jamk.fi

Pilvipalvelut

Contriboard asennus

Ryhmä 3
Alexander Andreev
Rami Ojala
Ilari Rajala
Asko Ropponen

Laboratorioraportti
TTTW0430 - Pilvipalvelut, Jarmo Viinikanoja
1.10.2019
Tieto- ja viestintätekniikan tutkinto-ohjelma

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

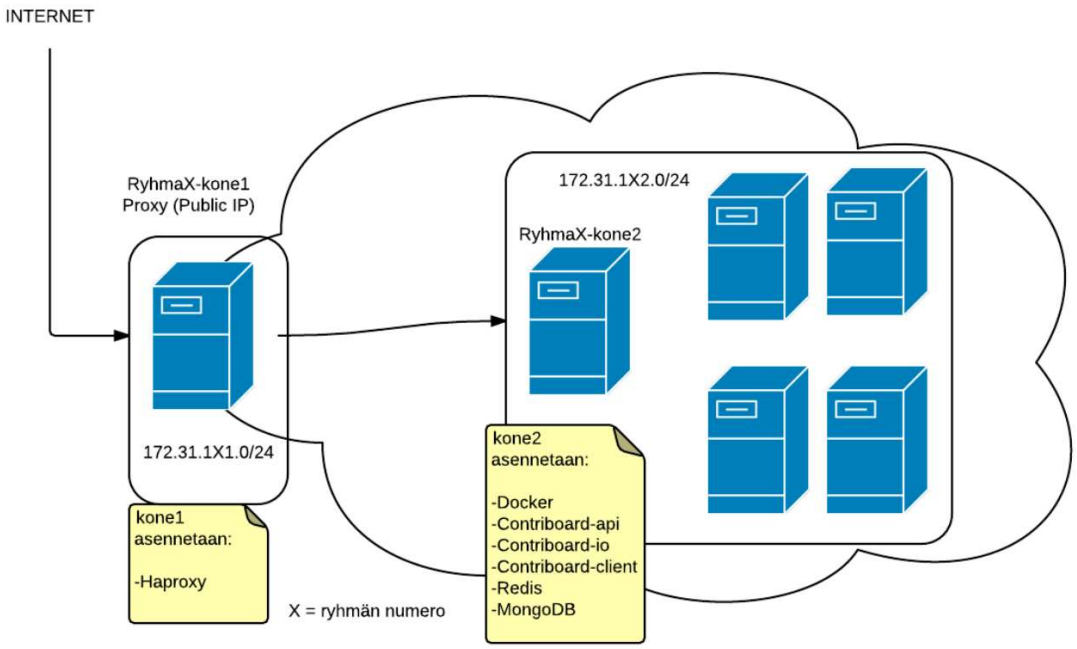
Sisältö

1 Johdanto

Pilvipalvelut laboratorio yksi työnä tehtiin Contriboard palvelun pystytys Amazon Web Services pilvipalveluun, ja siinä käytettiin aiemmin kurssilla opetettuja menetelmiä ja laboratoriotyön ohjeistusta. Opiskelijat jaettiin neljän hengen ryhmiin, jotka ryhmänä suorittivat laboratoriotyön sekä tekivät tämän kyseisen raportin aiheesta. Raportissa kerrotaan ja kuvaillaan Contriboardin asennuksen vaiheita.

2 Tehtävänanto

- Asenna HAProxy kone1:seen ja säädetään se kuuntelemaan porttia 80 ja ohjaamaan liikennettä eri komponenttien välillä.
- Ota yhteys koneen yksi kautta sisäverkon koneeseen.
- Asenna kone3:seen Docker ja Contriboardin kontit.
- Tee enviroment variableja käyttäviä tiedostoja, joissa määritetään konteille tarvittavat ympäristön asetukset.
- Kokeile että ryhmän julkisesta IP-osoitteesta pääsee toimivaan Contriboard palveluun.



Kuva 1 Contriboard palvelun rakennekaavio.

