

Labra 2

Palveluiden automatisointi

Markus Häkkinen

Ville Pulkkinen

Mikael Romanov

Laboratorioharjoitus

Helmikuu 2018

Tieto- ja viestintätekniikan koulutusohjelma

Tekniikan ja liikenteen ala

Sisälllys

1	Johdanto	3
2	Kontittaminen	4
2.1	Kontin erot virtuaalikoneeseen	4
3	Dockerin asennus	6
4	Palveluiden kontitus	10
4.1	Konttien luominen	10
4.2	Ntopng kontitus	11
4.3	Dockerfile	14
4.4	Elasticsearch kontitus	16
5	Docker Compose	21
5.1	Compose tiedoston luonti	21
6	Muutos-skenaario	21
7	Pohdinta	22
8	Lähteet	23

Kuviolista

Kuvio 1 Virtuaalikoneen konsepti	5
Kuvio 2 Kontin konsepti	5
Kuvio 3 Paketit	6
Kuvio 4 GPG	6
Kuvio 5 GPG Sormenjälki	7
Kuvio 6 Pakettilähteen lisäys	7
Kuvio 7 Apt-get update	7
Kuvio 8 Docker CE asennus	8
Kuvio 9 Docker version tarkistus.....	8
Kuvio 10 Docker versio	8
Kuvio 11 Dockerin testaus	9
Kuvio 12 Docker ps -a.....	9
Kuvio 13 Dockerhub.....	10
Kuvio 14 Docker pull	11
Kuvio 15 Docker images.....	11
Kuvio 16 Docker run.....	12
Kuvio 17 Isb-release	12
Kuvio 18 apt-get install wget	12
Kuvio 19 Deb paketti.....	13
Kuvio 20 Deb paketin asennus.....	13
Kuvio 21 apt-get update	13
Kuvio 22 ntopng paketit.....	14
Kuvio 23 Dockerfile ntop.....	15
Kuvio 24 Docker build	15
Kuvio 25 Docker images.....	15
Kuvio 26 Docker run ntopng	16
Kuvio 27 OpenJDK kontin lataus	16
Kuvio 28 Image IDn tarkistus	16
Kuvio 29 Kontin käynnistys	16
Kuvio 30 Perusohjelmien asennus	17
Kuvio 31 Elasticsearch asennus	17
Kuvio 323 Asetukset Elasticsearchille	18
Kuvio 334 Ympäristön muuttaminen uudeksi kontiksi	19
Kuvio 34 Elastic käynnistyy	20

1 Johdanto

Labran tavoitteena oli kontittaa aiemmin rakentamamme palvelut docker kontteihin. Docker kontit luotiin ensin käsin ja sen jälkeen käyttäen docker composea joka on useiden konttien ajamiseen käytettävä työkalu. Lopuksi kontteihin tuli suorittaa muutos-skenaariot muuttamalla dockerfileä tai ympäristömuuttujia konttien käynnistyksen yhteydessä.

2 Kontittaminen

Kontti on kevyt, erillinen, suoritettava osa ohjelmistoa joka sisältää kaiken tarvittavan sen suorittamiseen kuten: koodin, järjestelmätyökalut, kirjastot ja asetukset. Kontti toimii aina samalla tavalla riippuen sen isäntäympäristöstä. Konttien sisällä olevat sovellukset ovat eristetty muusta järjestelmän ympäristöstä (Kuvio 2 Kontin konsepti)

2.1 Kontin erot virtuaalikoneeseen

Virtuaalikoneilla on kokonainen käyttöjärjestelmä, mikä hallinnoi asennettua muistia ja asennettuja virtuaalisia asemia (Kuvio 1 Virtuaalikoneen konsepti). Virtuaalikoneen tärkeitä resursseja emuloi hypervisor ja käytössä oleva host käyttöjärjestelmä, mikä mahdollistaa useamman virtuaalikoneen samanaikaisen suorittamisen. Jokainen virtuaalikone toimii yksittäisenä kokonaisuutena eristettynä hostista.

Docker konteissa ei käytetä hypervisoria, vaan 'Docker moottorilla'. Kontit on paljon pienempiä kuin virtuaalikoneet. Kontit pystytään potkaisemaan nopeampaa käyntiin, joka edellyttää suorituskkyä ja vähentää eristämistä. Docker kontit edellyttää myös suurempaa yhteensopivuutta hosti koneen jakamisessa. Yksittäiseen palvelimeen voidaan pakata useampia kontteja, koska käyttöjärjestelmää ei duplikoida jokaiseen sovellukseen.

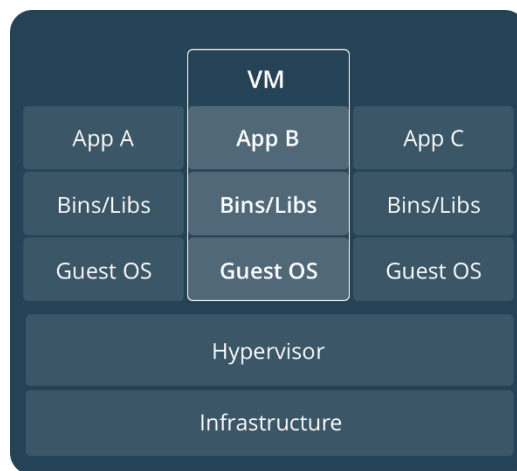
Docker kontiteissa on paljon enemmän potentiaalia kuin virtuaalikoneissa, sillä kontit voivat jakaa esim yhden ytimen ja sovelluskirjastoja. Sovellusten säilöminen konteissa on yleensä suorituskyyvyltään sama/parempi verrattuna virtuaalikoneeseen.

Docker on yhdessä asiassa virtuaalikoneita heikompi ja se on eristäminen. Intelin VT-d ja VT-x teknologiat toimittavat virtuaalikoneille ring-1 laitteisto eristykseen. Intelin teknologiat auttavat virtuaalikoneita hajoamasta ja aiheuttamasta häiriötä toisillensa. Docker kontit ei omaa laitteisto eristystä, mikä tekee niistä arkoja hyökkäyksille.

