

Opinnäytetyö (AMK)

Tieto- ja viestintäteknikka

2023

Matias Tieranta

# NetBox-ohjelmiston käyttönottaminen yrityksessä



Opinnäytetyö (AMK) | Tiivistelmä

Turun ammattikorkeakoulu

Tieto- ja viestintätekniikka

Toukokuu 2023 | 24 sivua

Matias Tieranta

## NetBox-ohjelmiston käyttöönottaminen yrityksessä

IT-alalla dokumentaation merkitys on kasvanut vuosi vuodelta. Dokumentaation tulee olla keskitettyä ja helposti löydettävissä. IT-infrastruktuurin dokumentaation kohdalla kuitenkin ollaan vielä lähtökuopissa.

Tämän opinnäytetyön tavoitteena oli tutkia yritykselle käyttöön tulevaa NetBox-ohjelmistoa ja selvittää miten alustan tarjoamaa NAPALM-tukea voitaisiin hyödyntää yrityksen tarpeisiin.

Dockeria ja virtuaalikonetta vertaillen tärkeimmiksi kohdiksi havaittiin Dockerin olevan kevyempi resurssi vaatimuksiltaan ja nopeampi asennuksissa. Virtuaalikone puolestaan oli resurssien näkökulmasta raskas ja asennus oli hidasta, mutta tieturvan näkökulmasta virtuaalikone on turvallisempi ilman muutoksia heti asennuksen jälkeen.

Opinnäytetyö toteutettiin asentamalla NetBox-ohjelmisto käyttäen Dockeria. Asennuksen jälkeen huomattiin, että alkuperäisestä Docker-imagesta puuttui tarvittavat ajurit ja imagea piti laajentaa. Näiden korjausten avulla niin NetBox kuin NAPALM saatiin toimimaan odotetusti.

Opinnäytetyö oli kokonaisuutena onnistunut ja alussa asetettuihin tavoitteisiin päästiin. Työstä saatiin hyvä käsitys siitä, miten NetBox-ohjelmistoa voidaan käyttää ja hyödyntää yrityksen toiminnassa nyt ja tulevaisuudessa.

Asiasanat:

NetBox, NAPALM, Docker

Bachelor's Thesis | Abstract

Turku University of Applied Sciences

Degree programme

2023 | 24 pages

Matias Tieranta

## Implementing Netbox software in a company

In the IT industry, the importance of documentation has grown year by year. Modern documentation should be centralized and easy to find. However, when it comes to IT infrastructure documentation, we are still at the starting point.

The aim of this thesis was to study the usage and viability of NetBox, and to investigate how its built-in NAPALM support could be used to fulfill company requirements.

When comparing Docker to a virtual machine, it was found that crucially Docker is a lighter resource in terms of requirements and faster in installations. The virtual machine, on the other hand, was heavy from the resources point of view and installation was slow, but from a security standpoint the virtual machine is more secure by default after first installation.

The thesis was overall a success, and set goals were achieved. The study supplied a good understanding of how NetBox can be used and utilized in the company's operation now and in the future.

Keywords:

NetBox, NAPALM, Docker

# Sisältö

<b>Käytetty sanasto</b>	<b>6</b>
<b>1 Johdanto</b>	<b>7</b>
<b>2 Työssä käytetyt ohjelmistot ja työkalut</b>	<b>8</b>
2.1 Docker-alusta	8
2.2 Docker vs. virtuaalikone	9
2.2.1 NetBox	11
<b>3 NetBoxin asennus ja käyttöönottaminen</b>	<b>14</b>
3.1 NetBox Docker -imagen asentaminen	14
3.2 NAPALM-kyvykkyyden lisääminen kytkimelle	17
3.3 Kytkimen ajurin lisääminen Docker-imageen	17
3.4 Prometheus-metriikoiden salliminen	19
<b>4 Tulokset ja pohdinta</b>	<b>21</b>
<b>5 Yhteenveto</b>	<b>22</b>
<b>Lähteet</b>	<b>24</b>

## Kuvat

Kuva 1. Kontti vs. virtuaalikone.	10
Kuva 2. Kuvaus NAPALMista.	13
Kuva 3. docker-compose.override.yml.	15
Kuva 4. Onnistunut Docker-kontin käynnistys.	15
Kuva 5. NetBox käyttöliittymä.	16
Kuva 6. Käyttäjän luominen kytkimellä NAPALMia varten.	17
Kuva 7. Status-välilehti virheilmoituksella.	18
Kuva 8. Jatkettu Dockerfile.	19
Kuva 9. Muokattu docker-compose.yml.	19

Kuva 10. Toimivan NAPALMin näkymä.

19

Kuva 11. NetBoxin metriikka.

20

## Käytetty sanasto

Lyhenne	Lyhenne selitettynä
Docker-image	Valmis paketti, joka sisältää ohjelman ja sen tarvitsemat riippuvuudet
Docker-kontti	Docker-imagen pohjalta käynnistetty sovellus
Hypervisor	Mahdollistaa usean virtuaalikoneen suorittamisen eri käyttöjärjestelmillä yhdellä serverialustalla
IP-osoite	Numeerinen osoite, joka identifioi laitteet verkossa
NAPALM	<i>Network Automation and Programmability Abstraction Layer with Multivendor support</i> . Python-kirjasto, joka mahdollistaa eri laitevalmistajien laitteiden kanssa kommunikoinnin käyttäen yhtenäistä API-rajapintaa.
REST API	<i>Representational state transfer</i> . Rajapinta-arkkitehtuuri. <i>Application programming interface</i> . Ohjelmointirajapinta. Rajapintamääritelmä, jonka avulla ohjelmat keskustelevat keskenään.
Valvonta	Havainnoi ja seuraa ohjelmistojen ja laitteistojen tilaa ja suorituskäykyä
Virtuaalikone	Virtuaalinen tietokone, joka toimii kuin fyysinen tietokone

# 1 Johdanto

Dokumentaatio on kriittisessä osassa IT-alalla. Yrityksellä tulee olla käsitys mitä laitteistoa sillä on käytössä niin työntekijän kuin IT-infrastruktuurinkin tasolla. Yleensä IT-alalla etenkin infrastruktuuri on kuvattuna Excel-taulukoissa, jotka voivat olla vaikeasti luettavissa tai ymmärrettävissä. Lisäksi dokumentaatiota voi olla ripoteltuna moneen paikkaan ja on vaikeasti löydettävissä.

