

Kodavi-sovellusprojekti

Lauri Antila
Outi Hilola
Antti Kauppi
Nuutti Rantanen
Anne Vaarala

Palvelimen asennusohje

Versio 1.0.0
Julkinen
2.12.2020

Jyväskylän yliopisto
Informaatioteknologian tiedekunta

Tietoja dokumentista

Tekijä: Outi Hilola `outi.m.hilola@student.jyu.fi`

Dokumentin nimi: Kodavi-sovellusprojekti, Palvelimen asennusohje

Sivumäärä: 16

Tiedoston nimi: `kodavi_palvelimen_asennusohje_1_0_0.docx`

Tiivistelmä: Kodavi-sovellusprojektissa kehitettiin vuonna 2020 Jyväskylän yliopiston liikuntatieteelliselle tiedekunnalle Study Health with HBSC -niminen WWW-sovellus, jonka avulla käyttäjät voivat hakea WHO-Koululaistutkimuksen tuloksia ja tarkastella niitä visuaalisesti. Dokumentissa kuvataan tietojärjestelmän osien asentaminen palvelimelle vaiheittain ja annetaan ohjeet niiden päivittämiseen.

Avainsanat: asennusohje, backend, frontend, palvelin, sovellus, sovellusprojekti, tietokanta.

Sisällysluettelo

1	Johdanto	1
2	Palvelimen ohjelmistot ja varmuuskopiointi.....	2
3	Palvelimelle kirjautuminen	3
4	Tietokannan asennus	4
5	Frontendin asennus	8
6	Backendin asennus	10
7	Tietokannan, frontendin ja backendin päivittäminen.....	16

1 Johdanto

Dokumentissa kuvataan vaiheittain Kodavi-sovellusprojektissa kehitetyn WWW-sovelluksen ensimmäinen asennuskerta sitä varten pystytetylle virtuaalipalvelimelle. Tietojärjestelmä koostuu käyttöliittymästä eli frontendistä, palvelinohjelmasta eli backendistä ja tietokannasta, joista kukin on asennettava erikseen palvelimelle. Lisäksi on tehtävä muutoksia palvelimelle asennettujen ohjelmistojen asetuksiin.

Kodavi-sovellusprojektissa palvelimelle asennettiin ensimmäisenä tietokanta, seuraavaksi frontend ja viimeisenä backend. Dokumentin ohjeet on laadittu samassa järjestyksessä ja kunkin tietojärjestelmän osan asennusohjeet ovat omassa luvussaan. Ohjeet on numeroitu kaksiosaisella numeroinnilla, jonka ensimmäinen osa on luvun numero ja jälkimmäinen on juokseva numero. Kunkin luvun kaikki ohjeet suositellaan suoritettavan numerojärjestyksessä ja yhdellä kertaa ongelmien välttämiseksi.

Asennukseen tarvitaan seuraavat ohjelmistot tai muut vastaavat ohjelmistot:

- Cisco AnyConnect -VPN-yhteysohjelma (otetaan yhteys Jyväskylän yliopiston verkkoon),
- IDE-kehitysympäristö (esim. IntelliJ IDEA) backendin paketointiin,
- koodieditori (esim. Visual Studio Code) frontendin kääntämistä varten,
- Pageant-autentikointiohjelma (tulee PuTTYn ja WinSCP:n mukana),
- PuTTY tai KiTTY -suorakäyttöohjelma sekä
- WinSCP-tiedonsiirto-ohjelma.

Dokumentin ohjeista on turvallisuussyistä piilotettu tarkkoja käyttäjätunnuksia, salasanoja ja tiedostojen nimiä ja korvattu ne kulmasulkeiden sisään kirjatulla selitteellä (esim. <käyttäjänimi>). Piilotetut tiedot on lähetetty Kodavi-sovellusprojektin projektiorganisaation sähköpostilistalle sen jälkeen, kun sähköpostilistan arkisto on kopioitu projekti-CD:lle. Projektiorganisaatioon kuuluneet henkilöt pääsevät käsiksi sähköpostiarkiston tietoihin kirjautumalla Jyväskylän yliopiston Korppi-järjestelmään.

