

Plusieurs serveurs Web avec une seule adresse IP publique : mise en place d'un reverse-proxy sous Ubuntu.

Nous disposons que d'une seule adresse IP publique et nous voulons permettre l'accès à nos différents serveurs web locaux depuis le réseau public selon le nom de domaine utilisé. Après avoir exposé les différentes solutions envisagées, cet article explique comment nous avons mis en place un reverse-proxy en recyclant un vieux PC, avec Linux (Ubuntu) et NGinx.

Le problème	1
Solutions	2
Reverse proxy, quesako ?	2
Solutions techniques	3
Dans la pratique	3
Installation Ubuntu	3
Adresse IP	3
Administration depuis un poste Windows	4
Installation et configuration NGinx	4
Sécurisation du serveur	6
Références	7

Le problème

Lorsque nous démarrons un nouveau projet pour un client, nous mettons aussitôt en place une plateforme de recette sur un de nos serveurs interne. Cette plateforme peut comprendre une ou plusieurs machines de façon à reproduire la plateforme de production. Dans la pratique ces machines sont souvent virtualisées mais ça ne change pas le problème : elles ont chacune une adresse IP locale différente et nous voulons les rendre visibles de l'extérieur pour que nos clients puissent tester les développements, mais nous ne disposons que d'une seule adresse IP publique. Comment associer plusieurs serveurs Web avec une seule adresse IP ?

Solutions

Il existe plusieurs solutions possibles : on pourrait utiliser des ports spécifiques pour chaque serveur (:8081, :8082...) mais ce n'est pas très élégant. Nous souhaitons donc associer des sous domaines à chaque machine. Nous avons donc créé autant de sous domaines que de projets : projet1.dev.clt-services.com, projet2.dev.clt-services.com.

Ce qu'il nous faut c'est le moyen de rediriger les flux http entrants vers une machine (virtuelle ou physique) ou une autre en fonction du sous domaine.

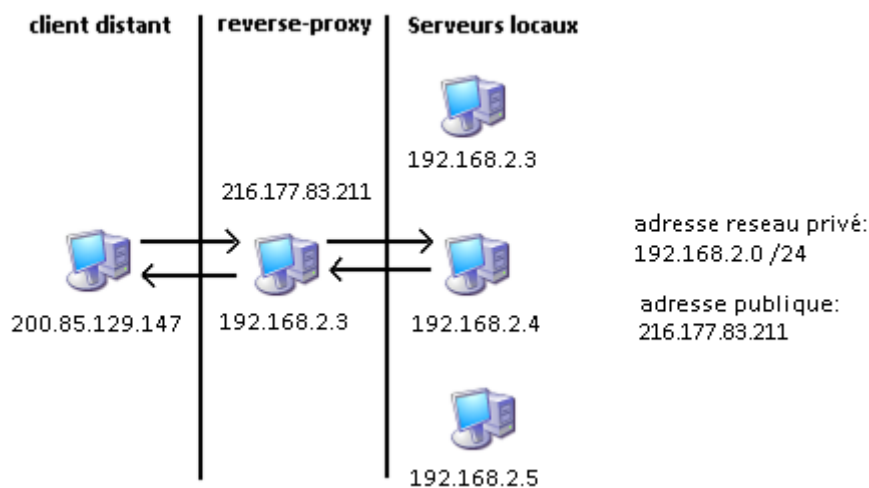
Deux possibilités :

- acheter un routeur pro qui offre cette fonction, mais c'est cher
- recycler un vieux pc qui ne sert plus en reverse proxy, c'est ce que nous allons faire.

Reverse proxy, quesako ?

Le reverse proxy comme son nom l'indique est l'inverse du proxy. Son rôle est de permettre l'accès d'un client distant vers un ou plusieurs serveurs locaux. L'intérêt principal du serveur reverse proxy est de pouvoir accéder via une seule adresse publique aux adresses privées du domaine local en se servant de l'entête http des paquets envoyés par le client distant. Ce système servira à rediriger l'url demandé vers le bon serveur web.

Fonctionnement du reverse proxy.



L'autre avantage du reverse proxy est de protéger les serveurs locaux des attaques externes ce qui accroît la sécurité du système d'information interne. En effet la machine qui va être exposée à l'extérieur est dédiée à cette seule fonction de routage et offre une surface d'exposition beaucoup plus réduite car elle a beaucoup moins de logiciels installés. Les reverse-proxy offrent aussi d'autres fonctions, comme un cache de requête ou des services de fail-over ou d'équilibrage de charge entre les serveurs, mais que nous n'utiliserons pas ici.

Le reverse-proxy prend en charge plusieurs protocoles dont http et SSL.