Tämän opinnäytetyön aiheena on tarkastella yritykselle käyttöön tulossa olevaan NetBoxiin ja sen mahdollistamaan NAPALM-tukeen. Lisäksi halutaan selvittää, että onko yrityksellä käytössä olevaan laitteistoon mahdollista lisätä tukea Prometheus-valvontaan NetBoxin välityksellä.

Työtä aloittaessa yrityksen dokumentaatiotaso oli hyvä, mutta sitä oli ripoteltu monelle eri alustalle. Suurin osa dokumentaatiosta on tehty perinteisille Excel-taulukoille, joita päivittää työntekijät. Excel-dokumentaatio muodostuu ongelmalliseksi, kun tiedostoja pitää olla rinnakkain useampi ja osa tiedostoista pääsee vanhentumaan.

Tavoitteena oli muodostaa käsitys NetBoxin tarpeellisuudesta, sen mahdollisista rajoituksista ja siitä, mitä sen käyttöönotossa tulisi huomioida. Lisäksi tavoitteena on selvittää, miten NetBoxista saadaan metriikkaa yrityksen Prometheus-valvontaa varten.

NetBoxista on Milla Karppinen tehnyt opinnäytetyön Kajaanin ammattikorkeakoulussa ”NetBox-sovelluksen hyödyntäminen yrityksen dokumentoinnissa”. (2022) Opinnäytetyö käsittelee yrityksen dokumentaatiota.

Työssä asennettiin NetBox Debianille ja tutustumalla sen käyttöönottoon. Lisäksi työssä oli käytettävänä yritykseltä lainaan saatu FiberStoren kytkin, jonka avulla voitiin testata NAPALMin toimivuutta ja käytännöllisyyttä.

## 2 Työssä käytetyt ohjelmistot ja työkalut

Kappaleessa käsitellään ja tutustaan Dockerin ja virtuaalikoneen keskeisiin eroihin ja miten valita kumpaa tulisi käyttää. Lisäksi esitellään työssä asennettavaa NetBoxia ja esitellään sen toimintaa.

### 2.1 Docker-alusta

Docker on avoimen lähdekoodin alusta, joka käyttää konttitekniologiaa. Dockerin avulla voidaan tuottaa, ajaa ja toimittaa sovelluksia. Käyttämällä Dockeria voidaan sovellus pakata niin, että siihen on valmiiksi sisällytetty tarvittavat riippuvuudet, eli ohjelmiston osat, jotka ovat riippuvaisia toisistaan, ja sen asetukset. Paketoinnin aikana saadaan vakioitua sovellus, joten se tulee toimimaan odotetulla tavalla, vaikka kohde vaihtuisikin. (IBM n.d)

Docker-sovelluksen käyttöönotto on suoraviivaista. Usein tarvittava Docker-image löytyy jo valmiina. Se sisältää sovelluksen, tarvittavat riippuvuudet, ja sen voi ladata [hub.docker.com](https://hub.docker.com)-osoitteesta. Mikäli kuitenkin valmista imagea ei ole saatavissa, tulee sellainen luoda.

Useimmiten Docker-image sisältää halutun ohjelmiston koodin, tarvittavat riippuvuudet ja tiedostot. Ohjelman sisältä voidaan löytää monia eri kerroksia. Kerroksilla tarkoitetaan Dockerin tapaa jaotella sen sisältämää lähdekoodia ja tarvitsemia asetuksia, sekä riippuvuuksia. Jokainen kerros on itsenäinen osa Docker ohjelmistoa. Docker-kontti rakennetaan kerrostaen alhaalta ylöspäin.

Imagen kerrostuksella voidaan varmistaa, että asennus onnistuu nopeasti ja mutkattomasti. Kerrostusta tehdessä on tärkeää tiedostaa, että kerrosten hierarkialla on suuri merkitys. Useasti muokattavien kerroksien sijainti tulee olla Dockerfilen viimeisillä riveillä. Dockerfile on tiedosto, joka kertoo Dockerille, miten image tulee rakentaa. (C 2023.)



Docker-imagen rakennuksen aikana jokainen rivi Dockerfilessä tarkoittaa yhtä uutta kerrosta. Kun Dockerfile on rakennettu oikeaoppisesti, ei mene aikaa hukkaan muuttumattomien kerrosten uudelleenrakentamisessa. (Chece 2023.)

