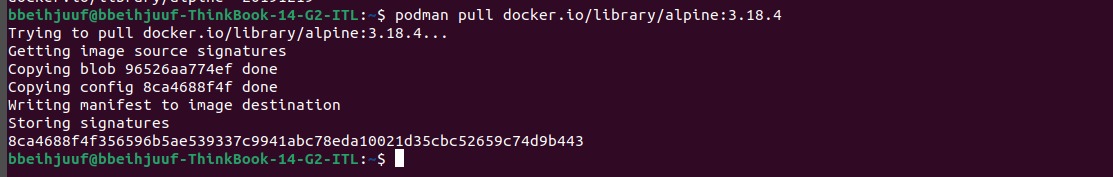
cle

|  |
| --- |
| Podman le moteur de conteneur alternatif à Docker  Les conteneurs sont devenus essentiels pour les environnements de développement, permettant le déploiement d’applications de manière rapide, portable et reproductible. Historiquement dominé par Docker, ce domaine a vu émerger d’autres outils répondant à des besoins spécifiques, notamment en matière de sécurité et de flexibilité.  Podman, acronyme de POD Manager, est une alternative open source qui se distingue par sa conception sans démon (daemonless), son exécution sans privilèges root (rootless) et son intégration native avec les pods, un concept clé de Kubernetes. Développé par Red Hat et soutenu par une large communauté, Podman se veut à la fois une solution moderne et compatible avec les outils existants comme Docker.  Un pod dans Podman est un groupe de conteneurs qui partagent les mêmes ressources, notamment les espaces de noms, le réseau et le contexte de sécurité, similaire au concept de pod dans Kubernetes. Chaque pod inclut un conteneur spécial appelé “infra” ou “pause”, qui maintient les espaces de noms associés au pod et permet à Podman de connecter d’autres conteneurs au pod. Ce conteneur “infra” ne fait rien d’autre que de rester en veille, mais il est essentiel pour la gestion des ressources partagées au sein du pod.  L’utilisation de pods présente plusieurs avantages, notamment la capacité de regrouper des conteneurs qui doivent interagir étroitement. Par exemple, si vous avez plusieurs conteneurs nécessitant l’accès à une base de données MariaDB, vous pouvez les placer dans un même pod. Ainsi, ces conteneurs peuvent communiquer entre eux via l’adresse localhost du pod, sans exposer la base de données sur un réseau accessible, renforçant ainsi la sécurité.  En résumé, les pods dans Podman offrent une gestion flexible et sécurisée des conteneurs, en permettant à plusieurs conteneurs de partager des ressources communes tout en maintenant une isolation appropriée. |
| Installation de Podman sur Linux Podman est largement compatible avec les principales distributions Linux et peut être installé à l'aide des gestionnaires de paquets de ces systèmes. Cependant, il arrive que les versions proposées dans les dépôts officiels ne soient pas toujours les plus récentes. Si vous souhaitez bénéficier des dernières fonctionnalités ou des corrections de bugs, il est possible de compiler Podman directement à partir de son code source. Installation depuis les dépôts officiels sudo apt update  sudo apt install -y podman Vérification de l’installation de Podman Il suffit de lancer la commande suivante :  podman –version  C:\Users\Aby Soumare Diouf\Desktop\Master1 - STIC UAHB\Pogman\WhatsApp Image 2025-03-29 à 00.43.55_fbf578b6.jpg Commandes de base de PodmanRechercher des images dans les registry Pour trouver des images dans la plupart des registry, on utilise la commande search :    Pour obtenir la liste des images officielles, on ajoute l’option --filter is-official :  podman search --filter is-official alpine  C:\Users\Aby Soumare Diouf\Desktop\Master1 - STIC UAHB\Pogman\WhatsApp Image 2025-03-30 à 03.13.30_ec136b2d.jpg  Pour obtenir la liste des tags disponibles, on ajoute l’option --list-tags :  podman search --list-tags --limit 10 docker.io/library/alpine  C:\Users\Aby Soumare Diouf\Desktop\Master1 - STIC UAHB\Pogman\WhatsApp Image 2025-03-29 à 00.43.55_32a44bc6.jpg Télécharger une image |

