Curs 2023 - 2024

Proxy invers amb Nginx

Guillermo Vidal Frasquet

Desplegament d'Aplicacions Web

Pràctica

IES Jaume II El Just
Tavernes de la Valldigna

Continguts

1 Proxy invers amb Nginx

1.1 Requisits abans de començar la pràctica

Atenció, molt important abans de començar

- La pràctica 2.1 ha d'estar funcionant correctament.
- No començar la pràctica abans de tindre la 2.1 **funcionant i comprovada**.

1.2 Introducció

1.2.1 Què és un servidor proxy?

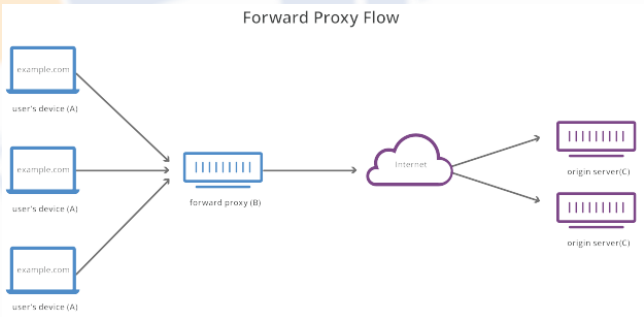
Un **proxy de reexpedició**, sovint anomenat **proxy**, **servidor proxy** o **proxy web**, és un servidor que es troba enfront d'un grup de màquines client. Quan aquestes màquines realitzen sol·licituds a llocs i serveis en Internet, el servidor proxy intercepta aquestes sol·licituds i després es comunica amb els servidors web en nom d'aquests clients, com un intermediari.

Per exemple, prenguem com a exemple 3 màquines involucrades en una comunicació típica de **proxy de reexpedició**:

A: Aquesta és la màquina d'un usuari.

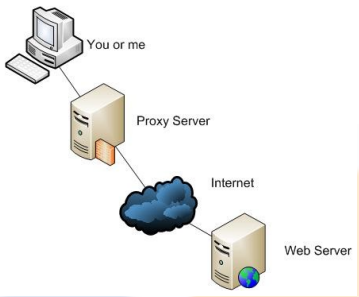
B: Aquest és un servidor proxy de reexpedició.

C: Aquest és el servidor d'origen d'un lloc web (on s'emmagatzemen les dades del lloc web).



En una comunicació estàndard per Internet, la màquina A es comunicaria directament amb la màquina C, amb el client enviant sol·licituds al servidor d'origen i el servidor d'origen responent al client. Quan hi ha un **proxy de reexpedició**, A enviarà sol·licituds a B, que després reexpedirà la sol·licitud a C. C enviarà una resposta a B, que reexpedirà la resposta a A.

Per què agregar aquest intermediari addicional a la nostra activitat en Internet?



Hi ha algunes raons per les quals un podria voler usar un **proxy de reexpedició**:

- **Per a evitar restriccions de navegació estatals o institucionals**: alguns governs, escoles i altres organitzacions usen **firewalls** per a donar als seus usuaris accés a una versió limitada d'Internet. Es pot usar un **proxy de reexpedició** per a sortejar aquestes restriccions, ja que permeten que l'usuari es connecte al proxy en lloc de directament als llocs que està visitant.
- **Per a bloquejar l'accés a un cert contingut**: al revés, els proxy també es poden configurar per a bloquejar l'accés d'un grup d'usuaris a uns certs llocs. Per exemple, una xarxa escolar pot estar configurada per a connectar-se a la web a través d'un proxy que habilita regles de filtrat de contingut, negant-se a reexpedir respostes de Facebook i altres llocs de xarxes socials.
- **Per a protegir la seua identitat en línia**: en alguns casos, els usuaris habituals d'Internet simplement desitgen un major anonimat en línia, però en altres casos, els usuaris d'Internet viuen en llocs on el govern pot imposar greus conseqüències als dissidents polítics. Criticar al govern en un fòrum web o en les xarxes socials pot donar lloc a multes o empresonament per a aquests usuaris. Si un d'aquests dissidents usa un **proxy de reexpedició** per a connectar-se a un lloc web on publica comentaris políticament sensibles, l'adreça IP utilitzada per a publicar els comentaris serà més difícil de rastrejar fins al dissident. Només estarà visible l'adreça IP del servidor proxy.

1.2.2 En què es diferencia un proxy invers?

Estaríem parlant del cas oposat a l'anterior.

