Déploiement Bookstack

Comment déployer bookstack en mode conteneur

Bookstack déploiement

Contexte

Déploiement d'un wiki BookStack sur VPS AlmaLinux 10 avec Podman (au lieu de Docker), Nginx comme reverse proxy, et nftables comme firewall.

Architecture

```
Internet → Nginx (443) → Podman (localhost:30080) → BookStack → MariaDB
                ↓
           Let's Encrypt SSL
```

- **BookStack** : Wiki de documentation
- **MariaDB** : Base de données
- **Nginx** : Reverse proxy avec SSL
- **Podman** : Alternative à Docker (rootless, sans daemon)

Prérequis

```
# Installer Podman et podman-compose
sudo dnf install podman podman-compose -y

# Vérifier l'installation
podman --version
podman-compose --version
```

Configuration finale qui fonctionne

Structure des dossiers

```
~/bookstack/
├── docker-compose.yml
```

docker-compose.yml

```
version: '3.8'

services:
  bookstack-db:
    image: mariadb:12.1.2
    container_name: bookstack-db
    environment:
      MYSQL_ROOT_PASSWORD: RootPassword123
      MYSQL_DATABASE: bookstack
      MYSQL_USER: bookstack
      MYSQL_PASSWORD: BookStackPass123
    volumes:
      - bookstack-db-data:/var/lib/mysql
    restart: unless-stopped

  bookstack:
    image: lscr.io/linuxserver/bookstack:latest
    container_name: bookstack
    environment:
      PUID: 1000
      PGID: 1000
      TZ: Europe/Paris
      APP_URL: https://wiki.sanjy.fr
      DB_HOST: bookstack-db
```

```
      DB_PORT: 3306
      DB_DATABASE: bookstack
      DB_USERNAME: bookstack
      DB_PASSWORD: BookStackPass123
      APP_KEY: base64:APP_KEY
    volumes:
      - bookstack-config:/config
    ports:
      - "30080:80"
    depends_on:
      - bookstack-db
    restart: unless-stopped

volumes:
  bookstack-db-data:
  bookstack-config:
```

Points importants :

- Pas de guillemets autour des variables (évite les problèmes d'échappement)
- Mots de passe simples sans caractères spéciaux
- `depends_on` assure que MariaDB démarre avant BookStack
- Port 30080 choisi pour éviter les conflits

Configuration Nginx

Fichier : `/etc/nginx/conf.d/wiki.conf`

```
server {
    listen 80;
    listen [::]:80;
    server_name wiki.sanjy.fr;
    return 301 https://$server_name$request_uri;
}

server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name wiki.sanjy.fr;
```

```
    ssl_certificate /etc/letsencrypt/live/sanjy.fr/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/sanjy.fr/privkey.pem;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers on;

    add_header Strict-Transport-Security "max-age=31536000" always;

    location / {
        proxy_pass http://localhost:30080;
        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;

        # Important pour les uploads de fichiers
        client_max_body_size 100M;
    }
}
```

Appliquer la configuration :

```
sudo nginx -t
sudo systemctl reload nginx
```

SELinux : Autoriser Nginx à se connecter au réseau

```
sudo setsebool -P httpd_can_network_connect 1
```

Déploiement complet - Procédure pas à pas

1. Préparation

```
# Créer le dossier
mkdir -p ~/bookstack
cd ~/bookstack

# Créer le docker-compose.yml (voir ci-dessus)
nano docker-compose.yml
```

2. Générer l'APP_KEY

```
APP_KEY=$(podman run --rm --entrypoint /bin/bash lscr.io/linuxserver/bookstack:latest appkey |
tr -d '\r')
echo "APP_KEY généré: $APP_KEY"

# Éditer docker-compose.yml et remplacer APP_KEY
nano docker-compose.yml
```

3. Configuration DNS (OVH)

Ajouter un enregistrement A :

```
Type: A
Sous-domaine: wiki
Cible: [IP du VPS]
TTL: 3600
```

4. Règles firewall nftables

```
# Trouver le subnet après le premier lancement
podman-compose up -d
SUBNET=$(podman network inspect bookstack_default | grep -i subnet | awk '{print $2}' | tr -d
'",')

# Ajouter les règles
sudo nft add rule inet filter input ip saddr $SUBNET accept
sudo nft add rule inet filter forward ip saddr $SUBNET accept
sudo nft add rule inet filter forward ip daddr $SUBNET accept
```

```
# Sauvegarder
sudo nft list ruleset | sudo tee /etc/nftables/ruleset.nft
```

5. Lancer BookStack

```
cd ~/bookstack
podman-compose up -d

# Vérifier les logs
podman-compose logs -f
```

Attendre 30 secondes que MariaDB s'initialise complètement.

