

VINCENT STEBEN

Fonctions de travail et multimédia

420-0D3-GG gr. 03405

PROJET 1 : CRÉATION ET GESTION D'UN ENVIRONNEMENT VIRTUEL MULTI-SYSTÈME

Travail présenté à

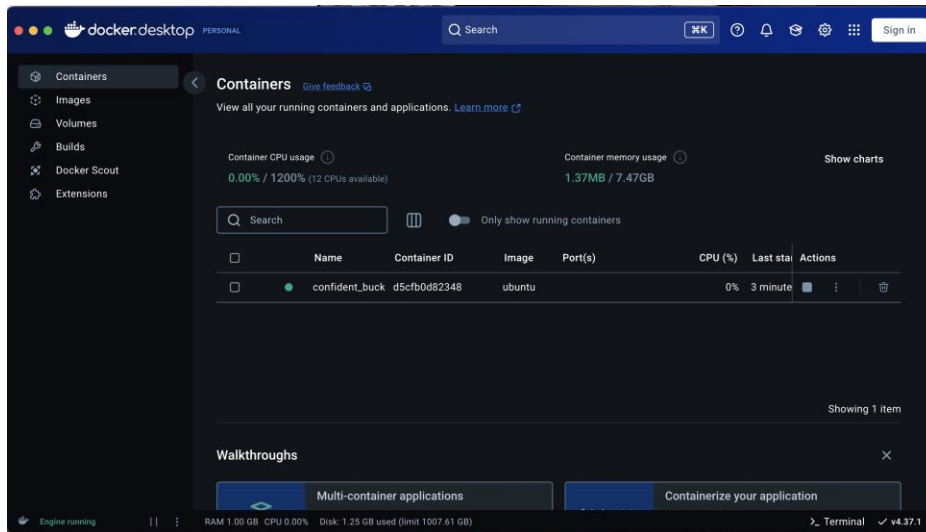
Madame Amal Boukdhir

Développement d'applications mobiles

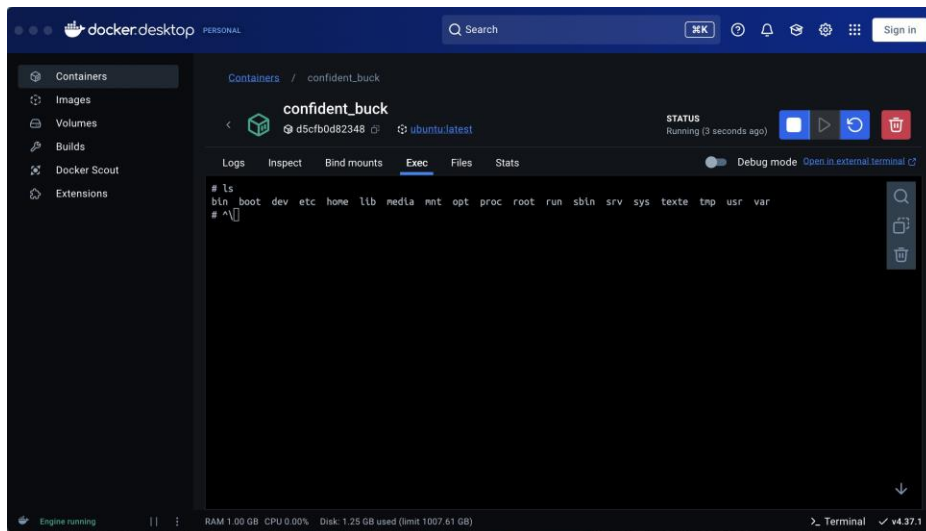
Cégep Gérard-Godin

6 janvier 2025

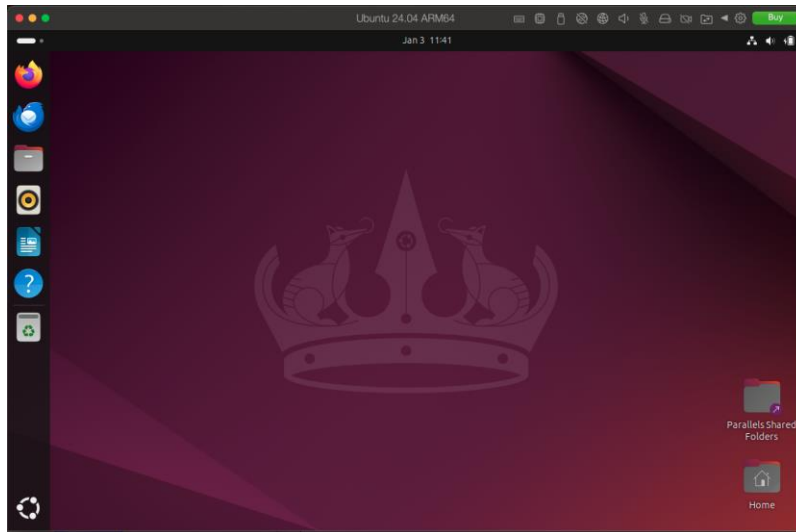
1. Installation de Docker sur macOS



2. Installation d'un système d'exploitation Linux sur Docker



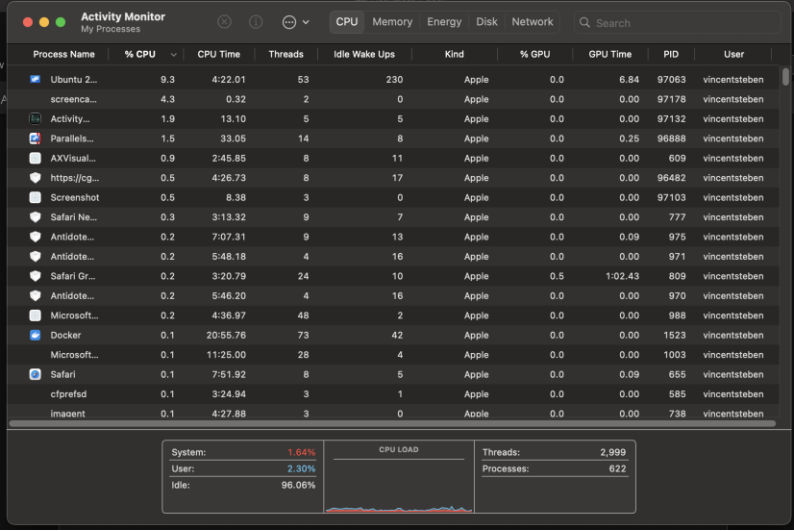
