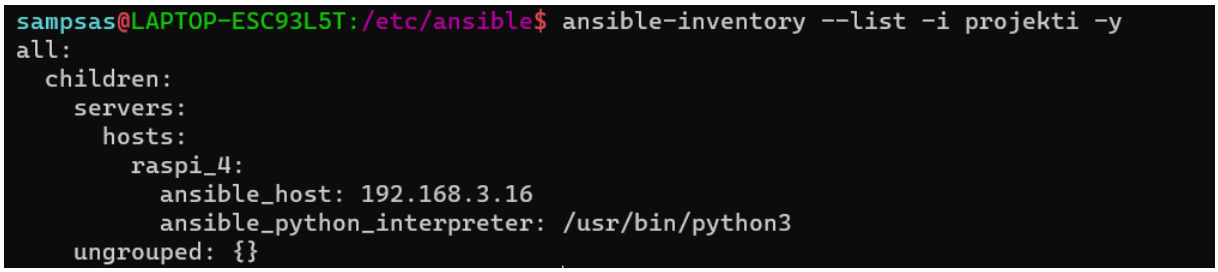


Projekti-tiedosto tallennettiin painamalla Ctrl+O, Enter, ja poistuttiin tiedostosta painamalla Crtl+X. Tiedoston sisältöä listatessa komentokehoteikkunaan, muistettiin lisätä komentoon -i parametri inventaariolistan sijaintiin. Komentoa ajettaessa oltiin polussa /etc/ansible/, joten polku voitiin jättää pois komennosta. Listaus YAML-muodossa (Kuva 13) saatiin antamalla Ansible ad-hoc -komento Komento 9.

Komento 9 Inventaariotiedoston listaus YAML-muodossa

```
$ ansible-inventory --list -i projekti -y
```

Kuva 13 Inventaariotiedosto listattuna YAML-muodossa



4.2.2 Ansible-käyttäjän lisääminen hallintakoneelle ja yhteystesti RasPiin

Hallintakoneelle luotiin Ansiblea varten uusi käyttäjä nimeltä asentaja, ja tämä käyttäjä lisättiin sudo-ryhmään. Sitten kirjauduttiin hallintakoneelle asentajakäyttäjänä, siirryttiin kotikansioon, ja tarkistettiin polku. Nämä tehtiin komennoilla Komento 10.

Komento 10 Uuden käyttäjän luominen ja liittäminen sudo-ryhmään

```
$ sudo adduser asentaja
$ sudo usermod -a -G sudo asentaja
$ su asentaja
$ cd
$ pwd
```

Ennen kuin Ansible-yhteystesti voisi onnistua RasPiin, oli ensin luotava WSL2-hallintakoneella asentajakäyttäjälle SSH-avainpari, jonka jälkeen käyttäjän julkinen avain lähetettiin RasPin Ubuntu-käyttäjälle. RSA-avainparin salauksen tasoksi valittiin 4096-bittiä. Kun SSH-yhteys oli kunnossa, voitiin ajaa Ansible-yhteystesti hallintakoneelta RasPiin. Edellä mainitut komennot tehtiin komennoilla Komento 11, jossa -m tarkoittaa Ansible-moduulia, -u tarkoittaa RasPi-käyttäjää, ja -K määrittää Ansiblea kysymään RasPin pääkäyttäjän salasanaa ennen toiminnon suorittamista. Salasanan ja kysytyn tunnuslauseen syöttämisen jälkeen yhteystesti onnistui ja antoi kuvan (Kuva 14) mukaisen palautteen.

Komento 11 Komennot SSH-yhteyden luomiseen ja Ansible-yhteystestille

```
$ ssh-keygen -b 4096
      Hyväksyttiin polku painamalla Enter
      asetettiin yhteydelle tunnuslause
      asetettiin yhteydelle tunnuslause uudelleen
$ ssh-copy-id ubuntu@192.168.x.x
      Hyväksyttiin varoitus syöttämällä yes
$ ansible -i /etc/ansible/projekti all -m ping -u ubuntu -K
      syötettiin ubuntu-käyttäjän salasana
      syötettiin yhteyden tunnuslause
```

Kuva 14 Informaatio onnistuneesta Ansible-yhteystestistä

```
asentaja@LAPTOP:~$ ansible -i /etc/ansible/projekti all -m ping -u ubuntu -K
BECOME password:
Enter passphrase for key '/home/asentaja/.ssh/id_rsa':
raspi_4 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

4.2.3 Ansible Vault -salasanatiedoston luominen

Ansible-vault käyttää oletuksena vi-editoria. Vaihdettiin komennoilla Komento 12 oletukseksi käyttöön nano-editori. Tämä tehtiin lisäämällä bashrc-tiedoston loppuun rivi: export EDITOR=nano . (Ellingwood, 2022)

Komento 12 Komennot oletuseditorin vaihtamiseksi nano-editoriin

```
asentaja@LAPTOP:~$ nano ~/.bashrc
asentaja@LAPTOP:~$ . ~/.bashrc
asentaja@LAPTOP:~$ echo $EDITOR
nano
```

Seuraavaksi luotiin käyttäjän kotikansioon uusi kansio, johon salasanasuojattu ja salattu vault-salasanatiedosto tallennettiin. Ubuntu-käyttäjän salasana lisättiin tiedostoon selkokielisenä. Myöhemmin pelikirjan tehtävässä RasPille lisättävälle asentaja-käyttäjälle luotiin tiedostoon satunnainen näppäimistöllä kirjoitettava vahva salasana. Satunnaisen salasanan luomiseen käytettiin siihen tarkoitettua Ubuntu-käyttöjärjestelmään sisältyvää openssl-työkalua toisella avoimella Windows Terminalin Ubuntu-välilehdellä. Lopuksi tarkastettiin salasanatiedosto ja mahdollisuus tarvittaessa muokata tiedoston sisältöä. Komentoina käytettiin Komento 13. Käyttäjät nimettiin salasana.yml tiedostoon (Kuva 15) mukaisesti, ja samassa muodossa, kuin salasanoja tullaan myöhemmin käyttämään yhteysmuuttujissa.

Komento 13 Vault-tiedoston luonti- määritys- ja tarkastuskomennot

```
$ cd /home/asentaja
$ mkdir holvi
$ ansible-vault create salasana.yml

# Salasanan luonti toisella Ubuntu-välilehdellä
$ openssl rand -base64 14

# Tiedoston sisällön tarkastelu
$ ansible-vault view salasana.yml

# Tiedoston sisällön muokkaaminen
$ ansible-vault edit salasana.yml
```

Kuva 15 Ansible-vault salasanatiedoston simuloitu sisältö

