# Deployment documentation

## Server info

Ubuntu 22.04 server on Digital Ocean

## Node installation

**Installing nvm ( https://github.com/nvm-sh/nvm)**

curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.2/install.sh | bash

source ~/.bashrc

**Checking installation**

command -v nvm

*output should be* nvm

**Checking which node versions are available**

nvm ls-remote

**Installing latest LTS version (pick from list generated by the command above)**

nvm install <version>

## Node process management (pm2) installation

**Installation**

npm install -g pm2

**Ensuring that pm2 starts up at boot**

pm2 startup systemd → *then follow the instructions of the output*

## Building the Spring Boot project

1. Installed Java: **sudo apt install -y openjdk-17-jdk**
2. Cloned the project from github into **/home/erik/Git/springProjekti**
3. Added the **secrets.application file**
4. Installed maven: **sudo apt install maven**
5. Tested that the project can run:  
    **./mvnw spring-boot:run**   
   **curl http://localhost:8080/api/products/**
6. Built the project: **mvn clean install**
7. Tested that the generated jar file works: **java -jar restdemo-0.0.1-SNAPSHOT.jar**
8. Copied jar file to **/home/ryhma4/www/jarri**
9. Started jar file with pm2: **pm2 start java -- -jar demo-0.0.1-SNAPSHOT.jar**
10. Tested that it is running: **curl http://localhost:8080/api/products/**

## Getting a domain from Godaddy

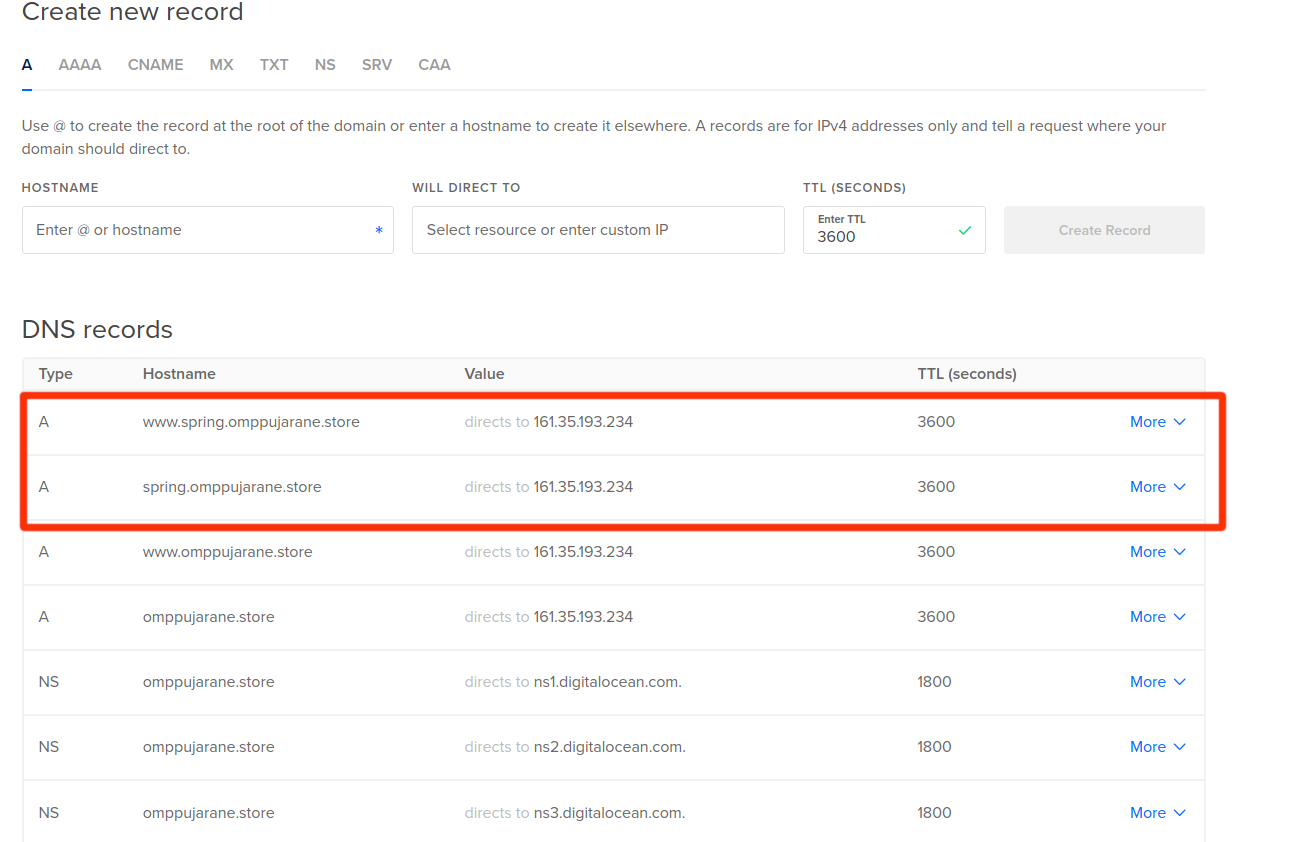
1. Bought the domain omppujarane.store
2. As the project is hosted on a DigitalOcean server the following changes are needed in the Godaddy settings:  
     
