

## **Exercice 2 : Manipulation avancée des conteneurs Docker**

### **Questions :**

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Étapes réalisées</b>	<b>2</b>
2.1	1. Lancement d'un conteneur Ubuntu en mode interactif .....	2
2.2	2. Installation de curl et vim dans le conteneur .....	3
2.3	3. Création d'un fichier test.txt .....	3
2.4	4. Sortie du conteneur sans l'arrêter.....	4
2.5	5. Copie du fichier test.txt vers la machine hôte.....	5
2.6	6. Modification du fichier et recopie dans le conteneur .....	5
2.7	7. Reconnexion au conteneur et vérification.....	5
2.8	8. Création d'une nouvelle image à partir du conteneur modifié .....	6
2.9	9. Lancement d'un nouveau conteneur basé sur l'image personnalisée.....	6
2.10	10. Vérification des modifications dans le nouveau conteneur .....	7
2.11	Bonus : Exploration des statistiques en temps réel.....	8
<b>3</b>	<b>Conclusion</b>	<b>9</b>

# 1 Introduction

Cet exercice avait pour objectif de se familiariser avec les fonctionnalités avancées de Docker, notamment la manipulation des conteneurs en mode interactif, la persistance de fichiers, la création d'images personnalisées et l'exploration des statistiques en temps réel.

## 2 Étapes réalisées

### 2.1 1. Lancement d'un conteneur Ubuntu en mode interactif

Commande utilisée :

```
docker run -it ubuntu:latest
```

```
PS C:\Users\user\Desktop\docker-exercice2-tp1> docker run -it --name ubuntu-exercice ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
4b3ffd8ccb52: Pull complete
Digest: sha256:66460d557b25769b102175144d538d88219c077c678a49af4afca6fbfc1b5252
Status: Downloaded newer image for ubuntu:latest
```

FIGURE 1 – Conteneur Ubuntu lancé en mode interactif

### 2.2 2. Installation de curl et vim dans le conteneur

Commandes exécutées dans le conteneur : apt

```
update
apt install -y curl vim
```

```

root@a5202d3f0d2b:/# apt update
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [33.1 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1167 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1587 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2624 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2754 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [35.9 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1937 kB]
Get:16 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1952 kB]
Get:17 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [49.4 kB]
Get:18 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [33.9 kB]
Fetched 34.4 MB in 27s (1297 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.

```

```

root@a5202d3f0d2b:/# apt install -y curl vim
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
ca-certificates krb5-locales libbrotlii libcurl4t64 libexpat1 libgpm2 libgssapi-krb5-2 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0
libldap-common libldap2 libnghtp2-14 libpsl5t64 libpython3.12-minimal libpython3.12-stdlib libpython3.12t64 libreadline8t64 librtmp1 libsasl2-2
libsasl2-modules libsasl2-modules-db libssodium23 libsqlite3-0 libssh-4 media-types netbase openssl publicsuffix readline-common tzdata
vim-common vim-runtime xxd
Suggested packages:
gpm krb5-doc krb5-user libsasl2-modules-gssapi-mit | libsasl2-modules-gssapi-heimdal libsasl2-modules-ldap libsasl2-modules-otp
libsasl2-modules-sql readline-doc ctags vim-doc vim-scripts
The following NEW packages will be installed:
ca-certificates curl krb5-locales libbrotlii libcurl4t64 libexpat1 libgpm2 libgssapi-krb5-2 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0
libldap-common libldap2 libnghtp2-14 libpsl5t64 libpython3.12-minimal libpython3.12-stdlib libpython3.12t64 libreadline8t64 librtmp1 libsasl2-2
libsasl2-modules libsasl2-modules-db libssodium23 libsqlite3-0 libssh-4 media-types netbase openssl publicsuffix readline-common tzdata vim
vim-common vim-runtime xxd
0 upgraded, 37 newly installed, 0 to remove and 2 not upgraded.
Need to get 19.9 MB of archives.
After this operation, 81.3 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 openssl amd64 3.0.13-0ubuntu3.6 [1003 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble/main amd64 ca-certificates all 20240203 [159 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 krb5-locales all 1.20.1-6ubuntu2.6 [14.8 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libexpat1 amd64 2.6.1-2ubuntu0.3 [88.8 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libkrb5support0 amd64 1.20.1-6ubuntu2.6 [34.4 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libk5crypto3 amd64 1.20.1-6ubuntu2.6 [82.0 kB]

```

FIGURE 2 – Installation de curl et vim

### 2.3 3. Cration d'un fichier test.txt

Commande :

```
echo "Ceci est un fichier test" > test .txt
```

## 2.4 4. Sortie du conteneur sans l'arrêter

Commande : Ctrl+P  
puis Ctrl+Q

## 2.5 5. Copie du fichier test.txt vers la machine hôte

Commande sur la machine hôte :  
docker cp <container\_id>:/test.txt ./fichiers/test.txt

```
PS C:\Users\user\Desktop\docker-exercice2-tp1> docker cp ubuntu-exercice:/test.txt ./fichiers/test.txt
Successfully copied 2.05kB to C:\Users\user\Desktop\docker-exercice2-tp1\fichiers\test.txt
```

