

Université Cheikh Anta Diop



École Supérieure Polytechnique Département Génie Informatique

Année Universitaire 2024-2025

DIC2 IABD

Infrastructure de Virtualisation

Atelier 01 : Plateforme de Conteneurisation Docker

Dr Mandicou BA

L'objectif de cet atelier est de vous permettre de se familiariser avec la plateforme de conteneurisation Docker.

À la fin de cet atelier chaque étudiant doit être en mesure de pouvoir :

- ✓ Installer et de configurer correctement Docker
- ✓ Gérer efficacement des conteneurs en mode cli et graphique (**Docker cli, Portainer**).
- ✓ Télécharger ou téléverser des images Docker dans le Hub de Docker (**Docker Hub**)
- ✓ Gérer l'interconnexion entre conteneurs et leurs déploiements (**Docker Compose**)
- ✓ Orchestrer et gérer des clusters avec (**Docker Swarm**)

Pré-requis :

1. Une machine virtuelle ou physique tournant sous l'os Debian 12 (Bookworm).
2. Une connexion internet sur la machine
3. un compte sur Docker Hub

Ressources :

1. <https://www.mandicouba.net/tools/virtualbox>
2. <https://www.mandicouba.net/os/debian>
3. <https://hub.docker.com/>

Tâche 1 : Mise en place d'une plateforme Docker sous Debian 12

1. Mise à jour de la liste de paquets existante

Avant de procéder à l'installation de Docker, commencer par mettre à jour la liste des paquets de Debian.

```
mandicou@Mandicou :~$ *****
```

2. Permettre à la commande apt d'utiliser les paquets via HTTPS

Installer **Docker** depuis le référentiel officiel afin d'être sûr de disposer la dernière version car les dépôts de **Debian** peuvent ne pas posséder la dernière version. Commencer par installer les prérequis qui vont vous permettre d'utiliser la commande **apt** pour installer des paquets à l'aide du protocole **https**.

```
mandicou@Mandicou :~$ *****
```

3. Ajout de la clé GPG du référentiel officiel de Docker

La commande **apt** peut utiliser **https** comme mode de transport des paquets. Ajouter la clé **GPG** du répertoire officiel de Docker afin de permettre le chiffrement et la signature des données.

```
mandicou@Mandicou :~$ *****
```

Ajouter le dépôt officiel de **Docker** dans la liste des **dépôts** du système.

```
mandicou@Mandicou :~$ *****
```

4. Mise à jour de la liste de paquets

Mettre à jour la liste des **paquets**

```
mandicou@Mandicou :~$ *****
```

5. Installation de Docker

Tout est prêt, passer maintenant à l'installation de **Docker** en utilisant la commande **apt**

```
mandicou@Mandicou :~$ *****
```

6. Vérification de l'état de Docker

Vérifier si tout fonctionne correctement par exemple vérifier que le démon Docker a bien démarré de mêmes si le processus de démarrage automatique est activé afin de permettre a Docker de se lancer directement après le démarrage du Système.

```
mandicou@Mandicou :~$ *****
```

7. Utiliser Docker sans Sudo

Par défaut, une commande **Docker** est toujours précédée du mot clé **sudo** autrement dit une commande **Docker** ne peut être exécutée que par le super admin **root** ou par un utilisateur du groupe docker. Lever cette contrainte afin d'utiliser **Docker** sans disposer des privilèges du super admin. Par défaut une fois que Docker est installé le groupe docker est automatiquement créé dans le doute créer le groupe.

```
mandicou@Mandicou :~$ *****
```

De même le socket utilisé par **Docker** doit appartenir au groupe docker. Définir l'appartenance.

```
mandicou@Mandicou :~$ *****
```

Ajouter l'utilisateur courant au groupe **docker** pour lui permettre d'utiliser les commandes docker sans au préalable taper **sudo**

```
mandicou@Mandicou :~$ *****
```

```
mandicou@Mandicou :~$ *****
```

Appliquer les changements et redémarrer **Docker**

```
mandicou@Mandicou :~$ sudo systemctl restart docker
```

8. L'aide sous Docker

Afficher la liste des commandes Docker

```
mandicou@Mandicou :~$ *****
```

Pour obtenir de l'aide sur une commande docker il faut utiliser la commande docker help suivis de la commande dont on veut afficher le manuel :

```
mandicou@Mandicou :~$ *****
```

```
mandicou@Mandicou:~$ *****
```

Contenu du fichier Dockerfile

```
*****
*****
*****
*****
*****
*****
*****
```

```
mandicou@Mandicou :~$ *****
```

PS : Afin de s'assurer que vous avez réussi parfaitement la **Tâche 2**, vous devez accéder au serveur web depuis le navigateur ou bien depuis le terminal de la machine Docker

