Universiteit Gent

Software Engineering Lab 2

Installatiehandleiding

Groep 2

Academiejaar 2020-2021

Contents

1 Inleiding

Dit document dient als een installatiehandleiding van het project en als documentatie van systeembeheer. De documentatie beschrijft de omgeving waarin het project wordt geïnstalleerd.

De eerstvolgende sectie is een beschrijving van de infrastructuur. Dit is de info die wij ontvangen hebben over de infrastructuur op de website van het vak.

De tweede en derde secties zijn de installatiestappen voor de back- en front-end.

Alle andere secties beschrijven hoe er omgegaan wordt met veiligheid, back-ups en gebruikersbeheer.

2 Beschrijving infrastructuur

Elke groep krijgt toegang tot een eigen (virtuele) server waarop de software van het project uiteindelijk zal moeten draaien. Dezelfde server dient tegelijkertijd als verslaginstrument. De groep staat zelf in voor het beheer van deze server: gebruikersbeheer, bestandsbeheer, veiligheid, enz. We raden heel sterk aan om back-ups te nemen van de belangrijkste bestanden op een andere plaats dan op deze server.

Bij de start van het project krijgt de systeemadministrator van de groep een root password toegewezen van een minimaal geïnstalleerd systeem. De groep moet dan zelf de nodige gebruikers aanmaken, verdere software installeren en configureren.

De docenten en assistenten moeten ten allen tijde toegang tot deze server krijgen (via reeds vooraf geconfigureerde SSH public keys), voor noodgevallen. Onze server heeft een vast IP-adres en een vaste naam: `sel2-2.ugent.be`.

Het spreekt voor zich dat deze server enkel mag gebruikt worden voor activiteiten die rechtstreeks met het project te maken hebben. (Externe) firewallsoftware zal ervoor zorgen dat het aantal geëxporteerde poorten beperkt is tot HTTP (poort 80) en HTTPS (poort 443). Daarnaast is SSH beschikbaar via VPN op de standaard poort 22. Je kan SSH bijkomend laten luisteren op poort 2002, welke ook van buitenaf bereikbaar is zonder VPN.

Opgelet: als jullie gebruik zouden willen maken van Docker, zorg er dan voor dat Docker een andere lokale interface dan de standaard range 172.17.*.* gebruikt. Want anders zal jullie server niet bereikbaar zijn via eduroam omdat eduroam dezelfde private ip range gebruikt en hiermee conflicteert.

3 Installatiegids: Vue als front-end

3.1 Installatie van Node.js

Node.js kan op twee manieren geïnstalleerd worden. Via de package manager van Ubuntu of via NVM[1]. NVM heeft als voordeel dat er snel tussen verschillende versies van Node.js gewisseld kan worden. NVM is ook ondersteund door Jenkins 9. We kiezen daarom voor NVM.

Installatiecommando NVM:
`curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.37.2/install.sh | bash`

3.2 Installatie van Vue

De installatie van Vue[2] verloopt volgens volgende korte stappen:
`npm install vue`

De volgende stap is het installeren van de node modules vanuit de root-map van het project:
`npm install`

Dan is het mogelijk de development server op te starten met:
`npm run serve`
De front-end zou dan beschikbaar moeten zijn op localhost via poort 8080. (http://localhost:8080)

3.3 Structuur van het project

In dit project gebruiken we Vue in combinatie met Vuex en de Vue router. De "pagina's" kunnen gevonden worden in de `‘src/views‘` map. In deze views gebruiken we componenten die zich bevinden in de `‘src/components‘` map.
De navigatie tussen de pagina's wordt beschreven in het bestand `‘src/router/index.js‘`.
De store en modules (`‘src/store‘`) worden gebruikt om een centrale state te behouden wanneer de gebruiker tussen pagina's navigeert.

3.4 Vue en Nginx

We creëren een uitvoerbaar bestand van onze Vue-bestanden via:
`npm run build`
Dit wordt dan statisch geserved door Nginx.

