FR

Guide Pratique

Backup automatise OPNsense

Strategie 3-2-1, Git, S3 et DR < 15 min

Mars 2026

Sommaire

Pourquoi sauvegarder la config OPNsense

OPNsense stocke toute sa configuration dans un seul fichier XML : /conf/config.xml. Ce fichier contient vos regles de pare-feu, VLANs, tunnels VPN, certificats SSL, utilisateurs et alias — potentiellement des jours de travail condenses en quelques kilooctets.

Sans backup, un crash disque, une mise a jour ratee ou une corruption systeme peut effacer toute votre configuration reseau en quelques secondes. Avec 200+ regles de firewall et une dizaine de VLANs, on parle d'une journee entiere de travail perdu.

Ce qu'il faut sauvegarder :

* config.xml — L'integralite de la configuration : regles, interfaces, VPN, certificats, utilisateurs

* Packages installes — Liste des plugins (os-zerotier, os-wireguard, os-acme...) pour reinstallation rapide

* RRD data — Graphiques de performance reseau (optionnel)

* Certificats additionnels — Inclus dans config.xml mais a exporter separement si utilises ailleurs

i OPNsense expose une API REST complete pour automatiser entierement ce processus — sans plugin tiers, sans SSH.

Backup manuel via l'interface OPNsense

Dans l'interface OPNsense, naviguez vers Diagnostics -> Backup. Trois options principales :

* Download configuration as XML : telecharge le config.xml complet, chiffre ou non

* Include extra data : inclut les donnees RRD (graphiques de performance)

* Include package info : liste les packages installes pour documentation

```
# Backup one-shot via API OPNsense
curl -k -u 'votre-apikey:votre-apisecret' \
  -o opnsense-backup-$(date +%Y%m%d).xml \
  'https://192.168.1.1/api/core/backup/download/this'
```

Automatiser avec l'API REST OPNsense

Creez d'abord une cle API dans System -> Access -> Users -> [votre user] -> API keys. Le secret n'est affiche qu'une seule fois — gardez-le en lieu sur.

```
# Endpoints essentiels
GET  /api/core/backup/download/this      # telecharger config courante
GET  /api/core/backup/list               # lister backups internes
POST /api/core/backup/revertto/{id}      # restaurer un backup par ID
POST /api/core/backup/deleteback/{id}    # supprimer un backup par ID

# Authentification : HTTP Basic Auth (apikey:apisecret)
# Ne jamais utiliser le mot de passe admin — les cles API sont revocables
```

Script de backup quotidien vers Git prive

Stocker la config OPNsense dans un repo Git prive offre : historique complet, diff lisible entre versions, rollback instantane et audit trail.

```
#!/bin/bash
# /usr/local/bin/opnsense-backup.sh
set -euo pipefail

OPNSENSE_HOST='192.168.1.1'
API_KEY='votre-api-key'
API_SECRET='votre-api-secret'
BACKUP_DIR='/opt/opnsense-backups'
GIT_REPO='git@github.com:yourorg/opnsense-configs.git'
DATE=$(date +%Y-%m-%d_%H-%M)

if [ ! -d "$BACKUP_DIR/.git" ]; then
  git clone "$GIT_REPO" "$BACKUP_DIR"
fi
cd "$BACKUP_DIR" && git pull --rebase origin main

curl -sk -u "${API_KEY}:${API_SECRET}" \
  "https://${OPNSENSE_HOST}/api/core/backup/download/this" \
  -o "config-${DATE}.xml"

if ! grep -q '<opnsense>' "config-${DATE}.xml" 2>/dev/null; then
  echo "ERREUR: config.xml invalide"; rm -f "config-${DATE}.xml"; exit 1
fi

ln -sf "config-${DATE}.xml" config-latest.xml
git add -A
if ! git diff --staged --quiet; then
  git commit -m "backup: ${DATE} [auto]"
  git push origin main
fi

# Rotation : garder les 30 derniers XML
ls -t config-20*.xml 2>/dev/null | tail -n +31 | xargs -r rm -f
git add -A && git diff --staged --quiet || git commit -m "rotation: nettoyage"
git push origin main 2>/dev/null || true

# crontab -e (en tant que root)
0 2 * * * /usr/local/bin/opnsense-backup.sh >> /var/log/opnsense-backup.log 2>&1
```

Backup vers stockage distant

Option A — Amazon S3 (ou Wasabi, MinIO)

```
apt install awscli -y && aws configure

# Ajouter dans opnsense-backup.sh apres le commit Git :
aws s3 cp "config-${DATE}.xml" \
  s3://mon-bucket-backup/opnsense/config-${DATE}.xml \
  --storage-class STANDARD_IA
# Politique S3 Lifecycle : retention 90 jours automatique
```

Option B — SFTP vers NAS ou serveur distant

```
ssh-keygen -t ed25519 -f ~/.ssh/backup_opnsense -N ''
ssh-copy-id -i ~/.ssh/backup_opnsense.pub backup@nas.local

# Dans opnsense-backup.sh :
sftp -i ~/.ssh/backup_opnsense -b - backup@nas.local << EOF
  put config-${DATE}.xml /backups/opnsense/
  bye
EOF
```

Option C — Proxmox Backup Server (PBS)

