Infrastructure VPN Sécurisée sur Proxmox / OVH

Déploiement Pritunl · Configuration Réseau · Durcissement Avancé CIS Level 2

Auteur	AFFANOU Eric Toussaint Yao — SysAdmin Réseau / DevOps
Version initiale	v1.0 — 8 mai 2025 (rédaction initiale)
Version mise à jour	v2.0 — 21 mai 2026 (déploiement réel + durcissement complet)
Statut	☑ **Validé pour mise en production**
Classification	Confidentiel — Usage interne
Environnement cible	Serveur dédié OVH — Proxmox VE 8.x — Ubuntu 24.04 LTS
Protocoles VPN	OpenVPN (1194/UDP) & WireGuard (14623/UDP)
Niveau durcissement	Avancé — CIS Level 2, AIDE, Auditd, Logwatch, Fail2ban, Sysctl

Historique des versions : v1.0 (mai 2025) — Rédaction initiale · v2.0 (mai 2026) — Déploiement réel, corrections MongoDB 8.0, durcissement SSH, Fail2ban, AIDE, Logwatch, sysctl, NAT complet

Table des matières

1. Introduction et Contexte

Ce guide technique de référence pour le déploiement d'une infrastructure VPN sécurisée basée sur Pritunl, hébergée sur un serveur dédié OVH exploitant l'hyperviseur Proxmox VE. Il couvre l'intégralité du processus : de la conception réseau jusqu'au durcissement avancé de la machine virtuelle serveur, en conformité avec le CIS Ubuntu 24.04 LTS Benchmark (Level 2), NIST SP 800-123 et les recommandations ANSSI.

> **🖧 Rôle central de la VM tourou**
>
> tourou est la VM pivot de toute l'infrastructure. Elle assume simultanément trois rôles critiques :
>
> **1. Serveur VPN (Pritunl)** — Point de terminaison OpenVPN et WireGuard. Gère l'authentification, le chiffrement et la distribution des profils clients.
>
> **2. Routeur / Passerelle LAN** — Assure le routage entre le réseau public (ens18 / IP Failover OVH), le réseau LAN interne (ens19 / 20.20.20.1/24) et les tunnels VPN (72.16.0.0/24 et 72.17.0.0/24). Toutes les VM internes passent par tourou pour atteindre internet.
>
> **3. Accès de secours Proxmox** — En cas de problème sur une VM interne ou sur le réseau LAN, tourou garantit un accès SSH direct via l'IP Failover OVH (port 2235), permettant à l'administrateur de se connecter à Proxmox et de rétablir les services sans intervention physique sur le serveur.
>
> *tourou est donc le seul point d'accès externe à toute l'infrastructure. Sa disponibilité et son durcissement sont critiques pour la continuité opérationnelle.*

1.1 Objectifs

- Fournir un accès VPN sécurisé (OpenVPN et WireGuard) aux machines virtuelles internes via Pritunl
- Isoler le trafic réseau par segmentation Bridge WAN (vmbr0) / Bridge LAN (vmbr1)
- Permettre l'accès SSH aux VM internes (VM2, VM3) exclusivement via le tunnel VPN
- Appliquer un durcissement avancé conforme CIS Benchmark Ubuntu 24.04 LTS — Level 2
- Intégrer AIDE, Logwatch, Auditd, Fail2ban et AppArmor pour une surveillance complète
- Documenter toutes les étapes de manière reproductible, auditable et scientifiquement rigoureuse

1.2 Périmètre technique

Hyperviseur	Proxmox VE 8.x — serveur dédié OVH
OS VM VPN	Ubuntu 24.04 LTS (Noble Numbat) — Server edition
Serveur VPN	Pritunl Community Edition
IP publique	1 IP principale (eth0 Proxmox) + 1 IP Failover OVH (VM Pritunl)
Réseau LAN interne	vmbr1 — 20.20.20.0/24
Réseau tunnel VPN	72.16.0.0/24 (géré automatiquement par Pritunl)
Protocoles VPN	OpenVPN UDP:1194 + WireGuard UDP:14623

Niveau durcissement	Avancé — CIS Level 2, AIDE, Logwatch, Auditd, AppArmor, Fail2ban

Note importante : Toutes les adresses IP présentes dans ce document (ex. <IP_PRINCIPALE>, <IP_FAILOVER>, <GATEWAY_OVH>) sont des variables à remplacer par vos valeurs réelles OVH avant application.

2. Architecture Réseau — Conception et Justification

L'architecture repose sur une double segmentation réseau au sein de Proxmox : un bridge WAN (vmbr0) exposé à internet via l'IP failover OVH, et un bridge LAN (vmbr1) pour le réseau interne. La VM Pritunl est positionnée en tant que passerelle et point de terminaison VPN entre ces deux zones.

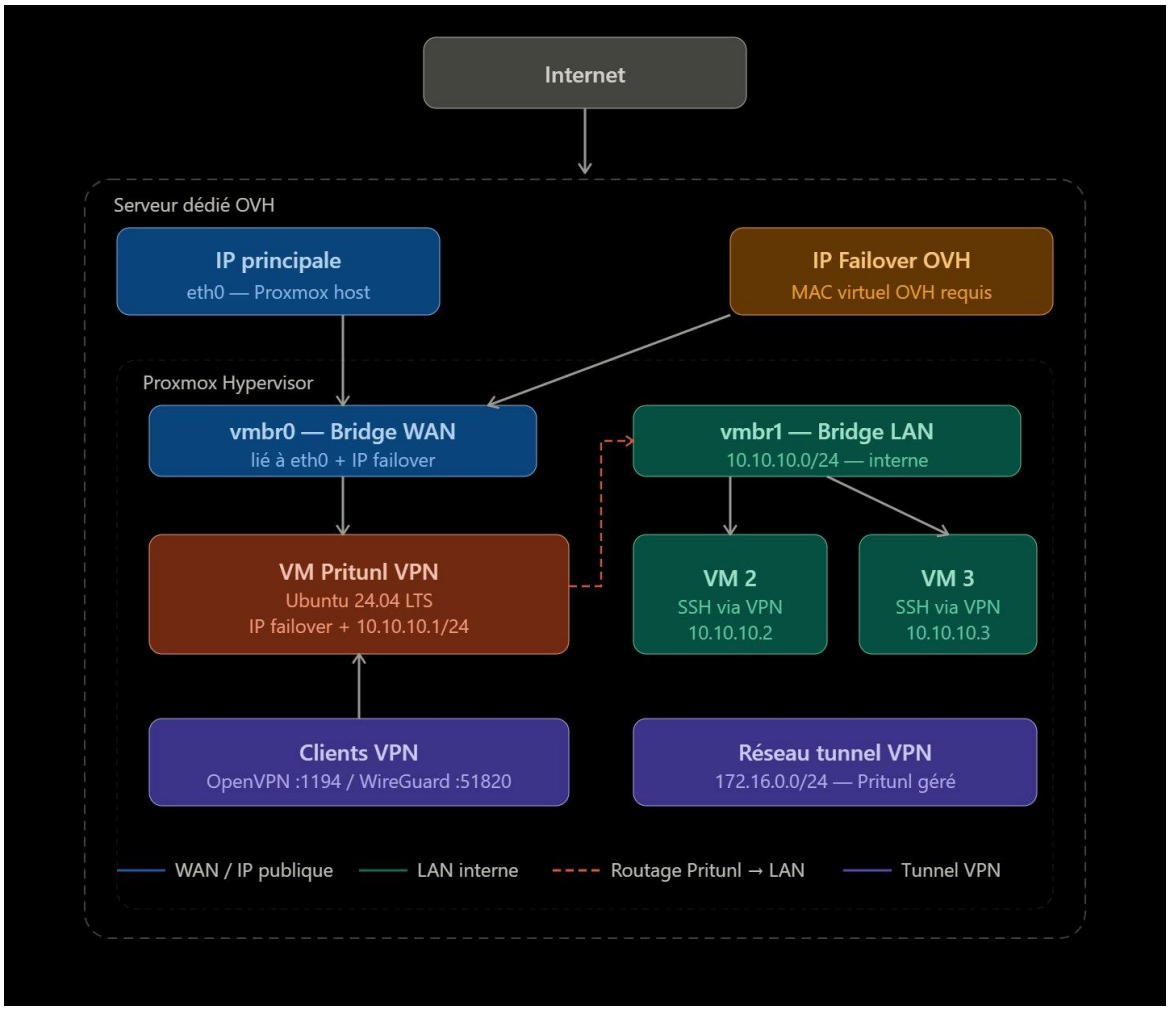

Figure 1 — Architecture réseau : Proxmox/OVH avec VM Pritunl VPN, Bridge WAN (vmbr0), Bridge LAN (vmbr1) et réseau tunnel 72.16.0.0/24