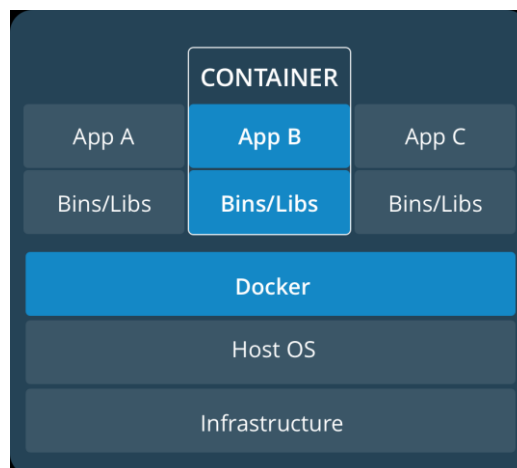
Virtuaalikoneisiin verrattuna kontit vievät vähemmän resursseja ja ovat usein nopeampia, niin kauan kun käyttäjä pysyy yhdellä alustalla. Usein virtuaalikoneiden luominen, kloonaminen ja käynnistäminen vie useita minutteja, kun kontit voidaan

luoda ja potkasta käyntiin vain muutamassa sekunnissa. Konteissa säilytetyt sovellukset/palvelut tarjoavat erinomaista suoristuskkyä vrt. virtuaalikoneen ajossa.

Dockerin mukaan konteissa olevat sovellukset voivat olla jopa 2 kertaa nopeampia kuin virtuaalikoneissa. Tosin virtuaalikoneita ja kontteja voidaan käyttää ns 'hybridinä', milloin saadaan molempien tuomat hyödyt ja samalla parantaa tehokkuutta. Virtuaalikoneet tuovat joustavuutta heterogeenisissä toimintaympäristöissä, kun taas docker kontit keskittyvät sovelluksiin ja niiden riippuvuuksiin.



Kuvio 1 Virtuaalikoneen konsepti



Kuvio 2 Kontin konsepti

3 Dockerin asennus

Dockerista on saatavilla 2 eri versiota, "Enterprise Edition" sekä "Community Edition". Docker Community Edition on suositeltu kehittäjillä sekä pienille tiimeille jotka haluavat oppia kontittamaan palveluitaan. Enterprise on maksullinen versio dockerista joka sisältää enemmän ominaisuuksia ja paremman tuen. Tässä työssä käytettiin Dockerin CE versiota Ubuntu 16.04 jakelun kanssa.

Docker voidaan asentaa pakettienhallinnasta, tai suoraan binääreistä. Ensiksi päivitämme pakettilistat ubuntuissa komennolla "apt-get update". Docker repositoryn lisäämistä varten pitää asentaa seuraavat paketit: "apt-transport-https", "ca-certificates", "curl" sekä "software-properties-common"(Kuvio 3 Paketit)

```
ubuntu@ubuntu:~$ sudo apt-get install \
> apt-transport-https \
> ca-certificates \
> curl \
> software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
apt-transport-https is already the newest version (1.2.24).
ca-certificates is already the newest version (20170717~16.04
.1).
curl is already the newest version (7.47.0-1ubuntu2.5).
software-properties-common is already the newest version (0.9
6.20.7).
0 upgraded, 0 newly installed, 0 to remove and 54 not upgrade
d.
ubuntu@ubuntu:~$
```

Kuvio 3 Paketit

Seuraavaksi pitää hakea ja asentaa dockerin virallinen GPG avain. Avaimella voidaan varmistaa pakettien oikeellisuus dockerin pakettihakemistossa (Kuvio 4 GPG)

```
ubuntu@ubuntu:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
ubuntu@ubuntu:~$
```

Kuvio 4 GPG

```
ubuntu@ubuntu:~$ sudo apt-key fingerprint 0EBFCD88
pub  4096R/0EBFCD88 2017-02-22
     Key fingerprint = 9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88
uid          Docker Release (CE deb) <docker@docker.com>
sub  4096R/F273FCD8 2017-02-22

ubuntu@ubuntu:~$
```

Kuvio 5 GPG Sormenjälki

Dockerin pakettihakemisto voidaan lisätä komennolla "sudo add-apt-repository \
hakemiston_nimi"(Kuvio 6 Pakettilähteen lisäys)

```
ubuntu@ubuntu:~$ sudo add-apt-repository \
> "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
> $(lsb_release -cs) \
> stable"
ubuntu@ubuntu:~$
```

Kuvio 6 Pakettilähteen lisäys

Nyt voidaan päivittää uusien pakettien sijainti komennolla "sudo apt-get
update"(Kuvio 6 Apt-get update)

```
ubuntu@ubuntu:~$ sudo apt-get update
Hit:1 http://security.ubuntu.com/ubuntu xenial-security InRelease
Hit:2 http://fi.archive.ubuntu.com/ubuntu xenial InRelease
Hit:3 http://fi.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:4 http://fi.archive.ubuntu.com/ubuntu xenial-backports InRelease
Get:5 https://download.docker.com/linux/ubuntu xenial InRelease [65.8 kB]
Get:6 https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages [3,150 B]
Fetched 69.0 kB in 0s (92.4 kB/s)
Reading package lists... Done
ubuntu@ubuntu:~$
```

Kuvio 7 Apt-get update

Nyt docker voidaan asentaa yksinkertaisesti komennolla "sudo apt-get install docker-
ce"(Kuvio 8 Docker CE asennus)


```
ubuntu@ubuntu:~$ sudo apt-get install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  aufs-tools cgroupfs-mount libltdl7
Suggested packages:
  mountall
The following NEW packages will be installed:
  aufs-tools cgroupfs-mount docker-ce libltdl7
0 upgraded, 4 newly installed, 0 to remove and 54 not upgraded.
Need to get 30.3 MB of archives.
After this operation, 149 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Kuvio 8 Docker CE asennus

Dockerin saatavilla olevat versiot voidaan tarkastella komennolla ”apt-cache madison docker-ce”(Kuvio 9 Docker version tarkistus)

```
ubuntu@ubuntu:~$ apt-cache madison docker-ce
docker-ce | 17.12.0~ce-0~ubuntu | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages
docker-ce | 17.09.1~ce-0~ubuntu | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages
docker-ce | 17.09.0~ce-0~ubuntu | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages
docker-ce | 17.06.2~ce-0~ubuntu | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages
docker-ce | 17.06.1~ce-0~ubuntu | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages
docker-ce | 17.06.0~ce-0~ubuntu | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages
docker-ce | 17.03.2~ce-0~ubuntu-xenial | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages
docker-ce | 17.03.1~ce-0~ubuntu-xenial | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages
docker-ce | 17.03.0~ce-0~ubuntu-xenial | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages
ubuntu@ubuntu:~$
```

Kuvio 9 Docker version tarkistus

Docker asensi automaattisesti uusimman version, joten meidän ei tarvitse päivittää sitä (Kuvio 10 Docker versio)

