

Podman ou Docker : la checklist pour choisir sans regret

Utilisez cette fiche avant d'installer, de migrer ou de standardiser votre moteur de conteneurs Linux. Elle vous aide à choisir selon vos contraintes réelles : droits, écosystème, exploitation et compatibilité.

Au sommaire

1. Clarifier votre contexte
2. Repères rapides : quel moteur correspond à votre priorité ?
3. Valider la sécurité et l'exploitation
4. Tester avant toute migration vers Podman
5. Décision finale à consigner

1. Clarifier votre contexte



Identifier les hôtes concernés

Serveurs Linux, postes de développement ou les deux.



Lister les applications à conteneuriser

Services isolés, applications multi-conteneurs, bases de données, cache.



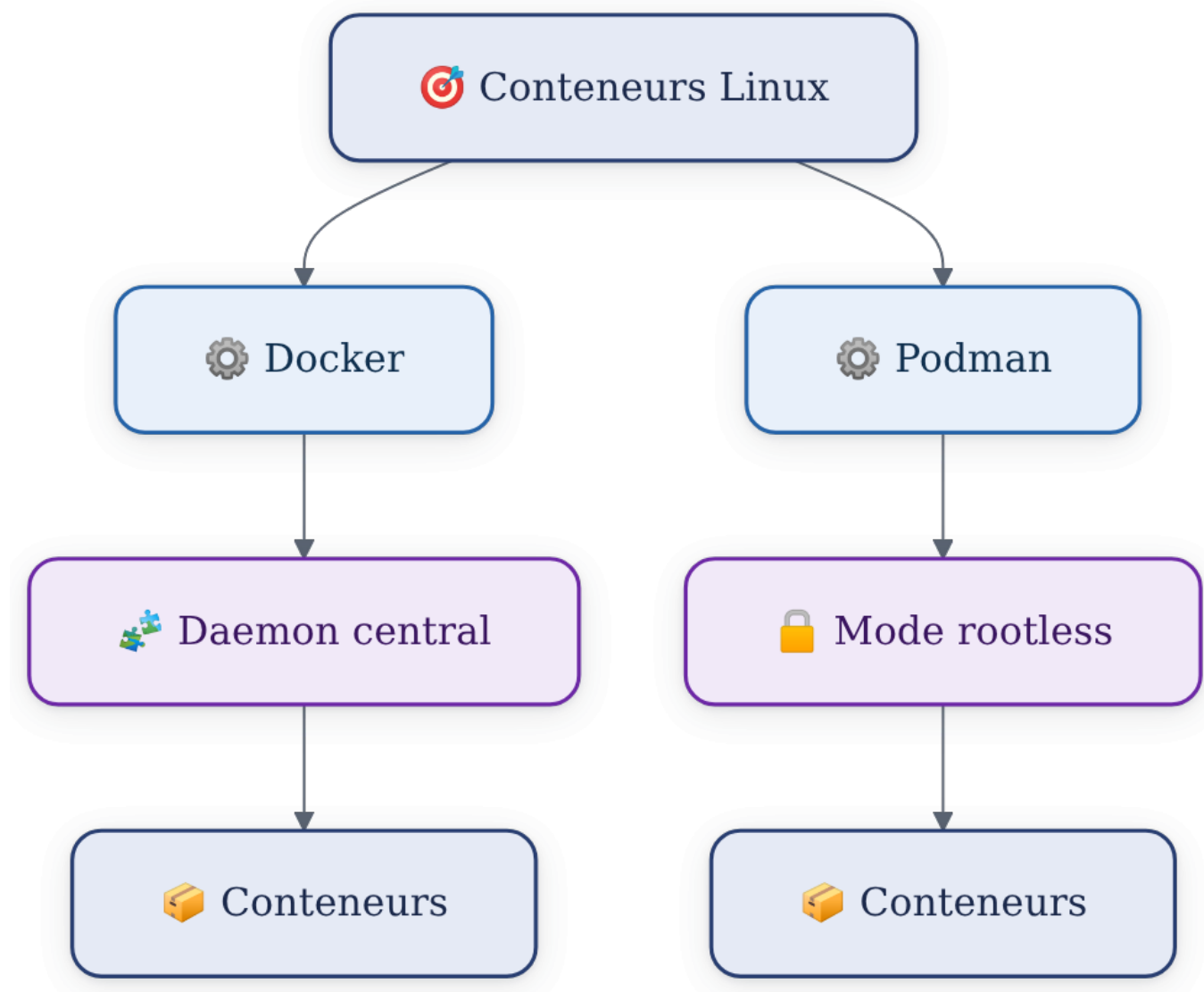
Recenser les outils déjà utilisés par l'équipe

Docker Compose, Docker Desktop, scripts, intégrations internes et API.