Pour télécharger l’image Ubuntu possédant le tag 3.18.4 depuis le Docker Hub, on utilise la commande pull :

podman pull docker.io/library/alpine:3.18.4



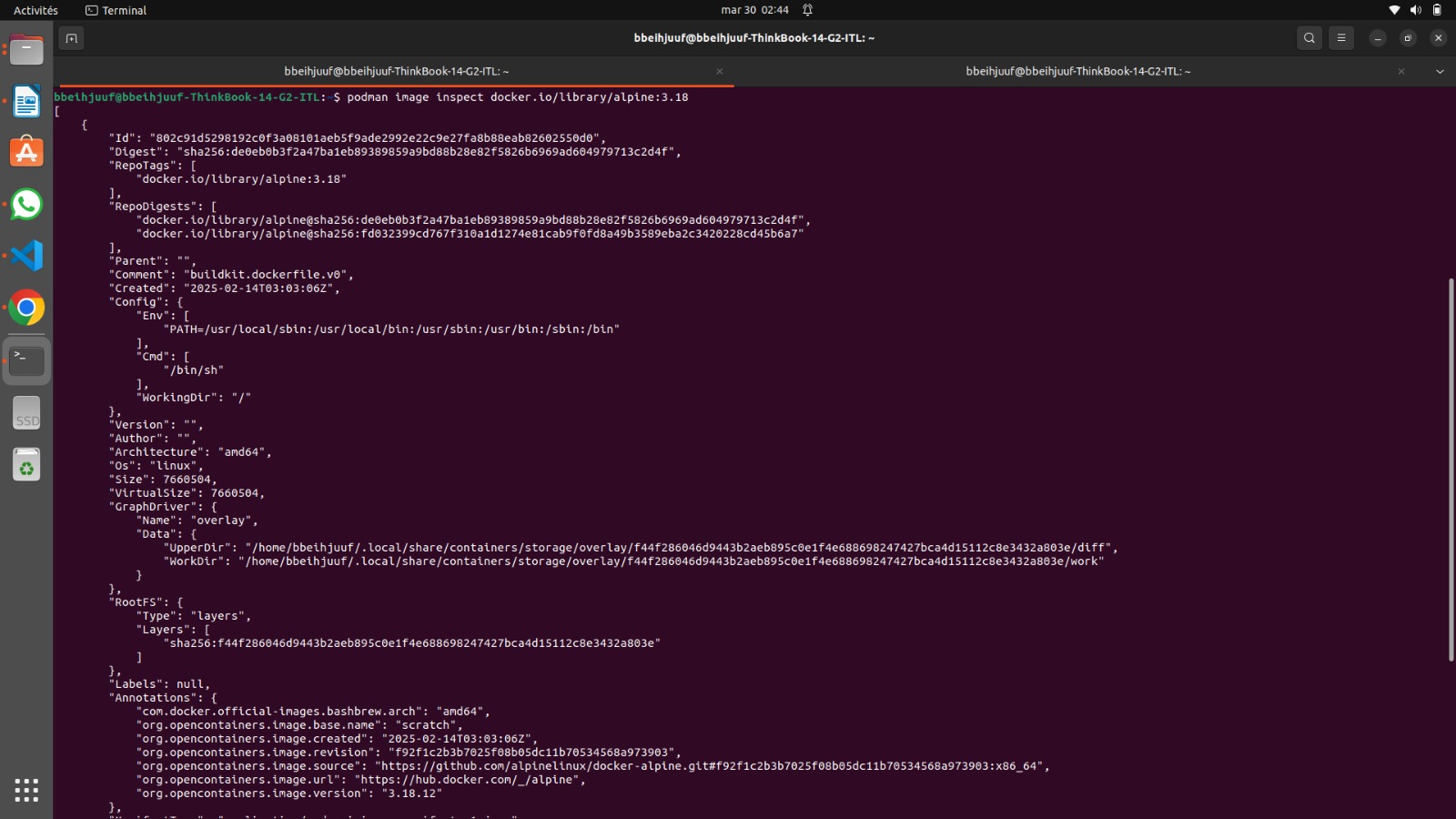
## Lister les images

Pour afficher toutes les images disponibles localement, on utilise la commande images :



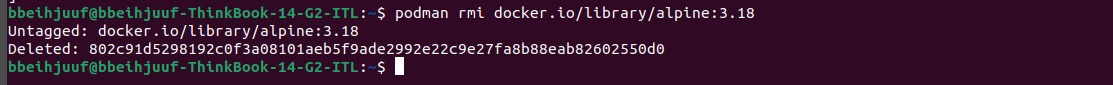
Pour obtenir des informations sur une image, on utilisera la commande image inspect :

podman image inspect docker.io/library/alpine:3.18



Pour détruire les images locales, on utilise la commande rmi :

