

# Keepalived et la Haute Disponibilité du Reverse-Proxy avec Apache

## 1. Introduction

Dans les architectures réseau, **Keepalived** est un outil clé pour garantir la **haute disponibilité** (HA) des services critiques tels que les reverse-proxies. Associé à Apache, Keepalived permet de gérer automatiquement le basculement (failover) entre plusieurs serveurs en cas de défaillance.

---

## 2. Concepts Clés

### Haute Disponibilité (HA)

- Garantit la continuité du service même si un serveur tombe en panne.
- Implémente une redondance via plusieurs serveurs actifs et un mécanisme de basculement (failover).

### Keepalived

- Utilise le protocole **VRRP** (Virtual Router Redundancy Protocol) pour offrir une IP virtuelle flottante (VIP) qui peut être associée dynamiquement à un serveur.
- Fonctionne avec une configuration **master-slave** ou **actif-actif** pour les serveurs.

### IP Virtuelle (VIP)

- Une adresse IP virtuelle partagée entre plusieurs serveurs.
  - Les clients se connectent à la VIP, et Keepalived assure que l'un des serveurs répond toujours.
- 

## 3. Cas d'Usage : Reverse Proxy Apache avec Keepalived

### Objectif

- Configurer deux serveurs Apache comme reverse proxies avec Keepalived.
- Garantir que l'adresse IP virtuelle reste accessible même en cas de panne de l'un des serveurs.

### Architecture

- **Serveurs :**
    - **Apache-1 (Master)** : 192.168.1.101
    - **Apache-2 (Backup)** : 192.168.1.102
  - **VIP** : 192.168.1.200
  - **Backends :**
    - Serveurs backend à équilibrer via Apache.
- 

## 4. Installation de Keepalived

Sur les deux serveurs Apache, installez **Keepalived** :

```
sudo apt update
sudo apt install keepalived
```

---

## 5. Configuration de Keepalived

### Sur le Serveur Master (Apache-1)

Éditez le fichier de configuration `/etc/keepalived/keepalived.conf` :

```
vrrp_instance VI_1 {
    state MASTER
    interface eth0 # Interface réseau associée
    virtual_router_id 51
    priority 100 # Priorité la plus élevée pour le master
    advert_int 1

    authentication {
        auth_type PASS
        auth_pass secretpass
    }

    virtual_ipaddress {
        192.168.1.200 # IP virtuelle
    }

    track_script {
        chk_apache
    }
}

# Script pour vérifier si Apache fonctionne
vrrp_script chk_apache {
    script "/usr/bin/systemctl is-active apache2"
    interval 2
    weight 2
}
```

### Sur le Serveur Backup (Apache-2)

Éditez `/etc/keepalived/keepalived.conf` :

```
vrrp_instance VI_1 {
    state BACKUP
    interface eth0
    virtual_router_id 51
    priority 90 # Priorité inférieure pour le backup
    advert_int 1
```

```
authentication {
    auth_type PASS
    auth_pass secretpass
}

virtual_ipaddress {
    192.168.1.200
}

track_script {
    chk_apache
}

}

vrrp_script chk_apache {
    script "/usr/bin/systemctl is-active apache2"
    interval 2
    weight 2
}
```

## Redémarrez Keepalived

Sur les deux serveurs :

```
sudo systemctl enable keepalived
sudo systemctl start keepalived
```

---

## 6. Configuration d'Apache comme Reverse Proxy

Configurez Apache sur les deux serveurs pour agir comme reverse proxies.

### VirtualHost Configuration

Éditez `/etc/apache2/sites-available/000-default.conf` :

```
<VirtualHost *:80>
    ServerName 192.168.1.200

    ProxyPreserveHost On

    <Proxy "balancer://mycluster">
        BalancerMember http://backend1.local
        BalancerMember http://backend2.local
        ProxySet lbmethod=byrequests
    </Proxy>

    ProxyPass / balancer://mycluster/
```

```
ProxyPassReverse / balancer://mycluster/  
  
ErrorLog ${APACHE_LOG_DIR}/proxy_error.log  
CustomLog ${APACHE_LOG_DIR}/proxy_access.log combined  
</VirtualHost>
```

Redémarrez Apache sur les deux serveurs :

```
sudo systemctl restart apache2
```

---

## 7. Tests

### 1. Tester la VIP :

- Accédez à <http://192.168.1.200> dans un navigateur ou utilisez `curl` :

```
curl http://192.168.1.200
```

### 2. Simuler une Panne du Master :

- Arrêtez Apache ou Keepalived sur le serveur Master (Apache-1) :

```
sudo systemctl stop apache2
```

- Vérifiez que la VIP passe au serveur Backup (Apache-2).

### 3. Retour du Master :

- Relancez Apache sur le Master et observez que la VIP revient automatiquement sur le Master :

```
sudo systemctl start apache2
```

---

## 8. Surveillance et Debugging

### 1. Logs de Keepalived :

- Vérifiez les logs pour identifier les problèmes :

```
sudo journalctl -u keepalived
```

## 2. Surveillance Active :

- Intégrez Keepalived à un système de monitoring (Nagios, Prometheus).

## 3. Verifier les États VRRP :

- Utilisez `ip` pour vérifier l'attribution de la VIP :

```
ip addr show
```

---

## 9. Bonnes Pratiques

### 1. Redondance Réseau :

- Configurez plusieurs interfaces réseau pour éviter un point de défaillance unique.

### 2. Scripts de Santé :

- Personnalisez le script de vérification de santé pour inclure d'autres vérifications, comme la disponibilité des backends.

### 3. Priorités Ajustées :

- Ajustez les priorités des serveurs en fonction de leurs capacités.

### 4. Évitez les Conflits IP :

- Assurez-vous que la VIP n'est pas utilisée ailleurs dans le réseau.

---

## 10. Résumé

Avec Keepalived et Apache :

- La **haute disponibilité** est assurée via une **VIP** flottante.
- Les **basculements automatiques** (failover) garantissent la continuité du service.
- Apache, configuré comme **reverse proxy**, assure l'équilibrage de charge entre les backends.