

RF L1low -ohjelmistokehitysympäristö

Tiivistelmä

Tekijä(t) Kiukas, Jonne	Julkaisun laji Opinnäytetyö, AMK Sivumäärä 27	Valmistumisaika 2021
Työn nimi RF L1low -ohjelmistokehitysympäristö		
Tutkinto ja koulutusala Tradenomi (AMK)		
Toimeksiantajan nimi, titteli ja organisaatio Esa Bogdanoff, Ryhmänjohtaja, Nokia Oyj		
Tiivistelmä <p>Opinnäytetyö tehtiin toimeksiantona Nokia Oyj:lle, ja tarkoituksena oli luoda käyttöönotto-opas, jonka avulla uudet työntekijät voivat ottaa tietokoneella käyttöön ohjelmistokehitysympäristön, jota tarvitaan RF L1low -nimisen ohjelmiston kehittämiseen. Liitteenä olevan RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaan tarkoitus on säästää aikaa ohjelmistokehitysympäristön käyttöönottamisessa.</p> <p>Opinnäytetyön teoriaosuus toteutettiin keräämällä aiheeseen liittyvää tietoa verkkolähteistä ja haastatteleamalla Nokia Oyj:n työntekijöitä, jotka työskentelevät RF L1low -ohjelmiston kanssa. Apuna oikean tiedon löytämisessä oli myös Confluence-tietopalvelu, joka on tarkoitettu Nokia Oyj:n työntekijöille tiedon hakemista varten. RF L1low -ohjelmistokehitysympäristön käyttöönotto-opas toteutettiin puolestaan kirjaamalla toimenpiteet ylös sitä mukaan, kun yksittäisessä toimenpiteessä edettiin. Opinnäytetyössä oleva palaute käyttöönotto-oppaasta saatiin haastatteleamalla uusia työntekijöitä, jotka tulivat Nokia Oyj:n Oulun toimipisteelle kehittämään RF L1low -ohjelmistoa.</p> <p>Uudet työntekijät saivat ohjelmistokehitysympäristön käyttöön RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaan avulla, ja he pystyivät aloittamaan RF L1low -ohjelmiston kehittämisen. Tämän perusteella pystytään toteamaan, että opinnäytetyötä voidaan hyödyntää myös tulevaisuudessa, kun Nokia Oyj:lle saapuu uusi työntekijä kehittämään RF L1low -ohjelmistoa.</p>		
Asiasanat rf l1low, docker, linsee-palvelin, ssh, ohjelmistokehitysympäristö		

Abstract

Author(s) Kiukas, Jonne	Type of Publication Thesis, UAS	Published 2021
	Number of Pages 27	
Title of Publication RF L1low Software Development Environment		
Degree and field of study Bachelor of Business Administration (UAS)		
Name, title and organisation of the client Esa Bogdanoff, Squad Group Lead, Nokia Corporation		
Abstract <p>The purpose of the thesis was to create a guide which instructs new employees to start using the software development environment which, in turn, is intended to develop a software called RF L1low. Nokia Corporation is developing the previously mentioned software and the thesis has been commissioned by Nokia Corporation. RF L1low Software Development Environment Introduction Guide, which is an appendix, aims to save time once the software development environment is taken into use.</p> <p>The theory of the thesis has been compiled from various sources. It consists of information from the internet and Nokia Corporation employee interviews. In addition, accurate information has been found from the Confluence information service, which Nokia Corporation employees have the right to access. The RF L1low Software Development Environment Introduction Guide has been created by documenting the required tasks as individual actions were performed. The feedback regarding the guide has been obtained from new employees who came to Nokia Corporation Oulu office to develop the RF L1low software.</p> <p>The new employees adopted the software development environment successfully by following the RF L1low Software Development Environment Introduction Guide and were able to begin their work. The study found that the thesis can be used in the future when a new employee arrives at Nokia Corporation to develop the RF L1low software.</p>		
Keywords rf l1low, docker, linsee-server, ssh, software development environment		

Sisällys

1	Johdanto.....	1
1.1	Opinnäytetyön tausta, tutkimuskysymykset ja tavoitteet	1
1.2	Opinnäytetyön rakenne ja työmenetelmät.....	1
2	RF L1low	3
3	RF L1low -ohjelmistokehitysympäristön teknologiat	5
3.1	Docker ja LinSEE-palvelin	5
3.2	SSH.....	5
3.3	CLion-ohjelmistokehitysympäristö ja Gerrit-koodinarviointipalvelu	6
4	RF L1low -ohjelmistokehitysympäristön käyttöönoton vaiheet.....	8
4.1	Tietokoneen käyttöönottoaminen.....	8
4.2	SSH-työkalun asentaminen ja kirjautuminen LinSEE-palvelimelle	8
4.3	RF L1low -repositorion kloonaaminen ja käyttäjätietojen päivittäminen.....	9
4.4	SSH-avaimien ja JFrog Artifactory API -avaimien generointi.....	9
4.5	Docker containerin käynnistäminen ja RF L1low -repositorion tarkistaminen	10
4.6	Clion-ohjelmistokehitysympäristön konfigurointi.....	11
5	RF L1low -ohjelmiston kehitysprosessi	12
5.1	Haarat eli branches.....	12
5.2	Koodin testaaminen	13
5.3	Koodin arviointiprosessi.....	13
6	Koodin lähettäminen arvioitavaksi.....	15
6.1	Tarvittavat Git-komennot.....	15
6.2	Jira-tehtävienhallintapalvelu.....	18
6.3	Commit-viesti.....	19
7	Palaute RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaasta	21
8	Yhteenveto ja pohdinta	23
	Lähteet	25

Liitteet

Liite 1. RF L1low -ohjelmistokehitysympäristön käyttöönotto-opas

1 Johdanto

1.1 Opinnäytetyön tausta, tutkimuskysymykset ja tavoitteet

Opinnäytetyön aiheena on RF L1low -ohjelmistokehitysympäristö, ja aihe on saatu Nokia Oyj:ltä. Aloitin 3.5.2021 Nokia Oyj:n Oulun toimipisteellä kesäharjoittelun, jonka tarkoituksena oli kehittää ohjelmistoa nimeltä RF L1low. Ennen kuin pääsin aloittamaan työnteon, minun oli tehtävä erilaisia toimenpiteitä, jotta sain ohjelmistokehitysympäristön käyttöön tietokoneelle. Toimenpiteiden suorittamisen jälkeen sain idean, että voisin tehdä opinnäytetyöni siitä, mitä uuden työntekijän täytyy tehdä, jotta hän voi aloittaa RF L1low -ohjelmiston kehittämisen. Ajatukseksi muodostui myös, että opinnäytetyössä voisi kertoa RF L1low -ohjelmistosta ja eri teknologioista, jotka liittyvät ohjelmistokehitysympäristön käyttöönottamiseen. Esittelin aihe-ehdotuksen ryhmänjohtajalleni Esa Bogdanoffille ja sain häneltä luvan toteuttaa opinnäytetyön.

Opinnäytetyön ensimmäinen tutkimuskysymys on, mitkä toimenpiteet uuden työntekijän täytyy suorittaa, jotta hän saa RF L1low -ohjelmistokehitysympäristön käyttöön tietokoneelle. Toinen tutkimuskysymys on, saako työntekijä ohjelmistokehitysympäristön käyttöön tehdyn käyttöönotto-oppaan avulla. Tavoitteena on vastata esitettyihin tutkimuskysymyksiin ja tavoitteisiin on päästy, jos uusi työntekijä onnistuu saamaan ohjelmistokehitysympäristön käyttöön RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaan (Liite 1) avulla. Tarkoituksena on, että työntekijä voi opasta seuraamalla edetä omaan tahtiinsa, jolloin hänen ei tarvitse kysyä kokeneemalta työntekijältä apua suorittaakseen tietyn toimenpiteen. Tällöin niin uudelta kuin myös vanhemmaltakin työntekijältä säästyy aikaa. Opinnäytetyön tavoitteena on myös ylläpitää työntekijän motivaatiota, kun hän aloittaa RF L1low -ohjelmiston kehittämisen. Opinnäytetyön lukemisen jälkeen uudella työntekijällä on käsitys kehitettävästä RF L1low -ohjelmistosta, erilaisista teknologioista ja ohjelmistokehitysympäristön käyttöönoton vaiheista. Kyseisten asioiden selvittämisellä on positiivinen vaikutus uuden työntekijän halukkuuteen aloittaa työnteke.

1.2 Opinnäytetyön rakenne ja työmenetelmät

RF L1low -ohjelmistokehitysympäristön käyttöönotto-opas (Liite 1) on pyritty toteuttamaan niin, että huomioidaan kaikki toimenpiteet, jotka vaaditaan ohjelmistokehitysympäristön käyttöönottoprosessissa. RF L1low -ohjelmistokehitysympäristön käyttöönotto-opas (Liite 1) on rajattu alkamaan siitä pisteestä, kun uusi työntekijä on saanut itselleen Nokia Oyj:ltä tietokoneen. Opas taas päättyy siihen kohtaan, kun työntekijä on saanut suoritettua ohjelmistokehitysympäristön konfiguroinnit, joiden jälkeen hän on valmis aloittamaan RF L1low

-ohjelmiston kehittämisen. RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaassa (Liite 1) on myös ohjeet toimenpiteille, joiden suorittamisen jälkeen työntekijä voi kehittää RF L1low -ohjelmistoa etätyöskentelynä. Opinnäytetyössä kerrotaan olennaisin RF L1low -ohjelmistosta, avataan ohjelmistokehitysympäristön käyttöönottamiseen liittyvien teknologioiden merkitystä ja kerrotaan RF L1low -ohjelmistokehitysympäristön käyttöönoton vaiheista. Opinnäytetyössä on myös luku, jossa kerrotaan RF L1low -ohjelmiston kehitysprosessista ja luku, jossa kerrotaan, kuinka RF L1low -ohjelmistoon tehty uusi ominaisuus saadaan lähetettyä arvioitavaksi. Lisäksi opinnäytetyössä tuodaan esiin palaute RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaan (Liite 1) toimivuudesta, jonka uudet työntekijät antoivat sen jälkeen, kun he olivat ottaneet tietokoneella RF L1low -ohjelmistokehitysympäristön käyttöön. Lopuksi yhteenvedossa käydään tiivistetysti läpi opinnäytetyö ja pohditaan sopivaa jatkotutkimusideaa.

Opinnäytetyön teoriaosuutta varten on etsitty tietoa verkkolähteistä ja haastateltu Nokia Oyj:n työntekijöitä, jotka työskentelevät RF L1low -ohjelmiston parissa. Nokia Oyj:n työntekijöillä on pääsy Confluence-nimiseen tietopalveluun, joka sisältää tietoa Nokia Oyj:llä kehitettävistä tuotteista ja niihin liittyvistä teknologioista. Kyseinen tietopalvelu on ollut apuna, kun on etsitty tietoa, jota ei ole muualta löytynyt. RF L1low -ohjelmistokehitysympäristön käyttöönotto-opas (Liite 1) on taas toteutettu tekemällä täysin samat toimenpiteet, jotka uuden työntekijänkin täytyy tehdä, ennen kuin hän pääsee kehittämään RF L1low -ohjelmistoa. Kaikki toimenpiteet suoritettiin alusta saakka ja kun yhden toimenpiteen yksittäinen kohta saatiin tehtyä, kirjattiin tiedot ylös. Näin lopulta syntyi vaihe vaiheelta seurattava käyttöönotto-opas. Tulokset käyttöönotto-oppaan onnistumisesta saatiin myös haastattelulla ja haastateltavina olivat uudet työntekijät, jotka tulivat Nokia Oyj:n Oulun toimipisteelle kehittämään RF L1low -ohjelmistoa.

2 RF L1low

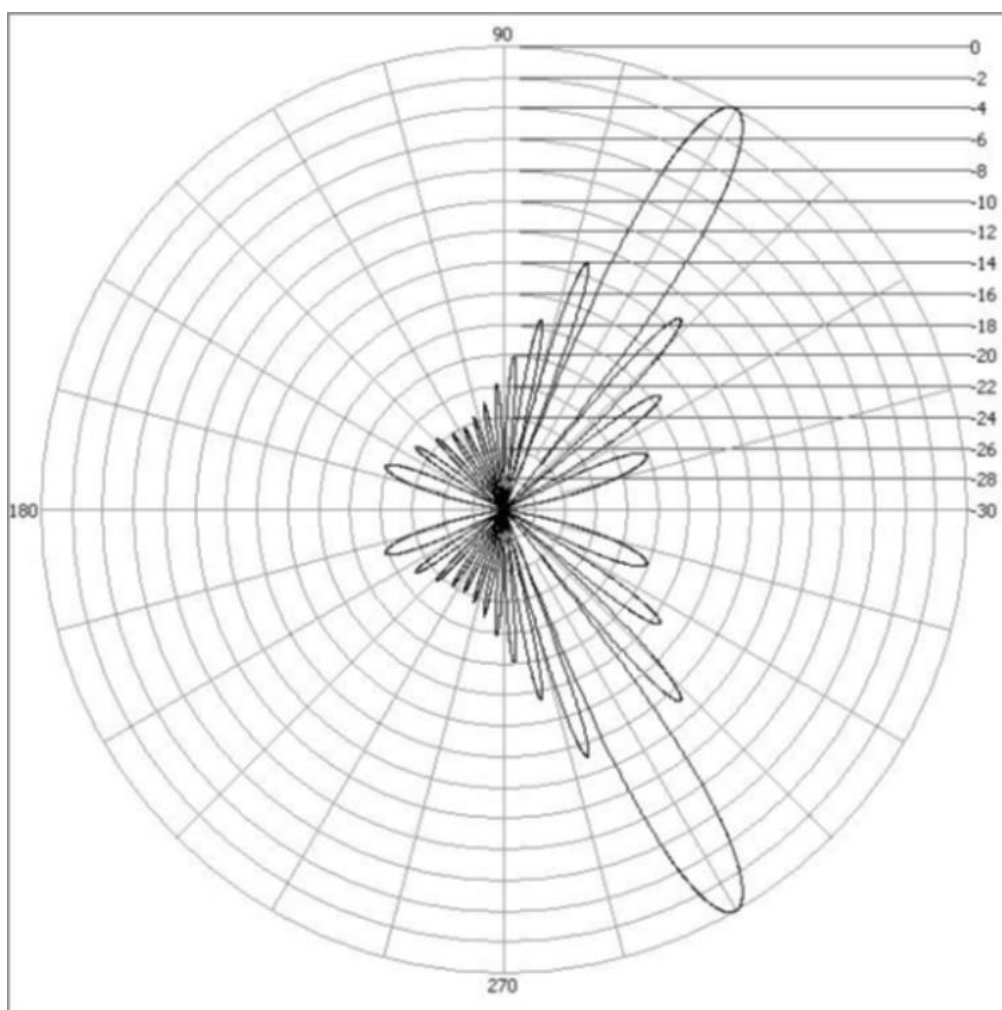
Nokia Oyj suunnittelee ja kehittää Oulun toimipisteellä uuden sukupolven radioverkkotekniikkaa, jolla voidaan suorittaa niin kutsuttua säteen muodostamista. Tavallisesti säteen muodostamisesta käytetään englanninkielistä termiä beamforming. Toisin sanoen Nokia Oyj valmistaa antennejä, jotka mahdollistavat viidennen sukupolven datayhteyden eli 5G-verkon toimivuuden. Nämä 5G-antennit vaativat toimiakseen erilaisia ohjelmistoja, joista yksi on nimeltään RF L1low. Nokia Oyj:n valmistamat 5G-antennit on toteutettu System on Chip -ratkaisulla, jossa RF L1low on jaettu laitteistoon ja ohjelmistoon. Kyse on siis sulautetusta järjestelmästä. (Hoglund 2019.)

RF L1low on radio-ohjelmistokomponentti, jonka tehtävänä on määritellä, mihin suuntaan useista osista koostuva 5G-antenni lähettää 5G-verkkoa. RF L1low -ohjelmiston toiminnallisuus perustuu siihen, että se asettaa 5G-antennin osia lähettämään eri radiotaajuuksia, jolloin osat lähettävät 5G-verkkoa yhteen suuntaan korkealla radiotaajuudella ja toiseen suuntaan matalalla radiotaajuudella. Mitä korkeampaa radiotaajuutta lähetetään, sitä pidempi 5G-verkon kantavuus on. Tällä periaatteella saadaan muodostettua pidemmälle kantava keilan muotoinen 5G-verkko, kun antennin osat lähettävät tiettyyn suuntaan korkeaa radiotaajuutta ja muihin suuntiin taas matalaa radiotaajuutta. Tätä kutsutaan myös beamforming-teoriaksi. (Arvio 2021.)

5G-verkkoa täytyy lähettää keilan muotoisena säteilynä, koska tällöin voidaan rajata tarkasti alue, jossa 5G-verkkoyhteyden halutaan sijaitsevan. Ei ole kannattavaa lähettää 5G-verkkoa tasaisena 360 asteen muodostelmana, kuten tavallista radioaaltoa lähetetään, koska 5G-verkko vaatii toimiakseen korkean radiotaajuuden. Mitä korkeampaa radiotaajuutta käytetään, sitä enemmän tarvitaan energiaa. Jos 5G-verkkoa lähetettäisiin tasaisena ympyränä, sen kantavuus olisi huomattavasti lyhyempi kuin lähetettäessä keilan muotoisena alueena. Keilan muoto tuo myös sen edun, että 5G-verkko voidaan suunnata juuri sinne, jossa sitä tarvitaan, jolloin hyötysuhteesta saadaan optimaalinen. (Arvio 2021.)