```
  GNU nano 4.8                  /home/asentaja/
raspi_4_sudo_passwd: salainensalasana
ubuntu_asentaja_passwd: $6$y.S4MbVqHElb$CP
```

4.2.4 Yhteysmuuttujien lisääminen projekti-inventaariotiedostoon

Yhteysmuuttajat lisättiin projekti-inventaariotiedostoon niitä varten luodun servers:vars -muuttujaryhmän sisään. Muuttujien asetuksilla määritettiin Ansible käyttämään SSH-yhteyteen RasPin ubuntu-käyttäjää, ja nostamaan käyttäjän oikeudet sudo-tasolle. Jotta tämä olisi mahdollista, tuli muuttujaan määrittää vault-salasanatiedostossa oleva salasana. Tiedostoon lisättiin myös (Kuva 16) mukaisesti asentaja-käyttäjä ja käyttäjän salasana, jotta tiedot olisivat käytössä myöhemmin luotavassa toisessa pelikirjan ajossa. Asetuksien määrittämiseen käytettiin apuna Vivek Giten esimerkkiä (Gite, 2019), sekä Ansiblen dokumentaatiota yhteysmuuttujista. (Ansible, n.d.-t)

Kuva 16 Yhteysmuuttujat projekti-inventaariotiedostossa

```
# servers-ryhmä
[servers]
raspi_4 ansible_host=192.168.3.16

# servers-ryhmän yhteysmuuttujat
[servers:vars]
ansible_user=ubuntu
ansible_become=yes
ansible_become_method=sudo
ansible_become_password='{{raspi_4_sudo_passwd}}'
user=asentaja
user_passwd='{{ubuntu_asentaja_passwd}}'

# Kaikille ryhmille yhteiset muuttujat
[all:vars]
ansible_python_interpreter=/usr/bin/python3
```

4.3 Ansible-pelikirjojen luominen ja ajaminen

4.3.1 update-upgrade.yml -pelikirjan käyttö testiajona

Kun Vault-salasanatiedosto ja projekti-inventaariolista sisälsivät tarvittavat tiedot, voitiin luoda pelikirja, jonka tehtävät suorittavat valitun inventaariolistan kaikkien palvelinkäyttöjärjestelmien pakettien päivittämisen viimeisimpien versioiden mukaiseksi, mikäli julkaistu päivitys on yhtä tuntia vanhempi. Pelikirjan Ohjelmakoodi 2 tehtävissä määritettiin myös vanhojen ja käyttämättömien pakettien puhdistaminen ja poistaminen, sekä päivityksen tapahtumista informoivan viestin lisääminen ajoraporttiin.

Ohjelmakoodi 2 update-upgrade -pelikirja (Liite 2)

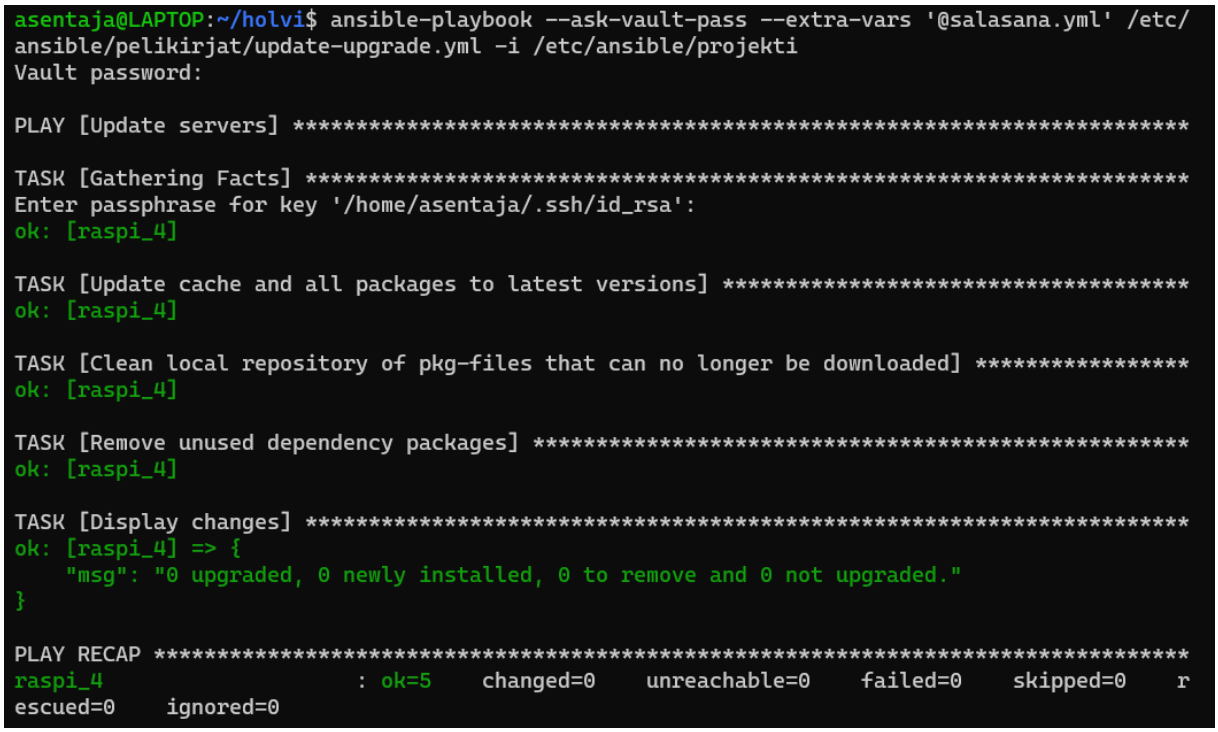
```
  GNU nano 4.8                          update-upgrade.yml
---
- name: Update servers
  hosts: all

  tasks:
  - name: Update cache and all packages to latest versions
    register: updatesys
    ansible.builtin.apt:
      name: "*"
      state: latest
      update_cache: yes
      cache_valid_time: 3600

  - name: Clean local repository of pkg-files that can no longer be downloaded
    ansible.builtin.apt:
      autoclean: yes

  - name: Remove unused dependency packages
    ansible.builtin.apt:
      autoremove: yes

  - name: Display changes
    ansible.builtin.debug:
      msg:  "{{updatesys.stdout_lines|last}}"
```

Pelikirja ajettiin lisäämällä ansible-playbook -komentoon --ask-vault-pass -argumentti kysymään käyttäjältä vault-salasanaa. Tiedostossa oleva salasana välitettiin komennon --extra-vars -argumentilla projekti-inventaariotiedostossa olevaan yhteysmuuttujaan. Jotta testiajo saatiin suoritettua, oli komentoon vielä lisättävä polut sekä pelikirjan että projekti-inventaariotiedoston sijainteihin. Pelikirjan ajoon käytettiin komentoa Komento 14, ja se täytyi ajaa polusta, jossa salasanatiedosto sijaitsi. Ajon suorittaminen edellytti vielä käyttäjänsyötteenä SSH-yhteyden tunnuslauseen. Onnistuneen pelikirjan ajon tuloksena oli (Kuva 17) mukainen raportti.

Komento 14 update-upgrade -pelikirjan ajokomento