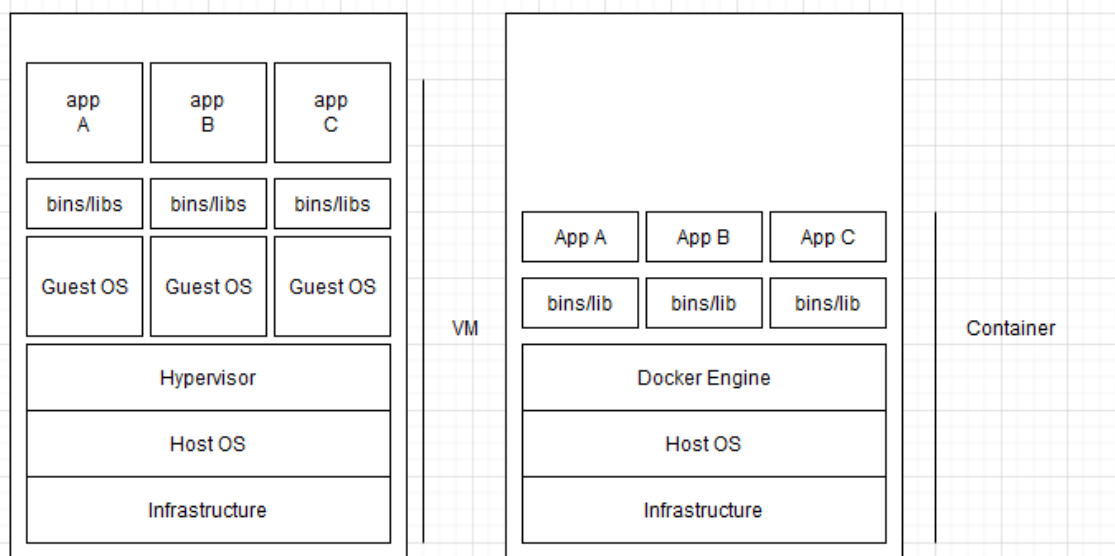
## 2.2 Docker vs. virtuaalikone

Aloitettaessa uuden ohjelmiston asentamista tulee aina harkita, onko tarpeellista tehdä sovellukselle oma virtuaalikone vai riittääkö mahdollisesti vain Docker-kontti, eli imagen pohjalta rakennettu sovellus. Molemmille on aikansa ja paikkansa.

Karkeasti sanottuna päätöksen voi tehdä sillä, kuinka usein on tarvetta tehdä muutoksia kyseisessä sovelluksessa. Mikäli sovellus tulee olemaan staattinen ja siihen ei haluta tehdä muutoksia, usein on järkevämpää mennäkö perinteisellä virtuaalikoneella. Mikäli sovellusta halutaan muokata, ja se on aktiivisessa kehityksessä, kannattaa valita Docker. (Arora 2022.)

**Virtuaalikone** sisältää kuusi kerrosta: palvelimen, palvelimen käyttöjärjestelmän, hypervisorin, virtuaalikoneen käyttöjärjestelmän, tarvittavat riippuvuudet ja itse sovelluksen. (Kuva 1.)

**Docker** sisältää viisi kerrosta: palvelimen, palvelimen käyttöjärjestelmän, Docker enginen, tarvittavat riippuvuudet ja itse sovelluksen. (Kuva 1.)



Kuva 1. Kontti vs. virtuaalikone.

Dockerin suurimpana etuna vertailtaessa virtuaalikoneeseen on, että Docker ei tarvitse käyttöjärjestelmää virtuaalisointiin, sillä se jakaa palvelimen käyttöjärjestelmän. Näin voidaan pienentää huomattavasti resurssien käytön tarvetta, käynnistymisen hitautta ja vähentää yrityksen taloudellisia menoja, sillä Dockerissa ei tarvitse maksaa lisenssimaksuja jokaisesta uudesta kontista.

Mikäli tarkoituksena on ajaa vain yhtä sovellusta palvelimella, voi olla helpointa käyttää virtuaalikonetta. Lisäksi joissain yrityksissä luotetaan virtuaalikoneisiin enemmän, sillä ne ovat heti asennuksen jälkeen tietoturvaisempia ilman sen suurempia muutoksia. (Aurora 2022.)

Vertailtaessa Dockeria ja virtuaalikonetta on hyvä ottaa huomioon muutama avaintekijä molemmista teknologioista.

Virtuaalikoneen käyttöönotossa on hyvä huomioida, että asentaminen on aikaa vievää ja kooltaan tiedostot voivat olla suuriakin verrattuna Dockeriin. Lisäksi virtuaalikoneen käynnistäminen vie pidempään kuin Docker-kontin pystyttäminen. Molempia vertaillessa Dockerin kohdalla puhutaan kymmenistä sekunneista, kun virtuaalikoneen asentaminen ja käynnistäminen vie minutteja. (Aurora 2022.)

Tietoturvan näkökulmasta virtuaalikoneissa palvelimen käyttöjärjestelmää ei ole jaettu virtuaalikoneen kanssa. Tietoturvallisemman tästä tekee se, että vaikka itse palvelimen käyttöjärjestelmässä olisikin haavoittuvuus niin sieltä ei ole suoraa pääsyä esimerkiksi toiseen virtuaalikoneeseen. Docker-konteilla on oletuksena pääsy muihin samalla palvelimella käynnissä oleviin kontteihin, joka heikentää tietoturvaa. (C 2022)

Valmiin virtuaalikoneen siirrettävyyden kannalta virtuaalikone erottuu edukseen. Kone voidaan siirtää kokonaisuudessaan mukaan lukien sen sisältämät tiedostot ja ohjelmat. Dockerissa siirrettävyyttä ei sinänsä tunneta, sillä kontteja ei ole suunniteltu siirrettäväksi. Uuden palvelinalustan kohdalla Docker-kontti uudelleenrakennetaan, ja kun vanhaa ei enää tarvita, se puretaan. (C 2022)

Mikäli palvelimella on tarpeellista virtualisoida montaa eri käyttöjärjestelmää, tulee valita virtuaalikone. Dockerissa ei ole mahdollista virtualisoida kuin UNIX-pohjaisia käyttöjärjestelmiä. Tämä johtuu siitä, että palvelimen käyttöjärjestelmä on jaettu Dockerin kanssa, ja esimerkiksi Debian-palvelimella ei voida suorittaa Windows-kontteja. (C 2022)

### 2.2.1 NetBox

NetBox on avoimenlähdekoodin ohjelmisto, jolla voidaan keskistetysti dokumentoida yrityksen IT-infrastruktuuria, IP-osoitteiden hallintaa ja dokumentoida datakeskusten laitteistonhallintaa.