Tâche 3 : Docker Hub

1. Recherche d'une image

Docker Hub est le dépôt distant géré par l'organisation Docker, vous pouvez y déposer vos images ou bien même télécharger des images depuis les serveurs Docker Hub. Rechercher une image MYSQL dans le Docker Hub.

```
mandicou@Mandicou :~$ *****
```

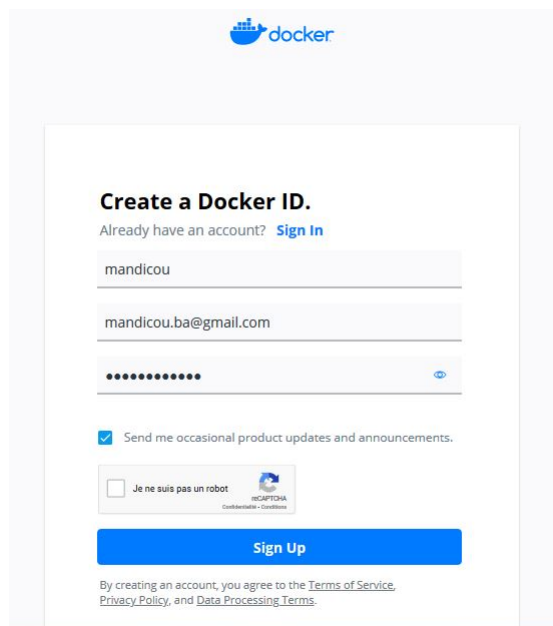
2. Téléchargement du conteneur

Télécharger une image MYSQL depuis Docker Hub.

```
mandicou@Mandicou :~$ *****
```

3. Téléverser une image dans le docker Hube

Docker Hub ne permet pas uniquement de télécharger des images mais permet aussi de sauvegarder en ligne nos propres images. Pour cela il faut disposer d'un compte docker hub. Créer un compte Docker Hub depuis le site officiel de docker.



The screenshot shows the Docker Hub registration page. At the top is the Docker logo. The main heading is 'Create a Docker ID.' Below it, a link says 'Already have an account? Sign In'. There are three input fields: a username field containing 'mandicou', an email field containing 'mandicou.ba@gmail.com', and a password field with masked characters and a strength indicator. Below the password field is a checkbox labeled 'Send me occasional product updates and announcements.' which is checked. Underneath is a CAPTCHA section with the text 'Je ne suis pas un robot' and a CAPTCHA image. At the bottom is a large blue 'Sign Up' button. Below the button, small text states: 'By creating an account, you agree to the Terms of Service, Privacy Policy, and Data Processing Terms.'



Welcome Back

Sign in with your Docker ID

mandicou

.....

Sign In

[Forgot Docker ID or Password?](#) | [Sign Up](#)

Search for great content (e.g., mysql)

[Explore](#) [Repositories](#) [Organizations](#) [Get Help](#) mandicou

Welcome to Docker Hub

Download and Take a Tutorial

Get started by downloading Docker Desktop, and learn how you can build, tag and share a sample image on Hub.

Get started with Docker Desktop

How likely are you to recommend Docker Hub to a co-worker?

Not at all likely 0 1 2 3 4 5 6 7 8 9 10 Extremely likely

Une fois le compte crée, se connecter a votre dépôt distant docker

```
mandicou@Mandicou :~$ *****
```

Âpres avoir connecter à votre dépôt distant, il faut y déposer une image pour cet exemple il faut sauvegarder sur Docker Hub votre image **premiere-image**.

```
mandicou@Mandicou :~$ *****
```

Tâche 4 : Docker Compose

Docker compose est un outil qui permet de gérer les conteneurs comme un ensemble de services inter connectés à l'aide d'un fichier YAML. Pour cet exemple il faut gérer l'interconnexion des deux connecteurs précédemment créés à l'aide du fichier **Dockerfile** et de l'image **MYSQL** téléchargée depuis Docker Hub.

1. Installation de docker compose

Docker compose ne faisant pas parti des outils de base l'or de l'installation de Docker, il faut l'installer.

```
mandicou@Mandicou :~$ *****
```

```
mandicou@Mandicou :~$ *****
```

Vérifier que docker compose a bien été installé.

```
mandicou@Mandicou :~$ *****
```

2. création du fichier docker-compose.yml

Docker compose permet de gérer en même temps plusieurs images Docker à l'aide du fichier **docker-compose.yml**. Utiliser les deux images précédemment créer à l'aide du fichier **Dockerfile** et l'image **MySQL** obtenu à l'aide de Docker Hub. Une fois le fichier vous pouvez démarrer et interconnecter les conteneurs en une seule commande. Créer le fichier **docker-compose.yml**

```
mandicou@Mandicou :~$ *****
```

Contenu du fichier dockerfile

```
*****
*****
*****
*****
*****
*****
*****
*****
```

Contenu du fichier Index.php

```
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

3. lancement du docker compose

Lancer **Docker Compose** une fois lancé les deux conteneur seront automatiquement lancés.