2.1 Composants réseau Proxmox

2.1.1 Bridge vmbr0 — Interface WAN

Le bridge vmbr0 est lié à l'interface physique eth0 du serveur dédié OVH. Il porte à la fois l'IP principale (accès Proxmox) et l'IP failover OVH (exposée par la VM Pritunl). La configuration de l'IP failover nécessite l'attribution d'une adresse MAC virtuelle par OVH via le Manager.

```
# /etc/network/interfaces — Proxmox host
auto lo
iface lo inet loopback
```

```
auto eth0
iface eth0 inet manual

auto vmbr0
iface vmbr0 inet static
    address <IP_PRINCIPALE>/32
    gateway <GATEWAY_OVH>
    bridge-ports eth0
    bridge-stp off
    bridge-fd 0
    # Routage IP failover vers VM Pritunl
    post-up ip route add <IP_FAILOVER>/32 dev vmbr0
    post-up ip rule add from <IP_FAILOVER> lookup 1
```

2.1.2 Bridge vmbr1 — Interface LAN interne

Le bridge vmbr1 constitue le réseau privé 20.20.20.0/24. Il n'est pas rattaché à une interface physique (bridge interne uniquement). Toutes les VM du réseau interne y sont connectées. La VM Pritunl y possède l'adresse 20.20.20.1/24 et fait office de gateway pour ce sous-réseau.

```
auto vmbr1
iface vmbr1 inet static
    address 10.10.10.254/24
    bridge-ports none
    bridge-stp off
    bridge-fd 0
```

2.2 Plan d'adressage IP complet

Composant	Interface	Adresse IP / Masque	Rôle
Proxmox host	eth0 / vmbr0	<IP_PRINCIPALE>/32	Accès hyperviseur & SSH admin
tourou	ens18 (WAN)	<IP_FAILOVER>/32	Exposition VPN public + HTTPS
tourou	ens19 (LAN)	20.20.20.1/24	Gateway réseau interne
VM 2	ens18 (LAN)	10.10.10.2/24	SSH via VPN uniquement
VM 3	ens18 (LAN)	10.10.10.3/24	SSH via VPN uniquement
Tunnel VPN	tun0 / wg0	72.16.0.0/24	Réseau clients VPN Pritunl

3. Configuration de l'IP Failover OVH

L'IP Failover OVH est une adresse IP publique additionnelle qui peut être routée vers n'importe quelle VM du serveur dédié. Elle nécessite l'utilisation d'une adresse MAC virtuelle assignée par OVH et configurée sur l'interface WAN de la VM Pritunl.

3.1 Obtention de la MAC virtuelle OVH

1. Se connecter au Manager OVH : https://www.ovh.com/manager
2. Naviguer vers : Bare Metal Cloud > Serveurs dédiés > [votre serveur]
3. Section « IPs » > cliquer sur l'IP failover > « Ajouter une MAC virtuelle »
4. Sélectionner le type : OVH (pour VM KVM/Proxmox)
5. Récupérer la MAC virtuelle générée (format : 02:00:00:XX:XX:XX)
6. Dans Proxmox, attribuer cette MAC à l'interface vmbr0 de la VM Pritunl lors de sa création

> ⚠ **Attention :** Sans la MAC virtuelle OVH correctement renseignée sur l'interface WAN de la VM, l'IP failover ne sera pas routable et la VM ne sera pas accessible depuis internet.

3.2 Configuration réseau de la VM Pritunl (Netplan)

```
# /etc/netplan/00-installer-config.yaml — VM Pritunl
network:
  version: 2
  ethernets:
    ens18:
      # Interface WAN — IP Failover OVH
      addresses:
        - <IP_FAILOVER>/32
      routes:
        - to: default
          via: <GATEWAY_OVH>
          on-link: true
      nameservers:
        addresses: [8.8.8.8, 1.1.1.1]
    ens19:
      # Interface LAN — Réseau interne 20.20.20.0/24
      addresses:
        - 20.20.20.1/24
```

```
# Appliquer et vérifier
sudo netplan apply
ip addr show ens18   # => <IP_FAILOVER>/32
ip addr show ens19   # => 20.20.20.1/24
ping -c 4 8.8.8.8    # Test connectivité WAN
```

4. Création des Machines Virtuelles dans Proxmox

4.1 VM tourou — Paramètres de création

Nom VM	tourou
OS	Ubuntu 24.04 LTS Server (ISO officiel)
CPU	2 vCPU minimum — 4 recommandé en production
RAM	2 Go minimum — 4 Go recommandé
Disque	30 Go — thin provisioning (LVM ou ZFS)
Interface 1 (WAN)	Bridge : vmbr0 | MAC : MAC virtuelle OVH | Modèle : VirtIO
Interface 2 (LAN)	Bridge : vmbr1 | MAC : auto | Modèle : VirtIO
QEMU Guest Agent	Activé (apt install qemu-guest-agent)
Démarrage automatique	Activé — Order 1 (premier démarrage)

4.2 VM internes (VM2, VM3) — Paramètres

OS	Ubuntu 24.04 LTS Server
CPU / RAM	1-2 vCPU — 1-2 Go RAM
Disque	20 Go
Interface réseau	Bridge : vmbr1 UNIQUEMENT — Pas de connexion WAN
Sécurité	SSH accessible UNIQUEMENT via tunnel VPN (72.16.0.0/24)

Isolation réseau : Les VM2 et VM3 n'ont aucune interface sur vmbr0. Elles ne sont joignables que depuis le réseau LAN interne (20.20.20.0/24), lui-même accessible uniquement aux clients VPN authentifiés.

4.3 Activation du routage IP sur le Proxmox host

```
# Activation permanente du routage IPv4
echo 'net.ipv4.ip_forward=1' >> /etc/sysctl.conf
sysctl -p

# Vérification
sysctl net.ipv4.ip_forward
# Résultat attendu : net.ipv4.ip_forward = 1
```

5. Installation et Configuration de Pritunl

5.1 Préparation du système

```
# Mise à jour complète du système
sudo apt update && sudo apt upgrade -y
sudo apt install -y curl gnupg2 software-properties-common apt-transport-https ca-certificates
```

5.2 Installation de MongoDB 8.x

```
curl -fsSL https://www.mongodb.org/static/pgp/server-8.0.asc | \
    sudo gpg -o /usr/share/keyrings/mongodb-server-8.0.gpg --dearmor

echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-8.0.gpg ] \
    https://repo.mongodb.org/apt/ubuntu noble/mongodb-org/8.0 multiverse" | \
    sudo tee /etc/apt/sources.list.d/mongodb-org-8.0.list

sudo apt update && sudo apt install -y mongodb-org
sudo systemctl enable --now mongod
sudo systemctl status mongod
```

5.3 Installation de Pritunl

```
sudo tee /etc/apt/sources.list.d/pritunl.list << EOF
deb https://repo.pritunl.com/stable/apt noble main
EOF

gpg --keyserver hkp://keyserver.ubuntu.com --recv-keys E68D5086
gpg --armor --export E68D5086 | sudo tee /etc/apt/trusted.gpg.d/pritunl.asc

sudo apt update && sudo apt install -y pritunl
sudo systemctl enable --now pritunl

# Générer la clé de configuration initiale
sudo pritunl setup-key
```

5.4 Configuration initiale (interface Web)

7. Accéder à : https://<IP_FAILOVER> depuis votre navigateur
8. Saisir la setup-key affichée dans le terminal
9. Définir le mot de passe administrateur
10. Settings > Configuration : renseigner l'IP publique (<IP_FAILOVER>)
11. Organization : créer une organisation (ex. : MonEntreprise)
12. Server > Add Server : configurer le serveur avec les paramètres ci-dessous

Nom du serveur	vpn-server-prod
OpenVPN	Port 1194/UDP — activé
WireGuard	Port 51820/UDP — activé
Réseau tunnel	72.16.0.0/24 (auto-géré par Pritunl)

DNS	8.8.8.8, 1.1.1.1
Route LAN	Ajouter : 20.20.20.0/24 pour accès aux VM internes
Organisation liée	MonEntreprise

5.5 Configuration NAT et routage LAN

```
# Activation routage IPv4 sur VM Pritunl
sudo sysctl -w net.ipv4.ip_forward=1
echo 'net.ipv4.ip_forward=1' | sudo tee -a /etc/sysctl.conf

# Règles NAT : clients VPN (72.16.0.0/24) -> LAN (20.20.20.0/24)
sudo iptables -t nat -A POSTROUTING -s 72.16.0.0/24 -o ens19 -j MASQUERADE
sudo iptables -A FORWARD -i tun+ -o ens19 -j ACCEPT
sudo iptables -A FORWARD -i ens19 -o tun+ -m state --state RELATED,ESTABLISHED -j
ACCEPT

# Persistance des règles
sudo apt install -y iptables-persistent
sudo netfilter-persistent save
```

6. Durcissement Avancé de la VM Pritunl (CIS Benchmark Level 2)

⚠ Cette section présente la configuration théorique planifiée. Pour la configuration réelle appliquée sur la VM tourou (avec corrections et ajustements), voir la section 10.