NetBox on alun perin Jeremy Strechin kehittämä. Idea sovelluksen kehittämiseen lähti IT-infrastruktuuritiimin omasta aloitteesta, perustuen heidän tarpeisiinsa niin infrastruktuurin kuin tietoverkkojenkin dokumentaation osalta. Sovelluksesta tehtiin avoimen lähdekoodin projekti vuonna 2015. (Strech 2022)

Sovellus perustuu avoimeen lähdekoodiin, mutta sen kehityksestä vastuuseen on nimetty NS1:n työntekijä Jeremy Strech. Hänen vastuullaan on kaikki koodimuutokset. (NetBox Documentation n.d.d)

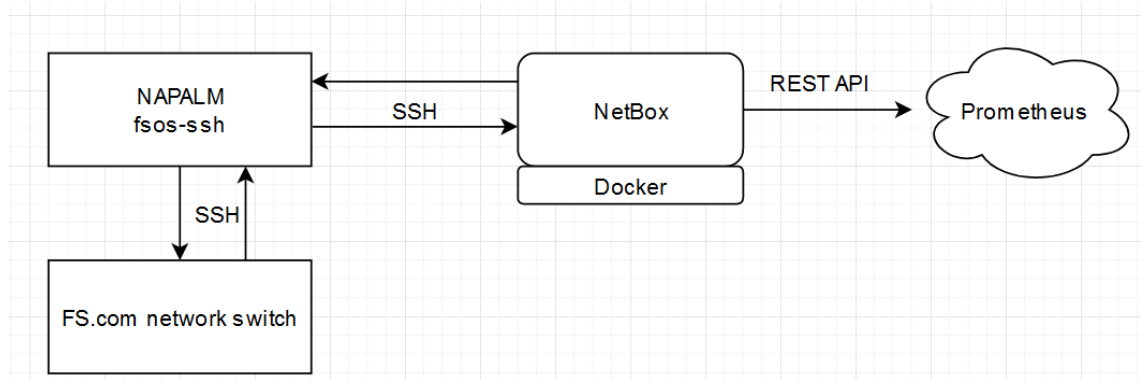
NetBox tarjoaa laajalti mahdollisuuksia IT-infrastruktuurin dokumentaation ja IP-osoitteiden hallintaan. Tässä työssä huomio kiinnittyi lähinnä IP-osoitteiden ja laitteiston hallintaan.

IP-osoitteiden hallinnalla voidaan keskitetysti dokumentoida yrityksen käytössä olevat verkot, sekä mitkä osuudet niistä on jo käytössä. NetBoxista voidaan tarkastaa, että palvelimelle suunniteltu IP-osoite, jolla numeraalisesti identifioidaan laite verkossa, on käytettävissä, eikä samaa IP-osoitetta käytetä vahingossa kahteen kertaan. (NetBox Documentation n.d.b)

Verkkolaitteiston dokumentaatiolla voidaan varmistua yrityksen käytössä olevasta laitteistosta. Kriittisen tästä tekee se, että konesalit voivat sijaita pitkienkin matkojen päässä, jopa toisessa maassa. Yrityksen on tärkeä tietää mahdollisten rikkoutumisien varalta, että mikä laite on hajonnut ja onko esimerkiksi konesalissa varalaitetta. (*NetBox Documentation n.d.c*)

NetBox tarjoaa lisäksi tuen REST API:lle, joka mahdollistaa esimerkiksi laitteiston lisäämisen yrityksen valvontaympäristöön. Lisäksi API:n avulla voidaan niin muokata kuin hakea NetBoxin sisältöä. API on rajapintamääritelmä, jonka avulla ohjelmat keskustelevat keskenään. REST on yksi lukuisista rajapinta-arkkitehtuureista. (*NetBox Documentation n.d.a*)

NetBox tarjoaa tukea NAPALMin käyttöön. NAPALM (Network Automation and Programmability Abstraction Layer with Multivendor support) on Python-kirjasto, joka mahdollistaa eri laitevalmistajien ja käyttöjärjestelmien kommunikaation NetBoxin kautta käyttäen yhtenäistä API-rajapintaa. (Kuva 2.)



Kuva 2. Kuvaus NAPALMista.

## 3 NetBoxin asennus ja käyttöönottaminen

Kappaleessa käydään läpi mitä kaikkea tulee ottaa huomioon NetBoxin käyttöön ottamisessa ja sen asentamisesta käyttäen Dockeria. Lisäksi käydään läpi ja esitellään NetBoxin tarjoamaa tukea NAPALMille.

Opinnäytetyön tavoitteena on saada selkeä käsitys NetBoxin käyttöön ottamisesta ja sen mahdollistamasta tuesta NAPALMille ja miten siitä voidaan kerätä arvokasta metriikkaa.

### 3.1 NetBox Docker -imagen asentaminen

NetBoxin asennustavaksi valikoitiin Docker. Yrityksellä on kokemusta Docker-pohjaisista ohjelmistoista ja niiden ylläpitämisestä. Lisäksi samalla palvelimella tulee olemaan muitakin UNIX-pohjaisia käyttöjärjestelmiä ja ohjelmistoja.

NetBoxin asentaminen palvelimelle oli helppoa seuraamalla NetBox-communityn ohjeita. (netbox-community n.d.)

Aluksi kohdepalvelimelle ladataan ja asennetaan Docker ja Docker Compose. Tämän jälkeen luodaan haluttu asentamispolku ja ladataan projekti GitHubista. Latauksen valmistuttua palvelimella on uusi hakemisto nimeltä "netbox-docker".

