Curs 2023 - 2024

Proxy invers i balanceig de càrrega amb SSL en NGINX

Guillermo Vidal Frasquet

Desplegament d'Aplicacions Web

Pràctica

IES Jaume II El Just
Tavernes de la Valldigna

Continguts

1 Proxy invers i balanceig de càrrega amb SSL en NGINX **2**
1.1 Requisits abans de començar la pràctica 2
1.2 Introducció 2
1.2.1 Certificats 3
1.3 Tasca 1 4
1.3.1 Creació de certificat autosignat 4
1.3.2 Configuració SSL en el proxy invers 6
1.3.3 Comprovacions 7
1.4 Tasca 2 8
1.4.1 Redirecció forçosa a HTTPS 8

1 Proxy invers i balanceig de càrrega amb SSL en NGINX

1.1 Requisits abans de començar la pràctica

Atenció, molt important abans de començar

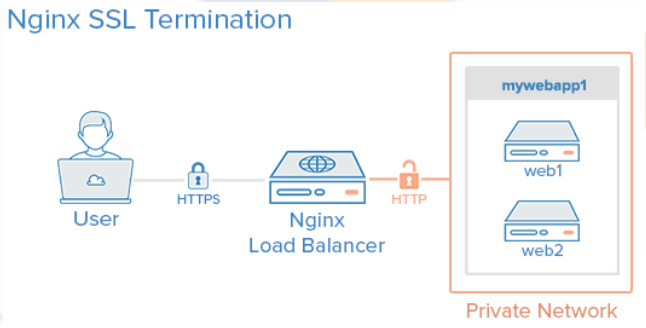
- La pràctica 2.4 ha d'estar funcionant correctament.
- No començar la pràctica abans de tindre la 2.4 **funcionant i comprovada**.

1.2 Introducció

A partir de les pràctiques anteriors hem arribat a un escenari on un **proxy invers** actua d'intermediari entre dos servidors web Nginx, **balancejant la càrrega** entre ells.

Ja vam dir que una important funció que podia tindre un proxy invers era realitzar el xifratge i desxifrat de SSL per a utilitzar HTTPS en els servidors web. D'aquesta manera s'alleujava la càrrega de treball dels servidors web, ja que és una tasca que consumeix recursos.

En definitiva, tindríem un esquema com aquest:



Podria arribar-se a pensar que en termes de seguretat no és adequat que el trànsit de xarxa entre el **balancejador de càrrega** i els servidors web vaja sense xifrar (HTTP). No obstant això, pensant en un cas real, la xarxa privada i el **proxy invers/balancejador de càrrega**, a més d'estar en la mateixa xarxa privada, solen estar administrats per les mateixes persones de la mateixa empresa, per la qual cosa no suposa un perill real que aquest trànsit vaja sense xifrar.

Podria xifrar-se si fora necessari, però llavors perd sentit que el proxy invers s'encarregue del xifratge SSL per a HTTPS, ja que faríem el mateix treball dues vegades.

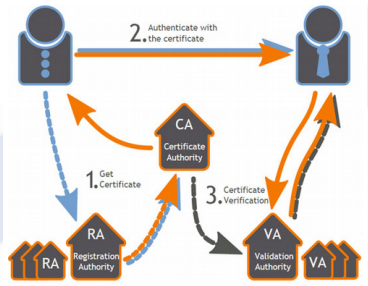
Així les coses, ens quedarem amb l'esquema de la imatge de més amunt per a la pràctica.

1.2.1 Certificats

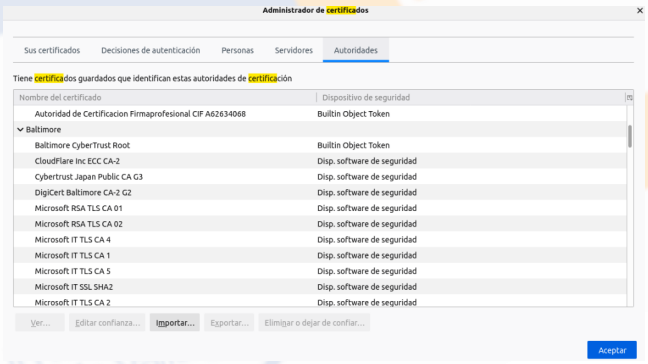
HTTPS es basa en l'ús de certificats digitals.

Grosso modo, quan entrem en una web via HTTPS, aquesta ens presenta un certificat digital per a assegurar que és qui diu ser. Com sabem que aquest certificat és vàlid? Hem de consultar a l'**Autoritat de Certificació (CA)** que va emetre aqueix certificat si és vàlid.

