



# HAUTE DISPONIBILITÉ ET ÉQUILIBRAGE DE CHARGE

heartbeat et haproxy

Préparé par  
**COSTA DUARTE Diogo**

**BTS SIO SISR**

CFA INSTA - 2023

# Haute disponibilité avec HeartBeat (PCA : Plan de continuité d'activité)

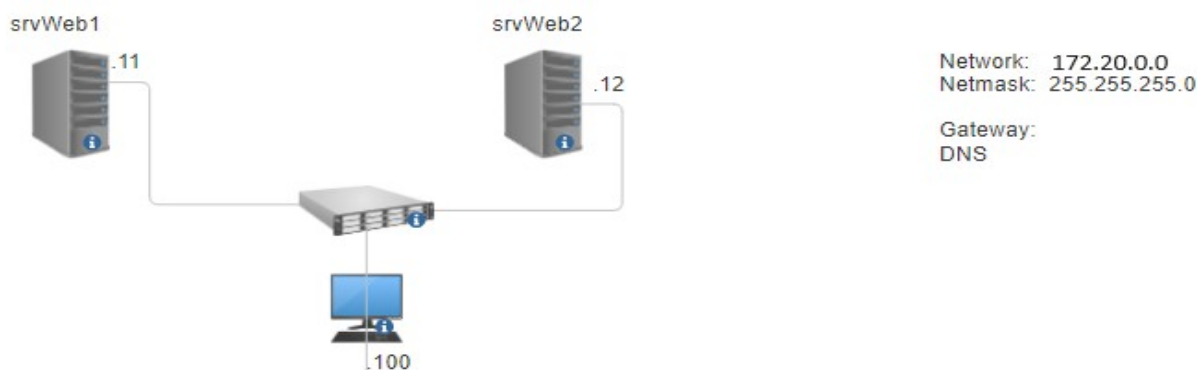
## Table des matières :

1. Introduction au Réseau
  - a) Infrastructure du Réseau
  - b) Les machines du réseau
2. Installation et configuration du 1<sup>er</sup> Serveur Web (srvWeb1)
  - a) Configuration du Serveur srvWeb1
  - b) Installation d'Apache 2
  - c) Mise en place du Site Web « The Grill-Master»
3. Installation et configuration du 2<sup>nd</sup> Serveur Web (srvWeb2)
  - a) Configuration du Serveur srvWeb2
  - b) Installation d'Apache 2
  - c) Mise en place du Site Web « The Grill-Master»
4. 1<sup>er</sup> Teste de connectivité entre les Machines
  - a) Ping entre les machines srvWeb1, srvWeb2 et client Windows
  - b) Accessibilité sur les site web des 2 serveurs
5. Configuration de HearBeat sur le 1<sup>er</sup> Serveur Web (srvWeb1)
  - a) Configuration du fichier Host et test de connectivité
  - b) Installation de HearBeat
  - c) Configuration de HearBeat
6. Configuration de HearBeat sur le 2<sup>nd</sup> Serveur Web (srvWeb2)
  - a) Configuration du fichier Host et test de connectivité
  - b) Installation de HearBeat
  - c) Configuration de HearBeat
7. Validation du fonctionnement de la Haute Disponibilité.

## 1. Introduction au Réseau

### a. Infrastructure du Réseau

Le réseau sera composé de 3 machines différentes. Chaque machines aura son rôle à jouer dans le réseau, nous aurons un cluster Web composé de 2 nœuds permettant un équilibrage de charge et ainsi une Haute disponibilité. Une troisième machine qui sera un Client Windows 10 aura pour but le test du bon fonctionnement de l'équilibrage de charge du Site Web.



Le réseau aura donc cette forme.

## b. Les machines du réseau

Les 3 machines de notre réseau auront les configuration suivantes :

Serveur Web 1 :

- hostname : srvWeb1
- Interface IP: 172.20.0.11 (LAN segment)
- netmask : 255.255.255.0

Serveur Web 2 :

- hostname : srvWeb2
- Interface IP: 172.20.0.12 (LAN segment)
- netmask : 255.255.255.0

Client :

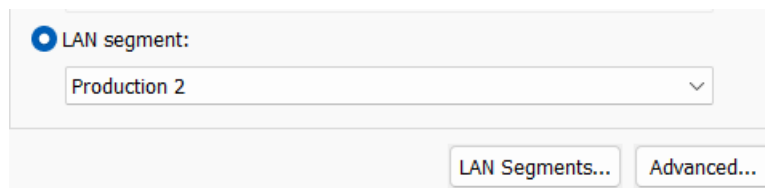
- hostname : clientw10
- Interface IP: 172.20.0.100 (LAN segment)
- Netmask : 255.255.255.0

## 2. Installation et configuration du 1<sup>er</sup> Serveur Web (srvWeb1)

### a. Configuration du Serveur srvWeb1

Ajout de la carte réseau dans le Lan Segment « Production 2 »

(**attention** une fois que la machine est sur le Lan segment celle-ci n'aura plus accès à internet donc plus accès aux mises à jours.). Pour cela nous vous conseillons l'installation en avance des packages : unzip, apache2, heartbeat



Changement du hostname de la machine

- Sur le terminal Linux entrez la commande suivante :
  - hostnamectl set-hostname srvWeb1

```
root@debian:~# hostnamectl set-hostname _
```

Configuration de l'interface réseau du Serveur srvWeb1

- Sur le terminal Linux entrez la commande suivante :
  - nano /etc/network/interfaces

```
root@srvWeb1:~# nano /etc/network/interfaces
```

- Une fois dans le fichier de configuration des interfaces, modifiez le avec les paramètres suivants :

```
# The primary network interface
allow-hotplug ens33
iface ens33 inet static
address 172.20.0.11
netmask 255.255.255.0
```