```
$ ansible-playbook --ask-vault-pass --extra-vars '@salasana.yml'
/etc/ansible/pelikirjat/update-upgrade.yml -i /etc/ansible/projekti
```

Kuva 17 Raportti update-upgrade -pelikirjan ajosta



4.3.2 Yhteys- käyttäjä- ja konfigurointiasetukset -pelikirja

Pelikirjalle (Liite 3) annettiin nimeksi server-setup.yml ja sen muokkaamiseen käytettiin Microsoft Visual Studio Code -ohjelmaa Red Hatin YAML-kielituella, josta pelikirja kopioitiin manuaalisesti nano-editoriin. Pelikirjassa käytettiin ohjelmakoodia Ohjelmakoodi 3 lisäämään RasPille asentaja-niminen käyttäjä sudo-oikeuksilla. Ansible lisää käyttäjälle automaattisesti myös kotikansion, ellei sen luontia erikseen kielletä. Käyttäjänimi otettiin projekti-inventaariotiedoston muuttujasta, ja käyttäjän salasanaksi määritettiin vault-tiedostosta muuttujaan sijoitettu salasana. Salasanan salaamiseen käytettiin tehtävän sisällä 512-bittistä salausalgoritmia.

Ohjelmakoodi 3 Lisää sudokäyttäjä -tehtävä

```
- name: Add Ansible user with a bash shell, appending to the group 'sudo'
  ansible.builtin.user:
    name: "{{ user }}"
    password: "{{ user_passwd | password_hash('sha512') }}"
    update_password: on_create
    state: present
    shell: /bin/bash
    groups: sudo
    append: yes
```

Tarkoituksena oli, että kaikki Ansiblella suoritettavat lisäkonfiguroinnit ja -asennukset tehtäisiin jatkossa käyttäen kyseistä asentaja-käyttäjätiliä, joten pelikirjan tehtävässä luotiin SSH-tunneliyhteys käyttäjien välille. Tämä tehtiin kopioimalla WSL2 asentaja-käyttäjän julkinen avain RasPin asentaja-käyttäjän authorized_keys -tiedostoon käyttämällä pelikirjan tehtävässä ohjelmakoodia Ohjelmakoodi 1. Authorized_keys -tiedoston tietoturvaa parannettiin määrittämällä asentaja-käyttäjälle luku- ja kirjoitusoikeuden sijaan vain lukuoikeus Ohjelmakoodi 4.

Ohjelmakoodi 4 authorized_keys -tiedoston käyttöoikeuksien muokkaustehtävä

```
- name: Set file permissions on /home/asentaja/.ssh/authorized_keys
  ansible.builtin.file:
    path: /home/asentaja/.ssh/authorized_keys
    owner: asentaja
    group: asentaja
    mode: 0400
```

Muita yleisiä ja tietoturvaa parantavia konfigurointiasetuksia tehtiin sshd_config -tiedostoon. Ensin varmistettiin, että root-käyttäjän kirjautuminen palvelimelle käyttämällä salasanaa on estetty, ja sitten estettiin kaikkien salasanojen käyttö SSH-yhteyden avaamiseksi laitteiden välille, jolloin yhteys muodostetaan jatkossa vain SSH-avainparia ja SSH-yhteyden tunnuslausetta käyttäen.

Ensimmäistä pelikirjaa ajettaessa todettiin, ettei ubuntu-käyttäjältä vaadittu pääkäyttäjän salasanaa sudo-komentoja annettaessa, joten asetusta kovennettiin 90-cloud-init-users -tiedostossa. Muutokset sshd_config -tiedostossa tulevat voimaan vasta kun sshd-palvelu käynnistetään uudelleen, joten palvelun uudelleenkäynnistämistä varten lisättiin pelikirjaan oma käsittelijänsä. Pelikirjaan lisättiin ohjelmakoodit Ohjelmakoodi 5.

Ohjelmakoodi 5 Tehtävät sshd_config ja 90-cloud-init-users -tiedostojen asetuksille

```
- name: Disable password authentication for root
  ansible.builtin.lineinfile:
    path: /etc/ssh/sshd_config
    state: present
    regexp: '^#?PermitRootLogin'
    line: 'PermitRootLogin prohibit-password'
  notify:
    - restart sshd

- name: Disable tunneled clear-text passwords
  ansible.builtin.lineinfile:
    path: /etc/ssh/sshd_config
```

```
    state: present
    regexp: '^#?PasswordAuthentication'
    line: 'PasswordAuthentication no'

  notify:
    - restart sshd


- name: Ask sudo password from user 'ubuntu'
  ansible.builtin.lineinfile:
    path: /etc/sudoers.d/90-cloud-init-users
    state: present
    regexp: 'ubuntu'
    line: 'ubuntu ALL=(ALL) PASSWD:ALL'


handlers:
  - name: restart sshd
    ansible.builtin.service:
      name: sshd
      state: restarted
    when: reboot_required_file.stat.exists == false
```

Myös Ubuntuun sisäänrakennettu Ufw-palomuuri otettiin käyttöön, ja lisättiin sääntö joka sallii SSH-yhteyden RasPiin ohjelmakoodilla Ohjelmakoodi 6.

Ohjelmakoodi 6 Ufw-palomuurin sääntö sallia SSH-yhteys

```
- name: Allow OpenSSH
  community.general.ufw:
    rule: allow
    name: OpenSSH

- name: Enable ufw
  community.general.ufw:
    state: enabled
```

Pelikirjaan lisättiin kokonaisuudessaan myös kaikki update-upgrade.yml -pelikirjan tehtävät. Päivytyksien yhteydessä palvelin on usein käynnistettävä uudelleen. Jotta pelikirjan ajo voisi tarvittaessa käynnistää palvelimen uudelleen, sitä varten lisättiin oma tehtävänsä jonka moduuli rekisteröi uudelleenkäynnistämisen tarpeen, ja oma tehtävänsä, joka suorittaa itse uudelleenkäynnistämisen. Ohjelmakoodi 7

Ohjelmakoodi 7 Uudelleenkäynnistämisen tehtävät