   Omat tuotteet → omppujarane.store → Hallitse → Hallitse verkkotunnusta → Hallitse DNS:ää → Nimipalvelimet → Vaihda → Anna omat nimipalvelimet →Lisää seuraavat:  
   **ns1.digitalocean.com**

**ns2.digitalocean.com**

**ns3.digitalocean.com**

## Digital Ocean settings

Add the following A-records in Digital Ocean



## nginx installation and configuration

*We install nginx, a web server. We also install Certbot in order to serve https pages.*

**Installation**

sudo apt install nginx certbot python3-certbot-nginx

**Firewall**

sudo ufw allow "Nginx Full"

## Editing the nginx configuration

**Editing /etc/nginx/nginx.conf**

See appendix

**Creating /etc/nginx/sites-available/koiravaatekauppa**

See appendix

**Checking to see if nginx configuration is ok**

sudo nginx -t

**Rebooting nginx**

sudo systemctl restart nginx

**Check that the site is online**

161.35.193.234 shows the website in the browser

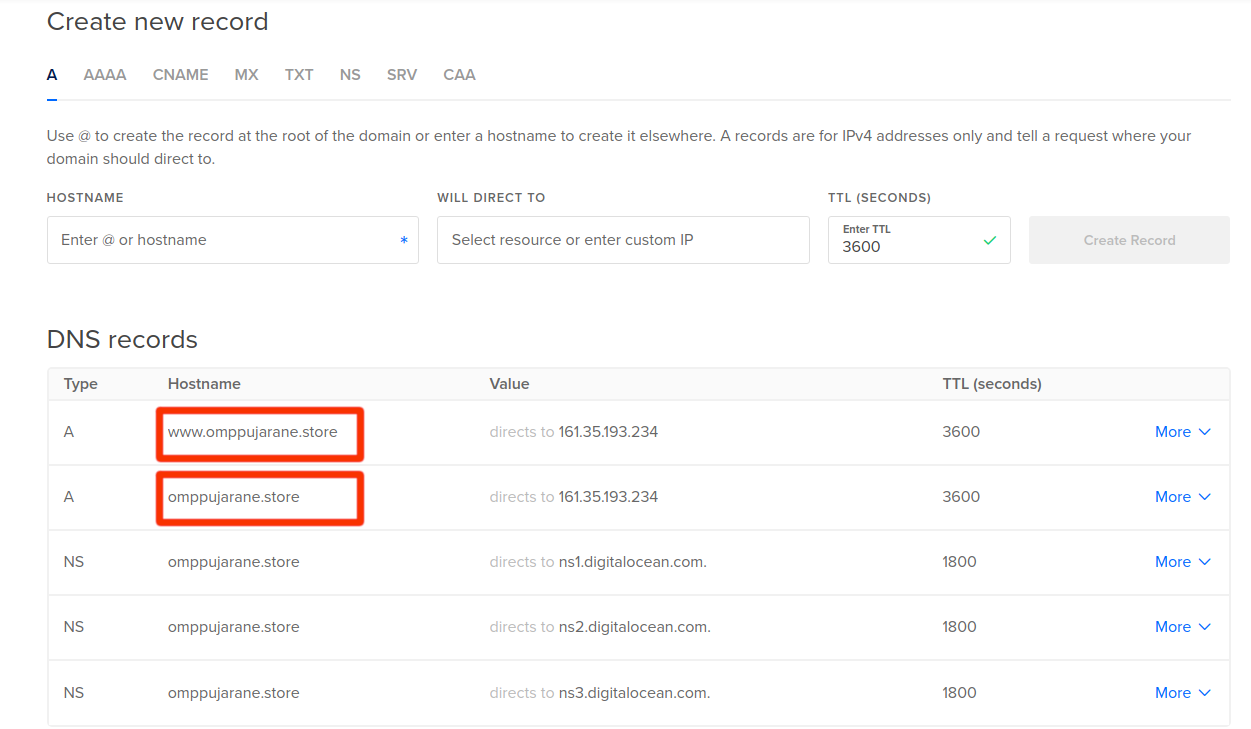
**Adding https**

sudo certbot --nginx -d spring.omppujarane.store

**Checking that the browser shows the site at spring.omppujarane.store**

# Deploying the React Frontend

**Added the following A-records to Digital Ocean**



**Cloned the frontend repository**

**Built project**

npm run build

**Installed server for React**

npm install -g serve

**Started server using pm2**

pm2 serve -s build --name frontend

*Note: The React server was on port 8080 and I wasn’t sure how to change that. To avoid collision, I changed the port of the Spring Boot project to 8081. I did this by adding* ***server.port=8081*** *to the application.properties file.*

**Creating /etc/nginx/sites-available/frontend**

See Appendix

**Https activated for React frontend using Certbot**

sudo certbot --nginx -d omppujarane.store

# Appendix

## The file /etc/nginx/nginx.conf file. The red parts are modified compared to the default:

*Note: the part about Auth0 and file uploads is probably not necessary, just a leftover from another project. Left it in there as we just in case we need it in sprint 4.*

user www-data;

worker\_processes auto;

pid /run/nginx.pid;