## b. Installation du Service Apache 2

Mise a jours de la liste des package disponible

- Sur le terminal Linux entrez la commande suivante :
  - apt-get update

```
root@srvWeb1:~# apt-get update
Atteint :1 http://security.debian.org/debian-security bullseye-security InRelease
Atteint :2 http://deb.debian.org/debian bullseye InRelease
Atteint :3 http://deb.debian.org/debian bullseye-updates InRelease
Lecture des listes de paquets... Fait
root@srvWeb1:~#
```

Mise a jours de tous les packages déjà présent sur la machine :

- Sur le terminal Linux entrez la commande suivante :
  - apt-get upgrade

```
root@srvWeb1:~# apt-get upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
Les paquets suivants ont été conservés :
 linux-image-amd64
0 mis à jour, 0 nouvellement installés, 0 à enlever et 1 non mis à jour.
root@srvWeb1:~#
```

Installation du package apache2

- Sur le terminal Linux entrez la commande suivante :
  - apt-get install apache2 -y

```
root@srvWeb1:~# apt-get install apache2 -y
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
apache2 est déjà la version la plus récente (2.4.56-1~deb11u2).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 1 non mis à jour.
root@srvWeb1:~# _
```

### c. Mise en place du Site Web « The Grill-Master»

Suppression de la page html par défaut de Apache 2

- Sur le terminal linux entrez la commande suivante :
  - `rm /var/www/html/index.html`

```
root@srvWeb1:~# rm /var/www/html/index.html
root@srvWeb1:~#
```

Rendez-vous dans le dossier hébergeant les pages html :

- Sur le terminal linux entrez la commande suivante :
  - `cd /var/www/html`

```
root@srvWeb1:~# cd /var/www/html/
root@srvWeb1:/var/www/html#
```

Installez le dossier compresseur thgrill possédant les fichier du site « The Grille-Master » depuis github :

- Sur le terminal linux entrez la commande suivante :
  - `wget https://github.com/technext/thegrill/archive/master.zip`

```
root@srvWeb1:/var/www/html# wget https://github.com/technext/thegrill/archive/master.zip
--2023-11-17 20:12:51-- https://github.com/technext/thegrill/archive/master.zip
Résolution de github.com (github.com)... 140.82.121.3
Connexion à github.com (github.com)|140.82.121.3|:443... connecté.
requête HTTP transmise, en attente de la réponse... 302 Found
Emplacement : https://codeload.github.com/technext/thegrill/zip/refs/heads/master [suivant]
--2023-11-17 20:12:51-- https://codeload.github.com/technext/thegrill/zip/refs/heads/master
Résolution de codeload.github.com (codeload.github.com)... 140.82.121.9
Connexion à codeload.github.com (codeload.github.com)|140.82.121.9|:443... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : non indiqué [application/zip]
Sauvegarde en : « master.zip »

master.zip          [  <=>          ]  4,32M  7,13MB/s  ds 0,6s

2023-11-17 20:12:52 (7,13 MB/s) - « master.zip » sauvegardé [4534581]

root@srvWeb1:/var/www/html#
```

Installer le package unzip :

- Sur le terminal linux entrez la commande suivante :
  - `apt-get install unzip`

```
root@srvWeb1:/var/www/html# apt-get install unzip
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Paquets suggérés :
  zip
Les NOUVEAUX paquets suivants seront installés :
  unzip
0 mis à jour, 1 nouvellement installés, 0 à enlever et 1 non mis à jour.
Il est nécessaire de prendre 172 ko dans les archives.
Après cette opération, 393 ko d'espace disque supplémentaires seront utilisés.
Réception de :1 http://deb.debian.org/debian bullseye/main amd64 unzip amd64 6.0-26+deb11u1 [172 kB]
172 ko réceptionnés en 0s (1 720 ko/s)
Sélection du paquet unzip précédemment désélectionné.
(Lecture de la base de données... 34409 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de ../unzip_6.0-26+deb11u1_amd64.deb ...
Dépaquetage de unzip (6.0-26+deb11u1) ...
Paramétrage de unzip (6.0-26+deb11u1) ...
Traitement des actions différées (« triggers ») pour mailcap (3.69) ...
Traitement des actions différées (« triggers ») pour man-db (2.9.4-2) ...
root@srvWeb1:/var/www/html# _
```

## Décompresser le fichier zip

- Sur le terminal linux entrez la commande suivante :
  - unzip master.zip

```
root@srvWeb1:/var/www/html# unzip master.zip _
```

Nous avons maintenant 2 fichiers : Master.zip qui est le fichier zip de base et thegrill-master qui le notre dossier contenant le site Web

```
root@srvWeb1:/var/www/html# ls
master.zip  thegrill-master
root@srvWeb1:/var/www/html# rm master.zip _
```

## Changement du doucement de configuration de Apache 2 afin de donner l'accessibilité au site

- Sur le terminal linux entrez la commande suivante :
  - nano /etc/apache2/sites-available/000-default.conf

```
root@srvWeb1:~# nano /etc/apache2/sites-available/000-default.conf _
```

Dans le fichiers de configuration, ajoutez le nom du dossier "thegrill-master" dernier /var/www/html/

```
GNU nano 5.4 /etc/apache2/sites-available/000-default.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/thegrill-master

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

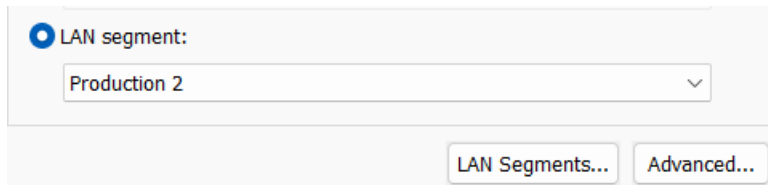
[ Lecture de 31 lignes ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper   ^T Exécuter  ^C EmplacementM-U Annuler
^X Quitter   ^R Lire fich.^N Remplacer ^U Coller   ^J Justifier ^_ Aller ligneM-E Refaire
```

## 7. Installation et configuration du 2nd Serveur Web (srvWeb2)

### a. Configuration du Serveur srvWeb2

Ajout de la carte réseau dans le Lan Segment « Production 2 »

(attention une fois que la machine est sur le Lan segment celle-ci n'aura plus accès à internet donc plus accès aux mises à jours.)



