Déploiement VPS

Sécurisation et Configuration d'un VPS AlmaLinux

Guide de déploiement et sécurisation d'un VPS AlmaLinux chez OVH avec Nginx et site web statique.

Table des matières

Configuration DNS

Chez OVH

Enregistrements DNS de base :

Type	Nom	Valeur	TTL
A	@	<IP_VPS>	3600
A	www	<IP_VPS>	3600
AAAA	@	<IPv6_VPS>	3600 (optionnel)

Vérification DNS :

```
# Vérifier la propagation DNS
dig sanjy.fr
dig www.sanjy.fr

# Ou avec nslookup
nslookup sanjy.fr
```

Délai de propagation : 0-24h (généralement quelques minutes avec OVH)

Sécurisation SSH

1. Changer le port SSH par défaut

Éditer la configuration SSH :

```
sudo nano /etc/ssh/sshd_config
```

Modifications :

```
# Changer le port (éviter 22) (sécurité par l'obscurantisme)
Port XXXXX

# Désactiver le login root
PermitRootLogin no

# Désactiver l'authentification par mot de passe (recommandé après setup de clés)
PasswordAuthentication no

# Autoriser uniquement certains utilisateurs
AllowUsers votre_user

# Désactiver les connexions X11 forwarding (si non utilisé)
X11Forwarding no

# Limiter les tentatives
MaxAuthTries 3
```

```
MaxSessions 2
```

Redémarrer SSH :

```
sudo systemctl restart sshd
```

⚠ **IMPORTANT** : Gardez une session SSH ouverte et testez la connexion sur le nouveau port dans une nouvelle fenêtre avant de fermer la session actuelle !

Tester la nouvelle configuration :

```
ssh -p XXXXX user@sanjy.fr
```

2. Configuration des clés SSH (si pas déjà fait)

Sur votre machine locale :

```
# Générer une paire de clés (si nécessaire)
ssh-keygen -t rsa

# Copier la clé publique sur le serveur (remarque peut se faire en IHM depuis OVH)
ssh-copy-id -p XXXXX user@sanjy.fr
```

Sur le serveur, vérifier les permissions :

```
chmod 700 ~/.ssh
chmod 600 ~/.ssh/authorized_keys
```

3. SELinux et changement de port SSH

Problème courant : SELinux bloque SSH sur un port non-standard (bizarremment j'ai pas eu à le faire au final)

Solution :

```
# Ajouter le nouveau port à la politique SELinux
sudo semanage port -a -t ssh_port_t -p tcp XXXXX

# Vérifier
sudo semanage port -l | grep ssh
```

```
# Redémarrer SSH
sudo systemctl restart sshd
```

Configuration Firewall (nftables)

Configuration complète

Fichier : `/etc/nftables/ruleset.nft`

```
#!/usr/sbin/nft -f

# Flush des règles existantes
flush ruleset

table inet filter {
    # Chaîne INPUT - Trafic entrant
    chain input {
        type filter hook input priority filter
        policy drop

        # Loopback
        iif "lo" accept comment "Autoriser loopback"

        # Connexions établies
        ct state established,related accept comment "Autoriser connexions établies"

        # SSH sur port custom
        tcp dport XXXXX accept comment "Autoriser SSH sur port XXXXX"

        # ICMP (ping)
        ip protocol icmp accept comment "Autoriser ICMP/ping IPv4"
        ip6 nexthdr ipv6-icmp accept comment "Autoriser ICMPv6/ping IPv6"

        # DHCP client
        udp sport 67 udp dport 68 accept comment "Autoriser DHCP client"

        # HTTP/HTTPS
```

```
        tcp dport 80 ct state established,new accept comment "Accès site web HTTP"
        tcp dport 443 ct state established,new accept comment "Accès site web HTTPS"

        # NoMachine (optionnel)
        tcp dport 4000 accept comment "Connexion avec NoMachine"
    }

    # Chaîne FORWARD - Pas utilisée pour site statique
    chain forward {
        type filter hook forward priority filter
        policy drop
    }

    # Chaîne OUTPUT - Trafic sortant
    chain output {
        type filter hook output priority filter
        policy accept
    }
}
```

Gestion du firewall

```
# Appliquer les règles
sudo nft -f /etc/nftables/ruleset.nft

# Vérifier les règles actives
sudo nft list ruleset

# Activer au démarrage
sudo systemctl enable nftables
sudo systemctl start nftables

# Recharger après modification
sudo systemctl reload nftables
```

Commandes utiles nftables

```
# Lister les règles de façon lisible
sudo nft list ruleset
```

```
# Vider toutes les règles
sudo nft flush ruleset

# Ajouter une règle temporaire (test)
sudo nft add rule inet filter input tcp dport 8080 accept

# Sauvegarder la config actuelle
sudo nft list ruleset > /etc/nftables/backup-ruleset.nft
```

