

Documentation Technique

Haute Disponibilité avec Keepalived

VRRP · vrrp_track_process · NAT VIP · MariaDB HA

Adrien Lefebvre

Mars 2026

Hyper-V / Debian

1. Contexte et Objectifs

Ce document décrit la mise en place d'une infrastructure de haute disponibilité (HA) basée sur Keepalived et le protocole VRRP. L'objectif est de garantir la continuité de service en cas de panne d'un serveur, grâce au basculement automatique (failover) de l'adresse IP virtuelle vers un serveur de secours.

⚠ v2 — Cette version v2 intègre les corrections issues des tests en production : remplacement de `vrrp_script/pidof` par `vrrp_track_process`, `weight` négatif pour le failover, NAT sur la VIP, et accès MariaDB depuis le BACKUP.

Objectifs

- Cloner une VM existante et reconfigurer son adressage réseau
- Mettre en place un mécanisme de failover automatique avec Keepalived
- Surveiller les services critiques via `vrrp_track_process` (nginx, vsftpd, BIND)
- Garantir la continuité de service via une VIP (Virtual IP)
- Configurer le NAT Cisco sur la VIP plutôt que sur l'IP fixe du MASTER
- Autoriser MariaDB depuis le BACKUP (10.51.10.12)

Services déployés

Service	Statut	Rôle
nginx	✓ Actif	Serveur web / reverse proxy
vsftpd	✓ Actif	Serveur FTP
named (BIND)	✓ Actif	Serveur DNS
php8.2-fpm	✓ Actif	Moteur PHP pour Nginx
keepalived	✓ Actif	Gestion VIP / failover VRRP
mariadb	✓ Actif	Base de données (accès HA via VIP)

snmpd	<input checked="" type="checkbox"/> Actif	Supervision SNMP
ssh	<input checked="" type="checkbox"/> Actif	Accès administration sécurisé

2. Architecture Réseau

2.1 PLAN D'ADRESSAGE IP

Hôte	Rôle	Adresse IP	Interface
VM1 (Original)	MASTER	10.51.10.11	eth0
VM2 (Clone)	BACKUP	10.51.10.12	eth0
VIP (flottante)	VRRP Virtual IP	10.51.10.67	eth0 (logique)
Passerelle	—	10.51.10.254	—
Réseau	—	10.51.0.0/16	—

2.2 LOGIQUE DE FAILOVER VRRP

Événement	Comportement	VIP sur
Démarrage normal	MASTER prend la VIP	VM1 (10.51.10.11)
nginx / vsftpd / named DOWN	Priorité - 11 → 89 < 90 → BACKUP prend la VIP	VM2 (10.51.10.12)
MASTER down complet	BACKUP → MASTER state	VM2 (10.51.10.12)
MASTER revient	Préemption : VIP rebascule (priorité 100 > 90)	VM1 (10.51.10.11)

3. Clonage de la VM (Hyper-V)

⚠ Après un clonage, deux problèmes critiques doivent être résolus : conflit de machine-id et adresse IP identique.

3.1 ÉTAPES DANS HYPER-V

- Exporter la VM source depuis le Gestionnaire Hyper-V
- Importer la VM exportée (option « Copier » pour créer un nouvel ID)
- Attribuer une nouvelle adresse MAC à la carte réseau du clone
- Démarrer le clone, se connecter en console

3.2 CORRECTION DU MACHINE-ID

Chaque VM doit avoir un identifiant système unique :

```
sudo rm /etc/machine-id
sudo systemd-machine-id-setup
```

3.3 CHANGEMENT D'ADRESSE IP (VM BACKUP)

Modifier `/etc/network/interfaces` sur le clone :

```
sudo nano /etc/network/interfaces
```

Contenu à appliquer :

```
auto eth0
iface eth0 inet static
    address 10.51.10.12
    netmask 255.255.0.0
    gateway 10.51.10.254
    dns-nameservers 8.8.8.8 8.8.4.4
```

```
sudo systemctl restart networking
ping 10.51.10.11 # ping le MASTER
```


4. Configuration Keepalived


4.1 INSTALLATION

À exécuter sur les deux VMs :

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

4.2 CONFIGURATION MASTER — VRRP_TRACK_PROCESS

 **Fix Debian** — Version v2 : `vrrp_script` avec `pidof` remplacé par `vrrp_track_process`. Sur Debian, `pidof` est un symlink vers `killall5` qui n'accepte pas de nom de processus — les scripts échouaient silencieusement.