Kuvassa 1 on havainnollistettu RF L1low -ohjelmiston säätlemiä keilan muotoisia 5G-verkon kuuluvuusalueita. 5G-verkon kuuluvuutta ei ole alueilla, joihin keilat eivät ylety. Kuvasta 1 voidaan huomata selkeästi kaksi suurempaa keilan muotoista aluetta. Alueilla voidaan ajatella olevan enemmän käyttäjiä, jotka vaativat nopeampaa tiedonsiirtoa, joten 5G-verkko kannattaa suunnata pidemmälle näihin suuntiin. Kuvassa 1 on siis havainnollistettu 5G-verkon suuntaaminen kahteen olennaisimpaan suuntaan, mutta siinä näkyy myös pienempiä keilan muotoisia alueita. Näilläkin alueilla on toimiva 5G-verkko, mutta sen kantavuus on selkeästi lyhyempi. Tämä johtuu siitä, kuten aiemmin mainittiin, että RF L1low -ohjelmisto asettaa 5G-antennin osia lähettämään eri radiotaajuuksia, jolloin 5G-verkko ulottuu

pidemmälle niille alueille, joissa tarvitaan tiedonsiirtoa varten nopeampia yhteyksiä. (Drozdy 2018.)



Kuva 1. Havainnekuva keilan muotoisista 5G-verkon kuuluvuusalueista (Drozdy 2018)

RF L1low -ohjelmiston nimi tulee englannin kielen sanoista Radio Frequency Layer 1 ja low viittaa L1-protokollan alempiin kerroksiin, jotka toimivat yhdessä 5G-antennin radiotaajuusosien kanssa (Aho ym. 2019). Nokia Oyj:n työntekijöiden keskuudessa RF L1low -ohjelmistosta käytetään usein nimitystä Beaver application. Tämä nimitys tulee vastaan, kun tutkitaan RF L1low -ohjelmiston kansiorakennetta. Beaver application -nimi on johdettu beam-forming-teoriasta, jota kutsutaan sanalla beamer ja se taas muistuttaa beaver-sanaa. (Hoglund 2019.)

3 RF L1low -ohjelmistokehitysympäristön teknologiat

3.1 Docker ja LinSEE-palvelin

Docker on olennaisessa roolissa työskenneltäessä RF L1low -ohjelmiston kanssa. Docker on työkalu, jonka avulla kehitetyt sovellukset voidaan julkaista niin, että ne toimivat useilla laitteilla. Työkalun toiminnallisuus perustuu ohjeeseen nimeltä Docker image, jolla käynnistetään Docker container. Docker container taas sisältää yksittäisen ohjelmiston lähdekoodin, riippuvuudet ja kaiken muun tarvittavan, jotta ohjelmisto toimii toisella laitteella. Docker container on hyödyllinen tilanteessa, jossa ohjelmisto halutaan jakaa toiselle ohjelmistokehittäjälle. Tällöin ei tarvitse huolehtia siitä, että kahdella laitteella olisi käytössä täysin samat ominaisuudet, asetukset tai asennukset, joiden eroavaisuudet voivat vaikuttaa ohjelmiston toimivuuteen. Työkalun avulla voidaan siis helposti jakaa ohjelmisto muille työntekijöille kehitettäväksi, jolloin työntekijät saavat samalla tavalla toimivan ohjelmiston. (Docker overview; What is a Container? A standardized unit of software.)

Nokia Oyj:llä käytetään palvelinta nimeltä LinSEE, joka mahdollistaa virtuaalisen etätyöskentelyn (Louargant 2018). LinSEE on lyhenne sanoista Linux Software Engineering Environment ja se tarjoaa nimensä mukaisesti Linux-pohjaisen ohjelmistoympäristön. Jotta tietokoneella saadaan etäyhteys Linux-käyttöjärjestelmään, on SSH-työkalulla kirjauduttava sisään LinSEE-palvelimelle. (LinSEE User Guide 2016.) Kyseiselle palvelimelle on tehtävä kopio RF L1low -repositoriosta, joka pitää sisällään Docker-tiedoston, joka on puolestaan ohje Docker imagen pystyttämiseksi. Docker imagen avulla käynnistetään Docker container, joka pitää sisällään Linux-käyttöjärjestelmän ja CLion-ohjelmistokehitysympäristön. Sen jälkeen, kun Docker container on käynnistetty, tarvitaan VNC Viewer -ohjelmisto, jotta RF L1low -ohjelmistoa voidaan kehittää. VNC Viewer -ohjelmiston avulla saadaan tietokoneelle näkymä LinSEE-palvelimella olevasta Linux-käyttöjärjestelmästä. (Arvio 2021.) Tämän virtuaalisen etätyöskentelyn käyttöä perustellaan helppokäyttöisyydellä, alhaisella viiveellä ja yksinkertaisella asennusprosessilla. Aiemmin RF L1low -ohjelmiston kehittäminen tapahtui pilvipalveluympäristössä, mutta siitä luovuttiin erilaisten ongelmien takia, joihin kuului muun muassa liian suuri viive. (Väisänen 2019.)

3.2 SSH

SSH, eli Secure Shell, on protokolla, joka mahdollistaa kahden laitteen välisen salatun kommunikoinnin. SSH:sta puhuttaessa tarkoitetaan nimenomaan verkossa tapahtuvaa laitteiden kommunikointia. Tavallisin tilanne on se, että lokaalilla tietokoneella halutaan lähettää tietoa palvelimelle ja vastaanottaa palvelimelta tietoa. SSH:n hyötynä on, että sen avulla tiedot salataan vaikeasti ymmärrettäväksi koodiksi, jota ei pystytä kääntämään ilman siihen

kuuluvia valtuuksia. Tällöin tiedot pysyvät vain niillä henkilöillä, joille ne kuuluvat. SSH:n toiminta perustuu avaimiin, joiden avulla salattu tieto saadaan muotoon, jota ihmiset ymmärtävät. SSH-työkalun avulla generoidaan kaksi kappaletta avaimia, joiden nimet ovat public key ja private key. Public key -avain annetaan palvelimelle, johon halutaan olla yhteydessä, ja Private key -avain pidetään omalla tietokoneella. Kun tietokoneelta lähetetään tietoa palvelimelle, tieto on salattua ja salauksen saa pois ainoastaan generoiduilla avaimilla. (Automation Step by Step 2020.)

Seuraavan esimerkin avulla voidaan helpoiten ymmärtää, kuinka SSH toimii. Kuvitellaan, että tietokoneelta lähetettävä tärkeä tieto on laatikon sisällä. Laatikko lukitaan lukolla, jonka saa auki vain ja ainoastaan Private key -avaimella. Kun palvelin vastaanottaa laatikon, se ei saa sitä auki, koska sillä on Public key -avain, joka on väärä laatikon lukkoon. Palvelimella laatikkoon asetetaan toinen lukko, jonka saa auki Public key -avaimella. Seuraavaksi palvelin lähettää laatikon takaisin tietokoneelle, josta laatikko on alun perin lähtenyt. Seuraavaksi tietokone avaa asettamansa lukon Private key -avaimella ja lähettää laatikon uudelleen palvelimelle. Nyt palvelin saa tiedon itselleen laatikon sisältä avaamalla oman lukkonsa Public key -avaimella. Näin tieto on siirtynyt tietokoneelta palvelimelle salattuna koko ajan, eikä kenelläkään ulkopuolisella ole ollut mahdollisuutta päästä tietoon käsiksi. Samalla periaatteella toimittaisiin myös tilanteessa, jossa palvelimelta haluttaisiin siirtää tietoa tietokoneelle. (Automation Step by Step 2020.)

3.3 CLion-ohjelmistokehitysympäristö ja Gerrit-koodinarviointipalvelu

RF L1low -ohjelmistoa kehitetään CLion-ohjelmistokehitysympäristön avulla. CLion on JetBrains nimisen yhtiön kehittämä ohjelmisto, joka on tarkoitettu C++-ohjelmointikielen kirjoittamiseen, lukemiseen ja muokkaamiseen (JetBrains). RF L1low -ohjelmisto on kehitetty C++-ohjelmointikielillä, jonka osaamista tarvitaan, jotta työskentely on mahdollista. CLion-ohjelmistokehitysympäristön asentamisesta ei tarvitse huolehtia, vaan se on valmiiksi asennettu Docker containerissa sijaitsevassa Linux-käyttöjärjestelmässä.

Kirjoitetun koodin arviointiin käytetään palvelua nimeltään Gerrit. Kyseinen palvelu on tarkoitettu käyttäjille, jotka työskentelevät ryhmänä ohjelmistoprojektin parissa. Palvelussa ryhmän jäsenet voivat arvioida toistensa muutoksia, jotka on tehty ohjelmistoon. (Charles 2020.) Kun RF L1low -ohjelmistoon on valmistunut uusi ominaisuus tai tehty jokin korjaus, kirjoitettu koodi on arvioitava. Nokia Oyj:llä on käytössä standardi, jonka mukaan koodin täytyy noudattaa tiettyjä ehtoja. Ohjelmistoon tehdyt muutokset lisätään Gerrit-palveluun Git-versionhallintatyökalulla käyttämällä Git-komentoja. Gitin avulla pysytään ajan tasalla siitä, milloin muutokset on tehty, millaisia ne ovat olleet, kuka ne on tehnyt, miksi ne on tehty ja mitä tiedostoja on muutettu (GitHub Training & Guides 2013). Gitin toiminta perustuu

siihen, että se tallentaa tiedoston kokonaan, kun se lähetetään versionhallintaan ensimmäistä kertaa. Tämän jälkeen koko tiedoston sisältöä ei tallenneta toista kertaa, vaan pelkästään tehdyt muutokset. Tällöin versionhallintaan ei kerry samasta tiedostosta useita eri versioita, vaan yhteen tiedostoon lisätään aina uudet muutokset. (Arvio 2021.)

4 RF L1low -ohjelmistokehitysympäristön käyttöönoton vaiheet

4.1 Tietokoneen käyttöönotto

Ennen kuin RF L1low -ohjelmiston kehittäminen voidaan aloittaa, tietokoneelle täytyy asentaa Microsoft Windows 10 -käyttöjärjestelmä. Nokia Oyj tilaa jokaiselle uudelle työntekijälle kannettavan tietokoneen ja käyttöjärjestelmän. Huomioitavaa Microsoft Windows 10 -käyttöjärjestelmän asentamisessa on se, että toimenpiteen voi tehdä vain Nokia Oyj:n Oulun toimipisteellä. (Laihiainen 2021.) Kun käyttöjärjestelmä on asennettu tietokoneelle ja työntekijä kirjautuu sisään ensimmäistä kertaa, tietokoneen kovalevy kryptataan eli salataan. Toimenpide käynnistyy automaattisesti. Kun kovalevyn kryptaus on suoritettu, lukon kuva ilmestyy kovalevykuvakkeen päälle, joka löytää resurssienhallinnasta Tämä tietokone -sivulta. (Savusalo 2021.)

Kryptaamisella viitataan tiedon salaamiseen. Kun tietokoneen kovalevy kryptataan, sen sisältämä tieto käännetään vaikeasti ymmärrettäväksi koodiksi. Vain ihmiset ja järjestelmät, joilla on pääsyoikeus kovalevyn sisältöön, voivat lukea tietoa niin, että se on ymmärrettävää. (Pekkarinen 2020.) Kovalevyn kryptaamisella siis suojataan tietoa, jonka ei haluta joutuvan väärin käsiin. Usein työtietokoneet sisältävät yrityksen olennaisia liikesalaisuuksia ja tärkeitä tietoja, jotka on pidettävä salassa. (Record Nations.) Tavallisimmin ajatellaan, että tietovuodot johtuvat murtautujasta, joka on päässyt käsiksi tietojärjestelmään, mutta tiedot voivat vuotaa myös huolimattoman tietokoneen suojauksen takia (Grand Canyon University 2019). Tutkimusten mukaan joka 53. sekunti varastetaan kannettava tietokone. Laitetta suojaava salasana tai pääsykoodi on hyvä olla käytössä, mutta ne eivät kuitenkaan ole takuuvarmoja keinoja pitämään tiedot salassa. (Olenski 2017.) Nokia Oyj:ltä saadun tietokoneen kovalevyn kryptaaminen on siis keino, jolla kovalevyn sisältämät tiedot pidetään tiedossa vain niillä henkilöillä, joille ne kuuluvat.

4.2 SSH-työkalun asentaminen ja kirjautuminen LinSEE-palvelimelle

Jotta etäyhteys voidaan muodostaa Linux-käyttöjärjestelmään ja käyttää CLion-ohjelmistokehitysympäristöä RF L1low -ohjelmiston kehittämiseen, tarvitaan SSH-työkalu, jolla kirjaututaan sisään LinSEE-palvelimelle (Arvio 2021). RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaassa (Liite 1) on neuvottu, kuinka SSH-työkalu nimeltään MobaXterm asennetaan, mutta muillakin SSH-työkaluilla voidaan kirjautua sisään LinSEE-palvelimelle. MobaXtermin käyttöä kuitenkin suositellaan sen graafisen käyttöliittymän takia.

Kun MobaXterm SSH-työkalu on asennettu tietokoneelle, voidaan kirjautua sisään LinSEE-palvelimelle. SSH-työkalussa on määriteltävä LinSEE-palvelin, johon ollaan yhteydessä.

Yksi Nokia Oyj:n Oulun toimipisteellä käytettävistä LinSEE-palvelimista on oulinb114.emea.nsn-net.net. Jos tälle LinSEE-palvelimelle ei pysty kirjautumaan sisään, palvelin saattaa olla poissa käytöstä. Verkko-osoitteessa <https://confluence.ext.net-nokia.com/display/LinSEE/Oulu+LinSEE+servers> on määritelty LinSEE-palvelimet, joita Nokia Oyj:n Oulun toimipisteellä käytetään. (Impola 2021.) Vain Nokia Oyj:n työntekijöillä on pääsy äsken mainittuun verkko-osoitteeseen.

4.3 RF L1low -repositorion kloonaminen ja käyttäjätietojen päivittäminen

Tavallisesti repositorio kloonataan eli kopioidaan lokaalille tietokoneelle Git-komennolla `git clone` (Chacon & Straub 2014, 27). RF L1low -ohjelmiston repositorio kuitenkin kloonataan LinSEE-palvelimelle Gerrit-palvelusta (Impola 2021). Repositorio on kansio, joka pitää sisällään ohjelmiston ja siihen liittyvät olennaiset tiedostot. Jos ohjelmiston kehittämisessä käytetään Git-versionhallintatyökalua, repositorio sisältää silloin myös `.git`-kansion, jonka tehtävänä on tallentaa ohjelmiston historia aina, kun ohjelmisto muuttuu. (Bilde 2016.) Git-versionhallintaa käyttävät ohjelmistoprojektien repositoriot sijaitsevat tavallisimmin palveluissa, jotka tarjoavat paikan repositorioiden säilyttämiseen.

RF L1low -repositorion kloonamisen jälkeen on päivitettävä omat käyttäjätiedot `user.conf`-tiedostoon. Tiedot päivitetään tunnistamista varten ja näin pystytään yksilöimään Nokia Oyj:n Oulun toimipisteellä työskentelevä työntekijä, että hänellä on käytössään Docker container. (Arvio 2021.)

4.4 SSH-avaimien ja JFrog Artifactory API -avaimien generointi

SSH-avaimien generoimisen tarkoituksena on saada pääsyoikeus Docker containeriin, joka pitää sisällään Linux-ympäristön, jossa RF L1low -ohjelmistoa kehitetään. SSH-avaimien generointiprosessissa luodaan julkinen ja yksityinen SSH-avain sekä `.ssh`-niminen kansio. SSH-avaimet generoituvat kotikansion sisälle ja ne on siirrettävä `.ssh`-kansion sisälle Linux-komennon avulla. Yksityinen SSH-avain pidetään tallessa omalla tietokoneella ja julkinen SSH-avain lisätään Gerrit-koodinarviointipalveluun, GitLab-versionhallintapalveluun ja JFrog Artifactory -palveluun. Julkisen SSH-avaimen lisäämisen myötä pystytään tunnistautumaan ja todentamaan käyttöoikeus äsken lueteltuihin palveluihin (Arvio 2021.)

Nokia Oyj:llä on käytössä kaksi GitLab-versionhallintapalvelua, mutta kumpaakaan niistä ei käytetä RF L1low -ohjelmiston versionhallintaan. Kyseiset GitLab-versionhallintapalvelut kuitenkin sisältävät ohjelmistokomponentteja, jotka liittyvät muihin Nokia Oyj:llä kehitettäviin ohjelmistoihin. Työntekijä saattaa siirtyä kehittämään jotakin toista ohjelmistoa uransa aikana Nokia Oyj:llä ja tästä syystä julkinen SSH-avain on hyvä olla asetettuna valmiiksi

molempiin GitLab-versionhallintapalveluihin. Julkinen SSH-avain on lisättävä kahteen GitLab-versionhallintapalveluun myös sen takia, jotta RF L1low -projekti voidaan alustaa valmiiksi kehitystä varten. (Arvio 2021.)

Julkinen SSH-avain täytyy käydä lisäämässä myös JFrog Artifactory -palveluun. Kyseiseen palveluun tallentuvat binääritiedostot, joita ohjelmistokehityksen aikana syntyy, eikä niiden haluta päätyvän Gerrit-koodinarviointipalveluun. Myös JFrog Artifactory -palveluita on Nokia Oyj:llä käytössä kaksi kappaletta ja tämä johtuu suurten binääritiedostojen lukumäärästä. (Arvio 2021.)

JFrog Artifactory -palveluissa generoidaan myös JFrog Artifactory API -avaimet. Näiden avaimien tarkoituksena on tunnistaa työntekijä oikeutetuksi palvelun sisältämiin riippuvuuksiin, joita tarvitaan RF L1low -projektin alustamiseen. JFrog Artifactory API -avaimien generoimisen jälkeen, avaimet on siirrettävä LinSEE-palvelimelle omissa tiedostoissa SSH-työkalun avulla. Kun Docker container käynnistetään ensimmäistä kertaa, API-avaimet kopioidaan LinSEE-palvelimelta Docker containerin sisälle. (Arvio 2021.)

