

Ohjelmistokehityksen teknologioita - Seminaarityö

Openshift

2.Infra ja DevOps

Matias Sanden

Sisältö

<i>Tiivistelmä</i>	<i>1</i>
<i>1 Johdanto/Ylätason esittely</i>	<i>1</i>
<i>2 Käytetyt tekniikat</i>	<i>2</i>
<i>2.1 Paikallisen kehityksen työkalut</i>	<i>2</i>
<i>2.2 Verkkokehityksen työkalut</i>	<i>2</i>
<i>3 Paikallinen kehitys</i>	<i>3</i>
<i>3.1 Sovelluksen valmistelu</i>	<i>3</i>
<i>3.2 Tietokannan valmistelu</i>	<i>4</i>
<i>3.3 Konttien yhteistoiminta</i>	<i>4</i>
<i>4 Openshift</i>	<i>5</i>
<i>4.1 Docker image</i>	<i>5</i>
<i>4.2 Openshift konfiguraatio</i>	<i>6</i>
<i>4.3 Tietokantakontti Openshiftissä</i>	<i>6</i>
<i>4.4 Sovellus Openshiftissä</i>	<i>7</i>
<i>4.5 Kokonaisuuden hallinta Openshiftissä</i>	<i>9</i>
<i>5 Yhteenveto</i>	<i>9</i>
<i>5.1 Lopputulokset ja analysointi</i>	<i>9</i>
<i>5.2 Oppimisen reflektointi</i>	<i>10</i>
<i>Lähdeluettelo</i>	<i>11</i>

Tiivistelmä

Tämän seminaarityön tavoitteena on opetella käytännössä, miten kontitetaan sovelluksia, ja miten niitä voidaan käyttää kontinhallinta-alustoilla, kuten Openshiftissä. Tässä työssä on käytetty dockeria, sekä CSC:n Rahti 1-palvelua (Rahti-palvelu, s.a.) Openshiftin hyödyntämiseen. Paikallista kehitystä on toteutettu Microsoft Subsystem for Linux:lla. Kohdesovellus, johon konttitekniologiaa on hyödynnetty, on Java-pohjainen ja se käyttää MySQL-tietokantaa. Sovellukseen tehtäviin muutoksiin tuotantoon vientiä varten on käytetty Eclipseä ja Visual Studio Codea. Lisäksi projektissa on hyödynnetty CSC:n dokumentaatiota eri palveluista ja käytännöistä (CSC, s.a.).

Projekti on edennyt vaiheittain paikallisesta kehityksestä tuotantoympäristöön asti. Aluksi konfiguroitiin sovellus konttiympäristöön sopivaksi. Sen jälkeen luotiin ensin paikallisesti sovelluskuvat, joiden perusteella docker-kontit rakennetaan. Tämän jälkeen tietokantakontti ja sovelluskontti yhdistetään samaan docker-verkkoon, jonka jälkeen niitä voidaan käynnissä ollessaan testata verkkoselaimella ja komentoriviltä. Kun paikallinen kokoonpano on saatu toimimaan, voidaan konfiguraatiot viedä CSC Rahtiin, ja sovellusta voidaan alkaa valmistelemaan julkaisua varten. Tällä hetkellä sovellus on käynnissä osoitteessa: <https://departures-container-ohke-teknologiat-matias.rahtiapp.fi/login> . Lopputuotoksena on kaksi konttia, joita hallinnoidaan ja ylläpidetään Openshiftillä.

1 Johdanto/Ylätason esittely

Tässä projektissa tullaan rakentamaan dockerilla konttiarkkitehtuuri, jossa suoritetaan yksinkertaista java-ohjelmaa. Pääteemana on Konttiarkkitehtuurin luonti ja hallinta Openshiftissä, sekä mitä eri vaihtoehtoja ja ratkaisuja projektissa on mahdollista hyödyntää. Lopuksi käsitellään myös eri ratkaisujen järkevyyttä pienikokoisessa ja yksinkertaisessa arkkitehtuurissa.

Tämän projektin oletuksena on, että lukija omaa riittävän tuntemuksen docker enginestä, sekä tuntee virtualisoinnin ja dockerin peruskäsitteitä ja toimintoja. Lisäinformaatiota voi tarvittaessa etsiä dockerin dokumentaatiosta (Docker, s.a.).

