Configuration Pare-feu

Par-feu OVH

```
#!/usr/bin/env bash
# ================================================================================
# Configuration UFW — VPS OVH
# Interface physique : eth0   |   IP WireGuard : 10.0.0.1
# ================================================================================
#
# À exécuter en root APRÈS avoir démarré le conteneur WireGuard (wg0 doit exister).
#   sudo bash ovh-ufw-setup.sh
#
# Ce script :
#   1. Active le forwarding IP au niveau hôte
#   2. Ajoute le masquerading (NAT) pour les clients exit-node OVH
#   3. Pose une politique FORWARD = DROP et n'autorise que le trafic VPN voulu
#   4. Ouvre les ports nécessaires (tunnel, SSH, Portainer UI sur le mesh)
#
# IMPORTANT : ne casse PAS Docker. UFW pose FORWARD à DROP, ce que Docker
# fait déjà de son côté ; Docker gère ses propres règles ACCEPT via
# DOCKER-USER. La communication entre conteneurs reste fonctionnelle.
# ================================================================================
set -euo pipefail

IFACE_PHYS="eth0"
WG_IFACE="wg0"
MESH="10.0.0.0/24"

echo "[1/5] Activation du forwarding IP (hôte)"
# net/ipv4 syntaxe attendue par /etc/ufw/sysctl.conf
grep -q "net/ipv4/ip_forward=1" /etc/ufw/sysctl.conf || \
  echo "net/ipv4/ip_forward=1" >> /etc/ufw/sysctl.conf
grep -q "net/ipv6/conf/all/forwarding=1" /etc/ufw/sysctl.conf || \
  echo "net/ipv6/conf/all/forwarding=1" >> /etc/ufw/sysctl.conf
# src_valid_mark requis par wg-quick (routage marqué)
echo "net.ipv4.conf.all.src_valid_mark=1" > /etc/sysctl.d/99-wireguard.conf
```

```
sysctl -p /etc/sysctl.d/99-wireguard.conf >/dev/null

echo "[2/5] Politique FORWARD par défaut = DROP"
sed -i 's/^DEFAULT_FORWARD_POLICY=.*/DEFAULT_FORWARD_POLICY="DROP"/' /etc/default/ufw

echo "[3/5] Masquerading NAT dans before.rules"
# On insère un bloc *nat en tête de /etc/ufw/before.rules s'il n'y est pas déjà.
if ! grep -q "WG-MASQUERADE-OVH" /etc/ufw/before.rules; then
  cp /etc/ufw/before.rules /etc/ufw/before.rules.bak.$(date +%s)
  cat > /tmp/wg-nat.rules <<EOF
# WG-MASQUERADE-OVH (ne pas supprimer cette ligne, sert de marqueur)
*nat
:POSTROUTING ACCEPT [0:0]
# Le trafic des clients VPN qui sort par eth0 prend l'IP publique d'OVH
-A POSTROUTING -s ${MESH} -o ${IFACE_PHYS} -j MASQUERADE
COMMIT

EOF
  # Préfixe le fichier existant avec le bloc nat
  cat /tmp/wg-nat.rules /etc/ufw/before.rules > /tmp/before.rules.new
  mv /tmp/before.rules.new /etc/ufw/before.rules
  rm -f /tmp/wg-nat.rules
fi

echo "[4/5] Ports d'entrée (INPUT)"
ufw allow OpenSSH                                    # ne pas se verrouiller dehors
ufw allow 51820/udp comment 'WireGuard tunnel'
# Portainer UI : accessible UNIQUEMENT via le mesh (interface wg0)
ufw allow in on ${WG_IFACE} to any port 9000 proto tcp comment 'Portainer UI (mesh)'
# SSH via le mesh (administration interne)
ufw allow in on ${WG_IFACE} to any port 22 proto tcp comment 'SSH mesh'

echo "[5/5] Règles de routage VPN (FORWARD)"
# --- Client ovh-exitnode (10.0.0.10) ---
# Sortie internet par OVH
ufw route allow in on ${WG_IFACE} out on ${IFACE_PHYS} from 10.0.0.10 comment 'ovh-exitnode
internet'
# Accès au reste du mesh (relais vers MRS/Albi)
ufw route allow in on ${WG_IFACE} out on ${WG_IFACE} from 10.0.0.10 to ${MESH} comment 'ovh-
exitnode mesh'
```

```
# --- Client private-only entré via OVH (10.0.0.40) ---
# Mesh uniquement, AUCUNE sortie eth0 (pas d'internet, pas de LAN)
ufw route allow in on ${WG_IFACE} out on ${WG_IFACE} from 10.0.0.40 to ${MESH} comment
'private-only-ovh mesh'

echo "Activation/rechargement d'UFW"
ufw --force enable
ufw reload

echo
echo "=== Terminé. Vérifs utiles : ==="
echo "  ufw status verbose"
echo "  ufw route status"
echo "  iptables -t nat -L POSTROUTING -n | grep MASQUERADE"
echo
echo "Si les conteneurs Docker perdent le réseau, redémarrez Docker UNE fois"
echo "pour qu'il réinsère ses règles au-dessus de celles d'UFW :"
echo "  systemctl restart docker"
```

Par-feu Marseille