FIGURE 3 – Copie du fichier test.txt vers la machine hôte

## 2.6 6. Modification du fichier et recopie dans le conteneur

Modifications sur la machine hôte puis :  
docker cp ./test.txt <container\_id>:/test.txt

```
PS C:\Users\user\Desktop\docker-exercice2-tp1> echo "" >> ./fichiers/test.txt
PS C:\Users\user\Desktop\docker-exercice2-tp1> echo "==== MODIFICATION SUR MACHINE ===" >> ./fichiers/test.txt
PS C:\Users\user\Desktop\docker-exercice2-tp1> echo "Ajout de texte depuis l'hôte" >> ./fichiers/test.txt
PS C:\Users\user\Desktop\docker-exercice2-tp1> cat ./fichiers/test.txt
==== Fichier cree dans le conteneur ===
Date : Mon Oct 27 10:14:15 SAST 2025

==== MODIFICATION SUR MACHINE ===

Ajout de texte depuis l hôte

PS C:\Users\user\Desktop\docker-exercice2-tp1> docker cp ./fichiers/test.txt ubuntu-exercice:/test.txt
Successfully copied 2.05kB to ubuntu-exercice:/test.txt
```

FIGURE 4 – Modification du fichier et recopie dans le conteneur

## 2.7 7. Reconnexion au conteneur et vérification

```
PS C:\Users\user\Desktop\docker-exercice2-tp1> docker exec -it ubuntu-exercice bash
root@a5202d3f0d2b:/# cat test.txt
==== Fichier cree dans le conteneur ===
Date : Mon Oct 27 10:14:15 SAST 2025

==== MODIFICATION SUR MACHINE ===
Ajout de texte depuis l hôte
root@a5202d3f0d2b:/#
```

FIGURE 5 – Vérification des modifications dans le conteneur

## 2.8 8. Création d'une nouvelle image à partir du conteneur modifié

Commande :

```
docker commit <container_id> mon_ubuntu_personnalise
```

```
PS C:\Users\user\Desktop\docker-exercice2-tp1> docker commit ubuntu-exercice mon-ubuntu-custom:v1
sha256:01013a507eaed07e6ac44fe66ea0d0b765c6c6ae9dadce8789f4a6caa5fe928c
PS C:\Users\user\Desktop\docker-exercice2-tp1> docker images
REPOSITORY                                     TAG      IMAGE ID      CREATED       SIZE
mon-ubuntu-custom                            v1      01013a507ea   28 seconds ago  329MB
```

FIGURE 6– Vérification des modifications dans le conteneur

## 2.9 9. Lancement d'un nouveau conteneur basé sur l'image personnalisée

Commande :

```
docker run -it      mon_ubuntu_personnalise
```

```
PS C:\Users\user\Desktop\docker-exercice2-tp1> docker run -it --name test-image mon-ubuntu-custom:v1 bash
root@42d86dic373c:/# cat test.txt
== Fichier cree dans le conteneur ==
Date : Mon Oct 27 10:14:15 SAST 2025

== MODIFICATION SUR MACHINE ==
Ajout de texte depuis l hôte
root@42d86dic373c:/# curl --version
curl 8.5.0 (x86_64-pc-linux-gnu) libcurl/8.5.0 OpenSSL/3.0.13 zlib/1.3 brotli/1.1.0 zstd/1.5.5 libidn2/2.3.7 libpsl/0.21.2 (+libidn2/2.3.7) libssh/0
.10.6/openssl/zlib nghttp2/1.59.0 librtmp/2.3 OpenLDAP/2.6.7
Release-Date: 2023-12-06, security patched: 8.5.0-2ubuntu10.6
Protocols: dict file ftp ftps gopher gophers http https imap imaps ldap ldaps mqtt pop3 pop3s rtmp rtsp scp sftp smb smbs smtp smt
Features: alt-svc AsynchDNS brotli GSS-API HSTS HTTP2 HTTPS-proxy IDN IPv6 Kerberos Largefile libz NTLM PSL SPNEGO SSL threadsafe TLS-SRP UnixSocket
s zstd
root@42d86dic373c:/# vim --version
VIM - Vi IMproved 9.1 (2024 Jan 02, compiled Sep 05 2025 19:44:46)
Included patches: 1-16, 647, 678, 697
Modified by team+vim@tracker.debian.org
Compiled by team+vim@tracker.debian.org
```

FIGURE 7 – Nouveau conteneur basé sur l'image personnalisée

## 2. 10. 10. Vérification

### - des modifications dans le nouveau conteneur

Commande :

```
cat test .txt
```

```
PS C:\Users\user\Desktop\docker-exercice2-tp1> docker run -it mon-ubuntu-custom:v1 bash -c "cat test.txt"
== Fichier cree dans le conteneur ==
Date : Mon Oct 27 10:14:15 SAST 2025
== MODIFICATION SUR MACHINE ==
```

FIGURE 8 – Vérification du fichier dans le nouveau conteneur

## 2.11 Bonus : Exploration des statistiques en temps réel

Commande :

```
docker stats
```

```
PS C:\Users\user\Desktop\docker-exercice2-tp1> docker stats ubuntu-exercice --no-stream
CONTAINER ID  NAME          CPU %     MEM USAGE / LIMIT      MEM %     NET I/O            BLOCK I/O      PIDS
5202d3f0d2b  ubuntu-exercice  0.00%    80.45MiB / 15.46GiB  0.51%    56.9MB / 833kB  0B / 203MB  3
```

FIGURE 9 – Statistiques des conteneurs en temps réel

## 3 Conclusion

Cet exercice a permis de se familiariser avec les fonctionnalités avancées de Docker :

- Gestion interactive des conteneurs
- Persistance des fichiers via docker cp
- Création d’images personnalisées
- Lancement de conteneurs basés sur des images modifiées
- Surveillance des conteneurs en temps réel avec docker stats

Ces compétences sont essentielles pour l’administration et le déploiement d’applications conteneurisées.