 **Fix weight** — `weight -11` (négatif) : soustrait quand le service est DOWN. Avec `weight +11` (positif), la priorité montait à 122 au démarrage, rendant tout failover impossible.

Fichier : `/etc/keepalived/keepalived.conf` (MASTER — 10.51.10.11)

```
vrrp_track_process chk_nginx {
    process nginx
    weight -11
}
vrrp_track_process chk_vsftpd {
    process vsftpd
    weight -11
}
vrrp_track_process chk_bind {
    process named
    weight -11
}

vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 51
    priority 100
```

```

advert_int 1
authentication {
    auth_type PASS
    auth_pass 1234
}
virtual_ipaddress {
    10.51.10.67
}
track_process {
    chk_nginx
    chk_vsftpd
    chk_bind
}
}

```

Calcul de priorité

Situation	Priorité effective	Résultat
Tous les services UP	100 (aucune soustraction)	MASTER ✓
nginx DOWN	$100 - 11 = 89 < 90$ (BACKUP)	Failover vers BACKUP ✓
vsftpd DOWN	$100 - 11 = 89 < 90$	Failover vers BACKUP ✓
named DOWN	$100 - 11 = 89 < 90$	Failover vers BACKUP ✓

4.3 CONFIGURATION BACKUP — MINIMALE

i Le BACKUP n'a pas besoin de surveiller les services. Il attend les adverts VRRP du MASTER et prend la main si celui-ci disparaît ou passe sous sa priorité.

Fichier : `/etc/keepalived/keepalived.conf` (BACKUP — 10.51.10.12)

```

vrrp_instance VI_1 {
    state BACKUP
    interface eth0
    virtual_router_id 51
    priority 90
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1234
    }
    virtual_ipaddress {
        10.51.10.67
    }
}

```

4.4 DÉMARRAGE ET ACTIVATION

Sur les deux VMs :

```

sudo systemctl enable keepalived
sudo systemctl start keepalived

```

```
sudo journalctl -u keepalived -f
```

5. NAT Cisco — Règles sur la VIP

i VIP NAT — Les règles NAT pointent désormais sur la VIP 10.51.10.67 (et non plus sur l'IP fixe du MASTER 10.51.10.11). La VIP suit automatiquement le nœud MASTER actif, rendant le NAT transparent au failover.

5.1 INTERFACE WAN 172.18.55.X

Suppression des règles port-par-port sur .11, remplacement par une règle full-IP sur la VIP :

```
no ip nat inside source static tcp 10.51.10.11 21 172.18.55.51 21 extendable
no ip nat inside source static tcp 10.51.10.11 80 172.18.55.51 80 extendable
no ip nat inside source static tcp 10.51.10.11 443 172.18.55.51 443 extendable
```

```
ip nat inside source static 10.51.10.67 172.18.55.51
```

5.2 INTERFACE WAN 192.168.4.X


```
no ip nat inside source static tcp 10.51.10.11 80 192.168.4.51 80 extendable
no ip nat inside source static tcp 10.51.10.11 443 192.168.4.51 443 extendable
```

```
ip nat inside source static 10.51.10.67 192.168.4.51
```

Les règles sur 10.52.10.31 (192.168.4.52) ne sont pas modifiées — elles concernent un autre cluster.

```
write memory
```

6. MariaDB — Accès depuis le BACKUP

 Erreur rencontrée : SQLSTATE[HY000] [1130] Host '10.51.10.12' is not allowed to connect to this MariaDB server

