

Harjoitustyö 4

Palvelu Googlen pilvipalveluun + oma rekisteri

Tuukka Bordi, TTV18S1

Tatu Alatalo, TTV19SMM

Timo Lehosvuori, TTV18S1

Harjoitustyö

Palveluiden Automatisointi, Juho Pekki

Huhtikuu 2021

Tekniikan ja liikenteen ala

Sisältö

1	Johdanto	3
2	Tehtävänanto	3
2.1	Vaihe 1	3
2.2	Vaihe 2	10
2.3	Vaihe 3	12
3	Pohdinta.....	16
	Lähteet	17

Kuviot

Kuvio 1.	MySQL vm koneen luominen	3
Kuvio 2.	SSH avainten lisääminen MySQL koneeseen.....	4
Kuvio 3.	MySQL kone käynnissä	5
Kuvio 4.	Testikirjautuminen selaimella	5
Kuvio 5.	”Create Similar”-painikkeen avulla luodaan wordpress vm instanssi.....	6
Kuvio 6.	Wordpress instanssi luotu	6
Kuvio 7.	Dockerin asennus Wordpress-vm:ään	7
Kuvio 8.	Openrekisteri sertifikaatin siirto	7
Kuvio 9.	MySQL dockerize skriptin ajaminen	8
Kuvio 10.	Myslikontti käynnissä	8
Kuvio 11.	Wp_dockerize skriptin ajo	9
Kuvio 12.	Wpkontti käynnissä	9
Kuvio 13.	Google cloud shellissä luotu palomuurisääntö	9
Kuvio 14.	WP-Kontti käynnissä osoitteessa 35.228.230.5	10
Kuvio 15.	Docker-registry vm instanssi luotu.....	11
Kuvio 16.	Oman sertifikaatin luonti	11
Kuvio 17.	Oman sertifikaatin uudelleen nimeäminen	11
Kuvio 18.	Sertifikaatin siirto omaan docker-rekisterissä	12
Kuvio 19.	Rekisterin käynnistäminen	12
Kuvio 20.	Itseluodun sertifikaattien siirto koneille	12
Kuvio 21.	Oman rekisterin lisäys mysql-koneen host tiedostoon.....	13

Kuvio 22. Imagien merkitseminen.....	13
Kuvio 23. WordPress imagen lähettäminen omaan rekisteriin	13
Kuvio 24. WordPressin lataaminen oman skriptin avulla	14
Kuvio 25. MySQL imagen lataaminen omasta rekisteristä	14
Kuvio 26. Myslikontin käynnistys omasta rekisteristä ladatulla imagella.....	14
Kuvio 27. Wpkontin käynnistäminen omasta rekisteristä ladatulla imagella	15
Kuvio 28. WP-Kontin sivut jälleen pystyssä.....	15

1 Johdanto

Tämä harjoitustyö on osa Jyväskylän Ammattikorkeakoulun Palveluiden Automatisointi -kurssia. Harjoituksen tarkoituksena on tutustua Googlen pilvipalveluun ja pystyttää oma Docker rekisteri.

2 Tehtävänanto

2.1 Vaihe 1

Aloitimme tehtävän luomalla projektin, lisäämällä muut ryhmäläiset siihen ja loimme Googlen Compute Engineen koneen nimellä “mysql”:

The screenshot displays the Google Cloud Platform interface for creating a new VM instance. On the left, a sidebar lists four options: 'New VM instance' (selected), 'New VM instance from template', 'New VM instance from machine image', and 'Marketplace'. The main area is titled 'Create an instance' and contains configuration fields. The 'Name' field is set to 'mysql'. The 'Region' is 'europe-north1 (Finland)' and the 'Zone' is 'europe-north1-a'. Under 'Machine configuration', the 'Machine family' is 'General-purpose', the 'Series' is 'E2', and the 'Machine type' is 'e2-micro (2 vCPU, 1 GB memory)'. The 'Boot disk' section shows a 'New 10 GB balanced persistent disk' with the image 'Ubuntu 18.04 LTS'. A price estimate of '\$7.83 monthly estimate' is displayed on the right.

Kuvio 1. MySQL vm koneen luominen

Loimme tarvitsemamme SSH –avaimet, jotta pystyimme yhdistämään koneeseen:

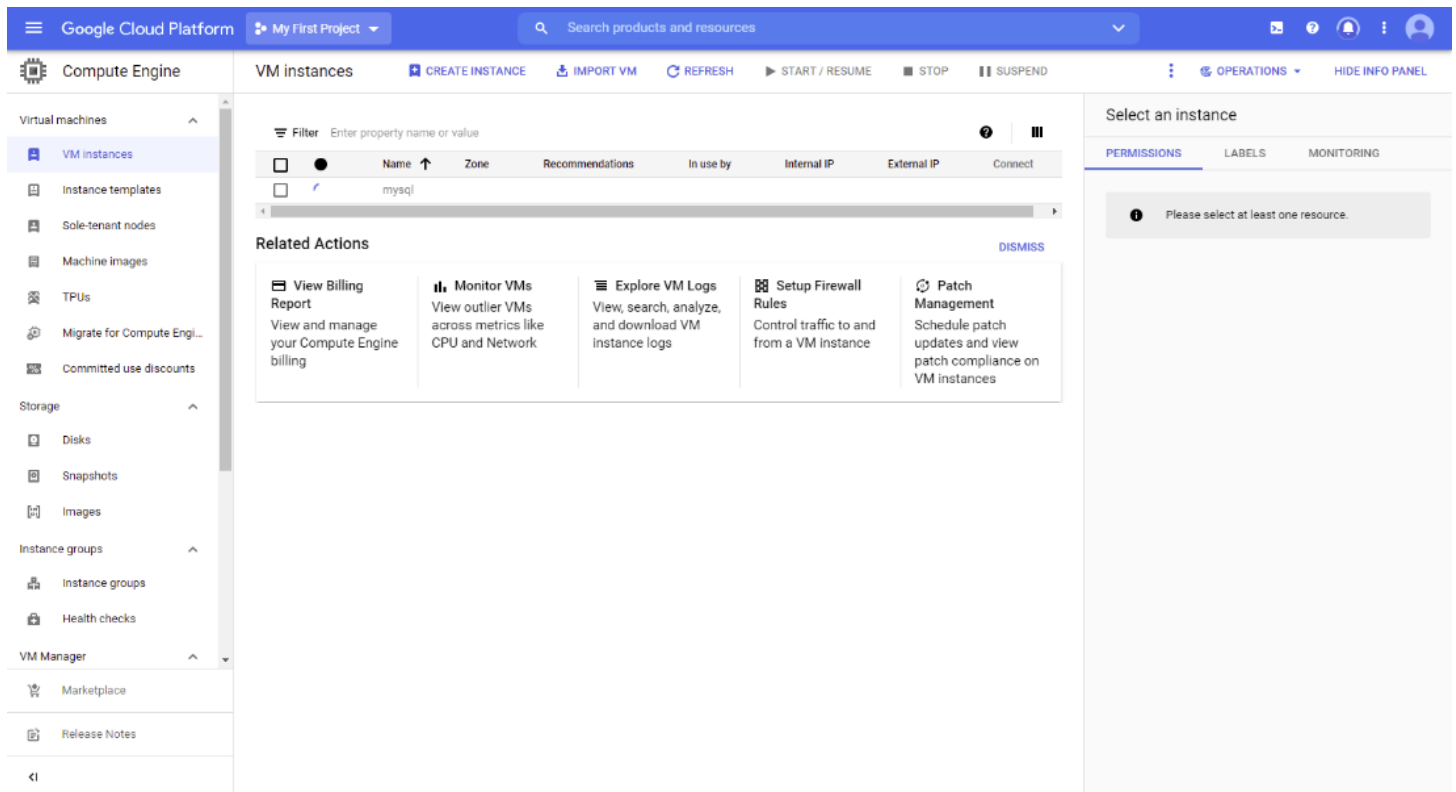
SSH Keys

☐ Block project-wide SSH keys

tuukka	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDQ...Z3yIVJ6jWYUtCKSaa53 tuukka@tuukka-N501VW
tuukka	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDQ... CLnI5F8WTZmTbzHWZ+xMr tuukka@zorin-E1050
tuukan-windows-20210219	ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQByPgA1...ZSTs1aPHeVnOT2x1 tuukan- windows-20210219
tatu	ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEAIE75... k0ObhKy2/l1XSmlBbjaJwREz6MC42g4aw== tatu
timo	ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEA1Zho... RVgKel2aIDdFKQVT/zfjl1pjRXxyjrnzw== timo

Kuvio 2. SSH avainten lisääminen MySQL koneeseen

Tämän jälkeen käynnistimme koneen ja testasimme, toimiiko yhteydet:



Kuvio 3. MySQL kone käynnissä

```

pa_group8@mysql: ~ - Google Chrome
ssh.cloud.google.com/projects/advance-elixir-302410/zones/europe-north1-a/instances/mysql?authuser=0&hl=en_US&projectNumber...
Connected, host fingerprint: ssh-rsa 0 6E:52:AD:5F:30:54:25:56:3B:AF:19:86:5C:04
:84:57:B9:2E:1C:69:B8:0B:9F:7E:A9:E5:6A:76:E5:96:02:F0
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1040-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Mar 29 07:38:16 UTC 2021

System load:  0.05          Processes:      121
Usage of /:   16.8% of 9.52GB Users logged in: 0
Memory usage: 20%          IP address for ens4: 10.166.0.2
Swap usage:   0%

0 packages can be updated.
0 of these updates are security updates.

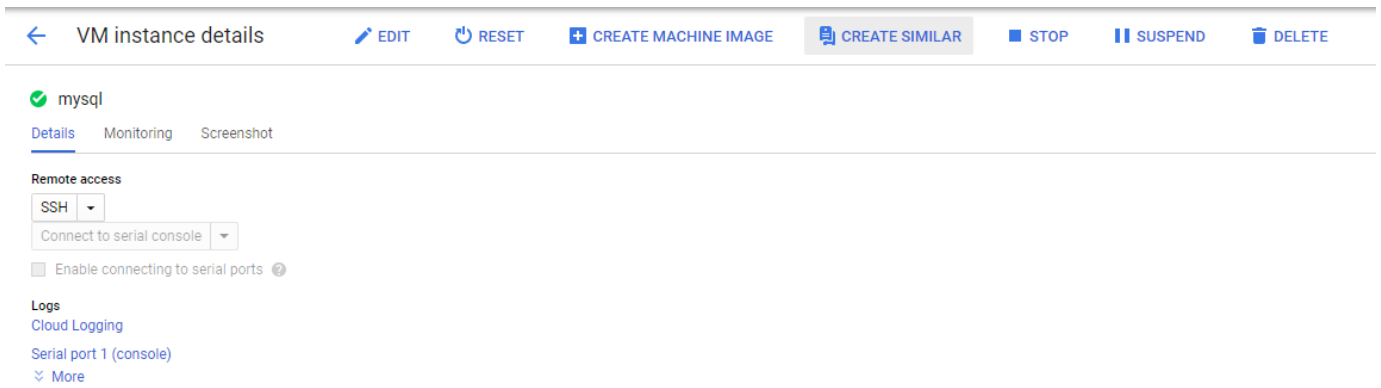
New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

pa_group8@mysql:~$ 

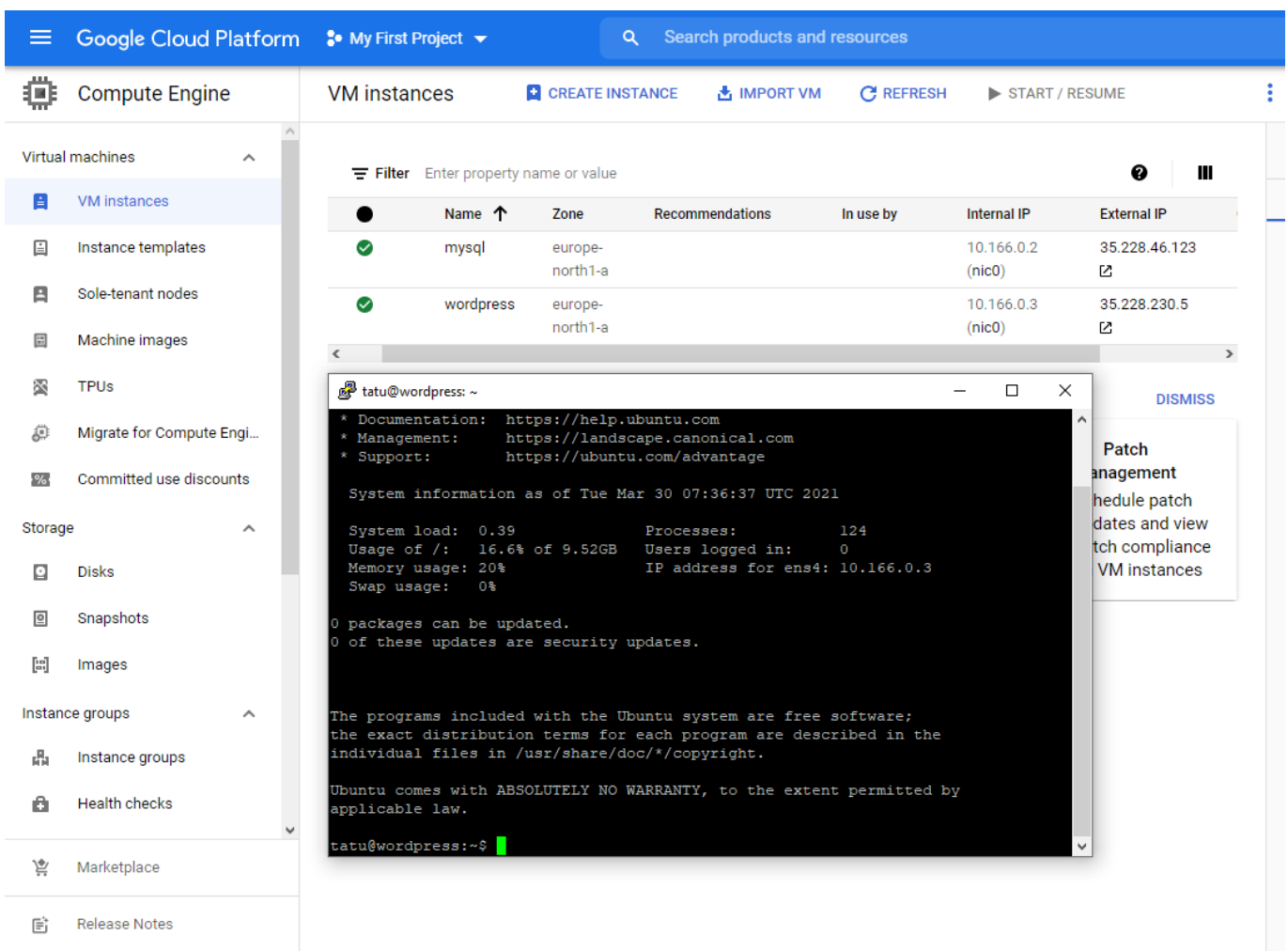
```

Kuvio 4. Testikirjautuminen selaimella

Kun saimme ensimmäisen koneen luotua (tässä vaiheessa koneessa ei ole vielä dok-keria), loimme tarvitsemamme WordPress koneen käyttäen “Create similar” paini-
ketta:



Kuvio 5. "Create Similar"-painikkeen avulla luodaan wordpress vm instanssi



Kuvio 6. Wordpress instanssi luotu

Koneen asennettua, asensimme siihen Dockerin manuaalisesti:

```
tatu@wordpress: ~
Unpacking docker-ce-cli (5:20.10.5~3-0~ubuntu-bionic) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../3-docker-ce_5%3a20.10.5~3-0~ubuntu-bionic_amd64.deb ...
Unpacking docker-ce (5:20.10.5~3-0~ubuntu-bionic) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../4-docker-ce-rootless-extras_5%3a20.10.5~3-0~ubuntu-bionic_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:20.10.5~3-0~ubuntu-bionic) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../5-libltdl7_2.4.6-2_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.6-2) ...
Setting up containerd.io (1.4.4-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up libltdl7:amd64 (2.4.6-2) ...
Setting up docker-ce-cli (5:20.10.5~3-0~ubuntu-bionic) ...
Setting up pigz (2.4-1) ...
Setting up docker-ce (5:20.10.5~3-0~ubuntu-bionic) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Progress: [ 87%] [#####.....]
```

Kuvio 7. Dockerin asennus Wordpress-vm:ään

Kun olimme asentaneet WordPress koneen ja Dockerin, siirsimme Openrekisteri sertifikaatin oikeaan polkuun MySQL koneessa:

```
tatu@mysql:~$ sudo nano /etc/hosts
tatu@mysql:~$ sudo mkdir -p /etc/docker/certs.d/openrekisteri.com:443/
tatu@mysql:~$ sudo cp /home/tuukka/ht3_vagrant/files/ca.crt /etc/docker/certs.d/openrekisteri.com:443/ca.crt
tatu@mysql:~$ ls -ls /etc/docker/certs.d/openrekisteri.com\443/
total 4
4 -rwxr-xr-x 1 root root 1984 Mar 30 08:11 ca.crt
tatu@mysql:~$
```