Projekti etenee seuraavissa vaiheissa:

- A. Sovelluksen valmistelu ja dockerfilen luonti.
- B. Tietokantakontin luonti.
- C. Konttien testaaminen paikallisessa ympäristössä.
- D. Sovelluskuvan vienti dockerhubiin.
- E. Openshiftin käyttöönotto
- F. Konttien luonti ja määritykset Openshiftissä
- G. Konttiarkkitehtuurin testaus.

2 Käytetyt tekniikat

2.1 Paikallisen kehityksen työkalut

- WSL 2
- Docker 25.0.1
- Eclipse 2023-03 (4.27.0)
- Visual Studio Code 1.89.1
- MySQL 8.4.0

2.2 Verkkokehityksen työkalut

- Rahti 1 / Openshift 3.11
- Dockerhub

3 Paikallinen kehitys

3.1 Sovelluksen valmistelu

Java sovellukselle on aivan aluksi määritettävä tietokantayhteys parametrillisesti vastamaan konttia, jossa tietokanta on käynnissä. Käytetään tietokantakontista nimeä mysql-container.

```
spring.datasource.url=jdbc:mysql://mysql-container:3306/matias
spring.datasource.username=matias
spring.datasource.password=password
spring.jpa.hibernate.ddl-auto=update
spring.data.rest.basePath=/api
spring.datasource.initialization-mode=always
spring.jpa.show-sql=true
server.port=8070
```

Jotta sovellus voi löytää tietokannan virtuaaliympäristössä, täytyy sille antaa tietokantakontin nimi osoitteeseen. Nyt sovellus yrittää yhdistää tietokannan porttiin 3306. Muut kirjautumistiedot on pidettävä myös tallessa, sillä niitä tarvitaan tietokantakontin määrittämisessä.

Tämän jälkeen sovellus voidaan paketoida yhdeksi jar:ksi. Näin sitä on helppo käyttää ja siirrellä. Kun sovellus on valmis, luodaan dockerfile, jonka perusteella sovelluskontti rakennetaan.

```
FROM openjdk:17-jdk-slim
```

```
ADD Departures-0.0.1-SNAPSHOT.jar app.jar
```

```
ENV SPRING_DATASOURCE_DRIVER_CLASS_NAME=com.mysql.cj.jdbc.Driver
ENV SPRING_DATASOURCE_URL=jdbc:mysql://mysql-container:3306/matias
ENV SPRING_DATASOURCE_USERNAME=matias
ENV SPRING_DATASOURCE_PASSWORD=password
ENV SPRING_JPA_HIBERNATE_DDL_AUTO=update
ENV SPRING_DATA_REST_BASEPATH=/api
ENV SPRING_DATASOURCE_INITIALIZATION_MODE=always
ENV SPRING_JPA_SHOW_SQL=true
ENV SERVER_PORT=8070
```

```
ENTRYPOINT ["java", "-jar", "/app.jar"]
```

Dockerfilessä on vielä erikseen määritelty sovelluksen parametrit ympäristömuuttujina. Tämä on hyödyllistä silloin, kun esimerkiksi testataan konttia, sillä ei ole tarvetta muuttaa itse sovelluksen parametrejä, koska ympäristömuuttujien arvot ylikirjoittavat sovelluksen sisällä määritetyt arvot. Päivitetystä dockerfilestä voidaan suoraan rakentaa uusi kontti.

3.2 Tietokannan valmistelu

Tietokantakontin luominen voidaan suorittaa kontin käynnistyksen yhteydessä, ja sille voidaan samalla määrittää myös kaikki parametrit.

```
root@MSI:/mnt/c/Windows/system32# docker run -d --name mysql-container network=app-network -p 3306:3306 -e MYSQL_ROOT_PASSWORD=password -e MYSQL_DATABASE=matias -e MYSQL_USER=matias -e MYSQL_PASSWORD=password mysql:latest
```

Käytetään MySQL:n viimeisintä versiota, ja määritellään tietokannalle parametrit siten, että ne vastaavat sovelluksen yhteysparametrejä. Kontti on nyt käynnissä.

```
263c4401e7ae  mysql:latest  "docker-entrypoint.s..."  2 days ago  Up 5 minutes
```

Huomioitavaa on, että tietokantakontti on käynnistettävä ennen sovelluskonttia, muuten sovelluskontti kaatuu, sillä sovellus ei pysty käsittelemään virhetilanteita yhteyksiin liittyen.