Changement du hostname de la machine

- Sur le terminal Linux entrez la commande suivante :
  - `hostnamectl set-hostname srvWeb1`

```
root@debian:~# hostnamectl set-hostname srvWeb2_
```

Configuration de l'adresse IP du Serveur srvWeb1

- Sur le terminal Linux entrez la commande suivante :
  - `nano /etc/network/interfaces`

```
root@srvWeb2:~# nano /etc/network/interfaces
```

- Une fois dans le fichier de configuration des interfaces, modifiez le avec les paramètres suivants :

```
# The primary network interface
allow-hotplug ens33
iface ens33 inet static
address 172.20.0.12
netmask 255.255.255.0
```

## b. Installation du Service Apache 2

Mise à jours de la liste des package disponible

- Sur le terminal Linux entrez la commande suivante :
  - apt-get update

```
root@srvWeb2:~# apt-get update
Atteint :1 http://security.debian.org/debian-security bullseye-security InRelease
Atteint :2 http://deb.debian.org/debian bullseye InRelease
Atteint :3 http://deb.debian.org/debian bullseye-updates InRelease
Lecture des listes de paquets... Fait
root@srvWeb2:~# _
```

Mise à jours des package déjà installés sur la machine

- Sur le terminal Linux entrez la commande suivante :
  - apt-get upgrade

```
root@srvWeb2:~# apt-get upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
Les paquets suivants ont été conservés :
 linux-image-amd64
0 mis à jour, 0 nouvellement installés, 0 à enlever et 1 non mis à jour.
root@srvWeb2:~# _
```

Installation du package apache2

- Sur le terminal Linux entrez la commande suivante :
  - apt-get install apache2 -y

```
root@srvWeb2:~# apt-get install apache2 -y
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
apache2 est déjà la version la plus récente (2.4.56-1~deb11u2).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 1 non mis à jour.
root@srvWeb2:~#
```



### c. Mise en place du Site Web « The Grill-Master »

Suppression de la page html par défaut de Apache 2

- Sur le terminal linux entrez la commande suivante :
  - `rm /var/www/html/index.html`

```
root@srvWeb2:~# rm /var/www/html/index.html
```

Rendez-vous dans le dossier hébergeant les pages html :

- Sur le terminal linux entrez la commande suivante :
  - `cd /var/www/html`

```
root@srvWeb2:~# cd /var/www/html/  
root@srvWeb2:/var/www/html#
```

Installez le dossier compresseur thgrill possédant les fichier du site « The Grille-Master » depuis github :

- Sur le terminal linux entrez la commande suivante :
  - `wget https://github.com/technext/thegrill/archive/master.zip`

```
root@srvWeb2:~# cd /var/www/html/  
root@srvWeb2:/var/www/html# wget https://github.com/technext/thegrill/archive/master.zip  
--2023-11-18 19:03:05-- https://github.com/technext/thegrill/archive/master.zip  
Résolution de github.com (github.com)... 140.82.121.4  
Connexion à github.com (github.com)|140.82.121.4|:443... connecté.  
requête HTTP transmise, en attente de la réponse... 302 Found  
Emplacement : https://codeload.github.com/technext/thegrill/zip/refs/heads/master [suivant]  
--2023-11-18 19:03:05-- https://codeload.github.com/technext/thegrill/zip/refs/heads/master  
Résolution de codeload.github.com (codeload.github.com)... 140.82.121.10  
Connexion à codeload.github.com (codeload.github.com)|140.82.121.10|:443... connecté.  
requête HTTP transmise, en attente de la réponse... 200 OK  
Taille : non indiqué [application/zip]  
Sauvegarde en : « master.zip »  
  
master.zip [ <=> ] 4,32M 14,8MB/s ds 0,3s  
  
2023-11-18 19:03:06 (14,8 MB/s) - « master.zip » sauvegardé [4534581]  
  
root@srvWeb2:/var/www/html#
```

Installer le package unzip :

- Sur le terminal linux entrez la commande suivante :
  - `apt-get install unzip`

```
root@srvWeb2:/var/www/html# apt-get install unzip  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait  
Lecture des informations d'état... Fait  
Paquets suggérés :  
  zip  
Les NOUVEAUX paquets suivants seront installés :  
  unzip  
0 mis à jour, 1 nouvellement installés, 0 à enlever et 1 non mis à jour.  
Il est nécessaire de prendre 172 ko dans les archives.  
Après cette opération, 393 ko d'espace disque supplémentaires seront utilisés.  
Réception de :1 http://deb.debian.org/debian bullseye/main amd64 unzip amd64 6.0-26+deb11u1 [172 kB]  
172 ko réceptionnés en 0s (2 082 ko/s)  
isélection du paquet unzip précédemment désélectionné.  
(Lecture de la base de données... 34409 fichiers et répertoires déjà installés.)  
Préparation du dépaquetage de .../unzip_6.0-26+deb11u1_amd64.deb ...  
Dépaquetage de unzip (6.0-26+deb11u1) ...  
Paramétrage de unzip (6.0-26+deb11u1) ...  
Traitement des actions différées (« triggers ») pour mailcap (3.69) ...  
Traitement des actions différées (« triggers ») pour man-db (2.9.4-2) ...  
root@srvWeb2:/var/www/html#
```

## Décompresser le fichier zip

- Sur le terminal linux entrez la commande suivante :
  - unzip master.zip

```
root@srvWeb2:/var/www/html# unzip master.zip
```

Nous avons maintenant 2 fichiers : Master.zip qui est le fichier zip de base et thegrill-master qui le notre dossier contenant le site Web

```
root@srvWeb2:/var/www/html# ls
master.zip  thegrill-master
root@srvWeb2:/var/www/html#
```

## Changement du doucement de configuration de Apache 2 afin de donner l'accessibilité au site

- Sur le terminal linux entrez la commande suivante :
  - nano /etc/apache2/sites-available/000-default.conf

```
root@srvWeb2:~# nano /etc/apache2/sites-available/000-default.conf
```