3. Installation d'un hyperviseur et d'une machine virtuelle



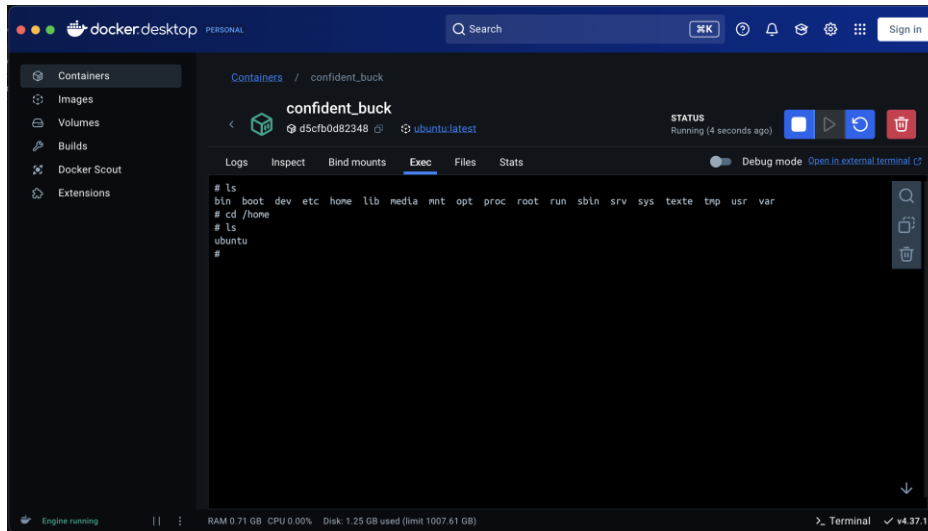
4. Comparaison des systèmes d'exploitation (Docker vs Hyperviseur)

Docker est un système de virtualisation par conteneurs qui agit comme intermédiaire entre ma machine (MacOS) et un autre système d'exploitation que je désirerais utiliser et qui est normalement incompatible (Linux/Ubuntu dans ce cas-ci). Comme l'application utilisée est indépendante, Docker lui-même ne consomme que très peu de ressources de l'ordinateur.