[1] https://github.com/nvm-sh/nvm
[2] `https://vuejs.org/`

4 Installatiegids: Java Spring als back-end

4.1 Installatie Maven

Voer volgende commando's uit om Maven te installeren:
`apt install default-jdk`
`apt install maven`
verificatie:
`mvn -version`

Als men de meest recente versie wil, dan kan het dat men van de bron moet bouwen. Raadpleeg dan `https://linuxize.com/post/how-to-install-apache-maven-on-ubuntu-18-04/` voor een handleiding.

4.2 Service file

```
[Unit]
Description=Selab api server

[Service]
EnvironmentFile=-/etc/default/apidev
Restart=always
StandardOutput=syslog
StandardError=syslog
Environment=SERVER_PORT=9090
SyslogIdentifier=Selab-API-server
ExecStart=/selabrepo-back-dev/backend/api/api/mvnw clean spring-boot:run
WorkingDirectory=/selabrepo-back-dev/backend/api/api

[Install]
WantedBy=multi-user.target
```

Belangrijk hierbij zijn de environnement variables, die toelaten de poort aan te passen. De service faalt echter wel als de `./mvnw` niet uitvoorbaar is. Je kan dit oplossen met `chmod +x ./mvnw`.

5 Nginx als web/proxy server

We kozen voor Nginx door de wijdverspreide documentatie en de goede performantie (response time) van de server.

5.1 Error pagina's

De server is zo ingesteld dat als er een $50x$ error optreedt, een statische errorpagina weergegeven wordt uit de `\htdocs\error\` directory.
Dit is handig omdat bij een "502 bad gateway"-error een pagina kan weergegeven worden in dezelfde huisstijl als de webapp. Dit zou kunnen optreden wanneer de Java-server niet meer draait.

Hetzelfde geldt voor de andere errorpagina's die kunnen optreden, zoals 404 en 401 errors.

5.2 Configuratie

De configuratie van Nginx is te vinden in appendix B. In die configuratie is Gzip[3]-compressie en SSL stapling ingeschakeld en HTTP2 kan gebruikt worden. De 'expires header' werd zodanig gezet zodat de cache optimaal gebruikt wordt (client-side) en alle security headers werden toegevoegd, zie sectie veiligheid 7.2.

6 Databank

6.1 PostgreSQL

PostgreSQL kan eenvouding geïnstalleerd worden aan de hand van volgende commando's:

```
sudo apt install Postgresql Postgresql-contrib
sudo -i -u Postgres
createuser --interactive -p
```

6.2 Inladen van de tabellen (DDL)

Het UML-diagram van de voorlopige databank is toegevoegd in de appendix. Dit diagram is gemaakt met Umbrello[4].

6.2.1 Stap 0 (Optioneel): Opstellen SQL-code voor tabeldefinities

Delen van de broncode worden gemaakt door Umbrello. Deze kun je steeds opnieuw maken door Umbrello te installeren en vervolgens code -> code generation wizard op te roepen. Voer `finalise_sql.sh` uit om `all_schemas.sql` aan te maken.

6.2.2 Stap 1: Installatie databank

Installeer PostgreSQL. Laat de PostgreSQL-server draaien op poort 2002. Maak een nieuwe databank aan. Noem deze "magdadatabase". Gebruik de SQL-code uit `database/src/code/create_user.sql` om de gebruiker aan te maken.

[3] `gzip.org`
[4] `https://umbrello.kde.org`

6.2.3 Stap 2: Tabellen opstellen

Gebruik de SQL-code uit `database/src/code/all_schemas.sql` om de tabellen op te stellen. Wanneer dit bestand niet bestaat, kan dit aangemaakt worden zoals beschreven in stap 0.

6.3 pgAdmin

Om makkelijk de databank te beheren en sql-scripts uit te voeren installeerden we pgAdmin4. Deze is via een reverse proxy beschikbaar op `sel2-2.ugent.be/db/` en wordt extra afgeschermd[5] met basic auth. Als extra veiligheidsstap zou deze webinterface normaal via ip-controle bereikbaar zijn. Dus de pagina is enkel beschikbaar als je in de whitelist staat, maar sinds dit niet haalbaar is (clienten hebben geen statische ip) laten we dit even achterwege.

