

## TP 1. – INSTALLATION DE DOCKER ET PREMIER PAS

Pour préparer la machine à installer docker, on va d'abord faire ces commandes :

```
apt update
```

Puis :

```
apt install ca-certificates curl gnupg
```

On va ensuite installer les certificats :

```
curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --dearmor -o  
/etc/apt/keyrings/docker.gpg
```

Puis on fait :

```
chmod a+r /etc/apt/keyrings/docker.gpg
```

Cette commande sert à donner les permissions au fichier docker.gpg

On va maintenant intégrer le dépôt docker dans le fichier source.list :

```
echo \
```

```
"deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg]
```

```
https://download.docker.com/linux/debian \
```

```
"$(. /etc/os-release && echo "$VERSION_CODENAME)" stable" | \
```

```
tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Malheureusement , il nous dit que il E: Entrée 1 mal formée dans list fichier  
/etc/apt/sources.list.d/docker.list (URI) E: La liste des sources ne peut être lue.

Le contenu du fichier /etc/apt/sources.list.d/docker.list doit être : deb [arch=amd64  
signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu jammy  
stable

On installe docker avec : apt install docker-ce docker-ce-cli containerd.io  
docker-buildx-plugin docker-compose-plugin

Ensuite pour voir si docker est bien actif, on met systemctl status docker

On ajoute l'utilisateur a docker

```
root@modele:/home/test# gpasswd -a test docker  
Ajout de l'utilisateur test au groupe docker
```

La on vérifie la version de docker

```
root@modele:/home/test# docker version
Client: Docker Engine - Community
Version:      28.0.1
API version:  1.48
Go version:   go1.23.6
Git commit:   068a01e
Built:        Wed Feb 26 10:41:08 2025
OS/Arch:      linux/amd64
Context:      default

Server: Docker Engine - Community
Engine:
Version:      28.0.1
API version:  1.48 (minimum version 1.24)
Go version:   go1.23.6
Git commit:   bbd0a17
Built:        Wed Feb 26 10:41:08 2025
OS/Arch:      linux/amd64
Experimental: false
containerd:
Version:      1.7.25
GitCommit:    bcc810d6b9066471b0b6fa75f557a15a1cbf31bb
runc:
Version:      1.2.4
GitCommit:    v1.2.4-0-g6c52b3f
docker-init:
Version:      0.19.0
GitCommit:    de40ad0
root@modele:/home/test# █
```

On va lancer un conteneur test

```

root@modele:/home/test# docker run --rm hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete
Digest: sha256:bfb0cc14f13f9ed1ae86abc2b9f11181dc50d779807ed3a3c5e55a6936dbdd5
Status: Downloaded newer image for hello-world:latest

```

Hello from Docker!  
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:  
\$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:  
<https://hub.docker.com/>

For more examples and ideas, visit:  
<https://docs.docker.com/get-started/>

On va chercher le mot ubuntu

```

root@modele:/home/test# docker search ubuntu
NAME                DESCRIPTION                STARS     OFFICIAL
ubuntu              Ubuntu is a Debian-based Linux operating sys... 17496     [OK]
ubuntu/squid        Squid is a caching proxy for the Web. Long-t... 107
ubuntu/nginx        Nginx, a high-performance reverse proxy & we... 127
ubuntu/cortex       Cortex provides storage for Prometheus. Long... 4
ubuntu/apache2      Apache, a secure & extensible open-source HT... 90
ubuntu/kafka        Apache Kafka, a distributed event streaming ... 53
ubuntu/bind9        BIND 9 is a very flexible, full-featured DNS... 102
ubuntu/prometheus   Prometheus is a systems and service monitori... 70
ubuntu/zookeeper    ZooKeeper maintains configuration informatio... 13
ubuntu/mysql        MySQL open source fast, stable, multi-thread... 67
ubuntu/postgres     PostgreSQL is an open source object-relatio... 41
ubuntu/redis        Redis, an open source key-value store. Long-... 23
ubuntu/jre          Distroless Java runtime based on Ubuntu. Lon... 19
ubuntu/dotnet-aspnet Chiselled Ubuntu runtime image for ASP.NET a... 25
ubuntu/grafana       Grafana, a feature rich metrics dashboard & ... 12
ubuntu/dotnet-deps   Chiselled Ubuntu for self-contained .NET & A... 16
ubuntu/cassandra    Cassandra, an open source NoSQL distributed ... 2
ubuntu/memcached     Memcached, in-memory keyvalue store for smal... 5
ubuntu/python       A chiselled Ubuntu rock with the Python runt... 20
ubuntu/dotnet-runtime Chiselled Ubuntu runtime image for .NET apps... 20
ubuntu/prometheus-alertmanager Alertmanager handles client alerts from Prom... 9
ubuntu/mlflow        MLFlow: for managing the machine learning li... 5
ubuntu/telegraf      Telegraf collects, processes, aggregates & w... 4
ubuntu/loki          Grafana Loki, a log aggregation system like ... 2
ubuntu/chiselled-jre [MOVED TO ubuntu/jre] Chiselled JRE: distrol... 3

```

