Déploiement du Vault

Déploiement HashiCorp Vault avec Podman Rootless

Architecture

- **Système** : AlmaLinux (RHEL-based)
- **Container Engine** : Podman (rootless)
- **Stockage** : Raft (backend intégré HA)
- **Sécurité** : SELinux activé
- **Reverse Proxy** : Nginx avec Let's Encrypt

Déploiement - Guide Complet

1. Préparation des volumes

```
# Créer les volumes nommés (recommandé pour Podman rootless)
podman volume create vault_config
podman volume create vault_data
```

2. Créer le fichier de configuration Vault

```
# Créer vault.hcl
cat > $(podman volume mount vault_config)/vault.hcl << 'EOF'
disable_cache = true
disable_mlock = true
ui = true
max_lease_ttl = "2h"
default_lease_ttl = "20m"
```

```
listener "tcp" {
  address     = "0.0.0.0:8200"
  tls_disable = true
}

storage "raft" {
  path    = "/vault/file"
  node_id = "vault-1"
}

api_addr = "http://127.0.0.1:8200"
cluster_addr = "https://127.0.0.1:8201"
EOF
```

3. Démarrer le conteneur Vault

```
podman run -d \
  --name vault \
  -p 8200:8200 \
  -p 8201:8201 \
  --cap-add=IPC_LOCK \
  -v vault_config:/vault/config \
  -v vault_data:/vault/file \
  docker.io/hashicorp/vault server
```

Gnagnagna pourquoi tu n'utilises pas un fichier de docker-compose ? Y'a un seul conteneur et 180 characters ... pourquoi utiliser un fichier docker-compose pour ça vraiment...

4. Vérifier le démarrage

```
# Vérifier les logs
podman logs vault

# Vérifier le statut
podman ps
```

5. Initialiser Vault (webui ou dans le container)

```
# Initialiser avec 1 clé (configuration single-admin)
vault operator init -key-shares=1 -key-threshold=1

# ⚠ IMPORTANT : Sauvegarder immédiatement dans un password manager :
# - Unseal Key (ex: abcd1234efgh5678...)
# - Initial Root Token (ex: s.xyz789abc123...)
```

6. Configuration de l'authentification AppRole

```
# WARNING! VAULT_ADDR and -address unset. Defaulting to https://127.0.0.1:8200.
export VAULT_ADDR=http://127.0.0.1:8200.

# Login avec le root token
vault login <ROOT_TOKEN>

# Activer AppRole
vault auth enable approle

# Créer la policy admin
vault policy write admin - << 'EOF'
path "*" {
  capabilities = ["create", "read", "update", "delete", "list", "sudo"]
}
EOF

# Créer le rôle AppRole avec la policy admin
vault write auth/approle/role/admin \
  token_policies="admin" \
  token_ttl=1h \
  token_max_ttl=4h
```

7. Récupérer les credentials AppRole

```
# Récupérer le RoleID (permanent)
ROLE_ID=$(vault read -field=role_id auth/approle/role/admin/role-id)
echo "RoleID: $ROLE_ID"

# Générer un SecretID (temporaire, à regénérer)
```

```
SECRET_ID=$(vault write -f -field=secret_id auth/approle/role/admin/secret-id)
echo "SecretID: $SECRET_ID"

# Sauvegarder le RoleID
echo "$ROLE_ID" > ~/.vault-role-id
```

8. Login avec AppRole

```
# Login et récupération du token
vault write auth/approle/login \
  role_id="$ROLE_ID" \
  secret_id="$SECRET_ID"

# Ou directement exporter le token
export VAULT_TOKEN=$(vault write -field=token auth/approle/login \
  role_id="$ROLE_ID" \
  secret_id="$SECRET_ID")
```

9. Configuration Nginx (Reverse Proxy avec SSL)

```
# Créer le fichier de configuration Nginx
sudo nano /etc/nginx/conf.d/vault.conf
```

```
server {
    listen 80;
    server_name vault.sanjy.fr;

    # Redirection HTTPS
    return 301 https://$server_name$request_uri;
}

server {
    listen 443 ssl http2;
    server_name vault.sanjy.fr;

    # Certificats SSL Let's Encrypt
    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 HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers on;

    # Headers de sécurité
    add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;
    add_header X-Frame-Options "DENY" always;
    add_header X-Content-Type-Options "nosniff" always;

    # Proxy vers Vault
    location / {
        proxy_pass 127.0.0.1:8200;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        # WebSocket support (pour l'UI)
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }
}
```

```
# Tester la configuration Nginx
sudo nginx -t

# Recharger Nginx
sudo systemctl reload nginx
```

10. Expansion du certificat Let's Encrypt

```
# Ajouter le sous-domaine vault au certificat existant
sudo certbot certonly --nginx \
  -d sanjy.fr \
  -d www.sanjy.fr \
```

```
  -d vault.sanjy.fr \
  --expand
```

11. Gestion SELinux pour les certificats (si besoin)

```
# Restaurer le contexte SELinux correct
sudo restorecon -Rv /etc/letsencrypt

# Rendre la configuration permanente
sudo semanage fcontext -a -t httpd_sys_content_t "/etc/letsencrypt(/.*)?"
sudo restorecon -Rv /etc/letsencrypt
```

13. Service Systemd (optionnel - pour auto-start)

```
# Générer le fichier systemd
podman generate systemd --new --name vault > ~/.config/systemd/user/vault.service

# Activer au démarrage
systemctl --user enable vault.service
systemctl --user start vault.service

# Activer le linger (conteneur démarre même si user pas connecté)
sudo loginctl enable-linger $USER
```

Ressources

- Documentation Vault
- Podman Documentation
- SELinux User Guide
- Site de l'auteur

Revision #7
Created 2025-11-28 01:05:18 UTC by Admin
Updated 2026-01-11 18:47:37 UTC by Admin