3 Teoreettiset lähtökohdat

Contriboard palvelu pystytetään Amazon Web Services palveluun, johon opettaja Viinikanoja oli jo valmiiksi luonut tunnukset ja avaimet. Työssä käytetään Amazon EC2 (Elastic Compute Cloud), joka tarjoaa muutettavissa olevaa laskenta kapasiteettiä pilvipalvelussa. EC2 koneissa käytetään korkea taajuisia Intelin Xeon prosessoreita ja niissä on tasapainotettu laskenta muisti ja verkkoresurssit. työssä käytettävissä t2.micro koneissa on 3,3GHz skaalautuva prosessori teho, muistia 1 GiB ja tallennus Amazon EBS (Elastic Block Store) (Amazon EC2 Instance IP Addressing).

Tietoturvallisuuden vuoksi vain kahdelle koneelle (ulkoverkko) on annettu mahdollisuus päästä suoraan internettiin julkisilla IP-osoitteilla ja niille asennetaan HAProxy, joiden kautta muut private IP koneet (sisäverkko) keskustelevat toistensa kanssa ja ovat yhteydessä internettiin.

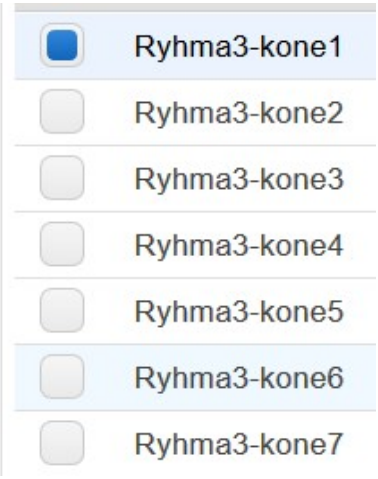
SSH yhteydellä ulkoverkon koneeseen saa ryhmälle valmiiksi määritetyllä PEM avaimella käyttäen PuTTY terminaali emulaattoria. Ulkokoneen kautta saadaan taas SSH yhteys muodostettua sisäverkon koneisiin käyttäen niiden private IP-osoitteita.

Contriboard asennetaan käyttäen Docker kontteja. Docker on joukko PaaS (Platform as a Service) tuotteita, ja ne käyttävät käyttöjärjestelmä tason virtualisointia ohjelmiston ja siihen tarvittavien kirjastojen ja riippuvuuksien toimittamiseen, joita sanotaan konteiksi. Niillä pystytään helposti jakamaan toiselle koneelle identtinen ohjelmisto ympäristöineen (What is a Conteiner).

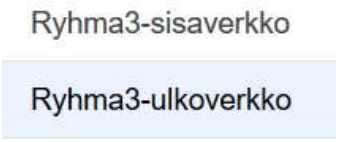
4 Työn kulku

4.1 AWS

Ohjeiden mukaan luottiin seitsemän virtuaalikonetta, joilla oli kaksi security groupia.



Kuva 2. Lista ryhmän koneista Amazon Web Services palvelussa.



Kuva 3. Lista Ryhmän verkoista Amazon Web Services palvelussa.

Laboratorio työssä tarvittiin vain koneita Ryhmä3-kone1 (julkinen IP) ja Ryhmä3-kone3 (private IP).

Sisäverkon koneelle annettiin seuraavat Security Group säännöt (Kuva 4).

Type	Protocol	Port Range	Source	Description
All traffic	All	All	172.31.131.0/24	
SSH	TCP	22	0.0.0.0/0	

Kuva 4. Sisäverkon Security Group säännöt.

Ulkoverkon koneelle seuraavat säännöt (Kuva 5).