4.5 Docker containerin käynnistäminen ja RF L1low -repositorion tarkistaminen

Kun RF L1low -repositorio on kloonattu LinSEE-palvelimelle, lisätty omat käyttäjätiedot user.conf-tiedostoon, generoitu SSH-avaimet ja ne on siirretty oikean kansion sisälle, lisätty julkinen SSH-avain Gerrit-, GitLab- ja JFrog Artifactory -palveluun sekä generoitu JFrog Artifactory API -avaimet ja ne on lisätty omiin tiedostoihin, voidaan käynnistää Docker container. On tärkeää, että kaikki äsken mainitut toimenpiteet on suoritettu, ennen kuin Docker container käynnistetään. Jos kaikkia toimenpiteitä ei ole suoritettu, Docker containerin käynnistämisessä ilmenee ongelmia, eikä se toimi halutulla tavalla. (Arvio 2021.)

Docker containerin käynnistämisen jälkeen on saatava Windows-tietokoneelle näkymä Linux-käyttöjärjestelmästä. Tämä onnistuu ohjelmistolla nimeltä VNC Viewer. Ohjelmisto ladataan Software Center -ohjelmistojakopalvelusta, joka löytyy valmiiksi asennettuna Nokia Oyj:ltä saadulta tietokoneelta. (Impola 2021.)

Kun VNC Viewer -ohjelmistolla on saatu Windows-tietokoneen näytölle näkymä Linux-työpöydästä, on tarkistettava, että RF L1low -repositorio on kloonautunut Docker containeriin. Kun Docker container käynnistetään ensimmäistä kertaa, on mahdollista, että RF L1low -repositorion kopioiminen epäonnistuu. Tämän takia on tarkistettava, että Linux-käyttöjärjestelmän tiedostonhallinnasta löytyy workspace-levyke. Jos levykkeen sisältä löytyy kansio nimeltä l1low ja tämän kansion sisällä on alikansioita, RF L1low -repositorion kopioiminen on onnistunut. (Impola 2021.) Tavallisin syy RF L1low -repositorion kopioinnin

epäonnistumiselle on epävakaata verkkoyhteys tai jonkin ohjelmistokehitysympäristön käyttöönottoprosessin toimenpiteen suorittaminen on ollut puutteellinen (Arvio 2021).

4.6 Clion-ohjelmistokehitysympäristön konfigurointi

CLion-ohjelmistokehitysympäristössä kirjoitetaan, käännetään ja testataan kehitettävän RF L1low -ohjelmiston koodi. Jotta CLion voi käänntää ja testata RF L1low -ohjelmiston, on tehtävä tietyt konfiguraatiot, johon kuuluu muun muassa RF L1low -projektin alustaminen. Lisäksi RF L1low -projektin juurikansio täytyy vaihtaa. Kun projekti avataan CMakeLists.txt -tekstitiedoston kautta, juurikansioksi valikoituu automaattisesti kansio nimeltä beaver-application. Tämä on kuitenkin väärä kansio juurikansioksi ja se täytyy vaihtaa. (Laihiainen 2021.) CMakeLists.txt -tekstitiedosto sisältää kuvauksen projektin lähdetiedoista (CMakeLists.txt 2021).

RF L1low -projektin alustamisen jälkeen, CLion-ohjelmistokehitysympäristöön on tuotava asetuksia kansioista nimeltä settings.zip. Kyseinen kansio sisältää muun muassa asetukset yksikkötestien ajamista varten, työkalun koodin optimointia varten ja työkalun merge-konfliktien selvittämiseen (Arvio 2021). Viimeinen CLion-ohjelmistokehitysympäristön konfiguraatiota koskeva toimenpide on CMake-asetusten päivittäminen, joka tapahtuu luomalla kaksi uutta asetusprofiilia. CMake on työkalu, jota käytetään C++-ohjelmointikielellä kirjoitetun lähdekoodin kääntämiseen (CMake Reference Documentation).

Kun lopulta on suoritettu kaikki Clion-ohjelmistokehitysympäristön konfigurointiin liittyvät toimenpiteet, ollaan valmiita kehittämään uutta ominaisuutta RF L1low -ohjelmistoon. Ohjeet konfigurointien ja muidenkin tarvittavien toimenpiteiden suorittamiseen löytyvät RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaasta (Liite 1).

5 RF L1low -ohjelmiston kehitysprosessi

5.1 Haarat eli branches

Ohjelmistokehityksessä käytetään haaroja, joiden englanninkielinen vastine on branches. Haara voidaan ajatella lineaariseksi viivaksi, joka sisältää yksittäisiä vaiheita ohjelmiston historiasta. Ohjelmiston kehittyessä tehdyt muutokset tallentuvat haaraan, ja tiettyyn muutokseen voidaan palata jälkepäin. Master-haara, usein myös pelkästään master, on haara, joka sisältää kaikki ohjelmistoon lisätyt ominaisuudet. Master-haarasta voi haarautua uusi haara, jolloin siihen kopioituu täysin sama ohjelmisto, joka on master-haarassa juuri sillä hetkellä, kun haara luodaan. Haaroja kannattaa käyttää tilanteessa, jossa halutaan lisätä ohjelmistoon jokin tietty toiminnallisuus, mutta ei olla täysin varmoja siitä, säilytetäänkö se pysyvästi. Master-haarasta erillisessä haarassa on turvallista kokeilla uusia ideoita, koska muutokset eivät vaikuta millään tavalla master-haaraan. Jos kävisi niin, että ohjelmisto lakkaisi toimimasta haarassa, joka on luotu uutta toiminnallisuutta varten, eikä sitä saataisiin korjattua, haara voidaan poistaa. Tämän jälkeen palataan takaisin master-haarassa sijaitsevaan ohjelmistoon, luodaan uusi haara ja yritetään tehdä toiminnallisuus uudestaan. Ennen haaran poistamista kannattaa uudet ominaisuudet tallentaa esimerkiksi tekstitiedostoon. Tällöin ei tarvitse tehdä samaa työtä uudestaan. Jos haaraa poistetaan ennen master-haaraan yhdistämistä, menetetään kaikki uudet muutokset, jotka on tehty. (The Coding Train 2016; TheCodex 2019.)

Jos taas haarassa luotu uusi ominaisuus toimii odotetulla tavalla, eikä se aiheuta ongelmia ohjelmiston toimivuuteen, haara voidaan yhdistää master-haaraan. Yhdistäminen, englanninkielinen vastine merge, on prosessi, jossa haarassa tehdyt muutokset tulevat osaksi master-haarassa sijaitsevaa ohjelmistoa. Kun haara on yhdistetty onnistuneesti, se voidaan poistaa. Tämän jälkeen voidaan tehdä taas uutta ominaisuutta uudessa haarassa. (The Coding Train 2016; TheCodex 2019.)

Myös RF L1low -ohjelmiston kehityksessä uudet ominaisuudet on tehtävä omiin haaroihin, koska tällöin uusia ominaisuuksia valmistuu nopealla aikataululla. Jos työntekijä lähettää joka kerta kirjoittamansa koodin suoraan master-haarasta Gerrit-palveluun arvioitavaksi, työstetyt ominaisuudet eivät päädy missään vaiheessa osaksi RF L1low -ohjelmistoa. Tämä johtuu siitä että, arviointiprosessi alkaa aina alusta, kun koodia lähetetään Gerrit-palveluun. Vaihtoehtona olisi jäädä odottamaan, että koodi arvioidaan hyväksytyksi, mutta arviointi vie aikaa. Ratkaisu tähän on se, että jokaiselle ominaisuudelle tehdään oma haaransa. Kun yksi ominaisuus on arviointiprosessissa, sillä välin voidaan työstää toista ominaisuutta toisessa haarassa. Haarojen käyttämisestä on myös se hyöty, että muutosten välille ei synny

riippuvuuksia, jotka voisivat estää muutosten yhdistämisen RF L1low -ohjelmistoon. (User Guide.)

5.2 Koodin testaaminen

Kun haarassa ei ole tarkoitus kirjoittaa enempää koodia, sen toimivuus täytyy testata ja arvioida, ennen kuin se yhdistetään master-haaraan. Koodin toimivuutta testataan jo valmiiksi rakennetuilla yksikkötesteillä ja automaatiotesteillä. Yksikkötestit ajetaan syöttämällä komento `./l1low_execute.sh ut-x86 [tuotteen nimi]` CLionissa olevaan terminaaliin, jonka jälkeen saadaan ilmoitus testin tuloksesta. Jos testi ei mene läpi, koodissa on toiminnan estäviä virheitä, jotka on korjattava. Läpäistyään yksikkötestit koodi voidaan lähettää arvioitavaksi Gerrit-palveluun Git-versionhallintatyökalun avulla. Komennossa `[tuotteen nimi]` tilalle on vaihdettava se RF L1low -ohjelmiston tuote, jolle ohjelmistoprojekti on alustettu. RF L1low -ohjelmisto koostuu useista eri tuotteista ja projekti täytyy alustaa tuotteelle, jota työntekijä on tullut kehittämään. Työntekijän täytyy keskustella linjamanagerin kanssa, mitä tuotetta kehitetään ja tehdä alustus sen perusteella. (Laihiainen 2021.) Ohjeet RF L1low -projektin alustamiseen löytyvät RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaan (Liite 1) luvusta 2.10.

Ennen arviointia koodi testataan Gerritissä Jenkinsin automaatioprosessissa. Jenkins on integraatiotyökalu, joka mahdollistaa ohjelmiston jatkuvan kehittämisen, testaamisen ja julkaisemisen (Simplilearn 2018). Koodin testaaminen yksikkötesteillä ei ole pakollista, mutta suositeltavaa ajan säästämiseksi. Jenkinsin testit kestävät kauemmin kuin yksikkötestit, joten myös tulosten saamisessa kestää kauemmin. Jos koodissa sattuu olemaan virheitä, niitä ei päästä korjaamaan yhtä nopeasti, kuin päästäisiin ajettaessa yksikkötestit lokaalisti omalla tietokoneella. Kun muutokset ovat lopulta läpäisseet myös Jenkinsin testit, siirrytään koodin arviointiprosessiin. (Laihiainen 2021.)

5.3 Koodin arviointiprosessi

Kuviossa 1 on kuvattuna RF L1low -ohjelmiston koodin arviointiprosessi. Vasemmassa reunassa oleva hahmo esittää työntekijää, joka lähettää RF L1low -ohjelmistoon tehdyt muutokset Gerrit-koodinarviointipalveluun. Tämän jälkeen käynnistyy Jenkins-testi, joka testaa ohjelmiston toimivuuden. Jos testissä havaitaan virheitä, arviointisivulle ilmestyy -1-merkinä. Tällöin on palattava korjaamaan virheet ja lähetettävä korjaukset Gerritiin. (Elo 2021.)

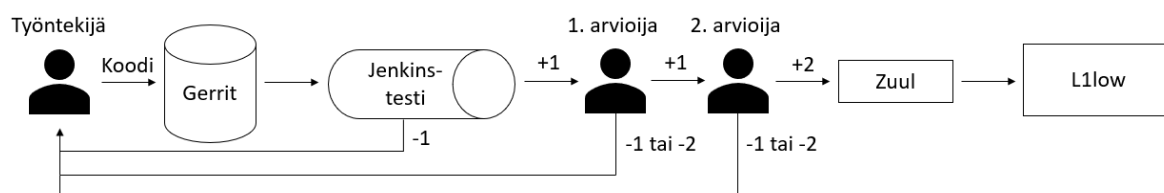
Kun Jenkins-testi on lopulta hyväksytysti läpi, koodi saa testin suorittamisesta +1-merkin. Tämän jälkeen kahden muun RF L1low -ohjelmistoa kehittävän työntekijän on arvioitava, että koodi on standardien mukaista. Gerrit lisää 2–3 koodinarvioijaa automaattisesti,

mutta heitä voi käydä lisäämässä useamman. Ei ole taattua, että Gerritin lisäämät arvioijat ehtisivät katsomaan muutoksia, joten arviointiprosessia voidaan nopeuttaa lisäämällä enemmän henkilöitä tarkastelemaan koodia. Työntekijöiden sähköpostiosoitteen lisääminen arviointisivulle lähettää kutsun arviointia varten. Myös työntekijät, joita ei ole kutsuttu sähköpostilla arvioijiksi, voivat käydä antamassa arviointimerkinnän. (Elo 2021.)

Jos ensimmäisen arvioijan mielestä koodi ei ole standardien mukaista, hän antaa tehdylle muutokselle -1- tai -2-merkinnän sen perusteella, kuinka paljon koodi eroaa Nokia Oyj:llä noudatettavasta koodistandardista. Miinusmerkkisen arvioinnin jälkeen on tehtävä tarvittavat muutokset, jotta koodi on tavoitellun mukaista. Jos ensimmäisen arvioijan mielestä koodissa ei olisikaan ollut mitään korjattavaa, antaisi hän sille +1-merkinnän. (Elo 2021.)

Ensimmäisen arvioijan annettua +1-merkinnän työntekijän kirjoittamalle koodille, on vuorossa toisen arvioijan arviointi. Menetelmä on lähes sama kuin ensimmäisellä arvioijalla. Jos koodi ei ole toisen arvioijan mielestä hyväksyttävää, hän antaa sille miinusmerkkisen arvioinnin. Koodi saa +2-merkinnän, jos myös toinen arvioija on sitä mieltä, ettei koodissa ole mitään korjattavaa. (Elo 2021.)

Kun lopulta ollaan siinä pisteessä, että koodi on saanut Jenkins-testistä +1-merkinnän, ensimmäinen arvioija on antanut +1-merkinnän ja toinen arvioija +2-merkinnän, Zuul niminen palvelu yhdistää muutokset automaattisesti osaksi RF L1low -ohjelmistoa. (Elo 2021.) Koodin arviointisivulta voi tarkistaa, ovatko muutokset päätyneet ohjelmistoon. Jos sivun vasemmassa ylänurkassa on pieni vihreä laatikko, jonka sisällä on Merged-teksti, muutokset ovat yhdistetty ohjelmistoon. Sininen laatikko, jonka sisällä on Active-teksti, tarkoittaa sitä, että koodin arviointi tai yhdistäminen on vielä kesken.



Kuvio 1. RF L1low -ohjelmiston koodin arviointiprosessi (mukailtu Elo 2021)

6 Koodin lähettäminen arvioitavaksi

6.1 Tarvittavat Git-komennot

Kirjoitettu koodi on arvioitava Gerrit-koodinarviointipalvelussa, ennen kuin se voidaan yhdistää RF L1low -ohjelmistoon. Jotta koodi voidaan lähettää Gerrit-palveluun, on syötettävä Git-komentoja terminaaliin, joka on CLion-ohjelmistossa. Taulukossa 1 on esitelty Git-komennot, jotka tarvitaan koodin lähettämiseksi Gerrit-palveluun.

Git-komento	Selite
git add <tiedoston nimi>	Komento lisää yhden muokatun tiedoston lähetettäväksi versionhallintaan, kun <tiedoston nimi> tilalle vaihdetaan tiedosto, jota on muokattu (Atlassian Plc 2017). Vaihtoehtoisesti komennolla git add -u voidaan lisätä kerralla kaikki muokatut tiedostot lähetettäväksi versionhallintaan (Laihiainen 2021).
git commit	Komento luo ohjelmistosta ikään kuin virsanpylvään, josta ilmenee ohjelmiston sen hetkinen tila (Git commit). Komennon hyötyinä on se, että voidaan palata taaksepäin tarkistamaan, miten ohjelmisto on muuttunut sen olemassaolon aikana.
git push origin HEAD:refs/for/master	Komento lähettää muutokset arvioitavaksi Gerritiin. Se on lähes sama kuin tavallinen git push -komento, mutta erona on se, että kohdehaaran eteen tulee etuliite refs/for/. Ilman etuliitettä koodi ei päädy Gerrit-palveluun. (Kempin.) Komennon perään voidaan kirjoittaa %wip, joka kertoo Gerritissä arvioijalle, että ominaisuuden tekeminen on vielä kesken. (Elo 2021). Tällöin arviointisivun vasempaan ylänurkkaan ilmestyy

	harmaa laatikko, jonka sisällä on Work in Progress -teksti. Kun ominaisuus on lopulta valmistunut, se pitää käydä Gerritissä merkkäämassa valmiiksi arviointia varten.
--	--

Taulukko 1. Olennaisimmat Git-komennot

Kun RF L1low -ohjelmiston uuden ominaisuuden koodia lähetetään Gerrit-palveluun ensimmäistä kertaa, käytetään git commit -komentoa. Jos ominaisuutta ei saada valmiiksi yhdellä kerralla, täytyy muilla kerroilla käyttää komentoa git commit --amend, kun samasta haarasta lähetetään koodia useamman kuin yhden kerran Gerritiin. Sen lisäksi, että komento git commit tekee ohjelmistosta uuden virstanpylvään, samalla Gerritiin luodaan uusi arviointisivu tehtyjä muutoksia varten. Jos git commit -komento suoritetaan joka kerta, kun koodia lähetetään Gerritiin samasta haarasta, lähetetyistä muutoksista generoituu omat arviointisivunsa. Tarkoituksena on, että uuden ominaisuuden kaikki koodi on yhdellä samalla arviointisivulla. Tämä onnistuu käyttämällä git commit --amend -komentoa, joka lisää muutokset yhdelle samalle arviointisivulle sen sijaan, että jokaiselle lähetetylle muutokselle tehtäisiin omat arviointisivut. Komento git commit --amend siis säilyttää kaikki tehdyt muutokset samassa haarassa. (Elo 2021.)