6.3.1 Installatie pgAdmin

Het gemakkelijkste is om pgAdmin te installeren via Python[6] in plaats van het apt commando. Dit is zo omdat de Ubuntu package verwacht dat men met Apache werkt.

```
sudo mkdir /var/lib/pgAdmin
mkdir /var/log/pgAdmin
sudo chown $USER /var/lib/pgAdmin
sudo chown $USER /var/log/pgAdmin
python3 -m venv pgAdmin4
source pgAdmin4/bin/activate
sudo apt install krb5-config krb5-user libkrb5-dev
(pgAdmin4)  pip install pgAdmin4
(pgAdmin4) $ pgAdmin4
NOTE: Configuring authentication for SERVER mode.


Enter the email address and password to use for the initial pgAdmin user account:


Email address: user@domain.com
Password:
Retype password:
Starting pgAdmin 4. Please navigate to http://127.0.0.1:5050 in your browser.
```

7 Veiligheid

7.1 SSL

SSL wordt geregeld via getssl[7], dit is een bash script dat certificaten tijdig vernieuwd via Let's Encrypt[8]. Het voordeel van een BASH-script is dat er zeer weinig dependencies zijn[9]. Dit zorgt ervoor dat het programma zeer betrouwbaar is in een productieomgeving. In tegenstelling tot de alternatieven, zal het

[5] bovenop het accountsysteem van pgAdmin
[6] `https://www.pgAdmin.org/download/pgAdmin-4-python/`
[7] https://github.com/srvrco/getssl
[8] https://letsencrypt.org/
[9] Enkel libcurl, wat vaak voorgeïnstalleerd is op linux.

script zelden breken wanneer een dependency geüpdatet wordt (bv. bij het upgraden van het systeem). De certificaten worden opgeslagen in de directory `/ssl` en er wordt dagelijks een back-up [10] genomen.

7.2 Security headers

In Nginx zijn er verschillende security headers ingesteld. Hoe ze worden toegevoegd staat in appendix B.

> Modern browsers support many HTTP headers that can improve web application security to protect against clickjacking, cross-site scripting, and other common attacks.[11]

7.2.1 Wat zijn security headers?

HTTP-security headers zijn een deelverzameling van HTTP-headers die worden uitgewisseld tussen de webclient (bijvoorbeeld de browser) en een server om de beveiligingsgerelateerde details van de HTTP-communicatie te specificeren. Sommige HTTP-headers zijn indirect gerelateerd aan privacy en beveiliging en kunnen ook een als een HTTP-security header beschouwd worden. Door het toevoegen van gepaste headers in een webapplicatie en webserver wordt je webapplicatie beschermt tegen vaak voorkomende aanvallen, zoals cross-site scripting (XSS) [12] en clickjacking[13].

HTTP-security headers geven een extra laag van beveiliging door restricties te plaatsen op het gedrag die de browser en server toestaan eens de webapplicatie draait. In veel gevallen is het implementeren van de correcte headers een cruciaal deel van een "best-practice application setup".

7.2.2 HTTP Strict Transport Security (HSTS)

De HSTS header verplicht het gebruik van geëncrypteerde HTTPS connecties in plaats van plain-text HTTP communicatie.

Deze is op de volledige server (dus elke pagina, ook al komt die van de proxy) aangezet, en wordt dus altijd teruggeven door de server.

7.2.3 Cross Site Scripting Protection (X-XSS)

Deze header is een functionaliteit van Internet Explorer, Chrome en Safari die het laden van pagina's stopt als een cross-site scripting (XSS) aanval gedetecteerd wordt.

7.2.4 Content-Security-Policy

De Content Security Policy (CSP) header is de beste manier om te beschermen tegen XSS aanvallen. Deze laat toe om contentbronnen te controleren en die eventueel te blokkeren.

7.2.5 X-Frame-Options

Dit zorgt ervoor dat de pagina niet in een iframe mag geladen worden. De webapplicatie mag dus niet ingevoegd worden in een andere pagina.