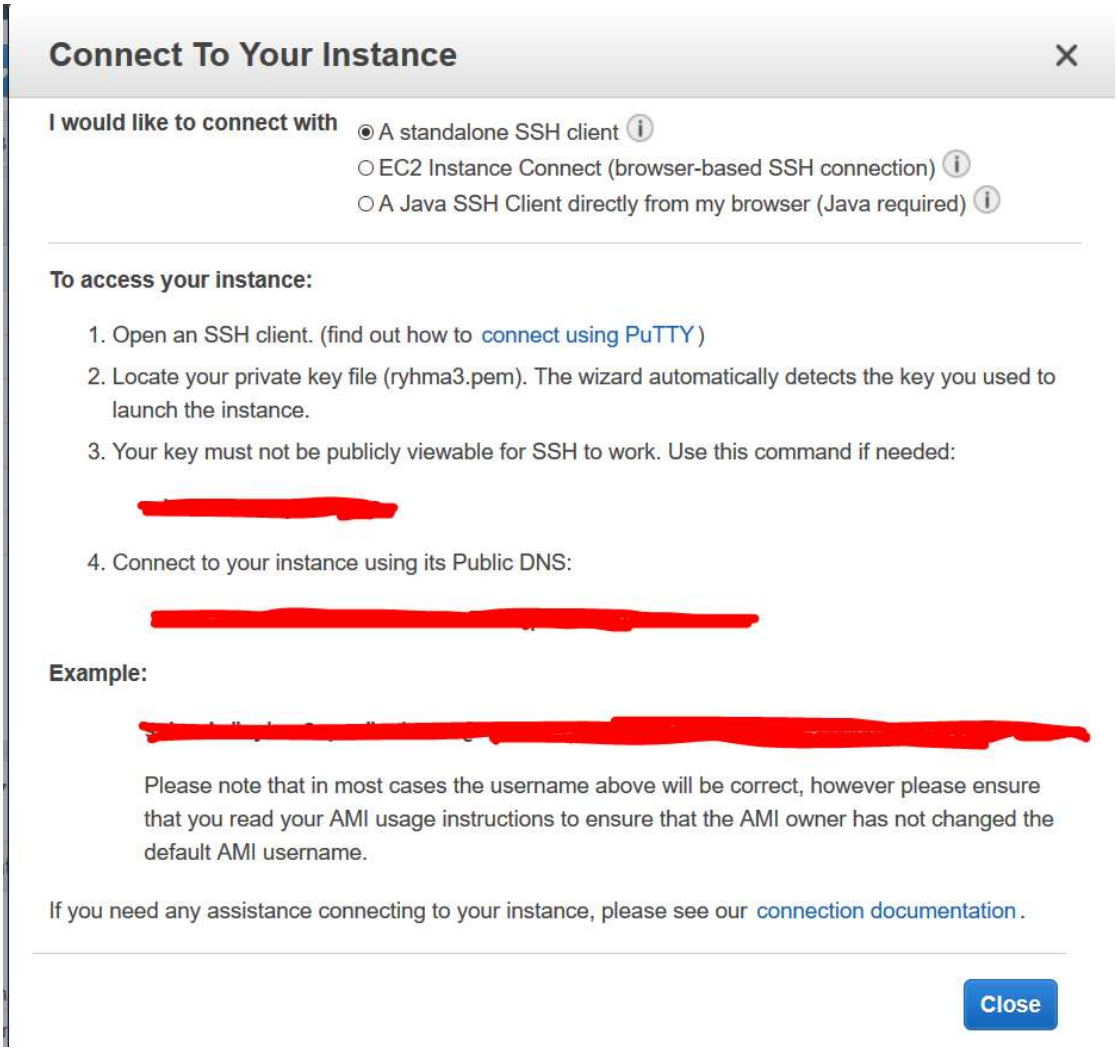
HTTP	TCP	80	0.0.0.0/0
All traffic	All	All	172.31.131.0/24
SSH	TCP	22	0.0.0.0/0

Kuva 5. Ulkoverkon Security Group säännöt.

Jokaiselle koneelle asennettiin Ubuntu 16.04 Server.

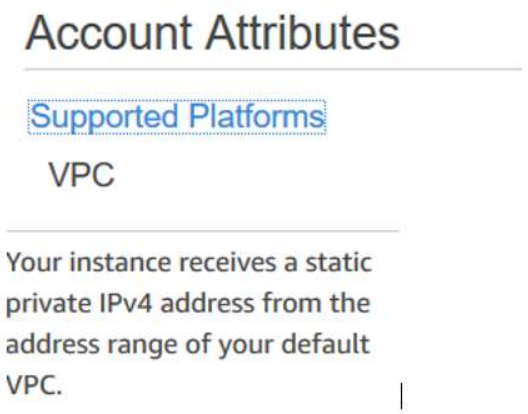
Yhteys koneeseen onnistui SSH-private avaimen avulla (Kuva 6) ja tarvittavat osoitteet löytyivät AWS – Connect valikosta.