Kuvio 8. Openrekisteri sertifikaatin siirto

Ajoimme aikaisemmassa harjoituksessa luomamme “docker_init.sh” skriptin, joka asensi koneeseen Dockerin ja “mysql_dockerize.sh” -skriptin, joka asensi koneeseen MySQL Dockerin ja käynnisti sen:


```
tatu@mysql: ~
4 -rwxrwxr-x 1 tuukka tuukka 943 Mar 30 07:08 docker_init.sh
4 -rw-rw-r-- 1 tuukka tuukka 951 Mar 30 07:16 docker_init_gce.sh
4 -rwxrwxr-x 1 tuukka tuukka 960 Mar 30 07:08 mysql_dockerize.sh
4 -rwxrwxr-x 1 tuukka tuukka 1147 Mar 30 07:08 wp_dockerize.sh
tatu@mysql:~$ nano /home/tuukka/ht3_vagrant/scripts/mysql_dockerize.sh
tatu@mysql:~$ sh /home/tuukka/ht3_vagrant/scripts/mysql_dockerize.sh
Checking if container is running...
/home/tuukka/ht3_vagrant/scripts/mysql_dockerize.sh: 6: [: !=: unexpected operator
No containers found. Downloading a new image "openrekisteri.com:443/ryhma8_mysql"
...
Using default tag: latest
latest: Pulling from ryhma8_mysql
d519e2592276: Extracting 24.48MB/26.71MB
d22d2dfcfa9c: Download complete
b3afe92c540b: Download complete
b66e1d55d718: Download complete
68bbd8eb6c49: Download complete
ba484d25fdf8: Download complete
f3721839f3a0: Download complete
0624c9c3aad2: Download complete
075626307450: Download complete
69098fc75611: Download complete
ab7243881889: Download complete
8634d847d477: Download complete
3d201307e5c6: Download complete
9454dfd86ead: Download complete
```

Kuvio 9. MySQL dockerize skriptin ajaminen

```
tatu@mysql: ~
Using default tag: latest
latest: Pulling from ryhma8_mysql
d519e2592276: Pull complete
d22d2dfcfa9c: Pull complete
b3afe92c540b: Pull complete
b66e1d55d718: Pull complete
68bbd8eb6c49: Pull complete
ba484d25fdf8: Pull complete
f3721839f3a0: Pull complete
0624c9c3aad2: Pull complete
075626307450: Pull complete
69098fc75611: Pull complete
ab7243881889: Pull complete
8634d847d477: Pull complete
3d201307e5c6: Pull complete
9454dfd86ead: Pull complete
Digest: sha256:227eeadb2b1a8742f479b694add28c240699488815765015cc24d436e35b887
Status: Downloaded newer image for openrekisteri.com:443/ryhma8_mysql:latest
mysql_data
mysql_conf
7d3d4c65a4ff6c16d2bc7c1204cc6d3dbcc5113d8d01aa4f6cf0506279b72e9c
tatu@mysql:~$ sudo docker ps -a
CONTAINER ID        IMAGE                                     COMMAND                  CREATED
7d3d4c65a4ff        openrekisteri.com:443/ryhma8_mysql      "docker-entrypoint.s..." 26 s
econds ago         Up 23 seconds      0.0.0.0:3306->3306/tcp    mysqlkontti
tatu@mysql:~$
```

Kuvio 10. Myslikontti käynnissä

”Myslikontin” käynnistämisen jälkeen ajoimme ”wp_dockerize.sh” -skriptin wordpress-koneella, joka asensi WordPress kontin ja käynnisti sen:

```
tatu@wordpress:~$ sh /home/tuukka/ht3_vagrant/scripts/wp_dockerize.sh "10.166.0.2" "35.228.230.5"
Checking if container is running...
Found container. Stopping and deleting wpkontti...
wpkontti
Deleted Containers:
80c2104b9f88d687ba175b4cb28a97ef91e41bfc35394d7ab3fc856830443704

Total reclaimed space: 5.568MB
openrekisteri.com:443/ryhma8_wp
Found Wordpress installation on /var/www/html/. Skipping installation...
/usr/local/bin/wp does not exist. Installing...
% Total      % Received % Xferd Average Speed Time Time Time Current
             Dload Upload Total Spent Left Speed
100 5437k 100 5437k 0 0 9.9M 0 --:--:-- --:--:-- --:--:-- 9.9M
/var/www/html/wp-config.php does not exist or it has wrong values. Creating a configuration file...
Success: Generated 'wp-config.php' file.
Database is empty. Creating tables...
sh: 1: -t: not found
Warning: Unable to create directory wp-content/uploads/2021/03. Is its parent directory writable by the server?
Success: WordPress installed successfully.
Changing WordPress theme...
Installing Graphene (2.8.4)
Downloading installation package from https://downloads.wordpress.org/theme/graphene.2.8.4.zip...
```

Kuvio 11. Wp_dockerize skriptin ajo

```
tatu@wordpress:~$ sudo docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED
STATUS        PORTS
330df057f3e3   openrekisteri.com:443/ryhma8_wp     "/bin/sh -c /wp.sh"     4 minutes
ago          Up 4 minutes   0.0.0.0:80->80/tcp, 443/tcp   wpkontti
tatu@wordpress:~$
```