Un **proxy invers** és un servidor que es troba enfront d'un o més servidors web, interceptant les sol·licituds dels clients. Això és diferent d'un **proxy de reexpedició**, on el proxy es troba enfront dels clients. Amb un **proxy invers**, quan els clients envien sol·licituds al servidor d'un lloc web, aquestes sol·licituds són interceptades a la frontera de la xarxa pel servidor **proxy invers**. El servidor **proxy invers** enviarà sol·licituds i rebrà respostes del servidor del lloc web.

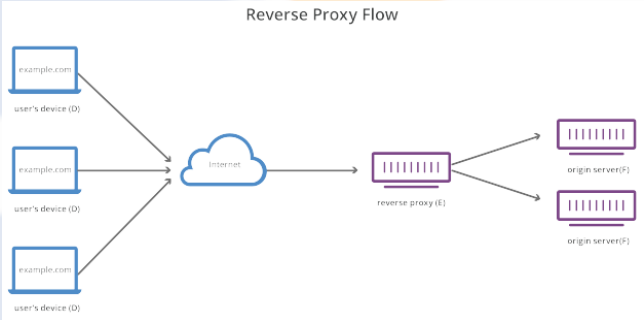
La diferència entre un proxy directe i invers és subtil però important. Una forma simplificada de resumir seria dir que un **proxy de reexpedició** es troba enfront d'un client i garanteix que cap servidor d'origen es comunique mai directament amb aqueix client específic. D'altra banda, un **proxy invers** es troba enfront d'un servidor d'origen i garanteix que cap client es comunique mai directament amb aqueix servidor d'origen.

Una vegada més, il·lustrem nomenant les màquines involucrades:

D: qualsevol nombre d'ordinadors domèstics dels usuaris.

E: aquest és un servidor **proxy invers**.

F: un o més servidors d'origen.



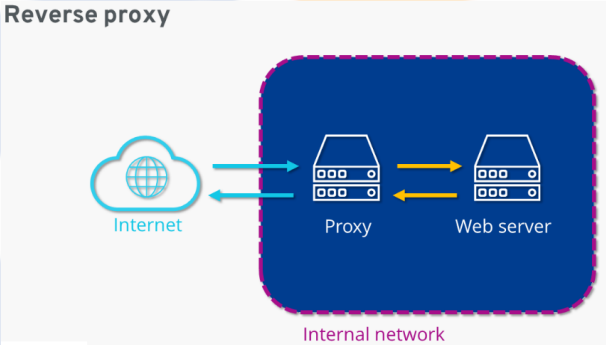
Normalment, totes les sol·licituds de D anirien directament a F, i F enviaria respostes directament a D. Amb un **proxy invers**, totes les sol·licituds de D aniran directament a E, i E enviarà les seues sol·licituds i rebrà respostes de F. E després transmeta les respostes apropiades a D.

A continuació es descriuen alguns dels beneficis d'un **proxy invers**:

- **Balanceig de càrrega**: és possible que un lloc web popular que rep milions d'usuaris tots els dies no puga manejar tot el trànsit entrant del lloc amb un solo servidor d'origen. En canvi, el lloc es pot distribuir entre un grup de servidors diferents, tots manejant sol·licituds per al mateix lloc. En aquest cas, un **proxy invers** pot proporcionar una solució de balanceig de càrrega que distribuirà el trànsit entrant de manera uniforme entre els diferents servidors per a evitar que un solo servidor se sobrecarregue. En el cas que un servidor falle per complet, altres servidors poden intensificar per a manejar el trànsit.
- **Protecció contra atacs**: amb un **proxy invers** en el seu lloc, un lloc web o servei mai necessita revelar l'adreça IP de la seua(s) servidor(s) d'origen. Això fa que siga molt més difícil per als atacants aprofitar un atac dirigit contra ells, com un atac DdoS.
- **Emmagatzematge en cau**: un **proxy invers** també pot emmagatzemar contingut en cau , la qual cosa resulta en un rendiment més ràpid. Per exemple, si un usuari a París visita un lloc web amb **proxy invers** amb servidors web a Los Angeles, l'usuari podria connectar-se a un servidor

proxy invers local a París, que després haurà de comunicar-se amb un servidor d'origen a Los Angeles. El servidor proxy després pot emmagatzemar en cau (o guardar temporalment) les dades de resposta. Els usuaris parisencs posteriors que naveguen pel lloc obtindran la versió en cau local del servidor **proxy invers** parisenc, la qual cosa donarà com a resultat un rendiment molt més ràpid.

- **Xifratge SSL** - Xifrat i desxifrat SSL (o TLS comunicacions) per a cada client poden ser computacionalment car per a un servidor d'origen. Es pot configurar un **proxy invers** per a desxifrar totes les sol·licituds entrants i xifrar totes les respostes sortints, alliberant valuosos recursos en el servidor d'origen.