```
- name: Check if reboot is reguired
  register: reboot_reguired_file
  ansible.builtin.stat:
    path: /var/run/reboot-reguired
```

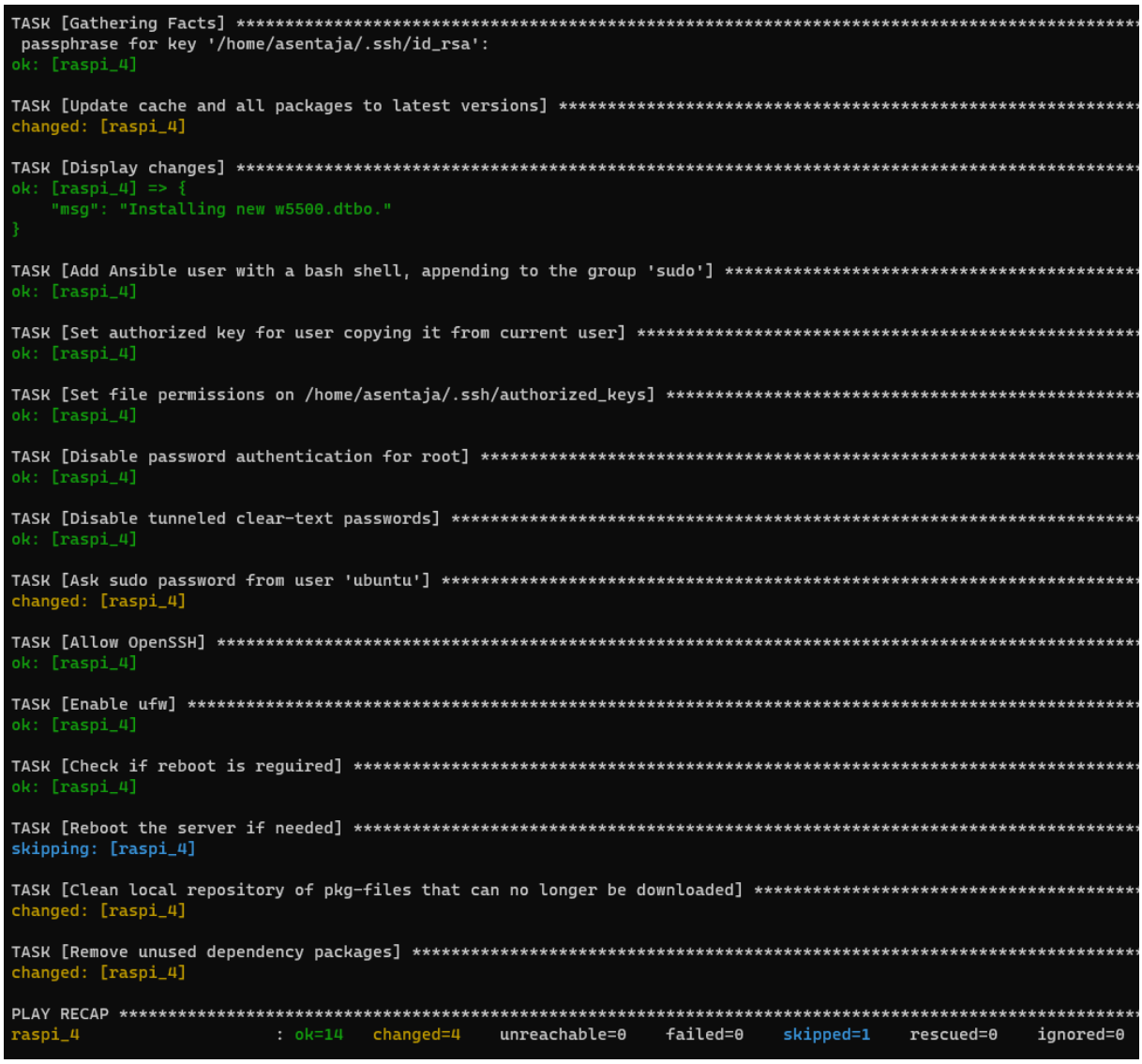
```
- name: Reboot the server if needed
  ansible.builtin.reboot:
    msg: "Reboot initiated by Ansible because of reboot required file."
    connect_timeout: 5
    reboot_timeout: 600
    pre_reboot_delay: 0
    post_reboot_delay: 30
    test_command: whoami
  when: reboot_reguired_file.stat.exists
```

Server-setup pelikirjaa ajettiin useita kertoja polusta jossa salasanatiedosto sijaitsi komennolla Komento 15, ja viimeisen ajon tuloksena saatiin ajoraporttina (Kuva 18).

Komento 15 server-setup -pelikirjan ajokomento

```
$ ansible-playbook --ask-vault-pass --extra-vars '@salasana.yml'
/etc/ansible/pelikirjat/server-setup.yml -i /etc/ansible/projekti
```

Kuva 18 Raportti server-setup -pelikirjan viimeisestä ajosta



4.3.3 Konfigurointien tarkistuksia

Pelikirjan ajon jälkeen tarkastettiin tehtyjen konfigurointien toteutuminen ja niiden oikeellisuus kirjautumalla Raspberry Pi 4:lle. Komennoilla Komento 16 tarkistettiin ensin että asentaja-käyttäjä on lisätty sudo-ryhmään, käyttäjälle on lisätty authorized_keys-tiedosto ja sinne julkinen avain, ja tiedoston käyttöoikeudet on rajoitettu vain käyttäjän luettavaksi.

Komento 16 Luodun asentaja-käyttäjän tarkistuskomennot

```
ubuntu@ubuntu:~$ getent group sudo
sudo:x:27:ubuntu,asentaja
ubuntu@ubuntu:~$ sudo ls -al /home/asentaja/.ssh
total 12
drwx------ 2 asentaja asentaja 4096 Feb 10 20:12 .
drwxr-xr-x 3 asentaja asentaja 4096 Feb 10 20:12 ..
-r-------- 1 asentaja asentaja  741 Feb 10 20:12 authorized_keys
```

Seuraavaksi tarkistettiin, että salasanalla kirjautuminen on estetty ja kirjautumisen olevan sallittu vain asentaja-käyttäjän SSH-avainparilla sekä antamalla lisäksi yksityiseen avaimeen liitetty tunnuslause. Tarkistukset tehtiin komennoilla Komento 17 .

Komento 17 Yhteyden tarkistuskomennot hallintakoneelta etäkoneelle.

```
Administrator: Windows PowerS    asentaja@LAPTOP: ~
sampsas@LAPTOP:/mnt/c/Users/samps$ cd
sampsas@LAPTOP:~$ ssh ubuntu@192.168.3.16
ubuntu@192.168.3.16: Permission denied (publickey).
sampsas@LAPTOP:~$ ssh asentaja@192.168.3.16
asentaja@192.168.3.16: Permission denied (publickey).
sampsas@LAPTOP:~$ su asentaja
Password:
asentaja@LAPTOP:/home/sampsas$ cd
asentaja@LAPTOP:~$ ssh asentaja@192.168.3.16
Enter passphrase for key '/home/asentaja/.ssh/id_rsa':
```

Seuraavaksi tarkistettiin komennoilla Komento 18, että ubuntu-käyttäjältä on määritetty kysyttäväksi sudo-salasanaa. Komennolla Komento 19 tarkistettiin Ufw-palomuurin tila.

Komento 18 Ubuntu-käyttäjän sudo-salasanan kysymisen tarkistuskomennot

```
asentaja@ubuntu:~$ su ubuntu
Password:
ubuntu@ubuntu:/home/asentaja$ cd
ubuntu@ubuntu:~$ sudo cat /etc/sudoers.d/90-cloud-init-users
[sudo] password for ubuntu:
# Created by cloud-init v. 21.2-3-g899bfaa9-0ubuntu2~20.04.1