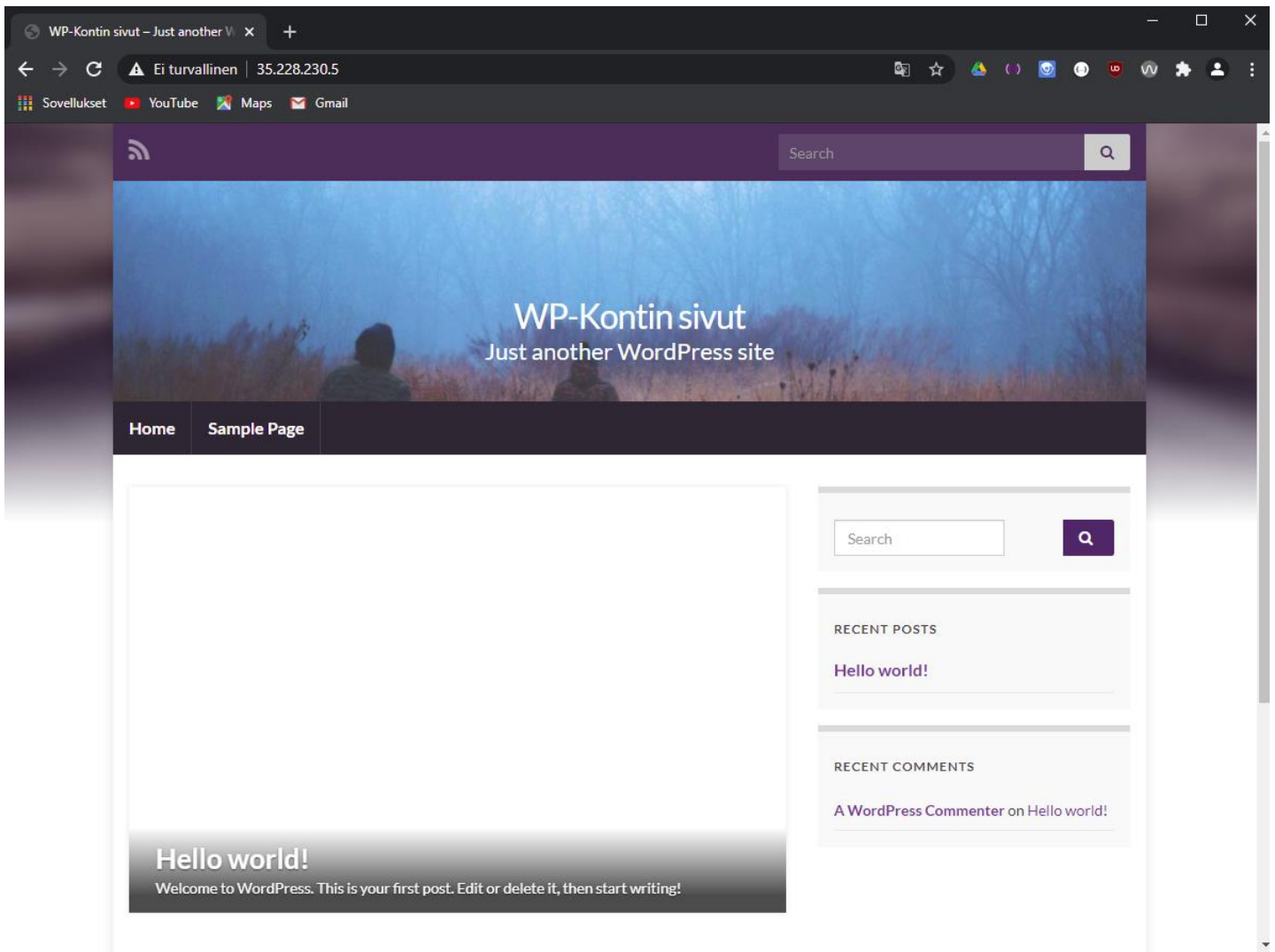
Kuvio 12. Wpkontti käynnissä

Huomasimme kuitenkin, että ”wpkontti” ei saanut yhteyttä mysqlkonttiin, joten loimme Google Cloud Consoleen palomuurisäännön. Tähän otimme mallia Google Cloudin omasta dokumentaatiosta (Firewall considerations in production environments). Huomasimme kuitenkin myöhemmin, että wpkontti käytti tietokantaan väärää IP-osoitetta, eikä meidän olisi tarvinnut luoda sääntöjä lainkaan:

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to advance-elixir-302410.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
pa_group8@cloudshell:~ (advance-elixir-302410) $ gcloud compute instances add-tags mysql --tags mysql-server --zone=europe-north1-a
Updated [https://www.googleapis.com/compute/v1/projects/advance-elixir-302410/zones/europe-north1-a/instances/mysql].
pa_group8@cloudshell:~ (advance-elixir-302410) $ gcloud compute instances add-tags wordpress --tags mysql-client --zone=europe-north1-a
Updated [https://www.googleapis.com/compute/v1/projects/advance-elixir-302410/zones/europe-north1-a/instances/wordpress].
pa_group8@cloudshell:~ (advance-elixir-302410) $ gcloud compute firewall-rules create "mysql-remote-access" \
> --allow tcp:3306 --source-tags "mysql-client" \
> --target-tags "mysql-server"
Creating firewall...Created [https://www.googleapis.com/compute/v1/projects/advance-elixir-302410/global/firewalls/mysql-remote-access].
Creating firewall...done.
NAME          NETWORK  DIRECTION  PRIORITY  ALLOW  DENY  DISABLED
mysql-remote-access  default  INGRESS    1000      tcp:3306  False
```

Kuvio 13. Google cloud shellissä luotu palomuurisääntö

”Wpkontti” tarvitsi tietokannan IP-osoitteeksi mysql-koneen sisäisen IP-osoitteen 10.166.0.2 ulkoisen 35.228.46.123 sijaan. Vaihdoin IP-osoitteen oikeaksi, jonka jälkeen yhteys toimi ja palvelu saatiin käyntiin:



Kuvio 14. WP-Kontti käynnissä osoitteessa 35.228.230.5

2.2 Vaihe 2

Rekisterin pystytyksessä noudatimme opettajan ohjeita (Test an insecure registry n.d, Deploy a registry server n.d). Alussa loimme rekisteripalvelimen nimellä “docker-registry”:

VM instances							
CREATE INSTANCE IMPORT VM REFRESH START / RESUME STOP SUSPEND							
Filter Enter property name or value							
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP
<input type="checkbox"/>	<input checked="" type="checkbox"/>	docker-registry	europe-north1-a			10.166.0.4 (nic0)	35.228.233.219 ↗
<input type="checkbox"/>	<input checked="" type="checkbox"/>	mysql	europe-north1-a			10.166.0.2 (nic0)	35.228.46.123 ↗
<input type="checkbox"/>	<input checked="" type="checkbox"/>	wordpress	europe-north1-a			10.166.0.3 (nic0)	35.228.230.5 ↗

Kuvio 15. Docker-registry vm instanssi luotu

Loimme seuraavaksi sertifikaatin ohjeiden mukaan (Test an insecure registry n.d):