3.3 Konttien yhteistoiminta

Jotta kontit voivat "löytää" toisensa, tulee niille ensiksi määritellä vielä yhteinen docker-verkko. Tässä verkossa molemmilla konteilla on omat ip-osoitteet, ja niitä voi turvallisesti testata ilman vaikutusta verkon ulkopuolisiin komponentteihin. Nimetään docker-verkko "app-network" nimiseksi.

Verkkomäärittelyn ja tietokantakontin luomisen jälkeen voidaan sovelluskontti käynnistää.

```
80ab92cabfdf  departures2  "java -jar /app.jar"  2 days ago  Up 4 minutes  
/tcp          departures-container
```

Sovelluksen lokeista voidaan todeta, että tietokantayhteys todella toimii. Lisäksi sovelluksen toiminnan voi tässä vaiheessa testata selaimella osoitteessa "localhost:8070".

```
hibernate: select f1_0.id,f1_0.arrival,f1_0.company,f1_0.departure,f1_0.estdep  
hibernate: select t1_0.typeid,t1_0.name from type t1_0 where t1_0.typeid=?  
hibernate: select t1_0.typeid,t1_0.name from type t1_0 where t1_0.typeid=?  
024-05-10T15:50:40.200Z INFO 1 [main] app DeparturesApplicati
```

localhost:8070/flightlist

[Maps](#)
[Opiskelijalle | Haaga...](#)
[Etusivu | Haaga-Helia](#)
[Atkins ry | Haaga-H...](#)

Flights

Welcome **Admin!**

[Sign Out](#)

Aikataulussa
Myöhässä

Number	Company	Estimated Departure	Departure	Type	Remove Flight Data	Edit Flight Data
AY805CD	Finnair	10.05.2024	10.05.2024	Commercial	Delete	Edit
AY805CLNG	Finnair	11.05.2024	12.05.2024	Commercial	Delete	Edit
AY805CLNG	Test_	15.03.1974	21.06.2046	Commercial	Delete	Edit
999858	Suomen ilmavoimat	05.05.2024	05.05.2024	Military	Delete	Edit

[Add New Flight Data](#)

4 Openshift

4.1 Docker image

Jotta voidaan rakentaa kontti sovelluksesta Openshiftissä, täytyy sovelluskuva julkaista dockerhubissa (Docker Hub, s.a.).

dockerhub

[Explore](#)
[Repositories](#)
[Organizations](#)

ctrl+K

[vehiclepickkb](#) / [Repositories](#) / [volvos60b5234](#) / [General](#)
Using 0 of 1 private repositories. [Get more](#)

[General](#)
[Tags](#)
[Builds](#)
[Collaborators](#)
[Webhooks](#)
[Settings](#)

vehiclepickkb/volvos60b5234

Updated 2 days ago

Rahti project image

This repository does not have a category

Docker commands

To push a new tag to this repository:

docker push vehiclepickkb/volvos60b5234:tagname

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
0.0.1		Image	---	2 days ago

[See all](#)