Solutions techniques

Le reverse proxy le plus fréquemment cité dans les articles web c'est Apache qui permet aussi cette option en plus de faire serveur web.

Pour éviter de mal configurer Apache et de laisser des trous de sécurité, et aussi parce qu'on a juste besoin de router le trafic entrant, pas d'héberger un site ou de faire tourner php sur ce pc, on a choisit d'utiliser plutôt un logiciel plus light, spécialisé dans la fonction de reverse proxy.

On en a trouvé 3 : Squid, Pound et nGinx. On a retenu ce dernier qui est à la fois simple à utiliser et qui a fait ces preuves sur de gros sites.

Pour aller plus loin : voir le site Nginx (<http://nginx.net>) et la doc (en anglais) : <http://wiki.codemongers.com/Main>

Pour le PC, on a retenu la distribution Ubuntu serveur (Feisty Fawn) mais les étapes de config seraient à peu près les mêmes sur d'autres distrib.

Dans la pratique

Installation Ubuntu

Pour simplifier l'installation du reverse proxy et son administration il est préférable d'installer un système d'exploitation de type GNU Linux qui possède un plus léger, ce qui allège considérablement la charge de calcul dédié au système. L'installation d'Ubuntu peut se faire sur un ordinateur ancien ou bon marché ayant une capacité de mémoire vive minimale égale 64 mo un disque dur de capacité suffisante pour installer le système (comptez plus de 2Go). Rendez-vous sur <http://doc.ubuntu-fr.org> plus d'informations sur la configuration minimale et l'installation du système d'exploitation.

Note pour ceux qui ne connaissent rien à Linux ; apt-get et sudo

Adresse IP

Pour le bon fonctionnement de votre système reverse-proxy il vous faudra obligatoirement assigner une adresse IP fixe à votre serveur. Vous pouvez néanmoins choisir le réseau local auquel il appartiendra tout en assurant la communication entre le serveur reverse-proxy et les serveurs web locaux (s'ils appartiennent à un autre sous-réseau). Il est conseillé d'utiliser une connexion filaire pour relier votre serveur proxy à vos serveurs web.

Dans cet exemple les serveurs web et le serveur reverse-proxy sont sur la même segmentation réseau.

Informations du réseau :

Adresse réseau : 192.168.2.0

Masque de sous réseau : 255.255.255.0

Passerelle par défaut : 192.168.2.1

Ouvrez le terminal et éditez le fichier de configuration d'interfaces réseau.

```
sudo vim /etc/network/interfaces
```

Repérez la ligne correspondant à la carte réseau reliait au réseau local. Si une adresse IP dynamique a été attribuée à votre carte réseau la ligne devrait normalement ressembler à ceci.

```
Iface eth0 inet dhcp
```

eth0 est le nom de la carte réseau de notre exemple.

Modifier la ligne pour quelle telle que suit :

```
Iface eth0 inet static 192.168.2.2 #Adresse IP du serveur reverse-proxy.  
    netmask 255.255.255.0 #Masque de sous-réseau.  
    gateway 192.168.2.1    #Passerelle par défaut.
```

Enregistrer les modifications apportées au fichier. Sous vim appuyez sur la touche « Echap » puis entrer « :wq ». Redémarrer ensuite votre carte réseau en entrant la commande :

```
sudo /etc/init.d/networking restart
```

Si la commande ne marche pas redémarrez votre ordinateur.

Administration depuis un poste Windows

SSH + Putty pour faciliter l'admin depuis un autre poste (pratique par exemple pour copier/coller les instructions qui suivent)

Maintenant vous pouvez accéder au shell de votre serveur reverse-proxy depuis Windows. Putty est un freeware qui vous permettra d'accéder au terminal de votre serveur depuis Windows, il est disponible gratuitement sur internet et est très simple d'installation et d'utilisation. A l'ouverture de Putty spécifiez l'adresse réseau du serveur reverse proxy et la connexion en mode ssh.

Note sur la sécurité : Pour faciliter la mise au point de la configuration, on va installer le reverse proxy en laissant toutes les connexions ouvertes ; une fois que notre serveur sera réglé, on le sécurisera en fermant les ports qui ne sont pas nécessaires et en restreignant les adresses ip. Nous verrons ça dans la dernière partie de cet article.

Installation et configuration NGinx

Pour installer correctement NGINX suivez les étapes suivantes :

Etape 3 : Télécharger la dernière version de NGYNX avec la commande d'installation « apt-get » puis patientez jusqu'à la fin de l'installation.

```
[Ubuntu@ubuntu ~]$ sudo apt-get install nginx
```

La plupart des applications et en particulier les applications client-serveur fonctionnant sur un système GNU\ Linux possèdent un ou plusieurs fichiers de configuration. En règle générale, le nom du fichier de configuration principale se termine par un « .conf », c'est le cas pour le fichier de configuration de NGINX.