Le durcissement avancé de la VM Pritunl suit le CIS Ubuntu Linux 24.04 LTS Benchmark (Level 2), complété par les recommandations ANSSI-PB-079 et NIST SP 800-123. Chaque sous-section précise la référence CIS applicable.

6.1 Gestion des mises à jour système (CIS 1.8)

```
sudo apt update && sudo apt upgrade -y && sudo apt dist-upgrade -y
sudo apt autoremove --purge -y

# Mises à jour automatiques de sécurité
sudo apt install -y unattended-upgrades apt-listchanges
sudo dpkg-reconfigure -plow unattended-upgrades
sudo unattended-upgrade --dry-run --debug  # Test
```

6.2 Sécurisation SSH (CIS 5.2)

La configuration SSH est un point critique du durcissement. Les paramètres ci-dessous désactivent l'authentification par mot de passe, restreignent les algorithmes cryptographiques aux suites approuvées par le NIST et limitent l'accès aux utilisateurs autorisés.

Configuration /etc/ssh/sshd_config — Paramètres de sécurité
Protocol 2
Port 2222 # Port non standard (adapter UFW)
PermitRootLogin no # CIS 5.2.10
PasswordAuthentication no # CIS 5.2.11 — Clé SSH obligatoire
PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys
PermitEmptyPasswords no
X11Forwarding no # CIS 5.2.6
MaxAuthTries 3 # CIS 5.2.7
LoginGraceTime 30
ClientAliveInterval 300
ClientAliveCountMax 2
AllowUsers <votre_utilisateur>
Banner /etc/issue.net
Algorithmes cryptographiques approuvés NIST/CIS
KexAlgorithms curve25519-sha256,diffie-hellman-group16-sha512
Ciphers aes256-gcm@openssh.com,chacha20-poly1305@openssh.com
MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com

```
sudo sshd -t && sudo systemctl restart ssh
```

6.3 Pare-feu UFW (CIS 3.5)

⚠ ATTENTION — Conflit UFW / iptables-persistent : choisir UN SEUL outil

UFW et iptables-persistent sont deux systèmes de gestion du firewall qui **ne peuvent pas coexister** sur le même serveur. Les utiliser simultanément provoque des conflits de règles, des comportements imprévisibles et peut verrouiller l'accès SSH.

Option A — UFW (plus simple, recommandé pour débutants) :

- Utiliser uniquement les commandes sudo ufw allow / deny
- Ne pas installer iptables-persistent en même temps

Option B — iptables-persistent (plus puissant, utilisé sur tourou) :

- Utiliser uniquement les commandes sudo iptables + sudo netfilter-persistent save
- Désactiver UFW : sudo ufw disable ou sudo systemctl disable ufw
- Pritunl gère ses propres règles iptables automatiquement — ne pas les modifier manuellement

Sur tourou : iptables-persistent a été choisi. UFW est désactivé (les chaînes UFW restent présentes en mémoire mais sont vides). Ne pas réactiver UFW sans désactiver iptables-persistent au préalable.

Configuration UFW — Politique stricte

```
sudo apt install -y ufw

# Politique par défaut
sudo ufw default deny incoming
sudo ufw default allow outgoing

# Règles d'accès
sudo ufw allow 2222/tcp        # SSH (port personnalisé)
sudo ufw allow 1194/udp        # OpenVPN
sudo ufw allow 51820/udp       # WireGuard
sudo ufw allow 443/tcp         # Interface Web Pritunl (HTTPS)

sudo ufw enable
sudo ufw status verbose
```

6.4 Fail2ban — Protection brute-force (CIS Best Practice)

Configuration /etc/fail2ban/jail.local

```
[DEFAULT]
bantime  = 3600
findtime = 600
maxretry = 5
backend  = systemd

[sshd]
```

```
enabled  = true
port     = 2222
logpath  = /var/log/auth.log
maxretry = 3
bantime  = 86400
```

```
sudo systemctl enable --now fail2ban
sudo fail2ban-client status sshd
```

6.5 AIDE — Intégrité du système de fichiers (CIS 1.4.1)

AIDE (Advanced Intrusion Detection Environment) établit une base de référence cryptographique (SHA-256 + SHA-512) de l'ensemble du système de fichiers et alerte en cas de modification non autorisée.

Installation et planification AIDE

```
sudo apt install -y aide aide-common

# Initialiser la base de données de référence
sudo aideinit
sudo cp /var/lib/aide/aide.db.new /var/lib/aide/aide.db

# Vérification quotidienne automatique (4h00)
echo '0 4 * * * root /usr/bin/aide --check | \
    mail -s "[AIDE] Rapport $(hostname) $(date +%F)" admin@example.com' \
    | sudo tee /etc/cron.d/aide-daily

# Vérification manuelle immédiate
sudo aide --check
```

6.6 Logwatch — Rapports de surveillance (Best Practice)

Configuration /etc/logwatch/conf/logwatch.conf

```
Output   = mail
Format   = html
MailTo   = admin@example.com
MailFrom = logwatch@$(hostname)
Range    = yesterday
Detail   = Med
Service  = All
```

```
sudo apt install -y logwatch
sudo logwatch --detail High --range today --output stdout | head -80  # Test
```

6.7 Auditd — Audit noyau Linux (CIS 4.1)

Règles /etc/audit/rules.d/hardening.rules

```
# Surveillance des fichiers sensibles
-w /etc/passwd -p wa -k identity
-w /etc/shadow -p wa -k identity
-w /etc/sudoers -p wa -k sudoers
-w /var/log/auth.log -p wa -k auth_log
-w /etc/ssh/sshd_config -p wa -k sshd_config

# Surveillance des appels système
-a always,exit -F arch=b64 -S execve -k exec
-a always,exit -F arch=b64 -S open -F exit=-EACCES -k access
-a always,exit -F arch=b64 -S chmod -S fchmod -k perm_mod
-a always,exit -F arch=b64 -S mount -k mounts

# Rendre les règles immuables (reboot requis pour modifier)
-e 2
```

```
sudo augenrules --load
sudo systemctl enable --now auditd
sudo ausearch -k identity | tail -20  # Vérifier les événements
```

6.8 Durcissement du noyau — sysctl (CIS 3.x)

Fichier /etc/sysctl.d/99-hardening.conf

```
# Protection anti-spoofing IP
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1

# Désactivation des ICMP redirects
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv6.conf.all.accept_redirects = 0

# Désactivation du source routing
net.ipv4.conf.all.accept_source_route = 0
net.ipv6.conf.all.accept_source_route = 0

# Protection SYN flood
net.ipv4.tcp_syncookies = 1

# Désactivation IPv6 (si inutilisé)
net.ipv6.conf.all.disable_ipv6 = 1
```

```
# ASLR — Address Space Layout Randomization
kernel.randomize_va_space = 2

# Restrictions d'accès aux informations noyau
kernel.dmesg_restrict = 1
kernel.kptr_restrict = 2
```

```
sudo sysctl -p /etc/sysctl.d/99-hardening.conf
```

6.9 Gestion des comptes et privilèges (CIS 5.x)

Politique de mots de passe — /etc/security/pwquality.conf

```
# Installation
sudo apt install -y libpam-pwquality

# Paramètres
minlen   = 14
minclass = 4
maxrepeat = 3
dcredit  = -1
ucredit  = -1
ocredit  = -1
lcredit  = -1
```

```
# Expiration des mots de passe
sudo chage --maxdays 90 --mindays 7 --warndays 14 <utilisateur>

# Comptes sans mot de passe (à corriger si résultat non vide)
sudo awk -F: '($2 == "") {print $1}' /etc/shadow

# Désactiver les comptes de service inutilisés
sudo usermod -s /sbin/nologin <compte_service>
```

6.10 Désactivation des services inutiles (CIS 2.x)

```
sudo systemctl disable --now avahi-daemon   # mDNS inutile
sudo systemctl disable --now cups            # Impression
sudo systemctl disable --now bluetooth       # Bluetooth
sudo systemctl disable --now rpcbind         # NFS client

# Vérifier les ports en écoute
sudo ss -tlnp
```

6.11 AppArmor — Contrôle d'accès mandataire (CIS 1.6.1)

```
sudo apt install -y apparmor apparmor-utils
sudo systemctl enable --now apparmor
sudo aa-enforce /etc/apparmor.d/*
sudo aa-status | grep enforce
```

6.12 Banner légal d'avertissement (CIS 5.1)

Contenu du fichier /etc/issue.net

```
**************************************************************************
*    AVERTISSEMENT — Systeme d'information a acces restreint             *
*    Tout acces non autorise est strictement interdit et poursuivi       *
*    conformement aux dispositions legales en vigueur.                   *
*    Toutes les activites sont enregistrees et auditees.                 *
**************************************************************************
```