Kuva 6 SSH-private avain.



Kuva 7. Uusi yhteys instanssiin.

Koska alustan tyyppi oli VPC, oli käytössämme staattiset sisäiset IP-osoitteet, joten hostname muokkaus ei ollut välttämätöntä (Amazon EC2 Instance IP Addressing).



Kuva 8. Käyttäjän asetukset.

4.2 HAProxy

Kone1 asetettiin ulkoverkkoon ja siihen asennettiin HAProxy, jonka tehtävänä oli järjestelmän kuormantasaaminen. Järjestelmään tuleva liikenne ohjattiin HAProxylla Contriboardin eri instansseille. Tässä tapauksessa sillä ohjattiin myös järjestelmän sisäistä liikennettä IO-komponentille.

HAProxyn asennus Ubuntu ympäristössä oli hyvin yksinkertainen ja se tapahtui yhdellä komennolla:

```
sudo apt install haproxy
```

Tämä asentaa HAProxyn ja luo vaadittavat servicet ja muut sellaiset. HAProxy palvelua voi ohjata seuraavilla komennoilla:

1. sudo service haproxy stop
2. sudo service haproxy start
3. sudo service haproxy restart

HAProxyn konfiguraatio tiedostot sijaitsevat polussa *"/etc/haproxy/haproxy.cfg".* Tästä tiedostosta oli hyvä ottaa kopio komennolla:

```
sudo apt cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.bak
```

Laboratorion materiaaleissa annetun HAProxyn konfiguraation pystyi kopioimaan täysin, mutta muutama IP-osoite piti korjata, jotta Contriboard toimi oikein.

backend API:

server API1 172.31.132.111:9001 weight 1 maxconn 2048 check

backend WS:

server WebSocket1 172.31.132.111:9002 weight 1 maxconn 2048 check

backend CLIENT

server client 172.31.132.111:80 weight 1 maxconn 2048 check

Nämä kolme riviä muutettiin siten, että IP osoite vastasi sisäverkon (Kone3) IP-osoitetta. Tämän jälkeen HAProxy oli palvelun uudelleenkäynnistämisen jälkeen valmis.

4.3 Docker

Dockerin asennus sujui Dockerin tarjoamilla ohjeilla. Ohjeet oli tarkoitettu Ubuntun versioille 16.04 LTS, 18.04 LTS, 18.10 sekä 19.04. Koneessamme oli Ubuntu Server 16.04. Asennus sujui seuraavalla tavalla.

1. Päivitetään Ubuntun paketticache:

```
$ sudo apt-get update
```

2. Asennetaan paketit, jotta APT voi käyttää repositorioita http:n yli:

```
sudo apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg-agent \
  software-properties-common
```

3. Lisää Dockerin GPG-avain, jotta voidaan asentaa paketteja Dockerin repoista ja tarkistetaan että avain on ok:

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

```
$ sudo apt-key fingerprint 0EBFCD88

pub   rsa4096 2017-02-22 [SCEA]
      9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88
uid           [ unknown] Docker Release (CE deb) <docker@docker.com>
sub   rsa4096 2017-02-22 [S]
```

4. Lisätään Dockerin repositoriot Ubuntuun:

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

   Tämä siis x86_64 prosessoriarkkitehtuurille, jos eri, niin vaihtaa arch=amd64 johonkin toiseen, esim. arm64, jos 64-bittinen arm-prosessori.

   Meidän serverimme oli x86_64.

5. Sitten päivitetään paketticache:

   sudo apt-get update

6. Ja asennetaan Dockerin paketit:

```
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

4.4 Contriboard

4.4.1 Asennuksen valmistelu

Contriboard koostuu viidestä eri Docker-kontista:

1. Client
2. API
3. IO
4. REDIS
5. MongoDB

Kontit tarvitsevat erinäisiä ympäristömuuttujia, jotka määritellään tiedostoissa, jotka luetaan kontin ajon yhteydessä. Tehdään kansio, jossa tiedostot ovat, meidän tapauksessamme se oli /envit/ (Kuva 9).