Ouvrez maintenant le fichier de configuration avec l'éditeur choisi en mode administrateur.

```
[Ubuntu@ubuntu ~]$ sudo vim /etc/nginx/nginx.conf
```

***Attention** : Vous devez disposer des droits administrateurs pour éditer le fichier de configuration des périphériques réseau ou du moins le modifier. Pour cela la commande sudo doit précéder la commande ouvrant le fichier avec l'éditeur vim. Vous pouvez néanmoins ouvrir le fichier avec votre éditeur préféré.*

Repérer les lignes suivantes :

```
server {
    listen      80;
    server_name localhost;

    access_log  /var/log/nginx/localhost.access.log;

    location / {
        root    /var/www/nginx-default;
        index   index.html index.htm;
    }
}
```

listen : Port d'écoute du protocole HTTP.

serverName : URL du serveur web.

access_log : Fichier de journalisation du serveur.

root : chemin vers le répertoire web (option à choisir si le site est interne au serveur reverse-proxy).

Index, index.html et index.htm sont les pages à charger par défaut du site web.

Pour arriver à configurer correctement votre reverse-proxy vous devez définir un serveur avec comme suit :

```
server
{
    listen [Port d'ecoute];
    server_name [url du serveur web];
    access_log [chemin du fichier de configuration]

    location \{
        proxy_pass [adresse IP du serveur local a atteindre]
    }
}
```

Notez que la propriété « proxy_pass » pointe vers l'adresse IP du serveur web local que l'on veut atteindre avec l'url spécifié sur le port donné. De plus le chemin du fichier de journalisation a été configuré par défaut lors de l'installation de NGINX sur votre machine, mais il vous ait normalement possible de redéfinir le chemin au fichier .log. Le numéro de port d'écoute est généralement 80.

Les blocs server et location possèdent bien évidemment d'autre paramètres que vous pouvez retrouver sur le site web d'Igor Sysoev, l'auteur de NGINX (<http://nginx.net>).

Sécurisation du serveur

Lorsque vous connectez votre serveur reverse-proxy a internet certaines failles de sécurité peuvent apparaitre si le port SSH est ouvert. En effet, par default l'accès au Shell de votre système d'exploitation est accessible à distance ce qui laisse la possibilité a certains internautes malveillants de prendre le contrôle de votre serveur s'ils prennent connaissance des informations utilisateur de votre machine.

Certaines distributions linux possèdent **Netfilter** un module qui fournit au système des fonctions de pare-feu. **IPtables** est la commande qui permet de configurer **Netfilter**.

Suivez les étapes suivantes pour configurer le pare-feu de votre reverse-proxy dans votre réseau local :

Restriction des connexions entrantes

Pour éviter cela il vous faut à nouveau configurer IPTABLES pour contrôler l'ouverture du port SSH de votre serveur reverse-proxy.

- Vérifier tout d'abord que l'INPUT Policy est a DROP pour interdire toutes les connexions entrantes non spécifié.

Etape 1 : Ouvrez le terminal de votre distribution linux et faites en sorte d'interdire toutes les connexions entrantes.

```
sudo iptables -P INPUT DROP
```

Etapes 4 : Autorisez les paquets entrant des connexions déjà établie.

```
sudo iptables -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Ouvrir le trafic HTTP depuis l'extérieur

Etape 3 : Autorisez les paquets entrant de n'importe quelle source utilisant le port 80 (HTTP) ou le cas échéant du port du protocole choisi.

```
sudo iptables -A INPUT -p tcp -i eth0 --dport http -j ACCEPT
```

Ouvrir le trafic SSH depuis les postes autorisés à administrer

- Autoriser la connexion SSH à l'adresse IP du poste que vous avez choisi, cela permettra à l'ordinateur autorisé de se connecter au Shell de votre reverse proxy.

```
sudo iptables -A INPUT -i eth0 -p tcp --dport ssh -s[ADRESSE IP] -j ACCEPT
```

- Il est très important de définir une règle anti bruteforce SSH. Ces deux règles ci-dessous déconnectent le client au bout de trois tentatives de connexions manquées ou après 300 secondes d'inactivité.

```
-I INPUT -p tcp --dport 22 -i eth0 -m state --state NEW -m recent --set  
-I INPUT -p tcp --dport 22 -i eth0 -m state --state NEW -m recent --  
update --seconds 300 --hitcount 3 -j DROP
```

Références

Pour approfondir vos connaissances sur le serveur proxy rendez-vous sur les sites web suivants :

<http://wiki.codemongers.com/Main>

<http://www.hsc.fr/ressources/breves/pourquoi-relais-inverse.html.en>