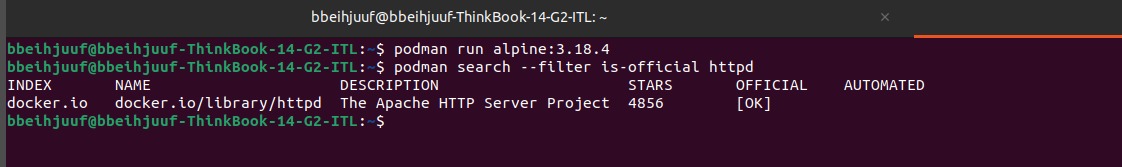
podman rmi docker.io/library/alpine:3.18



# Gestion des conteneurs

## Lancer une image de conteneur

Pour lancer par exemple une image de conteneur alpine on utilise la commande run :

podman run alpine:3.18.4

## Lancer le conteneur en mode interactif

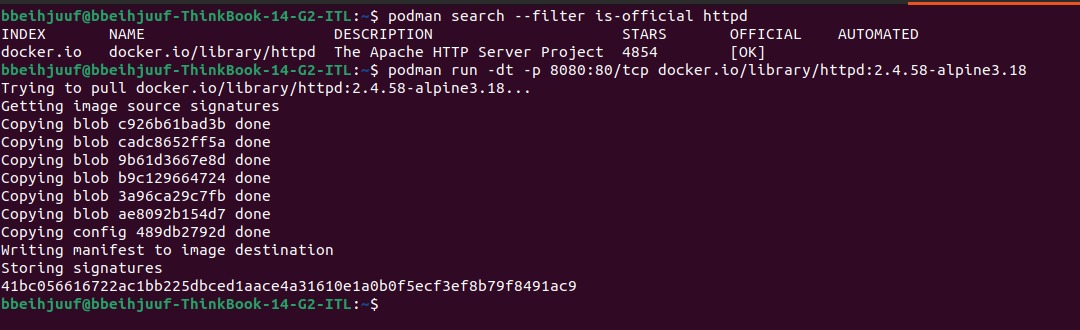
Pour lancer une exécution interactive, c’est-à-dire qu’on se retrouve avec l’invite de commande du conteneur, on utilise l’option -it :

podman run -it alpine:3.18.4 sh

## Lancer le conteneur en tâche de fond

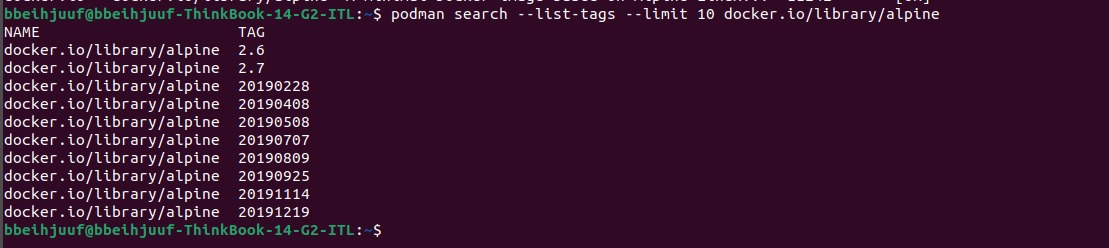
Pour lancer une image de conteneur en tache de fond, il faut utiliser l’option -dt. Pour cela, nous aurons besoin d’un conteneur httpd :

podman search --filter is-official httpd



Il est possible aussi de le lancer dans unautre port

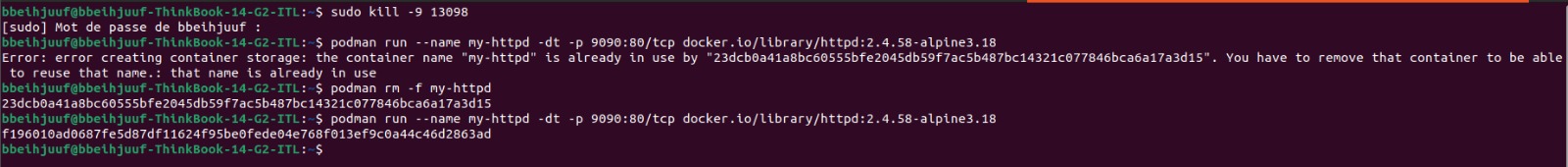
podman run -dt -p 9090:80/tcp docker.io/library/httpd:2.4.58-alpine3.18