```
ubuntu@ip-172-31-132-111:~$ ls /envit/
api.txt  client.txt  io.txt
```

Kuva 9 envit kansio.

Eli api.txt sisältää ympäristömuuttujat API-kontille, client.txt Client-kontille jne. Niistä muutettiin lähinnä IP-osoitteet: api.txt tiedostosta MongoDB-kontin sekä

REDIS-kontin osoitteet, client.txt tiedostosta API- ja IO-konttien osoitteet ja io.txt tiedostosta REDIS- ja API-konttien osoitteet.

API-kontin täytyi tietää MongoDB-kontin sijainti, ja tässä tapauksessa se oli samalla koneella eli käytimme koneen omaa IP-osoitetta, myös REDIS-kontti sijaitsi samalla koneella (Kuva 10).

```
ubuntu@ip-172-31-132-111:~$ cat /envit/api.txt
NODE_ENV=production
MONGODB_URL=mongodb://172.31.132.111:27017/CB
REDIS_HOST=172.31.132.111
REDIS_PORT=6379
PORT=9001
TOKEN_SECRET=keksijotain
NEW_RELIC_LICENSE_KEY=XXXXXXXXXXXXXXXXXX
NEW_RELIC_LOG=/home/teamboard/logs/new_relig_api.log
NEW_RELIC_ENABLED=false
```

Kuva 10 api.txt sisältö.

Client-kontin asetuksissa (Kuva 11) täytyi huomioida, että client latautuu selaimeen ja se kommunikoi API:n ja IO:n kanssa ulkoapäin. Eli kyseiset osoitteet ovat HAProxy-koneen julkiset IP-osoitteet. Tällöin Client ottaa yhteyttä API:iin ja IO:hon HAProxyn kautta, joka ohjaa liikenteen oikealle koneelle.

```
ubuntu@ip-172-31-132-111:~$ cat /envit/client.txt
NODE_ENV=production
IO_URL=http://34.244.248.43
IO_PORT=80
API_URL=http://34.244.248.43/api
API_PORT=80
```

Kuva 11 client.txt.

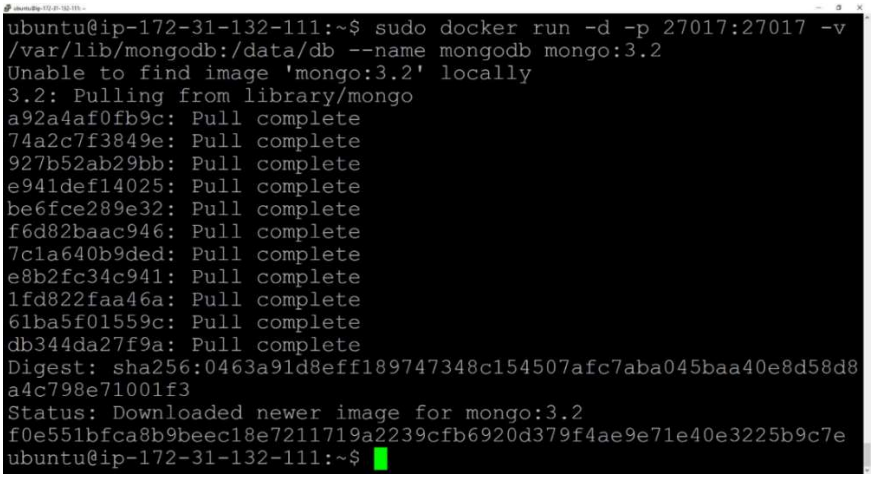
IO-kontissa määritetään REDIS- ja API-konttien osoite. IO:n ja API:n osoitteen täytyi osoittaa HAProxy-palvelimeen, jotta reaaliaikainen päivitys onnistui (Kuva 12).