Aiemmin mainituilla Git-komennoilla voidaan lähettää koodia Gerrit-koodinarviointipalveluun. RF L1low -ohjelmiston kehityksessä tarvitaan myös muita Git-komentoja, jotka helpottavat työskentelyä. Samoja komentoja voi käyttää kaikissa versionhallintaa vaativissa ohjelmistokehitysprojekteissa, joten Git-komentojen osaaminen on yleisesti hyödyllistä. Taulukossa 2 on listattuna Git-komentoja, joiden avulla versionhallintaa voidaan käyttää sujuvasti.

Git-komento	Selite
git status	Komento listaa kaikki muutoksia sisältävät tiedostot, joita ei ole vielä valmisteltu lähetettäväksi versionhallintaan git add -komentolla (Atlassian Plc 2017).
git pull	Komento hakee repositoriosta ohjelmiston viimeisimmän version ja yhdistää sen

	<p>automaattisesti kloonattuun ohjelmistoon, joka on lokaalilla tietokoneella (Chacon & Straub 2014, 54).</p>
<p>git checkout <haaran nimi></p>	<p>Komento vaihtaa yhdestä haarasta toiseen. Haara, johon halutaan vaihtaa, on oltava olemassa, jotta komennon suorittaminen onnistuu. (Chacon & Straub 2014, 68.) Komennossa <haaran nimi> tilalle on vaihdettava se haaran nimi, johon halutaan vaihtaa.</p>
<p>git checkout -b <haaran nimi></p>	<p>Komento luo uuden haaran ja samalla vaihtaa pois vanhasta haarasta uuteen luotuun haaraan (Chacon & Straub 2014, 72). Komennossa <haaran nimi> tilalle on syötettävä haaran nimi, joka halutaan luoda.</p>
<p>git checkout master <tiedostopolku></p>	<p>Komento hakee yksittäisen tiedoston master-haarasta. Komento on hyödyllinen tilanteissa, jossa useita tiedostoja on muokattu, mutta yksittäinen tiedosto halutaan lopulta palauttaa alkuperäiseen muotoon. Tällöin ei tarvitse muistaa palautettavan tiedoston alkuperäistä sisältöä, vaan se voidaan hakea master-haarasta. Komennossa <tiedostopolku> tilalle on vaihdettava palautettavan tiedoston sijainti tietokoneessa.</p>
<p>git rebase <haaran nimi></p>	<p>Komento lisää yhdessä haarassa tehdyt muutokset toiseen haaraan. Esimerkiksi, kun haarassa, nimeltään Malli-haara, suoritetaan komento git rebase master, master-haaran lisätyt uusimmat ominaisuudet tulevat osaksi Malli-haaraa. Komennossa <haaran nimi> on vaihdettava sen haaran nimeen, josta halutaan tuoda uusimmat</p>

	muutokset haaraan, jossa käyttäjä on sillä hetkellä. (Chacon & Straub 2014, 98.)
git branch -d <haaran nimi>	Komento poistaa haaran, joka on yhdistetty master-haaraan (Chacon & Straub 2014, 76). Jos haara halutaan poistaa ennen kuin se on yhdistetty master-haaraan, on suoritettava komento git branch -D <haaran nimi> (Chacon & Straub 2014, 82). Komenossa <haaran nimi> tilalle on vaihdettava haaran nimi, joka halutaan poistaa. Huomioitavaa RF L1low -ohjelmiston kehitystä varten tehtyjen haarojen poistamisessa on se, että ne täytyy poistaa aina käyttämällä poistokomentoa, joka sisältää ison D-kirjaimen. Koska haarassa tehdyt muutokset yhdistetään automaattisesti master-haaraan Zuul-palvelun avulla, lokaali Git-versionhallinta tietokoneella ei ymmärrä, että haara on jo yhdistetty. Tästä syystä Git ei salli haaran poistamista poistokomennolla, joka sisältää pienen d-kirjaimen. (Elo 2021.)
git branch	Komento listaa nykyiset haarat allekkain ja näyttää *-merkillä haaran, jossa käyttäjä on sillä hetkellä (Chacon & Straub 2014, 81).

Taulukko 2. Versionhallintaan liittyviä Git-komentoja

6.2 Jira-tehtävienhallintapalvelu

Jira-tehtävienhallintapalvelu on ohjelmistotuote, joka tarjoaa työryhmille alustan, jossa voidaan hallinnoida työstettävää projektia. Jira-tehtävienhallintapalvelu mahdollistaa projektin suunnittelun, seurannan ja tehtävien määrittämisen tietyille ryhmänjäsenille. (Guide 1: Get started with Jira Software.)

RF L1low -ohjelmistoon tehtävät uudet ominaisuudet on määritetty Jira-tehtävienhallintapalvelussa. Suurin osa tehtävistä määritetään aina tietyille henkilöille, mutta on myös yleisiä

avoimia tehtäviä, joita ei ole erikseen määrätty kenellekään. Tarkoituksena on, että pysytään ajan tasalla työstettävistä ominaisuuksista, jolloin tiedetään, mitä RF L1low -ohjelmistoon valmistuu, millaisella aikataululla, kuka ominaisuuden on tehnyt ja miten ominaisuuden toteuttaminen on edennyt. Näiden tietojen perusteella projektista vastaavat henkilöt osaa- vat jatkossa suunnitella paremmin uuden ominaisuuden kehitysprosessin. Ohjelmistokehit- täjän ei siis tarvitse huolehtia uusien tehtävien lisäämisestä palveluun, sillä tietyt henkilöt organisaatiossa ovat asiasta vastuussa. (Impola 2021.)

Yksittäistä tehtävää Jira-tehtävienhallintapalvelussa kutsutaan englanninkielisellä termillä ticket ja se sisältää tiedot ominaisuudesta, joka täytyisi saada valmiiksi arvioidussa ajassa. Jira-tehtävä yksilöidään merkkijonolla eli Jira-tunnuksella. Kyseinen tunnus on aina laiteta- tava commit-viestissä varattuun paikkaan, jotta tiedetään mihin Jira-tehtävään ominaisuus kuuluu. Työntekijä löytää Nokia Oyj:llä käytettävän Jira-palvelun osoitteesta <https://jiradec.ext.net.nokia.com>. (Impola 2021.)

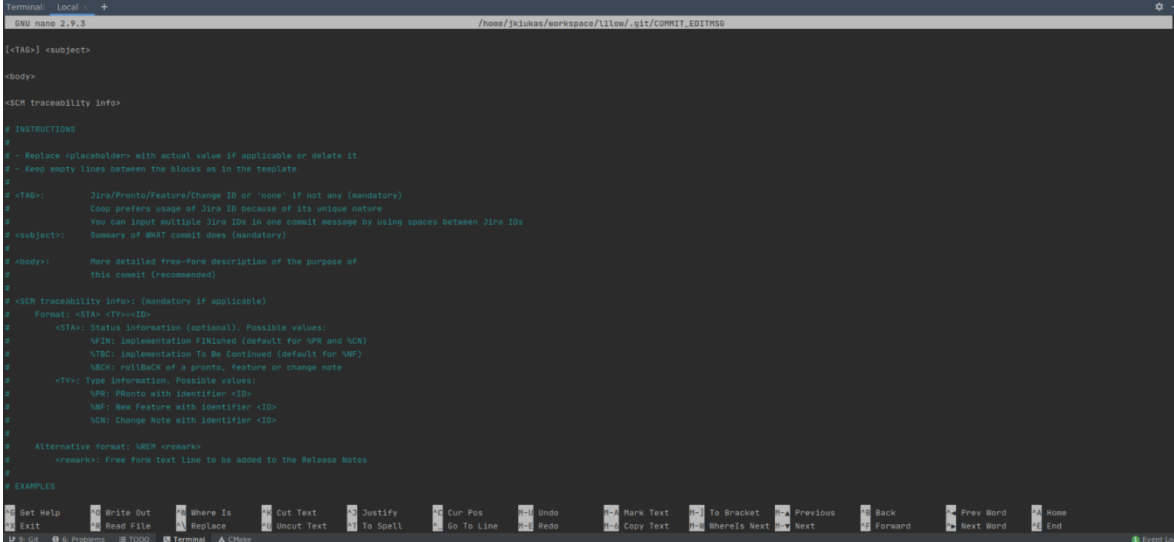
6.3 Commit-viesti

Komentojen git commit ja git commit --amend jälkeen on kirjoitettava commit-viesti. Työs- kentely ohjelmistoprojektissa on ryhmätyötä, joten selkeät commit-viestit tekevät työsken- telystä helpompaa ja tehokkaampaa. Kirjoitetusta viestistä tulisi käydä selkeästi ilmi, millai- sia muutoksia on tehty ja miksi. Tällöin toisen työntekijän on helpompi ymmärtää, mitä aiem- min ohjelmistoon on tehty ja jatkaa työskentelyä pisteestä, johon on jääty. Näin ollen hel- posti ymmärrettävät commit-viestit ovat osoitus omasta yhteistyökyvystä. (Ayodeji 2019.)

Commit-viestin tulisi alkaa yhden rivin mittaisella otsikolla, josta ilmenee tiivistetysti tehdyt muutokset. Otsikon perään ei lisätä pistettä. Tämän jälkeen jätetään yksi tyhjä riviväli, jota seuraa lisätietoa tehdyistä muutoksista. Jos viestin kirjoittamiseen on käytettävä useampaa kuin yhtä kappaletta, ne täytyy erottaa toisistaan tyhjällä rivivälillä. Myös ranskalaisia viivoja voi käyttää viestissä. Koska Nokia Oyj on kansainvälinen yritys, RF L1low -ohjelmistoa on kehittämässä ihmisiä useista eri maista. Tästä syystä commit-viestit täytyy kirjoittaa aina englanniksi, jolloin kenellekään ei jää epäselväksi millaisia muutoksia ohjelmistoon on tehty. (Chacon & Straub 2014, 133; Ayodeji 2019; Laihiainen 2021.)

Kirjoitettuaan komennon git commit Clionin terminaaliin, työntekijän tietokoneen näytölle avautuu kuvan 2 mukainen tekstieditori, johon kirjoitetaan commit-viesti. Hakasulkeiden si- sältä on poistettava <TAG> ja lisättävä tilalle Jira-tehtävän tunnus. Jos tehty muutos ei kuulu mihinkään Jira-tehtävään, hakasulkeiden sisään kirjoitetaan none. Työntekijän täytyy vaihtaa <subject> tilalle commit-viestin otsikko, josta käy lyhyesti ilmi, mitä muutoksessa on tehty. Tarkemmat tiedot muutoksesta kirjoitetaan <body> tilalle. Työntekijä voi poistaa

<SCM traceability info> kokonaan, eikä hänen tarvitse lisätä sen tilalle mitään. Rivit, jotka alkavat #-merkillä, eivät sisälly commit-viestiin. Kyseiset rivit ovat ohjeistusta varten ja niitä ei huomioida, kun commit-viesti tallennetaan. Kun commit-viesti on kirjoitettu valmiiksi, se tallennetaan painamalla näppäinyhdistelmää Ctrl + O ja lopuksi vielä enter-näppäimellä tallennetaan commit-viestin nimi. Tämän jälkeen työntekijä voi painaa näppäinyhdistelmää Ctrl + X, joka sulkee tekstieditorin. Suoritettaessa komento git commit --amend, Jira-tunnusta ja otsikkoa ei tarvitse kirjoittaa uudestaan. Pelkästään commit-viestin sisältö on kirjoitettava uusiksi, jotta tiedetään, millaisia muutoksia ohjelmistoon on tehty. Kun työntekijä on kuvailut tarpeeksi tarkasti tekemänsä muutokset RF L1low -ohjelmistoon, hän voi lähettää kirjoittamansa koodin arvioitavaksi Gerrit-palveluun.



```

Terminal Local +
GNU nano 2.9.3 /home/jkukas/workspace/l1low/.git/COMMIT_EDITMSG

[<TAB>] <subject>
<body>
<SCM traceability info>
# INSTRUCTIONS
#
# - Replace <placeholder> with actual value if applicable or delete it
# - Keep empty lines between the blocks as in the template
#
# <TAB>:      Jira/Pronto/feature/Change ID or 'none' if not any (mandatory)
#
# @@@@ prefers usage of Jira ID because of its unique nature
# You can input multiple Jira IDs in one commit message by using spaces between Jira IDs
# <subject>:  Summary of WHAT commit does (mandatory)
#
# <body>:    More detailed free-form description of the purpose of
#           this commit (recommended)
#
# <SCM traceability info>: (mandatory if applicable)
# Format: <STA> <TR> <ID>
#
# <STA>: Status information (optional). Possible values:
#       *FIN: Implementation FINISHED (default for WIP and WCN)
#       *TBC: Implementation To Be Continued (default for WIP)
#       *WIP: rollback of a pronto, feature or change note
# <TR>: Type information. Possible values:
#       *PR: Pronto with Identifier <ID>
#       *WF: New Feature with Identifier <ID>
#       *CN: Change Note with Identifier <ID>
#
# Alternative format: WCN <pronto>
# <pronto>: Free form text line to be added to the Release Notes
#
# EXAMPLES
#
# [F1] Help      [F2] Write Out  [F3] Where Is  [F4] Cut Text  [F5] Justify   [F6] Cut Pos  [F7] Undo     [F8] Mark Text  [F9] To Bracket [F10] Previous  [F11] Back     [F12] Prev Word [F13] Home
# [Esc] Exit    [Ctrl] Read File [Ctrl] Replace [Ctrl] Uncut Text [Ctrl] To Spell [Ctrl] Go To Line [Ctrl] Redo    [Ctrl] Copy Text [Ctrl] WhereIs Next [Ctrl] Next     [Ctrl] Forward  [Ctrl] Next Word [Ctrl] End
# [Ctrl] O      [Ctrl] Problems  [Ctrl] TOOD    [Ctrl] Terminal [Ctrl] Mouse
# [Ctrl] Q      [Ctrl] Event Log
  
```

Kuva 2. Tekstieditori Clion-ohjelmistossa

7 Palaute RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaasta

RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaan (Liite 1) toimivuutta kokeiltiin elokuussa 2021. Nokia Oyj:n Oulun toimipisteelle saapui kaksi uutta työntekijää Juha Kortesalmi ja Pasi Riikonen. Opinnäytetyö ja sen liite 1 lähetettiin molemmille työntekijöille sähköpostitse ja he ryhtyivät suorittamaan toimenpiteitä, joilla ohjelmistokehitysympäristö on tarkoitus saada käyttöön. Suoritettuaan tarvittavat toimenpiteet, Kortesalmi ja Riikonen saivat tietokoneillensa ohjelmistokehitysympäristön käyttöön ja pystyivät aloittamaan RF L1low -ohjelmiston kehittämisen. Kumpikaan työntekijä ei tarvinnut ulkopuolisen henkilön apua käyttöönottoprosessin aikana. Käyttöönoton jälkeen Kortesalmi ja Riikonen antoivat palautetta oppaasta.

Kortesalmi (2021) sanoo, että vastaavanlaisten järjestelmien käyttöönottaminen ei ole täysin helppoa, mutta verrattaessa aikaisempiin kokemuksiin, RF L1low -ohjelmistokehitysympäristön käyttöönottaminen oli helppoa. Hän kokee RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaan (Liite 1) säästäneen aikaa verrattaessa tilanteeseen, jossa toimenpiteet olisi täytynyt suorittaa omin avuin. Kortesalmi kuvailee RF L1low -ohjelmistokehitysympäristön käyttöönotto-opasta (Liite 1) selkeimmäksi ja parhaimmaksi ohjeeksi, mitä hän on koskaan nähnyt. Kortesalmi kertoo olevansa täysin ammattilainen tietotekniikassa ja hänellä on kokemusta alasta 20-vuotta. Näihin vuosiin kuuluu useita luettuja ohjeita, jotka ovat opastaneet erilaisten järjestelmien käyttöönottamisessa. Kortesalmen aiemmin lukemat oppaat ovat sisältäneet vain ranskalaisia viivoja yksittäisistä tehtävistä, eikä niissä ole selitetty tarkasti suoritettavia toimenpiteitä.

Kortesalmi (2021) arvioi, että RF L1low -ohjelmistokehitysympäristön voi ottaa käyttöön RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaan (Liite 1) avulla sellainen henkilö, joka omaa häneen verrattavissa olevan taustan. Hän mainitsee, ettei tämä kuitenkaan pois sulje sitä, etteikö harjoitteluun astuva henkilö voisi ymmärtää ohjeen sisältöä. Kortesalmi tuo esiin, että uudella työntekijällä olisi hyvä olla jo entuudestaan kokemusta Linux-ympäristöstä. Tietyille henkilöille saattaa olla epäselvää Linuxin käyttäminen, jos heillä ei ole siitä aikaisempaa kokemusta. Kortesalmi perustelee väitettään sillä, että yleisessä käytössä oleva Windows-käyttöjärjestelmä on erilainen verrattuna Linux-käyttöjärjestelmään, jonka käyttäminen perustuu hyvin pitkälti erilaisiin Linux-komentoihin. Kokemattomuus Linuxista ja sen komennoista voivat taas johtaa siihen, että uusi työntekijä joutuu kysymään kokenemmalta työntekijältä useasti apua tai perehtymään Linuxin perusteisiin, jotta hän saa ohjelmistokehitysympäristön käyttöönsä.

Kortesalmi (2021) kommentoi RF L1low -ohjelmistokehitysympäristön käyttöönotto-opasta (Liite 1) sanomalla, että sen rakenne on looginen, millä hän viittaa siihen, että opas etenee

siinä järjestyksessä, jossa toimenpiteet täytyy suorittaa. Hänen mielestään numeroidut toimenpiteet ovat toimiva ratkaisu, sillä ne kertovat lukijalle, että toimenpiteet täytyy suorittaa järjestyksessä vaihe vaiheelta. Kortosalmi mainitsee RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaan (Liite 1) hyväksi puoleksi sen, että uudelle työntekijälle kerrotaan selkeästi mitä täytyy tehdä, ja opinnäytetyön hyväksi puoleksi taas sen, että siinä on selitetty yleisellä tasolla toimenpiteiden merkitys. Lopuksi Kortosalmi sanoo, että toimenpiteiden merkitysten avaaminen vähentää uuden työntekijän niin kutsuttua henkistä kuormaa. Tällä Kortosalmi viittaa siihen, että jos toimenpiteiden merkitystä ei kerrota työntekijälle, ne jäisivät mietityttämään häntä. Tällöin työntekijä voi jäädä pohtimaan asiaa, jota ei ole ymmärtänyt, pitkäksi ajaksi. Kortosalmi myös pohtii, että ymmärtämättömyys voi hidastaa työnteon nopeutta.