[10] Zie sectie 12
[11] https://www.netsparker.com/blog/web-security/http-security-headers/
[12] https://owasp.org/www-community/attacks/xss/
[13] https://owasp.org/www-community/attacks/Clickjacking

7.2.6 X-Content-Type-Options

Deze header verplicht webbrowsers om strict de MIME-types gespecificeerd in de Content-Type header te volgen. Dit beschermt websites tegen cross-site scripting aanvallen die gebruik maken van "MIME sniffing" om kwaadaardige code uit te voeren die zich voordoet als een niet-uitvoerbaar MIME-type.

7.2.7 Expect-CT

Om "website certificate spoofing" te vermijden, wordt de Expect-CT header toegevoegd om aan te duiden dat enkel nieuwe certificaten die in de transparante logs van de CA zijn opgenomen, toegelaten worden. Let's Encrypt hanteert ook ct-logs[14] en deze header kan dus aangezet worden op elke pagina.

7.2.8 Referrer-Policy

Hiermee controleren we hoeveel referrer-informatie aan de webserver wordt doorgegeven.

7.3 ConfigServer Security Firewall

ConfigServer Security & Firewall (CSF)[15] heeft een steilere "learning curve" dan Uncomplicated Firewall (UFW), die standaard met Ubuntu wordt meegeleverd. CSF heeft echter wel meer mogelijkheden: het is niet enkel een stateful packet inspection firewall (SPI) zoals UFW, maar ook een intrusion detection system (IDS). Het heeft ook een login failure daemon en DDOS-protectie.

7.3.1 Gebruikte functionaliteit

Hier lijsten we de belangrijkste features op, die we effectief gaan inzetten om beter de veiligheid te garanderen.

1. SSH en SU login notification (Alle mails worden naar het e-mailadres van de systeembeheerder verzonden, zie sectie 8.1)
2. Excessive connection blocking
3. Block traffic on unused server IP addresses
4. Alert when end-user scripts sending excessive emails per hour (Identificeren van spamming scripts)
5. Suspicious process reporting (reports potential exploits running on the server)
6. Excessive user processes reporting
7. Suspicious file reporting - reports potential exploit files in /tmp and similar directories
8. Directory and file watching - reports if a watched directory or a file changes
9. Block traffic on a variety of Block Lists including DShield Block List and Spamhaus DROP List
10. BOGON packet protection
11. IDS (Intrusion Detection System) - De onderste meldingsdienst die ons op de hoogte brengt als er aanpassingen zijn aan het systeem of application binaries
12. SYN Flood protection

[14] https://letsencrypt.org/docs/ct-logs/
[15] https://www.configserver.com/cp/csf.html

13. Ping of death protection
14. Port Scan tracking and blocking
15. Account modification tracking - stuurt meldingen als er wijzigingen zijn aan de gebruikersaccounts
16. Country Code blocking - weiger verbindingen uit gekozen landen
17. Port Flooding Detection - connection flooding detection per IP en poort

7.3.2 Lfd

Om de ConfigServer Firewall (CSF) te ondersteunen,is er een Login Failure Daemon (lfd) proces dat periodiek de laatste logbestanden scant en zoekt naar aanmeldingspogingen die snel falen in een korte tijd. Dit zijn vaak "Brute-force attacks". Het daemon proces reageert zeer snel op zulke patronen en blokkeert de IP's waar die pogingen vandaan komen.

7.4 Web Application Firewall

Dit wordt geïmplementeerd in Nginx voor milestone 2. Een Web application Firewall (WAF) is een extra beveiligingslaag en staat tussen de firewall en de applicatieserver. De WAF probeert patronen te herkennen in de pakketten naar de applicatieserver. Het blokkeert bijvoorbeeld SQL-injecties en ongeldige JSON.

7.5 Nginx Misconfiguratie

Om Nginx misconfiguratie tegen te gaan, is het pakket gixy[16] geïnstalleerd. Dit controleert een Nginx-configuratie op fouten. Iedere dag om midderdacht wordt de Nginx-configuratie gecontroleerd. Het resultaat wordt dan via een cron alert naar het e-mailadres van de systeembeheerder verstuurd.