7. Vérification, Tests et Validation

7.1 Tests de connectivité VPN

13. Connecter un client externe avec OpenVPN Connect ou le client WireGuard
14. Vérifier l'attribution d'une IP dans 72.16.0.0/24 : ip addr show tun0 ou ip addr show wg0
15. Pinger la gateway LAN : ping 20.20.20.1
16. Pinger VM2 : ping 10.10.10.2 et VM3 : ping 10.10.10.3
17. Se connecter en SSH sur VM2 : ssh user@10.10.10.2 (via VPN)
18. Vérifier qu'une connexion SSH à VM2 SANS VPN est impossible (firewall)

7.2 Audit de conformité — Lynis

```
sudo apt install -y lynis
sudo lynis audit system
# Rapport : /var/log/lynis.log
sudo lynis show summary
# Score cible après durcissement : > 75 / 100
```

7.3 Checklist de validation finale

Point de contrôle	CIS Réf.	Statut
SSH : root login désactivé	CIS 5.2.10	[ ] Validé
SSH : authentification par clé uniquement	CIS 5.2.11	[ ] Validé
UFW actif et configuré	CIS 3.5	[ ] Validé
Fail2ban actif (SSH)	Best Practice	[ ] Validé
AIDE initialisé + cron configuré	CIS 1.4.1	[ ] Validé
Auditd actif avec règles CIS	CIS 4.1	[ ] Validé
Logwatch configuré (email)	Best Practice	[ ] Validé
Sysctl durcissement appliqué	CIS 3.x	[ ] Validé
AppArmor enforce actif	CIS 1.6.1	[ ] Validé
MAJ automatiques activées	CIS 1.8	[ ] Validé
Services inutiles désactivés	CIS 2.x	[ ] Validé
Politique de mots de passe configurée	CIS 5.4	[ ] Validé
Banner légal configuré	CIS 5.1	[ ] Validé
Lynis score > 75/100	Audit	[ ] Validé
SSH via VPN vers VM2 et VM3	Fonctionnel	[ ] Validé
Accès SSH sans VPN bloqué	Sécurité	[ ] Validé

7.4 Connexion à l’interface Web Pritunl

Pritunl expose une interface web HTTPS sur l'IP Failover OVH. Cette section décrit la procédure complète de première connexion, la configuration initiale et la navigation dans le tableau de bord.

7.4.1 Générer le mot de passe par défaut

Avant d'accéder à l'interface, récupérer la setup-key et le mot de passe initial depuis la VM tourou :

```
# &#xc9;tape 1 — G&#xe9;n&#xe9;rer la cl&#xe9; de configuration initiale (usage
unique)

sudo pritunl setup-key
# Exemple de sortie : ef5e2ae11293466daba9c20d12cfc50e

# &#xc9;tape 2 — G&#xe9;n&#xe9;rer le mot de passe administrateur par d&#xe9;faut
sudo pritunl default-password
# Sortie :
# Administrator default password:
#   username: "pritunl"

#   password: "i8g1JLkZQmdl"
```

⚠ La setup-key est à usage unique. Une fois saisie dans l'interface web, elle ne peut plus être réutilisée. Conserver le mot de passe par défaut jusqu'à la finalisation du setup.

7.4.2 Accès et setup initial via le navigateur

Ouvrir un navigateur et suivre les étapes suivantes :

- Naviguer vers : https://<IP_FAILOVER> (ou http://<IP_FAILOVER> en premier accès)
- Accepter l'avertissement de certificat auto-signé — normal lors de la première connexion
- Saisir la setup-key générée à l'étape précédente, vérifier que l'URI MongoDB pointe sur localhost, puis cliquer sur Save
- Sur la page de login, saisir le username pritunl et le mot de passe par défaut généré par default-password, puis cliquer sur Sign In

7.4.3 Configuration initiale — Initial Setup Dialog

Apres la première connexion, Pritunl affiche une fenêtre Initial Setup. Renseigner les champs suivants :

Paramètre	Valeur / Action
New Password	Définir un mot de passe fort (≥ 16 caractères, mixte) — remplace le mot de passe par défaut

Public IP / Verify	Vérifier que l'IP publique affichée correspond à <IP_FAILOVER>
Let's Encrypt Domain	Optionnel — renseigner votre domaine DNS (ex. : vpn.tourou.com) pour certificat TLS automatique
Email (notifications)	admin@example.com — alertes système Pritunl

- Cliquer sur Save pour appliquer la configuration — l'interface est maintenant opérationnelle

ℹ Pour le certificat Let's Encrypt, le port 80/tcp doit être temporairement ouvert (sudo ufw allow 80/tcp) et le domaine doit pointer sur l'IP Failover. Une fois le certificat généré, le port 80 peut être refermé.

7.4.4 Structure de Pritunl — Concepts clés

Avant de créer des ressources, il est essentiel de comprendre la hiérarchie Pritunl :

Concept	Rôle
Server	Point de terminaison VPN — héberge OpenVPN/WireGuard, gère le routage et le chiffrement
Organization	Espace de gestion des utilisateurs — CA indépendante qui émet les certificats clients
User	Profil individuel avec certificat unique — se connecte via le profil .tar généré par Pritunl

⚠ Ordre obligatoire : créer l'Organisation AVANT le Serveur, puis attacher l'Organisation au Serveur AVANT de le démarrer. Sans organisation attachée, le bouton Start Server est inactif.

7.5 Création du Serveur Virtuel VPN (OpenVPN + WireGuard simultanés)

Pritunl permet de faire fonctionner OpenVPN et WireGuard en parallèle sur le même serveur virtuel. Les deux protocoles partagent la même organisation et les mêmes utilisateurs — chaque client choisit son protocole lors de l'import du profil.

7.5.1 Créer l'Organisation

L'organisation est le conteneur qui regroupe les utilisateurs VPN. Elle est obligatoire avant de pouvoir démarrer un serveur.

- Aller dans l'onglet Users dans la barre de navigation

- Cliquer sur Add Organization
- Saisir le nom de l'organisation (ex. : TourouCorp)
- Cliquer sur Add — l'organisation apparaît dans la liste avec le statut actif

7.5.2 Créer le Serveur dual-protocole

Aller dans l'onglet Servers → cliquer sur Add Server. Renseigner les paramètres suivants dans le formulaire :

Paramètre	Valeur / Configuration
Name	vpn-prod-dual
DNS Server	8.8.8.8 (ou 1.1.1.1) — serveur DNS attribué aux clients
Port	1194 — port OpenVPN principal
Protocol	UDP — recommandé pour la performance
Virtual Network	72.16.0.0/24 — sous-réseau privé attribué aux clients VPN
WireGuard	Activer le toggle WireGuard → port automatique : 51820/UDP
Route LAN interne	Ajouter manuellement : 20.20.20.0/24 pour accès aux VM internes

- Cliquer sur ADD pour enregistrer la configuration du serveur
- Vérifier le message : Successfully added server.

> **i** Le toggle WireGuard active le dual-protocole : OpenVPN écoute sur 1194/UDP et WireGuard sur 51820/UDP — les deux sont actifs simultanément. Chaque client choisit son protocole lors du téléchargement du profil dans le client Pritunl.

7.5.3 Configurer l'authentification et la durée de session

Dans l'interface de création/édition du serveur (icône crayon), section Advanced Settings, configurer :

Paramètre	Valeur / Justification
Auth Token Duration	10080 minutes = 7 jours (1 semaine) — durée de session avant re-authentification
Two-Factor Authentication (OTP)	Activer — code TOTP requis à chaque nouvelle session (OpenVPN et WireGuard)
User Authentication	Activer — chaque client doit avoir un compte Pritunl valide
SSO (Single Sign-On)	Optionnel — intégration Google / Okta / SAML si disponible

✓ 10080 minutes = 7 j x 24 h x 60 min = 10080. Après expiration, le client Pritunl redemande automatiquement le code TOTP. L'administrateur peut révoquer une session à tout moment depuis Users → Sessions.

Vérification de la durée de session via Settings (valeur globale en secondes) :

```
# Settings > Token Expiration

# 604800 secondes = 7 jours (valeur globale)

# Auth Token Duration dans le serveur = 10080 minutes (prioritaire)
```

7.5.4 Attacher l'Organisation au Serveur

Le serveur ne peut pas démarrer sans organisation attachée. Suivre la procédure ci-dessous :

- Dans l'onglet Servers, sélectionner le serveur vpn-prod-dual
- Cliquer sur Attach Organization
- Dans la fenêtre, sélectionner l'organisation (ex. : TourouCorp) et le serveur cible
- Cliquer sur Attach pour confirmer le lien Organisation ↔ Serveur

7.5.5 Démarrer le Serveur et vérifier les ports

- Cliquer sur Start Server (triangle vert) dans l'onglet Servers
- Vérifier que le statut passe à Online (point vert) — le serveur est actif

Vérifier que les deux protocoles sont bien en écoute sur la VM tourou :

```
# V&#xe9;rification des ports UDP en &#xe9;coute

sudo ss -ulnp | grep -E '1194|51820'
# R&#xe9;sultat attendu :
# UNCONN  0.0.0.0:1194   (openvpn)
# UNCONN  0.0.0.0:51820  (wireguard)

# Autoriser les ports dans UFW si ce n'est pas d&#xe9;j&#xe0; fait
sudo ufw allow 1194/udp    # OpenVPN
sudo ufw allow 51820/udp   # WireGuard

sudo ufw reload
```

⚠ Chaque serveur Pritunl utilise un port spécifique. Vérifier dans Servers le port configuré et s'assurer qu'il est autorisé dans UFW. Sans cette règle, les clients ne peuvent pas établir la connexion VPN.