```
mandicou@Mandicou :~$ *****
```

PS : Afin de s'assurer que vous avez réussi parfaitement la **Tâche 4**, vous devez accéder au serveur web depuis le navigateur ou bien depuis le terminal de la machine Docker et voir le contenu de la base de données MySQL.

Tâche 5 : Docker Swarm

L'objectif est de créer un cluster Docker et y déployer un service avec le système d'orchestration de conteneurs natif de Docker(Docker Swarm).

Prérequis : En plus de la machine Debian 12 où Docker est installé qui sera notre nœud principal (Master), il nous faut également une deuxième machine avec un OS de votre choix où Docker est correctement installé. Cette deuxième machine sera notre nœud de travail (Worker).

PS : Pour gagner du temps, vous pouvez utiliser **Docker Machine** pour créer le nœud de travail (Worker)

1. Initialiser Docker Swarm

La phase d'initialisation se fait sur le nœud principal et va vous permettre de récupérer le token utilisé pour ajouter des nœuds de travail."

```
mandicou@Mandicou :~$ *****
```

2. Ajouter un nœud worker au Swarm

Utiliser le token fourni par la commande précédente et ajouter la deuxième machine en tant que nœud de travail.

```
mandicou@Mandicou :~$ *****
```

3. Afficher les informations du clusters

```
mandicou@Mandicou :~$ *****
```

4. Déploiement d'un service sur le cluster

Déployer un service simple basé sur l'image nginx qui sera accessible depuis le port 80 :

```
mandicou@Mandicou :~$ *****
```

5. Vérifier l'état du service

```
mandicou@Mandicou :~$ *****
```

6. Mise à l'échelle du service

Augmenter ou diminuer le nombre de réplicas de votre service en utilisant la commande scale.

```
mandicou@Mandicou :~$ *****
```

PS : Afin de s'assurer que vous avez réussi parfaitement la **Tâche 5**, vous devez accéder au serveur Nginx depuis un navigateur ou bien depuis le terminal d'une des machine du cluster et vérifier la tolérance aux pannes.

Tâche 6 : Portainer

Il est possible de gérer les conteneurs docker depuis un interface graphique, plusieurs outils permettent de le faire nous allons prendre un des outils qui s'appelle **Portainer** qui va permettre depuis son interface web de gerer les images de Docker.

1. installation de Portainer

Commencer par créer un volume qui va accueillir les données de **Portainer**.

```
mandicou@Mandicou :~$ *****
```

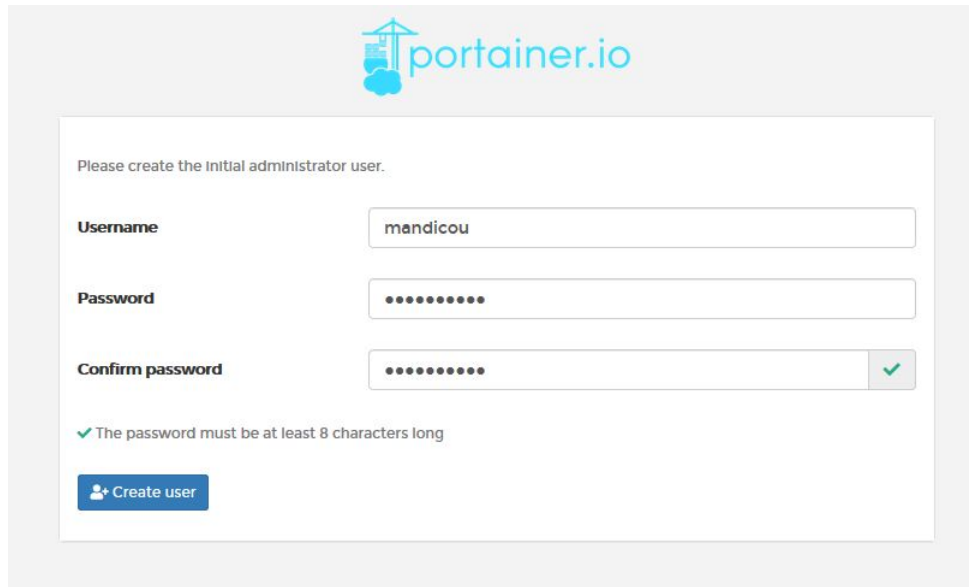
Après avoir créer le volume, démarrer Portainer depuis le port 9000