Dans le fichiers de configuration, ajoutez le nom du dossier "thegrill-master" dernier /var/www/html/

```
GNU nano 5.4 /etc/apache2/sites-available/000-default.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/thegrill-master

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

[ Lecture de 31 lignes ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C EmplacementM-U Annuler
^X Quitter   ^R Lire fich.^N Remplacer  ^U Coller    ^J Justifier  ^_ Aller ligneM-E Refaire
```

## 4. 1<sup>er</sup> Teste de connectivité entre les Machines

### a. Ping entre les machines srvWeb1, srvWeb2 et client Windows

Ping du srvWeb1 (172.20.0.11) vers le srvWeb2 (172.20.0.12)

- Sur le terminal linux entrez la commande suivante :
  - ping 172.20.0.12

```
root@srvWeb1:~# ping 172.20.0.12
PING 172.20.0.12 (172.20.0.12) 56(84) bytes of data.
64 bytes from 172.20.0.12: icmp_seq=1 ttl=64 time=0.278 ms
64 bytes from 172.20.0.12: icmp_seq=2 ttl=64 time=0.504 ms
64 bytes from 172.20.0.12: icmp_seq=3 ttl=64 time=0.538 ms
64 bytes from 172.20.0.12: icmp_seq=4 ttl=64 time=0.283 ms
64 bytes from 172.20.0.12: icmp_seq=5 ttl=64 time=0.402 ms
^C
--- 172.20.0.12 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4104ms
rtt min/avg/max/mdev = 0.278/0.401/0.538/0.108 ms
root@srvWeb1:~#
```

Ping du srvWeb2 (172.20.0.12) vers le srvWeb1 (172.20.0.11)

- Sur le terminal linux entrez la commande suivante :
  - ping 172.20.0.11

```
root@srvWeb2:~# ping 172.20.0.11
PING 172.20.0.11 (172.20.0.11) 56(84) bytes of data.
64 bytes from 172.20.0.11: icmp_seq=1 ttl=64 time=0.371 ms
64 bytes from 172.20.0.11: icmp_seq=2 ttl=64 time=0.554 ms
64 bytes from 172.20.0.11: icmp_seq=3 ttl=64 time=0.196 ms
64 bytes from 172.20.0.11: icmp_seq=4 ttl=64 time=0.342 ms
64 bytes from 172.20.0.11: icmp_seq=5 ttl=64 time=0.401 ms
^C
--- 172.20.0.11 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4076ms
rtt min/avg/max/mdev = 0.196/0.372/0.554/0.114 ms
root@srvWeb2:~# _
```

Ping du srvWeb2 (172.20.0.12) et du srvWeb1 (172.20.0.11) vers le clientW10 (172.20.0.100)

- Sur le terminal linux entrez la commande suivante :
  - ping 172.20.0.100

```
root@srvWeb1:~# ping 172.20.0.100
PING 172.20.0.100 (172.20.0.100) 56(84) bytes of data.
64 bytes from 172.20.0.100: icmp_seq=1 ttl=128 time=0.531 ms
64 bytes from 172.20.0.100: icmp_seq=2 ttl=128 time=0.677 ms
^C
--- 172.20.0.100 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1021ms
rtt min/avg/max/mdev = 0.531/0.604/0.677/0.073 ms
root@srvWeb1:~#
```

```
root@srvWeb2:~# ping 172.20.0.100
PING 172.20.0.100 (172.20.0.100) 56(84) bytes of data.
64 bytes from 172.20.0.100: icmp_seq=1 ttl=128 time=0.713 ms
64 bytes from 172.20.0.100: icmp_seq=2 ttl=128 time=0.420 ms
^C
--- 172.20.0.100 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1032ms
rtt min/avg/max/mdev = 0.420/0.566/0.713/0.146 ms
root@srvWeb2:~#
```

Ping du clientW10 (172.20.0.100) vers le srvWeb1 (172.20.0.11) et le srvWeb2 (172.20.0.12)