7.6 Gestion des Utilisateurs VPN

Chaque utilisateur possède ses propres certificats et un profil de connexion (.tar) qui contient à la fois la configuration OpenVPN et WireGuard. Pritunl les génère automatiquement.

7.6.1 Créer un Utilisateur

- Aller dans l'onglet Users
- Cliquer sur Add User dans l'organisation concernée

Champ	Valeur
Name	Nom de l'utilisateur (ex. : eric.affanou)
Organization	Sélectionner l'organisation (ex. : TourouCorp)
Email	Adresse email — reçoit le lien de téléchargement du profil
PIN	Code PIN fort — requis à chaque connexion VPN (s'ajoute au TOTP)

- Cliquer sur Add — le compte est créé et apparaît dans la liste de l'organisation

> **i** Le PIN est requis à chaque connexion VPN depuis le client. Il s'ajoute au code TOTP si le 2FA est activé sur le serveur. Ne pas confondre avec le mot de passe du compte administrateur Pritunl.

7.6.2 Télécharger et distribuer le profil

Pritunl génère un profil universel (.tar) contenant la configuration OpenVPN et WireGuard. Deux méthodes de distribution :

Méthode A — Téléchargement direct depuis le tableau de bord :

- Dans Users, cliquer sur l'icône Download Profile (flèche vers le bas) à droite de l'utilisateur
- Sélectionner le serveur vpn-prod-dual dans la liste
- Télécharger le fichier .tar et le transmettre à l'utilisateur de manière sécurisée (canal chiffré)

Méthode B — Lien public temporaire (Pritunl Client) :

- Cliquer sur Link dans les options de l'utilisateur
- Copier le lien généré — l'utilisateur peut l'ouvrir directement dans le client Pritunl pour importer son profil
- Le lien expire automatiquement après utilisation

> **i** Le client Pritunl (Windows, macOS, Linux, iOS, Android) gère nativement les deux protocoles et la re-authentification à l'expiration de session. Téléchargement : https://client.pritunl.com

7.6.3 Première connexion utilisateur — Procédure TOTP

Lors de la première connexion, l'utilisateur doit configurer son authentificateur TOTP :

- Télécharger et installer le client Pritunl depuis https://client.pritunl.com
- Ouvrir Pritunl Client → cliquer sur Import en haut du menu
- Cliquer sur Browse, sélectionner le fichier .tar téléchargé, puis cliquer sur Import
- Vérifier les informations (nom d'utilisateur, nom du serveur) puis cliquer sur Connect
- Lors de la première connexion : scanner le QR code affiché avec Google Authenticator, Aegis ou tout client TOTP compatible
- Saisir le PIN défini à la création de l'utilisateur pour valider la connexion

✓ Une fois connecté, vérifier l'IP attribuée dans 72.16.0.0/24 et tester ping 20.20.20.1 pour confirmer l'accès au réseau LAN interne. La session est valide 7 jours sans nouvelle authentification.

7.6.4 Révoquer un utilisateur

Pour supprimer l'accès d'un utilisateur immédiatement :

- Aller dans Users → sélectionner l'utilisateur à révoquer
- Cliquer sur Delete Selected en haut à droite du tableau de bord
- Confirmer la suppression — tous les tokens et certificats de cet utilisateur sont immédiatement invalidés

⚠ La suppression d'un utilisateur révoque instantanément tous ses profils VPN actifs sur OpenVPN et WireGuard. Les sessions en cours sont terminées sans préavis.

8. Retour d'expérience — Déploiement réel sur tourou

Cette section documente les problèmes rencontrés et résolus lors du déploiement réel de Pritunl sur la VM tourou (Proxmox / OVH). Elle sert de guide pour éviter les mêmes écueils lors de futures installations.

8.1 Problèmes rencontrés et résolus

Problème	Cause / Solution
MongoDB 7.0 — dépôt introuvable (404)	MongoDB 7.0 n'existe pas pour Ubuntu 24.04 noble. Solution : utiliser MongoDB 8.0
mongod crash signal=ILL	CPU Proxmox de type kvm64 n'expose pas AVX. Solution : passer le CPU en type host dans Proxmox
pritunl default-password — Empty host	mongodb_uri vide dans /etc/pritunl.conf. Solution : sudo pritunl set-mongodb mongodb://localhost:27017/pritunl
WireGuard — FileNotFoundError: wg	wg non trouvé dans le PATH de Pritunl. Solution : sudo ln -sf /usr/bin/wg /usr/local/bin/wg
tun0 absent — OpenVPN sans IP client	Module TUN non exposé à la VM. Solution : activer TUN dans Proxmox + sudo modprobe tun
UFW inactif au démarrage	UFW non activé après installation. Solution : sudo ufw enable + ouvrir les ports
Port WireGuard dynamique	Pritunl utilise un port aléatoire différent de 51820. Solution : sudo wg show pour trouver le port réel
NAT WireGuard manquant	Pritunl ne pousse pas les routes WireGuard automatiquement. Solution : règles iptables manuelles

8.2 Configuration CPU Proxmox — Prérequis AVX

MongoDB 8.0 requiert les instructions AVX du processeur. Par défaut, Proxmox expose un CPU kvm64 qui ne les fournit pas. Cette configuration doit être appliquée AVANT l'installation de MongoDB.

Depuis le shell Proxmox (remplacer 101 par l'ID réel de la VM tourou) :

```
# Changer le type CPU pour exposer les instructions du processeur physique

qm set 101 --cpu host
qm reboot 101

# V&#xe9;rifier depuis tourou apr&#xe8;s red&#xe9;marrage
grep -o 'avx[^ ]*' /proc/cpuinfo | sort -u

# R&#xe9;sultat attendu : avx (et id&#xe9;alement avx2)
```

⚠ Ne pas utiliser qm set <VMID> --args '-device tun' — ce paramètre est invalide et empêche la VM de démarrer (QEMU exited with code 1).

8.3 Activation du module TUN — Prérequis OpenVPN

OpenVPN nécessite l'interface tun0. Si /dev/net/tun existe mais que tun0 n'est pas créé au démarrage du serveur, charger le module manuellement :

```
# Charger le module TUN

sudo modprobe tun

# Rendre le chargement permanent
echo 'tun' | sudo tee /etc/modules-load.d/tun.conf

# V&#xe9;rifier
lsmod | grep tun

ls -la /dev/net/tun
```

✓ Résultat attendu : crw-rw-rw- 1 root root 10, 200 /dev/net/tun — tun0 sera créé automatiquement par OpenVPN au démarrage du serveur Pritunl.

8.4 Configuration NAT complète — OpenVPN + WireGuard + LAN

Pritunl gère le NAT OpenVPN automatiquement, mais pas le NAT WireGuard ni le NAT du réseau LAN interne. Les règles suivantes sont obligatoires pour un fonctionnement complet :

```
# Activation routage IPv4 (v&#xe9;rifier d'abord)

sysctl net.ipv4.ip_forward
# Si = 0, activer :
sudo sysctl -w net.ipv4.ip_forward=1
echo 'net.ipv4.ip_forward=1' | sudo tee -a /etc/sysctl.conf

# NAT OpenVPN (72.16.0.0/24) vers Internet et LAN
sudo iptables -t nat -A POSTROUTING -s 72.16.0.0/24 -o ens18 -j MASQUERADE
sudo iptables -t nat -A POSTROUTING -s 72.16.0.0/24 -o ens19 -j MASQUERADE
sudo iptables -A FORWARD -i tun0 -o ens19 -j ACCEPT
sudo iptables -A FORWARD -i ens19 -o tun0 -m state --state RELATED,ESTABLISHED -j
ACCEPT

# NAT WireGuard (72.17.0.0/24) vers Internet et LAN
sudo iptables -t nat -A POSTROUTING -s 72.17.0.0/24 -o ens18 -j MASQUERADE
sudo iptables -t nat -A POSTROUTING -s 72.17.0.0/24 -o ens19 -j MASQUERADE
sudo iptables -A FORWARD -i wg0 -o ens19 -j ACCEPT
sudo iptables -A FORWARD -i ens19 -o wg0 -m state --state RELATED,ESTABLISHED -j
ACCEPT

# NAT LAN interne (20.20.20.0/24) vers Internet
sudo iptables -t nat -A POSTROUTING -s 20.20.20.0/24 -o ens18 -j MASQUERADE

# Persitance des r&#xe8;gles
```

```
sudo apt install -y iptables-persistent
sudo netfilter-persistent save

# V&#xe9;rification

sudo iptables -t nat -L POSTROUTING -n -v
```