Il faut déjà vérifier si sur le port aucun processus n’y est lancée si c’est le cas faire un kill

sudo lsof -i D:9090

sudo kill -9 13098



On peut également le lancer avec curl mais préablement il faut l’installer

curl <http://localhost:8080>

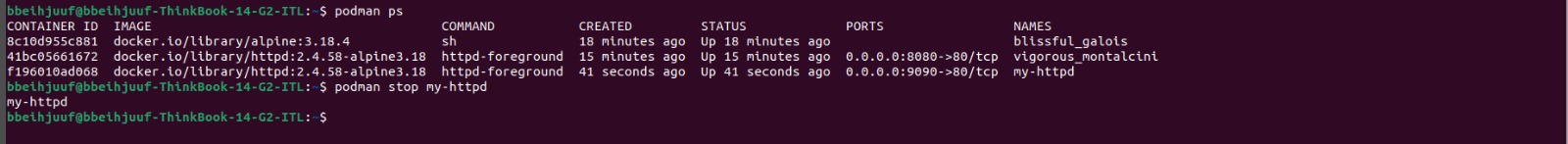
Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

## Lister les conteneurs en cours d’exécution

Pour lister les conteneurs en cours, il faut utiliser commande ps :

podman ps



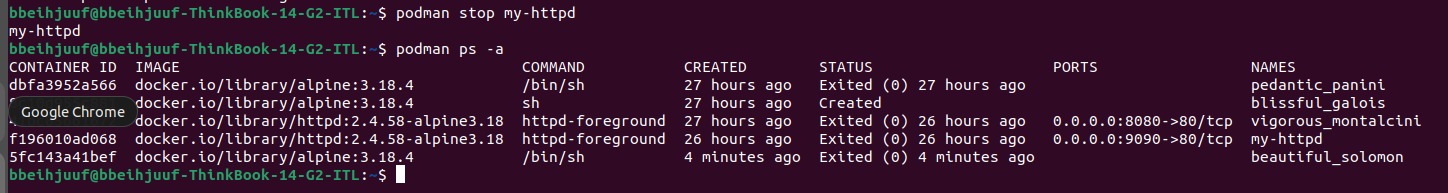
## Stopper/Démarrer un conteneur

Pour stopper un conteneur, il faut utiliser commande stop :

podman stop my-httpd

Pour vérifier que notre conteneur juste stoppé, il faut utiliser l’option -a à la commande ps :

podman ps –a



## Afficher les logs d’un conteneur

Pour afficher les logs du conteneur, on utilise la commande logs :

podman logs my-httpd

## Se connecter à un conteneur en cours d’exécution

Pour se connecter à un conteneur tournant en tache de fond :

podman exec -it my\_httpd sh

## Détruire un conteneur

Pour détruire un conteneur, il faut utiliser la commande rm :

podman rm my-httpd

Si vous voulez détruire un conteneur en cours d’execution il faudra ajouter un –f

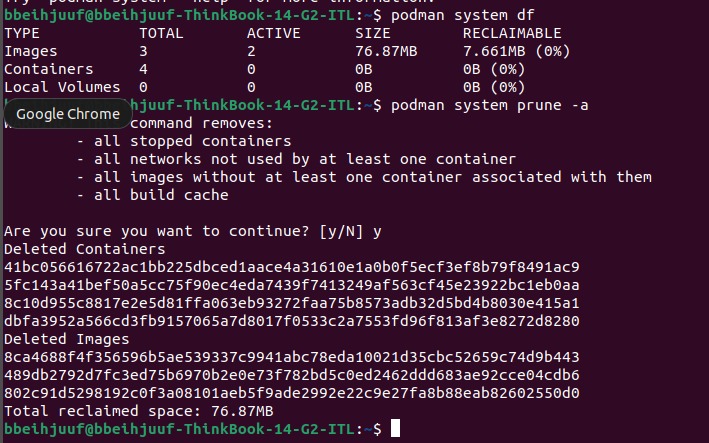
podman rm -f my-httpd

## Faire le ménage

Podman comme Docker utilise pas mal d’espace disque. Heureusement, nous avons des commandes pour contrôler l’évolution de l’utilisation de l’espace disque. À commencer par la commande system ps.

podman system

podman system prune df



## Gestion des Pods sous Podman

Pour créer un pod, il faut utiliser la commande create auquel on ajoute l’option --name pour le nommer :

podman pod create --name myFirstPod -p 8080:80

## Attacher un conteneur à un pod

Pour attacher un conteneur à un pod, il faut, au moment du lancement du conteneur, ajouter l’option --pod :

podman run --pod myFirstPod --name my-hhtpd -dt docker.io/library/httpd:2.4.58-alpine3.18

## Lister les pods

Pour lister les pods en cours, on utilise la commande ls :

podman pod ls