```
ubuntu@ubuntu:~$ docker version
Client:
 Version:      17.12.0-ce
 API version:  1.35
 Go version:   go1.9.2
 Git commit:   c97c6d6
 Built: Wed Dec 27 20:11:19 2017
 OS/Arch:     linux/amd64
```

Kuvio 10 Docker versio

Dockerin toimivuuden voi testata luomalla yksinkertaisen kontin joka tulostaa ruudulle ”Hello from docker” jos dockerin asennus on onnistunut (Kuvio 11 Dockerin testaus). Käynnissä olevia kontteja voi tarkastella komennolla ”docker ps -a”(Kuvio 12 Docker ps -a)

```
ubuntu@ubuntu:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
ca4f61b1923c: Pull complete
Digest: sha256:66ef312bbac49c39a89aa9bcc3cb4f3c9e7de3788c944158df3ee0176d32b751
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

Kuvio 11 Dockerin testaus

```
ubuntu@ubuntu:~$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e0351e9a2ddf	hello-world	"/hello"	9 minutes ago	Exited (0) 9 minutes ago		awesome_perlman

```
ubuntu@ubuntu:~$
```

Kuvio 12 Docker ps -a

4 Palveluiden kontitus

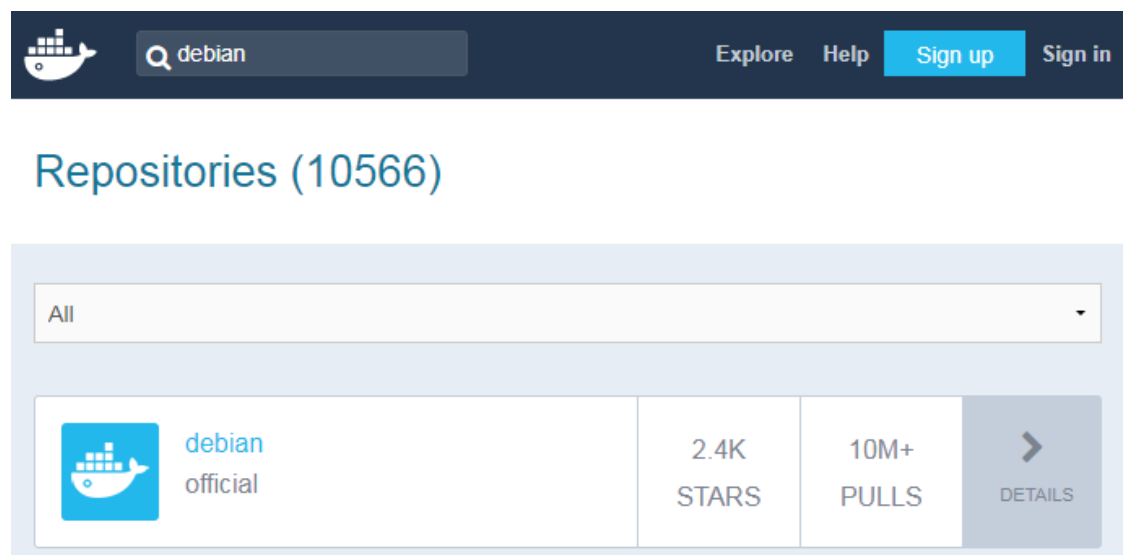
Valitsemamme palvelut tulisi nyt siirtää docker kontteihin. Palvelut piti siis tehdä käsin docker-konteiksi ja sen jälkeen luoda dockerfilet palveluista.

4.1 Konttien luominen

Docker konttien luominen aloitetaan valitsemalla mihin pohjaan rakennamme sovelluksen. Dockerissa palvelu rakennetaan kerroksittain, joten käyttöjärjestelmä olisi ensimmäinen kerros sovelluksessamme. Aiemmassa labrassa rakensimme ntopng palvelun Debian 9 jakelun päälle, joten aloitamme docker-kontin rakentamisen debian 9 pohjalle. Käyttöjärjestelmistä ja sovelluksista on saatavilla valmiita docker imageja dockerhubissa (Kuvio 13 Dockerhub)

<https://hub.docker.com/>

Dockerhubista löytyy virallinen debian image josta on saatavilla useita versioita.

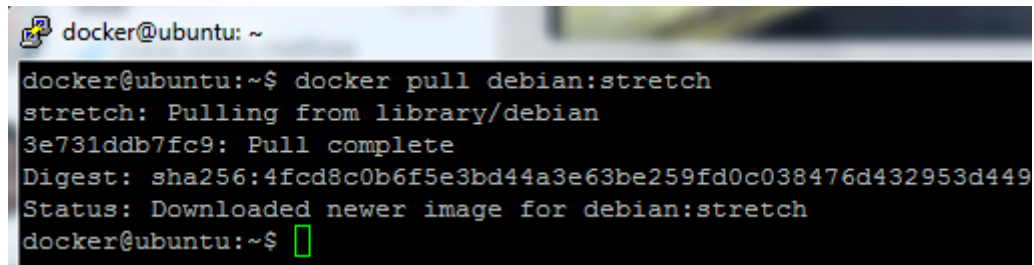


Kuvio 13 Dockerhub

Ntopng oli rakennettu Debian 9 eli "stretch" version päälle joten lataamme dockerhubista debian:stretch imagen. Docker hakee automaattisesti imagea dockerhubista jos sitä ei löydy paikallisesti käytettävästä koneesta.

4.2 Ntopng kontitus

Ntopng palvelua varten haemme dockerhubista debian:stretch docker imagen komennolla *"docker pull debian:stretch"*(Kuvio 14 Docker pull)



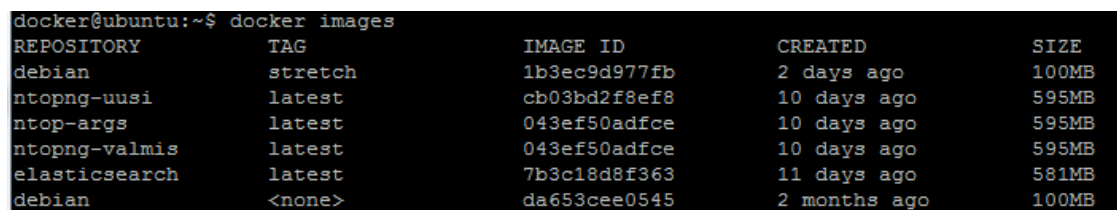
```