```

root@modele:/home/test# docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
5a7813e071bf: Pull complete
Digest: sha256:72297848456d5d37d1262630108ab308d3e9ec7ed1c3286a32fe09856619a782
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

```

Docker télécharge les différentes couches qui forment l'image.

```
root@modele:/home/test# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	a04dc4851cbc	6 weeks ago	78.1MB
hello-world	latest	74cc54e27dc4	7 weeks ago	10.1kB

On va récupérer une autre version de l'image

```
root@modele:/home/test# docker pull ubuntu:lunar
lunar: Pulling from library/ubuntu
6360b3717211: Pull complete
Digest: sha256:5a828e28de105c3d7821c4442f0f5d1c52dc16acf4999d5f31a3bc0f03f06edd
Status: Downloaded newer image for ubuntu:lunar
docker.io/library/ubuntu:lunar
```

```
root@modele:/home/test# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	a04dc4851cbc	6 weeks ago	78.1MB
hello-world	latest	74cc54e27dc4	7 weeks ago	10.1kB
ubuntu	lunar	f4cdeba72b99	15 months ago	70.3MB

docker rmi hello-world

Ensuite, on fait : docker run ubuntu

Ici, on liste les conteneurs actifs

```
root@modele:/home/test# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

Ensuite, on va lister les docker actif

```
root@modele:/home/test# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
f329ae601ca2	ubuntu	"/bin/bash"	14 seconds ago	Exited (0) 14 seconds ago		nifty_sutherland

docker run ubuntu

docker ps -a

```
root@modele:/home/test# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
6144af5359bf	ubuntu	"/bin/bash"	6 seconds ago	Exited (0) 5 seconds ago		angry_clarke
f329ae601ca2	ubuntu	"/bin/bash"	7 minutes ago	Exited (0) 7 minutes ago		nifty_sutherland

docker run ubuntu cat /etc/lsb-release

```
root@modele:/home/test# docker run ubuntu cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=24.04
DISTRIB_CODENAME=noble
DISTRIB_DESCRIPTION="Ubuntu 24.04.1 LTS"
```

docker ps -a

```
root@modele:/home/test# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e55fc581fc0e	ubuntu	"cat /etc/lsb-release"	3 minutes ago	Exited (0) 3 minutes ago		gracious_hawking
6144af5359bf	ubuntu	"/bin/bash"	4 minutes ago	Exited (0) 4 minutes ago		angry_clarke
f329ae601ca2	ubuntu	"/bin/bash"	11 minutes ago	Exited (0) 11 minutes ago		nifty_sutherland

docker rm <id conteneur> ou docker rm <nom du conteneur>

```

root@modele:/home/test# docker rm 6144af5359bf
6144af5359bf
root@modele:/home/test# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS      PORTS      NAMES
e55fc581fc0e   ubuntu   "cat /etc/lsb-release"   6 minutes ago   Exited (0) 6 minutes ago           gracious_hawking
f329ae601ca2   ubuntu   _/bin/bash              14 minutes ago   Exited (0) 14 minutes ago           nifty_sutherland

```

`docker run --rm ubuntu cat /etc/lsb-release`

Cette commande ci dessus permet de programmer la suppression

`docker ps -a`

Cette commande permet de voir qu'il n'y a pas de nouveau conteneur

```

test@modele:~$ sudo bash
[sudo] Mot de passe de test :
root@modele:/home/test# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS      PORTS      NAMES
e55fc581fc0e   ubuntu   "cat /etc/lsb-release"   13 days ago   Exited (0) 13 da
ys ago          gracious_hawking
f329ae601ca2   ubuntu   _/bin/bash              13 days ago   Exited (0) 13 da
ys ago          nifty_sutherland
root@modele:/home/test#

```

Pour relancer un conteneur, on va utiliser cette commande :

```
docker start <id conteneur> ou docker start <nom du conteneur>
```

Pour lancer le conteneur au 1er plan, on va utiliser la commande :