Source / Sortie	Rôle
72.16.0.0/24 → ens18	Clients OpenVPN → Internet
72.16.0.0/24 → ens19	Clients OpenVPN → LAN 20.20.20.0/24
72.17.0.0/24 → ens18	Clients WireGuard → Internet
72.17.0.0/24 → ens19	Clients WireGuard → LAN 20.20.20.0/24
20.20.20.0/24 → ens18	VM LAN internes → Internet

8.5 Port WireGuard dynamique — Identification et ouverture

Pritunl attribue un port WireGuard aléatoire différent du port configuré dans l'interface web. Il est indispensable d'identifier le port réel et de l'ouvrir dans le firewall.

```
# Identifier le port WireGuard r&#xe9;ellement utilis&#xe9;

sudo wg show
# Chercher : listening port: XXXXX

# Exemple de sortie :
# interface: wg0
#   listening port: 14623

# Ouvrir le port dans UFW
sudo ufw allow 14623/udp
sudo ufw reload

# V&#xe9;rification

sudo ufw status | grep udp
```

⚠ Ce port change à chaque redémarrage du serveur Pritunl. Après chaque redémarrage, vérifier sudo wg show et mettre à jour la règle UFW si nécessaire.

8.6 Limitation WireGuard — Routes LAN non poussées automatiquement

Contrairement à OpenVPN, WireGuard ne supporte pas le mécanisme 'push route'. Les routes additionnelles (ex. : 20.20.20.0/24) configurées dans l'interface web Pritunl sont ignorées pour WireGuard.

Fonctionnalité	OpenVPN vs WireGuard

Routes LAN automatiques	OpenVPN : ☑ push route | WireGuard : ✗ NAT obligatoire
Accès Internet	OpenVPN : ☑ auto | WireGuard : ☑ avec NAT
Accès LAN 20.20.20.0/24	OpenVPN : ☑ route | WireGuard : ☑ MASQUERADE ens19
Auth TOTP	Les deux : ☑ supporté
Port	OpenVPN : 1194/UDP fixe | WireGuard : dynamique (wg show)

ℹ Cette limitation est inhérente au protocole WireGuard et ne sera pas corrigée par Pritunl. La solution par NAT (section 8.4) est la méthode recommandée et fonctionnelle.

8.7 Checklist de validation WireGuard

Point de contrôle	Commande de vérification
wg command disponible	which wg && wg --version
Module TUN chargé	lsmod | grep tun
Interface wg0 UP	ip link show wg0
Port WireGuard ouvert UFW	sudo wg show + sudo ufw status | grep udp
NAT WireGuard actif	sudo iptables -t nat -L POSTROUTING | grep 72.17
FORWARD WireGuard actif	sudo iptables -L FORWARD | grep wg
Client WireGuard connecté	sudo wg show (peers avec handshake récent)
Client ping LAN	depuis client : ping 20.20.20.1
Client accès Internet	depuis client : ping 8.8.8.8
Règles persistées	sudo netfilter-persistent save

9. Durcissement SSH — Configuration réelle sur tourou

Cette section documente la configuration SSH appliquée réellement sur la VM tourou, basée sur le retour d'expérience du déploiement. L'objectif est d'éliminer tout accès par mot de passe et de sécuriser l'accès administrateur.

9.1 Stratégie d'accès adoptée

Paramètre	Valeur
Port SSH	**2235 (non standard — évite les scans automatiques)**
Utilisateur de connexion	**alibora (compte non-root avec sudo)**
Authentification	**Clé SSH ED25519 uniquement — mot de passe désactivé**
Accès root direct	**Interdit (PermitRootLogin no)**
Accès root indirect	**sudo su depuis alibora — demande le mot de passe alibora**
Clé publique	**ED25519 — générée sur le poste Windows (VICTUS)**

9.2 Génération de la clé SSH — Poste Windows

Exécuter depuis PowerShell sur le poste client :

```
# G&#xe9;n&#xe9;rer la cl&#xe9; ED25519

ssh-keygen -t ed25519 -C "tourou-admin"
# Appuyer Entr&#xe9;e pour les valeurs par d&#xe9;faut

# Afficher la cl&#xe9; publique &#xe0; copier sur le serveur
cat ~/.ssh/id_ed25519.pub

# Exemple : ssh-ed25519 AAAAC3Nz... victus@LAPTOP-FAINJCAQ
```

9.3 Installation de la clé sur tourou

```
# Cr&#xe9;er le r&#xe9;pertoire SSH pour alibora

mkdir -p /home/alibora/.ssh
chmod 700 /home/alibora/.ssh

# Ajouter la cl&#xe9; publique
```

```
echo "ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIGgEkTKwzVSDb9s0t2m1P5Fom63Ga3L5jfC6CJfTPTxH victus@LAPTOP-
FAINJCAQ" \
  >> /home/alibora/.ssh/authorized_keys

chmod 600 /home/alibora/.ssh/authorized_keys

chown -R alibora:alibora /home/alibora/.ssh
```

9.4 Configuration SSH durcie

Fichier : /etc/ssh/sshd_config.d/hardening.conf

```
Port 2235

PermitRootLogin no
PasswordAuthentication no
PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys
MaxAuthTries 3
ClientAliveInterval 300
ClientAliveCountMax 2
X11Forwarding no

AllowTcpForwarding no
```

⚠ Le fichier 50-cloud-init.conf écrase les paramètres SSH par défaut. Il doit être neutralisé : sudo mv /etc/ssh/sshd_config.d/50-cloud-init.conf /etc/ssh/sshd_config.d/50-cloud-init.conf.bak

⚠ UFW reste actif en arrière-plan même si la commande ufw n'est pas disponible. Pour ouvrir le port 2235, insérer la règle directement dans la chaîne UFW : sudo iptables -I ufw-before-input 1 -p tcp --dport 2235 -j ACCEPT

9.5 Procédure de connexion

```
# Connexion normale (depuis Windows)

ssh -i ~/.ssh/id_ed25519 -p 2235 alibora@<IP_FAILOVER>

# Devenir root (mot de passe alibora requis)
sudo su

# Connexion root directe — REFUS&#xc9;E
ssh -i ~/.ssh/id_ed25519 -p 2235 root@<IP_FAILOVER>

# R&#xe9;sultat : Permission denied (publickey)
```

✓ Test validé : root@<IP_FAILOVER>: Permission denied (publickey) — root SSH complètement bloqué. Seul alibora peut se connecter par clé puis escalader avec sudo su + mot de passe.

10. Durcissement Avancé CIS Level 2 — Configuration réelle

> ℹ Note : La section 6 décrit la configuration théorique prévue (CIS Benchmark). Cette section (10) documente la configuration réelle appliquée sur tourou lors du déploiement. Les commandes ici sont celles réellement exécutées et validées, avec les corrections apportées par rapport au plan initial.

Cette section documente le durcissement complet appliqué sur tourou lors du déploiement réel. Toutes les configurations sont actives et validées.

10.1 Fail2ban — Protection SSH et Pritunl

Fail2ban protège contre les attaques par force brute. Le rapport Logwatch a confirmé 17 IPs bannées le premier jour de déploiement.

```
sudo apt install -y fail2ban


# Configuration /etc/fail2ban/jail.local
[DEFAULT]
bantime  = 3600
findtime = 600
maxretry = 5
backend  = systemd

[sshd]
enabled  = true
port     = 2235
logpath  = %(sshd_log)s

[pritunl]
enabled  = true
port     = 443
logpath  = /var/log/pritunl.log
maxretry = 5
bantime  = 3600

filter   = pritunl
```

```
# Filtre Pritunl /etc/fail2ban/filter.d/pritunl.conf

[Definition]
failregex = \[.*\]\[ERROR\].*Failed to authenticate.*&lt;HOST&gt;
            \[.*\]\[ERROR\].*User auth.*failed.*&lt;HOST&gt;
ignoreregex =

# D&#xe9;marrer
sudo systemctl enable --now fail2ban

sudo fail2ban-client status
```

10.2 Auditd — Journalisation des événements système

```
sudo apt install -y auditd audispd-plugins


# R&#xe8;gles /etc/audit/rules.d/hardening.rules
-w /etc/passwd -p wa -k identity
-w /etc/shadow -p wa -k identity
-w /etc/sudoers -p wa -k sudoers
-w /var/log/pritunl.log -p rwa -k pritunl
-w /etc/pritunl.conf -p wa -k pritunl
-w /etc/ssh/sshd_config -p wa -k sshd
-w /sbin/iptables -p x -k firewall
-a always,exit -F arch=b64 -S execve -k exec_commands

sudo systemctl enable --now auditd

sudo service auditd restart
```