- Sur le terminal client du clientW10 entrez la commande suivante :
  - ping 172.20.0.11
  - et
  - ping 172.20.0.12

```
C:\Users\Client>ping 172.20.0.11

Envoi d'une requête 'Ping' 172.20.0.11 avec 32 octets de données :
Réponse de 172.20.0.11 : octets=32 temps<1ms TTL=64
Réponse de 172.20.0.11 : octets=32 temps<1ms TTL=64

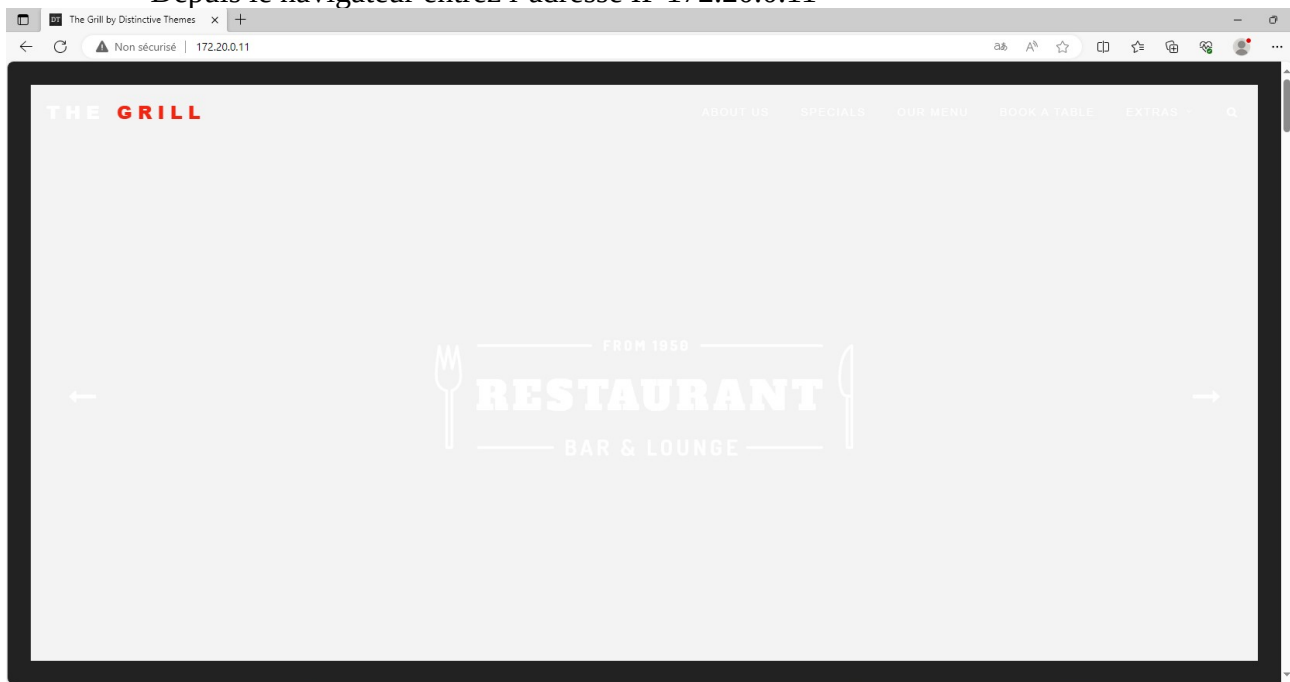
Statistiques Ping pour 172.20.0.11:
    Paquets : envoyés = 2, reçus = 2, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms
Ctrl+C
^C
C:\Users\Client>ping 172.20.0.12

Envoi d'une requête 'Ping' 172.20.0.12 avec 32 octets de données :
Réponse de 172.20.0.12 : octets=32 temps<1ms TTL=64
Réponse de 172.20.0.12 : octets=32 temps<1ms TTL=64

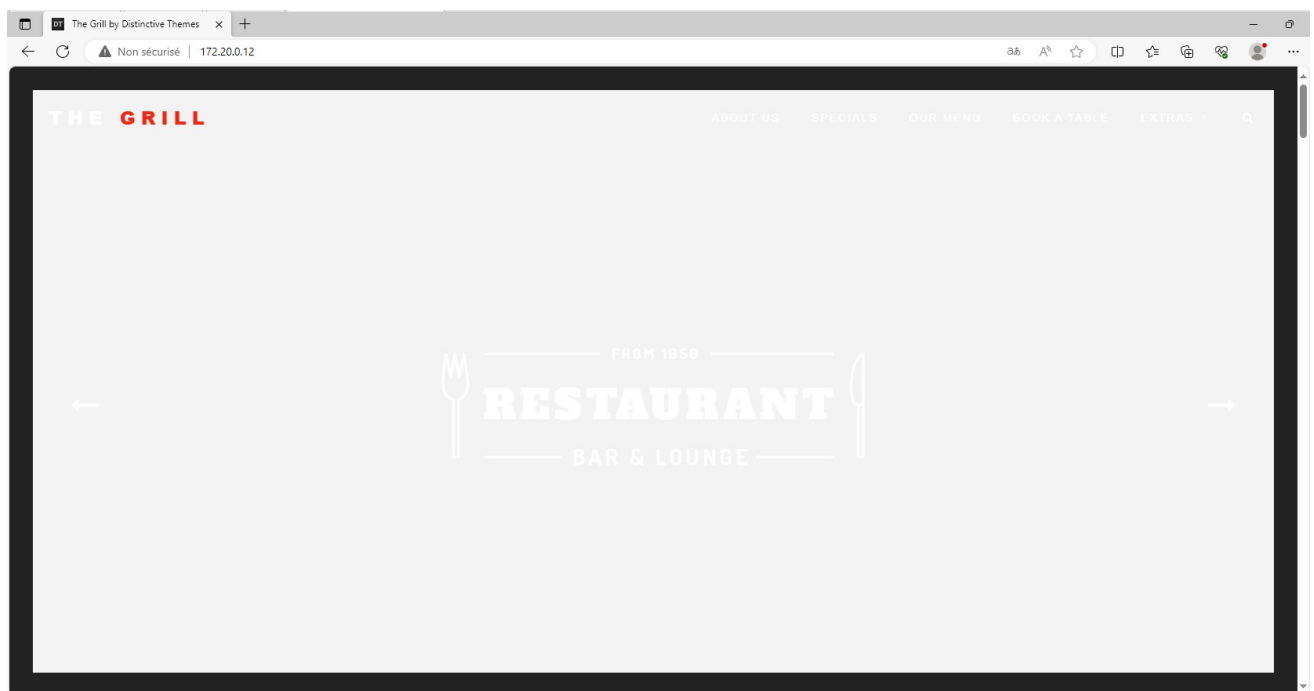
Statistiques Ping pour 172.20.0.12:
    Paquets : envoyés = 2, reçus = 2, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms
Ctrl+C
^C
C:\Users\Client>
```

## b. Accessibilité sur les site web des 2 serveurs

- Depuis le clientW10 faite une requête http vers le serveur srvWeb1
  - Depuis le navigateur entrez l'adresse IP 172.20.0.11



- Depuis le clientW10 faite une requête http vers le serveur srvWeb1
  - Depuis le navigateur entrez l'adresse IP 172.20.0.12



Maintenant que toutes les machines communiquent bien entre elle nous allons pouvoir mettre en place la haute Disponibilité.

## 5. Configuration du Service HearBeat sur le 1<sup>er</sup> Serveur Web (srvWeb1)

### a. Configuration du fichier Host et test de connectivité

Le fichier Host sur linux se trouve dans le répertoire suivant : /etc/hosts

Cette configuration permettra de faire en sorte que la machine puisse comprendre que une « une adresse ip » = « nom de machine » (le même principe qu'un DNS).