# User rules for ubuntu
ubuntu ALL=(ALL) PASSWD:ALL
```

Komento 19 Ufw-palomuurin tilan tarkistuskomento

```
ubuntu@ubuntu:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
```

5 Johtopäätökset ja pohdinta

Opinnäytetyön tavoitteena oli luoda uudelleenajettavia Ansible-pelikirjoja, joiden avulla Linux-palvelimelle tehtiin automatisoidusti yhteys- käyttäjä- ja koventamiskonfigurointeja. Oma osaamiseni pohjautuu HAMKin tietojenkäsittelyn koulutuksessa ja järjestelmän hallintaan profiloivissa opinnoissa hankittuihin tietoihin ja taitoihin. Koulutuksessa Ansible mainittiin työkaluna, jonka opiskelu olisi looginen seuraava vaihe, ja jonka osaajista olisi kysyntää. Tätä taustaa vasten valitsin opinnäytetyöni keskiöön Ansiblen.

Raspberry Pi 4 valikoitui laitteeksi, johon Ansiblella etähallittava palvelinkäyttöjärjestelmä asennettiin, koska tarkoituksena on tulevaisuudessa harjoitella lisää siihen tehtäviä asennuksia omassa paikallisessa lähiverkossa. Minitietokoneen uusimman 4-version tekniset ominaisuudet ovat edistyneet jo sille tasolle, että laitetta voidaan käyttää 64-bittisyyttä edellyttäville asennuksille. Raspberry Pi:lle asennettu 64-bittinen Ubuntu Server 20.04.3 LTS on yleisesti käytössä oleva palvelinkäyttöjärjestelmä ja siihen tehdyt asennukset toimivat hyvinä harjoitteina työelämää ajatellen. Arviolta noin 95-prosenttia kaikista Ubuntu-asennuksista on pitkäaikaisesti tuettuja LTS-versioita, niiden ollessa yrityskäyttötasoisia ja eniten käytettyjä Ubuntu-käyttöjärjestelmiä.

Ansiblen ydintoimintojen kuvaaminen teoriaosuuteen osoittautui haastavaksi tehtäväksi virallisen dokumentaation ollessa melko hajanaista. Onnistuin kuitenkin mielestäni avaamaan lukijalle näitä Ansiblen toiminnan peruspilareita tavalla, joka tukee käytännön osuudessa tehtävien määritysten ja pelikirjojen ymmärtämistä. Ansiblen ydintoimintoihin kuuluu myös ansible.cfg -konfigurointitiedosto. Jätin tämän tiedoston kuvaamisen teoriaosuuden ulkopuolelle, koska dokumentaatio painottaa tiedoston käyttöön liittyviä tietoturvariskejä WSL-ympäristössä. Riskit ovat dokumentaation mukaan vältettävissä. Tämä opinnäytetyö ei kuvaa näiden toimien toteutusta. Asennettu Ansible 5.2.0 community ei sisältänyt automaattisesti mainittua ansible.cfg -tiedostoa, vaan se olisi pitänyt ladata käyttöön erillisellä komennolla. Tämä kuvaa mielestäni sitä, kuinka tärkeää Ansiblen kehittäjien taholta on varmistaa tiedoston harkittu käyttöönotto ja käyttäminen. Tiedostoon tehtävällä määrityksellä voisi esimerkiksi määrittää Ansible lukemaan inventaariotiedosto halutusta polusta, mikä helpottaisi komentokehotteessa annettavan pelikirjan ajokomennon

antamista lyhyemmässä muodossa, mutta olisi samalla väärissä käsissä vakava tietoturvariski.

Ansiblen dokumentaatiota seuraten toteutettu asennus Ubuntulle lisäsi järjestelmään ppa.launchpad.net -lähteen (Personal Package Archive), jonka kautta Ansiblen-asennusta tullaan jatkossa päivittämään normaalien järjestelmäpäivitysten yhteydessä. Lähteen lisääminen vaikutti aluksi epäilyttävältä, koska lähdesivustolla mainittiin PPA:n olevan epäluotettava, ja ettei järjestelmää päivittäviä paketteja tuettaisi, ja että PPA-asennukset tehtäisiin omalla vastuulla. Koska kyseessä on Ansiblen virallinen asennuskanava Ubuntulle, ja sen käyttäessä PPA-avainta varmentamaan sen, ettei asennettaviin paketteihin ole koskettu niiden luomisen jälkeen, päätin pitää asennuskanavaa luotettavana.

Hallintakoneena käytetty Windows 10, ja sen WSL2 ympäristöön Ubuntulle asennettu Ansible ovat myös oma tietoturvakysymyksensä, johon tämä opinnäytetyö ei täysin vastaa. Tietoturvan tasoa parantavana ratkaisuna WSL2-Ubuntulle tuli kirjautua salasanaa käyttäen, josta tuli edelleen kirjautua Ansiblen käyttöön luodulle salasanasuojatulle käyttäjätilille, jota käytettiin pelikirjojen ajamiseen tunnuslauseella suojattua SSH-yhteyttä käyttäen. Ohjelman luonteen ollessa etäkoneita hallitseva, tulisi hallintakoneen tietoturvan tasoon kiinnittää vielä lisää huomiota.

Työssä tehtyjä asennuksia Ubuntu-palvelimelle olisi myös voinut harjoitella esimerkiksi HAMKin tietojenkäsittelyn opiskelijoille tarjoamassa virtuaalisessa Embotics vCommander-palvelussa, jossa käyttöjärjestelmät olisi tilattu käyttöön virtuaalikoneina. Raspberry Pi 4:n toimiessa työssä harjoituslaitteena oli varmuuskopioinnista huolehdittava itse ottamalla microSD-kortista kopioita palautuspisteeksi, josta olikin kerran hyötyä, kun onnistuin estämään väärä asetus tekemällä itseltäni pääsyn järjestelmään.

Onnistuneiden yhteys- käyttäjä- ja koventamiskonfigurointien ajaminen etäkoneelle edellytti ensinnäkin etäkoneen lisäämisen inventaariotiedostoon, ja toiseksi pelikirjan luomisen, jossa määriteltiin etäkoneelle ajettavat konfiguroinnit. Ansiblen oman Vault-salasanatiedoston käyttöönotto varmisti sen, että etäkoneelle kirjautumiseen käytetty salasana haettiin tästä salatusta tiedostosta. Myös etäkoneelle luodulle käyttäjälle asetettava salasana haettiin samasta Vault-tiedostosta. Pelikirjoilla etäkoneelle ajetut konfiguroinnit olivat perustasoa ja

esimerkinomaisia. Järjestelmän kattava koventaminen vaatisi lisää tehtäviä pelikirjaan, kuten esimerkiksi SSH-portin vaihtamisen ja Fail2Ban-ohjelman asentamisen kontrolloimaan palvelimelle kirjautumisia.

WSL2-Ubuntu ja Raspberry Pi 4 toimivat yhdessä mielestäni hyvin paikallisena harjoitusympäristönä Ansiblen toiminnallisuuksien opiskeluun, ja luodut pelikirjat toimivat tehtyjen konfiguraatioiden dokumentaationa ja ovat uudelleenajettavissa sekä muokattavissa erilaisiin tarpeisiin. Tavoitteenani oli, että opinnäytetyön käytännön osuudessa kuvatut asennukset ja määritykset toimisivat ohjeistuksena myös muille järjestelmän hallintaa opiskeleville henkilöille Ansible-harjoitusympäristön luomiseksi. Tämän tavoitteen toteutumista ei kuitenkaan todennettu toisen opiskelijan tekemällä asennuksella luomaani ohjeistusta seuraten.

Tutkimuskysymyksiin vastauksia etsiessäni vastaan tuli myös muita mielenkiintoisia näkökulmia ja tapoja toimia, kuten tapa automatisoida pilvessä olevan Ubuntu-palvelimen konfigurointi ensimmäisen käynnistyksen yhteydessä käyttämällä Ubuntun omaa Cloud-init asetustiedostoa. Toimintatavalla varmistetaan, että haastavassa pilviympäristössä käynnistyvän Ubuntu-palvelimen tietoturva on heti alussa halutulla tasolla.