docker@ubuntu: ~
docker@ubuntu:~$ docker pull debian:stretch
stretch: Pulling from library/debian
3e731ddb7fc9: Pull complete
Digest: sha256:4fcd8c0b6f5e3bd44a3e63be259fd0c038476d432953d449
Status: Downloaded newer image for debian:stretch
docker@ubuntu:~$

```

Kuvio 14 Docker pull

Docker imaget voi listata komennolla *"docker images"*. Komennolla näemme images repositoryn, tagin eli sille annetun nimen, docker imagen id numeron, images luomispäivän, sekä koon megabitteinä(Kuvio 15 Docker images)



REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
debian	stretch	1b3ec9d977fb	2 days ago	100MB
ntopng-uusi	latest	cb03bd2f8ef8	10 days ago	595MB
ntopng-args	latest	043ef50adfce	10 days ago	595MB
ntopng-valmis	latest	043ef50adfce	10 days ago	595MB
elasticsearch	latest	7b3c18d8f363	11 days ago	581MB
debian	<none>	da653cee0545	2 months ago	100MB

Kuvio 15 Docker images

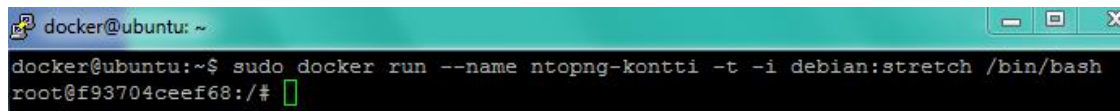
Seuraavaksi meidän pitää käynnistää docker kontti ja mennä sen sisään, jotta voidaan asentaa tarvittavat paketit kontin sisälle. Kontin ajaminen tapahtuu komennolla *"docker run imagen_nimi"*(Kuvio 15 Docker run). Lisäparametreilla voidaan antaa kontille lisää ominaisuuksia.

Docker run komennon rakenne on seuraavanlainen

docker run [OPTIONS] IMAGE [COMMAND] [ARG...], jossa OPTIONS parametreina ovat:

--name asettaa kontille nimen
--tty, -t asettaa virtuaali terminaali linjan
--interacative, -i pitää STDIN:in avoinna

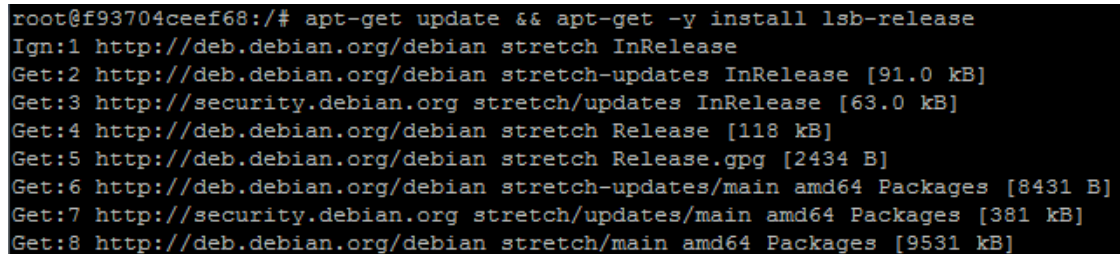
Komennon lopussa */bin/bash* tarkoittaa, että mikä ohjelma käynnistetään ensimmäisenä kontissa. Bash on Linuxissa oletuksena oleva komenotulkki jonka avulla voidaan asentaa kaikki tarvittavat paketit palvelun suorittamiseksi.



```
docker@ubuntu: ~$ sudo docker run --name ntopng-kontti -t -i debian:stretch /bin/bash
root@f93704ceef68:/#
```

Kuvio 16 Docker run

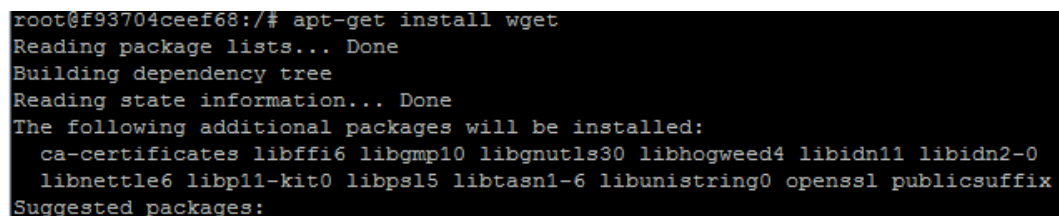
Ensimmäiseksi koneelle haetaan hakemistolistojen päivitykset ja asennetaan *lsb-release* paketti, joka tarjoaa infrastruktuurin LSB sovellusten asentamiseksi Debianiin(Kuvio 17 lsb-release)



```
root@f93704ceef68:/# apt-get update && apt-get -y install lsb-release
Ign:1 http://deb.debian.org/debian stretch InRelease
Get:2 http://deb.debian.org/debian stretch-updates InRelease [91.0 kB]
Get:3 http://security.debian.org stretch/updates InRelease [63.0 kB]
Get:4 http://deb.debian.org/debian stretch Release [118 kB]
Get:5 http://deb.debian.org/debian stretch Release.gpg [2434 B]
Get:6 http://deb.debian.org/debian stretch-updates/main amd64 Packages [8431 B]
Get:7 http://security.debian.org stretch/updates/main amd64 Packages [381 kB]
Get:8 http://deb.debian.org/debian stretch/main amd64 Packages [9531 kB]
```

Kuvio 17 lsb-release

Seuraavaksi täytyy asentaa wget paketti komennolla *apt-get -y install wget* (Kuvio 18 apt-get install wget), jolla voidaan ladata ntopng repositoryn asentamiseen tarkoitettu deb-paketti (Kuvio 19 Deb paketti)



```
root@f93704ceef68:/# apt-get install wget
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates libffi6 libgmp10 libgnutls30 libhogweed4 libidn11 libidn2-0
  libnettle6 libp11-kit0 libpsl5 libtasn1-6 libunistring0 openssl publicsuffix
Suggested packages:
```

Kuvio 18 apt-get install wget

