

Podman - Conteneurisation sans démon

Code : SYS-PODMAN-100 Durée : 3 jours (21 heures)

Podman est l'alternative moderne à Docker pour la gestion de conteneurs. **Sans démon et rootless par défaut**, il offre une sécurité renforcée tout en conservant une compatibilité CLI avec Docker. Cette formation vous apprend à créer des images avec Buildah, orchestrer des services avec Podman Compose, gérer les pods et intégrer vos conteneurs avec systemd et Kubernetes.

Vous connaissez Docker ? Podman se distingue par son **architecture sans démon**, son **exécution rootless par défaut**, le concept de **pods** (groupes de conteneurs), l'utilisation de **Buildah** pour la construction d'images, et son **intégration native avec systemd et Kubernetes**.

Public visé

DevOps, développeurs, architectes et administrateurs systèmes. Accessible aux utilisateurs de Docker souhaitant migrer vers Podman.

Prérequis

Avoir suivi cette formation, ou en maîtriser les thèmes abordés :

- Linux - les fondamentaux

Objectifs pédagogiques

À l'issue de cette formation, vous serez capable de :

- Expliquer les avantages et inconvénients de la conteneurisation
- Déployer et administrer Podman en mode rootless
- Configurer et utiliser Podman en ligne de commande
- Construire des images avec Buildah et les Containerfile
- Interagir avec les registres d'images OCI (publics et privés)
- Mettre en oeuvre et configurer des conteneurs et des pods
- Organiser la gestion des réseaux et du stockage
- Maintenir et surveiller une infrastructure de conteneurs en production

Programme

Jour 1 - Matin

Introduction aux conteneurs et à Podman

- Présentation de la conteneurisation et cas d'utilisation
- Modèles d'activation et licensing des conteneurs
- ↔ **Docker** — Architecture de Podman : **sans démon** (pas de service central), exécution directe via fork/exec
- ↔ **Docker** — Conformité OCI : images et runtimes standardisés, compatibilité avec les images Docker
- Comparaison des performances : Podman vs Docker
- Vue d'ensemble du cycle de vie d'un conteneur

Déployer et gérer les conteneurs

- Préparation au déploiement des conteneurs
- Installation de Podman sur les principales distributions Linux
- ↔ **Docker** — **Mode rootless par défaut** : user namespaces, subordinate UIDs/GIDs
- Mise en oeuvre, configuration et gestion des conteneurs
- Compatibilité CLI : commandes Podman vs commandes Docker

Exemples de travaux pratiques (à titre indicatif) :

- Déploiement de Podman et prise en main de la CLI
- Déploiement des premiers conteneurs en mode rootless
- Comparaison pratique : même commande sous Docker et Podman

Jour 1 - Après-midi

Interaction avec les conteneurs

- Gestion et création d'images de conteneurs
- ↔ **Docker** — Introduction à **Buildah** : construction d'images sans démon
- Gérer le réseau et les communications inter-conteneurs
- Exploration des logs et gestion des processus dans les conteneurs
- Comprendre les différents runtimes OCI (crun, runc)

Jour 2 - Matin

La gestion du réseau avec Podman

- ↔ **Docker** — Stack réseau Podman : **Netavark** et Aardvark-dns (remplacement de la stack Docker)
- Configuration des réseaux bridge, macvlan et host
- Communication inter-conteneurs et résolution DNS
- Sécurisation des communications entre conteneurs

La gestion du stockage avec Podman

- Introduction à la persistance des données
- Déploiement des volumes et bind mounts
- ↔ **Docker** — Gestion du stockage avec les drivers **Overlay** et **VFS**

Les images de conteneurs

- ↔ **Docker** — Construction d'images avec **Buildah** :
 - À partir d'un conteneur (méthode interactive)
 - À partir d'un Containerfile/Dockerfile
 - Commandes Buildah sans Containerfile (scripting)
- ↔ **Docker** — Gestion des registres avec **Skopeo** : inspection, copie et synchronisation d'images
- Sécurisation des connexions aux registres (TLS, authentification)
- Stocker et récupérer des images depuis des registres publics et privés

Jour 2 - Après-midi

Monitoring et maintenance des conteneurs

- Vue d'ensemble des outils de monitoring (podman stats, podman healthcheck)
- Sauvegarder et restaurer un environnement de conteneurs
- ↔ **Docker** — Gestion des conteneurs avec **systemd** : génération de fichiers unit avec `podman generate systemd`

Exemples de travaux pratiques (à titre indicatif) :

- *Création de conteneurs avec Podman et Buildah*
- *Mise en oeuvre :*
 - *Des réseaux inter-conteneurs avec Netavark*
 - *Des différentes méthodes de stockage*
 - *D'un service systemd pour un conteneur*

Jour 3 - Matin

Les pods et Podman Compose

- ↔ **Docker** — Concept de **pods** : regroupement de conteneurs partageant le même espace réseau
- Création et gestion de pods avec Podman
- ↔ **Docker** — **Podman Compose** : orchestration de services multi-conteneurs
- Notion d'architectures N-tiers avec les pods

Intégration Kubernetes et fonctionnalités avancées

- ↔ **Docker** — Génération de manifestes **Kubernetes YAML** depuis les pods (`podman generate kube`)
- ↔ **Docker** — Déploiement de manifestes Kubernetes dans Podman (`podman play kube`)
- Intégration de Podman dans les pipelines CI/CD
- Introduction à la migration Docker vers Podman

Jour 3 - Après-midi

Sécurité et bonnes pratiques

- ↔ **Docker** — Sécurité renforcée par défaut : rootless, absence de démon privilégié
- SELinux et Podman : confinement des conteneurs
- Gestion des secrets et configurations sensibles (`podman secret`)
- Bonnes pratiques de construction d'images et réduction de surface d'attaque
- Vue d'ensemble de l'écosystème : Podman Desktop, Cockpit

Exemples de travaux pratiques (à titre indicatif) :

- *Orchestration de conteneurs avec les pods et Podman Compose*
- *Génération et déploiement de manifestes Kubernetes*
- *Mise en place d'un service conteneurisé géré par systemd*
- *Analyse de sécurité d'un conteneur rootless vs root*
- *Mise en place d'un pipeline CI/CD avec Podman*

Modalités d'évaluation des acquis

En cours de formation, par des études de cas ou des travaux pratiques. En fin de formation, par un questionnaire d'auto-évaluation.

Voir aussi

Vous serez peut-être intéressés par les formations suivantes :

- [Docker - Conteneurisation et Docker Compose](#)
- [Docker - Fonctionnalités avancées](#)
- [Infrastructure as Code \(IaC\)](#)
- [LXC - Virtualisation par conteneurs Linux](#)
- [Stack Prometheus Grafana](#)
- [Virtualisation avec Proxmox](#)