Antony Kervazo-Canut

Podman for Teenagers

CONVIENT AUSSI AUX ADULTES

Introduction

Podman (Pod Manager) est un outil de gestion de conteneurs et d'images de conteneurs compatible avec OCI (Open Container Initiative). Il est conçu pour trouver, exécuter, créer et partager des conteneurs et des pods. Podman est souvent comparé à Docker, mais il fonctionne sans démon et permet de gérer des conteneurs en tant qu'utilisateur non privilégié.

La principale différence entre Podman et Docker réside dans leur architecture. Podman fonctionne sans démon, ce qui signifie que chaque conteneur est exécuté dans son propre processus, directement géré par l'utilisateur. Cela offre plusieurs avantages :

- Sécurité améliorée : Pas de démon central, réduisant le risque d'attaques ciblant le démon de conteneur.
- Compatibilité avec Docker : Podman peut exécuter presque toutes les commandes Docker grâce à une interface CLI similaire, facilitant la migration de Docker à Podman.
- Gestion en tant qu'utilisateur non privilégié : Les conteneurs peuvent être exécutés sans nécessiter de droits d'administrateur, améliorant ainsi la sécurité et la facilité d'utilisation.

Podman est particulièrement populaire dans les environnements où la sécurité est une priorité, ainsi que chez les utilisateurs préférant une solution plus légère et flexible que Docker.

Il est développé par des ingénieurs Red Hat® et des membres de la communauté Open Source.

Installation de Podman

L'installation de Podman varie selon le système d'exploitation. Voici comment l'installer sur les systèmes Linux les plus populaires, ainsi que sur macOS et Windows.

```
# Installation de Podman sur Fedora
sudo dnf install -y podman
# Installation de Podman compose
sudo dnf install -y podman-compose

# Mise à jour de la liste des paquets et installation de
Podman sur Ubuntu, Debian
sudo apt-get update && sudo apt-get install -y podman
# Installation de Podman compose
sudo apt-get update && sudo apt-get install -y podman-compose

# Installation de Podman sur macOS avec Homebrew
brew install podman
# Installation de Podman compose
brew install podman-compose

# Pour Windows, passez par une WSL2, guide dans l'URL
suivante :
https://github.com/containers/podman/blob/main/docs/tutorials
/podman-for-windows.md
```

DockerFile

L'utilisation d'un Dockerfile reste essentielle pour la création d'images personnalisées, que vous utilisiez Docker ou Podman. Podman est entièrement compatible avec les Dockerfile, ce qui signifie que vous pouvez construire vos images de conteneurs avec Podman exactement de la même manière que vous le feriez avec Docker.

Avantages de l'utilisation de Podman pour les Dockerfiles

Mode Rootless : Un avantage clé de Podman est sa capacité à construire et exécuter des conteneurs sans nécessiter de privilèges de superutilisateur, renforçant ainsi la sécurité.

Compatibilité : La compatibilité directe des Dockerfile avec Podman simplifie la transition pour les équipes et les projets déjà habitués à Docker, permettant une migration douce sans nécessiter de réécriture des Dockerfiles.

Pas de Daemon : Podman fonctionne sans daemon, ce qui signifie que chaque conteneur est exécuté comme un processus direct de l'utilisateur, réduisant la complexité et les risques associés à un processus daemon centralisé.

DockerFile
Exemple

```
# Utilise une image de base officielle Ubuntu comme point de
départ
FROM ubuntu:20.04

# Met à jour les paquets et installe nginx
RUN apt-get update && apt-get install -y nginx

# Copie le contenu du dossier courant dans le conteneur
COPY . /var/www/html

# Expose le port 80 pour permettre l'accès au serveur web
EXPOSE 80

# Exécute Nginx en avant-plan lorsque le conteneur démarre
CMD ["nginx", "-g", "daemon off;"]
```

Gestion des conteneurs

Impression de déjà-vus ? C'est normal, Podman facilite la transition depuis docker en intégrant les mêmes commandes.

```
# Construire une image à partir d'un Dockerfile.
# podman build -t [nom_de_l'image] [chemin_du_dossier]
# -t : nomme l'image construite.
podman build -t monimage .

# Affiche la liste des images avec des détails comme ID,
taille, tag, etc.
podman images

# Supprimer une image spécifique.
# podman rmi [ID_ou_nom_de_l'image]
podman rmi monimage

# Liste détaillée des couches de l'image et de leurs
changements.
# podman history [nom_de_l'image]
podman history nginx
```

Gestion des images

```
# Construire une image à partir d'un Dockerfile.
# podman build -t [nom_de_l'image] [chemin_du_dossier]
# -t : nomme l'image construite.
podman build -t monimage .

# Affiche la liste des images avec des détails comme ID,
taille, tag, etc.
podman images

# Supprimer une image spécifique.
# podman rmi [ID_ou_nom_de_l'image]
podman rmi monimage

# Liste détaillée des couches de l'image et de leurs
changements.
# podman history [nom_de_l'image]
podman history nginx
```