Dokumentin luvussa 2 kerrotaan palvelimen ominaisuuksista ja varmuuskopioinnista. Luvussa 3 kerrotaan palvelimelle kirjautumisesta. Tietokannan asennuksen vaiheet kuvataan luvussa 4, frontendin asennuksen vaiheet luvussa 5 ja backendin asennuksen vaiheet luvussa 6. Luvussa 7 kerrotaan, kuinka tietojärjestelmän osia päivitetään.

2 Palvelimen ohjelmistot ja varmuuskopiointi

Jyväskylän yliopiston Digipalvelut pystyttivät tuotantopalvelimena käytetyn virtuaalipalvelimen. Palvelimen käyttöjärjestelmä on Red Hat Enterprise Linux 8 (minor-versio 8.2) ja sen nimi on `studyhealth.sport.jyu.fi`. Palvelimessa on neliydinsuoritin ja kahdeksan gigabittia keskusmuistia. Digipalvelut asensivat Kodavi-projektiryhmän pyynnöstä palvelimelle valmiiksi seuraavat ohjelmistot:

- Apache Tomcat 9,
- Java JRE 11 (Open JDK),
- Java 1.8.0 (pakettiriippuvuudet vaativat),
- Nginx (SSL) sekä
- PostgreSQL 12 -tietokannanhallintajärjestelmä.

Lisäksi projektiryhmä pyysi Digipalveluita lisäämään käyttäjille juurikäyttäjiä vastaavat oikeudet tai mahdollisuuden muokata asetuksia ja luoda esimerkiksi sovelluskäyttäjä Kodavi-ohjelmalle oikeuksien rajoittamiseksi.

Palvelin varmuuskopioidaan joka yö sekä virtuaalikonetasolla että tiedostotasolla. Lisäksi PostgreSQL:n tietokannoista ajetaan SQL-dumpit talteen joka ilta hakemistoon `/mnt/backup/postgresql`, jossa niitä säilytetään viikon ajan. Ne varmuuskopioidaan sieltä vielä erikseen nauhalle, josta voidaan tarvittaessa palauttaa vanhempiakin versioita.

Palvelimen komentohistoria tallentuu sijaintiin `/root/.history_files/`. Historiatietoja pääsee lukemaan komennolla `collect_history.sh`, joka näyttää oletuksena kuluvan kuukauden komentohistorian kaikilta tunnuksilta. Komennolle voi antaa lokitiedostoja argumentteina, jolloin pääsee lukemaan esimerkiksi tietyn käyttäjän komentohistoriaa tai edellisten kuukausien komentohistoriaa. Esimerkiksi käyttäjän `<käyttäjätunnus>` komentohistorian vuoden 2020 ajalta pääsee näkemään seuraavalla komennolla:

```
collect_history.sh /root/.history_files/*<käyttäjätunnus>*2020*
```

Kaikista Emacsilla, vi:llä tai nanolla editoiduista tiedostoista tehdään automaattisesti varmuuskopio hakemistoon `/root/emacsbackups/`.

3 Palvelimelle kirjautuminen

Palvelimelle kirjaudutaan Jyväskylän yliopiston Digipalvelujen erikseen luomalla käyttäjätunnuksella. Käyttäjän on pyydettävä itselleen tunnus ottamalla yhteyttä palvelimen hallinnolliseen yhteyshenkilöön, Jyväskylän yliopiston liikuntatieteellisessä tiedekunnassa työskentelevään tutkijatohtori Nelli Lyyraan ja Digipalveluihin. Digipalvelut luovat tunnuksen, mikäli Lyyra on vahvistanut, että kyseisellä henkilöllä on lupa kirjautua palvelimelle.