8 Waarschuwing

Als er iets foutloopt op de server, dan is het natuurlijk belangrijk dat we zo snel mogelijk op de hoogte gebracht worden.

8.1 Mail

Na sommige testen[17] bleek dat alle externe Mail transfer agents niet bereikbaar zijn. Enkel `cypress.ugent.be` is op het lokale UGent-netwerk en dus wel bereikbaar. Alle mails van het systeem worden dus naar het UGent e-mailadres van de systeembeheerder verstuurd. Daar wordt het dan via regels doorgestuurd naar het correcte adres.

8.2 Alerts

Voor de volgende zaken worden er mails verstuurd:

- SSH logins

[16] https://github.com/yandex/gixy
[17] Telnet naar mijn eigen MTA (Postfix)

- Suspicious programs
- Veranderde hash van belangrijke bestanden
- Gefaalde cronjobs

9 Jenkins

9.1 Installatie van Jenkins

```
wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
sudo apt-get update
sudo apt-get install jenkins
```

10 Blue Ocean

Om snel te integeren met GitHub Enterprise en een flexibel multibranch project aan te maken, maken we gebruik van de Blue Ocean plugin. Dit is ook een makkelijke pipeline editor. Het Jenkinsbestand op GitHub kan dus eenvoudig ingelezen en aangepast worden. Daarnaast biedt het een makkelijkere en mooiere interface om de statussen op te volgen. De installatie gebeurt door Blue Ocean te installeren vanuit de Jenkins plugins sectie. Door het Blue Ocean stappenplan te volgen kan er snel een project opgezet worden.

11 NVM en jenkins

NVM is ondersteund door een plugin nvm-wrapper[18]. Zo kunnen we snel van Node.js versie wisselen in Jenkins zelf.

12 Backup

Er wordt dagelijks een backup gemaakt van alle belangrijke bestanden op de server. Dit doen we via een cronjob en git. Git werkt via ssh en geeft dus een beveiligde (versleutelde) verbinding. We moeten dus enkel de gevoelige informatie versleutelen, zoals de databank. Door de firewallbeperkingen hebben niet gekozen voor rsync[19] of rsnapped[20], hoewel deze zeer handig zijn voor incrementele back-ups. Een database cluster voor redundantie lukt ook niet door de firewallbeperkingen.

- Service files, api, api dev, pgAdmin, ..
- Nginx config
- Versleutelde Databank
- Crontab

[18] https://plugins.jenkins.io/nvm-wrapper/
[19] https://www.digitalocean.com/community/tutorials/how-to-use-rsync-to-sync-local-and-remote-directories
[20] https://github.com/ohitz/rsnap