Gestion du réseau

La gestion du réseau est légèrement différente avec Podman. C'est un point de vigilance en cas de migration d'un outil à l'autre.

```
# Affiche la liste des réseaux avec des détails comme ID, nom, driver, etc.
podman network ls

# Créer un nouveau réseau de type bridge (par défaut)
podman network create monreseau

# Créer et connecter un conteneur à un réseau spécifique lors de son exécution
podman run --name monconteneur --network monreseau -d nginx

# Affiche des informations détaillées sur le réseau, comme les conteneurs connectés, les options, l'IP, etc.
# podman network inspect [nom_du_réseau]
podman network inspect monreseau
```

Gestion des données

```
# Créer un nouveau volume Podman pour le stockage persistant.
# podman volume create [nom_du_volume]
podman volume create monvolume

# Affiche la liste des volumes avec des détails comme le nom,
le driver, etc.
podman volume ls

# Attacher un volume à un conteneur pour le stockage
persistant.
# podman run -v [nom_du_volume]:[chemin_dans_le_conteneur]
[nom_de_l'image]
# -v : attache le volume nommé à un chemin spécifique dans le
conteneur.
podman run -v monvolume:/data nginx

# Supprimer un volume Podman spécifique.
# podman volume rm [nom_du_volume]
podman volume rm monvolume
```

Monitoring & Logs

```
# Affiche les logs de sortie du conteneur spécifié.
# podman logs [ID_ou_nom_du_conteneur]
podman logs monconteneur

# Fournit un aperçu en temps réel de l'utilisation des
ressources pour tous les conteneurs en cours d'exécution.
podman stats

# Ouvre un shell interactif dans le conteneur, si bash est
utilisé comme commande.
# docker exec [options] [ID_ou_nom_du_conteneur] [commande]
# -it : attache une session interactive TTY.
podman exec -it monconteneur bash

# Affiche un JSON détaillé avec des informations complètes
sur le conteneur, y compris l'état du réseau, les volumes
montés,
# les paramètres de configuration, etc.
# podman inspect [ID_ou_nom_du_conteneur]
podman inspect monconteneur
```

Debogage

```
# Affiche un JSON détaillé avec des informations complètes
sur le conteneur, y compris l'état du réseau, les volumes
montés,
# les paramètres de configuration, etc.
# podman inspect [ID_ou_nom_du_conteneur]
podman inspect monconteneur

# Sauvegarder un conteneur dans une nouvelle image. Utile
pour créer une image à partir d'un conteneur modifié.
# podman commit [ID_ou_nom_du_conteneur] [nom_de_l'image]
podman commit monconteneur monimage

# Copier des fichiers ou dossiers depuis et vers un
conteneur. Très utile pour le transfert de données.
# Pour copier du conteneur vers l'hôte:
# podman cp [ID_ou_nom_du_conteneur]:
[chemin_dans_le_conteneur] [chemin_local]
podman cp monconteneur:/fichier.txt /mon/dossier/local

# Pour copier de l'hôte vers le conteneur:
# podman cp [chemin_local] [ID_ou_nom_du_conteneur]:
[chemin_dans_le_conteneur]
podman cp /mon/dossier/local monconteneur:/fichier.txt

# Liste les fichiers et dossiers qui ont été ajoutés,
modifiés ou supprimés dans le conteneur.
# podman diff [ID_ou_nom_du_conteneur]
podman diff monconteneur
```

Nettoyage

```
# Nettoyer les ressources inutilisées telles que les
conteneurs arrêtés, les réseaux non utilisés, et les images
en cache.
podman system prune

# Supprimer les conteneurs arrêtés, les volumes non utilisés,
les réseaux non utilisés,
# et les images non utilisées (y compris les images en
cache).
podman system prune -a

# Supprimer toutes les images non utilisées, pas seulement
les images en cache.
podman image prune -a

# Supprimer tous les conteneurs arrêtés.
podman container prune

# Supprimer tous les volumes non utilisés.
podman volume prune
```

Gestion des registres

```
# Connexion à DockerHub pour permettre le push et le pull
d'images.
# podman login -u [nom_utilisateur] -p [mot_de_passe]
podman login -u monnomutilisateur -p monmotdepasse

# podman login -u [nom_utilisateur] -p [token_d'accès]
podman login -u monnomutilisateur -p
9f86d081884c7d659a2feaa0c55ad015a

# Télécharger une image depuis DockerHub sur le système
local.
# podman pull [nom_de_l'image]
podman pull ubuntu

# Pousser une image locale sur DockerHub.
# podman push [nom_utilisateur/nom_de_l'image]
podman push monutilisateur/monimage

# Recherche d'images sur DockerHub.
# podman search [terme_de_recherche]
podman search mysql
```

Podman Compose

Podman Compose est un outil qui permet de définir et de gérer des applications multi-conteneurs avec Podman, similaire à Docker Compose. Il utilise un fichier YAML pour configurer les services de l'application.