Ennen tunnuksen luomista käyttäjän on luotava itselleen SSH-avainpari, tallennettava julkinen ja salainen avain tiedostoihin sekä pistettävä keksimänsä passphrase hyvään talteen. SSH-avainparin julkinen avain toimitetaan Digipalveluille. Digipalvelut luovat käyttäjälle käyttäjätunnuksen palvelimelle ja yhdistävät toimitetun SSH-avaimen käyttäjätunnukseen. Tämän jälkeen käyttäjä pääsee kirjautumaan palvelimelle käyttäjätunnuksen, salaisen SSH-avaimen ja passphrasen yhdistelmällä. Avainparin luontiin voi käyttää esimerkiksi PuTTYn mukana tulevaa PuTTYgen-ohjelmaa.

Palvelimelle kirjaudutaan suorittamalla vaiheet 3.1–3.3. Etsi tarvittaessa lisäohjeita VPN-yhteyden avaamiseen ja PuTTY-ohjelman käyttöön Jyväskylän yliopiston verkkosivuilta jyu.fi.

3.1 Avaa VPN-yhteys yliopiston verkkoon.

3.2 Lisää luomasi SSH-avain Pageantiin seuraavasti:

- Avaa Pageant tietokoneellasi. (Huomaa, että ohjelma käynnistyy taustalle.)
- Klikkaa Pageant-kuvaketta hiiren oikealla painikkeella ja valitse *Add key*.
- Valitse tiedosto, joka sisältää salaisen SSH-avaimen ja syötä passphrase Pageantin pyytäessä.

3.3 Kirjautu PuTTY-ohjelmalla palvelimelle seuraavasti:

- Lisää *SSH Auth* -välilehdelle tiedosto, joka sisältää salaisen SSH-avaimen.
- Lisää *Session*-välilehdelle *Host Name* -tekstikenttään `<käyttäjätunnus>@studyhealth.sport.jyu.fi` ja valitse *open*.

Kirjautuminen on onnistunut, jos terminaaliin tulee näkyviin seuraava rivi:

```
[<käyttäjätunnus>@studyhealth ~]$
```

4 Tietokannan asennus

Tietokannan luomista varten on järkevää **luoda erillinen käyttäjärooli**, jolla on postgres-käyttäjää rajatut oikeudet. Käyttäjä luodaan suorittamalla ohjeet 4.1–4.6.

- 4.1 Mene palvelimen pääkäyttäjäksi (root) seuraavalla komennolla:

```
sudo su -
```

- 4.2 Ota yhteys PostgreSQL-palvelimeen postgres-käyttäjänä seuraavalla komennolla:

```
psql -U postgres
```

- 4.3 Luo tietokantapalvelimelle erillinen käyttäjärooli seuraavalla komennolla:

```
CREATE ROLE <käyttäjänimi> WITH NOSUPERUSER CREATEDB LOGIN;
```

Kyseisessä komennossa NOSUPERUSER tarkoittaa, että roolille ei anneta pääkäyttäjäoikeuksia. LOGIN mahdollistaa roolin kirjautumisen ja palvelimelle yhdistämisen itsenäisenä käyttäjänä. CREATEDB sallii roolin luoda uusia tietokantoja. Se on tässä kontekstissa hyödyllinen, muttei täysin välttämätön ominaisuus.

- 4.4 Lisää luodulle käyttäjälle salasana seuraavalla komennolla:

```
ALTER ROLE <käyttäjänimi> WITH PASSWORD '<salasana>';
```

- 4.5 Luo tietokanta seuraavalla komennolla:

```
CREATE DATABASE <tietokannan nimi> WITH OWNER = <käyttäjänimi>;
```

- 4.6 Lopeta yhteys PostgreSQL-palvelimeen seuraavalla komennolla:

```
\q
```


Tietokantaan kirjautumisen asetuksia muutettiin ensimmäisen asennuksen yhteydessä ohjeiden 4.7–4.11 mukaisesti. Taulujen ja näkymien lisääminen myöhemmin ei vaadi kyseisten asetusten muuttamista uudelleen.