```

root@modele:/home/test# docker start -a gracious_hawking
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=24.04
DISTRIB_CODENAME=noble
DISTRIB_DESCRIPTION="Ubuntu 24.04.1 LTS"

```

On va supprimer le conteneur non actif :

```

root@modele:/home/test# docker rm nifty_sutherland
nifty_sutherland

```

On lance un conteneur en interactif :

```
docker run --name servUbuntu -it ubuntu
```

On sera connecté au conteneur donc on va faire :

```

apt update
apt dist-upgrade
apt install openssh-server

```

Ensuite, on va sortir du conteneur en faisant : `exit`

On va réactiver le conteneur avec cette commande :

```

root@modele:/home/test# docker start servUbuntu
servUbuntu

```

Et pour accéder au conteneur on va faire :

```
docker attach servUbuntu
```

Pour voir les processus lancés dans un conteneur on fait cette commande :

docker top <id conteneur> ou docker top <nom du conteneur>

En l'occurrence on va faire : docker top servUbuntu

Ne pas oublier de lancer le conteneur :

```
root@modele:/home/test# docker start servUbuntu
servUbuntu
root@modele:/home/test# docker top servUbuntu
UID                PID                PPID                C
STIME              TTY                TIME               CMD
root               27122             27103              0
08:50              pts/0              00:00:00           /bin/bash
```

On va activer le service ssh :

docker attach servUbuntu

```
root@b613ba3f3d1c:/# service ssh start
```

```
* Starting OpenBSD Secure Shell server sshd
```

[ OK ]

Dans un autre terminal :

```
test@modele:~$ docker top servUbuntu
UID                PID                PPID                C                STIME              TTY                TIME               CMD
root               27122             27103              0                08:50              pts/0              00:00:00           /bin/bash
root               28437             27122              0                08:53              ?                  00:00:00           sshd: /usr/sbin/sshd [listener] 0 of 10-100 startups
```

On peut créer une nouvelle image à partir d'un conteneur à l'aide de cette commande :

docker commit <id conteneur> <nom image> ou docker commit <nom conteneur> <nom image>

On va faire cette commande qui va générer un hash :

```
root@modele:/home/test# docker commit servUbuntu aporaf/ubuntu:ssh
sha256:ae7ab5f059ffccd994a3ae221773ff2203cae84ae167fecfe341e3fc15b1f0d4
```

À l'aide de la commande docker images on peut voir qu'il y a une nouvelle image

```
root@modele:/home/test# docker images
REPOSITORY          TAG                IMAGE ID            CREATED             SIZE
aporaf/ubuntu        ssh                ae7ab5f059ff       About a minute ago  300MB
ubuntu               latest            a04dc4851cbc       8 weeks ago        78.1MB
hello-world          latest            74cc54e27dc4       2 months ago       10.1kB
ubuntu               lunar             f4cdeba72b99       16 months ago      70.3MB
```

Pour sauvegarder une image en local :

docker save <nom image> > <nom\_fichier.tar>

Et pour restaurer une image :

docker load -i <nom\_fichier.tar>

Docker dispose d'une interface réseau sur la machine hôte :

```
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
    N group default
    link/ether ca:9c:61:75:38:63 brd ff:ff:ff:ff:ff:ff
    inet 172.18.0.1/16 brd 172.18.255.255 scope global docker0
        valid_lft forever preferred_lft forever
    inet6 fe80::c89c:61ff:fe75:3863/64 scope link
        valid_lft forever preferred_lft forever
```

docker run -d -p <IP:port-hôte:port-conteneur> --name <nom conteneur> <nom image>

COMMANDE

```
root@modele:/home/test# docker run -d -p 172.17.123.58:999:22 --name servssh aporaf/ubuntu:ssh /usr/sbin/sshd -D
8b8ebca6273851038108272e048aa035325b2e4eb0097feb56151b0f417abd10
```

```
root@modele:/home/test# ssh user@172.17.123.58 -p 999
The authenticity of host '[172.17.123.58]:999 ([172.17.123.58]:999)' can't be established.
ED25519 key fingerprint is SHA256:eY3F8UYxtywj5SSStKDCG/WPrkYKrZTyQeUPeLJcu0zM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[172.17.123.58]:999' (ED25519) to the list of known hosts.
```

On fait docker ps :

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8b8ebca62738	aporaf/ubuntu:ssh	"/usr/sbin/sshd -D"	2 minutes ago	Up 2 minutes	172.17.123.58:999->22/tcp	servssh

Et pour afficher les logs du conteneur : docker logs servssh  
Fin DU TP