```

root@f93704ceef68:/# wget http://apt-stable.ntop.org/stretch/all/apt-ntop-stable.deb
--2018-02-17 20:53:28-- http://apt-stable.ntop.org/stretch/all/apt-ntop-stable.deb
Resolving apt-stable.ntop.org (apt-stable.ntop.org)... 151.11.50.180
Connecting to apt-stable.ntop.org (apt-stable.ntop.org)[151.11.50.180]:80... connecte
d.
HTTP request sent, awaiting response... 302 Found
Location: http://packages.ntop.org/apt-stable/stretch/all/apt-ntop-stable.deb [follow
ing]
--2018-02-17 20:53:28-- http://packages.ntop.org/apt-stable/stretch/all/apt-ntop-sta
ble.deb
Resolving packages.ntop.org (packages.ntop.org)... 151.11.50.180
Reusing existing connection to apt-stable.ntop.org:80.
HTTP request sent, awaiting response... 200 OK
Length: 6586 (6.4K) [application/x-debian-package]
Saving to: 'apt-ntop-stable.deb'

apt-ntop-stable.deb  100%[=====>]  6.43K  --.-KB/s    in 0.001s

2018-02-17 20:53:28 (5.98 MB/s) - 'apt-ntop-stable.deb' saved [6586/6586]

root@f93704ceef68:/# █

```

Kuvio 19 Deb paketti

deb paketin asentaminen vaatii gnupg paketin toimiakseen. Gnupg voidaan asentaa komennolla *"apt-get -y install gnupg"*. Nyt voimme asentaa ntopng:n deb paketin komennolla *"dpkg -i apt-ntop-stable.deb"* (Kuvio 20 Deb paketin asennus)

```

root@f93704ceef68:/# dpkg -i apt-ntop-stable.deb
(Reading database ... 8333 files and directories currently installed.)
Preparing to unpack apt-ntop-stable.deb ...
Unpacking apt-ntop-stable (2.6-916) over (2.6-916) ...
upgrade
Setting up apt-ntop-stable (2.6-916) ...
Adding ntop key to apt keyring
Warning: apt-key should not be used in scripts (called from postinst maintainerscript
of the package apt-ntop-stable)
OK
root@f93704ceef68:/# █

```

Kuvio 20 Deb paketin asennus

Paketin asentamisen jälkeen hakemistolistat on taas päivitettävä (Kuvio 21 apt-get update)

```

root@f93704ceef68:/# apt-get update
Ign:1 http://deb.debian.org/debian stretch InRelease
Hit:2 http://deb.debian.org/debian stretch-updates InRelease
Hit:3 http://deb.debian.org/debian stretch Release
Hit:4 http://security.debian.org stretch/updates InRelease
Get:6 http://packages.ntop.org/apt-stable/stretch x64/ InRelease [2137 B]
Get:7 http://packages.ntop.org/apt-stable/stretch all/ InRelease [2137 B]
Get:8 http://packages.ntop.org/apt-stable/stretch x64/ Packages [9782 B]
Get:9 http://packages.ntop.org/apt-stable/stretch all/ Packages [2351 B]
Fetched 16.4 kB in 0s (24.1 kB/s)
Reading package lists... Done
root@f93704ceef68:/# █

```

Kuvio 21 apt-get update

Varsinaiset ntopng paketit voidaan asentaa nyt pakettien hallinnasta komennolla *"apt-get install pfring ntopng ntopng-data"* (Kuvio 22 ntopng paketit)

```
root@f93704ceef68:/# apt-get -y install pfring ntopng ntopng-data
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils bridge-utils cpp cpp-6 cron dkms ethtool exim4 exim4-base exim4-c
  exim4-daemon-light fakeroot fontconfig fontconfig-config fonts-dejavu-core
  gcc-6 geoip-database guile-2.0-libs kmod krb5-locales libasan3 libatomic1
```

Kuvio 22 ntopng paketit

Kaikki tarvittavat paketit palvelua varten on nyt asennettu. Konfiguraatioita voidaan muokata kontin sisällä mutta labran tarkoituksena on saada konfiguraatiomuutokset toimimaan ympäristömuuttujia käyttäen.

Ntopng vaatii *"redis-server"* toimiakseen, joten redis täytyy käynnistää ennen ntopia. Redis käynnistetään komennolla *"/etc/init.d/redis-server start"*, jonka jälkeen ntop voidaan käynnistää suoraan bashissa komennolla *"ntopng"*. Ntopin tarvittavat parametrit voidaan antaa suoraan komentoriviltä käyttäen yhtä ympäristömuuttujaa.

4.3 Dockerfile

Dockerfile on tiedosto, jonka mukaan luodaan docker-image. Dockerfileen määritellään ohjeet, miten docker-image tulee rakentaa. Dockerfilen tulisi sisältää kaikki komennot, millä palvelu saadaan asennettua manuaalisesti. *"docker build"* komento rakentaa dockerfilen pohjalta docker-imagen.

Ensiksi dockerfileen määritellään pohjaimage, josta se alkaa kasaamaan uutta imagea. Tiedostoon määritellään kaikki samat komennot kuin manuaalisessa asennuksessaakin. Lopuksi imagelle kerrotaan portti mitä se kuuntelee sekä entrypoint, eli niin sanottu sisääntulopiste kontille. Dockerfilessä on yksi ympäristömuuttuja *"\$@"* joka ottaa vastaan joukon muuttujia ja ohjaa ne ntopng:lle (Kuvio 23 Dockerfile ntop)

```

GNU nano 2.5.3                               File: Dockerfile
FROM debian:stretch
RUN apt-get update && apt-get -y -q install lsb-release
RUN apt-get -y install wget
RUN wget http://apt-stable.ntop.org/stretch/all/apt-ntop-stable.deb
RUN apt-get -y install gnupg
RUN apt --fix-broken install
RUN dpkg -i apt-ntop-stable.deb
RUN apt --fix-broken install
RUN dpkg -i apt-ntop-stable.deb
RUN apt-get update
RUN apt-get -y install pfring ntopng ntopng-data
EXPOSE 3000
RUN echo '#!/bin/bash\n/etc/init.d/redis-server start\ntopng "$@"' > /tmp/run.sh
RUN chmod +x /tmp/run.sh
ENTRYPOINT ["/tmp/run.sh"]

```

Kuvio 23 Dockerfile ntop

kun dockerfile on valmis niin voimme luoda docker-imagen siitä komennolla *"docker build -t imagen_nimi."*(Kuvio 23 Docker build)