Configuration du fichier host

- Sur le terminal linux entrez la commande suivante :
  - nano /etc/hosts
- ajoutez-y les paramètres suivants :
  - 127.0.0.1 localhost
  - 172.20.0.11 srvWeb1
  - 172.20.0.12 srvWeb2

```
127.0.0.1 localhost
#127.0.1.1 debian
172.20.0.11 srvWeb1
172.20.0.12 srvWeb2
```

Test de connectivité en utilisant les nom host.

- Sur le terminal linux entrez la commande suivante :
  - ping srvWeb1
  - et
  - ping srvWeb2

```
root@srvWeb1:~# ping srvWeb1
PING srvWeb1 (172.20.0.11) 56(84) bytes of data.
64 bytes from srvWeb1 (172.20.0.11): icmp_seq=1 ttl=64 time=0.014 ms
^C
--- srvWeb1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.014/0.014/0.014/0.000 ms
root@srvWeb1:~# ping srvWeb2
PING srvWeb2 (172.20.0.12) 56(84) bytes of data.
64 bytes from srvWeb2 (172.20.0.12): icmp_seq=1 ttl=64 time=0.416 ms
64 bytes from srvWeb2 (172.20.0.12): icmp_seq=2 ttl=64 time=0.452 ms
^C
--- srvWeb2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1029ms
rtt min/avg/max/mdev = 0.416/0.434/0.452/0.018 ms
root@srvWeb1:~# _
```

## b. Installation du Service HeartBeat

Mise a jours de la liste des package disponible

- Sur le terminal Linux entrez la commande suivante :
  - apt-get update

```
root@srvWeb1:~# apt-get update
Atteint :1 http://security.debian.org/debian-security bullseye-security InRelease
Atteint :2 http://deb.debian.org/debian bullseye InRelease
Atteint :3 http://deb.debian.org/debian bullseye-updates InRelease
Lecture des listes de paquets... Fait
root@srvWeb1:~#
```

Mise a jours de tous les packages déjà présent sur la machine :

- Sur le terminal Linux entrez la commande suivante :
  - apt-get upgrade

```
root@srvWeb1:~# apt-get upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
Les paquets suivants ont été conservés :
 linux-image-amd64
0 mis à jour, 0 nouvellement installés, 0 à enlever et 1 non mis à jour.
root@srvWeb1:~#
```

Installation du package heartbeat

- Sur le terminal Linux entrez la commande suivante :
  - apt-get install heartbeat -y

```
root@srvWeb1:~# apt-get install heartbeat -y
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
heartbeat est déjà la version la plus récente (1:3.0.6-11+deb11u1).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 1 non mis à jour.
root@srvWeb1:~# _
```

### c. Configuration du Service HearBeat

Suite à l'installation de heartbeat, un lien symbolique /etc/heartbeat est créé, pointant vers /etc/ha.d

Vous pouvez le vérifier à l'aide de la commande suivante : `ls -l /etc/heartbeat`

Dans le dossier /etc/ha.d, nous allons créer 3 fichiers :

- ha.cf : fichier de configuration principal
- haresources : fichier de configuration des ressources.
- authkeys : information d'authentification

Création des 3 fichiers dans le dossier /etc/ha.d:

- Sur le terminal linux entrez la commande suivante :
  - `touch ha.cf`
  - `touch haresources`
  - `touch authkeys`

Configuration du premier fichier (ha.cf) :

- Sur le terminal linux entrez la commande suivante :
  - `nano /etc/ha.d/ha.cf`
- Entrez y les paramètres suivants :
  - # Les fichiers logs de heartbeat (événements relatifs à heartbeat)  
`logfile /var/log/heartbeat`  
`logfacility local0`

```
# Intervalle entre deux battements de cœur en seconde  
keepalive 5
```

```
# Temps nécessaire avant de considérer qu'un serveur (noeud) est mort (en seconde)  
deadtime 30
```

```
# Interface d'écoute  
bcast ens33
```

```
# liste des nœuds utilisées pour la Haute Disponibilité  
node srvWeb1 srvWeb2
```

```
# comportement si le nœud revient dans le réseau  
auto_failback on
```



Configuration du deuxième fichier (haresources) :

- Sur le terminal linux entrez la commande suivante :
  - nano /etc/ha.d/ haresources
- Entrez y les paramètres suivants :
  - # Active l'interface IP Virtuelle avec comme noeud principal srvWeb1  
# syntaxe :hostname IPaddr::IPvirtuelle/CIDR/interface service
  - #Hostname = Hostname du nœud principale (srvweb1)
  - #Ipaddr = L'ip du nœud principale(172.20.0.11)  
srvWeb1 IPaddr::172.20.0.10/24/ens33 apache2

Configuration du troisième fichier (authkeys) :

- Sur le terminal linux entrez la commande suivante :
  - nano /etc/ha.d/authkeys
- Entrez y les paramètres suivants :
  - auth 1  
1 md5 greta

modification les droits d'accessibilité au troisième fichier :

- Sur le terminal linux entrez la commande suivante :
- chmod 600 /etc/ha.d/authkeys

Désactivation du démarrage automatique d'apache car c'est heartbeat (haresources) qui va gérer le lancement de ce dernier

- Sur le terminal linux entrez la commande suivante :
  - update-rc.d apache2 remove

```
root@debian:/etc/ha.d# update-rc.d apache2 remove
```

Arrêtez apache2 :

- Sur le terminal linux entrez la commande suivante :
  - service heartbeat stop

```
root@debian:/etc/ha.d# service apache2 stop
```

Arrêtez heartbeat

- Sur le terminal linux entrez la commande suivante :
  - service heartbeat stop

```
root@debian:/etc/ha.d# service heartbeat stop_
```

## 6. Configuration du Service HearBeat sur le 2nd Serveur Web (srvWeb2)

### a. Configuration du fichier Host et test de connectivité

Le fichier Host sur linux se trouve dans le répertoire suivant : /etc/hosts

Cette configuration permettra de faire en sorte que la machine puisse comprendre que une « une adresse ip » = « nom de machine » (le même principe qu'un DNS).