4.7 Siirry kansioon `/var/lib/pgsql/12/data` ja avaa haluamallasi tekstieditorilla tiedosto `pg_hba.conf`.

4.8 Lisää tiedostoon rivin `# Local & JYU LAN users can connect if username matches the database name` alle viimeiseksi seuraava rivi:

```
host <tietokannan nimi> <käyttäjänimi> 127.0.0.1/32 md5
```

Tehty lisäys sallii kirjautumisen tietokantaan `<tietokannan nimi>` käyttäjältä `<käyttäjänimi>` paikallisesta osoitteesta `127.0.0.1/32` salasanan ollessa oikea.

4.9 Tallenna tiedosto ja sulje tekstieditori.

4.10 Käynnistä PostgreSQL uudelleen seuraavalla komennolla:

```
systemctl restart postgresql-12.service
```

4.11 Poistu root-tilasta seuraavalla komennolla:

```
exit
```

Käyttäjäroolin luomisen jälkeen voidaan **tuoda tiedot tietokantaan**. Tietojen siirron vaiheet on kuvattu ohjeissa 4.12–4.24.

4.12 Avaa WinSCP-ohjelma ja kirjaudu palvelimelle palvelimen nimen `studyhealth.sport.jyu.fi` ja käyttäjätunnuksesi avulla. Salasanaa ei tarvita, jos olet käyttänyt Pageantia kirjautuessasi palvelimelle.

4.13 Kopioi `.pgsql`-päätteinen tietokantatiedosto WinSCP-ohjelmalla palvelimelle kotikansioosi `/adminhome/<käyttäjätunnus>/`.

4.14 Varmista, että olet merkittynä tiedoston omistajaksi eli käyttäjätunnuksesi lukee *owner*-sarakkeessa. Jos näin ei ole, muuta itsesi tiedoston omistajaksi *chown*-komennon avulla.

4.15 Siirry takaisin PuTTY-ikkunaan.

4.16 **Huomio!** Älä tee seuraavia asioita pääkäyttäjänä, vaan omalla käyttäjätunnuksellasi.

4.17 Siirry kotihakemistoon seuraavalla komennolla:

```
cd ~
```

4.18 Ensimmäisen palvelimen asennuskerran yhteydessä muokattiin tekstieditorilla tietokantatiedostoa korvaamalla *replace*-komennolla ilmaukset *OWNER TO postgres* ilmauksilla *OWNER TO <käyttäjänimi>* sekä siirrettiin kommentteihin tiedoston alussa olevat *SET*-lauseet.

4.19 Ota yhteys PostgreSQL-palvelimeen aiemmin luomasi roolin käyttäjänimellä ja salasanalla sekä putkita se lukemaan tietokantatiedoston tiedot seuraavalla komennolla:

```
PGPASSWORD='<salasana>' psql -h 127.0.0.1 -U <käyttäjänimi> -d  
<tietokannan nimi> -w <tietokantatiedosto>
```

Huomaa, että salasana saattaa jäädä komentotulkin historiaan. Se kannattaa poistaa sieltä myöhemmin, syöttää jollakin muulla tavalla (esimerkiksi *.pgpass*-tiedoston avulla) tai muuttaa tietokannasta jälkikäteen.

4.20 Ensimmäisellä palvelimen asennuskerralla saatiin komentoa suoritettaessa seuraava virheviesti ja korjausehdotus:

```
ERROR: permission denied to create extension "uuid-oss"
HINT: must be superuser to create this extension.
```

Ongelma korjattiin luomalla tarvittava laajennos pääkäyttäjänä.

4.21 Ota yhteys PostgreSQL-palvelimeen *postgres*-käyttäjänä seuraavalla komennolla:

```
psql -U postgres
```

4.22 Suorita käyttäjänä `postgres` seuraavat tietokantatiedoston sisältämät rivit, joiden suorittaminen epäonnistui aiemmin:

```
CREATE EXTENSION IF NOT EXISTS "uuid-oss" WITH SCHEMA public;
--
-- Name: EXTENSION "uuid-oss"; Type: COMMENT; Schema: -; Owner:
--
COMMENT ON EXTENSION "uuid-oss" IS 'generate universally unique
identifiers (UUIDs)';
```

4.23 Tietokantatiedoston sisältämä data on nyt tietokannassa. Voit halutessasi tarkistaa asennuksen onnistumisen tekemällä tietokantahaun esimerkiksi tauluun `respondent`.