Riikonen (2021) kertoo, että hänellä on kokemusta tietotekniikasta seitsemän vuotta, joista kolme on kertynyt työelämän aikana ja neljä opiskeluaikana. Kokemusta vastaavanlaisten ympäristöjen, kuten RF L1low -ohjelmistokehitysympäristön, käyttööntamisesta Riikosella ei ollut aiemmin. Muissa projekteissa, joissa Riikonen on ollut mukana, tarvittavat työkalut ovat olleet valmiiksi asennettuina tietokoneelle. Riikonen kommentoi RF L1low -ohjelmistokehitysympäristön käyttöönotto-opasta (Liite 1) erittäin hyväksi. Hänen mielestään uusi työntekijä ei voi epäonnistua ohjelmistokehitysympäristön käyttööntamisessa, jos opasta seurataan vaihe vaiheelta. Riikoselle ei myöskään jäänyt epäselväksi suoritettujen toimenpiteiden merkitys, koska hänen mielestään ne on selitetty ymmärrettävästi opinnäytetyössä.

Riikonen (2021) arvioi, että RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaan (Liite 1) avulla säästettiin huomattavasti aikaa ohjelmistokehitysympäristön käyttööntamisessa. Riikosella ei ollut minkäänlaista tietoa siitä, mitä ympäristöä käytetään RF L1low -ohjelmiston kehittämiseen ja ilman ohjeistusta hän ei olisi tiennyt mistä aloittaa. Riikonen sanoo, jos käyttöönotto-opasta ei olisi ollut käytettävissä, kokeneemman työntekijän olisi täytynyt kertoa hänelle suoritettavat toimenpiteet. Tämä taas olisi vienyt työntekijän aikaa pois muista työtehtävistä.

Riikosen (2021) mielestä uuden työntekijän ei tarvitse olla tietotekniikan ammattilainen, jotta tarvittavan ohjelmistokehitysympäristön saa otettua käyttöön. Henkilö, joka vasta opiskelee tietotekniikkaa ja hallitsee tietokoneen käytön, saa suoritettua RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaan (Liite 1) avulla helposti tarvittavat toimenpiteet. Riikonen mainitsee, että uudella työntekijällä ei välttämättä tarvitse olla kokemusta Linux-käyttöjärjestelmästä, jotta ohjelmistokehitysympäristön saa käyttöön. RF L1low -ohjelmistokehitysympäristön käyttöönotto-opas (Liite 1) ohjeistaa riittävän Linuxin käytön, jotta työntekijä voi aloittaa uuden ominaisuuden kehittämisen RF L1low -ohjelmistoon.

8 Yhteenveto ja pohdinta

RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaan (Liite 1) tarkoituksena oli selvittää, millaisilla toimenpiteillä Nokia Oyj:n Oulun toimipisteelle saapunut työntekijä saa tietokoneelle käyttöönsä ohjelmistokehitysympäristön. Kyseinen ohjelmistokehitysympäristö vaaditaan, jotta työntekijä voi kehittää ohjelmistoa nimeltä RF L1low. RF L1low -ohjelmistokehitysympäristön käyttöönotto-opas (Liite 1) neuvoo uutta työntekijää etenemään vaihe vaiheelta, kunnes ohjelmistokehitysympäristö on saatu käyttöön onnistuneesti. Tämän jälkeen työntekijä voi aloittaa uuden ominaisuuden kehittämisen RF L1low -ohjelmistoon. RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaan (Liite 1) hyötynä on se, että työntekijä voi ottaa ohjelmistokehitysympäristön käyttöön omaan tahtiinsa. Tämä taas säästää aikaa käyttöönottoprosessissa, kun uuden työntekijän ei tarvitse olla jatkuvasti kysymässä neuvoa suorittaakseen tietyn toimenpiteen. Kokeneemmat työntekijät voivat siis keskittyä omiin tehtäviinsä, ja uudet työntekijät pääsevät nopeammin varsinaisen työn pariin.

RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaan (Liite 1) lisäksi opinnäytetyö sisältää kuvauksen kehitettävästä RF L1low -ohjelmistosta, tietoja ohjelmistokehitysympäristön käyttöönottoon liittyvistä teknologioista, selvityksen RF L1low -ohjelmistokehitysympäristön käyttöönoton vaiheista, kuvauksen RF L1low -ohjelmiston kehitysprosessista ja selvityksen kuinka RF L1low -ohjelmistoon tehty uusi ominaisuus saadaan lähetettyä Gerrit-palveluun arvioitavaksi. Erilaisten asioiden selvittäminen opinnäytetyössä varmistaa sen, että uusi työntekijä tietää, mitä hänen pitää tehdä ja mikä merkitys milläkin toimenpiteellä on. Tällä on pyritty siihen, että työntekijällä olisi mahdollisimman hyvät lähtökohdat aloittaa työnteko.

Opinnäytetyön ensimmäinen tutkimuskysymys oli, mitkä toimenpiteet uuden työntekijän täytyy suorittaa, jotta hän saa RF L1low -ohjelmistokehitysympäristön käyttöön tietokoneelle. Toinen tutkimuskysymys oli, saako työntekijä ohjelmistokehitysympäristön käyttöön tehdyn RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaan (Liite 1) avulla. Molempiin tutkimuskysymyksiin on saatu vastaus. RF L1low -ohjelmistokehitysympäristön käyttöönotto-opas (Liite 1) sisältää tarvittavat toimenpiteet, joiden avulla ohjelmistokehitysympäristö saadaan käyttöön. Tämä käy ilmi RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaan (Liite 1) toimivuuden kokeilemisesta uusilla työntekijöillä, jotka saapuivat Nokia Oyj:n Oulun toimipisteelle kehittämään RF L1low -ohjelmistoa. Uudet työntekijät saivat tarvittavan ohjelmistokehitysympäristön käyttöönsä RF L1low -ohjelmistokehitysympäristön käyttöönotto-oppaan (Liite 1) avulla ja pystyivät aloittamaan RF L1low -ohjelmiston kehittämisen. Uusien työntekijöiden ei myöskään tarvinnut kysyä ulkopuoliselta henkilöltä neuvoa

käyttöönottoprosessin aikana. Tämä tarkoittaa sitä, että käyttöönotto-opas on toteutettu onnistuneesti siihen tarkoitukseen, johon se on suunniteltu.

Uusilta työntekijöiltä saadun palautteen perusteella pystytään tekemään johtopäätös, että RF L1low -ohjelmistokehitysympäristön käyttöönotto-opasta (Liite 1) voidaan hyödyntää tulevaisuudessa uuden työntekijän perehdyttämisessä, kun hän saapuu Nokia Oyj:n Oulun toimipisteelle kehittämään RF L1low -ohjelmistoa. Perehdyttämistä tukee myös opinnäytetyössä oleva luku RF L1low -ohjelmistokehitysympäristön käyttöönoton vaiheista, joka selvittää työntekijälle vaadittavien toimenpiteiden merkitykset. Uuteen työhön astuvan työntekijän aikaisemalla kokemuksella ei ole merkittävää vaikutusta siihen, että hän voisi ymmärtää RF L1low -ohjelmistokehitysympäristön käyttöönotto-opaan (Liite 1) sisällön. Ohjelmistokehitysympäristön voi ottaa käyttöön RF L1low -ohjelmistokehitysympäristön käyttöönotto-opaan (Liite 1) avulla niin ensimmäiseen kesäharjoitteluun tullut opiskelija kuin myös henkilö, jolla on jo monen vuoden kokemus ohjelmistokehityksestä.

Jatkotutkimusidea tälle opinnäytetyölle on, että RF L1low -ohjelmiston ja 5G-antennin arkkitehtuurista kerrottaisiin enemmän. RF L1low -ohjelmiston esittely opinnäytetyössä on pieni kuvaus suuremmasta kokonaisuudesta. Jatkotutkimuksessa voitaisiin kertoa syvemmin ohjelmiston ja laitteiston toiminnallisuudesta ja siitä, kuinka ne toimivat yhdessä. Tällaisen tutkimuksen toteuttajalta vaadittaisiin erittäin laaja teknillisen tason tuntemus muun muassa 5G-verkosta, radioverkkoteknologiasta ja sulautetuista järjestelmistä. En pysty toteuttamaan esitettyä ideaa, koska huolellinen perehtyminen äsken mainittuihin tekniikoihin vaatii pidemmän ajan, mitä on käytettävissä tämän opinnäytetyön kirjoittamiseen. Minulla ei myöskään ole aikaisempaa kokemusta tekniikoista, joita tarvittaisiin ehdotetun jatkotutkimuksen toteuttamista varten, koska ne eivät sisälly opetussuunnitelmaani.

Lähteet

Aho, J., Rytönen, J., Salmi, J. & Väisänen, N. 2019. Beamer L1low SW Intro for Newcomers. Video. Nokia Oyj. Viitattu 20.7.2021. Saatavissa vain Nokia Oyj:n työntekijöiden sisäiseen käyttöön https://confluence.ext.net.nokia.com/download/attachments/858306129/beamer_l1low_sw_intro_for_newcomers.mp4?version=1&modificationDate=1561541855000&api=v2

Arvio, A. 2021. Insinööri. Nokia Oyj. Haastattelu 12.7.2021.

Atlassian Plc. 2017. Basic Git commands. Viitattu 24.6.2021. Saatavissa <https://confluence.atlassian.com/bitbucketserver/basic-git-commands-776639767.html>

Automation Step by Step. 2020. School Of Basics | What is SSH | How SSH works. Youtube-video. Viitattu 2.7.2021. Saatavissa <https://www.youtube.com/watch?v=IRMAJwMQ0Vc>

Ayodeji, B. 2019. How to Write Good Commit Messages: A Practical Git Guide. freeCodeCamp. Viitattu 25.6.2021. Saatavissa <https://www.freecodecamp.org/news/writing-good-commit-messages-a-practical-guide/>

Bilde, L. 2016. What is a Repository. Youtube-video. Viitattu 25.6.2021. Saatavissa <https://www.youtube.com/watch?v=9A26ybw6tGY>

Chacon, S. & Straub, B. 2014. Pro Git. Apress LLC. Viitattu 25.6.2021. Saatavissa <https://git-scm.com/book/en/v2>

Charles, D. 2020. Gerrit User Guide. Nokia Oyj. Viitattu 21.6.2021. Saatavissa vain Nokia Oyj:n työntekijöiden sisäiseen käyttöön <https://confluence.ext.net.nokia.com/display/EE-GERRIT/Gerrit+User+Guide>

CMakeLists.txt. 2021. JetBrains. Viitattu 13.7.2021. Saatavissa <https://www.jetbrains.com/help/clion/cmakelists-txt-file.html>

CMake Reference Documentation. Kitware. Viitattu 13.7.2021. Saatavissa <https://cmake.org/cmake/help/v3.21/>

Docker overview. Docker Inc. Viitattu 20.6.2021. Saatavissa <https://docs.docker.com/get-started/overview/>

Drozdy, A. 2018. Beamforming theory basics. Video. Nokia Oyj. Viitattu 19.7.2021. Saatavissa vain Nokia Oyj:n työntekijöiden sisäiseen käyttöön

<https://nokialearn.csod.com/LMS/Video/LaunchVideo.aspx?loid=00342956-f51d-45c1-bc91-1de32e3fcea1>

Elo, J. 2021. Gerrit How To for L1SW5G, L1 Platform. Nokia Oyj. Viitattu 26.6.2021. Saatavissa vain Nokia Oyj:n työntekijöiden sisäiseen käyttöön <https://confluence.ext.net.nokia.com/display/CUP/Gerrit+How+To+for++L1SW5G%2C+L1+Platform>

Git commit. Atlassian Plc. Viitattu 24.6.2021. Saatavissa <https://www.atlassian.com/git/tutorials/saving-changes/git-commit>

GitHub Training & Guides. 2013. What is VCS? (Git-SCM) • Git Basics #1. Youtube-video. Viitattu 21.6.2021. Saatavissa <https://www.youtube.com/watch?v=8oRjP8yj2Wo>

Grand Canyon University. 2019. A Lost Laptop Is a Cybersecurity Threat. Viitattu 1.7.2021. Saatavissa <https://www.gcu.edu/blog/engineering-technology/lost-laptop-cybersecurity-threat>

Guide 1: Get started with Jira Software. Atlassian Plc. Viitattu 19.7.2021. Saatavissa <https://www.atlassian.com/software/jira/guides/getting-started/overview#jira-software-hosting-options>

Hoglund, C. 2019. Welcome to Beaver Team. Nokia Oyj. Viitattu 21.7.2021. Saatavissa vain Nokia Oyj:n työntekijöiden sisäiseen käyttöön <https://confluence.ext.net.nokia.com/display/L1RDOUL/Welcome+to+Beaver+Team>

Impola, L. 2021. Insinööri. Nokia Oyj. Haastattelu 5.7.2021.

JetBrains. Viitattu 20.6.2021. Saatavissa <https://www.jetbrains.com/clion/>

Kempin, E. Gerrit Concepts and Workflows. Google LLC. Viitattu 26.6.2021. Saatavissa https://docs.google.com/presentation/d/1C73UgQdzZDw0qzpaEqlC6SPu-jZJhqamyqO1XOHjH-uk/edit#slide=id.g4d6c16487b_31_60

Kortesalmi, J. 2021. Insinööri. Nokia Oyj. Haastattelu 10.8.2021.

Laihiainen, J. 2021. Insinööri. Nokia Oyj. Haastattelu 2.8.2021.

Louargant, M. 2018. AFT/HART R&D DevOps Framework Training Virtual LinSEE Servers. Nokia Oyj. Viitattu 20.6.2021. Saatavissa vain Nokia Oyj:n työntekijöiden sisäiseen käyttöön <https://nokialearn.csod.com/ui/lms-learning-details/app/material/c12ce749-7883-49a5-9a47-3029ace8689c>

LinSEE User Guide. 2016. Nokia Oyj. Viitattu 21.6.2021. Saatavissa vain Nokia Oyj:n työntekijöiden sisäiseen käyttöön

https://nokia.sharepoint.com/:p:/r/sites/airscale13shard/_layouts/15/Doc.aspx?sourcedoc=%7B4708C593-45FC-46F9-AAE9-D6F0EE345AF3%7D&file=13.%20EE%20Lin-SEE%20Usage%20Guide.pptx&action=edit&mobileredirect=true&DefaultItemOpen=1

Olenski, S. 2017. Is The Data On Your Business' Digital Devices Safe? Forbes LLC. Viitattu 1.7.2021. Saatavissa <https://www.forbes.com/sites/steveolenski/2017/12/08/is-the-data-on-your-business-digital-devices-safe/?sh=55c05d124c6a>

Pekkarinen, L. 2020. Hard Drive and Full Disk Encryption: What, Why, and How? Miradore Ltd. Viitattu 1.7.2021. Saatavissa <https://www.miradore.com/blog/hard-drive-encryption-full-disk-encryption/>

Record Nations. Why You Should Encrypt Your External Hard Drive. Viitattu 1.7.2021. Saatavissa <https://www.recordnations.com/articles/encrypt-external-hard-drive/>

Riikonen, P. 2021. Insinööri. Nokia Oyj. Haastattelu 19.8.2021.

Savusalo, T. 2021. Insinööri. Nokia Oyj. Haastattelu 2.8.2021.