Toinen mielenkiintoinen vastaan tullut näkökulma oli Ubuntun esiin nostama Raspberry Pi 4:llä ARM64-prosessoripohjaiselle käyttöjärjestelmälle kehitetyn, ja konttiteknologiaa käyttävän sovelluksen nostaminen Amazonin Graviton2-pilveen. Vaikuttaakin, että Raspberry Pi 4 soveltuu hyvin harjoitus- ja kehitysympäristöksi niin järjestelmän hallintaa, kuin sovelluskehitystäkin opiskeleville henkilöille.

6 Yhteenveto

Projektiin valittu Raspberry Pi 4 minitietokone on suosittu alusta erilaisille harrasteprojekteille. Keskityin kuitenkin enemmän kuvaamaan laitteen teknisiä ominaisuuksia, sen toimiessa Ansiblella hallittavana etäkoneena, johon on asennettu yleisesti pilvipalveluntarjoajien käytössä oleva Linux-palvelin.

Laitteeseen vertailun pohjalta asennettavaksi valittu käyttöjärjestelmä Ubuntu Server 20.04.3 LTS oli ennalta arvattava ja turvallinen valinta sen täyttäessä kaikki asetetut valintakriteerit. Vaatimus Docker Engine yhteensopivuudelle oli puolestaan tekijä, joka määritti käyttöjärjestelmän 64-bittisyyden. Laitevalmistajan oma Raspberry Pi OS (64-bit) vakaa versio julkaistiin 28.1.2022 kesken tämän työn tekemisen. Koska työ oli jo pitkällä, ja vertailun pohjalta valittu käyttöjärjestelmä oli asennettuna, en ottanut sitä mukaan vertailuun.

Ansiblen dokumentaatio on yhtä laaja kuin on systeemityön työkenttä, ja siinä esitettyjä malleja tulee osata soveltaa ja yhdistellä halutun toiminnallisuuden saavuttamiseksi. Dokumentaatio on myös hieman hajanaista ja vaati allekirjoittaneelta syvällistä paneutumista. Kuitenkin, kun WSL2 ympäristöön asennetun hallintakoneen on saanut valmisteltua pelikirjojen ajoa varten, ja soveltamalla dokumentaatiota ja internetistä löytyviä ohjeita on rakentanut pelikirjan, jonka ajon edistymistä voi seurata komentokehoteikkunassa, on Ansiblen käyttö nopeasti hyvin palkitsevaa. Inhimillisille virheille herkät manuaaliset konfiguroinnit on automatisoitu, ja samalla dokumentoitu, sekä muokattavissa. Samaa pelikirjaa ajamalla voisi ajaa asetukset isoon joukkoon etäkoneita. Pelikirjojen uudelleenajettavuus mahdollistaa ja helpottaa konfigurointien lisäämisen ja muokkaamisen saman pelikirjan sisällä, koska etäkoneella jo halutussa tilassa olevia asetuksia ei ajeta uudelleen. Tämä Ansiblen ominaisuus on käytännöllinen, koska pelikirjaa voi koeajaa jatkuvasti, ja todeta muokkausten toimivuus.

Opin työn tekemällä Ansiblen peruskäyttöä ja tarkoitus on tulevaisuudessa asentaa Ansiblea käyttäen seuraavaksi Raspberry Pi 4:lle Docker Engine. Mahdollisia tulevaisuuden asennuksia harjoitusympäristöön ovat järjestelmää monitoroiva Prometheus sekä sen tuottaman datan esittäminen Grafana-kojelaudan avulla.

Lähteet

Abraham, J. (18.10.2019). *Basic server hardening using Ansible*. https://able.bio/jibiabraham/basic-server-hardening-using-ansible--127r833

Ansible-cmdb. (n.d.). *About.* Ansible-CMDB. Haettu 7.2.2022 osoitteesta https://ansible-cmdb.readthedocs.io/en/latest/

Ansible-community. (12.1.2022). *ansible-build-data/CHANGELOG-v5.rst.* https://github.com/ansible-community/ansible-build-data/blob/main/5/CHANGELOG-v5.rst#release-summary

Ansible. (n.d.-a). *ansible.builtin.apt – Manages apt-packages.* Ansible Documentation. Haettu 7.2.2022 osoitteesta https://docs.ansible.com/ansible/latest/collections/ansible/builtin/apt_module.html#ansible-collections-ansible-builtin-apt-module

Ansible. (n.d.-b). *ansible.builtin.debug – Print statements during execution.* Ansible Documentation. Haettu 7.2.2022 osoitteesta https://docs.ansible.com/ansible/latest/collections/ansible/builtin/debug_module.html#ansible-collections-ansible-builtin-debug-module

Ansible. (n.d.-c). *ansible.builtin.file – Manage files and file properties.* Ansible Documentation. Haettu 13.2.2022 osoitteesta https://docs.ansible.com/ansible/latest/collections/ansible/builtin/file_module.html#ansible-collections-ansible-builtin-file-module

Ansible. (n.d.-d). *ansible.builtin.lineinfile – Manage lines in text files.* Ansible Documentation. Haettu 13.2.2022 osoitteesta https://docs.ansible.com/ansible/latest/collections/ansible/builtin/lineinfile_module.html

Ansible. (n.d.-e). *ansible.builtin.reboot – Reboot a machine.* Ansible Documentation. Haettu 13.2.2022 osoitteesta https://docs.ansible.com/ansible/latest/collections/ansible/builtin/reboot_module.html#ansible-collections-ansible-builtin-reboot-module

Ansible. (n.d.-f). *ansible.builtin.service – Manage services.* Ansible Documentation. Haettu 13.2.2022 osoitteesta https://docs.ansible.com/ansible/latest/collections/ansible/builtin/service_module.html#ansible-collections-ansible-builtin-service-module

Ansible. (n.d.-g). *ansible.builtin.stat – Retrieve file or file system status.* Ansible Documentation. Haettu 13.2.2022 osoitteesta https://docs.ansible.com/ansible/latest/collections/ansible/builtin/stat_module.html#ansible-collections-ansible-builtin-stat-module

Ansible. (n.d.-h). *ansible.builtin.user – Manage user accounts.* Ansible Documentation. Haettu 13.2.2022 osoitteesta https://docs.ansible.com/ansible/latest/collections/ansible/builtin/user_module.html#ansible-collections-ansible-builtin-user-module

Ansible. (n.d.-i). *ansible.posix.authorized_key – Adds or removes an SSH authorized key.* Ansible Documentation. Haettu 31.1.2022 osoitteesta https://docs.ansible.com/ansible/latest/collections/ansible/posix/authorized_key_module.html#ansible-posix-authorized-key-adds-or-removes-an-ssh-authorized-key

Ansible. (n.d.-j). *Ansible - Product Pricing*. Haettu 27.1.2022 osoitteesta https://www.ansible.com/products/pricing

Ansible. (n.d.-k). *Ansible architecture.* Ansible Documentation. Haettu 7.2.2022 osoitteesta https://docs.ansible.com/ansible/latest/dev_guide/overview_architecture.html#ansible-architecture

Ansible. (n.d.-l). *Ansible Documentation.* Ansible Documentation. Haettu 27.1.2022 osoitteesta https://docs.ansible.com/ansible/latest/index.html

Ansible. (n.d.-m). *Collection Index.* Ansible Documentation. Haettu 1.2.2022 osoitteesta https://docs.ansible.com/ansible/latest/collections/index.html

Ansible. (n.d.-n). *community.general.ufw – Manage firewall with UFW.* Ansible Documentation. Haettu 13.2.2022 osoitteesta https://docs.ansible.com/ansible/latest/collections/community/general/ufw_module.html

Ansible. (n.d.-o). *Encrypting content with Ansible Vault.* Ansible Documentation. Haettu 3.2.2022 osoitteesta https://docs.ansible.com/ansible/latest/user_guide/vault.html