4.24 Lopeta yhteys PostgreSQL-palvelimeen seuraavalla komennolla:

```
\q
```

Tietokannan asennus on valmis.

5 Frontendin asennus

Sovelluksen käyttöliittymä eli frontend asennetaan palvelimelle suorittamalla ohjeet 5.1–5.8.

- 5.1 Luo frontendistä `build`-kansio suorittamalla haluamasi koodieditorin (esim. Visual Studio Code) terminaalissa `frontend`-kansiossa seuraava komento:

```
npm run build
```

- 5.2 Avaa WinSCP-ohjelma. Jos et ole vielä kirjautuneena palvelimelle, kirjaudu palvelimen nimen `studyhealth.sport.jyu.fi` ja käyttäjätunnuksesi avulla. Salasanaa ei tarvita, jos olet käyttänyt Pageantia.

- 5.3 Kopioi koko `build`-kansio WinSCP-ohjelmalla palvelimelle omaan kotikansioosi `/adminhome/<käyttäjätunnus>/`.

- 5.4 Siirry takaisin PuTTY-ikkunaan.

- 5.5 Tarkista tiedostosta `/etc/nginx/nginx.conf` root-määritelmä, jonka mukaisesta kansiossa Nginx hakee oletuksena `html`-tiedoston. Luodun `build`-kansion sisältö halutaan kopioida kyseiseen kansioon. Käytä tiedoston lukemiseen esimerkiksi seuraavaa komentoa:

```
less /etc/nginx/nginx.conf
```

- 5.6 Mene palvelimen pääkäyttäjäksi käyttämällä seuraavaa komentoa:

```
sudo su -
```

- 5.7 Siirry omaan kotikansioosi kopioimaasi `build`-kansioon seuraavalla komennolla:

```
cd /adminhome/<käyttäjätunnus>/build
```

- 5.8 Kopioi rekursiivisesti koko hakemiston sisältö `nginx.conf`-tiedoston root-määritelmässä ilmoitettuun kansioon seuraavalla komennolla:

```
cp -r * <root-määritelmässä ilmoitettu kansio>
```

Esimerkiksi seuraavasti: `cp -r * /usr/share/nginx/html/`

Anna lupa kirjoittaa yli aiempi sisältö vastaamalla `y`.

Frontendin asennus on valmis. Jos asennus onnistui, sovellus on nähtävissä selaimella osoitteessa `studyhealth.sport.jyu.fi`.

6 Backendin asennus

Sovelluksen backend on paketoitava `war`-tiedostoksi ennen kuin se voidaan asentaa palvelimelle.

Backendin paketoinnin vaiheet on kuvattu ohjeissa 6.1–6.7.

- 6.1 Avaa backend käyttämässäsi kehitysympäristössä (esim. IntelliJ IDEA).
- 6.2 Täydennä asetusmuuttujat tiedostoon `application.yml` lisäämällä PostgreSQL-palvelimelle luodun käyttäjän tiedot. Tiedoston oikea syntaksi on seuraavanlainen:

```
app:
  datasource:
    jdbc-url: <jdbc-url>
    username: <käyttäjänimi>
    password: <salasana>
    pool-size: 30
```

Huomio! Asetusmuuttujia ei saa päästä julkiseksi esimerkiksi versiohallintaan. Oikeaoppisempi tapa saattaa tietokannan salasana backendin tietoon olisi lisätä `application properties` erikseen Tomcat-yhteyteen.

6.3 Lisää `pom.xml`-tiedostoon seuraavat rivit:

- Lisää kohdan `project` alle seuraava rivi:

```
<packaging>war</packaging>
```

- Lisää kohdan `dependencies` alle seuraavat rivit:

```
<dependency>
```

```
<groupId>org.springframework.boot</groupId>
```

```
<artifactId>spring-boot-starter-tomcat</artifactId>
```

```
<scope>provided</scope>
```

```
</dependency>
```

- Lisää kohdan `build` alle seuraava rivi:

```
<finalName>${project.artifactId}</finalName>
```

6.4 Täydennä `KodaviApplication`-tiedostossa `KodaviApplication`-luokan esittelyrivi seuraavaan muotoon:

```
public class KodaviApplication extends SpringBootServletInitializer
```

6.5 Lisää `KodaviApplication`-tiedoston alkuun seuraava import-rivi:

```
import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;
```

6.6 Suorita kehitysympäristön terminaalissa `backend`-kansiossa seuraava komento:

```
mvnw clean package
```

Käännetty tiedosto löytyy `target`-kansioista.