"netbox-docker" -hakemistoon luodaan seuraavaksi "docker-compose.override.yml" -tiedosto, johon määritellään vakioidut parametrit. Tähän voidaan käyttää "tee" -komentoa, jolla voidaan luoda tiedosto, tallentaa se ja varmistaa, että se on luotu oikein. (Kuva 3.)

```

~/netbox/netbox-docker# tee docker-compose.override.yml <<EOF
version: '3.4'
services:
  netbox:
    ports:
      - 8000:8080
EOF
version: '3.4'
services:
  netbox:
    ports:
      - 8000:8080
~/netbox/netbox-docker#

```

Kuva 3. docker-compose.override.yml.

Viimeiseksi tulee ladata Docker-image palvelimelta komennolla "docker compose pull". Tämän valmistuttua voidaan kyseinen image käynnistää komennolla "docker compose up -d". Käyttämällä -d-vipua voidaan Docker-kontti käynnistää taustalla ja näin palvelimella voidaan suorittaa monta eri Docker-konttia samanaikaisesti. (Kuva 4.)

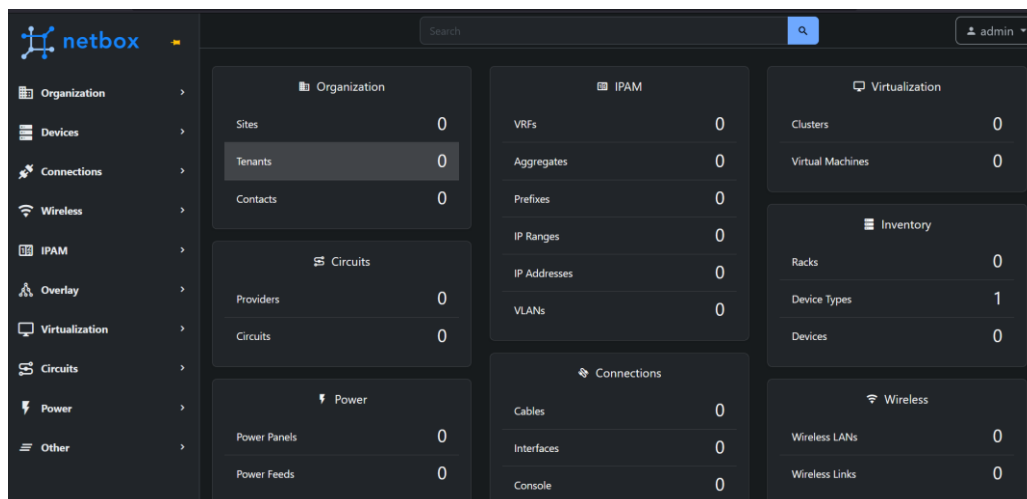
```

root@DESKTOP-1HHT08D:~/netbox/netbox-docker# docker-compose up -d
[+] Running 6/6
# Container netbox-docker-redis-cache-1      Started           0.9s
# Container netbox-docker-redis-1           Started           0.7s
# Container netbox-docker-postgres-1        Started           0.9s
# Container netbox-docker-netbox-1          Healthy          18.8s
# Container netbox-docker-netbox-worker-1   Started           19.4s
# Container netbox-docker-netbox-housekeep-1 Started           19.4s

```

Kuva 4. Onnistunut Docker-kontin käynnistys.

Onnistuneen Docker-kontin käynnistyksen jälkeen voidaan kirjautua ensimmäisen kerran NetBoxiin. Käyttäjätunnukset on luotu asennuksen aikana. Käyttäjätunnuksena toimii "admin" ja salasanana "admin". (Kuva 5.)



Kuva 5. NetBox-käyttöliittymä.

Onnistuneen sisäänkirjautumisen jälkeen tulee NetBoxiin luoda ensimmäinen laite laitteiston hallintaan. Ennen ensimmäisen laitteen lisäämistä NetBoxiin, tulee määrittellä seuraavat arvot:

- Sijainti ("Site")

Konehuoneen sijainti.

- Laitteen valmistaja ("Manufacturer")

Laitteiston valmistajan nimi

- Laitteen rooli ("Device Role")

Laitteiston roolit konehuoneessa. Näitä voivat olla esimerkiksi hallintakytkin, reititin ja palomuuuri.

- Laitteen tila ("Status")

Laitteen tila, onko laite käytössä, vai onko se esimerkiksi varalla mahdollisia rikkoontumisia varten.

Lisäksi määriteltiin:

- Alusta ("Platform")

Alustaksi tulee valita kytkimen käyttämä ajuri. Tässä tapauksessa "fsos\_ssh"



- Liitäntä ("Interface")

Laitteistolle tulee luoda sitä vastaavat portit.

- Ensisijainen IPv4 ("Primary IPv4")

Viimeiseksi tuli määrittää NetBoxiin kytkimen hallinta IP-osoite.

### 3.2 NAPALM-kyvykkyyden lisääminen kytkimelle

NAPALMin asentaminen tulee aloittaa kytkimeltä. Työssä käytettävät kytkimet ovat FiberStoren kytkimiä, joten ensimmäisenä tuli kirjautua kytkimelle ja varmistaa, että kytkimen käyttöjärjestelmän versio on vähintään 1.7.3 ajurin dokumentaation mukaisesti. (Ortega 2023)

Jotta kytkin voi toimia NAPALMin kanssa tulee siitä kytkeä SSH-palvelin päälle. Lisäksi kytkimelle tulee luoda käyttäjä NAPALMia varten (Kuva 6.)

Ajuria varten käyttäjällä tulisi olla ainakin lukuoikeudet. Työssä käytetyssä kytkimessä ei kuitenkaan ollut mahdollista käyttää alle pääkäyttäjätason oikeuksia, sillä silloin ei NAPALM ei toiminut. (Kuva 6.)