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

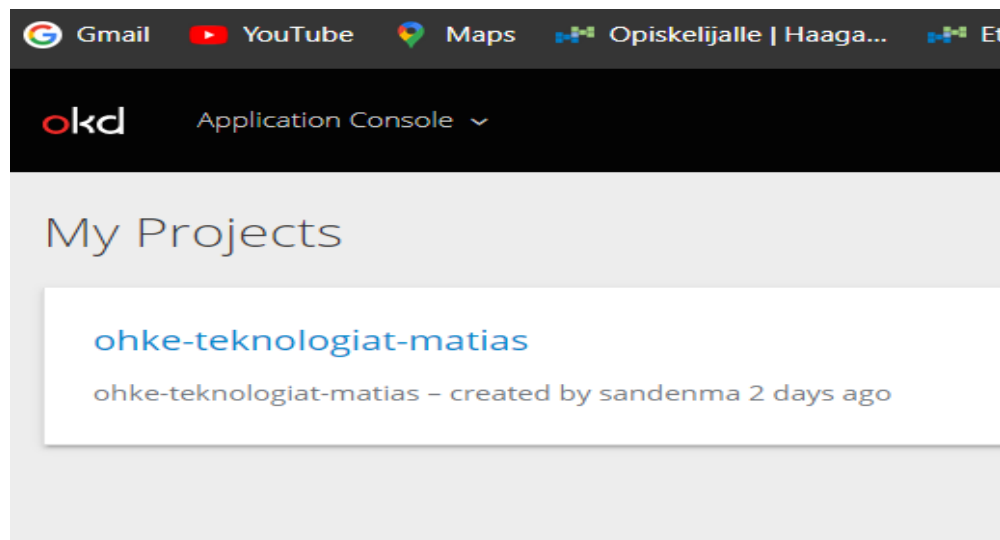
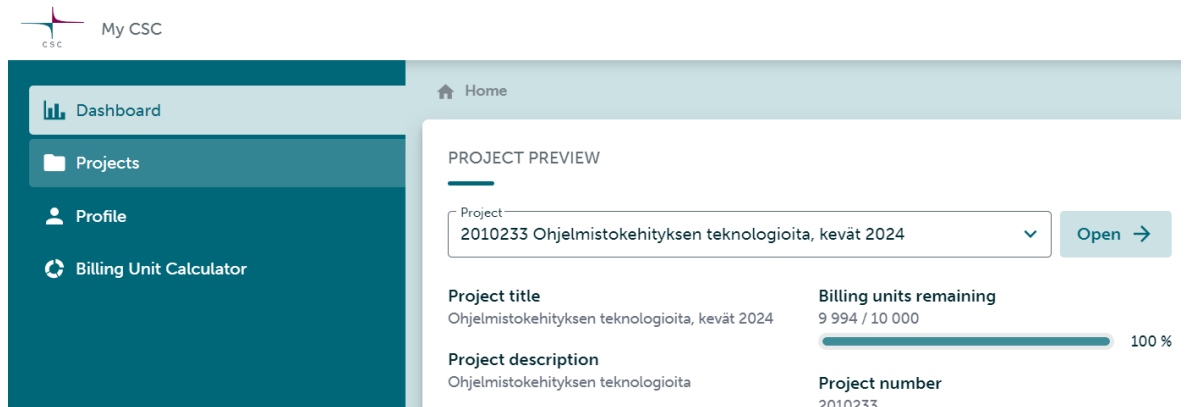
Available with Pro, Team and Business subscriptions. [Read more about automated builds](#)

[Upgrade](#)

Dockerhubiin voidaan kirjautua komentoriviltä, ja samoin voidaan myös image julkaista suoraan komentoriviltä käyttäen docker tag ja docker push-komentoja.

4.2 Openshift konfiguraatio

Openshiftiä käyttääkseen on kirjauduttava ensin CSC:n pääpalveluun, ja sen jälkeen vielä CSC:n rahti palveluun. CSC-pääpalvelussa on oltava projekti. Tämän jälkeen Rahtiin luodaan oma projekti tulevia kontteja varten. Tämä projekti toimii käytännössä tietynlaisena docker-verkkona tai ympäristönä, ja kaikki siihen liitetyt kontin konfiguroidaan perustasolla toimimaanyhdessä automaattisesti.



4.3 Tietokantakontti Openshiftissä

Rahdissa tietokantakontin luonti on pitkälti automatisoitu ja tehty varsin käyttäjäystävälliseksi. Noudatetaan sovelluksen kannalta edullista etenemisjärjestystä ja rakennetaan tietokanta jälleen ensin.

MySQL

Information Configuration Binding Results

1 2 3 4

* Memory Limit
512Mi
Maximum amount of memory the container can use.

Namespace
openshift
The OpenShift Namespace where the ImageStream resides.

* Database Service Name
mysql
The name of the OpenShift Service exposed for the database.

MySQL Connection Username
Username for MySQL user that will be used for accessing the database.

MySQL Connection Password
Password for the MySQL user that will be used for accessing the database.

Cancel < Back Next >

Annetaan tietokannalle samat arvot, kuin mitä aikaisemmin paikallisessa määrittelyssä. Tämän jälkeen tietokantakontti käynnistyy ja sen toimintaa voi tarkastella esimerkiksi selaimessa integroidulla komentorivillä.

```
sh-4.2$ mysql -u matias -p matias
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6854
Server version: 5.7.21 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

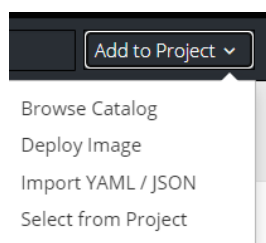
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

4.4 Sovellus Openshiftissä

Sovelluskontti luodaan Openshiftiin importoimalla image dockerhubista, ja luomalla kontti sen pohjalta. Rahti 1 tarjoaa myös oman image registry palvelun (Rahti-palvelu, s.a.), josta sovelluskuva on suhteellisen helppo tuoda omaan projektiin. Kun sovelluskuva on viety registryyn, voidaan kontti luoda projektin konsolista valitsemalla "Deploy image".



Tämän jälkeen kontin rakennus alkaa automaattisesti. Automaatiikka ei kuitenkaan osaa rakentaa konttia täysin valmiiksi ulkoista käyttöä varten, joten siihen on tehtävä muutamia muutoksia, jotta sovellusta voidaan käyttää esimerkiksi selaimessa.