Simplilearn. 2018. What Is Jenkins? | What Is Jenkins And How It Works? | Jenkins Tutorial For Beginners | Simplilearn. Youtube-video. Viitattu 21.6.2021. Saatavissa <https://www.youtube.com/watch?v=LFDnKpOTg>

The Coding Train. 2016. 1.2: Branches - Git and GitHub for Poets. Youtube-video. Viitattu 22.6.2021. Saatavissa <https://www.youtube.com/watch?v=oPpnCh7InLY>

TheCodex. 2019. Git and GitHub Tutorials #3 - What are Git Branches? Youtube-video. Viitattu 22.6.2021. Saatavissa <https://www.youtube.com/watch?v=iJKlXrJ40ss>

User Guide. Nokia Oyj. Viitattu 10.7.2021. Saatavissa vain Nokia Oyj:n työntekijöiden sisäiseen käyttöön <https://gerrit.ext.net.nokia.com/gerrit/Documentation/intro-user.html>

Väisänen, N. 2019. L1low software building and testing demo. Video. Nokia Oyj. Viitattu 17.7.2021. Saatavissa vain Nokia Oyj:n työntekijöiden sisäiseen käyttöön https://confluence.ext.net.nokia.com/download/attachments/858306129/l1low_sw_building_and_testing_demo.mp4?version=1&modificationDate=1561541855000&api=v2

What is a Container? A standardized unit of software. Docker Inc. Viitattu 20.6.2021. Saatavissa <https://www.docker.com/resources/what-container>

Liite 1. RF L1low -ohjelmistokehitysympäristön käyttöönotto-opas

1 Ensimmäiset askeleet

1.1 Microsoft Windows 10 -käyttöjärjestelmän asentaminen

Asentaakseen Microsoft Windows 10 -käyttöjärjestelmän tietokoneelle, työntekijän on suoritettava alla olevat kohdat numerjärjestyksessä:

1. Tietokone täytyy olla liitettynä verkkovirtaan käyttöjärjestelmän asennuksen ajaksi. Asennusprosessi saattaa kestää useita tunteja. Jos tietokoneesta loppuu akku asennuksen aikana, asentaminen keskeytyy.
2. Tietokone täytyy olla liitettynä Nokia Oyj:n Oulun toimipisteen verkkoyhteyteen Ethernet-kaapelilla. Jos tietokoneessa ei ole paikkaa Ethernet-kaapelille, on käytettävä adapteria apuna langallisen verkkoyhteyden muodostamiseksi.
3. Tietokone käynnistetään virtapainikkeesta ja tämän jälkeen on painettava F12-näppäintä toistuvasti, kunnes päädytään BIOS-laiteohjelmistoon. Huomioitavaa on, että F12-näppäimen painaminen toimii Lenovo-merkkisissä tietokoneissa, kun halutaan päästä BIOS-laiteohjelmistoon. Jos käytössä on HP-merkinen tietokone, käynnistysvaiheessa täytyy painaa toistuvasti F9-näppäintä, jotta päästään BIOS-laiteohjelmistoon. (Savusalo 2021.)
4. Kun on päädytty BIOS-laiteohjelmistoon, on valittava PCI LAN, IPv4 tai jokin muu vastaava verkon käynnistysvaihtoehto.
5. Kun tietokoneen näytölle ilmestyy ikkuna nimeltä Task Sequence Wizard, on klikattava Next-painiketta.
6. Tietokoneen näytölle ilmestyy uusi samanniminen ikkuna, kuin kohdassa 5. Seuraavaksi on valittava valikosta Nokia Win 10 PC Image x.x.x ja klikattava Next-painiketta.
7. Tietokoneen näytölle ilmestyy varoitusviesti, jossa kerrotaan uuden käyttöjärjestelmän asentamisesta ja tämän toimenpiteen myötä kaikki vanhat tiedot häviävät, jos niistä ei ole tehty varmuuskopioita. Varoitusviestin voi ohittaa turvallisesti, koska, käyttöönotettava tietokone on täysin uusi, joten ei ole minkäänlaista vaaraa siitä, että tärkeitä tiedostoja menetettäisiin. Jotta käyttöjärjestelmä asentuu uudelle tietokoneelle, on klikattava Confirm-painiketta. Asennus voi kestää 2–8 tuntia ja tähän vaikuttavat käytössä oleva tietokone sekä verkkoyhteyden nopeus.

8. Jos tietokoneen näytölle ilmestyy Windows 10 sisäänkirjautumisnäkyvä, käyttöjärjestelmän asentaminen on tapahtunut onnistuneesti. Tämän jälkeen voidaan kirjautua sisään tietokoneelle käyttämällä NSN-INTRA käyttäjätunnusta ja salasanaa. (Getting Started Guide, New Windows PC Hardware 2017.) Käyttäjätunnus on muotoa: etunimen ensimmäinen kirjain pienellä ja sukunimi pienellä kirjoitettuna. Esimerkiksi henkilön Jonne Kiukas NSN-INTRA käyttäjätunnus olisi jkiukas. Jos työntekijä ei ole saanut NSN-INTRA käyttäjätunnusta ja salasanaa tai ne eivät toimi sisään kirjaututtaessa, hänen täytyy ottaa yhteys linjamanageriin.

1.2 Pääsyoikeuksien tarkistaminen

Kun tietokoneelle on kirjaututtu sisään, täytyy varmistaa pääsyoikeudet eri palveluihin, joita tarvitaan RF L1low -ohjelmiston kehittämisessä. Samoja pääsyoikeuksia tarvitaan myös ohjelmistokehitysympäristön käyttöönottamisessa. Kaikki oikeudet kannattaa tarkistaa kerralla, koska se säästää aikaa, eikä työntekijän tarvitse palata anomaan pääsyä tiettyyn palveluun kesken kehitysympäristön käyttöönoton. (Laihiainen 2021.)

Työntekijä tarkistaa linjamanagerin kanssa pääsyoikeudet alla oleviin palveluihin. Myöhemmässä vaiheessa käyttöönotto-opasta oletetaan, että työntekijällä on pääsy kyseisiin palveluihin. Jos pääsy kuitenkin evätään johonkin tiettyyn palveluun, työntekijän on otettava yhteys linjamanageriin. (Laihiainen 2021.)

Työntekijä tarvitsee pääsyoikeuden

- LinSEE-palvelimelle
- Confluence-tietopalveluun
- Jira-tehtävienhallintapalveluun
- GitLab-versionhallintapalveluun
- Gerrit-koodinarviointipalveluun
- espoo1 Artifactory -palveluun
- espoo2 Artifactory -palveluun (Laihiainen 2021).

2 RF L1low -ohjelmistokehitysympäristön käyttöönotaminen

2.1 SSH-työkalun asentaminen

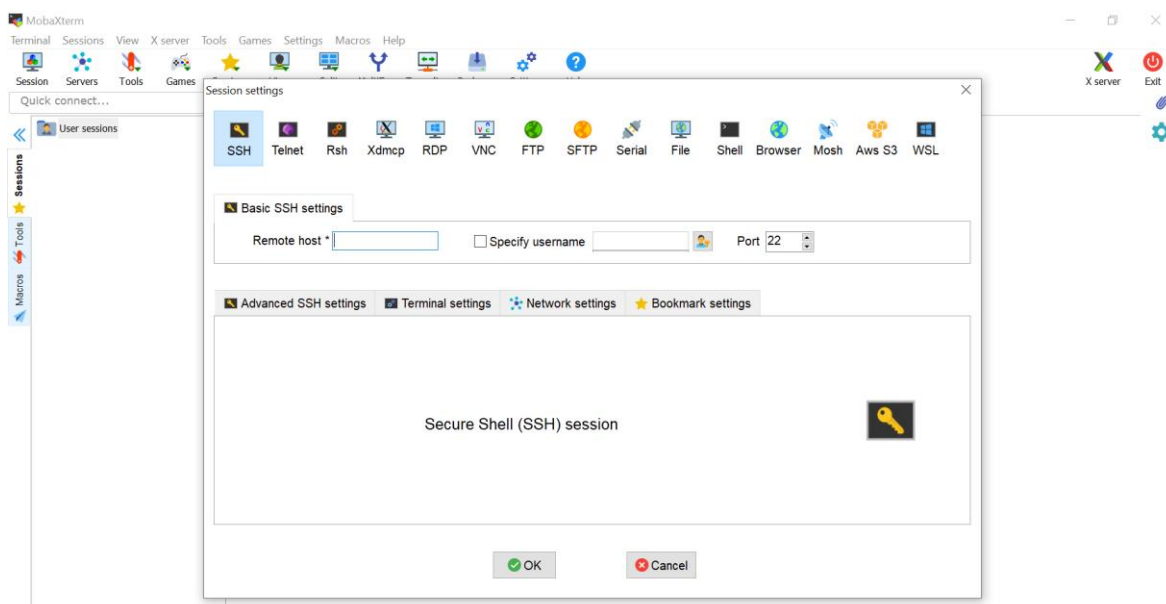
Asentaakseen MobaXterm-ohjelmiston, työntekijän on suoritettava alla olevat kohdat numerojärjestyksessä:

1. Työntekijän täytyy siirtyä osoitteeseen <https://mobaxterm.mobatek.net/download.html>.
2. Työntekijän täytyy klikata latauspainiketta, joka on Home Edition -laatikon sisällä.
3. Työntekijän täytyy valita Installer edition -latausvaihtoehto.
4. Seuraavaksi on avattava ZIP-kansio, jonka verkkoselain on ladannut, ja kaksoisklikattava asennusohjelmistoa nimeltä MobaXterm_installer.msi.
5. Tietokoneen näytölle avautuu MobaXterm Setup -ikkuna. Next-painiketta klikkaamalla voidaan siirtyä asennuksen aloittamiseen.
6. Kun MobaXtermin lisenssisopimus on luettu ja hyväksytty, on klikattava Next-painiketta.
7. Seuraavaksi täytyy valita tietokoneesta sijainti, jonne MobaXterm-ohjelmisto halutaan asentaa, ja klikata Next-painiketta.
8. Jotta SSH-työkalun asentaminen voidaan aloittaa, on klikattava Install-painiketta.
9. Kun Windows varmistaa, että saako MobaXterm installer -sovellus tehdä muutoksia laitteeseen, on klikattava Kyllä-painiketta. Jos tämän jälkeen tietokoneen näytölle ilmestyy virheilmoitus, että tiedostoa ei löytynyt, täytyy ladatun ZIP-kansion sisältä viedä tiedosto nimeltä MobaXterm_installer.dat oikean kansion sisälle. Oikea kansio, johon äsken mainittu tiedosto täytyy siirtää, löytyy tietokoneen sijainnista C:\Käyttäjät\[Käyttäjän nimi]\AppData\Local\Temp\Temp1_MobaXterm_Installer.zip. Kun MobaXterm_installer.dat-tiedosto on viety oikean kansion sisälle, on klikattava Retry-painiketta.
10. Kun asennus on suoritettu, lopuksi on klikattava Finish-painiketta.

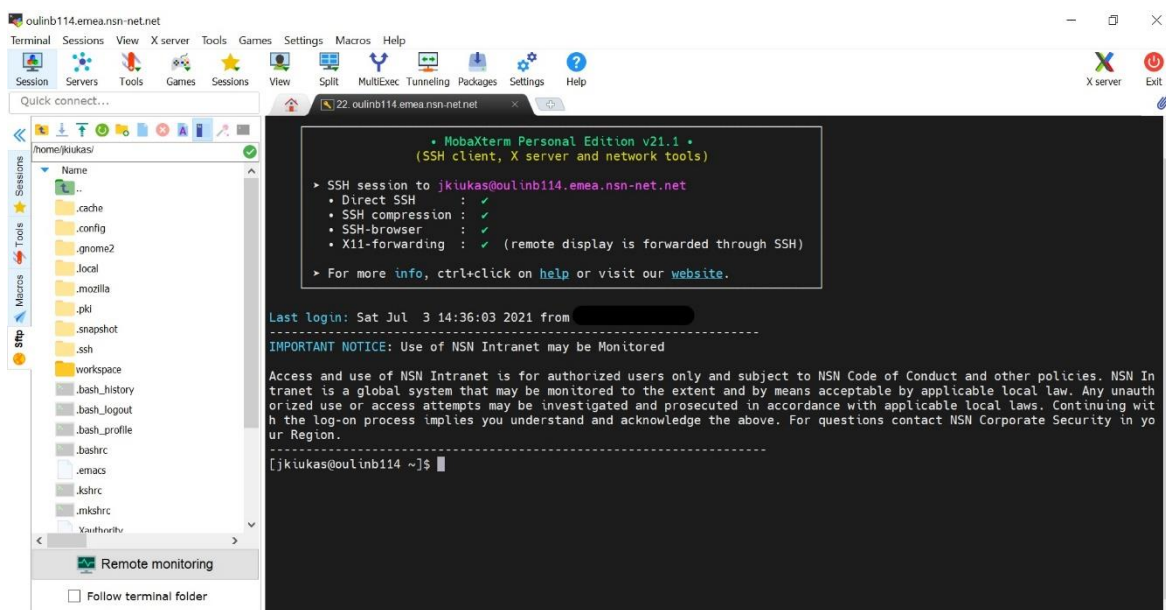
2.2 Kirjautuminen LinSEE-palvelimelle

Kirjautuakseen sisään LinSEE-palvelimelle, työntekijän on suoritettava alla olevat kohdat numerojärjestyksessä:

1. Tietokoneella täytyy avata MobaXterm SSH-työkalu ja painaa vasemmasta ylänurkasta Sessions-painiketta. Tietokoneen näytölle avautuu uusi Sessions settings -ikkuna.
2. Sessions settings -ikkunan vasemmasta ylänurkasta on painettava SSH-painiketta, jonka jälkeen näytölle avautuu kuvan 1 kaltainen näkymä.
3. Remote host -tekstikenttään on kirjoitettava oulinb114.emea.nsn-net.net tai jokin muu toimiva LinSEE-palvelin.
4. Halutessaan työntekijä voi laittaa rastin ruutuun kohdassa Specify username ja kirjoittaa Nokia-käyttäjätunnuksensa tekstikenttään. Tällöin ei tarvitse kirjoittaa käyttäjätunnusta joka kerta, kun kirjaudutaan LinSEE-palvelimelle sisään.
5. Port-tekstikentässä on oltava luku 22.
6. Seuraavaksi on klikattava OK-painiketta, jonka jälkeen näytölle ilmestyy mustalla pohjalla oleva tekstieditori.
7. LinSEE-palvelimelle kirjaudutaan sisään syöttämällä Nokia-tunnukset tekstieditoriin ja painamalla enter-näppäintä. Jos tietokoneen näytölle ilmestyy kuvan 2 mukainen ilmoitus, LinSEE-palvelimelle on kirjauduttu onnistuneesti.



Kuva 1. LinSEE-palvelimen määrittäminen MobaXtermissä



Kuva 2. Ilmoitus LinSEE-palvelimelle kirjautumisesta

2.3 RF L1low -repositorion kloonaminen

Kloonataksaan RF L1low -repositorion LinSEE-palvelimelle, työntekijän on suoritettava alla olevat kohdat numerojärjestyksessä:

1. SSH-työkalun mustalle tekstieditorille on syötettävä Linux-komento `mkdir workspace`, joka luo uuden kansio nimeltä `workspace`. Kyseisen kansion sisälle kloonataan RF L1low -repositorio (Impola 2021). `workspace`-kansio on luotava kotikansion sisällä. Työntekijä on automaattisesti kotikansion sisällä, kun LinSEE-palvelimelle kirjaututaan sisään. Kotikansion sisälle voidaan palata mistä tahansa kansiosista syöttämällä SSH-työkalun tekstieditoriin Linux-komento `cd` (Parker 1998, 157).
2. Seuraavaksi on siirryttävä `workspace` kansioon syöttämällä Linux-komento `cd workspace` (Parker 1998, 156, 161).
3. RF L1low -repositorio on kloonattava Git-komennolla `git clone https://gerite1.ext.net.nokia.com/RFSW/INTERNAL/l1low`. Lisäksi on syötettävä Nokia-tunnukset. Jos RF L1low -repositorion kloonaminen ei onnistu, työntekijän täytyy tarkistaa linjamanagerin kanssa pääsyoikeudet Gerrit-palveluun. (Impola 2021.) Huomioitavaa on, ettei `git clone` -komennossa olekaan URL-osoitteeseen tule tavuvii-vaa. Jos repositorion kloonaminen on tapahtunut onnistuneesti, SSH-työkaluun ilmestyy kuvan 3 mukainen ilmoitus.

```

remote: Counting objects: 1755, done
remote: Finding sources: 100% (75/75)
remote: Total 213025 (delta 1), reused 212971 (delta 1)
Receiving objects: 100% (213025/213025), 200.45 MiB | 23.95 MiB/s, done.
Resolving deltas: 100% (159642/159642), done.
Updating files: 100% (7474/7474), done.

```

Kuva 3. Ilmoitus RF L1low -repositorion kloonamisesta

2.4 Käyttäjätietojen päivittäminen

Päivittääkseen omat käyttäjätietonsa user.conf-tiedostoon, työntekijän on suoritettava alla olevat kohdat numerojärjestyksessä:

1. SSH-työkalun tekstieditoriin täytyy syöttää Linux-komento `cd workspace/l1low/docker/dev/dockerBeaver/shared`, jotta päädytään kansioon, jossa user.conf-tiedosto sijaitsee.
2. Tiedosto on avattava muokkaustilaan syöttämällä Linux-komento `nano user.conf` (Rantala 2003, 70).
3. Tiedostoon täytyy lisätä Nokia-sähköpostiosoite, etu- ja sukunimi ja Nokia-käyttäjä-tunnus niille tarkoitetuille paikoille.
4. Muokattu tiedosto on tallennettava painamalla näppäinyhdistelmää `Ctrl + O`. Tämän jälkeen täytyy painaa enter-näppäintä, jolla hyväksytään tiedoston tallennusnimi. Lopuksi on painettava vielä näppäinyhdistelmää `Ctrl + X`, jolla poistutaan tiedostosta. (Rantala 2003, 70.)

2.5 SSH-avaimien generointi ja SSH-avaimen lisääminen tarvittaviin palveluihin

Kun työntekijä on kirjautunut sisään LinSEE-palvelimelle, kloonannut LinSEE-palvelimelle RF L1low -repositorion ja päivittänyt käyttäjätietonsa user.conf-tiedostoon, on seuraavaksi generoitava SSH-avaimet.

Generoidakseen SSH-avaimet, työntekijän on suoritettava alla olevat kohdat numerojärjestyksessä:

1. SSH-työkalun tekstieditoriin täytyy syöttää kotikansion sisällä komento `ssh-keygen -q -t rsa -C "Nokia-sähköpostiosoite"`. Komennossa korvataan Nokia-sähköpostiosoite omalla Nokia-sähköpostiosoitteella. Lainausmerkit ovat osa komentoa, joten niitä ei saa jättää pois.

2. Graafisen kansiorakenteen yläpuolelta täytyy painaa vihreää nappia, joka päivittää kotikansion. Kansion sisältä pitäisi löytyä nyt alikansio nimeltä .ssh. Kotikansion sisältä pitäisi löytyä myös julkinen SSH-avain tiedostosta nimeltä id_rsa.pub ja yksityinen SSH-avain tiedostosta nimeltä id_rsa. Molemmat tiedostot voi avata ja tarkistaa, että ne sisältävät numeroista, kirjaimista ja erikoismerkeistä koostuvan merkkijonon. Julkinen SSH-avain voidaan jakaa esimerkiksi muihin palveluihin, mutta yksityisen SSH-avaimen sisältö on pidettävä salassa muilta.
3. Kotikansioon generoitunut julkinen SSH-avain on siirrettävä .ssh-kansion sisälle syöttämällä tekstieditoriin Linux-komento `mv id_rsa.pub .ssh` (Parker 1998, 162).
4. Kolmas kohta täytyy toistaa uudestaan, mutta tällä kertaa Linux-komentoon vaihdetaan tiedosto nimeltä id_rsa, jossa on yksityinen SSH-avain.

