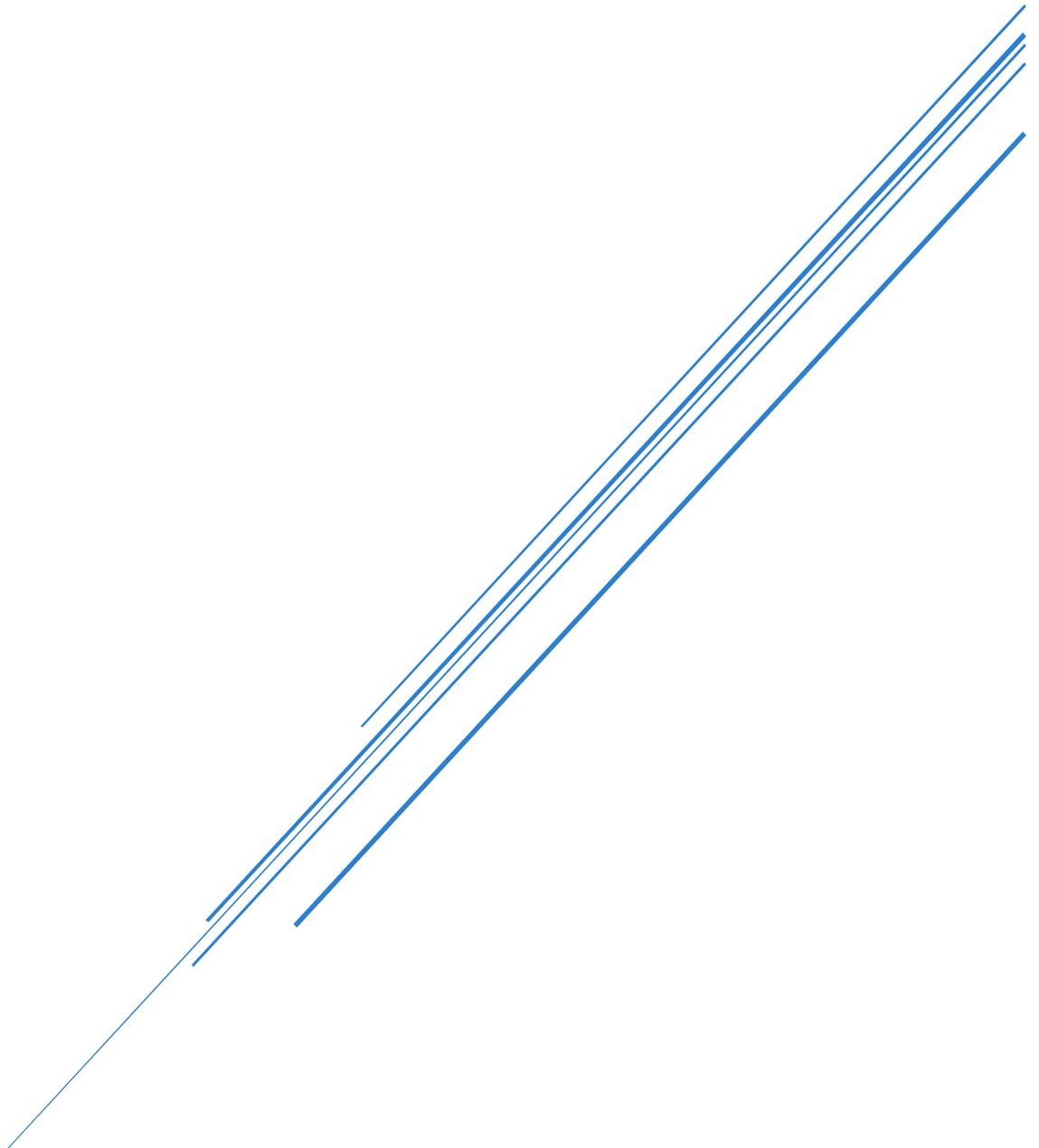


# PROJET HAPROXY ET KEEPALIVED

Réalisé lors du second stage de BTS SIO



Étudiant en BTS-SIO lycée Édouard Gand  
par Paris Steevy, Mohamed Toure

# 1. Introduction

Dans le cadre de ce projet, nous avons mis en place une solution garantissant la haute disponibilité d'un service web en utilisant **HAProxy** et **Keepalived**. L'objectif est d'assurer un équilibrage de charge efficace et un basculement automatique en cas de défaillance d'un serveur, améliorant ainsi la résilience et la continuité du service. Grâce à HAProxy, nous répartissons intelligemment le trafic entre plusieurs serveurs, tandis que Keepalived permet de gérer la redondance et d'assurer une adresse IP virtuelle flottante. Ce document détaille l'installation, la configuration et les bonnes pratiques pour déployer cette architecture robuste et performante.

## 2. Prérequis

Pour réaliser ce projet, il est nécessaire de disposer du matériel et des logiciels suivants :

### Matériel requis

- **1 ordinateur** (capable d'exécuter des machines virtuelles)

### Logiciels et configurations nécessaires

- **2 machines virtuelles sous Debian 12**
- **VirtualBox** (ou tout autre hyperviseur compatible)
- **Apache2 , HAProxy et Keepalived** (pour la gestion de la haute disponibilité)

# 1. Configuration de apache2

**sur une machine virtuelle « web1 » sur debian**

**Installer Apache**

> **sudo apt install apache2**

**Déposer les fichiers du site dans :**

> **sudo mkdir /var/www/html**

> **sudo chmod 777 index.html**

**Pour finir :**

> **sudo systemctl reload apache2**

> **sudo a2ensite site1.conf**

> **sudo a2ensite site2.conf**

**Cloner la machine virtuelle « web 1 » en « web2 » sur debian**

# 1. Configuration de Haproxy

> **sudo apt-get install haproxy**

**Modifier le fichier de configuration de haproxy**

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

Ajouter les ligne suivant a la fin du fichier :