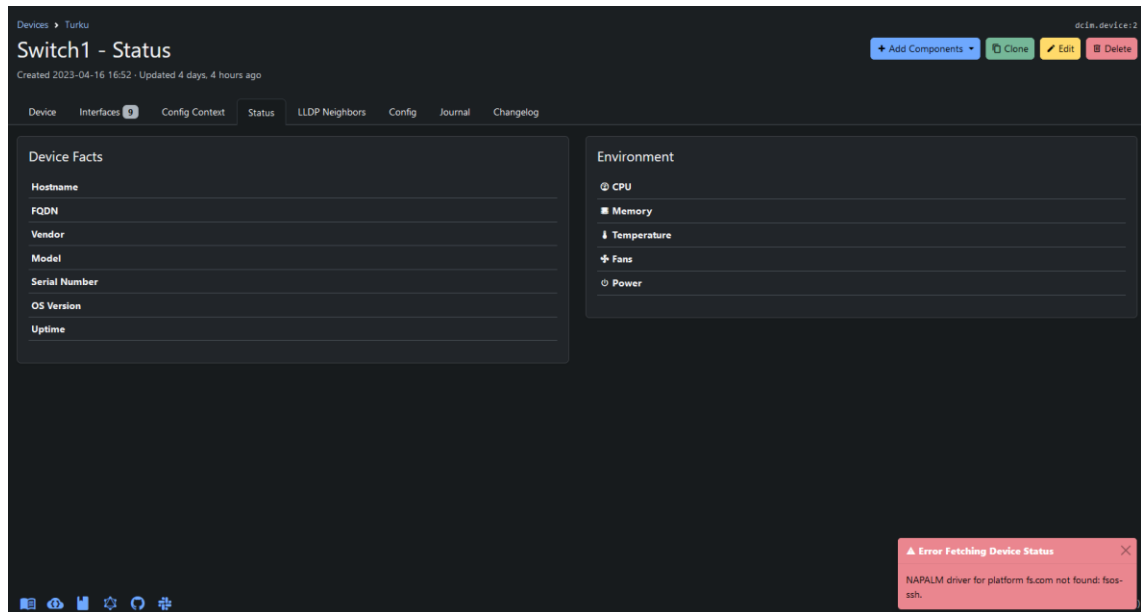
```
#ip ssh server enable
#username napalm password 0 napalmtest
#username napalm privilege 15
```

Kuva 6. Käyttäjän luominen kytkimellä NAPALMia varten.

### 3.3 Kytkimen ajurin lisääminen Docker-imageen

Onnistuneiden määritysten jälkeen voidaan NAPALMin toiminnan testaus aloittaa ensimmäistä kertaa. NetBoxissa valitaan juuri luotu laite ja sivun yläpalkista avataan Status-välilehti. Ongelmaksi muodostui kuitenkin se, että

kytkimelle tarvittu ajuri puuttui Docker-kontista ja kyseinen ajuri tulee asentaa erikseen. (Kuva 7.)



Kuva 7. Status-välilehti virheilmoituksella.

Ongelman korjaamiseksi tuli laajentaa Docker-imagea. Docker-imagen laajentamista varten luotiin uusi tyhjä Dockerfile, jossa määriteltiin haluttu pohjaimage ja kytkimen ajuri. (Kuva 8.)

Seuraavaksi tuli muokata docker-compose.yml -tiedostoa. Tiedoston neljäs rivi voitiin poistaa kokonaisuudessaan, sillä image haluttiin rakentaa juuri määritellyn Dockerfilen mukaisesti.

Build-määrittelyllä Docker-image rakennettiin käyttämällä samasta hakemistosta löytyvää Dockerfileä. Muita muutoksia docker-compose.yml -tiedostoon ei tarvinnut tehdä. (Kuva 9.)

Tämän jälkeen Docker-kontti rakennettiin uudestaan ja NAPALM toimi odotetusti. (Kuva 10.)

```

GNU nano 5.4
#Alkuperäinen NetBoxin docker image
FROM netboxcommunity/netbox:v3.4-2.5.2

#Kytkimelle tarvittavat ajurit
RUN pip3 install napalm-fsos-ssh

```

Kuva 8. Jatkettu Dockerfile.

```

version: '3.4'
services:
  netbox: &netbox
  build: .

```

Kuva 9. Muokattu docker-compose.yml.

Switch1 - Status  
Created 2023-04-16 16:52 - Updated 4 days, 4 hours ago

Device | Interfaces | Config Context | Status | LLDP Neighbors | Config | Journal | Changelog

Device Facts	
Hostname	SwitchA
FQDN	
Vendor	Fiberstore
Model	
Serial Number	CG2012245876N00049
OS Version	1.7.3
Uptime	16 days, 5 hours, 25 minutes, 24 seconds 2023-04-04 18:04:21 GMT+3 2007-04-04 16:04:21 UTC

Environment	
CPU	
Name Usage	
Memory	
Available Used	
214315008 322555904	
Temperature	
Sensor Value	
1 51 °C	
Fans	
Fan Status	
Power	
PSU Status	
1 ✓	