```
ubuntu@ip-172-31-132-111:~$ cat /envit/io.txt
NODE_ENV=production
REDIS_HOST=172.31.132.111
REDIS_PORT=6379
API_URL=http://172.31.131.239:80/api
PORT=9002
NEW_RELIC_LOG=/home/teamboard/logs/new_relig_io.log
NEW_RELIC_ENABLED=false
```

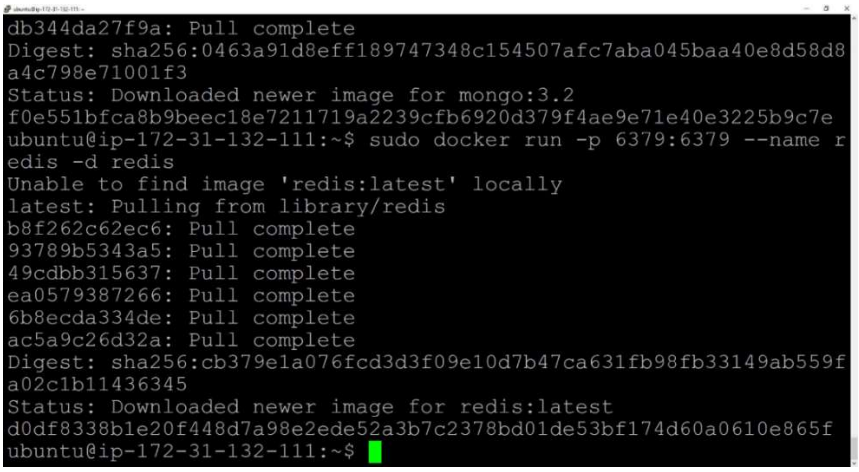
Kuva 12 io.txt.

4.4.2 Konttien käynnistys

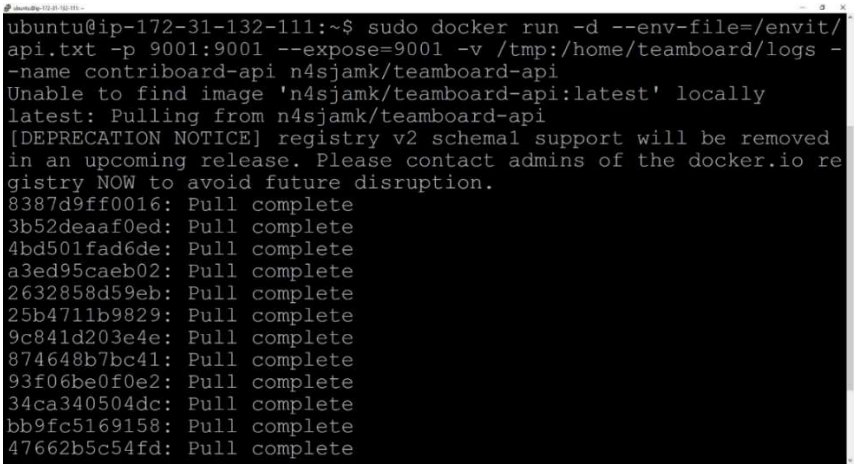
Kun konttien ympäristömuutujat oli määritelty, voitiin käynnistää kontit. Jos konttia ei ole lokaalisti, Docker lataa sen. Kuvissa 13-17 näytetään millä käskyillä kontit käynnistetään.



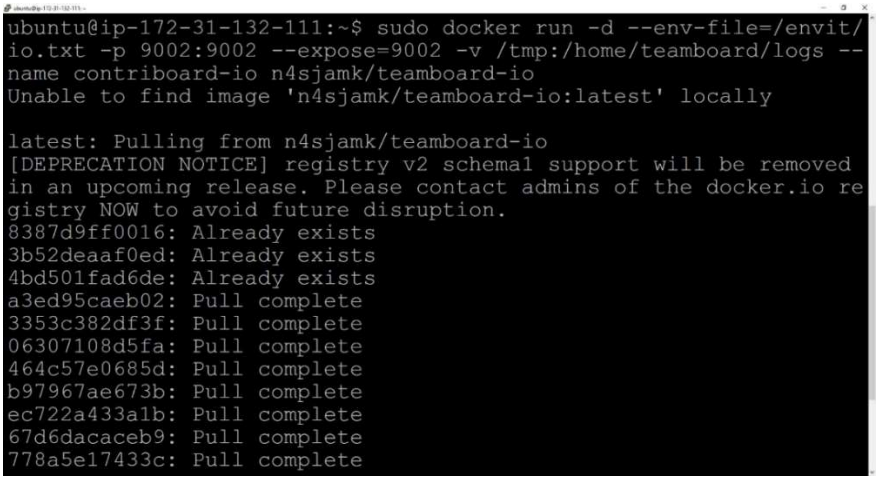
Kuva 13 MongoDB-kontin käynnistys.



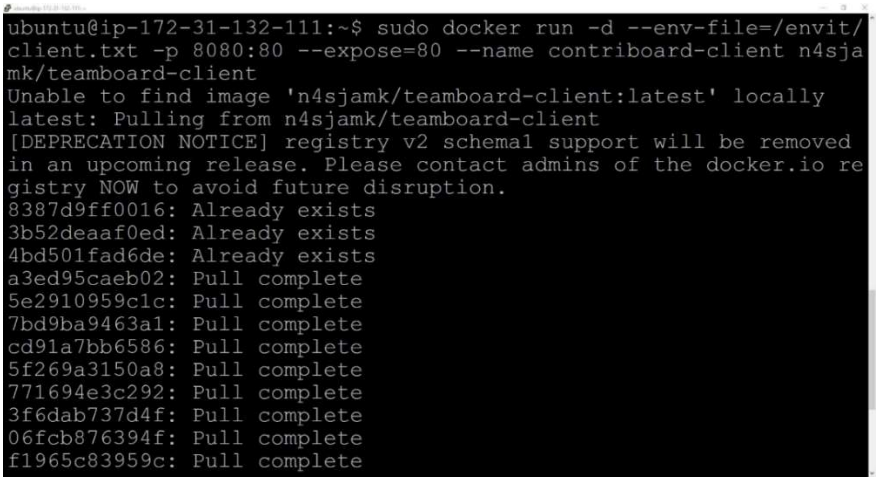
Kuva 14 REDIS-kontin käynnistys.



