

# INSTALLATION D'UN SERVEUR REVERSE-PROXY SUR Debian 12 EN UTILISANT Nginx ET haproxy

Un **reverse proxy** est un serveur à l'avant des services Web et qui transmet les demandes des clients à ces serveurs web. Il est utilisé pour distribuer la charge d'une manière qui maximise l'expérience de l'utilisateur.

tout au long de cette documentation, nous installerons un service web, un logiciel d'équilibrage de charge et .

nous utiliserons comme serveur web **Nginx** et comme logiciel d'équilibrage de charge **Haproxy**.

**Prérequis:**

**Création de trois machines virtuelles sous Debian 12 : VM-1 VM-2 VM-3**

**Installation des serveurs web sur VM-2 et VM-3**

**Installation de Haproxy sur VM-1**

## Installation du serveur web Nginx

**Etape 1 :** Installation des serveurs web sur VM-2 VM-3

Avant tout, il faut installer les différentes mise à jour

Sudo apt update && sudo apt upgrade

Installation des paquets de **Nginx**

sudo apt install nginx -y

Après pour l'installation de **Nginx** il va falloir démarrer le service

sudo systemctl enable nginx

Après il va falloir vérifier le statut du service, s'il fonctionne correctement

Sudo systemctl status nginx

## Modification des pages web

## **Etape 2 : Modification des pages d'accueil des serveurs web sur VM-2 et VM-3**

le site web du service web se trouve dans le dossier **/var/www/html/** cd

**/var/www/html/**

Nous listons les fichiers et supprimons le fichier **index.html**

**ls et sudo rm index.html**

Nous le rééditons et créons notre fichier html.

**sudo nano index.html**

Pour être sûr que nos modifications sont prises en charge, nous redémarrons le serveurs

**systemctl restart nginx**

## **Installation du logiciel d'équilibrage**

### **Etape 3 : Installation du logiciel d'équilibrage **haproxy** sur VM-1**

**sudo apt update && sudo apt upgrade**

**sudo apt install haproxy -y**

## **Configuration de haproxy**

### **Etape 4 : Configuration de **haproxy****

**sudo nano /etc/haproxy/haproxy.cfg**

#### **Note**

```
# configuration personnalisé
Frontend front_webservers
bind *: 80
default_backend backend_webservers
option forwardfor
backend backend_webservers
```

```
balance roundrobin  
server srv-web-1 adresse ip_web_serveur_1:80 check  
server srv-web-2 adresse_ip_web_serveur_2:80 check
```

Pour être sur que nos modifications sont prises en compte, nous redémarrons le service

```
`systemctl restart haproxy`
```

**ici nous utilisons la méthode de répartition Round Robin, il existe cependant d'autres méthodes comme : Least connections qui est une méthode qui pourra repartir le flux vers le serveur ayant le moins de connexion active ou encore le least Bandwidht qui attribuera les nouvelles requêtes sur le serveur consommant le moins de bande passante.**

## Eventuels problèmes

### Si lors des commandes sudo il vous ai renvoyé

user n'est pas dans le groupe sudoers

alors il va falloir ajouter cet utilisateur au groupe sudoers. Il faut :

```
Su -  
nano /etc/sudoers  
user ALL=(ALL:ALL) ALL
```

### Si lors du démarrage du serveur Nginx est impossible

vérifier qu'il n'y a pas d'autre serveur web avec un service actif. Si c'est le cas, il faudra l'arrêter pour faire tourner notre serveur web

```
`sudo service 'serveur web' stop`  
`sudo service nginx start`
```

### les commandes

sudo apt update && sudo apt upgrade  
doivent être utilisées qu'une seule fois