A Databankdiagram

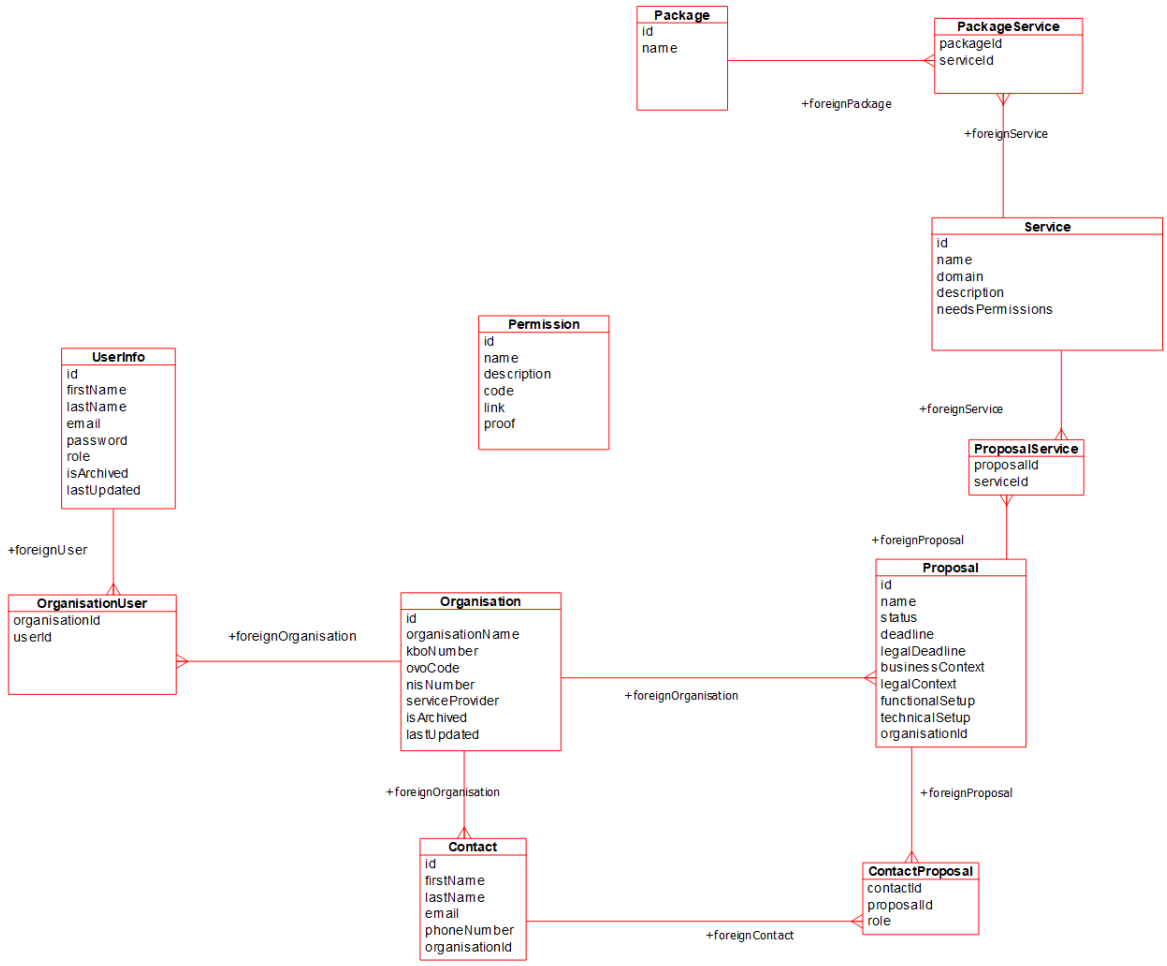

Figure 1: Het databankdiagram, gemaakt met Umbrello

B Nginx

```
upstream jenkins {
  keepalive 32; # keepalive connections
  server 127.0.0.1:7070; # jenkins ip and port
}

# Required for Jenkins websocket agents
map $http_upgrade $connection_upgrade {
  default upgrade;
  '' close;
}

# Expires map
map $sent_http_content_type $expires {
    default                    off;
    text/html                  epoch;
    text/css                   max;
    application/javascript     max;
```

```
    ~image/                     max;
}

    server {
        listen 80 default_server;
        listen [::]:80 default_server;

        # SSL configuration
        listen 443 ssl http2 default_server;
        listen [::]:443 ssl http2 default_server;

            gzip  on;
            # compress proxied requests too.
# it doesn't actually matter if the request is proxied, we still want it
    compressed.
gzip_proxied any;

# a pretty comprehensive list of content mime types that we want to
    compress
# there's a lot of repetition here because different applications might use
     different
# (and possibly non-standard) types. we don't really care, we still want
    them included
# don't include text/html -- it is always included anyway
gzip_types
        text/css
        text/plain
        text/javascript
        application/javascript
        application/json
        application/x-javascript
        application/xml
        application/xml+rss
        application/xhtml+xml
        application/x-font-ttf
        application/x-font-opentype
        application/vnd.ms-fontobject
        image/svg+xml
        image/x-icon
        application/rss+xml
        application/atom_xml;

# increase the compression level, at the expense of additional CPU
# cpu cycles are cheap virtually everywhere now, bandwidth not nearly as
    much
gzip_comp_level 9;

# the default is to gzip only HTTP 1.1 requests
# we want to gzip http 1.0 requests, too, so lower the level required
gzip_http_version 1.0;

# increase the size of the buffers which hold responses to make sure larger
     content can be compressed too
# this means there are 16 buffers and they can each hold 8k
# if you serve a lot of ridiculously large text (like combined CSS) you
    might consider upping this slightly
gzip_buffers 16 8k;
```

```
# up the minimum length a little to account for gzip overhead
# this means anything smaller than 50 bytes won't be compressed.
# the default is 20 bytes, which is sooo tiny it's a waste to compress
gzip_min_length 50;


        server_name sel2-2.ugent.be;
        more_set_headers "Server: no server";
        ssl_certificate /ssl/fullchain.crt;
        ssl_certificate_key /ssl/private.key;
        ssl_session_timeout 5m;
        ssl_session_tickets off;
        ssl_stapling on;
        ssl_stapling_verify on;
        ssl_prefer_server_ciphers on;
        ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-SHA512:
            ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-
            AES256-SHA384;
        ssl_protocols TLSv1.1 TLSv1.2 TLSv1.3;
        ssl_session_cache shared:SSL:10m;
        ssl_ecdh_curve secp384r1;

        # HSTS header, eigenlijk ook een security header
        add_header Strict-Transport-Security "max-age=15768000";

        # Security headers