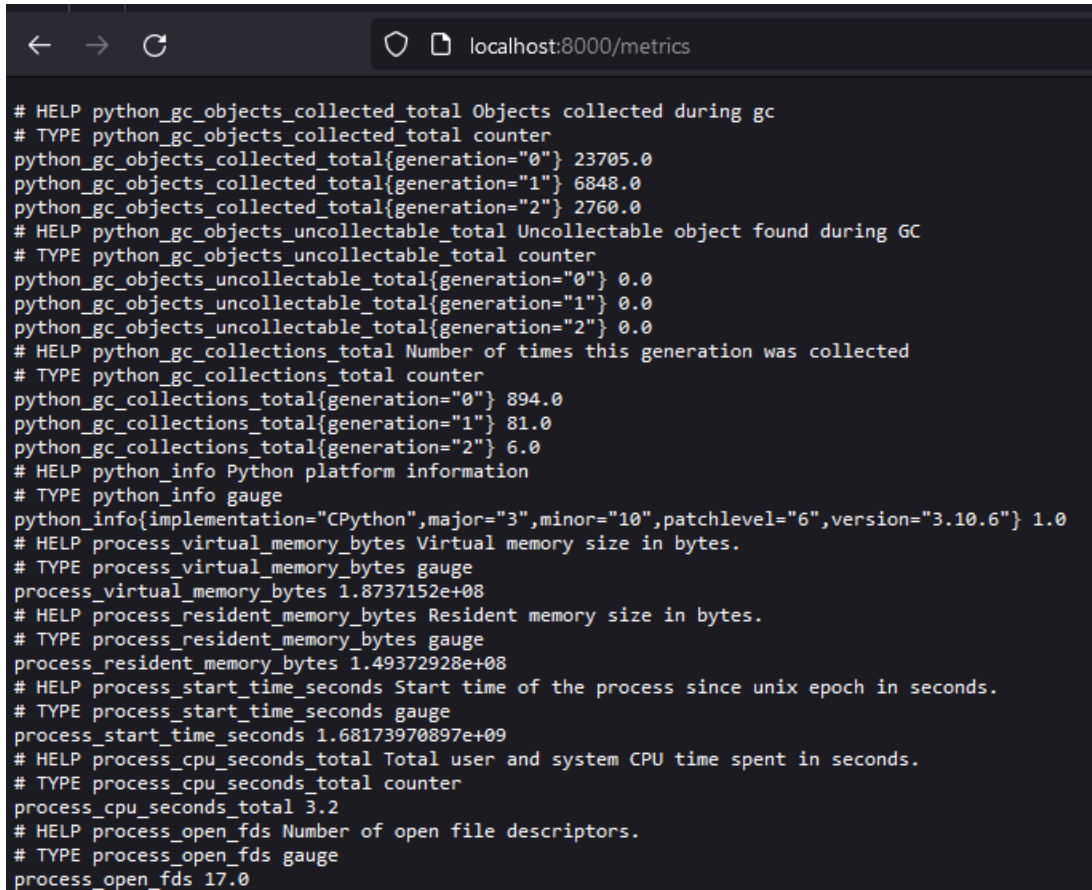
Kuva 10. Toimivan NAPALMin näkymä.

### 3.4 Prometheus-metriikoiden salliminen

Työn viimeisenä tehtävänä oli selvittää, miten mahdollistetaan kytkinten yhdistäminen yrityksen käyttämään Prometheus-valvontatyökaluun, jonka avulla kytkimiä voidaan valvoa. Valvonnalla voidaan laitteiston tilaa seurata ulkoisella työkalulla ja saada ilmoituksia, mikäli laitteistossa ilmenee ongelmia. Ominaisuus on sisäänrakennettu NetBoxiin, joten tämä oli suoraviivainen vaihe.

Tutkiessa NetBoxin configuration.py -tiedostoa voi nähdä, että metriikan käyttöönotto onnistuu lisäämällä netbox.env -tiedostoon muuttujan "METRICS\_ENABLED" arvolla "TRUE".

Tämän jälkeen Docker-kontti käynnistettiin uudestaan ja metriikat olivat saatavilla. (Kuva 11.)



```
# HELP python_gc_objects_collected_total Objects collected during gc
# TYPE python_gc_objects_collected_total counter
python_gc_objects_collected_total{generation="0"} 23705.0
python_gc_objects_collected_total{generation="1"} 6848.0
python_gc_objects_collected_total{generation="2"} 2760.0
# HELP python_gc_objects_uncollectable_total Uncollectable object found during GC
# TYPE python_gc_objects_uncollectable_total counter
python_gc_objects_uncollectable_total{generation="0"} 0.0
python_gc_objects_uncollectable_total{generation="1"} 0.0
python_gc_objects_uncollectable_total{generation="2"} 0.0
# HELP python_gc_collections_total Number of times this generation was collected
# TYPE python_gc_collections_total counter
python_gc_collections_total{generation="0"} 894.0
python_gc_collections_total{generation="1"} 81.0
python_gc_collections_total{generation="2"} 6.0
# HELP python_info Python platform information
# TYPE python_info gauge
python_info{implementation="CPython",major="3",minor="10",patchlevel="6",version="3.10.6"} 1.0
# HELP process_virtual_memory_bytes Virtual memory size in bytes.
# TYPE process_virtual_memory_bytes gauge
process_virtual_memory_bytes 1.8737152e+08
# HELP process_resident_memory_bytes Resident memory size in bytes.
# TYPE process_resident_memory_bytes gauge
process_resident_memory_bytes 1.49372928e+08
# HELP process_start_time_seconds Start time of the process since unix epoch in seconds.
# TYPE process_start_time_seconds gauge
process_start_time_seconds 1.68173970897e+09
# HELP process_cpu_seconds_total Total user and system CPU time spent in seconds.
# TYPE process_cpu_seconds_total counter
process_cpu_seconds_total 3.2
# HELP process_open_fds Number of open file descriptors.
# TYPE process_open_fds gauge
process_open_fds 17.0
```

Kuva 11. NetBoxin metriikka.

## 4 Tulokset ja pohdinta

Asennusvaiheessa kaikki sujui alussa ongelmitta. Docker-kontti saatiin pystyyn onnistuneesti seuraamalla ohjeita GitHubista. NAPALMia määriteltessä ongelmia alkoi muodostumaan. Alkuperäisessä Docker-imagessa ei ollut tarvittavia ajureita, joiden vuoksi NetBoxin Status-välilehti näytti virheilmoitusta. Ongelman korjaamiseksi Docker-imagea laajennettiin, sisällyttämällä siihen tarvittavat ajurit. Myös alkuperäistä docker-compose.yml tiedostoa muokattiin käyttämään laajennettua imagea. Korjausten jälkeen Docker-kontti rakennettiin uudestaan ja NAPALM toimi kuten oli oletettu.