include /etc/nginx/modules-enabled/\*.conf;

events {

worker\_connections 768;

# multi\_accept on;

}

http {

##

# Basic Settings

##

#Added to make Auth0 callback possible

proxy\_buffers 8 16k;

proxy\_buffer\_size 32k;

#Added to aid large file uploads

proxy\_connect\_timeout 600;

proxy\_send\_timeout 600;

proxy\_read\_timeout 600;

client\_max\_body\_size 100M;

sendfile on;

tcp\_nopush on;

types\_hash\_max\_size 2048;

# server\_tokens off;

# server\_names\_hash\_bucket\_size 64;

# server\_name\_in\_redirect off;

include /etc/nginx/mime.types;

default\_type application/octet-stream;

##

# SSL Settings

##

ssl\_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE

ssl\_prefer\_server\_ciphers on;

##

# Logging Settings

##

access\_log /var/log/nginx/access.log;

error\_log /var/log/nginx/error.log;

##

# Gzip Settings

##

gzip on;

# gzip\_vary on;

# gzip\_proxied any;

# gzip\_comp\_level 6;

# gzip\_buffers 16 8k;

# gzip\_http\_version 1.1;

# gzip\_types text/plain text/css application/json application/javascript text/xml application/xml application/xml+rss text/javascript;

##

# Virtual Host Configs

##

include /etc/nginx/conf.d/\*.conf;

#Changed from enabled to available

include /etc/nginx/sites-available/\*;

}

#mail {

# # See sample authentication script at:

# # http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript

#

# # auth\_http localhost/auth.php;

# # pop3\_capabilities "TOP" "USER";

# # imap\_capabilities "IMAP4rev1" "UIDPLUS";

#

# server {

# listen localhost:110;

# protocol pop3;

# proxy on;

# }

#

# server {

# listen localhost:143;

# protocol imap;

# proxy on;

# }

#}

## The file /etc/nginx/sites-available/frontend

server {

server\_name omppujarane.store;

location / {

proxy\_pass http://localhost:8080; #whatever port your app runs on

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

}

## The file /etc/nginx/sites-available/koiravaatekauppa

server {

server\_name spring.omppujarane.store;

location / {

proxy\_pass http://localhost:8081; #whatever port your app runs on

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

}