        add_header X-Frame-Options SAMEORIGIN;
        add_header X-Content-Type-Options nosniff;
        add_header X-XSS-Protection "1; mode=block";
        add_header Permissions-Policy "geolocation=();midi=();notifications
            =();push=();sync-xhr=(self);microphone=();camera=();magnetometer
            =();gyroscope=();speaker=(self);vibrate=();fullscreen=(self);
            payment=();";
        add_header Referrer-Policy "strict-origin";
        add_header Expect-CT "max-age=96400, enforce";
        add_header Cross-Origin-Embedder-Policy "require-corp; report-to=
            default";

        # root en index definitie
        root   /htdocs/;
        index  index.html;

        # geen server identificatie
        server_tokens off;


        # regel cache, op bassis van mime type selecteer de beste cache TTL
        expires $expires;

        # Als http2 push enkele resources. Vooral gebruikt bij location
        #http2_push /resources/fonts/stereo.woff2;

        # definieer nieuwe error page
        error_page 404 /error/404.html;

        #redirect http naar https
        if ($scheme = http) {
                return 301 https://$server_name$request_uri;
```

```
}

# Haal ssl verificatie bestanden op van een andere locatie
location ^~ /.well-known/acme-challenge/ {
        default_type "text/plain";
        alias /var/www/acme-challenge/;
        try_files $uri $uri/ =404;
}

# haal de home uit de std /htdocs directory
location / {
        more_set_headers "Content-Security-Policy: default-src '
            self'";
        try_files $uri $uri/ =404;
}

# redirect to webapp
location = /app {
        return 301 https://$server_name$request_uri/;
}

# redirect to api
location = /api {
        return 301 https://$server_name$request_uri/;
}

# redirect to webapp
location = /app/dev {
        return 301 https://$server_name$request_uri/;
}
# redirect to webapp
location = /api/dev {
        return 301 https://$server_name$request_uri/;
}



# serve de webapp
location /app/ {
        auth_basic "Administrator's Area";
        auth_basic_user_file /htdocs/data/.htpasswd;

        more_set_headers "Content-Security-Policy: default-src '
            self'";

        alias /selabrepo/frontend/dist/;
        try_files $uri $uri/ /app/;
}

# serve de webapp van dev versie
location /app/dev/ {
        auth_basic "Administrator's Area";
        auth_basic_user_file /htdocs/data/.htpasswd;
        more_set_headers "Content-Security-Policy: default-src '
            self'";

        alias /selabrepo-front-dev/frontend/dist/;
        try_files $uri $uri/ /app/dev/;
```

```
}


# Alle bestanden vna de data server
location  /data/static/ {
        auth_basic "Administrator's Area";
        auth_basic_user_file /htdocs/data/.htpasswd;
        alias /selab/static/;
        more_set_headers "Content-Security-Policy: default-src '
            self'";
        try_files $uri $uri/ =404;
}

# Data server html (c++ program in /selab directory, )
location /data/ {
        auth_basic "Administrator's Area";
        auth_basic_user_file /htdocs/data/.htpasswd;
        proxy_pass http://localhost:3075;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $host;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Host $host;
        more_set_headers "Content-Security-Policy: default-src '
            self'";
}