1.3 Tasca

1.3.1 Configuracions

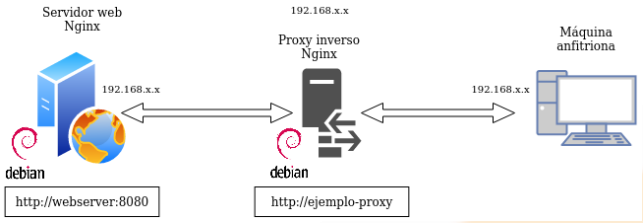
Servidor web Nginx

Configurarem dos Debian amb sengles servidors Nginx. Teniu la màquina virtual inicial i heu de clonar-la per a tindre una segona:

- Un servirà les pàgines web que ja hem configurat, així doncs utilitzarem el servidor que ja tenim configurat de la Pràctica 2.1.
- El nou servidor clon Debian amb Nginx configurat com proxy invers.
- Realitzarem les peticions HTTP des del navegador web de la nostra màquina física/amfitrió cap al **proxy clonat**, que ens redirigirà al servidor web original.

Compte
Ull en clonar les màquines virtuals perquè cal donar-li a crear una nova MAC, en cas contrari no tindreu IP en eixa màquina.

El diagrama de xarxa quedaria així:



Perquè tot quede més diferenciat i us quede més clar que la petició està passant pel **proxy invers** i arriba al servidor web destí, farem que cadascun dels servidors escolte les peticions en un port diferent.

1. En primer lloc, heu de canviar el nom que tinguera la vostra web pel de **webserver**, això implica:

   - Canviar el nom de l'arxiu de configuració de llocs disponibles per a Nginx.
   - Canviar el nom del lloc web dins d'aquest arxiu de configuració on faça falta.
   - No us oblideu d'eliminar el link simbòlic antic amb el comando `unlink nom_de el_link` dins de la carpeta `sites-enabled` i crear el nou per al nou nom d'arxiu.

2. En l'arxiu de configuració del lloc web, en lloc de fer que el servidor escolte en el port 80, canvieu-lo al 8080.

3. Reiniciar Nginx.

Proxy invers Nginx

Ara, quan intentem accedir a `http://exemple-proxy` (o el nom que tinguereu en la vostra web de les pràctiques anteriors), en realitat estarem accedint al proxy, que ens redirigirà a http://webserver:8080, el servidor web que acabem de configurar perquè escolte amb aquest nom en el port 8080.

Per a això:

- Crear un arxiu de configuració en `sites-available` amb el nom exemple-proxy (o el que tinguereu vosaltres).
- Aquest arxiu de configuració serà més simple, tindrà la següent forma:

```
server {
    listen __;
    server_name ____________;
    location / {
    proxy_pass http://_________:____;
    }
}
```

On, **mirant el diagrama de xarxa i tenint en compte la configuració feta fins ara**, heu de completar:

- El port on està escoltant el **proxy invers**.
- El nom del vostre domini o lloc web original al qual accedim en el proxy.
- La directiva `proxy_pass` indica a on es redirigiran les peticions, això és, al servidor web. Per tant, heu de posar la IP i número de port adequats del vostre lloc web configurat en l'apartat anterior.
- Crear el link simbòlic pertinent.

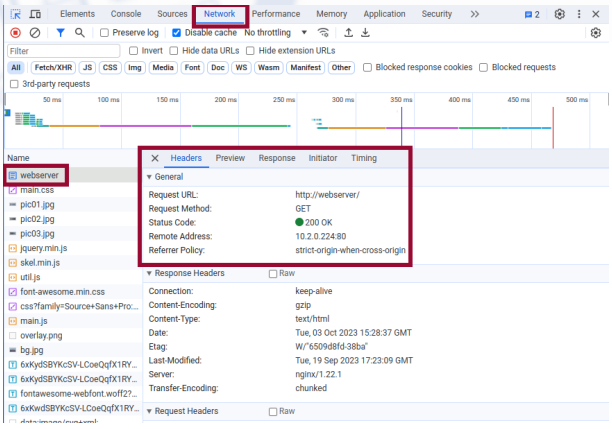
Això és per a simular la situació en la qual nosaltres, com a clients, quan accedim al nostre lloc web, no necessitem saber com està tot configurat, només necessitem saber el nom de la web.

Atenció, molt important
Heu de modificar l'arxiu host que vareu configurar en la pràctica 2.1. **Si mireu el diagrama de xarxa, ara el nom del vostre lloc web es correspondrà amb la IP de la nova màquina clon que fa de proxy**. Serà aquesta l'encarregada de redirigir-nos automàticament al vertader lloc web.