Configuration du fichier host

- Sur le terminal linux entrez la commande suivante :
  - nano /etc/hosts
- ajoutez-y les paramètres suivants :
  - 127.0.0.1 localhost
  - 172.20.0.11 srvWeb1
  - 172.20.0.12 srvWeb2

```
127.0.0.1 localhost
#127.0.1.1 debian
172.20.0.11 srvWeb1
172.20.0.12 srvWeb2
```

Test de connectivité en utilisant les nom host.

- Sur le terminal linux entrez la commande suivante :
  - ping srvWeb1
  - et
  - ping srvWeb2

```
root@srvWeb1:~# ping srvWeb1
PING srvWeb1 (172.20.0.11) 56(84) bytes of data.
64 bytes from srvWeb1 (172.20.0.11): icmp_seq=1 ttl=64 time=0.014 ms
^C
--- srvWeb1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.014/0.014/0.014/0.000 ms
root@srvWeb1:~# ping srvWeb2
PING srvWeb2 (172.20.0.12) 56(84) bytes of data.
64 bytes from srvWeb2 (172.20.0.12): icmp_seq=1 ttl=64 time=0.416 ms
64 bytes from srvWeb2 (172.20.0.12): icmp_seq=2 ttl=64 time=0.452 ms
^C
--- srvWeb2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1029ms
rtt min/avg/max/mdev = 0.416/0.434/0.452/0.018 ms
root@srvWeb1:~# _
```

## b. Installation du Service HeartBeat

Mise a jours de la liste des package disponible

- Sur le terminal Linux entrez la commande suivante :
  - apt-get update

```
root@srvWeb2:~# apt-get update
Atteint :1 http://security.debian.org/debian-security bullseye-security InRelease
Atteint :2 http://deb.debian.org/debian bullseye InRelease
Atteint :3 http://deb.debian.org/debian bullseye-updates InRelease
Lecture des listes de paquets... Fait
```

Mise a jours de tous les packages déjà présent sur la machine :

- Sur le terminal Linux entrez la commande suivante :
  - apt-get upgrade

```
root@srvWeb2:~# apt-get upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
Les paquets suivants ont été conservés :
 linux-image-amd64
0 mis à jour, 0 nouvellement installés, 0 à enlever et 1 non mis à jour.
```

Installation du package heartbeat

- Sur le terminal Linux entrez la commande suivante :
  - apt-get install heartbeat -y

```
root@srvWeb2:~# apt-get install heartbeat -y
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
heartbeat est déjà la version la plus récente (1:3.0.6-11+deb11u1).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 1 non mis à jour.
root@srvWeb2:~#
```

### c. Configuration du Service HearBeat

Suite à l'installation de heartbeat, un lien symbolique /etc/heartbeat est créé, pointant vers /etc/ha.d

Vous pouvez le vérifier à l'aide de la commande suivante : `ls -l /etc/heartbeat`

Dans le dossier /etc/ha.d, nous allons créer 3 fichiers :

- ha.cf : fichier de configuration principal
- haresources : fichier de configuration des ressources.
- authkeys : information d'authentification

Création des 3 fichiers dans le dossier /etc/ha.d:

- Sur le terminal linux entrez la commande suivante :
  - `touch ha.cf`
  - `touch haresources`
  - `touch authkeys`

Configuration du premier fichier (ha.cf) :

- Sur le terminal linux entrez la commande suivante :
  - `nano /etc/ha.d/ha.cf`
- Entrez y les paramètres suivants :
  - # Les fichiers logs de heartbeat (événements relatifs à heartbeat)  
`logfile /var/log/heartbeat`  
`logfacility local0`  
# Intervalle entre deux battements de cœur en seconde  
`keepalive 5`  
  
`# Temps nécessaire avant de considérer qu'un serveur (noeud) est mort (en seconde)`  
`deadtime 30`  
  
`# Interface d'écoute`  
  
`bcast ens33`  
  
`# liste des nœuds utilisées pour la Haute Disponibilité`  
`node srvWeb1 srvWeb2`  
  
`# comportement si le nœud revient dans le réseau`  
`auto_failback on`

Configuration du deuxième fichier (haresources) :

- Sur le terminal linux entrez la commande suivante :
  - nano /etc/ha.d/haresources
- Entrez y les paramètres suivants :
  - # Active l'interface IP Virtuelle avec comme noeud principal srvWeb2  
# syntaxe :hostname IPAddr::IPvirtuelle/CIDR/interface service  
#Hostname = Hostname du nœud principale (srvweb1)  
  
#Ipaddr = L'ip du nœud principale(172.20.0.11)
  - srvWeb1 IPAddr::172.20.0.10/24/ens33 apache2

Configuration du troisième fichier (authkeys) :

- Sur le terminal linux entrez la commande suivante :
  - nano /etc/ha.d/authkeys
- Entrez y les paramètres suivants :
  - auth 1  
1 md5 greta

modification les droits d'accessibilité au troisième fichier :

- Sur le terminal linux entrez la commande suivante :
- chmod 600 /etc/ha.d/authkeys

### **Désactivation du démarrage automatique d'apache**

Désactivation du démarrage automatique d'apache car c'est heartbeat (haresources) qui va gérer le lancement de ce dernier

- Sur le terminal linux entrez la commande suivante :
  - update-rc.d apache2 remove

Arrêtez apache2 :

- Sur le terminal linux entrez la commande suivante :
  - service apache2 stop
  - service heartbeat stop

## 7. Validation du fonctionnement de heartbeat.

Depuis les serveurs **srvweb1** et **srvweb2** veuillez démarrer apache2 et heartbeat :

- Sur le terminal linux entrez la commande suivante :
  - service apache2 start
  - service heartbeat start

Vérification de la présence de l'adresse ip virtuelle.

- A l'aide de la commande ip ad, vérifiez la présence d'une interface virtuelle ens33 d'adresse 172.20.0.10 au niveau de la machine srvWeb1 et srvWeb2.