Définir qui administre les conteneurs

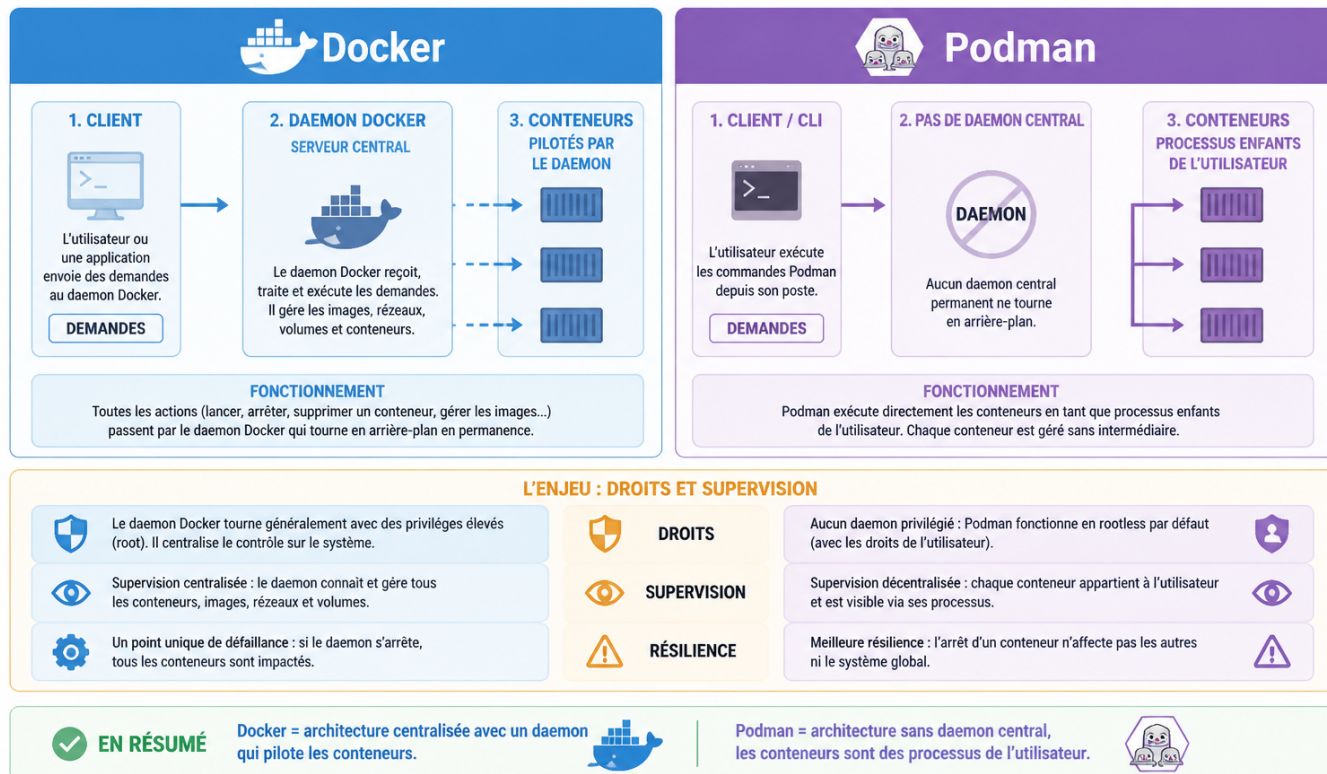
Compte utilisateur courant, administrateurs système ou pipeline d'automatisation.



INFOGRAPHIE Schéma Podman ou Docker montrant daemon central Docker et exécution rootless Podman