L'hyperviseur quant à lui (Parallels dans ce cas-ci) contient ses propres applications et interfaces comme si on installait un autre mini-ordinateur sur sa machine. Parallels offre même par défaut les options les plus populaires comme Windows 11 ou Ubuntu en lien direct sur sa page d'accueil. Il est donc beaucoup plus convivial et facile d'utilisation pour un débutant mais consomme par le fait même beaucoup plus de ressources sur l'ordinateur hôte. Près de 100 fois plus que Docker comme le démontre cette image (9,3% pour Parallels contre 0,1% pour Docker).



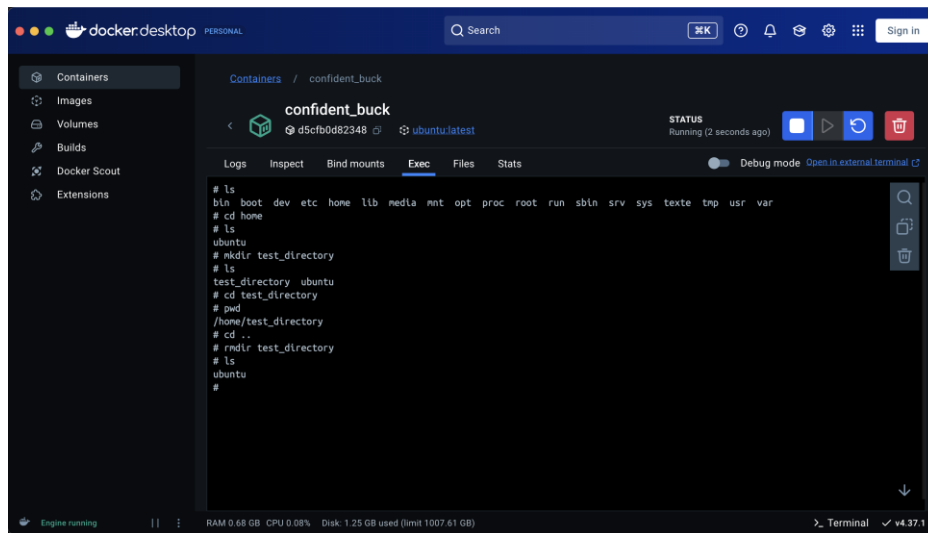
5. Utilisation de commandes de base Linux



The screenshot shows the Docker Desktop interface. On the left is a sidebar with navigation options: Containers, Images, Volumes, Builds, Docker Scout, and Extensions. The main area displays the 'confident_buck' container, which is running. Below the container name, there are tabs for Logs, Inspect, Bind mounts, Exec, Files, and Stats. The 'Exec' tab is active, showing a terminal window. The terminal output is as follows:

```
# ls
bin boot dev etc home lib media mnt opt proc root run/sbin srv sys texte tmp usr var
# cd /home
# ls
ubuntu
#
```

At the bottom of the Docker Desktop window, system statistics are visible: RAM 0.71 GB, CPU 0.00%, and Disk 1.25 GB used (limit 1007.61 GB). The version v4.37.1 is also shown.



This screenshot shows the same Docker Desktop interface as the previous one, but with more terminal output. The terminal session continues with the following commands and output:

```
# mkdir test_directory
# ls
test_directory ubuntu
# cd test_directory
# pwd
/home/test_directory
# cd ..
# rmdir test_directory
# ls
ubuntu
#
```

The system statistics at the bottom remain the same: RAM 0.68 GB, CPU 0.08%, and Disk 1.25 GB used (limit 1007.61 GB). The version v4.37.1 is also shown.

6. En ordre d'apparition sur la capture d'écran: ls (list) affiche le contenu du répertoire où vous vous trouvez, puis cd (change directory) permet d'entrer dans l'un de ces répertoires (home). Ls réaffiche ensuite le contenu de ce nouveau répertoire. Mkdir (make directory) crée un nouveau répertoire nommé test_directory et ls l'affiche pour confirmer qu'il est bien créé. Cd entre dans le nouveau répertoire et pwd (print working directory) affiche le nom complet du répertoire où nous nous trouvons pour confirmer le mouvement. Finalement, cd .. nous permet de ressortir du répertoire pour ensuite le supprimer avec rmdir (remove directory). Un dernier ls nous permet de constater que test_directory n'est plus là.