Gestion SELinux

Vérifier le statut

```
# Statut SELinux
sestatus

# Mode actuel
getenforce
```

Commandes SELinux courantes

```
# Passer en mode permissif (temporaire)
sudo setenforce 0

# Repasser en mode enforcing
sudo setenforce 1

# Voir les contextes de fichiers
ls -lZ /path/to/file

# Restaurer les contextes par défaut
sudo restorecon -Rv /path/to/directory

# Rendre permanent un contexte
sudo semanage fcontext -a -t httpd_sys_content_t "/var/www(/.*)?"
```

```
sudo restorecon -Rv /var/www
```

Problèmes SELinux courants

1. Nginx ne peut pas accéder aux certificats SSL

```
sudo restorecon -Rv /etc/letsencrypt
sudo semanage fcontext -a -t httpd_sys_content_t "/etc/letsencrypt(/.*)?"
```

2. Port non-standard bloqué par défaut sudo restorecon -Rv /path/to/directory

Rendre permanent un contexte

sudo semanage fcontext -a -t httpd_sys_content_t

```
# Pour SSH
sudo semanage port -a -t ssh_port_t -p tcp XXXXX

# Pour HTTP/HTTPS custom
sudo semanage port -a -t http_port_t -p tcp 8080
```

Debug SELinux

```
# Voir les dernières violations
sudo ausearch -m avc -ts recent
```

Accès distant avec NoMachine

Pourquoi NoMachine au lieu de VNC ?

Problème avec VNC sur AlmaLinux :

- TigerVNC et autres solutions VNC ont des problèmes de compatibilité avec AlmaLinux (surtout Wayland)
- Configuration complexe avec SELinux
- Performances médiocres

NoMachine :

- Installation simple
- Meilleure performance
- Compatible AlmaLinux out-of-the-box
- Chiffrement intégré
- N'est pas basé sur RDP (RDP de la m il semblerait :p)

Installation NoMachine

1. Télécharger et installer :

```
# Télécharger la version serveur
wget https://download.nomachine.com/download/9.2/Linux/nomachine_9.2.18_3_x86_64.tar.gz

# Extraire
tar -xzf nomachine_9.2.18_3_x86_64.tar.gz

# Installer
cd NX
sudo ./nxserver --install

# Vérifier le statut
sudo /etc/NX/nxserver --status
```

2. Configurer le firewall (sur VPS):

```
# Ajouter le port NoMachine (4000) dans nftables
sudo nft add rule inet filter input tcp dport 4000 accept

# Sauvegarder
sudo nft list ruleset > /etc/nftables/ruleset.nft
```

3. Configuration SELinux :

```
# Autoriser NoMachine
sudo semanage port -a -t http_port_t -p tcp 4000

# Ou si problème persistant
sudo setsebool -P nis_enabled on
```

4. Démarrer le service :

```
sudo systemctl enable nxserver
sudo systemctl start nxserver
```

Connexion depuis le client

Sur machine locale :

1. Installer le client NoMachine depuis nomachine.com
2. Créer une nouvelle connexion :
   - **Protocole** : NX
   - **Hôte** : sanjy.fr
   - **Port** : 4000
3. Se connecter avec vos identifiants Linux

Configuration avancée NoMachine

Fichier : `/usr/NX/etc/server.cfg`

```
# Éditer la config
sudo nano /usr/NX/etc/server.cfg

# Paramètres utiles :
# Changer le port d'écoute
CommandPort 4000

# Limiter les connexions simultanées
SessionLimit 2

# Activer le chiffrement
EnableEncryption 1
```

Redémarrer après modification :

```
sudo /etc/NX/nxserver --restart
```

Déploiement du site web

1. Installation de Nginx

```
# Installer Nginx
sudo dnf install nginx -y

# Activer et démarrer
sudo systemctl enable nginx
sudo systemctl start nginx

# Vérifier le statut
sudo systemctl status nginx
```

2. Structure des fichiers

```
/var/www/sanjy.fr/     # Racine du site
├── index.html
├── assets/
│   ├── css/
│   ├── js/
│   └── images/
/etc/nginx/
├── nginx.conf             # Config principale
└── conf.d/
    └── sanjy.fr.conf  # Config du site
```

3. Créer la structure

```
# Créer le dossier du site
sudo mkdir -p /var/www/sanjy.fr

# Définir les permissions
sudo chown -R nginx:nginx /var/www/sanjy.fr
sudo chmod -R 755 /var/www/sanjy.fr
```