Kuva 15 API-kontin käynnistys.

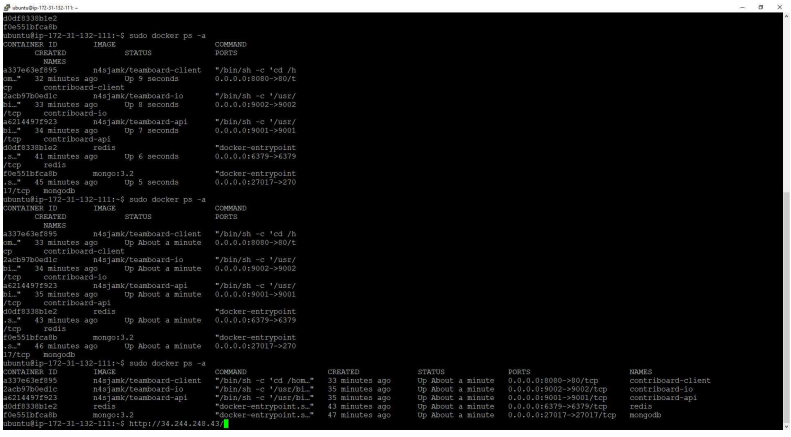


Kuva 16 IO-kontin käynnistys.



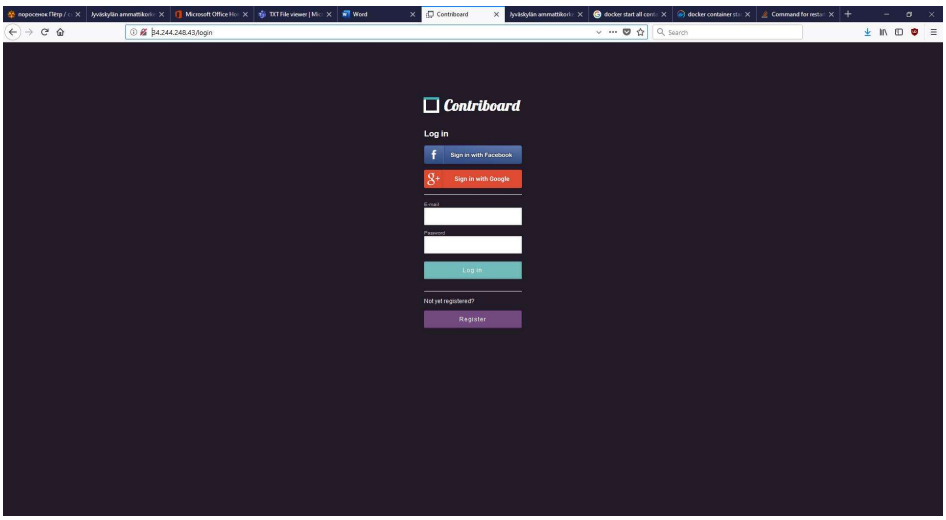
Kuva 17 Client-kontin käynnistys.

Komennolla ”sudo docker ps -a” (Kuva 18) voidaan tarkistaa, että kontit ovat käynnissä.

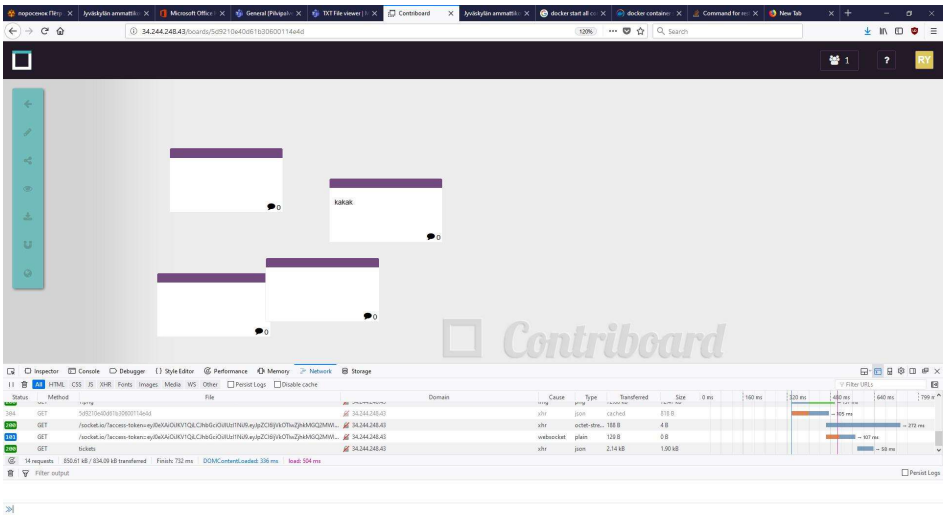


Kuva 18 Docker-kontit käynnissä.

Tämän jälkeen voitiin mennä selaimella IP-osoitteeseen, joka oli HAProxy-koneen julkinen IP. Tämä ohjasi Contriboard-Client:in näkymään selaimeen (Kuva 19 ja Kuva 20).



Kuva 19 Contriboard toiminnassa.



Kuva 20 Contriboard toiminnassa.

5 Pohdinta

Työn aloitus ja sen tekeminen kokonaisuudessaan oli hyvin sulava prosessi. Työssä käytettiin kuivaharjoittelussa kirjoitettuja muistiinpanoja, jotka nopeuttivat työn valmistumista huomattavasti. Contriboardin toiminta oli alussa hieman kankeaa, koska eräs asetus oli jäänyt väärään IP-osoitteeseen. Tästä johtuen, Contriboard ei päivittynyt reaaliajassa verkon ylitse. Tämä saatiin kuitenkin korjattua HAProxyn asetuksista helposti.

Lähteet

Amazon EC2 Instance IP Addressing, Artikkeli docs.aws.amazon.com sivustolla. Viitattu 1.10.2019, https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-instance-addressing.html#concepts-private-addresses.

What is a Conteiner, Artikkeli docker.com sivustolla. Viitattu 27.10.2019. https://www.docker.com/resources/what-container.