- 6.7 Ensimmäisellä asennuskerralla backendin paketointi epäonnistui aluksi. Virheilmoitusten avulla selvisi, että koodissa oli ilmoitettu väärä Java-versio. Terminaalissa tarkastettiin Javan versionumero seuraavalla komennolla:

```
java -version
```

Vastauksena saadun tiedon perusteella muutettiin tiedostossa `pom.xml` kohtaan `properties` Javan versionumeroksi 8. Tämän jälkeen paketointi onnistui.

Backendin paketoinnin jälkeen **backend asennetaan palvelimelle**. Asennuksen vaiheet on kuvattu ohjeissa 6.8–6.15.

- 6.8 Avaa WinSCP-ohjelma. Jos et ole vielä kirjautuneena palvelimelle, kirjaudu palvelimen nimen `studyhealth.sport.jyu.fi` ja käyttäjätunnuksesi avulla. Salasanaa ei tarvita, jos olet käyttänyt Pageantia.

- 6.9 Kopioi backendistä paketoitu `war`-tiedosto WinSCP-ohjelmalla palvelimelle omaan kotikansioosi `/adminhome/<käyttäjätunnus>/`.

- 6.10 Siirry takaisin PuTTY-ikkunaan.

- 6.11 Tarkista, että olet pääkäyttäjänä. Jos et ole, käytä seuraavaa komentoa:

```
sudo su -
```

- 6.12 Siirry omaan kotikansioosi seuraavalla komennolla:

```
cd /adminhome/<käyttäjätunnus>/
```

- 6.13 Kopioi `war`-tiedosto Tomcatin `webapps`-kansioon ja muuta tiedostonimeksi `ROOT.war` seuraavalla komennolla:

```
cp <war-tiedoston nimi> /var/lib/tomcat9/webapps/ROOT.war
```

Tomcat purkaa automaattisesti `war`-tiedoston kansioon `ROOT` ja käynnistää backendin.

6.14 Testaa backendin toimivuutta api-pyynnöllä, esimerkiksi seuraavalla komennolla:

```
curl 127.0.0.1:8080/api/variable/language/fi/theme/2
```

6.15 Ensimmäisellä kerralla curl-komennon vastauksena saatiin seuraava virheilmoitus:

```
error: Internal Server Error
```

```
message: Failed to get driver instance for jdbcUrl
```

Tomcat ei jostain syystä löytänyt PostgreSQL-ajuria, vaikka se löytyi kirjastoriippuvuuksista. Ongelma korjattiin siirtymällä kansioon `ROOT/WEB-INF/lib`, joka sisältää kaikki war-tiedostoon pakatut riippuvuudet ja kopioimalla sieltä tiedosto `postgresql-42.2.11.jar` Tomcatin library-kansioon seuraavalla komennolla:

```
cp postgresql-42.2.11.jar /usr/share/tomcat9/lib/
```

Tämän jälkeen Tomcat käynnistettiin uudelleen. Hetken kuluttua curl-komennon avulla tehdyt api-pyyntö alkoivat palauttaa vastauksia.

Nginxiä käytetään eräänlaisena proxynä, joka näkyy kaikille asiakkaille. Asiakkailta tulevat pyynnöt täytyy välittää edelleen tietokantaan. **Nginxin asetuksia täytyy muokata** ohjeiden 6.16–6.22 mukaisesti, jotta se toimii halutulla tavalla.