Connexion sur les sites web

- Depuis la machine cliente lancez Edge pour tapez l'adresse IP du serveur srvweb : <http://172.20.0.11>
- Depuis la machine cliente lancez Edge pour tapez l'adresse IP du serveur srvweb : <http://172.20.0.12>
- Depuis la machine cliente lancez Edge pour tapez l'adresse IP Virtuelle : <http://172.20.0.10> vous devez avoir la page d'accueil d'apahe2 The Grill 1 !

Déconnexion du serveur principale

- Déconnectez l'interface réseau de srvWeb1 dans les settings de la machine
- attendre un petit peu
- Depuis le srvWeb2 avec la commande « ip a », vous devez voir apparaître l'interface virtuelle 172.20.0.10 et le navigateur affiche The Grill 2, donc le **Load Balancing** est fonctionnel.

Si vous avez des erreurs n'hésitez pas à regarder les logs :

- Journalctl -xe

ou bien

- echo > /var/log/syslog
- service heartbeat restart
- nano /var/log/syslog

# HAProxy : Équilibrage de Charge et Haute Disponibilité

Table des matières :

1. Introduction
  - a) Présentation de HAProxy
  - b) Les machines du réseau
2. Configuration du Serveur srv-haproxy
  - a) Configuration du Serveur srv-haproxy
  - b) Mise à jours de la distribution
3. Installation et configuration du Service Proxy (HAProxy)
  - a) Installation de HAProxy
  - b) Configuration de HAProxy
4. Validation du fonctionnement de HAProxy

## 1. Introduction

### a. Présentation de HAProxy

HAProxy est un équilibreur de charge open-source, souvent utilisé pour distribuer le trafic web entre plusieurs serveurs, assurant ainsi la répartition de la charge. Il fonctionne en surveillant la disponibilité des serveurs en amont et en redirigeant intelligemment les requêtes des clients vers les serveurs opérationnels. Son utilisation permet d'optimiser les performances, d'assurer la haute disponibilité des services, et de faciliter la gestion du trafic dans les architectures distribuées. Nous allons configurer le service haproxy pour assurer le balancement (load balancing) entre les serveurs web du cluster.

Attention : il y'a deux segments LAN sur VMWare "LAN" et "WAN"

### b. Les machines du réseau

Nous utiliserons le réseau déjà créé pour la haute disponibilité avec Heartbeat et nous y ajouterons en plus notre serveur HAProxy afin de permettre l'équilibrage de charge.

#### **Serveur HAProxy:**

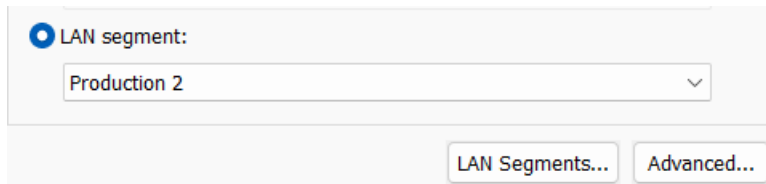
- hostname : srv-haproxy
- Interface IP: DHCP (NAT)
- Interface IP: 172.20.0.20 (Lan segment)
- netmask : 255.255.255.0

## 2. Installation et configuration du Serveur Proxy (srv-haproxy)

### a. Configuration du Serveur srv-haproxy

Ajout de la carte réseau dans le Lan Segment « Production 2 »

(**attention** une fois que la machine est sur le Lan segment celle-ci n'aura plus accès à internet donc plus accès aux mises à jours.). Pour cela nous vous conseillons l'installation en avance des packages : haproxy



Changement du hostname de la machine

- Sur le terminal Linux entrez la commande suivante :
  - hostnamectl set-hostname srv-haproxy

```
hostnamectl set-hostname srv-haproxy
```

Configuration des interfaces réseau du Serveur srv-haproxy

- Sur le terminal Linux entrez la commande suivante :
  - nano /etc/network/interfaces

```
root@srv-haproxy:~# nano /etc/network/interfaces
```

- Une fois dans le fichier de configuration des interfaces, modifiez le avec les paramètres suivants :

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens33
iface ens33 inet dhcp

#The second network interfaces
allow-hotplug ens36
iface ens36 inet static
address 172.20.0.20
netmask 255.255.255.0
```

### b. Mise à jour de la distribution

Mise à jour de la distribution

- Sur le terminal Linux entrez la commande suivante :
  - apt-get update && apt-get upgrade

```
root@srv-haproxy:~# apt-get update && apt-get upgrade
```



### 3. Installation et configuration du Service Proxy (HAProxy)

#### a. Installation de HAProxy

Installation de HAProxy

- Sur le terminal Linux entrez la commande suivante :
  - apt-get install haproxy

```
root@srv-haproxy:~# apt-get install haproxy
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
haproxy est déjà la version la plus récente (2.2.9-2+deb11u5).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 1 non mis à jour.
root@srv-haproxy:~#
```

#### b. Configuration de HAProxy

Éditer le fichier de configuration /etc/haproxy/haproxy.cfg et ajouter les lignes suivantes à la fin du fichier :

- Sur le terminal Linux entrez la commande suivante :
  - nano /etc/haproxy/haproxy.cfg

```
root@srv-haproxy:~# nano /etc/haproxy/haproxy.cfg
```

```
#Configuration du balancement
listen clusterWebStadiumCompany
bind 172.20.0.20:80 ← Adresse IP de notre serveur Proxy

#Mode d'écoute
mode http

#Option
option httpclose
option forwardfor

#liste des serveurs impliqués par le balancement
server srvweb1 172.20.0.11:80 check
server srvweb2 172.20.0.12:80 check

#Pour les statistique
stats enable
stats hide-version
stats refresh 30s
stats show-node
stats auth admin:password
stats uri /statistique
```

### 4. Validation du fonctionnement de HAProxy

Redémarrer HAProxy et vérifier son statut avec les commandes suivantes :

```
service haproxy restart
service haproxy status
```

Rendez-vous dans votre navigateur de recherche et rentrez-y l'address IP de vote serveur Proxy.