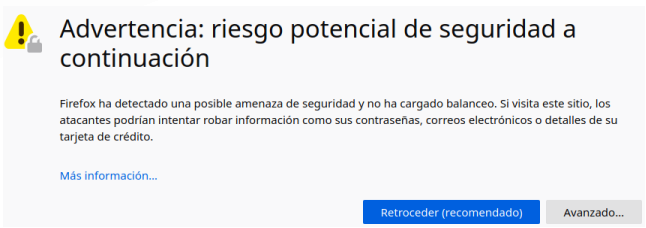
Les CA són entitats que emeten certificats i el seu funcionament es basa en la confiança. Confiem que els certificats emesos i signats per aqueixes entitats són reals i funcionals.



Els navegadors web tenen precarregades les **Autoritats de Certificació** en les quals confien per defecte a l'hora de navegar per webs HTTPS:



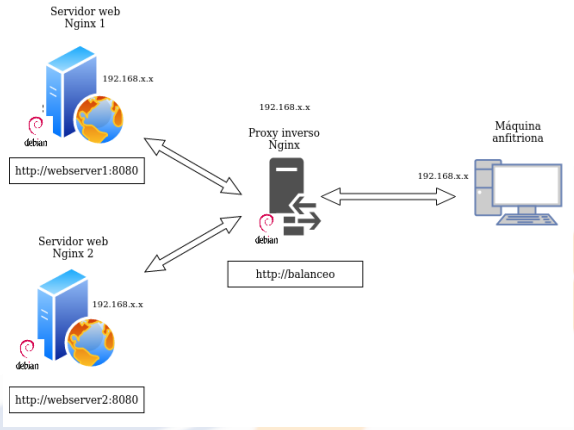
Si accedim a una web el certificat de la qual no haja sigut emés i signat per una d'aquestes entitats, ens saltarà el famós avís:



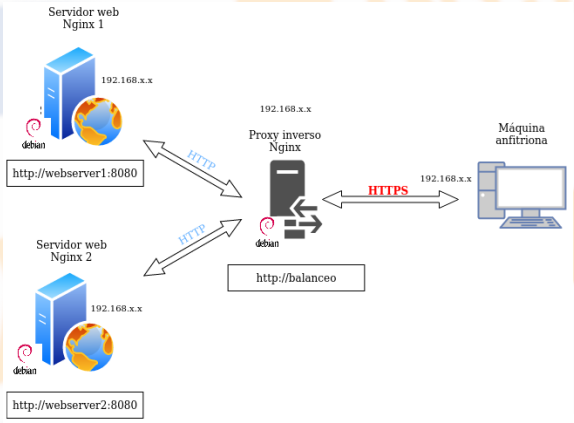
Ja que si el certificat no ha sigut emés i signat per una CA de confiança, pot ser que es tracte d'una web maliciosa que ens supose un risc de seguretat, com bé diu l'avís.

1.3 Tasca 1

Partim de la configuració exacta de la pràctica anterior, que recordem era aquesta:



Pel que en aquesta pràctica simplement hem d'afegir la configuració SSL per al xifratge en el **Proxy Invers**:



Tal com quedarà la configuració, des del client encara podríem accedir als dos servidors web amb HTTP (podeu provar-ho) però és una cosa que solucionarem en següents temes, configurant un firewall perquè només la IP del proxy invers puga accedir per HTTP als servidors web i ningú més.

1.3.1 Creació de certificat autosignat

Nosaltres no utilitzarem certificats de cap CA de confiança, bàsicament perquè:

- Els nostres serveis no estan publicats en Internet.
- Aquests certificats són de pagament.

Així doncs, nosaltres crearem els nostres propis certificats i els signarem nosaltres mateixos com si fórem una CA autèntica per a poder simular aquest escenari.

Això provocarà que quan accedim per HTTPS al nostre lloc web per primera vegada, ens salte l'avís de seguretat que es comentava en la introducció.

En aquest cas no hi haurà perill perquè estem 100% segurs que aquest certificat l'hem emés nosaltres per a aquesta pràctica, no hi ha dubtes.

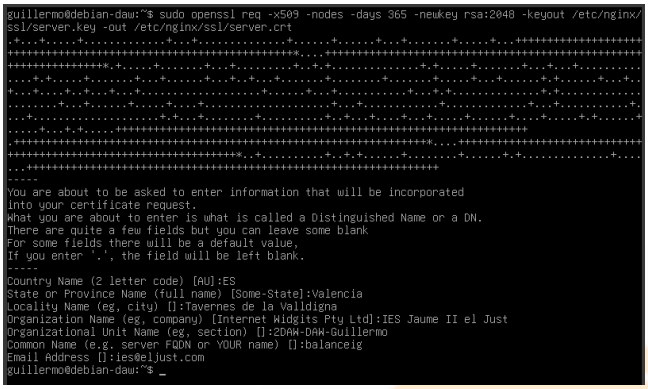
Vegem doncs el procés per a generar els certificats i les claus associades a ells (privada/pública). En primer lloc hem de crear el següent directori:

`/etc/nginx/ssl`

Podem crear el certificat i les claus de manera simultània amb un únic comando, on:

- **openssl**: aquesta és l'eina per línia de comandos bàsica per a crear i administrar certificats, claus i altres arxius **OpenSSL**.
- **req**: aquest subcomando s'utilitza per a generar una sol·licitud de certificats i també sol·licituds de signatura de certificats (CSR).
- **x509**: Això modifica encara més el subcomando anterior en dir-li a l'eina que volem crear un certificat autosignat en lloc de generar una sol·licitud de signatura de certificat, com succeiria normalment.
- **nodes**: Això li diu a OpenSSL que ometa l'opció d'assegurar el nostre certificat amb contrasenya. Necessitem que Nginx puga llegir l'arxiu sense la intervenció de l'usuari quan s'inicia el servidor. Una contrasenya evitaria que això succeïra ja que hauríem d'introduir-la a mà després de cada reinici.
- **days 365**: aquesta opció estableix el temps durant el qual el certificat es considerarà vàlid. Ho configurem per a un any.
- **newkey rsa: 2048**: Això especifica que volem generar un nou certificat i una nova clau al mateix temps. No creem la clau necessària per a signar el certificat en un pas anterior, per la qual cosa hem de crear-la juntament amb el certificat. La part **rsa: 2048** li diu que cree una clau RSA de 2048 bits de longitud.
- **keyout**: aquest paràmetre li diu a OpenSSL on col·locar l'arxiu de clau privada generat que estem creant.
- **out**: Això li diu a OpenSSL on col·locar el certificat que estem creant.

El comando complet seria així:



Us sol·licitarà que introduïu una sèrie de paràmetres, com veieu en el requadre roig de baix de la imatge. Heu d'introduir els mateixos paràmetres que en la imatge excepte en el "**Organizational Unit Name**" que veieu requadrat en groc. Ací haureu de posar **2DAW – DAW - Vostrenom**.

1.3.2 Configuració SSL en el proxy invers

De la pràctica anterior, dins del directori `/etc/nginx/sites-available` ja heu de tindre l'arxiu de configuració anomenat "**balanceig**". És precisament ací on realitzarem la configuració perquè l'accés al lloc web es realitze mitjançant SSL (HTTPS).

Dins del bloc `server {...}` heu de canviar el port d'escolta (listen 80) pel que veieu en el codi de baix, afegint les següents línies de configuració també, de tal forma que quede:

```
1 listen 443 ssl;
2 ssl_certificate /etc/nginx/ssl/server.crt;
3 ssl_certicate_key /etc/nginx/ssl/server.key;
4 ssl_protocols TLSv1.3;
5 ssl_ciphers ECDH-AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+
    AES128:ECDH+3DES:DH+3DES:RSA+AESGCM:RSA+AES:RSA+3DES:!aNULL:!MD5:!
    DSS;
6 server_name balanceig;
7 access_log /var/log/nginx/https_acces.log;
```

On li esteu dient que:

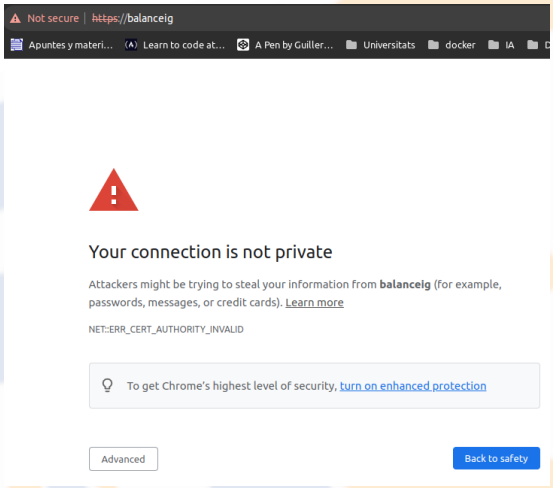
- Escolte en el port 443 → Port per defecte d'HTTPS.
- El directori on està el certificat que heu generat anteriorment.
- El directori on està la clau que heu generat anteriorment.
- Els protocols i tipus de xifratges que es poden utilitzar → Aquestes són les versions de protocols i els tipus de xifratges considerats segurs hui dia (hi ha molts més però no es consideren segurs actualment).

- **server_name** ja ho teníeu de la pràctica anterior, no fa falta tocar-ho.
- L'arxiu on es guarden els **logs** canvia de nom, ara serà `https_access.log`.