6. Vérifier le bon fonctionnement

```
# Vérifier les containers
podman-compose ps

# Tester la connexion réseau
podman exec -it bookstack nc -zv bookstack-db 3306

# Tester l'accès web
curl http://localhost:30080
```

7. Configuration Nginx et SSL

```
# Créer la config Nginx
sudo nano /etc/nginx/conf.d/wiki.conf
# (Voir configuration ci-dessus)

# Étendre le certificat SSL
sudo certbot certonly --nginx \
  -d sanjy.fr \
  -d www.sanjy.fr \
```

```
  -d jenkins.sanjy.fr \
  -d wiki.sanjy.fr \
  --expand

# Recharger Nginx
sudo nginx -t
sudo systemctl reload nginx
```

8. Première connexion

1. Accéder à `https://wiki.sanjy.fr`
2. Se connecter avec :
   - Email: `admin@admin.com`
   - Password: `password`
3. **Changer immédiatement le mot de passe !**

Backup et restauration

Backup de la base de données

```
# Backup automatique
podman exec bookstack-db mysqldump -u bookstack -pBookStackPass123 bookstack > backup-$(date +%Y%m%d).sql

# Ou avec root
podman exec bookstack-db mysqldump -u root -pRootPassword123 --all-databases > backup-full-$(date +%Y%m%d).sql
```

Backup des fichiers BookStack

```
# Les fichiers sont dans le volume bookstack-config
podman volume inspect bookstack_bookstack-config

# Backup du volume
```

```
sudo tar -czf bookstack-config-backup-$(date +%Y%m%d).tar.gz \
  $(podman volume inspect bookstack_bookstack-config -f '{{.Mountpoint}}')
```

Restauration

```
# Restaurer la base de données
cat backup-20241122.sql | podman exec -i bookstack-db mysql -u root -pRootPassword123
bookstack

# Restaurer les fichiers
sudo tar -xzf bookstack-config-backup-20241122.tar.gz -C /
```

Améliorations possibles

1. Service systemd pour auto-start

```
cd ~/bookstack
podman generate systemd --new --files --name bookstack
mkdir -p ~/.config/systemd/user/
mv *.service ~/.config/systemd/user/
systemctl --user enable container-bookstack.service
loginctl enable-linger $USER
```

2. Backups automatiques avec cron

```
# Éditer crontab
crontab -e

# Ajouter backup quotidien à 2h du matin
0 2 * * * cd ~/bookstack && podman exec bookstack-db mysqldump -u root -pRootPassword123
bookstack > ~/backups/bookstack-$(date +\%Y\%m\%d).sql
```

3. Monitoring

Intégrer BookStack dans Prometheus/Grafana pour surveiller :

- Uptime des containers
- Utilisation CPU/RAM
- Taille de la base de données

Leçons apprises

1. **Kubernetes (K3s) me kasse le kou** : Je trouverai une solution ultérieure pour l'install de k3s sous almalinux 10
2. **nftables et containers nécessitent des règles explicites** : Contrairement à iptables, nftables ne crée pas automatiquement les règles pour les containers
3. **Les volumes Podman persistent même après** `podman-compose down` : Toujours supprimer explicitement les volumes pour repartir à zéro
4. **Les caractères spéciaux dans les variables d'environnement causent des problèmes** : Utiliser des mots de passe alphanumériques simples dans docker-compose.yml
5. **SELinux doit être configuré pour Nginx → Containers** : `httpd_can_network_connect` est essentiel
6. **Let's Encrypt ne renouvelle que les domaines dans le certificat** : Utiliser `--expand` pour ajouter de nouveaux sous-domaines

---

Ressources

- BookStack Documentation
- Podman Documentation
- LinuxServer.io BookStack Image
- Let's Encrypt Documentation

Changelog

- **2024-11-22** : Déploiement initial avec Podman Compose
  - Abandon de K3s pour Podman

- Configuration nftables pour communication inter-containers
- Résolution problèmes authentification MariaDB
- Configuration SSL avec Let's Encrypt

Déployé sur : VPS OVH - AlmaLinux 10
URL : https://wiki.sanjy.fr
Auteur : Sanjy Andriamiseza

Problèmes rencontrés + troubleshooting

Problèmes rencontrés et solutions

1. ? Tentative initiale : Kubernetes (K3s)