```
spec:
  containers:
    - image: >-
      docker-registry.default.svc:5000
      imagePullPolicy: Always
      name: departures-container
      ports:
        - containerPort: 8070
          name: tomcat
          protocol: TCP
  spec:
    clusterIP: 172.30.206.85
    ports:
      - name: tomcat
        port: 8070
        protocol: TCP
        targetPort: 8070
```

<https://departures-container-ohke-teknologiat-matias.rahtiapp.fi>

✓ Exposed on router 'router' 10 hours ago

Details

Path: *none*
Service: [departures-container](#)
Target Port: tomcat

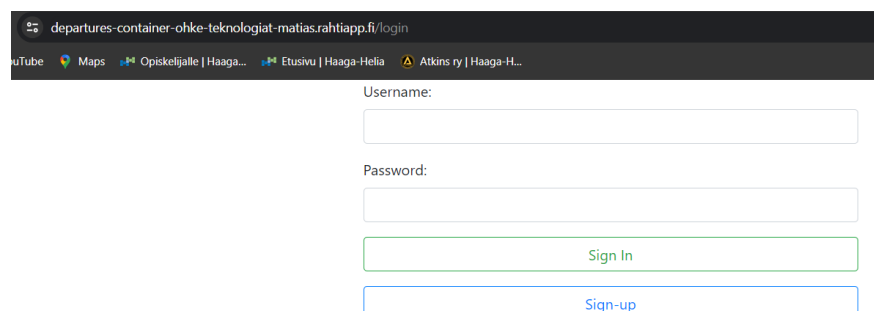
This target port will route to Service Port 8070 → Container Port 8070 (TCP).

Tarvitaan konfigurointi kontin yaml-tiedostoon, jossa määritellään vielä varmuuden vuoksi containerPort, mikä tarkoittaa kontin sisäistä porttia, johon tulee liikennettä sisältä ja ulkoa.

Tarvitaan myös konfigurointi kontin palveluun (service), jossa määritellään portti ja sille nimi. Portin nimi on erityisen tärkeä, sillä ilman sitä liikennettä ei voida ohjata tunnistettuun porttiin, ja selainnäkö ei toimi.

Lopuksi luodaan palvelulle reitti, josta pääsee sovellukseen käsiksi. Reitti ohjaa liikenteen porttien kautta siten, että kanava loppukäyttäjälle toimii.

Lopuksi voidaan todeta, että kontit toimivat pilvessä, kun navigoidaan reitistä saatuun osoitteeseen.



departures-container-ohke-teknologiat-matias.rahtiapp.fi/login

Username:

Password:

[Sign In](#)

[Sign-up](#)

4.5 Kokonaisuuden hallinta Openshiftissä

Kun konttiarkkitehtuuri on onnistuneesti rakennettu, voidaan nyt tehdä erilaisia toimenpiteitä ja konfiguraatioita konteille, esimerkiksi porttimuutoksia, suojausmuutoksia tai päivityksiä. Openshiftin hyvä puoli on se, että suuri osa konfiguraatioista on automaattisesti toimivia, tai vähintäänkin helposti integroitavia. Kontteja voi myös monistaa ja sammuttaa, tai arkkitehtuuriin voi lisätä helposti muita laajennuksia. Rahti 1:ssä on myös muita ominaisuuksia, jotka tekevät konttien hallinnasta helpompaa, esimerkiksi valmiit konttipohjat perinteisille sovelluksille (CSC, s.a.).

5 Yhteenveto

5.1 Lopputulokset ja analysointi

Näin konttiarkkitehtuuri on nyt onnistuneesti pilvessä. Se koostuu kahdesta kontista, sovelluksesta ja tietokannasta. Sovellusta voi käyttää selaimella kuka tahansa.

Huomioitavaa on, että Rahti 1 palvelu on jo vahentumassa, ja sen käyttö loppuu 2024 ke- säkuussa. Rahti 1:stä on mahdollista migratoida dataa Rahti 2 palveluun, mutta konfiguraatiot saattavat vaatia pieniä muutoksia siirron jälkeen (CSC, s.a.).