Vous pouvez ensuite créer un fichier docker-compose.yml pour votre projet. Voici un exemple simple :

```yaml
version: '3'
services:
  web:
    image: nginx
    ports:
      - "8080:80"
    volumes:
      - web-data:/usr/share/nginx/html
volumes:
  web-data:
```

Gestion de Podman Compose

```
# Démarre les services spécifiés dans le fichier docker-compose.yml en arrière-plan.
podman-compose up -d

# Arrêter et supprimer les ressources (conteneurs, réseaux, volumes) créées.
podman-compose down

# Affiche les projets Compose en cours d'exécution avec leur état.
podman-compose ls

# Afficher les logs de tous les services.
podman-compose logs

# Exécuter une commande unique dans un service.
# podman-compose run [nom_du_service] [commande]
podman-compose run webapp echo "Hello World"

# Mettre à jour et reconstruire un service spécifique.
# podman-compose up -d --no-deps [nom_du_service]
podman-compose up -d --no-deps webapp

# Arrêter un service spécifique.
# podman-compose stop [nom_du_service]
podman-compose stop webapp
```

Migration Docker vers Podman

La migration de Docker à Podman est un processus relativement simple grâce à la compatibilité des commandes et à l'architecture similaire des deux outils.

Points de vigilance

- Stockage et gestion des volumes : Bien que les commandes pour gérer les volumes soient similaires, le système de fichiers utilisé par Podman peut différer, surtout en mode rootless. Il est important de vérifier les chemins et les permissions des volumes et bind mounts après la migration.
- Réseaux : Podman gère les réseaux de manière légèrement différente. Si vous utilisez des configurations réseau complexes avec Docker, vous devrez peut-être ajuster ces configurations lors de la migration vers Podman.

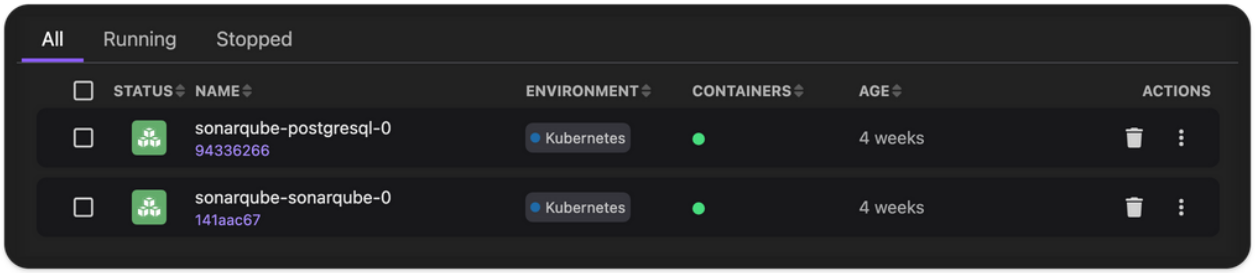
La migration de Docker à Podman est une opportunité de bénéficier d'une gestion de conteneurs plus sécurisée et flexible, en particulier pour les déploiements en mode rootless.

Podman Desktop

Podman Desktop est une interface utilisateur graphique (GUI) et une application de bureau pour Podman, visant à fournir une expérience utilisateur simplifiée pour la gestion des conteneurs et des images de conteneurs, similaire à ce que Docker Desktop offre pour Docker. Alors que Podman est traditionnellement utilisé via la ligne de commande (CLI), Podman Desktop vise à rendre la gestion des conteneurs plus accessible, surtout pour les utilisateurs qui préfèrent une interface graphique.

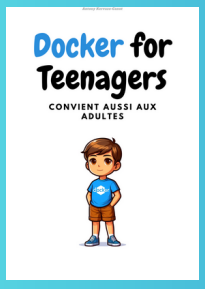
Contrairement à Docker Desktop qui nécessite une licence en entreprise, Podman Desktop est gratuit.

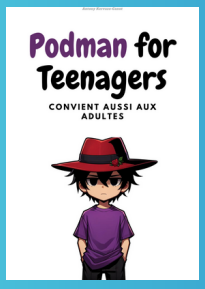
Podman Desktop peut récupérer des ressources précédemment installé avec un autre support et les afficher avec leur source.

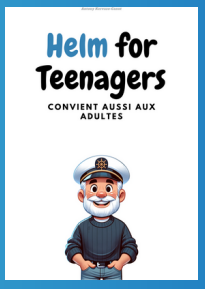


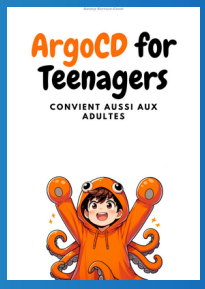
Dans la même collection

Orchestration et Gestion de Conteneurs

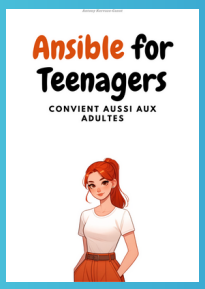








Infrastructure as Code



Sécurité & Gestion des secrets

Développement & CI/CD

ANTONYCANUT

ANTONY KERVAZO-CANUT