```
#!/usr/bin/env bash
# ============================================================================
# Configuration UFW — Serveur Marseille
# Interface physique : enp4s0   |   IP WireGuard : 10.0.0.2
# LAN local : 192.168.1.0/24
# ============================================================================
#
#   sudo bash marseille-ufw-setup.sh
#
# Accès LAN Marseille (192.168.1.0/24) autorisé UNIQUEMENT au client
# mrs-exitnode (10.0.0.20). Tout autre peer est bloqué par le FORWARD DROP.
# Aucune route statique sur la box n'est nécessaire (masquerading sur enp4s0).
# ============================================================================
set -euo pipefail

IFACE_PHYS="enp4s0"
WG_IFACE="wg0"
MESH="10.0.0.0/24"
LAN="192.168.1.0/24"
```

```
echo "[1/5] Forwarding IP (hôte)"
grep -q "net/ipv4/ip_forward=1" /etc/ufw/sysctl.conf || \
  echo "net/ipv4/ip_forward=1" >> /etc/ufw/sysctl.conf
grep -q "net/ipv6/conf/all/forwarding=1" /etc/ufw/sysctl.conf || \
  echo "net/ipv6/conf/all/forwarding=1" >> /etc/ufw/sysctl.conf
echo "net.ipv4.conf.all.src_valid_mark=1" > /etc/sysctl.d/99-wireguard.conf
sysctl -p /etc/sysctl.d/99-wireguard.conf >/dev/null

echo "[2/5] Politique FORWARD = DROP"
sed -i 's/^DEFAULT_FORWARD_POLICY=.*/DEFAULT_FORWARD_POLICY="DROP"/' /etc/default/ufw

echo "[3/5] Masquerading NAT (enp4s0)"
if ! grep -q "WG-MASQUERADE-MRS" /etc/ufw/before.rules; then
  cp /etc/ufw/before.rules /etc/ufw/before.rules.bak.$(date +%s)
  cat > /tmp/wg-nat.rules <<EOF
# WG-MASQUERADE-MRS (marqueur — ne pas supprimer)
*nat
:POSTROUTING ACCEPT [0:0]
# Couvre l'exit node (internet) ET l'accès au LAN : dans les deux cas le
# trafic sort par enp4s0 et prend l'IP locale du serveur (ex 192.168.1.100),
# ce qui évite toute route statique sur la box.
-A POSTROUTING -s ${MESH} -o ${IFACE_PHYS} -j MASQUERADE
COMMIT

EOF
  cat /tmp/wg-nat.rules /etc/ufw/before.rules > /tmp/before.rules.new
  mv /tmp/before.rules.new /etc/ufw/before.rules
  rm -f /tmp/wg-nat.rules
fi

echo "[4/5] Ports d'entrée (INPUT)"
ufw allow OpenSSH
ufw allow 51820/udp comment 'WireGuard tunnel'
# Portainer Agent : joignable uniquement par Portainer UI via le mesh
ufw allow in on ${WG_IFACE} to any port 9001 proto tcp comment 'Portainer Agent (mesh)'
# SSH via mesh — sert aussi aux sauvegardes rsync depuis Albi
ufw allow in on ${WG_IFACE} to any port 22 proto tcp comment 'SSH mesh / backups'

echo "[5/5] Règles de routage VPN (FORWARD)"
```

```
# --- Client mrs-exitnode (10.0.0.20) : internet + LAN Marseille ---
ufw route allow in on ${WG_IFACE} out on ${IFACE_PHYS} from 10.0.0.20 comment 'mrs-exitnode internet+LAN'
ufw route allow in on ${WG_IFACE} out on ${WG_IFACE} from 10.0.0.20 to ${MESH} comment 'mrs-exitnode mesh'