6.1 CRÉER UN UTILISATEUR AUTORISÉ DEPUIS LE BACKUP

Se connecter localement sur le serveur MariaDB (MASTER), puis :

```
mysql -u root -p
```

```
-- Créer l'utilisateur avec accès depuis le BACKUP uniquement
CREATE USER 'user'@'10.51.10.12' IDENTIFIED BY 'motdepasse';
GRANT ALL PRIVILEGES ON ma_base.* TO 'user'@'10.51.10.12';
FLUSH PRIVILEGES;
```

i Recommandation : les clients doivent se connecter via la VIP 10.51.10.67 pour bénéficier du HA. Créer également un utilisateur '@10.51.10.67' ou utiliser un wildcard '%' avec restriction réseau au niveau du firewall.

6.2 VÉRIFIER QUE MARIADB ÉCOUTE SUR LE RÉSEAU

```
grep bind-address /etc/mysql/mariadb.conf.d/50-server.cnf
```

Si la valeur est 127.0.0.1, changer en :

```
bind-address = 0.0.0.0
sudo systemctl restart mariadb
```

6.3 CONNEXION DEPUIS LE BACKUP

```
mysql -u user -p -h 10.51.10.67 # via la VIP (recommandé)
mysql -u user -p -h 10.51.10.11 # via IP fixe MASTER
```

7. Problèmes rencontrés et corrections

Problème	Cause	Solution v2
killall5: only one argument	pidof est un symlink vers killall5 sur Debian — ne supporte pas les noms de process	Remplacer vrrp_script + pidof par vrrp_track_process
Pas de failover malgré nginx arrêté	weight +11 : priorité montait à 122 au lieu de descendre	Passer à weight -11 : priorité descend à 89 < 90 BACKUP
BACKUP config inutilement complexe	vrrp_script vide + track_script vide laissés par copier-coller	Supprimer tout track_script du BACKUP — inutile
NAT ciblait l'IP fixe du MASTER	En cas de failover, le trafic externe continuait d'aller sur .11	Remplacer les règles tcp port-par-port par static sur la VIP .67
Host 10.51.10.12 not allowed (MariaDB)	Aucun user MariaDB créé pour le BACKUP	CREATE USER + GRANT depuis l'adresse du BACKUP

8. Commandes de Référence

KEEPALIVED

```
sudo systemctl start keepalived
sudo systemctl stop keepalived # déclenche failover si MASTER
sudo systemctl restart keepalived
sudo systemctl status keepalived
sudo journalctl -u keepalived -f # logs temps réel
```

VÉRIFIER LA VIP

```
ip a show eth0 # chercher 10.51.10.67
ip a | grep 10.51.10.67 # présence sur le nœud actif
```

SERVICES SURVEILLÉS

```
sudo systemctl status nginx vsftpd named
/usr/bin/pgrep nginx # alternative propre à pidof sur Debian
```

RÉSEAU

```
ip a # toutes les IPs
ip route show # table de routage
cat /etc/network/interfaces # config réseau
```

MARIADB

```
mysql -u user -p -h 10.51.10.67 # connexion via VIP (recommandé HA)
SHOW GRANTS FOR 'user'@'10.51.10.12';
```

Annexe — Récapitulatif des fichiers modifiés

Fichier	VM(s)	Modification v2
/etc/network/interfaces	BACKUP	IP statique 10.51.10.12
/etc/machine-id	BACKUP	Régénéré après clonage
/etc/keepalived/keepalived.conf	MASTER	vrrp_track_process + weight -11
/etc/keepalived/keepalived.conf	BACKUP	Config minimale, sans track_process
NAT Cisco (172.18.55.x)	Routeur	static 10.51.10.67 → 172.18.55.51
NAT Cisco (192.168.4.x)	Routeur	static 10.51.10.67 → 192.168.4.51
MariaDB users	MASTER	CREATE USER 'user'@'10.51.10.12'