```

root@ubuntu:~/ntopng# docker build -t ntopng-image-2018 .
Sending build context to Docker daemon  2.56kB
Step 1/15 : FROM debian:stretch
--> 1b3ec9d977fb
Step 2/15 : RUN apt-get update && apt-get -y -q install lsb-release
--> Running in f4c7aa280212
Ign:1 http://deb.debian.org/debian stretch InRelease
Get:2 http://security.debian.org stretch/updates InRelease [63.0 kB]
Get:3 http://deb.debian.org/debian stretch-updates InRelease [91.0 kB]
Get:4 http://deb.debian.org/debian stretch Release [118 kB]
Get:5 http://deb.debian.org/debian stretch Release.gpg [2434 B]
Get:6 http://security.debian.org stretch/updates/main amd64 Packages [381 kB]
Get:7 http://deb.debian.org/debian stretch-updates/main amd64 Packages [8431 B]
Get:8 http://deb.debian.org/debian stretch/main amd64 Packages [9531 kB]

```

Kuvio 24 Docker build

ntopng-image-2018 pitäisi ilmestyä docker images listaan (Kuvio 25 Docker images).

```

root@ubuntu:~# docker images

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ntopng-image-2018	latest	ddd1365ac1f1	12 seconds ago	596MB

Kuvio 25 Docker images

Nyt voimme käynnistää kontin komennolla *"docker run -d --name ntopng-valmis-2018 --net=host -p 3000:3000 ntopng-image-2018 -F="es;flows;ntopng-%Y.%m.%d;http://192.168.30.132:9200/_bulk;"* (Kuvio 26 Docker run ntopng)


```

root@ubuntu:~# docker run -d --name ntopng-valmis-2018 --net=host -p 3000:3000 ntopng
--image-2018 -F="es;flows;ntopng-%Y.%m.%d;http://192.168.30.132:9200/_bulk;"
9aa62dc33e470c3ce904f71aaa0aded987815fd311134394af4e7140e3c5effa
root@ubuntu:~# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS            PORTS              NAMES
9aa62dc33e47       ntopng-image-2018  "/tmp/run.sh -F=es;f..." 6 seconds ago
Up 5 seconds
ntopng-valmis-2018
root@ubuntu:~# █

```

Kuvio 26 Docker run ntopng

4.4 Elasticsearch kontitus

Elasticsearchin kontti rakennetaan openjdk:8-jre konttipohjan päälle (Kuvio 27 OpenJDK kontin lataus). Kyseinen kontti sisältää Debian 9 asennuksen, jossa on OpenJDK8 esiasennettuna.

```

docker@ubuntu:~$ docker pull openjdk:8-jre
8-jre: Pulling from library/openjdk
3e731ddb7fc9: Already exists
47cafa6a79d0: Downloading [=====>] 9.174MB/11.11MB
79fcf5a213c7: Download complete
fd532571c5d3: Download complete
root@ubuntu:~#

```

Kuvio 27 OpenJDK kontin lataus

Ladataan kontti esimerkin mukaisesti (Kuvio 27). Rakennetaan tästä pohjasta kokonaan uusi Docker kontti, joka sisältää Elasticsearch tiedonhakujärjestelmän. Elasticsearch vaatii hieman säätämistä, jotta se saadaan toimimaan kontissa. Rakennetaan kontti myös pakkaukseksi, jonka voi ottaa käyttöön muissa ympäristöissä ilman Dockerfilellä tehtyä automaattiasennusta.

```

root@ubuntu:~# docker images
REPOSITORY          TAG             IMAGE ID          CREATED          SIZE
openjdk              8-jre          e956268fd4ed     5 days ago      538MB

```

Kuvio 28 Image ID:n tarkistus

Tarkistetaan ladatun "*openjdk:8-jre*" imagen ID-numero (Kuvio 28 Image ID:n tarkistus). Tätä tunnusta tarvitsemme, kun muokkaamme konttia pysyvästi. Käynnistetään kontti ja hypätään kontin sisällä toimivaan komentokehotteeseen (Kuvio 29 Kontin käynnistys).

```

root@ubuntu:~# docker run -i -t e956268fd4ed /bin/bash
root@5f76c4d5807b:/# █

```

Kuvio 29 Kontin käynnistys

Kuten kuvasta huomaa, kontin käynnistys loi pohjakontista uuden kontin.

Pohjakontin ID oli `"e956268fd4ed"` kun nyt käynnistynyt kontti on ID

`"5f76c4d5807b"`. Kontit pohjautuvat samaan pohjajärjestelmään, mutta ovat erilliset olemassa olevat ympäristöt. Aloitetaan ympäristön asennus konttiin.

```
root@5f76c4d5807b:/# apt-get update && apt-get -y install apt-transport-https wget
Ign:1 http://deb.debian.org/debian stretch InRelease
Get:2 http://security.debian.org stretch/updates InRelease [63.0 kB]
Get:3 http://deb.debian.org/debian stretch-updates InRelease [91.0 kB]
Get:4 http://deb.debian.org/debian stretch Release [118 kB]
```

Kuvio 30 Perusohjelmien asennus

Aloitetaan asentamalla paketit *wget* ja *apt-transport-https* (Kuvio 30). Wget ohjelmaa tarvitsemme Elasticsearchin komponentin lataamiseen, kun taas *apt-transport-https* lisää APT-pakettienhallintatyökaluun HTTPS-tuen, jota myös Elasticsearch tarvitsee. Jatketaan ottamalla Elasticsearchin pakettilähteet käyttöön, sekä asentamalla Elasticsearch (Kuvio 31).

```
root@5f76c4d5807b:/# echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" >> /etc/apt/sources.list.d/elastic-6.x.list
root@5f76c4d5807b:/# wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | apt-key add -
OK
root@5f76c4d5807b:/# apt-get update && apt-get install elasticsearch
Ign:1 http://deb.debian.org/debian stretch InRelease
Hit:2 http://security.debian.org stretch/updates InRelease
Hit:3 http://deb.debian.org/debian stretch-updates InRelease
Hit:4 http://deb.debian.org/debian stretch Release
Ign:6 https://artifacts.elastic.co/packages/6.x/apt stable InRelease
```

Kuvio 31 Elasticsearch asennus

Kun Elasticsearch on asennettu, luodaan uusi käyttäjä Elasticsearchille (Kuvio 32), joka liitetään käyttäjäryhmään *root* (Kuvio 31). Lopuksi muokataan Elasticsearchin asetustiedostoja. Tämä tosin vaatii vielä tekstieditorin asentamisen, joten asennetaan nano.

```
root@5f76c4d5807b:/etc/elasticsearch# apt-get install nano
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  spell
The following NEW packages will be installed:
  nano
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded
```

Kuvio 32 Tekstieditorin asennus

Kun nano on asennettu (Kuvio 32), voidaan muokata Elasticsearchin asetustiedostoa. Asetustiedosto löytyy polusta `"/etc/elasticsearch/elasticsearch.yml"`. Elasticsearch

vaatii toimiakseen ympäristökohtaisia asetuksia, kuten Elasticsearch clusterin nimen, Elasticsearch koneen nimen, sekä IP-osoitteen jossa Elasticsearch toimii.