Problème : Communication réseau entre pods impossible

- **Erreur** : `SQLSTATE[HY000] [2002] No such file or directory`
- **Cause** : Module kernel `nf_conntrack` manquant
- **Erreur K3s** : `Extension conntrack is not supported, missing kernel module?`

Tentative de solution :

```
sudo modprobe nf_conntrack
sudo modprobe br_netfilter  # Module non disponible sur AlmaLinux 10
```

Il y a des soucis avec AlamaLinux 10 et K3S, je reviendrai dessus quand j'aurais le temps. **Décision** : Abandon de K3s au profit de Podman Compose (plus simple, moins de ressources)

---

2. ? Problème : Communication réseau entre containers Podman

Erreur : `nc: getaddrinfo for host "bookstack-mariadb" port 3306: Try again`

Cause : nftables bloque le trafic entre containers

Solution : Ajouter des règles nftables pour autoriser le subnet Podman

```
# Trouver le subnet Podman
podman network inspect bookstack_default | grep -i subnet
# Exemple de résultat : 5.1.0.0/24

# Ajouter les règles nftables
sudo nft add rule inet filter input ip saddr 5.1.0.0/24 accept
sudo nft add rule inet filter forward ip saddr 5.1.0.0/24 accept
sudo nft add rule inet filter forward ip daddr 5.1.0.0/24 accept

# Sauvegarder
sudo nft list ruleset | sudo tee /etc/nftables/ruleset.nft
```

Note importante : Docker et nftables ont des conflits connus. Podman gère mieux nftables mais nécessite quand même des règles explicites.

---

3. ? Problème : Erreurs d'authentification MariaDB

Erreur : `ERROR 1045 (28000): Access denied for user 'bookstack'@'X.X.X.X' (using password: YES)`

Causes multiples :

1. Volumes persistants gardant les anciens mots de passe
2. Incohérence entre `MYSQL_USER` et `DB_USERNAME`
3. Caractères spéciaux (`!`) mal échappés dans les mots de passe

Solution finale :

```
# 1. Arrêter et supprimer complètement les volumes
cd ~/bookstack
podman-compose down
podman volume rm -f bookstack_bookstack-db-data bookstack_bookstack-config

# 2. Vérifier que les volumes sont supprimés
podman volume ls | grep bookstack

# 3. Utiliser des mots de passe simples (sans caractères spéciaux)
# Voir docker-compose.yml ci-dessous
```

4. ? Problème : APP_KEY bookstack manquant

Erreur : `The application key is missing, halting init!`

Solution : Générer et ajouter l'APP_KEY

```
# Générer la clé
APP_KEY=$(podman run --rm --entrypoint /bin/bash lscr.io/linuxserver/bookstack:latest appkey |
tr -d '\r')

# Afficher pour vérification
echo "APP_KEY: $APP_KEY"

# L'ajouter dans docker-compose.yml sous environment de bookstack
```

5. ? Problème : Certificat SSL ne couvre pas wiki.sanjy.fr

Erreur : `SSL: no alternative certificate subject name matches target hostname`

Solution : Étendre le certificat existant

```
sudo certbot certonly --nginx \
  -d sanjy.fr \
  -d www.sanjy.fr \
  -d jenkins.sanjy.fr \
  -d wiki.sanjy.fr \
  --expand
```

Alternative : Certificat wildcard (recommandé pour plusieurs sous-domaines mais je préfère être exhaustif et être sur de fournir un certif pour chaque sous domaine)

```
sudo certbot certonly --manual --preferred-challenges dns \
  -d "*.sanjy.fr" -d "sanjy.fr"
```

Troubleshooting

Les containers ne démarrent pas

```
# Vérifier les logs
podman-compose logs

# Vérifier les volumes
podman volume ls

# Supprimer et recréer
podman-compose down
podman volume rm bookstack_bookstack-db-data bookstack_bookstack-config
podman-compose up -d
```

Erreur d'authentification MariaDB

Solution : Supprimer complètement les volumes et recréer

```
podman-compose down
podman volume rm -f bookstack_bookstack-db-data bookstack_bookstack-config
podman-compose up -d
```

Erreur 502 Bad Gateway sur Nginx

```
# Vérifier que BookStack tourne
podman-compose ps

# Vérifier SELinux
sudo setsebool -P httpd_can_network_connect 1

# Vérifier les logs Nginx
sudo tail -f /var/log/nginx/error.log
```

Problème de réseau entre containers

```
# Vérifier les règles nftables
sudo nft list ruleset | grep 10.89

# Tester la connexion
podman exec -it bookstack ping bookstack-db
podman exec -it bookstack nc -zv bookstack-db 3306
```