1.4 Comprobacions

Si accediu al vostre lloc web, heu de poder continuar accedint sense problemes.

- Comproveu en els `access.log` dels dos servidors que arriba la petició.
- Comproveu a més la petició i resposta amb les eines de desenvolupador del navegador en la màquina amfitriona. Prement F12 en el navegador us apareixeran aquestes eines.



En la primera petició (marcada en roig), utilitzant l'apartat "Xarxa" (també marcat en roig) i també en roig està assenyalat on es pot veure la resposta de la petició GET HTTP (200 OK).

També veiem les capçaleres que s'inclouen en la petició (mètode GET) i en la resposta a aquesta petició.

1.4.1 Afegint capçaleres

A més d'haver mirat els **logs**, demostrarem encara de forma més clara que la petició està passant pel **proxy invers** i que està arribant al servidor web i que torna pel mateix camí.

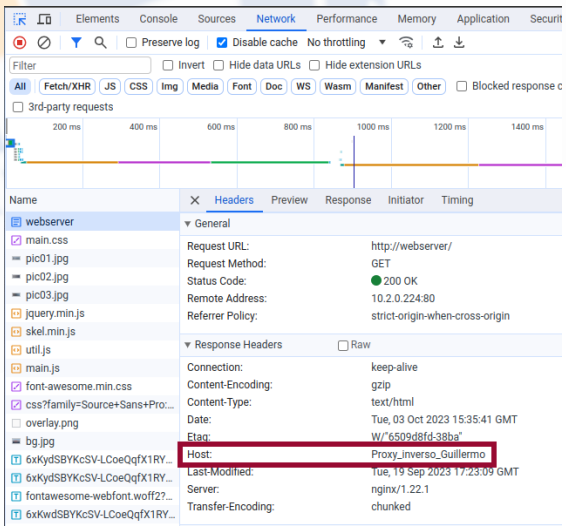
Si recordeu de teoria, el servidor web és capaç d'afegir capçaleres en les respostes a les peticions.

Així doncs, configurarem tant el **proxy invers** com el servidor web perquè afigen cadascun la capçalera "Host" que també vam veure en teoria.

Per a afegir capçaleres, en l'arxiu de configuració del lloc web hem d'afegir dins del bloc `location / { ...}` hem d'afegir la directiva:

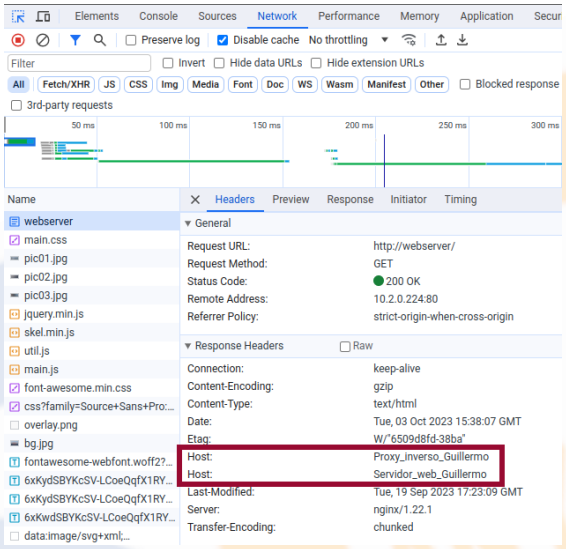
```
1  add_header Host nom_del_host;
```

1. Afegirem primer aquesta capçalera únicament en l'arxiu de configuració del lloc web del **proxy invers**. El Nom_del_host serà **Proxy_invers_vostre_nom**.
2. Reiniciem Nginx.
3. Comprovem que podem accedir al lloc web sense problemes.
4. Amb les eines de desenvolupador comprovem que la petició ha passat pel **proxy invers** que ha afegit la capçalera en la resposta:

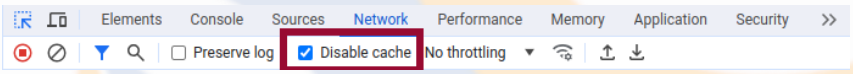


5. Fem el propi amb el servidor web. Ficant com a Nom_del_host serà **Servidor_web_vostre_nom**.

   Si tot està configurat correctament, en examinar les peticions i respostes, us apareixeran les dues capçaleres que han inclòs en la resposta tant el **proxy invers** com el servidor web.



És molt important que per a realitzar aquestes comprovacions tingueu marcat la casella de selecció **Desactivar cau** o en una finestra privada del navegador.



Si no marcau això, la pàgina es guardarà en la memòria cau del navegador i no estareu rebent la resposta del servidor sinó de la cau del navegador, la qual cosa pot donar lloc a resultats erronis.