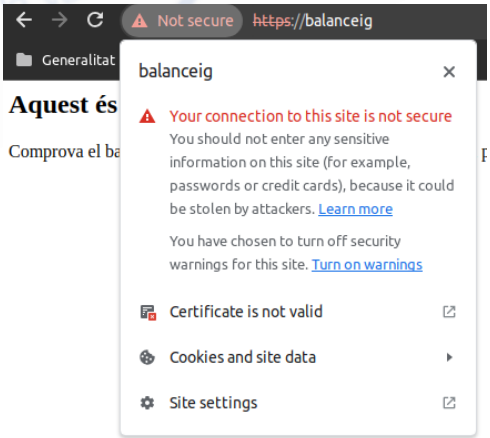
Recordeu que després de modificar qualsevol configuració d'un servei, cal reiniciar el servei, en aquest cas Nginx.

1.3.3 Comprovacions

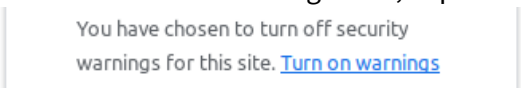
- Si accediu ara a `https://balanceig` us hauria de saltar un avís de seguretat pel fet que el nostre certificat és autosignat, com comentàvem anteriorment.



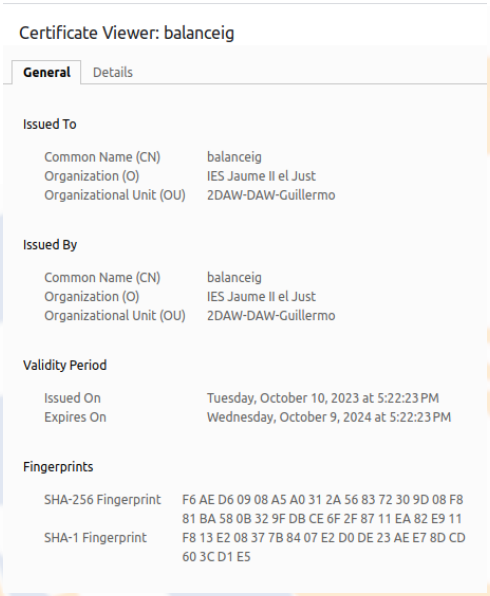
- Si afegiu una una excepció podreu accedir al lloc web i recarregant repetidament la pàgina amb F5, veureu que el balanceig de càrrega es fa correctament accedint mitjançant HTTPS.
- Per a comprovar que les dades del certificat són, efectivament, els vostres podeu comprovar-lo així. Prement en el cadenat de la barra de cerca:



Ací també podreu eliminar l'excepció que heu afegit en la pàgina de l'advertiment de seguretat, per si necessiteu reiniciar les proves.



I finalment, veure certificat:



Si ara intenteu accedir a `http://balanceig`, hauríeu de poder accedir? Comproveu-ho i descriviu què passa i per què.

1.4 Tasca 2

1.4.1 Redirecció forçosa a HTTPS

Perquè, indistintament de la forma per la qual accedim al lloc web **balanceig**, sempre es force a utilitzar HTTPS, necessitarem una configuració addicional.

Necessitem afegir un bloc "**server**" addicional i separat de l'altre, a l'arxiu de configuració de "**balanceig**". Una cosa així:

```
server {
        listen 80;
        server_name balanceig;
        access_log /var/log/nginx/http_access.log;
        return 301 https://balanceig;
}
```

Amb aquesta configuració li estem dient que:

- Escolte en el port 80 (HTTP).
- Que el nom al qual respondrà el servidor/lloc web és balanceig.
- Que guarde els logs d'aquest bloc en aquest directori i amb aquest nom.
- Quan es rep una petició amb les dues condicions anteriors, es retorna un codi HTTP 301:
  - **HTTP 301 Move Permanently** (Mogut permanentment en valencià) és un codi d'estat d'HTTP que indica que el host ha sigut capaç de comunicar-se amb el servidor però que el recurs sol·licitat ha sigut mogut a una altra direcció permanentement. Es molt important configurar les redireccions 301 en els llocs web i per a això hi ha diferents mètodes i sintaxis per a realitzar la redirecció 301.
  - La redirecció 301 és un codi o comando inserit per un **Webmaster** que permet redirigir als usuaris i cercadors d'un lloc web d'un lloc a un altre.

> És a dir, el que estem fent és que quan es reba una petició HTTP (port 80) en `http://balanceig`, es redirigisca a `https://balanceig` (HTTPS).

Per lliurar la tasca 2, cal fer:

- Afegiu el bloc **server{...}** per fer la redirecció a **HTTPS**.
- Reinicieu el servei.
- Comproveu ara que quan entreu en `http://balanceig`, automàticament us redirigeix a la versió segura de la web.
- Comproveu que quan realitzeu una petició en l'arxiu de log `http_access.log` apareix la redirecció 301 i que, de la mateixa manera, apareix una petició **GET** en `https_access.log`.
- Realitzeu les captures de pantalla necessàries per comprovar que tot funciona com cal, lliureu-les totes en un document pdf.