```
mandicou@Mandicou :~$ *****
```

2. Utilisation de Portainer

une fois que Portainer est lancé il faut ouvrir votre navigateur et saisir l'adresse suivante

mandicou@Mandicou :~\$ *****




portainer.io

Please create the initial administrator user.

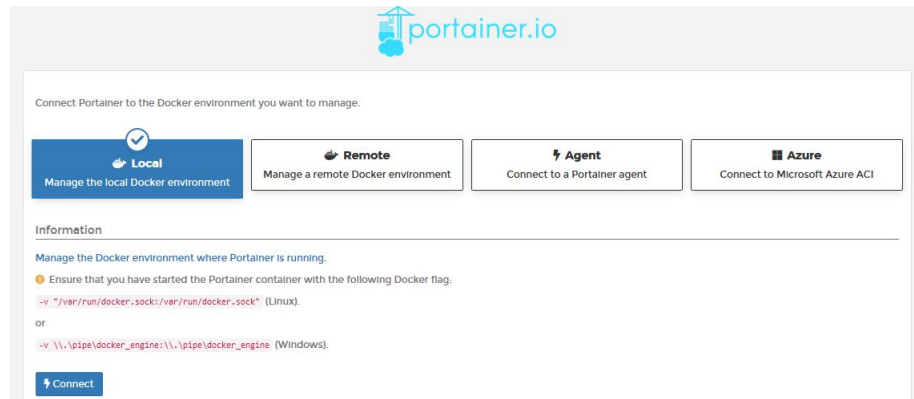
Username

Password

Confirm password 

✓ The password must be at least 8 characters long

[Create user](#)



portainer.io

Connect Portainer to the Docker environment you want to manage.

[Local](#) [Remote](#) [Agent](#) [Azure](#)

Manage the local Docker environment | Manage a remote Docker environment | Connect to a Portainer agent | Connect to Microsoft Azure ACI

Information

Manage the Docker environment where Portainer is running.

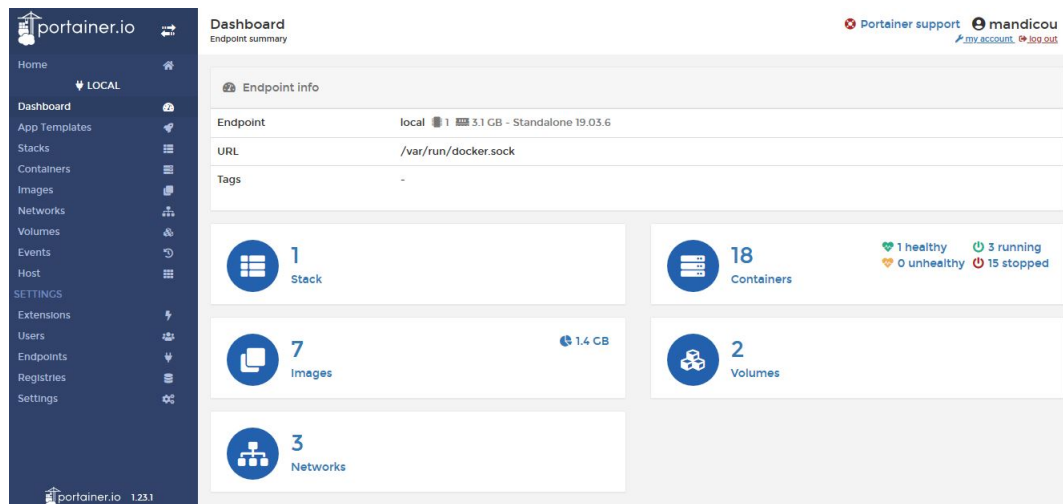
Ensure that you have started the Portainer container with the following Docker flag:

```
-v "/var/run/docker.sock:/var/run/docker.sock" (Linux).
```

or

```
-v "\\.\pipe\docker_engine:\\.\pipe\docker_engine" (Windows).
```

[Connect](#)



portainer.io

Dashboard
Endpoint summary

Portainer support | mandicou
[my account](#) | [log out](#)

Endpoint info

Endpoint	local	1	3.1 GB	Standalone 19.03.6
URL	/var/run/docker.sock			
Tags	-			

Stacks 1

Images 7 1.4 GB

Volumes 2

Containers 18
1 healthy | 0 unhealthy | 3 running | 15 stopped

Networks 3

portainer.io 1.23.1

PS : Afin de s'assurer que vous avez parfaitement réussi la **Tâche 6**, vous devez tester certaines fonctionnalités de Portainer, par exemple créer une image, lancer un conteneur, effectuer un déploiement rapide d'application à l'aide de l'interface graphique.