6.16 Siirry kansioon `/etc/nginx/` seuraavalla komennolla:

```
cd ~/etc/nginx/
```

6.17 Avaa tekstieditorissa tiedosto `nginx.conf` ja lisää SSL-yhteyksiä käsittelevään `server-`lohkoon seuraavat koodirivit ohjaamaan pyynnöt oikeaan osoitteeseen backendille:

```
# Proxy to API

location /theme/ {
    proxy_pass http://127.0.0.1:8080/api/theme/;
}

location /variable/ {
    proxy_pass http://127.0.0.1:8080/api/variable/;
}

location /response/ {
    proxy_pass http://127.0.0.1:8080/api/response/;
}

# Handle address rewriting

location /hakusivu {
    rewrite ^ /index.html break;
}
```

- 6.18 Lisää samaan SSL-yhteyksiä käsittelevään `server`-lohkoon `root`-määritelmän jälkeen seuraava rivi pakottamaan uudelleenohjaus paluukoodilla `301` suojattuun yhteyteen:

```
return 301 https://$host$request_uri;
```

- 6.19 Tallenna tiedosto ja sulje tekstieditori.

- 6.20 Käynnistä Nginx uudelleen seuraavalla komennolla:

```
systemctl restart nginx
```

- 6.21 Ensimmäisen asennuksen yhteydessä lisättiin SELinuxin boolean-arvo, joka sallii Nginxin yhdistää paikallisiin hosteihin palvelimen sisäisesti. Lisäys tehtiin seuraavalla komennolla:

```
setsebool -P httpd_can_network_connect 1
```

Digipalveluilta varmistettiin, että SELinuxin booleanin arvot säilyvät voimassa jatkossakin.

- 6.22 Voit nyt kirjautua ulos palvelimelta ja sulkea käyttämäsi ohjelmistot.

Backendin asennus on valmis.

7 Tietokannan, frontendin ja backendin päivittäminen

Tietokanta päivitetään ohjeiden 4.12–4.24 mukaisesti. Tietokantatiedosto kopioidaan käyttäjän omaan kotikansioon palvelimelle. Tämän jälkeen otetaan yhteys PostgreSQL-palvelimeen käyttämällä erikseen luodun käyttäjäroolin tunnuksia ja putkitetaan se lukemaan tietokantatiedoston data. Päivityksen onnistuminen kannattaa tarkistaa tekemällä tietokantahaku johonkin päivitetystä taulusta.

Frontend päivitetään ensimmäistä asennuskertaa vastaavalla tavalla suorittamalla ohjeet 5.1.–5.8. Frontendistä luodaan koodieditorin avulla `build`-kansio, joka kopioidaan käyttäjän omaan kotikansioon palvelimelle. Sieltä se kopioidaan pääkäyttäjänä rekursiivisesti tiedoston `/etc/nginx/nginx.conf` root-määritelmän mukaiseen kansioon ja korvataan samalla aiempi versio. Tämän jälkeen käyttöliittymän päivitetyn version pitäisi olla näkyvissä selaimella osoitteessa `studyhealth.sport.jyu.fi`.

Backendin päivittäminen aloitetaan paketoimalla projekti `war`-tiedostoksi. Tarkemmat ohjeet paketointiin on kuvattu ohjeissa 6.1–6.7. Paketoitu tiedosto kopioidaan käyttäjän omaan kotikansioon palvelimelle. Sieltä se kopioidaan pääkäyttäjänä Tomcatin `webapps`-kansioon ja muutetaan samalla tiedostonimeksi `ROOT.war`. Samalla korvataan aiempi `ROOT.war`-tiedosto. Tomcat purkaa kansion automaattisesti ja käynnistää backendin. Tarkemmat ohjeet backendin asennukseen löytyy ohjeista 6.8–6.15. Päivityksen onnistuminen kannattaa tarkistaa esimerkiksi seuraavalla api-pyyntöillä:

```
curl 127.0.0.1:8080/api/variable/language/fi/theme/2
```

Mikäli backendiin on tehty isoja muutoksia, voi olla tarpeen muuttaa Nginxin asetuksia. Lisätietoa asetusten muuttamiseen löytyy ohjeista 6.16–6.22.