# --- Client private-only entré via MRS (10.0.0.41) : mesh uniquement ---
# Pas de règle "out on enp4s0" => ni internet ni LAN pour ce client.
ufw route allow in on ${WG_IFACE} out on ${WG_IFACE} from 10.0.0.41 to ${MESH} comment 'private-only-mrs mesh'

echo "Activation/rechargement d'UFW"
ufw --force enable
ufw reload

echo
echo "=== Terminé. Vérifs : ufw status verbose ; ufw route status ==="
echo "Test attendu :"
echo "  - mrs-exitnode atteint 192.168.1.x : OUI"
echo "  - private-only / autres profils atteignent 192.168.1.x : NON"
echo
echo "Si Docker perd le réseau : systemctl restart docker (une fois)"
```

Par-feu Albi

```
#!/usr/bin/env bash
# ================================================================================
# Configuration UFW — Serveur Albi
# Interface physique : enp2s0   |   IP WireGuard : 10.0.0.3
# LAN local : 192.168.2.0/24
# ================================================================================
#
#   sudo bash albi-ufw-setup.sh
#
# Accès LAN Albi (192.168.2.0/24) autorisé UNIQUEMENT au client
# albi-exitnode (10.0.0.30).
#
# COHABITATION DOCKER : Home Assistant et Frigate ne sont pas affectés.
# Leur trafic est local (OUTPUT/INPUT) ou géré par Docker, pas par la
# chaîne FORWARD qu'on filtre ici. Voir la note Docker en fin de script.
# ================================================================================
```

```
set -euo pipefail

IFACE_PHYS="enp2s0"
WG_IFACE="wg0"
MESH="10.0.0.0/24"
LAN="192.168.2.0/24"

echo "[1/5] Forwarding IP (hôte)"
grep -q "net/ipv4/ip_forward=1" /etc/ufw/sysctl.conf || \
  echo "net/ipv4/ip_forward=1" >> /etc/ufw/sysctl.conf
grep -q "net/ipv6/conf/all/forwarding=1" /etc/ufw/sysctl.conf || \
  echo "net/ipv6/conf/all/forwarding=1" >> /etc/ufw/sysctl.conf
echo "net.ipv4.conf.all.src_valid_mark=1" > /etc/sysctl.d/99-wireguard.conf
sysctl -p /etc/sysctl.d/99-wireguard.conf >/dev/null

echo "[2/5] Politique FORWARD = DROP"
sed -i 's/^DEFAULT_FORWARD_POLICY=.*/DEFAULT_FORWARD_POLICY="DROP"/' /etc/default/ufw

echo "[3/5] Masquerading NAT (enp2s0)"
if ! grep -q "WG-MASQUERADE-ALBI" /etc/ufw/before.rules; then
  cp /etc/ufw/before.rules /etc/ufw/before.rules.bak.$(date +%s)
  cat > /tmp/wg-nat.rules <<EOF
# WG-MASQUERADE-ALBI (marqueur — ne pas supprimer)
*nat
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -s ${MESH} -o ${IFACE_PHYS} -j MASQUERADE
COMMIT

EOF
  cat /tmp/wg-nat.rules /etc/ufw/before.rules > /tmp/before.rules.new
  mv /tmp/before.rules.new /etc/ufw/before.rules
  rm -f /tmp/wg-nat.rules
fi

echo "[4/5] Ports d'entrée (INPUT)"
ufw allow OpenSSH
ufw allow 51820/udp comment 'WireGuard tunnel'
ufw allow in on ${WG_IFACE} to any port 9001 proto tcp comment 'Portainer Agent (mesh)'
ufw allow in on ${WG_IFACE} to any port 22 proto tcp comment 'SSH mesh / backups'
```

```
echo "[5/5] Règles de routage VPN (FORWARD)"
# --- Client albi-exitnode (10.0.0.30) : internet + LAN Albi ---
ufw route allow in on ${WG_IFACE} out on ${IFACE_PHYS} from 10.0.0.30 comment 'albi-exitnode internet+LAN'
ufw route allow in on ${WG_IFACE} out on ${WG_IFACE} from 10.0.0.30 to ${MESH} comment 'albi-exitnode mesh'
# --- Client private-only entré via Albi (10.0.0.42) : mesh uniquement ---
ufw route allow in on ${WG_IFACE} out on ${WG_IFACE} from 10.0.0.42 to ${MESH} comment 'private-only-albi mesh'

echo "Activation/rechargement d'UFW"
ufw --force enable
ufw reload

echo
echo "=== Terminé. ==="
echo "NOTE DOCKER (Home Assistant / Frigate) :"
echo "  Si après activation d'UFW les conteneurs perdent l'accès réseau,"
echo "  redémarrez Docker UNE fois pour qu'il réinsère ses règles FORWARD"
echo "  au-dessus de celles d'UFW :"
echo "    systemctl restart docker"
echo "  (cela redémarre Home Assistant et Frigate — à faire à un moment calme)"
```

Revision #1
Created 5 June 2026 09:16:45 by gpatruno
Updated 5 June 2026 09:18:45 by gpatruno