Julkinen SSH-avain, joka löytyy tiedostosta id_rsa.pub, on lisättävä eri palveluihin, joita Nokia Oyj:llä käytetään RF L1low -ohjelmiston kehityksessä. Tässä vaiheessa työntekijän täytyy kirjautua sisään LinSEE-palvelimelle ja avata id_rsa.pub-tiedosto. Ensimmäinen palvelu, johon julkinen SSH-avain täytyy käydä lisäämässä, on Gerrit-koodinarviointipalvelu.

Lisätäkseen julkisen SSH-avaimen Gerrit-palveluun, työntekijän on suoritettava alla olevat kohdat numerojärjestyksessä:

1. Työntekijän täytyy avata osoite <https://gerrite1.ext.net.nokia.com> verkkoselaimessa ja kirjautua palveluun sisään Nokia-tunnuksillaan.
2. Sisään kirjautumisen jälkeen täytyy klikata oikeassa yläkulmassa olevaa ratasta, josta päädytään asetuksiin.
3. Seuraavaksi on klikattava vasemmalla puolella olevaa tekstiä, jossa lukee SSH Keys.
4. Julkinen SSH-avain on kopioitava id_rsa.pub-tiedostosta ja liitettävä Public Key -tekstikenttään Gerrit-palvelun asetuksissa.
5. Algorithm-tekstikenttään on kirjoitettava ssh-rsa ja Comment-tekstikenttään Nokia-sähköpostiosoite.
6. Lopuksi on painettava SAVE CHANGES -nappia, jotta tehdyt muutokset tallentuvat.

GitLab on toinen palvelu, johon julkinen SSH-avain täytyy käydä lisäämässä ja Nokia Oyj:llä käytetään kahta GitLab-palvelua.

Lisätäkseen julkisen SSH-avaimen GitLab-palveluihin, työntekijän on suoritettava alla olevat kohdat numerojärjestyksessä:

1. Työntekijän täytyy avata osoitteet <https://gitlabe1.ext.net.nokia.com> ja <https://gitlabe2.ext.net.nokia.com> verkkoselaimessa omiin välilehtiin ja kirjautua palveluihin sisään Nokia-tunnuksillaan.
2. Sisään kirjautumisen jälkeen, molemmilta sivuilta on klikattava oikeasta ylänurkasta löytyvää avatar-kuvaketta ja sen jälkeen Settings-tekstiä.
3. Kun päädytään asetuksiin, on klikattava molempien sivujen vasemmasta reunasta SSH Keys -tekstiä, jonka jälkeen ollaan sivuilla, jonne lisätään SSH-avain.
4. Julkinen SSH-avain täytyy kopioida id_rsa.pub-tiedostosta ja liitettävä molempien GitLab-palveluiden Key-tekstikenttään.
5. Title-tekstikenttiin on syötettävä haluttu nimi SSH-avaimelle.
6. Expires at -tekstikenttiin ei tarvitse asettaa päivämäärää, jolloin SSH-avain vanhentuisi.
7. Molemmilta asetussivuilla täytyy painaa vihreää nappia, jossa lukee Add key, jotta tehdyt muutokset tallentuvat.

Viimeinen palvelu, johon julkinen SSH-avain täytyy käydä lisäämässä, on JFrog Artifactory. Myös JFrog Artifactory -palveluita on Nokia Oyj:llä käytössä kaksi kappaletta ja ne löytyvät osoitteista <https://artifactory-espoo1.int.net.nokia.com> ja <https://artifactory-espoo2.int.net.nokia.com>.

Lisätäkseen julkisen SSH-avaimen JFrog Artifactory -palveluihin, työntekijän on suoritettava alla olevat kohdat numerojärjestyksessä:

1. Työntekijän täytyy avata äsken mainitut linkit verkkoselaimessa omiin välilehtiin
2. Sivustoilla on klikattava Log In -tekstiä oikeasta yläkulmasta, jonka jälkeen on kirjaututtava sisään Nokia-tunnuksilla.
3. Sivujen oikeasta yläkulmasta on klikattava käyttäjänimeä ja kirjoitettava salasana uudelleen Current Password -tekstikenttiin.
4. Seuraavaksi on siirryttävä molempien sivujen loppuun ja lisättävä SSH-tekstikenttään julkinen SSH-avain.
5. SSH-avain on tallennettava kumpaankin JFrog Artifactory -palveluun, painamalla vihreää Save-painiketta.

2.6 JFrog Artifactory API -avaimien generointi ja lisääminen LinSEE-palvelimelle

JFrog Artifactory API -avaimet generoidaan osoitteissa <https://artifactory-espoo1.int.net.nokia.com> ja <https://artifactory-espoo2.int.net.nokia.com>. Jos molemmista JFrog Artifactory -palveluista oli ehditty jo kirjautua ulos SSH-avaimien lisäämisen jälkeen, palveluihin täytyy kirjautua sisään uudestaan ja siirtyä samaan paikkaan sivustoissa, jossa SSH-avain lisättiin.

Generoidakseen API-avaimet JFrog Artifactory -palveluissa ja lisätäkseen API-avaimet LinSEE-palvelimelle, työntekijän on suoritettava alla olevat kohdat numerjärjestyksessä:

1. Työntekijän täytyy klikata molemmilta JFrog Artifactory -asetussivuilta nappia, joka generoi API-avaimen.
2. Generoidut API-avaimet on tallennettava painamalla vihreää painiketta, jossa lukee Save.
3. Seuraavaksi tietokoneella on avattava MobaXterm SSH-työkalu ja luotava uusi tiedosto kotikansion sisällä kirjoittamalla Linux-komento `nano my_api_key_espoo1` (Rantala 2003, 70).
4. Generoitu API-avain on kopioitava osoitteesta <https://artifactory-espoo1.int.net.nokia.com>. Verkkosivulla API-avain on salattu ja salauksen saa pois klikkaamalla silmän muotoista kuvaketta.
5. Kopioitu API-avain on liitettävä `my_api_key_espoo1`-tiedostoon. Huomioitavaa on, että SSH-työkaluun ei voida liittää tekstiä käyttäen näppäinyhdistelmää `Ctrl + V`, vaan liittäminen on tehtävä näppäinyhdistelmällä `shift + Insert` tai hiiren kakkospainikkeen avulla.
6. Tiedosto täytyy tallentaa painamalla näppäinyhdistelmää `Ctrl + O` (Rantala 2003, 70).
7. Tiedoston tallennusnimi tallennetaan `enter`-näppäimellä ja tiedosto suljetaan `Ctrl + X` -näppäinyhdistelmällä (Rantala 2003, 70).
8. Seuraavaksi on luotava taas uusi tiedosto kotikansion sisällä kirjoittamalla Linux-komento `nano my_api_key_espoo2` (Rantala 2003, 70).
9. Tällä kertaa API-avain on kopioitava osoitteesta <https://artifactory-espoo2.int.net.nokia.com> ja liitettävä tiedostoon nimeltä `my_api_key_espoo2`.
10. Tiedosto täytyy tallentaa samalla tavalla kuin kohdassa kuusi. Kohta seitsemän täytyy myös toistaa uudestaan.

11. Graafisen kansiorakenteen yläpuolelta täytyy painaa vihreää nappia, joka päivittää kotikansion.
12. Lopuksi on tarkistettava, että tiedostot nimeltä my_api_key_espool ja my_api_key_espool2 löytyvät kotikansion sisältä. Tiedostojen olemassaolon voi myös tarkistaa syöttämällä kotikansion sisällä Linux-komennon ls -la, joka listaa kaikki kansiossa olevat tiedostot (Rantala 2003, 377).

2.7 Docker containerin käynnistäminen

Käynnistääkseen Docker containerin, työntekijän on suoritettava alla olevat kohdat numerorajestyksessä:

1. Työntekijän täytyy siirtyä tietokoneella kansioon sisälle, jossa Docker container käynnistetään, syöttämällä SSH-työkalun tekstieditoriin Linux-komento cd workspace/l1low/docker/dev/dockerBeaver.
2. Docker container täytyy käynnistää dockerBeaver-kansion sisällä syöttämällä komento ./dockerBeaverInServer.sh fullstart.
3. Jos Docker container käynnistyy onnistuneesti, pitäisi SSH-työkalun tekstieditoriin ilmestyä vastaava ilmoitus, kuten on kuvassa 4.

```
Starting SshAgent for dockerBeaver and adding keys: /home/jkortesa/.ssh/id_rsa
Agent pid 10859
Identity added: /home/jkortesa/.ssh/id_rsa (/home/jkortesa/.ssh/id_rsa)
Starting dockerBeaverServer
Create user specific network: beaver_network_for_jkortesa
bfad9ae53bcc5d683e33c1064f14cc636954a15d03cdc1f01e22679c186e80ae

Image found locally:rf-rules-local.esisoj71.emea.nsn-net.net/l1low/devenv-gui:4.0.5
Create user specific container: beaver_devenv-gui_for_jkortesa
Setting SSH_AUTH_SOCK to default
97354094071a281b23e12d54fe21b8e49d84300f8f866763066d5ec8fcb2f27c

Image name:      rf-rules-local.esisoj71.emea.nsn-net.net/l1low/devenv-gui:4.0.5
Container name:  beaver_devenv-gui_for_jkortesa
IP Address:
Vnc:             oulinb114.emea.nsn-net.net:5814
Rdp:             oulinb114.emea.nsn-net.net:3413
Ssh:             oulinb114.emea.nsn-net.net:2235
```

Kuva 4. Ilmoitus Docker containerin käynnistämisestä

2.8 VNC Viewer -ohjelmiston käyttäminen

Seuraavaksi työntekijän on ladattava VNC Viewer -ohjelmisto ja avattava se, kun asennus on suoritettu. VNC Viewer -ohjelmisto ladataan Software Center -ohjelmistojakopalvelusta

ja se löytyy valmiiksi asennettuna tietokoneelta, jonka työntekijä on saanut Nokia Oyj:ltä. (Imppola 2021.)

Saadakseen näkymän Linux-käyttöjärjestelmästä, työntekijän on suoritettava alla olevat kohdat numerojärjestyksessä:

1. Työntekijän täytyy klikata VNC Viewer -ohjelmiston vasemmassa yläkulmassa olevaa File-tekstiä ja valittava New connection.
2. VNC Server -tekstikenttään on kirjoitettava LinSEE-palvelin, jossa Docker container on käynnistetty, ja perään :5814. Numerosarja voi olla kuitenkin muuttunut, joten oikea tieto täytyy tarkistaa menemällä SSH-työkalussa kansiorakenteeseen workspace/l1low/docker/dev/dockerBeaver ja tämän jälkeen syöttää komento `./dockerBeaverInServer.sh info`. Syöte, joka täytyy antaa VNC Server -tekstikenttään, on SSH-työkalussa rivillä, jossa lukee VNC.
3. Name-tekstikenttään täytyy kirjoittaa yhteydelle vapaavalintainen nimi ja paina OK-painiketta.
4. Seuraavaksi on klikattava VNC Viewer -ohjelmistoon muodostunutta ikonia, josta aukeaa näytölle uusi ikkuna tunnistautumista varten.
5. Password-tekstikenttään täytyy kirjoittaa Nokia-käyttäjätunnus, ei siis salasanaa, ja painaa OK-painiketta. Tämän jälkeen on muodostettu yhteys Linux-käyttöjärjestelmään ja näytöllä pitäisi olla näkymä Linux-työpöydästä.

2.9 RF L1low -repositorion tarkistaminen

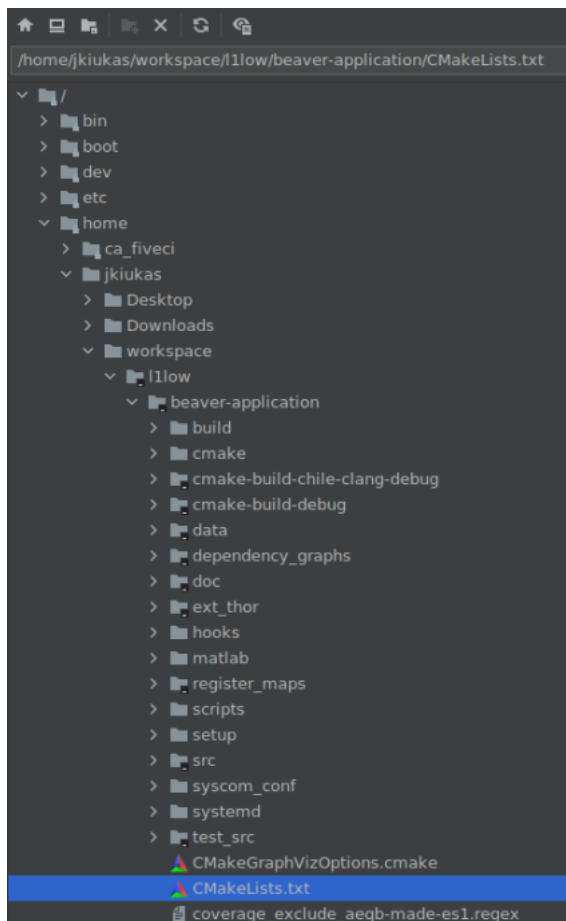
Jos Linux-käyttöjärjestelmän tiedostonhallinnasta ei löydy l1low-kansiota, työntekijän on suoritettava alla olevat kohdat numerojärjestyksessä kloonatakseen repositorion uudelleen:

1. Työntekijän täytyy avata Linux-käyttöjärjestelmässä terminaali ja syöttää Linux-komento `cd workspace`.
2. Seuraavaksi täytyy syöttää komento `git clone ssh://käyttäjätunnus@gerite1.ext.net.nokia.com:8282/RFSW/INTERNAL/l1low`. Äsken mainitussa komenossa työntekijän täytyy korvata käyttäjätunnus omalla Nokia-käyttäjätunnuksellaan.
3. Lopuksi on tarkistettava Linux-käyttöjärjestelmän tiedostonhallinnasta, että repositorio on kopioitu onnistuneesti. (Imppola 2021.)

2.10 CLion-ohjelmistokehitysympäristön konfigurointi

Konfiguroidakseen CLion-ohjelmistokehitysympäristön, työntekijän on suoritettava alla olevat kohdat numerojärjestyksessä:

1. Työntekijän täytyy avata CLion-ohjelmisto Linux-käyttöjärjestelmässä ja kaksoisklikattava näytölle avautuneesta ikkunasta kohtaa Open.
2. Työntekijän täytyy avata tekstiedosto nimeltä CMakeLists.txt kansiorakenteesta, joka on kuvattuna kuvassa 5.
3. Seuraavaksi CLion pyytää tekemään päivityksiä asetuksiin, mutta tämän toimenpiteen voi ohittaa tässä vaiheessa. Asetusten päivittämiseen palataan myöhemmin käyttöönotto-oppaassa ja osa asetuksista tulee ladattavan settings.zip-kansion kautta.
4. Työntekijän täytyy avata CLion-ohjelmistossa ylhäältä Tools-valikko ja siirtää hiiri CMake-tekstinpäälle, josta avautuu uusi valikko.
5. Avautuneesta valikosta on klikattava Change Project Root -tekstiä.
6. Juurikansion täytyy olla l1low-kansio, joten se on valittava näytölle avautuneesta listasta ja painettava oikeasta alanurkasta OK-painiketta.
7. Lopuksi on avattava CLion-ohjelmiston alareunasta terminaali ja syötettävä komento `./docker/build/run.sh init [tuotteen nimi]`. Alustuskomennossa [tuotteen nimi] tilalle täytyy vaihtaa se RF L1low -ohjelmiston tuote, jota on tarkoitus kehittää. Jos terminaaliin ilmestyy kuva majavasta, samanlainen kuin kuvassa 6, projektin alustus on onnistunut.



Kuva 5. RF L1low -ohjelmiston kansiorakenne



Kuva 6. Ilmoitus RF L1low -projektin alustamisesta

Tuodakseen tarvittavat asetukset CLion-ohjelmistokehitysympäristöön, työntekijän on suoritettava alla olevat kohdat numerojärjestyksessä:

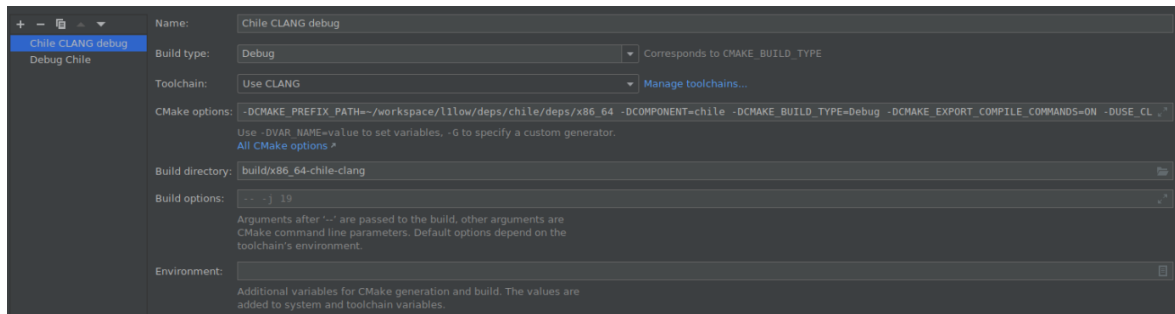
1. Työntekijän täytyy avata verkkoselain Linux-ympäristössä ja siirtyä osoitteeseen <https://confluence.ext.net.nokia.com/display/RFL1L/Clion+configuration+tips>.
2. Sivustolta on ladattava settings.zip-kansio, jonka jälkeen voi palata takaisin CLioniin.
3. Vasemmasta yläkulmasta täytyy klikata File-tekstiä ja viedä hiiri Manage IDE Settings -tekstin päälle.
4. Seuraavaksi on klikattava Import Settings -tekstiä, joka on oikealle puolelle avautuneessa valikossa.
5. Työntekijän täytyy valita settings.zip-kansio, joka löytyy Downloads-kansion sisältä, ja painaa OK-painiketta.