Elasticsearch ymmärtää ympäristömuuttujia (Environmental Variables)

asetustiedostossaan, joten se helpottaa kontin muodostamista runsaasti.

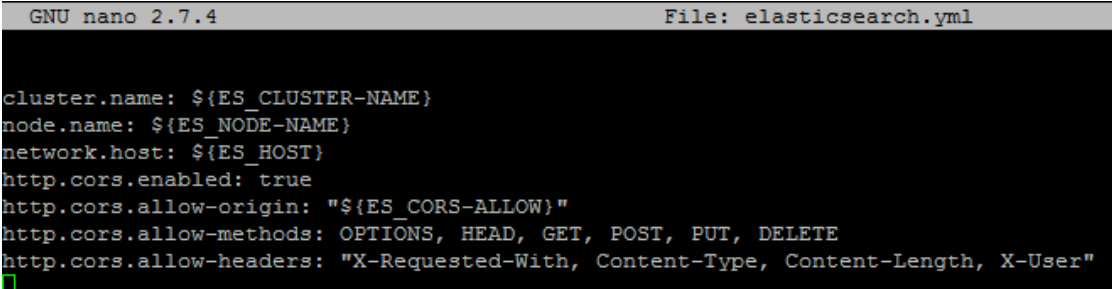
Ympäristömuuttujat ymmärretään konfiguraatiossa muodossa "asetus.on:

`${MUUTTUJANNIMI}`", jolloin ympäristömuuttuja MUUTTUJANNIMI kun asetetaan

arvoon esimerkiksi "on", vastaisi tämä samaa kuin asetustiedostoon olisi kirjoitettu

"asetus.on: on". Docker-kontteja käytettäessä, asetukset annetaan

ympäristömuuttujina.



```

GNU nano 2.7.4                                     File: elasticsearch.yml

cluster.name: ${ES_CLUSTER-NAME}
node.name: ${ES_NODE-NAME}
network.host: ${ES_HOST}
http.cors.enabled: true
http.cors.allow-origin: "${ES_CORS-ALLOW}"
http.cors.allow-methods: OPTIONS, HEAD, GET, POST, PUT, DELETE
http.cors.allow-headers: "X-Requested-With, Content-Type, Content-Length, X-User"

```

Kuvio 323 Asetukset Elasticsearchille

Lisätään *elasticsearch.yml* tiedoston loppuun kuvanmukaiset asetukset (Kuvio 33).

Nyt käytössä on ympäristömuuttujat *ES_CLUSTER-NAME*, *ES_NODE-NAME*, *ES_HOST* ja *ES_CORS-ALLOWED*. Kyseiset asetukset voidaan antaa Elasticsearchille konttia käynnistettäessä Dockerista. Tallennetaan asetustiedosto ja poistutaan shellistä komennolla *exit*.

Nyt alla olevaan konttiin on säädetty perusasiat kuntoon. Tässä vaiheessa

huomioidaan kontin ID-numero. Konttimme oli "5f76c4d5807b", joka on Dockerin

ympäristössä edelleen olemassa kaikkine muutoksineen. Kopioidaan kontti uudeksi

kokonaan uudeksi pohjaksi, nimeltään "es-engine".

```

root@ubuntu:~# docker commit 5f76c4d5807b es-engine
sha256:fd982dacf35d1e22494cf11da841389c6572ffc2cd349a3d9221a40abe510e5e
root@ubuntu:~# docker image
image images
root@ubuntu:~# docker images

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
es-engine	latest	fd982dacf35d	8 seconds ago	591MB
openjdk	8-jre	e956268fd4ed	5 days ago	538MB
ubuntu	16.04	0458a4468cbc	3 weeks ago	112MB
debian	stretch	da653cee0545	2 months ago	100MB
hello-world	latest	f2a91732366c	3 months ago	1.85kB
vostro/ntopng	latest	ed24ee80cc9d	5 months ago	468MB

```

root@ubuntu:~#

```

Kuvio 334 Ympäristön muuttaminen uudeksi kontiksi

Tarkistetaan "*docker images*" komennolla konttilista (Kuvio 34). Kontteihin on ilmestynyt edellämainittu "*es-engine*" joka on kokonaan uusi kontti, pohjautuen aiemmin tehtyihin muutoksiin kontissa "*5f76c4d5807b*". Seuraavaksi kun otamme konttia käyttöön, täytyy Elasticsearchille luoda virtuaalilevy, jonne data tallennetaan. Konttien ollessa vain-luku (muutokset säilyy vain per ympäristö), tarvitsee Elasticsearch erillisen pysyvän tallennustilan, joka kulkee kontin mukana erillisessä tiedostossa. Tätä kutsutaan volumeksi. Luodaan siis volume, jota voimme käyttää uudessa kontissa.

```

root@ubuntu:~# docker create -v /var/lib/elasticsearch --name esdata es-engine
30f4355b29d1f13e6015dd400ab9fdb3efbfd4d3a4020ec0e62775e157d87b8d
root@ubuntu:~#

```

Kuvio 35 Volumen luonti

Luodaan uusi volume kontille "*es-engine*" (Kuvio 35). Volumen nimeksi tulee "*esdata*" ja volume liitetään kontissa polkuun "*/var/lib/elasticsearch*", jonne Elasticsearch tallentaa tuodun datan. Seuraavaksi käynnistetään Elasticsearch kontti.