Ansible. (n.d.-p). *How to build your inventory.* Ansible Documentation. Haettu 25.1.2022 osoitteesta https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html

Ansible. (n.d.-q). *Installing Ansible.* Ansible Documentation. Haettu 7.2.2022 osoitteesta https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html#prerequisites

Ansible. (n.d.-r). *Installing Ansible.* Ansible Documentation. Haettu 30.1.2022 osoitteesta https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html#installing-ansible-on-ubuntu

Ansible. (n.d.-s). *Intro to playbooks.* Ansible Documentation. Haettu 2.2.2022 osoitteesta https://docs.ansible.com/ansible/latest/user_guide/playbooks_intro.html#playbooks-intro

Ansible. (n.d.-t). *Understanding privilege escalation: become.* Ansible Documentation. Haettu 6.2.2022 osoitteesta https://docs.ansible.com/ansible/latest/user_guide/become.html#become-connection-variables

Ansible. (n.d.-u). *Working with command line tools.* Ansible Documentation. Haettu 1.2.2022 osoitteesta https://docs.ansible.com/ansible/latest/user_guide/command_line_tools.html

Boelen, M. (7.7.2018). *Linux hardening steps for starters.* Linux Audit. https://linux-audit.com/linux-server-hardening-most-important-steps-to-secure-systems/

Debian. (n.d.). *DebianReleases.* Debian Wiki. Haettu 22.1.2022 osoitteesta https://wiki.debian.org/DebianReleases

Dineshbaburam. (13.11.2020). *Ansible Architecture. What is Ansible?.* https://dineshbaburam91.medium.com/ansible-architecture-a22d3b8af699

Docker. (n.d.). *Docker Documentation*. Haettu 21.1.2022 osoitteesta https://docs.docker.com/engine/install/ubuntu/

Ellingwood, J. (5.1.2022). *How To Use Ansible Vault to Protect Sensitive Playbook Data.* DigitalOcean. https://www.digitalocean.com/community/tutorials/how-to-use-vault-to-protect-sensitive-ansible-data

Gite, V. (2.1.2019). *How to set and use sudo password for Ansible Vault.* nixCraft. https://www.cyberciti.biz/faq/how-to-set-and-use-sudo-password-for-ansible-vault/

Heidi, E. (15.5.2020). *How To Install and Configure Ansible on Ubuntu 20.04.* DigitalOcean. https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-ansible-on-ubuntu-20-04

IBM. (2.12.2019). *Infrastructure as Code.* IBM. https://www.ibm.com/cloud/learn/infrastructure-as-code

Raspberrypi. (n.d.-a). *Buy a Raspberry Pi 4 Model B.* Raspberry Pi. Haettu 8.2.2022 osoitteesta https://www.raspberrypi.com/products/raspberry-pi-4-model-b/?variant=raspberry-pi-4-model-b-8gb

Raspberrypi. (n.d.-b). *Raspberry Pi Imager update to v1.6.* Raspberry Pi. Haettu 19.1.2022 osoitteesta https://www.raspberrypi.com/news/raspberry-pi-imager-update-to-v1-6/

Raspberrypi. (n.d.-c). *Raspberry Pi OS.* Raspberry Pi. Haettu 19.1.2022 osoitteesta https://www.raspberrypi.com/software/

Datasheets.Raspberrypi. (2021). *Raspberry Pi 4 Computer Model B*. https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf

Forums.Raspberrypi. (25.4.2021). *Raspberry Pi OS (64 bit) beta test version.* Raspberry Pi Forums. https://forums.raspberrypi.com/viewtopic.php?t=275370

Downloads.Raspberrypi. (8.11.2021). *Index of /raspios_arm64/images*. https://downloads.raspberrypi.org/raspios_arm64/images/

Red Hat Ansible. (n.d.). *What is Ansible?.* Ansible Quick Start Video. Haettu 1.2.2022 osoitteesta https://www.ansible.com/resources/videos/quick-start-video?extIdCarryOver=true&sc_cid=701f2000001OH6uAAG

RedHat. (9.7.2019). *IBM Closes Landmark Acquisition of Red Hat for $34 Billion; Defines Open, Hybrid Cloud Future*. https://www.redhat.com/en/about/press-releases/ibm-closes-landmark-acquisition-red-hat-34-billion-defines-open-hybrid-cloud-future

Sarav, A. (19.1.2022). *Ansible apt module Examples - install packages with apt | Devops Junction*. https://www.middlewareinventory.com/blog/ansible-apt-examples/#Differentiate_the_Update_and_Upgrade

Spoclearn. (n.d.). *DevOps Courses Finland.* Spoclearn. Haettu 27.1.2022 osoitteesta https://www.spoclearn.com/fi/devops-courses

Ubuntu. (n.d.-a). *About the Ubuntu project.* Ubuntu. Haettu 8.2.2022 osoitteesta https://ubuntu.com/about

Ubuntu. (n.d.-b). *How to install Ubuntu Server on your Raspberry Pi.* Ubuntu. Haettu 22.1.2022 osoitteesta https://ubuntu.com/tutorials/how-to-install-ubuntu-on-your-raspberry-pi#4-boot-ubuntu-server

Ubuntu. (n.d.-c). *Installation.* Ubuntu. Haettu 8.2.2022 osoitteesta https://ubuntu.com/server/docs/installation

Ubuntu. (n.d.-d). *Raspberry Pi with Ubuntu CLI.* Ubuntu. Haettu 22.1.2022 osoitteesta https://ubuntu.com/raspberry-pi/server

Ubuntu. (n.d.-e). *Releases.* Ubuntu Wiki. Haettu 22.1.2022 osoitteesta https://wiki.ubuntu.com/Releases

Ubuntu. (n.d.-f). *Ubuntu Core.* Ubuntu. Haettu 22.1.2022 osoitteesta https://ubuntu.com/core

Ubuntu. (n.d.-g). *Ubuntu release cycle.* Ubuntu. Haettu 9.2.2022 osoitteesta https://ubuntu.com/about/release-cycle#ubuntu

Upton, E. (24.5.2021). *Announcing the Raspberry Pi PoE+ HAT.* Raspberry Pi. https://www.raspberrypi.com/news/announcing-the-raspberry-pi-poe-hat/

Wilder, G. (18.11.2021). *How to Configure a New Ubuntu Server with Ansible.* Vultr. https://www.vultr.com/docs/how-to-configure-a-new-ubuntu-server-with-ansible/

Win32DiskImager. (n.d.). *Win32 Disk Imager Download (Latest Stable Version)*. Haettu 22.1.2022 osoitteesta https://win32diskimager.download/

Liite 1: Aineistonhallintasuunnitelma

Opinnäytetyössä ei käsitellä henkilötietoja tai muita luottamuksellisia tai salassa pidettäviä tietoja.

Kehitysprojektin aikana pidetään päiväkirjaa (aineisto), johon kerätään teknistä tietoa projektista. Tämä tieto analysoidaan opinnäytetyötä varten. Päiväkirjaa säilytetään tekijän tietokoneen C-asemalla, ja siitä otetaan säännöllisesti varmuuskopioita OneDrive-Hämeen ammattikorkeakoulu -kansioon. Päiväkirjaa säilytetään C-asemalla ainakin vuoden verran opinnäytetyön valmistumisesta.

Kehitysprojektin aikana pidetyistä kokouksista pidetään pöytäkirjoja. Niiden säilytys ja varmuuskopiointi tapahtuu samalla tavalla kuin päiväkirjan kanssa toimitaan.

Valmiin projektin onnistumisesta kerätään tietoa ottamalla kuvakaappauksia asennusten vaiheista. Niiden säilytys ja varmuuskopiointi tapahtuu samalla tavalla kuin päiväkirjan kanssa toimitaan.

Onnistuneista automatisoinneista Ansiblea käyttäen kertyy pelikirjoja. Myös niiden säilytys ja varmuuskopiointi tapahtuu samalla tavalla kuin päiväkirjan kanssa toimitaan. Pelikirjat liitetään opinnäytetyön liitteeksi.

Liite 2: update-upgrade.yml -pelikirja