6. Lopuksi on valittava Select Components to Import -ikkunassa kaikki asetukset ja painettava OK-painiketta.

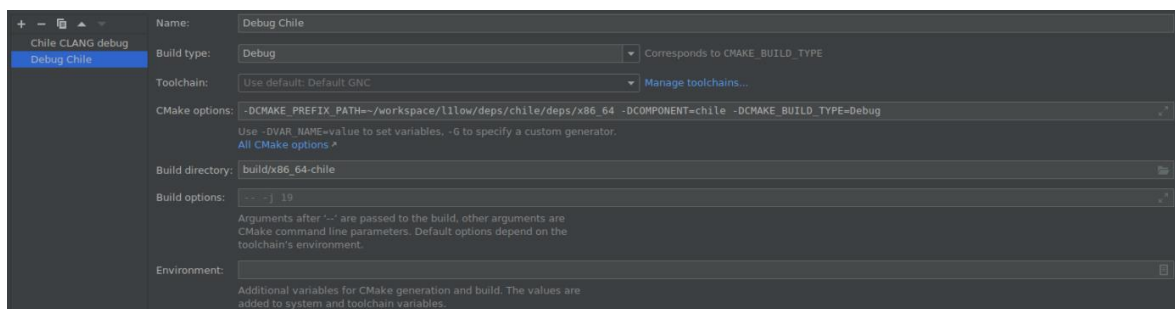
Päivittääkseen CMake-asetukset, työntekijän on suoritettava alla olevat kohdat numerojärjestyksessä:

1. Vasemmasta yläkulmasta on klikattava File-tekstiä ja avattava asetukset klikkaamalla Settings-tekstiä.
2. Tietokoneen näytölle avautuu uusi ikkuna, jonka vasemmalla puolella on Build, Execution, Deployment -otsikko. Työntekijän täytyy klikata otsikosta ja avata CMake-asetukset klikkaamalla listasta CMake-tekstiä.
3. Ensimmäinen profiili luodaan painamalla +-merkkiä CMake asetuksissa. Työntekijän täytyy lisätä Name-tekstikenttään Chile CLANG debug.
4. Build type -alasvetolaatikosta täytyy valita vaihtoehto Debug ja Toolchain-alasvetolaatikosta vaihtoehto Use CLANG.
5. CMake options -tekstikenttään on lisättävä merkkijono `-DCMAKE_PREFIX_PATH=~/workspace/l1low/deps/chile/deps/x86_64 -DCOMPONENT=chile -DCMAKE_BUILD_TYPE=Debug -DCMAKE_EXPORT_COMPILE_COMMANDS=ON -DUSE_CLANG=ON`. Kun merkkijono kopioidaan, täytyy huomioida, ettei sanoihin PREFIX ja COMMANDS tule tavuviivaa.
6. Build directory -tekstikenttään täytyy lisätä merkkijono `build/x86_64-chile-clang`.
7. Toinen asetusprofiili luodaan painamalla uudestaan +-merkkiä. Työntekijän täytyy lisätä Name-tekstikenttään Debug Chile.
8. Build type -alasvetolaatikosta täytyy valita taas vaihtoehto Debug, mutta kohdassa Toolchain on pidettävä se vaihtoehto, joka siinä on jo valmiina.
9. CMake options -tekstikenttään on lisättävä merkkijono `-DCMAKE_PREFIX_PATH=~/workspace/l1low/deps/chile/deps/x86_64 -DCOMPONENT=chile -DCMAKE_BUILD_TYPE=Debug`. Kun merkkijono kopioidaan, täytyy huomioida, ettei sanaan PREFIX tule tavuviivaa.
10. Build directory -tekstikenttään täytyy lisätä merkkijono `build/x86_64-chile`.

Kun työntekijä on suorittanut äskeiset kymmenen kohtaa, hänellä pitäisi olla sen jälkeen luotuna kaksi asetusprofiilia, jotka ovat samanlaiset, kuten kuvissa 7 ja 8. Lopuksi on painettava OK-painiketta, jotta asetukset tallentuvat. Tämän jälkeen työntekijä on valmis aloittamaan RF L1low -ohjelmiston kehittämisen.



Kuva 7. Ensimmäinen asetusprofiili



Kuva 8. Toinen asetusprofiili

3 VPN-yhteys

3.1 Autentikointiasetusten päivittäminen

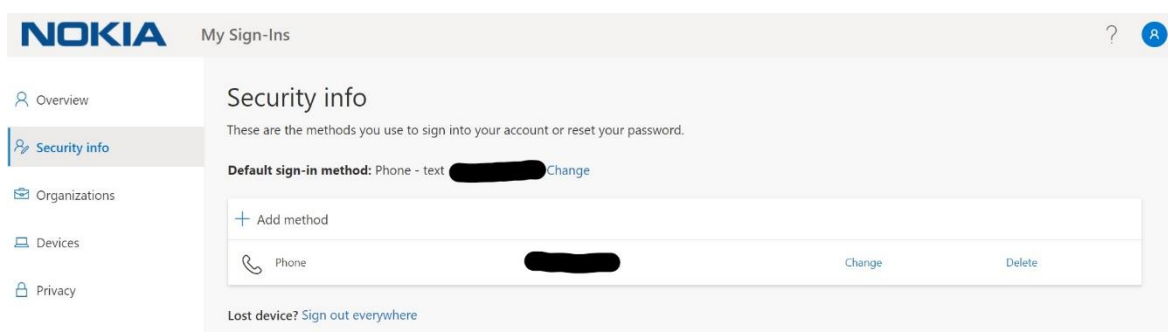
Nokia Oyj:n Oulun toimipisteellä pystytään kirjautumaan sisään LinSEE-palvelimelle ja saamaan VNC Viewer työkalulla näkymä Linux-käyttöjärjestelmästä. Tämä kuitenkin vaatii sen, että tietokone on yhdistetty Nokia Oyj:n omaan verkkoyhteyteen. Koska kyseistä verkkoyhteyttä ei ole saatavilla muualla kuin Nokia Oyj:n toimipisteellä, tarvitaan VPN-yhteys, jotta RF L1low -ohjelmiston kehittäminen on mahdollista myös etätyönä. (Laihiainen 2021.)

VPN-yhteys muodostetaan kirjautumalla Nokia-tunnuksilla ohjelmistoon nimeltä Cisco AnyConnect Secure Mobility Client ja se löytyy valmiiksi asennettuna tietokoneelta. VPN yhteyden muodostamiseen tarvitaan myös älypuhelinsovellus nimeltä Microsoft Authenticator. Sovellus on ladattavissa App Storesta iOS-laitteille ja Google Play Storesta Android-laitteille. Ennen kuin Microsoft Authenticator -sovellusta voidaan käyttää tunnistautumiseen ja VPN-yhteyden muodostamiseen, on ensin päivitettävä autentikointiasetukset. (Laihiainen 2021.)

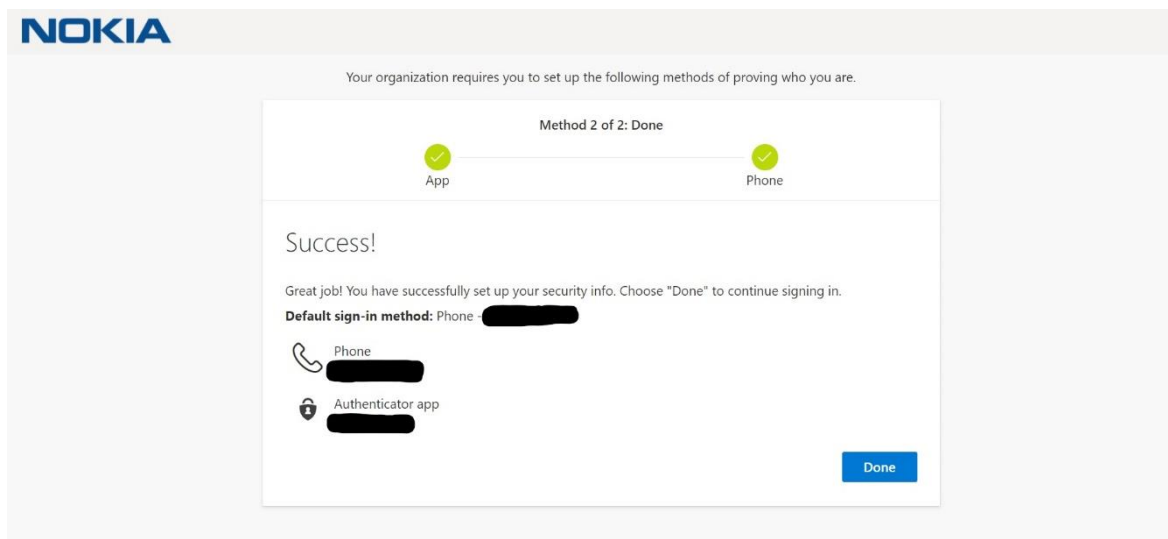
Päivittääkseen autentikointiasetukset, työntekijän on suoritettava alla olevat kohdat numerojärjestyksessä:

1. Työntekijän täytyy siirtyä osoitteeseen <https://mysignins.microsoft.com/security-info> ja kirjautua sisään Nokia-sähköpostilla ja -salasanalla. Tämän jälkeen hän päätyy kuvan 9 mukaiselle sivustolle.
2. Sivustolla täytyy klikata Add method -tekstiä, josta avautuu tietokoneen näytölle uusi ikkuna, jonka sisällä on alasvetovalikko.
3. Valikosta on valittava Authenticator app -vaihtoehto ja painettava Add-painiketta. Seuraavaksi työntekijää pyydetään lataamaan puhelimelle Microsoft Authenticator -sovellus.
4. Next-painiketta voidaan painaa siinä vaiheessa, kun työntekijä on asentanut puhelimelleen Microsoft Authenticator -sovelluksen.
5. Näkymässä, jonka otsikko on Set up your account, on painettava Next-painiketta.
6. Seuraavaksi on avattava Microsoft Authenticator -sovellus puhelimesta ja avattava QR-koodinlukija. Se, mistä tämä ominaisuus löytyy sovelluksessa, saattaa vaihdella käytössä olevan puhelimen ja julkaistun version perusteella.
7. Sovelluksen pyyntö käyttää puhelimen kameraa ja lähettää puhelimeen ilmoituksia täytyy hyväksyä, jotta työntekijä voidaan autentikoida käyttämään VPN-yhteyttä.

8. QR-koodi täytyy lukea tietokoneen näytöltä näkymässä Scan the QR code. Tämän jälkeen on painettava Next-painiketta.
9. Verkkosivuston lähettämä testi-ilmoitus täytyy hyväksyä puhelinsovelluksella. Tämän jälkeen tietokoneen näytölle ilmestyy vahvistusviesti suoritetusta toimenpiteestä.
10. Vahvistusviestin voi sulkea painamalla Next-painiketta ja lopuksi on päivitettävä verkkosivu. Tietokoneen näytölle ilmestyy kuvan 10 mukainen ilmoitus, että tietoturva koskevat asiat on päivitetty onnistuneesti.



Kuva 9. Autentikointi-asetusten päivityksiä koskevan sivuston näkymä



Kuva 10. Tietoturvan päivitysilmoitus

Jos työntekijä on tilanteessa, jossa hän voi lukea QR-koodia puhelimellaan, Microsoft Authenticator -sovellukseen täytyy kirjautua sisään Nokia-tunnuksilla. Myös tällä tavalla Microsoft Authenticator -sovellus saadaan käyttöön autentikointia varten.

Jos työntekijä ei onnistu lukemaan QR-koodia, hänen on suoritettava alla olevat kohdat numerojärjestyksessä, päivittääkseen autentikointiasetukset:

1. Microsoft Authenticator -sovellus on avattava puhelimella ja painettava Lisää työ- tai koulutili -napista.
2. Kirjaudu sisään -näkyvässä täytyy syöttää Nokia-sähköpostiosoite ja painaa Seuraava-painiketta.
3. Tämän jälkeen työntekijä päätyy Nokia Oyj:n kirjautumissivulle, jossa täytyy syöttää Nokia-tunnukset ja painaa Kirjaudu sisään -painiketta.
4. Työntekijä saa tekstiviestillä vahvistuskoodin, joka on syötettävä tekstikenttään, joka on Anna koodi -näkyvässä. Tämän jälkeen on painettava Tarkista-painiketta.
5. Seuraava-nappia on painettava näkyvässä, jonka otsikko on Lisätietoja tarvitaan.
6. Näkyvässä, jonka otsikko on Pidä tilisi suojattuna, on painettava oikeasta alakulmasta Ohita määrittely -tekstiä.
7. Näkyvässä, jonka otsikko on Tili lisättiin, täytyy painaa Valmis-nappia.
8. Lopuksi on päivitettävä verkkosivu, jonka jälkeen työntekijä saa tietokoneen näytölle ilmoituksen, että hän on onnistuneesti päivittänyt tietoturvaan liittyvät asiat. Ilmoitus on samanlainen, jos autentikointiasetukset päivitetäisiin lukemalla QR-koodi.

3.2 Vaihtoehdot sisäänkirjautumisen vahvistamiselle

Vaikka Microsoft Authenticator -sovellus on otettu käyttöön, osoitteessa <https://mysignins.microsoft.com/security-info> on vielä asetus päällä, joka lähettää vahvistuskoodin puhelimeen tekstiviestillä. Valittavissa on kaksi vaihtoehtoa, kuinka sisäänkirjautuminen vahvistetaan Microsoft Authenticator -sovelluksella, kun muodostetaan VPN-yhteys. Vaihtoehdot ovat valittavissa alavetovalikosta, joka tulee näkyviin painettaessa tietokoneen hiirellä sinistä Change-tekstiä. Työntekijän on valittava toinen vaihtoehtoista, jotka ovat esitely alapuolella, ja painettava Confirm-painiketta.

Ensimmäistä Microsoft Authenticator – notification -vaihtoehtoa käytettäessä sisäänkirjautuminen VPN-ohjelmistoon vahvistetaan ilmoituksella, joka tulee sovellukseen. Kun

työntekijä avaa sovelluksen, puhelimenäytölle ilmestyy ilmoitus, jossa häntä pyydetään hyväksymään sisäänkirjautuminen. VPN-yhteys muodostetaan heti, kun sisäänkirjautuminen hyväksytään. Jos työntekijä ei saa minkäänlaista ilmoitusta, vaikka on tämä vaihtoehto valittuna, hänen on tarkistettava puhelimen asetuksista, että Microsoft Authenticator -sovellus saa lähettää puhelimeen ilmoituksia.

Toista Authenticator app or hardware token – code -vaihtoehtoa käytettäessä, sisäänkirjautuminen vahvistetaan kuusinumeroisella koodilla, joka syötetään VPN-yhteyden tarjoavaan Cisco AnyConnect Secure Mobility Client -ohjelmistoon. Työntekijä löytää koodin avaamalla Microsoft Authenticator -sovelluksen ja painamalla etusivulta Nokia-sähköpostiosoitettaan. Koodi on voimassa vain 30 sekuntia, jonka aikana se on syötettävä, jotta VPN-yhteys muodostetaan. Jos koodia ei ehditä antaa tarpeeksi nopeasti, se vanhentuu, jonka jälkeen sovellus generoi uuden koodin. Koodin syöttämisen jälkeen, VPN-yhteys on käytettävissäsi.

3.3 VPN-yhteyden muodostaminen

Kun työntekijä on ladannut älypuhelimelleen Microsoft Authenticator -sovelluksen ja päivittänyt autentikointiasetukset, hän on valmis muodostamaan VPN-yhteyden. Yhteyden muodostaminen riippuu siitä, kumpi kahdesta vaihtoehdoista sisäänkirjautumisen vahvistamiseksi työntekijällä on käytössä. Jos työntekijä ei ole vielä valinnut jompaakumpaa vaihtoehtoa, hänen on palattava lukuun 3.2 ja valittava itselleen sopivin vaihtoehto.

Muodostaakseen VPN-yhteyden tietokoneeseen, työntekijän on suoritettava alla olevat kohdat numerojärjestyksessä:

1. Tietokoneella on avattava Cisco AnyConnect Secure Mobility Client -ohjelmisto.
2. Alasvetovalikosta on valittava Global – Finland – Helsinki -vaihtoehto ja painettava Connect-painiketta.
3. Näytölle avautuu uusi ikkuna, jossa täytyy syöttää Nokia-sähköpostiosoite ja -salausana niille tarkoitettuihin tekstikenttiin.
4. Lopuksi on hyväksyttävä sisäänkirjautuminen Microsoft Authenticator -sovelluksella.

VPN-yhteys muodostuu heti, kun sisäänkirjautuminen on hyväksytty. Jos yhteys halutaan katkaista, se onnistuu painamalla Disconnect-painiketta. Yhteys katkeaa myös silloin, kun tietokone sammutetaan.

Lähteet

Getting Started Guide, New Windows PC Hardware. 2017. Nokia Oyj. Viitattu 17.6.2021.
Saatavissa vain Nokia Oyj:n työntekijöiden sisäiseen käyttöön

Impola, L. 2021. Insinööri. Nokia Oyj. Haastattelu 5.7.2021.

Laihiainen, J. 2021. Insinööri. Nokia Oyj. Haastattelu 2.8.2021.

Parker, T. 1998. Linux Unleashed Third Edition. Carmel: Sams Publishing.

Rantala, A. 2003. Linux. Jyväskylä: Docendo Oy Finland.

Savusalo, T. 2021. Insinööri. Nokia Oyj. Haastattelu 2.8.2021.