Rahti 1:n käyttöliittymä on jokseenkin epäjohdonmukainen ja siellä käytettävä termistö ei vastaa täysin perinteistä docker-termistöä esimerkiksi komponenttien osalta, mikä saattaa aiheuttaa aluksi hieman ihmettelyä, kun ei tarkalleen tiedä, mistä aloittaa. Dokumentaatio on kyllä olemassa, mutta sen läpi kahlaaminen vain yksittäisen pienen asian takia saattaa olla turhauttavaa. Tämä voi osaltaan johtua siitä, että Rahti 1:n ei koskaan ollut tarkoitus siirtyä tuotantokäyttöön. (CSC, s.a.).

Rahti 2 tarjoaa uudemmat versiot Openshiftistä ja Kubernetesesta, kuin Rahti 1. Suurta merkitystä ominaisuuksissa ei kuitenkaan tämän projektin osalta ollut, ja Rahti 1:n oma image registry teki siitä luonnollisemman valinnan selkeytensä vuoksi. Lisäksi tällä projektilla ei toistaiseksi ole suunniteltua jatkoa kovinkaan pitkälle, joten migraatio ja ylläpito Rahti 1:n poistuessa käytöstä ei ole tarpeellista.

Konttiarkkitehtuuria rakennettaessa olisi mahdollista käyttää docker composea, mikä voisi olla hyvä vaihtoehto suurehkoissa projekteissa, tai jos kontteja olisi useampia ja ne vaatisivat verkolta lisäasetuksia. Tässä projektissa asetukset ovat kuitenkin niin yksinkertaisia, että docker composen käytöstä ei olisi saatu varsinaista hyötyä irti. Se olisi ollut vain toinen lähestymistapa. Openshift-ympäristö taas korvaa tarpeen docker composelle, sillä siellä konfigurointi yleisesti tapahtuu DeploymentConfig ja service ominaisuuksia käyttäen, tai vaihtoehtoisesti Kubernetesin manifest-tiedostoilla.

5.2 Oppimisen reflektointi

Tässä projektissa olen oppinut ja kerrannut hyödyllisiä dockerin peruskäytänteitä, kuten monia komentorivisyötteitä ja konttien toimintaa. Lisäksi uusia palveluita olivat dockerhub ja CSC Rahti, sekä Openshift palveluna yleisesti. Projekti on antanut kokonaisvaltaista ymmärrystä siitä, miten polku paikallisesta virtuaalikoneesta johtaa lopulta pilvialustalle, jossa valmis sovellus toimii. Openshiftiin ja CSC:n palveluihin olisi voinut perehtyä vielä syvemmin. Tässä projektissa oli mahdollisuus tutustua tekniikoihin vain sen sallimassa viitekehityksessä. Mahdollisuuksien mukaan kehitetylle palvelulle voisi nyt esimerkiksi rakentaa monimutkaisemman ja laajemman tietokannan, tai siihen voisi kehittää esimerkiksi frontend-käyttöliittymän.

Lähdeluettelo

CSC. (s.a.). CSC:n verkkosivusto. Luettavissa: <https://docs.csc.fi/>. Luettu: 10.05.2024.

Docker. (s.a.). Docker Documentation. Luettavissa: <https://docs.docker.com>. Luettu: 10.05.2024

Docker Hub. (s.a.). Docker Hub. Luettavissa: <https://hub.docker.com>. Luettu: 10.05.2024

Rahti-palvelu. (s.a.). Rahti. Luettavissa: <https://rahti.csc.fi>. Luettu: 10.05.2024

CSC. (s.a.). Rahti - What is Rahti?. Luettavissa: <https://docs.csc.fi/cloud/rahti/rahti-what-is/>. Luettu: 10.05.2024

CSC. (s.a.). Rahti - What is Rahti?. Luettavissa: <https://docs.csc.fi/cloud/rahti2/rahti-what-is>. Luettu: 10.5.2024

CSC. (s.a.). Rahti Migration Guide. Luettavissa: <https://docs.csc.fi/cloud/rahti/rahti-migration>. Luettu: 10.05.2024