# Data server html (c++ program in /selab directory, )
location /api/ {
        more_set_headers "Content-Security-Policy: default-src '
            self'";
more_set_headers "Access-Control-Allow-Origin: *";
more_set_headers "Access-Control-Allow-Methods: GET, POST, PATCH,
    OPTIONS";
more_set_headers "Access-Control-Allow-Headers: DNT,User-Agent,X-
    Requested-With,If-Modified-Since,Cache-Control,Content-Type,
    Range";
more_set_headers "Strict-Transport-Security: max-age=31536000";
more_set_headers "X-Frame-Options: deny";
more_set_headers "Access-Control-Allow-Credentials: true";
        more_set_headers "Access-Control-Expose-Headers: Content-
            Length,Content-Range";

proxy_pass http://localhost:8080/;
    proxy_set_header X-Script-Name /api/;
proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $host;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-Prefix /api/;
}

# Data server html (c++ program in /selab directory, )
location /api/dev/ {
proxy_pass http://localhost:9090/;
proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $host;
```

```
            proxy_set_header X-Forwarded-Proto $scheme;
            proxy_set_header X-Forwarded-Prefix /api/dev/;
            proxy_set_header Host $host;
            more_set_headers "Content-Security-Policy: default-src '
                self'";
    more_set_headers "Access-Control-Allow-Origin: *";
    more_set_headers "Access-Control-Allow-Methods: GET, POST, PATCH,
        OPTIONS";
    more_set_headers "Access-Control-Allow-Headers: DNT,User-Agent,X-
        Requested-With,If-Modified-Since,Cache-Control,Content-Type,
        Range";
    more_set_headers "Strict-Transport-Security: max-age=31536000";
    more_set_headers "X-Frame-Options: deny";
    more_set_headers "Access-Control-Allow-Credentials: true";
            more_set_headers "Access-Control-Expose-Headers: Content-
                Length,Content-Range";

    }


    error_page   500 502 503 504  /50x.html;
            location = /50x.html {
    }

    error_page   404  /404.html;
            http2_push /style.css;
            location = /404.html {
    }
    location /db/ {
            auth_basic "Administrator's Area";
            auth_basic_user_file /htdocs/data/.htpasswd;
    proxy_set_header X-Script-Name /db;
            proxy_set_header X-Scheme $scheme;
    proxy_set_header Host $host;
    proxy_pass http://localhost:5050/;
    proxy_redirect off;
    }

    location /jenkins/ {
  sendfile off;
  proxy_pass          http://jenkins;
  proxy_redirect      default;
  proxy_http_version 1.1;

  # Required for Jenkins websocket agents
  proxy_set_header   Connection        $connection_upgrade;
  proxy_set_header   Upgrade           $http_upgrade;
  proxy_set_header   Host              $host;
  proxy_set_header   X-Real-IP         $remote_addr;
  proxy_set_header   X-Forwarded-For   $proxy_add_x_forwarded_for;
  proxy_set_header   X-Forwarded-Proto $scheme;
  proxy_max_temp_file_size 0;

  #this is the maximum upload size
  client_max_body_size       10m;
  client_body_buffer_size    128k;

  proxy_connect_timeout      90;
```

```
        proxy_send_timeout          90;
        proxy_read_timeout          90;
        proxy_buffering             off;
        proxy_request_buffering     off; # Required for HTTP CLI commands
        proxy_set_header Connection ""; # Clear for keepalive
}

}
```