Asennuksessa käytettävää prosessia voidaan pitää luotettavana ja oikeaoppisena. Kaikki käytetyt ohjeet ja lähteet ovat peräisin NetBoxin ja NetBox-communityn GitHub-dokumentaatiosta.

Ottaen huomioon kaikki työssä ilmenneet asiat voidaan silti todeta, että yritykselle on kannattavaa ottaa NetBox osaksi päivittäistä toimintaansa, kunhan jatkotutkimusta ollaan valmiita tekemään.

Oppinäytetyön pohjalta voidaan yrityksen IT-infrastruktuurin dokumentoida keskitetysti ja välttää dokumentaation pirstaloituminen monelle eri alustalle. Näillä toimilla voidaan minimoida dokumentaation hakemiseen käytettyä aikaa ja samalla varmistua, että ongelmatilanteissa ei mene aikaa hukkaan dokumentaation etsimiseen.

## 5 Yhteenveto

Opinnäytetyössä käsiteltiin uuden dokumentaatioalustan NetBoxin käyttöönottoa ja sen mahdollistamia ominaisuuksia.

Tavoitteena työssä oli muodostaa käsitys NetBoxin tarpeellisuudesta, sen mahdollisista rajoituksista ja siitä, mitä sen käyttöönotossa tulisi huomioida. Lisäksi tavoitteena oli selvittää, miten NetBoxista saadaan metriikkaa yrityksen Prometheus-valvontaa varten.

Tuloksena NetBox osoittautui hyväksi ja käytännölliseksi tavaksi dokumentoida IT-infrastruktuurin laitteiston hallintaa ja IP-osoitteiden dokumentaatiota. NetBox mahdollistaa nykyaikaisen tavan ylläpitää yrityksen IT-infrastruktuurin dokumentaatiota ja tekee siitä keskitettyä.

Onnistumisina voitiin nähdä, että jokainen tavoite täytettiin ja työ tarjosi selkeän näkemyksen, miten asennusvaiheessa tulee toimia. Työssä ongelmia aiheutti puuttunut ajuri.

Mahdollisina rajoitteina NetBoxin kanssa voidaan nähdä, että sitä tulee silti ylläpitää ihan vastaavalla tavalla kuin perinteistä Excel-dokumentaatiotakin. Vaikka dokumentaatio olisikin keskitettyä, tulee sitä pitää silti ajan tasalla. Lisäksi tulee varmistua dokumentaation oikeanlaisuudesta.

Toisena rajoitteena voidaan nähdä, että NetBoxin selkeästä ulkonäöstä huolimatta on sen hallinnointi haastavaa ja sen oppimiseen tulee menemään aikaa. NetBoxin käyttäminen ilman aikaisempaa kokemusta voi olla hankalaa.

Opinnäytetyön tuloksia voidaan soveltaa tulevaisuudessa yrityksen toiminnassa. NetBoxin avulla yrityksen dokumentaatio voidaan siirtää perinteisistä Excel-taulukoista keskitetysti NetBoxiin.

Mikäli NetBoxia halutaan käyttää yrityksen toiminnassa jokapäiväisellä tasolla, tulisi siihen tehdä vielä muutoksia tietoturvallisuuden näkökulmasta. Esimerkiksi NAPALMin käyttämät käyttäjätunnukset ovat alkuperäisessä konfiguraatiossa laitettuna env-hakemistoon ja ne löytyvät sieltä selkotekstillä ilman salausta.

Metriikan osalta on tärkeää varmistaa, että onko metriikka halutun mukaista ja sellaista, että siitä on oikeasti hyötyä.

## Lähteet

Arora, S. (2022) *Docker vs. Virtual Machines: Differences You Should Know*. 2022.

Viitattu 22.4.2023

<https://cloudacademy.com/blog/docker-vs-virtual-machines-differences-you-should-know/>

C, B. P. (2022). *Docker vs Virtual Machine (VM) – Key Differences You Should Know*.

Viitattu 14.4.2023

<https://www.freecodecamp.org/news/docker-vs-vm-key-differences-you-should-know/>

Chece, J. (2023). *Understanding Docker Image Layers – Sweetcode.io*.

Viitattu 16.4.2023

<https://sweetcode.io/understanding-docker-image-layers/>

IBM. (n.d.). *What is Docker?*

Viitattu 10.4.2023

<https://www.ibm.com/topics/docker>

*Netbox-community* (n.d.) *netbox-docker*

<https://github.com/netbox-community/netbox-docker>

NetBox Documentation. (n.d.a) *API & Integration -*

Viitattu 22.4.2023

<https://docs.netbox.dev/en/stable/features/api-integration/>

*NetBox Documentation* (n.d.b) *Devices & Cabling*

Viitattu 21.4.2023

<https://docs.netbox.dev/en/stable/features/devices-cabling/>

NetBox Documentation. (n.d.c) *IPAM*

Viitattu 21.4.2023

<https://docs.netbox.dev/en/stable/features/ipam/>

NetBox Documentation (n.d.d) *Introduction*

Viitattu 20.4.2022

<https://docs.netbox.dev/en/stable/development/#governance>

Strech, J. (2022). *DigitalOcean Renews NetBox Sponsorship*.

Viitattu 19.4.2023

<https://netbox.dev/blog/posts/digitalocean-renews-sponsorship/>

Ortega, L. (2023). *napalm-automation-community/napalm-fsos-ssh: SSH-based driver for FSOS (related products: see fs.com)*.



Viitattu 13.4.2023

<https://Github.com/napalm-automation-community/napalm-fsos-ssh#readme>