10.3 Sysctl Hardening — Paramètres noyau

Fichier : /etc/sysctl.d/99-hardening.conf

```
# Routage IPv4 (conserv&#xe9; pour VPN)

net.ipv4.ip_forward = 1

# Protection r&#xe9;seau
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.tcp_syncookies = 1
net.ipv4.conf.all.log_martians = 1

# Protection m&#xe9;moire
kernel.randomize_va_space = 2
kernel.dmesg_restrict = 1
kernel.kptr_restrict = 2

# IPv6
net.ipv6.conf.all.accept_redirects = 0
net.ipv6.conf.default.accept_redirects = 0

# Appliquer

sudo sysctl -p /etc/sysctl.d/99-hardening.conf
```

10.4 AIDE — Détection d’intrusion par intégrité de fichiers

```
sudo apt install -y aide


# Initialisation de la base de r&#xe9;f&#xe9;rence (2-3 minutes)
sudo aideinit
sudo cp /var/lib/aide/aide.db.new /var/lib/aide/aide.db
```

```
# V&#xe9;rification manuelle
sudo aide --check

# Cron quotidien &#xe0; 3h du matin
echo "0 3 * * * root /usr/bin/aide --check >> /var/log/aide.log 2>&amp;1" \
  | sudo tee /etc/cron.d/aide-check
```

ℹ La base AIDE initialisée sur tourou fait 47 MB et couvre l'intégralité du système de fichiers. Toute modification non autorisée de fichiers système sera détectée au prochain check quotidien.

10.5 Logwatch — Rapports quotidiens

```
sudo apt install -y logwatch


# Configuration /etc/logwatch/conf/logwatch.conf
Output = stdout
Format = text
Range = yesterday
Detail = Med
Service = All

# Test

sudo logwatch --output stdout --range today --detail med
```

ℹ Le premier rapport Logwatch a révélé plus de 2996 tentatives de connexion SSH invalides et 17 IPs bannées par Fail2ban en une journée — ce qui confirme l'importance du changement de port SSH et du durcissement réalisé.

10.6 Firewall iptables — Règles finales

⚠ Sur tourou, iptables-persistent est l'outil retenu (Option B). UFW est désactivé pour éviter tout conflit. Voir section 6.3 pour le détail du choix UFW vs iptables-persistent.

Numéro	Port/Protocole	Service	Statut
1-6	Pritunl (wg0/tun0)	Règles VPN auto	☑ Géré par Pritunl
7-12	UFW chains	Héritage UFW	☑ Transparent
13	2235/tcp	SSH alibora	☑ Ouvert
14	443/tcp	Pritunl HTTPS	☑ Ouvert
15	1194/udp	OpenVPN	☑ Ouvert
16	14623/udp	WireGuard (port réel)	☑ Ouvert
17	loopback	Trafic local	☑ Accepté
18	RELATED,EST	Sessions établies	☑ Accepté

19-20	tun0/wg0 DROP	Isolation VPN	☑ Géré par Pritunl
21	ALL DROP	Tout le reste	☑ Bloqué

```
# Sauvegarder les r&#xe8;gles de mani&#xe8;re permanente
sudo netfilter-persistent save

# V&#xe9;rifier
sudo iptables -L INPUT -n -v --line-numbers
```

⚠ Le port WireGuard 14623 est dynamique — il est régénéré à chaque redémarrage de Pritunl. Après tout redémarrage, vérifier sudo wg show et mettre à jour la règle iptables si le port change.

10.7 Checklist durcissement final — tourou

Contrôle	Commande	Résultat attendu
SSH port 2235	sudo ss -tlnp | grep ssh	LISTEN 0.0.0.0:2235
Root SSH bloqué	ssh -p 2235 root@IP	Permission denied
Auth clé uniquement	sudo sshd -T | grep passwordauth	passwordauthentication no
Fail2ban actif	sudo fail2ban-client status	Jail list: sshd, pritunl
Auditd actif	sudo systemctl is-active auditd	active
Sysctl appliqué	sysctl net.ipv4.ip_forward	1
AIDE base initialisée	ls -lh /var/lib/aide/	aide.db 47MB
Logwatch configuré	sudo logwatch --range today	Rapport généré
Ports fermés	sudo iptables -L INPUT | tail -3	DROP all
WireGuard actif	sudo wg show	interface wg0
OpenVPN actif	sudo ss -ulnp | grep 1194	openvpn pid
NAT persisté	sudo netfilter-persistent save	Rules saved

11. Synthèse du Durcissement — Tableau récapitulatif

Le tableau suivant présente l'ensemble des mesures de durcissement appliquées sur la VM tourou, classées par domaine de sécurité, avec leur référence aux standards CIS Benchmark et ANSSI, leur statut de validation et la commande de vérification associée.

Domàne	Mesure appliquée	Standard	Référence	Statut
SSH	Port SSH changé vers 2235	CIS L2	CIS 5.2.2	☑ Validé
SSH	PermitRootLogin no	CIS L2 / ANSSI	CIS 5.2.10	☑ Validé
SSH	PasswordAuthentication no	CIS L2 / NIST	CIS 5.2.11	☑ Validé
SSH	Clé ED25519 uniquement	NIST SP 800-57	FIPS 186-5	☑ Validé
SSH	MaxAuthTries 3	CIS L1	CIS 5.2.7	☑ Validé
SSH	X11Forwarding no	CIS L1	CIS 5.2.6	☑ Validé
SSH	AllowTcpForwarding no	CIS L2	CIS 5.2.20	☑ Validé
SSH	ClientAliveInterval 300	CIS L1	CIS 5.2.16	☑ Validé
Comptes	Utilisateur non-root alibora	CIS L1 / ANSSI	CIS 5.4.2	☑ Validé
Comptes	sudo su avec mot de passe	CIS L1	CIS 5.3.1	☑ Validé
Firewall	Politique DROP par défaut	CIS L1 / ANSSI	CIS 3.5.1	☑ Validé
Firewall	Ports minimaux ouverts	ANSSI R27	CIS 3.5.2	☑ Validé
Firewall	NAT OpenVPN + WireGuard	RFC 3022	Custom	☑ Validé
Firewall	NAT LAN 20.20.20.0/24	RFC 3022	Custom	☑ Validé
Firewall	Port 22 fermé	CIS L1	CIS 3.5.2	☑ Validé
Noyau	ASLR activé (randomize_va_space=2)	CIS L1	CIS 1.6.3	☑ Validé
Noyau	SYN cookies activés	CIS L1	CIS 3.2.8	☑ Validé
Noyau	ICMP redirects désactivés	CIS L1	CIS 3.2.2	☑ Validé
Noyau	Source routing désactivé	CIS L1	CIS 3.2.1	☑ Validé
Noyau	dmesg restreint (kptr_restrict=2)	CIS L2	CIS 1.6.4	☑ Validé
Noyau	Log martians activé	CIS L1	CIS 3.2.4	☑ Validé
Surveillance	Fail2ban SSH (bantime 3600s)	CIS L1	CIS 5.3.4	☑ Validé

Surveillance	Fail2ban Pritunl (bantime 3600s)	Custom	Custom	☑ Validé
Surveillance	Auditd + règles hardening	CIS L2	CIS 4.1.x	☑ Validé
Surveillance	AIDE base initialisée (47MB)	CIS L2	CIS 1.3.1	☑ Validé
Surveillance	AIDE cron quotidien 03h00	CIS L2	CIS 1.3.2	☑ Validé
Surveillance	Logwatch rapport quotidien	ANSSI	Custom	☑ Validé
VPN	OpenVPN 1194/UDP actif	RFC 2246	Custom	☑ Validé
VPN	WireGuard 14623/UDP actif	RFC 8520	Custom	☑ Validé
VPN	TOTP 2FA (Google Auth)	RFC 6238	NIST SP 800-63B	☑ Validé
VPN	Session timeout 7 jours (604800s)	Custom	Custom	☑ Validé
VPN	TLS 1.2+ (OpenSSL 3.0)	NIST SP 800-52r2	CIS 3.5.3	☑ Validé
MongoDB	Écoute localhost uniquement	CIS L1	CIS 2.1.1	☑ Validé
MongoDB	Port 27017 non exposé	CIS L1	CIS 3.5.2	☑ Validé
Système	Anciens noyaux supprimés	Hygiène	Custom	☑ Validé
Système	Type CPU Proxmox = host	Proxmox best practice	Custom	☑ Validé
Système	Module TUN chargé au boot	Custom	Custom	☑ Validé
Système	Sym lien wg dans PATH	Custom	Custom	☑ Validé

12. Sources et Références

Les références suivantes constituent le corpus documentaire de ce guide technique. Elles couvrent les standards de sécurité appliqués, la documentation officielle des outils utilisés et les guides de bonnes pratiques reconnus.