```
frontend http_front
```

```
bind 10.0.2.16:8080 #IP WEB1
```

```
bind 127.0.0.1:8080 #IP WEB2
```

```
default_backend servers
```

```
backend servers
```

```
#envoie les nouvelles connexions au serveur qui a le moins de connexions actives
```

```
balance leastconn
```

```
cookie SERVERID insert indirect nocache
```

```
server site1 127.0.0.1:8081 check cookie S1
```

```
server site2 127.0.0.1:8082 check cookie S2
```

➤ **Sudo systemctl restart haproxy**

**Répéter cette opération pour les deux serveur Haproxy.**

*Haproxy définit un frontend qui équilibre la charge entre deux serveurs (site1 et site2), en utilisant la méthode leastconn pour diriger le trafic vers le serveur ayant le moins de connexions et en insérant un cookie pour gérer la persistance de session.*

## 2. Configuration de keepalive

**> sudo apt install keepalived**

**> sudo nano /etc/keepalived/keepalived.conf**

```
vrp_instance VI_1 {
state MASTER # ou BACKUP
interface enp0s3
virtual_router_id 51
priority 100 # modifier pour BACKUP a 99
advert_int 1
authentication {
    auth_type PASS
    auth_pass mypass12
}

virtual_ipaddress {
    10.0.2.16 # IP Virtuelle (VIP)
}
```

**Pour finir :**

**> sudo systemctl restart keepalived**

**> sudo systemctl stop keepalived**

**> sudo apt-get install ufw**

**> sudo ufw allow 80**

**> sudo ufw allow 443**

## Conclusion

**Grâce à l'utilisation conjointe de HAProxy pour l'équilibrage de charge et de Keepalived pour la gestion de la redondance via une IP flottante, nous avons mis en place une infrastructure web à haute disponibilité. Cette solution permet d'assurer une continuité de service même en cas de panne d'un serveur, tout en répartissant efficacement le trafic. En suivant les étapes d'installation et de configuration décrites dans ce document, vous obtenez une architecture robuste, évolutive et résiliente, prête à répondre aux exigences des environnements de production.**