DOCKER ET PODMAN : COMPARAISON DES ARCHITECTURES

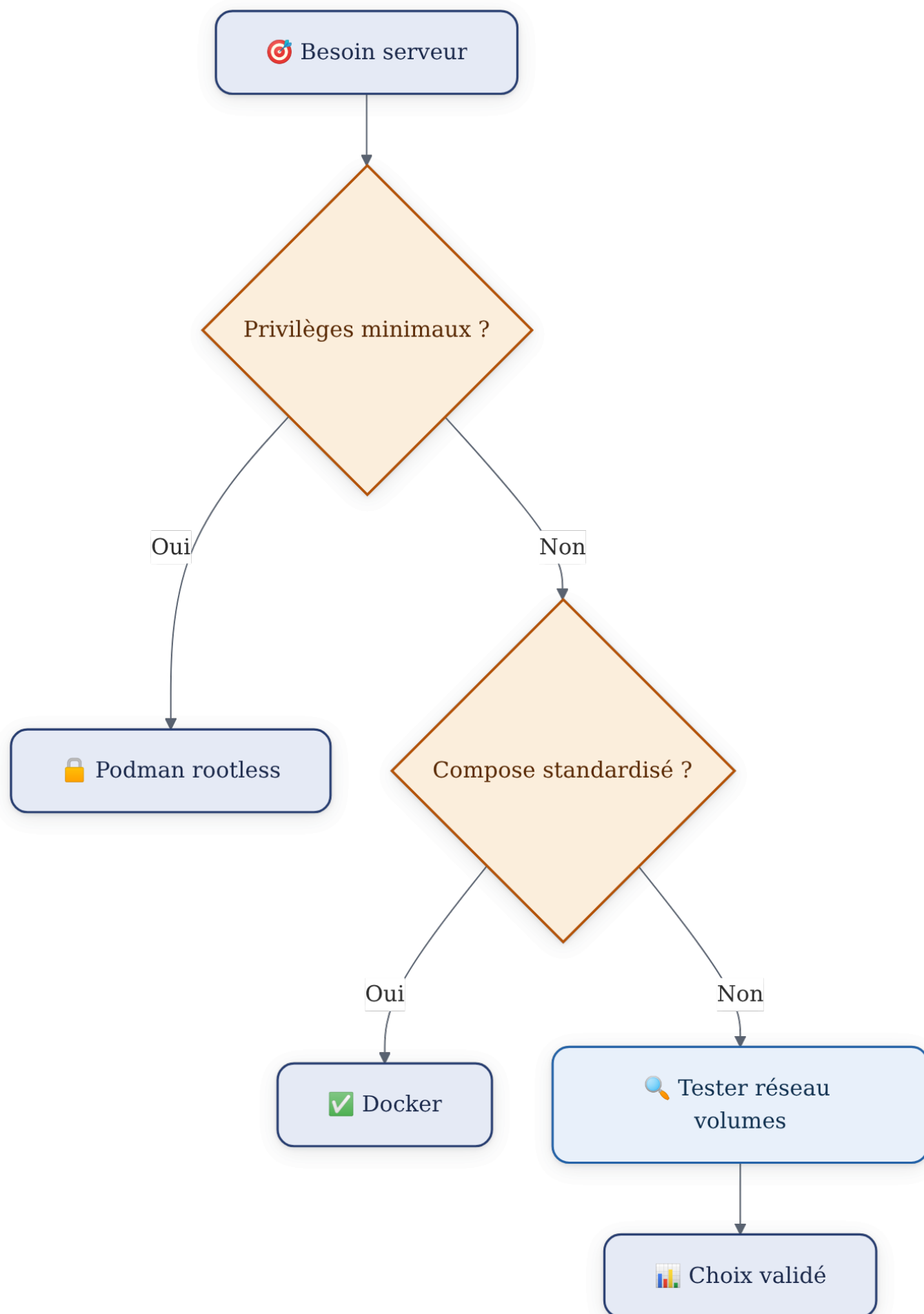
Deux approches pour exécuter des conteneurs Linux



INFOGRAPHIE Schéma comparatif Podman Docker montrant le daemon central de Docker et l'architecture daemonless de Podman.

2. Repères rapides : quel moteur correspond à votre priorité ?

Votre priorité	Orientation à privilégier	Point à vérifier
Limiter les privilèges sur un hôte Linux	Podman	Valider le mode rootless, les ports, les volumes et les besoins réseau.
Conserver un environnement Docker déjà établi	Docker	Vérifier les droits d'accès au daemon et au socket Docker.
S'appuyer fortement sur Docker Compose	Docker ou validation préalable avec Podman	Tester le fichier Compose, les réseaux et les dépendances multi-conteneurs.
Réutiliser des images OCI et des recettes de build existantes	Les deux	Tester les Dockerfile ou Containerfile dans votre contexte réel.
Intégrer les services aux mécanismes Linux	Podman	Adapter la supervision avec systemd à la version installée.



3. Valider la sécurité et l'exploitation

Choisir explicitement le modèle de droits

Avec Podman, évaluer l'exécution rootless sous le compte utilisateur courant.

Contrôler l'accès au moteur de conteneurs

Avec Docker, traiter l'accès au socket et au daemon comme un accès sensible à l'hôte.

Prévoir la supervision des services

Documenter le démarrage, l'arrêt, les journaux et le diagnostic d'incident.

Vérifier les mécanismes de confinement disponibles

Les droits du moteur ne remplacent pas les protections de l'hôte.

4. Tester avant toute migration vers Podman

Exécuter une image représentative

Confirmer que l'image OCI se récupère et démarre comme prévu.

Reconstruire l'application

Tester les Dockerfile ou Containerfile, les arguments de build et les scripts.

Valider les volumes

Contrôler les montages, les permissions et la persistance des données.

Valider les réseaux et les ports exposés

Tester les communications entre services et les accès attendus.

Tester le fonctionnement multi-conteneurs

Ne pas présumer de la compatibilité : vérifier le fichier Compose ou l'outil retenu.

Rejouer les procédures d'exploitation

Démarrage, arrêt, mises à jour, sauvegardes, journaux et retour arrière.

! Rootless ne signifie pas « sans risque »

Le mode rootless réduit le besoin d'utiliser les privilèges administrateur pour les opérations courantes, mais il ne dispense pas de maintenir les images à jour, filtrer le réseau, protéger les secrets, contrôler les volumes et limiter les capacités accordées aux conteneurs.

5. Décision finale à consigner

Choisissez Podman si la priorité est une exploitation Linux orientée privilèges minimaux et sans daemon central permanent.

Ne décidez pas sur la seule proximité des commandes : réseau, volumes, scripts et exploitation doivent être validés sur votre application.

Conservez Docker si votre efficacité dépend déjà fortement de Docker Compose, Docker Desktop ou d'outils construits autour de l'écosystème Docker.

Ni Podman ni Docker ne remplacent, à eux seuls, un orchestrateur de production.