```
---
- name: Update servers
  hosts: all

  tasks:
    # (Sarav, 2022)
    # (Ansible, ei pvm.-a)
  - name: Update cache and all packages to latest versions
    register: updatesys
    ansible.builtin.apt:
      name: "*"
      state: latest
      update_cache: yes
      cache_valid_time: 3600

  - name: Clean local repository of pkg-files that can no longer be downloaded
    ansible.builtin.apt:
      autoclean: yes

  - name: Remove unused dependency packages
    ansible.builtin.apt:
      autoremove: yes

    # (Ansible, n.d.-b)
  - name: Display changes
    ansible.builtin.debug:
      msg:  "{{updatesys.stdout_lines|last}}"
```

Liite 3: server-setup.yml -pelikirja

```
---
- name: Update, set Ansible-user, SSH, Basic security-settings, and reboot if
reguired
  hosts: all


  tasks:
    # (Ansible, n.d.-a)
    # (Sarav, 2022)
  - name: Update cache and all packages to latest versions
    register: updatesys
    ansible.builtin.apt:
      name: "*"
      state: latest
      update_cache: yes
      cache_valid_time: 3600


    # (Ansible, n.d.-b)
    # (Sarav, 2022)
  - name: Display changes
    ansible.builtin.debug:
      msg:  "{{updatesys.stdout_lines|last}}"


    # (Ansible, n.d.-h)
    # (Wilder, 2021)
  - name: Add Ansible user with a bash shell, appending to the group 'sudo'
    ansible.builtin.user:
      name: "{{ user }}"
      password: "{{ user_passwd | password_hash('sha512') }}"
      update_password: on_create
      state: present
      shell: /bin/bash
      groups: sudo
      append: yes


    # (Ansible, n.d.-i)
  - name: Set authorized key for user copying it from current user
    ansible.posix.authorized_key:
      user: "{{ user }}"
      state: present
      key: "{{ lookup('file', lookup('env','HOME') + '/.ssh/id_rsa.pub') }}"


    # (Ansible, n.d.-c)
    # (Abraham, 2019)
  - name: Set file permissions on /home/asentaja/.ssh/authorized_keys
    ansible.builtin.file:
      path: /home/asentaja/.ssh/authorized_keys
      owner: asentaja
      group: asentaja
      mode: 0400
```

```
  # (Ansible, n.d.-d)
  # (Wilder, 2021)
- name: Disable password authentication for root
  ansible.builtin.lineinfile:
    path: /etc/ssh/sshd_config
    state: present
    regexp: '^#?PermitRootLogin'
    line: 'PermitRootLogin prohibit-password'
  notify:
    - restart sshd


  # (Ansible, n.d.-d)
  # (Wilder, 2021)
- name: Disable tunneled clear-text passwords
  ansible.builtin.lineinfile:
    path: /etc/ssh/sshd_config
    state: present
    regexp: '^#?PasswordAuthentication'
    line: 'PasswordAuthentication no'
  notify:
    - restart sshd

  # (Ansible, n.d.-d)
- name: Ask sudo password from user 'ubuntu'
  ansible.builtin.lineinfile:
    path: /etc/sudoers.d/90-cloud-init-users
    state: present
    regexp: 'ubuntu'
    line: 'ubuntu ALL=(ALL) PASSWD:ALL'


  # (Ansible, n.d.-n)
- name: Allow OpenSSH
  community.general.ufw:
    rule: allow
    name: OpenSSH

- name: Enable ufw
  community.general.ufw:
    state: enabled


  # (Ansible, n.d.-g)
- name: Check if reboot is reguired
  register: reboot_reguired_file
  ansible.builtin.stat:
    path: /var/run/reboot-reguired_file get_md5=no


  # (Ansible, n.d.-e)
  # (Wilder, 2021)
- name: Reboot the server if needed
  ansible.builtin.reboot:
    msg: "Reboot initiated by Ansible because of reboot required file."
    connect_timeout: 5
    reboot_timeout: 600
    pre_reboot_delay: 0
    post_reboot_delay: 30
    test_command: whoami
  when: reboot_reguired_file.stat.exists
```

```
  # (Ansible, n.d.-a)
- name: Clean local repository of pkg-files that can no longer be downloaded
  ansible.builtin.apt:
    autoclean: yes

- name: Remove unused dependency packages
  ansible.builtin.apt:
    autoremove: yes


handlers:
    # (Ansible, n.d.-f)
    # (Wilder, 2021)
  - name: restart sshd
    ansible.builtin.service:
      name: sshd
      state: restarted
    when: reboot_required_file.stat.exists == false
```