```

root@ubuntu:~# docker run -t -i -e ES_CLUSTER-NAME='testcluster' -e ES_NODE-NAME='docker' -e ES_HOST='_global_' -e ES_CORS-ALLOW='*' --expose 9200-9300 --user elasticsearch --volumes-from esdata es-engine /usr/share/elasticsearch/bin/elasticsearch
[2018-02-20T21:42:28,702][INFO ][o.e.n.Node               ] [docker] initializing ...
[2018-02-20T21:42:29,938][INFO ][o.e.e.NodeEnvironment ] [docker] using [1] data paths, mounts [[/var/lib/elasticsearch (/dev/mapper/ubuntu--vg-root)]], net usable_space [9.4gb], net total_space [14.1gb], types [ext4]
[2018-02-20T21:42:29,940][INFO ][o.e.e.NodeEnvironment ] [docker] heap size [1007.3mb], compressed ordinary object pointers [true]
[2018-02-20T21:42:29,945][INFO ][o.e.n.Node               ] [docker] node name [docker], node ID [QW-0vsTVR9q-lMnL0kbXSQ]
[2018-02-20T21:42:29,946][INFO ][o.e.n.Node               ] [docker] version[6.2.2], pid[1], build[10b1edd/2018-02-16T19:01:30.685723Z], OS[Linux/4.4.0-112-generic/amd64], JVM[Oracle Corporation/OpenJDK 64-Bit Server VM/1.8.0_151/25.151-b12]
[2018-02-20T21:42:29,948][INFO ][o.e.n.Node               ] [docker] JVM arguments [-Xms1g, -Xmx1g, -XX:+UseConcMarkSweepGC, -XX:CMSInitiatingOccupancyFraction=75, -XX:+UseCMSInitiatingOccupancyOnly, -XX:+AlwaysPreTouch, -Xss1m, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Djna.nosys=true, -XX:-OmitStackTraceInFastThrow, -Dio.netty.noUnsafe=true, -Dio.netty.noKeySetOptimization=true, -Dio.netty.recycler.maxCapacityPerThread=0, -Dlog4j.shutdownHookEnabled=false, -Dlog4j2.disable.jmx=true, -Djava.io.tmpdir=/tmp/elasticsearch.nG3NPrzQ, -XX:+HeapDumpOnOutOfMemoryError, -XX:HeapDumpPath=/var/lib/elasticsearch, -XX:+PrintGCDetails, -XX:+PrintGCDateStamps, -XX:+PrintTenuringDistribution, -XX:+PrintGCApplicationStoppedTime, -Xloggc:/var/log/elasticsearch/gc.log, -XX:+UseGCLogFileRotation, -XX:NumberOfGCLogFiles=32, -XX:GCLogFileSize=64m, -Des.path.home=/usr/share/elasticsearch, -Des.path.conf=/etc/elasticsearch]
[2018-02-20T21:42:39,803][INFO ][o.e.p.PluginsService     ] [docker] loaded module [aggs-matrix-stats]
[2018-02-20T21:42:39,804][INFO ][o.e.p.PluginsService     ] [docker] loaded module [analysis-common]

```

Kuvio 34 Elastic käynnistyy

Elasticsearch käynnistyy onnistuneesti, kuten Kuvio 34stä näkyy.

5 Docker Compose

Docker compose on työkalu, jolla voidaan määritellä usean eri kontin ympäristö ja sen asetukset. Docker compose käyttää YAML tiedostoa ympäristön luomiseen.

Docker composen avulla voit pystyttää kokonaisen konttiympäristön yhdellä käskyllä.

Docker compose on erillinen paketti dockeriin joten sen voi yksinkertaisimmillaan asentaa linux jakeluun komennolla *"apt-get install docker-compose"*

5.1 Compose tiedoston luonti

Docker compose tiedostoon määritellään kaikki kontit ja niiden parametrit.

Yksinkertaisimmillaan compose tiedosto sisältää konttien nimet, pohjaimaget, ja portit. Compose tiedostoomme määritellään palveluiksi ntopng ja elasticsearch.

Molemmille palveluille määritetään pohjaimage sekä portit ja ympäristömuuttujat.

```
1  version: "2"
2  services:
3    ntopng:
4      image: ntopng
5      ports:
6        "3000:3000"
7      environment:
8    elasticsearch:
9      image: elasticsearch
10     ports:
11       "9200:9200"
12       "9300:9300"
13     environment:
14       ES_CLUSTER-NAME=klusteri
15       ES_NODE_NAME=node-1
16       ES_HOST=hosti-1
17       ES_CORS-ALLOW="*"
18
```

6 Muutos-skenaario

Ntopng:n kaikki parametrit pystytään määrittelemään kontin käynnistyksen

yhteydessä. Aiemman labran muutos-skenaario oli portin ja IP-osoitteen

vaihtuminen. Ntopin kaikki muutokset voidaan tehdä kontin käynnistyskomentoon.

```
"docker run -d --name ntopng-valmis-2018 --net=host -p 3000:3000 ntopng-image-2018 -F="es;flows;ntopng-%Y.%m.%d;http://192.168.30.132:9200/_bulk;"
```

Elasticsearchin muutokset voidaan myös kertoa suoraan ympäristömuuttujille.

```
"docker run -t -i -e ES_CLUSTER-NAME='testcluster' -e ES_NODE-NAME='docker' -e ES_HOST='global' -e ES_CORS-ALLOW='*' --expose 9200-9300 -p 9200:9200 --user elasticsearch --volumes-from esdata es-engine /usr/share/elasticsearch/bin/elasticsearch
```

Porttien vaihtamiset onnistuivat ympäristömuuttujia muokkaamalla.

7 Pohdinta

Labra jäi hieman keskeneräiseksi vielä ja palveluita olisi voinut hienosäätää lisää.

Opimme kuitenkin dockerin käyttöä sekä palveluiden kontittamista.

8 Lähteet

<https://www.jamescoyle.net/how-to/1503-create-your-first-docker-container>

<https://www.digitalocean.com/community/tutorials/how-to-work-with-docker-data-volumes-on-ubuntu-14-04>

<https://stackoverflow.com/questions/16047306/how-is-docker-different-from-a-normal-virtual-machine>

<https://nickjanetakis.com/blog/comparing-virtual-machines-vs-docker-containers>

<https://www.networkcomputing.com/cloud-infrastructure/comparing-docker-containers-virtual-machines/77530426>

Loput omasta päästä, ei tämä rakettitiedettä ole.