Si OPNsense tourne en VM Proxmox, PBS peut sauvegarder l'integralite de la VM via snapshots Proxmox. Restauration complete en 5 minutes.

```
# Proxmox : configurer backup job pour la VM OPNsense
# Datacenter -> Backup -> Add job
# Storage: PBS, Mode: snapshot, Schedule: daily 03:00
```

Strategie 3-2-1 appliquee a OPNsense

La strategie 3-2-1 est le standard de reference pour les backups professionnels :

* 3 copies : config.xml local + repo Git prive + bucket S3

* 2 supports differents : stockage disque + cloud object storage

* 1 hors site : S3 dans une region differente (ou SFTP serveur distant)

```
# Frequences recommandees
Backup local Git    : quotidien a 2h00 (cron)
Push S3             : integre dans le script Git (automatique)
Snapshot PBS/Proxmox: hebdomadaire (dimanche 3h00)
Test de restauration: trimestriel

# Cout mensuel estime : < $3/mois sur S3 Standard-IA pour 90 jours
```

Tester la restauration

Un backup non teste est un backup inutile. Testez la restauration complete au moins une fois par trimestre sur un OPNsense de test.

Methode 1 — Via l'interface web

1. Diagnostics -> Backup -> section Restore

2. Browse -> choisir le fichier config.xml sauvegarde

3. Cocher "Reboot after restore" -> Restore configuration

4. Attendre le redemarrage (~2 min) -> verifier regles, VLANs, VPN

Methode 2 — Via l'API REST

```
curl -k -u 'apikey:apisecret' \
  -X POST \
  -F 'conffile=@/opt/opnsense-backups/config-latest.xml' \
  'https://192.168.1.1/api/core/backup/restore'

curl -k -u 'apikey:apisecret' \
  'https://192.168.1.1/api/core/firmware/status'
```

Checklist post-restauration : regles firewall actives, VLANs routes, tunnels WireGuard actifs, DHCP fonctionnel, DNS Unbound, certificats SSL valides.

Integration Ansible — Disaster Recovery < 15 min

La combinaison backup automatise + playbooks Ansible permet un disaster recovery complet en moins de 15 minutes, meme sur un hardware neuf.

```
# playbook-opnsense-dr.yml
---
- name: OPNsense Disaster Recovery
  hosts: opnsense_new
  gather_facts: false
  vars:
    backup_source: 'git@github.com:yourorg/opnsense-configs.git'
    backup_file: 'config-latest.xml'

  tasks:
    - name: Clone le repo de backup
      git:
        repo: '{{ BACKUP_SOURCE }}'
        dest: /tmp/opnsense-backup
        version: main

    - name: Restaurer la configuration OPNsense
      uri:
        url: 'https://{{ INVENTORY_HOSTNAME }}/api/core/backup/restore'
        method: POST
        url_username: '{{ API_KEY }}'
        url_password: '{{ API_SECRET }}'
        body_format: form-multipart
        body:
          conffile:
            content: '{{ LOOKUP_FILE }}'
            filename: 'config.xml'
        validate_certs: false

    - name: Attendre le redemarrage OPNsense
      wait_for:
        host: '{{ INVENTORY_HOSTNAME }}'
        port: 443
        delay: 30
        timeout: 300

    - name: Verifier les regles firewall
      uri:
        url: 'https://{{ INVENTORY_HOSTNAME }}/api/firewall/filter/searchrule'
        url_username: '{{ API_KEY }}'
        url_password: '{{ API_SECRET }}'
        validate_certs: false
      register: rules_check

    - name: Rapport final
      debug:
        msg: "OPNsense restaure avec {{ RULES_COUNT }} regles firewall"
```

Timeline DR : 0-5 min installation OPNsense neuf | 5-8 min ansible-playbook restore | 8-12 min verification automatisee | 12-15 min tests manuels. Objectif : < 15 minutes.

Conclusion : votre filet de securite reseau

Backup XML automatise via API REST + versioning Git + replication S3 + strategie 3-2-1 + disaster recovery Ansible : la configuration OPNsense devient une infrastructure resiliente.

Cout en temps d'implementation : 2-3 heures. Le gain : meme en cas de catastrophe, le reseau sera de retour en moins de 15 minutes.

Serie Stack OPNsense Enterprise — Navigation

* Billet 1 : Installer OPNsense dans Proxmox
* Billet 2 : VLANs & Zero Trust
* Billet 3 : WireGuard VPN & failover LTE
* Billet 4 : WiFi & APs par VLAN
* Billet 5 : CrowdSec + fail2ban IDS/IPS
* Billet 6 : NAC avec FreeRADIUS & 802.1X
* Billet 7 : Suricata IDS/IPS
* Billet 8 : AdGuard Home + DNS over HTTPS
* Billet 9 : Monitoring Grafana + InfluxDB
* Billet 10 : CARP & Haute disponibilite
* Billet 11 : SIEM Wazuh
* Billet 12 : Ansible as Code
* Billet 13 (ce billet) : Backup automatise OPNsense

i Hub complet : blog.botum.ca/opnsense-stack-securite-enterprise-proxmox/

Article FR : blog.botum.ca/opnsense-backup-config-automatise

Site web : www.botum.ca • contact@botum.ca