4. Configuration Nginx

Fichier : `/etc/nginx/conf.d/sanjy.fr.conf`

```
server {
    listen 80;
    listen [::]:80;
    server_name sanjy.fr www.sanjy.fr;

    # Redirection HTTPS (après installation du certificat)
    # return 301 https://$server_name$request_uri;

    # Temporaire avant SSL
    root /var/www/sanjy.fr;
    index index.html index.htm;

    location / {
        try_files $uri $uri/ =404;
    }
}

# Configuration HTTPS (à décommenter après installation du certificat)
# server {
#     listen 443 ssl http2;
#     listen [::]:443 ssl http2;
#     server_name sanjy.fr www.sanjy.fr;
#
#     # Certificats SSL
#     ssl_certificate /etc/letsencrypt/live/sanjy.fr/fullchain.pem;
#     ssl_certificate_key /etc/letsencrypt/live/sanjy.fr/privkey.pem;
#
#     # Configuration SSL moderne
#     ssl_protocols TLSv1.2 TLSv1.3;
#     ssl_ciphers 'ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384';
#     ssl_prefer_server_ciphers off;
#
#     # Headers de sécurité
#     add_header Strict-Transport-Security "max-age=63072000; includeSubDomains; preload"
always;
#     add_header X-Frame-Options "SAMEORIGIN" always;
#     add_header X-Content-Type-Options "nosniff" always;
#     add_header X-XSS-Protection "1; mode=block" always;
#
```

```
#     # Racine du site
#     root /var/www/sanjy.fr;
#     index index.html index.htm;
#
#     location / {
#         try_files $uri $uri/ =404;
#     }
#
#     # Optimisations pour fichiers statiques
#     location ~* \.(jpg|jpeg|png|gif|ico|css|js|svg|woff|woff2|ttf)$ {
#         expires 1y;
#         add_header Cache-Control "public, immutable";
#     }
# }
```

Tester et recharger :

```
# Tester la config
sudo nginx -t

# Recharger
sudo systemctl reload nginx
```

5. Déployer le site

Exemple de fichier index.html :

```
sudo nano /var/www/sanjy.fr/index.html
```

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Mon Site</title>
</head>
<body>
    <h1>Bienvenue sur sanjy.fr</h1>
    <p>Site en construction...</p>
</body>
```

```
</html>
```

Ou upload depuis votre machine locale :

```
# Depuis votre machine
scp -P XXXXX -r /path/to/site/* user@sanjy.fr:/tmp/

# Sur le serveur
sudo mv /tmp/* /var/www/sanjy.fr/
sudo chown -R nginx:nginx /var/www/sanjy.fr
sudo restorecon -Rv /var/www/sanjy.fr
```

6. Certificats SSL avec Let's Encrypt

```
# Installer Certbot
sudo dnf install certbot python3-certbot-nginx -y

# Obtenir un certificat
sudo certbot certonly --manual --preferred-challenges dns -d sanjy.fr

Il faut ensuite prouver la possession du DNS en rajoutant une entrée DNS de type TXT !
Une fois validé il génère notre certif pem dans
/etc/letsencrypt/live/sanjypro.asz/fullchain.pem

# Etendre le certif sur de nouveau sous domaine on peut utiliser ex rajout de wiki.sanjy.fr
sudo certonly certbot --nginx -d sanjy.fr -d www.sanjy.fr -d wiki.sanjy.fr --expand



# Vérifier le renouvellement automatique
sudo certbot renew --dry-run

# Le renouvellement auto est configuré via systemd timer
sudo systemctl status certbot-renew.timer
```

Fix SELinux pour les certificats :

```
# Permettre à Nginx de lire les certificats
sudo restorecon -Rv /etc/letsencrypt
sudo semanage fcontext -a -t httpd_sys_content_t "/etc/letsencrypt(/.*)?"
sudo restorecon -Rv /etc/letsencrypt
```

Après installation du certificat :

1. Décommentez la section HTTPS dans `/etc/nginx/conf.d/sanjy.fr.conf`
2. Activez la redirection HTTP → HTTPS
3. Testez et rechargez : `sudo nginx -t && sudo systemctl reload nginx`

Checklist de sécurité

- ☑ SSH sur port non-standard (XXXXX)
- ☑ Authentification par clé SSH uniquement
- ☑ Root login désactivé
- ☑ Firewall nftables configuré (policy drop par défaut)
- ☑ SELinux en mode enforcing
- ☑ Certificats SSL Let's Encrypt
- ☑ Headers de sécurité HTTP configurés
- ☑ Mises à jour automatiques de sécurité
- ☑ Monitoring des logs
- ☑ Backups réguliers

Ressources

- AlmaLinux Documentation
- nftables Wiki
- SELinux Project
- NoMachine Documentation
- Let's Encrypt
- Nginx Documentation

Revision #8
Created 2025-11-28 01:39:05 UTC by Admin
Updated 2026-05-10 10:50:16 UTC by Admin