12.1 Standards et Cadres de Sécurité

[1] Center for Internet Security (CIS). (2024). CIS Ubuntu Linux 24.04 LTS Benchmark, Version 1.0. CIS Benchmarks. https://www.cisecurity.org/benchmark/ubuntu_linux

[2] National Institute of Standards and Technology (NIST). (2020). SP 800-53 Rev. 5: Security and Privacy Controls for Information Systems and Organizations. NIST. https://doi.org/10.6028/NIST.SP.800-53r5

[3] National Institute of Standards and Technology (NIST). (2017). SP 800-63B: Digital Identity Guidelines — Authentication and Lifecycle Management. NIST. https://doi.org/10.6028/NIST.SP.800-63b

[4] Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI). (2022). Guide de Recommandations de Sécurité Relatives à un Système GNU/Linux. ANSSI. https://www.ssi.gouv.fr/guide/recommandations-de-securite-relatives-a-un-systeme-gnulinux/

[5] Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI). (2021). Recommandations de sécurité relatives aux réseaux privés virtuels. ANSSI. https://www.ssi.gouv.fr/guide/recommandations-de-securite-relatives-aux-reseaux-prives-virtuels/

[6] National Institute of Standards and Technology (NIST). (2019). FIPS 186-5: Digital Signature Standard (DSS). NIST. https://doi.org/10.6028/NIST.FIPS.186-5

12.2 RFC et Standards Techniques

[7] Donenfeld, J. A. (2017). WireGuard: Next Generation Kernel Network Tunnel. Proceedings of the Network and Distributed System Security Symposium (NDSS 2017). https://www.wireguard.com/papers/wireguard.pdf

[8] Ylonen, T., & Lonvick, C. (2006). RFC 4251: The Secure Shell (SSH) Protocol Architecture. IETF. https://www.rfc-editor.org/rfc/rfc4251

[9] Kivinen, T., & Kojo, M. (2003). RFC 3526: More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE). IETF. https://www.rfc-editor.org/rfc/rfc3526

[10] M'Raihi, D., Bellare, M., Hoornaert, F., Naccache, D., & Ranen, O. (2005). RFC 4226: HOTP: An HMAC-Based One-Time Password Algorithm. IETF. https://www.rfc-editor.org/rfc/rfc4226

[11] M'Raihi, D., Machani, S., Pei, M., & Rydell, J. (2011). RFC 6238: TOTP: Time-Based One-Time Password Algorithm. IETF. https://www.rfc-editor.org/rfc/rfc6238

[12] Srisuresh, P., & Egevang, K. (2001). RFC 3022: Traditional IP Network Address Translator (Traditional NAT). IETF. https://www.rfc-editor.org/rfc/rfc3022

12.3 Documentation Officielle des Outils

[13] Pritunl. (2024). Pritunl VPN Server Documentation. Pritunl Inc. https://docs.pritunl.com

[14] MongoDB Inc. (2024). MongoDB 8.0 Manual — Security Hardening. MongoDB Documentation. https://www.mongodb.com/docs/v8.0/security/

[15] OpenVPN Inc. (2024). OpenVPN 2.6 Documentation. OpenVPN. https://openvpn.net/community-resources/

[16] WireGuard. (2024). WireGuard Official Documentation. Jason A. Donenfeld. https://www.wireguard.com/

[17] Proxmox Server Solutions GmbH. (2024). Proxmox VE Administration Guide. Proxmox. https://pve.proxmox.com/pve-docs/pve-admin-guide.html

[18] Ubuntu Security Team. (2024). Ubuntu 24.04 LTS Security Guide. Canonical Ltd. https://ubuntu.com/security

[19] Fail2ban Project. (2024). Fail2ban Documentation. https://www.fail2ban.org/wiki/index.php/Main_Page

[20] AIDE Project. (2024). AIDE — Advanced Intrusion Detection Environment Manual. https://aide.github.io/

12.4 Guides et Ressources Complémentaires

[21] OVHcloud. (2024). Documentation Bare Metal — Sécurité des serveurs dédiés. OVHcloud. https://help.ovhcloud.com/csm/fr-dedicated-servers-securing-server

[22] Vultr. (2025). How to Install and Setup Pritunl VPN Server on Ubuntu 24.04. Vultr Documentation. https://www.vultr.com/docs/install-pritunl-vpn-server-on-ubuntu-24-04/

[23] Canonical Ltd. (2024). Ubuntu Server Security Best Practices. Ubuntu Documentation. https://ubuntu.com/server/docs/security-introduction

[24] Linux Audit Project. (2024). Auditd — Linux Audit Daemon Manual. https://linux.die.net/man/8/auditd

[25] OWASP Foundation. (2024). OWASP Testing Guide v4.2 — Network Infrastructure Testing. OWASP. https://owasp.org/www-project-web-security-testing-guide/

12.5 Références complémentaires

[26] NIST. (2008). SP 800-123: Guide to General Server Security. National Institute of Standards and Technology. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-123.pdf

[27] ANSSI. (2020). Guide de configuration d'un système GNU/Linux (ANSSI-PB-079). https://www.ssi.gouv.fr/guide/configuration-recommandee-dun-systeme-gnulinux/

[28] OVHcloud. (2024). Configuration d'une IP Failover sur un serveur dédié. OVHcloud Documentation. https://docs.ovh.com/fr/dedicated/network-ipaliasing/

[29] Ubuntu Server Team. (2024). Ubuntu 24.04 LTS Server Guide. Canonical Ltd. https://ubuntu.com/server/docs

[30] Lynis. (2024). Lynis — Security Auditing Tool for Unix/Linux Systems. CISOfy. https://cisofy.com/lynis/

[31] AppArmor Project. (2024). AppArmor — Mandatory Access Control Framework. https://apparmor.net

[32] Linux Audit Project. (2024). Linux Audit Framework — auditd. https://linux-audit.com

13. Conclusion

Ce document a présenté de manière exhaustive et rigoureuse le déploiement d'une infrastructure VPN sécurisée basée sur Pritunl au sein d'un environnement Proxmox hébergé chez OVH. L'architecture retenue, articulée autour d'une double segmentation réseau (WAN/LAN) et d'une VM dédiée au service VPN, garantit une isolation stricte des machines internes tout en offrant un accès distant sécurisé via OpenVPN et WireGuard.

Le durcissement avancé appliqué à la VM Pritunl, conforme au CIS Ubuntu 24.04 LTS Benchmark (Level 2), ainsi que l'intégration d'outils de détection d'intrusion (AIDE), de surveillance (Logwatch, Auditd) et de protection active (Fail2ban, UFW, AppArmor), confèrent à cette infrastructure un niveau de sécurité opérationnel adapté aux environnements de production sensibles.

L'ensemble des configurations documentées est reproductible, auditable et constitue une base solide pour des évolutions futures : ajout de nouvelles VM, mise en cluster Proxmox haute disponibilité, intégration d'une solution SIEM ou déploiement d'un second serveur VPN en failover.

14. Annexes

Annexe A — Glossaire des termes techniques

Terme	Définition
AIDE	Advanced Intrusion Detection Environment — Outil de vérification d'intégrité par checksums cryptographiques
AppArmor	Système de contrôle d'accès mandataire intégré au noyau Linux, restreignant les capacités des processus
CIS Benchmark	Centre for Internet Security — Référentiel de durcissement système faisant autorité dans l'industrie
IP Failover	Adresse IP publique additionnelle OVH pouvant être routée dynamiquement vers une VM
Lynis	Outil d'audit de sécurité open-source pour systèmes Unix/Linux
MAC virtuelle	Adresse MAC assignée par OVH permettant le routage de l'IP failover vers une VM spécifique
Pritunl	Serveur VPN open-source avec interface web, basé sur OpenVPN et WireGuard
Proxmox VE	Hyperviseur de virtualisation open-source basé sur KVM et LXC
UFW	Uncomplicated Firewall — Interface simplifiée pour la gestion d'iptables sous Ubuntu
vmbr0	Bridge réseau Proxmox lié à l'interface physique eth0 (WAN)
vmbr1	Bridge réseau Proxmox interne pour le réseau LAN (20.20.20.0/24)
WireGuard	Protocole VPN moderne, intégré au noyau Linux depuis la version 5.6

Annexe B — Déclaration d'authenticité et mentions légales

Auteur principal	AFFANOU Eric Toussaint Yao — SysAdmin Réseau / DevOps
Date de rédaction	8 mai 2025
Version	v2.0 — 21 mai 2026
Langue de rédaction	Français
Classification	Confidentiel — Usage interne
Droits de reproduction	Tous droits réservés — Reproduction soumise à autorisation écrite de l'auteur

Ce document technique a été rédigé par AFFANOU Eric Toussaint Yao — SysAdmin Réseau / DevOps dans le cadre du déploiement d'une infrastructure réseau sécurisée. Toutes les configurations et recommandations présentées sont issues de sources officielles et de pratiques éprouvées en ingénierie des systèmes et de la sécurité informatique.