```
tuukka@docker-registry:~$ openssl req \
> -newkey rsa:4096 -nodes -sha256 -keyout certs/domain.key \
> -x509 -days 365 -out certs/domain.crt
Can't load /home/tuukka/.rnd into RNG
140273786024384:error:2406F079:random number generator:RAND_load_file:Cannot open file:../crypto/rand/randfile.c:88:Filename=/home/tuukka/.rnd
Generating a RSA private key
.....++++
.....++++
writing new private key to 'certs/domain.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FI
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:JAMK
```

Kuvio 16. Oman sertifikaatin luonti

Sen jälkeen nimesimme sertifikaatit omaan käyttöön sopiviksi:

```
tuukka@docker-registry:~$ mv certs/domain.crt certs/ryhma8.crt
tuukka@docker-registry:~$ ls certs/
domain.key  ryhma8.crt
tuukka@docker-registry:~$ mv certs/domain.key certs/ryhma8.key
tuukka@docker-registry:~$ ls certs/
ryhma8.crt  ryhma8.key
tuukka@docker-registry:~$
```

Kuvio 17. Oman sertifikaatin uudelleen nimeäminen

Seuraavaksi loimme Dockerin tarvitsemat kansiot rekisteriä varten. Otimme mallia harjoitustyö 2:sta, jossa tehtiin sama asia:

```
tatu@docker-registry:~$ sudo mkdir -p /etc/docker/certs.d/ryhma8-rekisteri.com:443
tatu@docker-registry:~$ sudo cp /home/tuukka/certs/ryhma8.crt /etc/docker/certs.d/ryhma8-rekisteri.com:443/ryhma8.crt
tatu@docker-registry:~$ ls -ls /etc/docker/certs.d/ryhma8-rekisteri.com:443/
total 4
4 -rw-r--r-- 1 root root 2106 Mar 31 07:32 ryhma8.crt
tatu@docker-registry:~$
```

Kuvio 18. Sertifikaatin siirto omaassa docker-rekisterissä

Viimeiseksi laitoimme rekisterin pyörimään ohjeiden mukaan sovellettuna (Deploy a registry server n.d). Alla tämä kuvattuna:

```
tuukka@docker-registry:~$ sudo docker run -d --restart=always --name registry
-v "$(pwd)/certs:/certs -e REGISTRY_HTTP_ADDR=0.0.0.0:443 -e REGISTRY_H
TTP_TLS_CERTIFICATE=certs/ryhma8.crt -e REGISTRY_HTTP_TLS_KEY=certs/ryhma8.key
-p 443:443 registry:2
626c3d2a12ce7284fa42d894033b549fe5c21591c2927022989b7145b4058114
tuukka@docker-registry:~$ sudo docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS
PORTS         NAMES
626c3d2a12ce   registry:2 "/entrypoint.sh /etc..." 5 seconds ago   Up 4 second
s
0.0.0.0:443->443/tcp, 5000/tcp registry
```

Kuvio 19. Rekisterin käynnistäminen

Näiden vaiheiden jälkeen rekisterin pitäisi olla toimintakunnossa! Testaamaan pääsemme seuraavassa vaiheessa.

2.3 Vaihe 3

Aluksi siirsimme sertifikaatit kohdekoneille *scp*-työkalulla:

```
tuukka@zorin-E1050:~$ scp -r docker_registry:/home/tuukka/certs/ .
ryhma8.key          100% 3272    50.0KB/s   00:00
ryhma8.crt          100% 2106    39.2KB/s   00:00
tuukka@zorin-E1050:~$ scp certs google_wordpress:~
certs: not a regular file
tuukka@zorin-E1050:~$ scp -r certs google_wordpress:~
ryhma8.key          100% 3272    52.7KB/s   00:00
ryhma8.crt          100% 2106    34.0KB/s   00:00
tuukka@zorin-E1050:~$ scp -r certs google_mysql:~
ryhma8.key          100% 3272    53.0KB/s   00:00
ryhma8.crt          100% 2106    39.1KB/s   00:00
tuukka@zorin-E1050:~$
```

Kuvio 20. Itseluodun sertifikaattien siirto koneille

Tämän jälkeen lisäsimme manuaalisesti rekisterin dns-nimen */etc/hosts* kansioon, jotta voimme käyttää hyväksi palvelimen domain-nimeä ip-osoitteen sijaan. Lisäksi sertifikaattiin on lisätty kyseinen domain-nimi, joten tämä toimenpide on muutenkin pakollinen:

```

GNU nano 2.9.3 /etc/hosts
127.0.0.1 localhost

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
169.254.169.254 metadata.google.internal metadata
35.228.57.182 openrekisteri.com
35.228.233.219 ryhma8-rekisteri.com

```

[Wrote 12 lines]

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

Kuvio 21. Oman rekisterin lisäys mysql-koneen host tiedostoon

Tagitimme imaget pushaamista varten seuraavassa vaiheessa:

```

tuukka@docker-registry:~$ sudo docker tag openrekisteri.com:443/ryhma8_wp ryhma8
-rekistrii.com:443/wordpress
tuukka@docker-registry:~$ sudo docker tag openrekisteri.com:443/ryhma8_mysql ryh
ma8-rekistrii.com:443/mysql
tuukka@docker-registry:~$

```

Kuvio 22. Imagien merkitseminen

Seuraavaksi pääsimme imagien pushaamiseen:

```

tuukka@docker-registry:~$ sudo docker push ryhma8-rekisteri.com:443/wordpress
Using default tag: latest
The push refers to repository [ryhma8-rekisteri.com:443/wordpress]
671cec6aad24: Pushing 80.55MB
f850f616bf83: Pushed
5dc980197467: Pushed
cc45506c4447: Pushed
6ec4d4ce53cc: Pushed
9a60d912a14f: Pushed
ce60a0c97d4a: Pushed
e1cd0107ea85: Pushed
914a1eddd57a: Pushed
0ff9183bd099: Pushed
d666585087a1: Pushed
bc0429138e0d: Pushing 23.77MB/45.89MB
02eef72b445f: Pushed
e45a78df7536: Pushing 19.69MB/227.3MB
ddcd8d2fcf7e: Pushed
87c8a1d8f54f: Waiting

```

Kuvio 23. WordPress imagen lähettäminen omaan rekisteriin

Sama toistettiin mysql-koneella ja lähetettiin mysql-image omaan rekisteriin. Tämän jälkeen testattiin imagien lataamista -eli pullaamista- omasta rekisteristä. Käytimme tähän harjoitustyö 3:ssa luomaamme skriptiä pienillä muokkauksilla. Skripti pysäyttää ja tuhoaa käynnissä olevat kontit, lataa uuden imagen ja käynnistää sen. Muokatussa skriptissä ladataan imaget omasta rekisteristä opettajan rekisterin sijaan:


```

tuukka@wordpress:~$ sudo sh ht4/wp_dockerize.sh
Checking if container is running...
Found container. Stopping and deleting wpkontti...
wpkontti
Deleted Containers:
330df057f3e3fb5f2702ca9c4519e6cd1a88f34b2b5caf98ac6390c2c889db

Total reclaimed space: 10.73MB
ryhma8-rekisteri.com:443/wordpress
Unable to find image 'ryhma8-rekisteri.com:443/wordpress:latest' locally
latest: Pulling from wordpress
Digest: sha256:bb1797fa5db4bf96ac164bc7e22dff82b7e847d2ccbb0c19f05a65eec27965e9
Status: Downloaded newer image for ryhma8-rekisteri.com:443/wordpress:latest
10a6a3ecaaeb62345392ccd93e04eca209c5d5a8e56e72c4170974f806fe5c31
tuukka@wordpress:~$ sudo docker image ls
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
openrekisteri.com:443/ryhma8_wp   latest         33ff496fa516    5 days ago     490MB
ryhma8-rekisteri.com:443/wordpress latest         33ff496fa516    5 days ago     490MB
tuukka@wordpress:~$

```

Kuvio 24. WordPressin lataaminen oman skriptin avulla

MySQL:n pullaamisen hoidimme sen sijaan manuaalisesti (kuva alla):

```

tatu@mysql:~$ sudo docker pull ryhma8-rekisteri.com:443/mysql
Using default tag: latest
latest: Pulling from mysql
Digest: sha256:227eeadb2b1a8742f479b694addd28c240699488815765015cc24d436e35b887
Status: Downloaded newer image for ryhma8-rekisteri.com:443/mysql:latest
ryhma8-rekisteri.com:443/mysql:latest
tatu@mysql:~$ sudo docker images -ls
unknown shorthand flag: 'l' in -ls
See 'docker images --help'.
tatu@mysql:~$ sudo docker image ls
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
openrekisteri.com:443/ryhma8_mysql   latest         220b02364a2f    2 weeks ago     352MB
ryhma8-rekisteri.com:443/mysql       latest         220b02364a2f    2 weeks ago     352MB
tatu@mysql:~$

```

Kuvio 25. MySQL imagen lataaminen omasta rekisteristä

Molemmat imaget ladattiin onnistuneesti. Seuraavaksi poistimme kaikki kontit ja imaget ja pyöritimme skriptit uudestaan kokeillaksemme, latautuvatko imaget ja toimivatko skriptit. Ensimmäisenä MySQL-imagien lataaminen ja ajaminen:

```

Digest: sha256:227eeadb2b1a8742f479b694addd28c240699488815765015cc24d436e35b887
Status: Downloaded newer image for ryhma8-rekisteri.com:443/mysql:latest
ryhma8-rekisteri.com:443/mysql:latest
mysql_data
mysql_conf
11bebe01e6976a5daba5a6513db13e880c2ca5aaf3daa319258405fb3b820afd
tuukka@mysql:~$ docker ps -a
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://%2Fvar%2Frun%2Fdocker.sock/v1.24/containers/json?all=1: dial unix /var/run/docker.sock: connect: permission denied
tuukka@mysql:~$ sudo docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED
STATUS        PORTS                NAMES
11bebe01e697  ryhma8-rekisteri.com:443/mysql      "docker-entrypoint.s..." 26 seconds ago
Up 24 seconds  0.0.0.0:3306->3306/tcp  myslikontti
tuukka@mysql:~$

```

Kuvio 26. Myslikontin käynnistys omasta rekisteristä ladatulla imagella

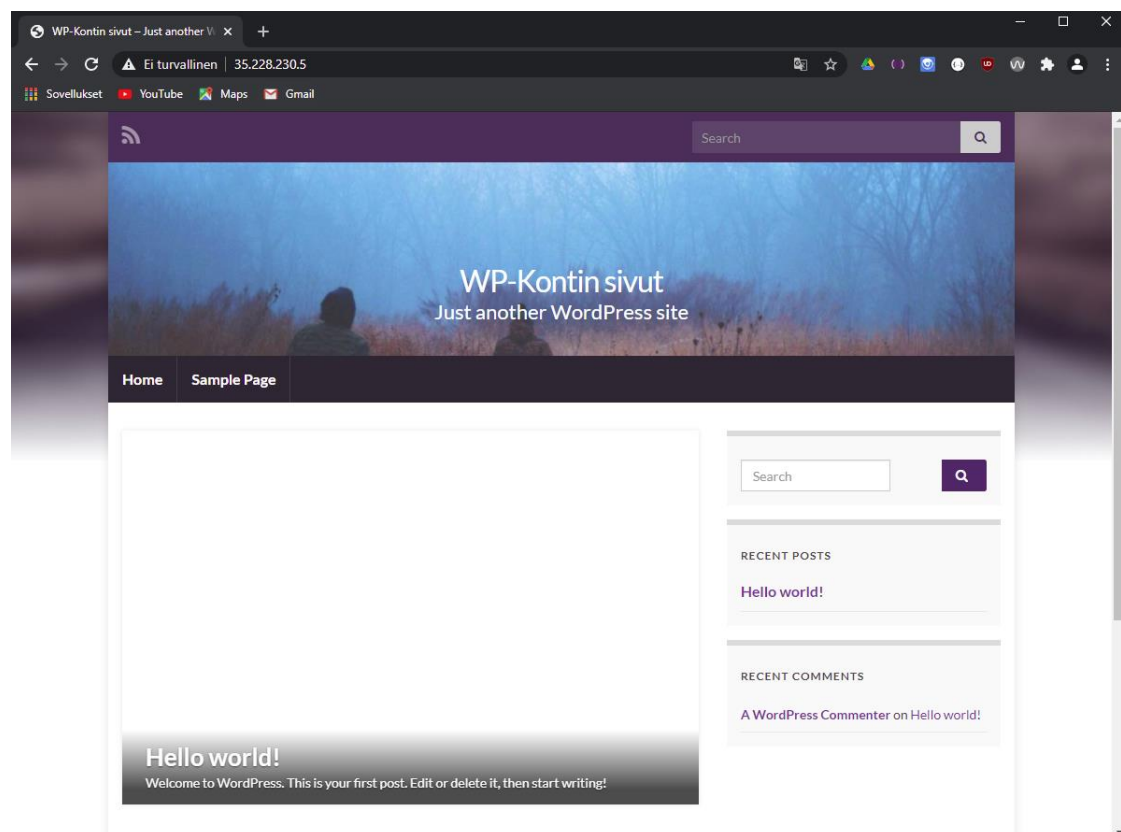
Kuvassa näkyy skriptin lopputulos. Skripti pyöri onnistuneesti ja käynnisti kontin.

Seuraavaksi teimme saman WordPressille:

```
tatu@wordpress: ~
3bb2e8051594: Pull complete
4c761b44e2cc: Pull complete
c2199db96575: Pull complete
1b9a9381eea8: Pull complete
fd07bbc59d34: Pull complete
72b73ab27698: Pull complete
983308f4f0d6: Pull complete
6c13f026e6da: Pull complete
e5e6cd163689: Pull complete
5c5516e56582: Pull complete
154729f6ba86: Pull complete
dd2992917525: Pull complete
6590df1a7e39: Pull complete
Digest: sha256:bb1797fa5db4bf96ac164bc7e22dff82b7e847d2ccbb0c19f05a65eec27965e9
Status: Downloaded newer image for ryhma8-rekisteri.com:443/wordpress:latest
ryhma8-rekisteri.com:443/wordpress:latest
wp
f62e8ff41abd1a5b2d36ef88388860bb64dc68fdbel03759dbe2713fabe21d02
tatu@wordpress:~$ sudo docker ps -a
CONTAINER ID        IMAGE                                     COMMAND                  CREATED             STATUS
PORTS              NAMES
f62e8ff41abd        ryhma8-rekisteri.com:443/wordpress    "/bin/sh -c /wp.sh"     18 seconds ago     Up 16
seconds           0.0.0.0:80->80/tcp, 443/tcp            wpkonttti
tatu@wordpress:~$
```

Kuvio 27. Wpkontin käynnistäminen omasta rekisteristä ladatulla imagella

Tämäkin skripti toimi odotetunlaisesti. Viimeiseksi katsoimme, onko WordPress saatavilla niin kuin pitääkin:



Kuvio 28. WP-Kontin sivut jälleen pystyssä

WordPress-sivu avautui odotetusti. Tehtävä valmis.

3 Pohdinta

Aiemmin tekemistämme skripteistä oli hyötyä tässä tehtävässä.

Meillä oli hieman ongelmia saada koneet keskustelemaan keskenään ja luulimme, että ongelma oli palomuurissa, ja teimmekin säännön, joka sallii SQL-liikenteen.

Ongelmaksi paljastuikin se, että käytimme väärää IP-osoitetta: julkista sisäisen IP:n sijaan. Sisäistä IP-osoitetta käyttäen kaikki toimikin hyvin.

Lähteet

Deploy a registry server. N.d. Artikkelin Dockerin sivuilla. Viitattu 1.4.2021.

<https://docs.docker.com/registry/deploying/#get-a-certificate>.

Firewall Considerations in production environments. 2021. Google Cloudin dokumentaatio kuinka luodaan palomuurisääntö MySQL:n etäkäyttöön. Viitattu 1.4.2021.

https://cloud.google.com/solutions/mysql-remote-access#firewall_considerations_in_production_environments.

Test an insecure registry. N.d. Artikkelin Dockerin sivuilla. Viitattu 1.4.2021

<https://docs.docker.com/registry/insecure/#use-self-signed-certificates>.