

Spécialité : Réseaux Intelligents et Cybersécurité

Présenté par : COULIBALY DAOUDA DAVID & WAHID Moncef

Conteneurisation avec Docker

Encadré par:

Pr. IDRISSI Hind

ENSA Khouribga

Année universitaire :2024/2025

I. INSTALLATION

- Installez Docker depuis les dépôts officiels :

```
try: sudo apt install <deb name>
david@david-virtual-machine:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Some packages could not be installed. This may mean that you have
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or have been moved out of Incoming.
```

- Vérifiez que Docker est en cours d'exécution :

```
david@david-virtual-machine:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Fri 2025-04-25 20:41:24 +01; 7min ago
   TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
   Main PID: 1533 (dockerd)
   Tasks: 9
   Memory: 95.3M
   CPU: 1.830s
   CGroup: /system.slice/docker.service
           └─1533 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont>

20:41:17 25 بريلي | david-virtual-machine dockerd[1533]: time="2025-04-25T20:41:1>
20:41:17 25 بريلي | david-virtual-machine dockerd[1533]: time="2025-04-25T20:41:1>
20:41:23 25 بريلي | david-virtual-machine dockerd[1533]: time="2025-04-25T20:41:2>
20:41:24 25 بريلي | david-virtual-machine systemd[1]: Started Docker Application >
lines 1-22/22 (END)
```

II. INSTALLATION (à partir du référentiel de Docker)

1. Installez quelques paquets prérequis qui permettent à « apt » d'utiliser les paquets via HTTPS :

```

Command not found
david@david-virtual-machine:~$ sudo apt install apt-transport-https ca-certific
ates
curl software-properties-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203~22.04.1).
Ubuntu 22.04.5 LTS amd64 already the newest version (2.4.13).
0 upgraded, 0 newly installed, 0 to remove and 101 not upgraded.

curl: (6) Could not resolve host: software-properties-common
david@david-virtual-machine:~$
david@david-virtual-machine:~$ S

```

2. Ajoutez ensuite la clé GPG du référentiel Docker officiel à votre système :

```

david@david-virtual-machine:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg
| sudo apt-key add -
curl: (6) Could not resolve host: xn--fssl-z86a
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBFit2ioBEADhWpZ8/wvZ6hUTiX0wQHxMAlaFHcPH9hAtr4F1y2+0YdbtMuth
lqqwp028AqyY+PRfVMtSYMbjuQuu5byyKR01BbqYhus3jttqQmLjZ/bJvXqnmVXh
38UuLa+z077PxyxQhu5BbqntTPQMfiyqEiU+BKbq2WmANUKQf+1AmZY/IruOXbnq
L4C1+gJ8vfmXQt99npCaxEjaNRVYF0S8QcixNzHUYNb6emjLANyEVLZzeqo7XKL7
UrvW5inawTSzWNvtjEjj4nJL8NsLwscpLPQUhtQ+7BbQXAwAmeHCUTQIvVwXqW0N
cmhh4HgeQscQHYg0JjjDVfoY5MucvglbIgcqfzAHW9jxmRL4qbMZj+b1XoePEtht
ku4bIQN1X5P07fNWzLgaRL5Z4POXDDZTLIQ/EL58j9kp4bnWRCJW0lya+f8ocodo
vZZ+Doi+fy4D5ZGrL4XEcIQP/Lv5uFyf+kQtl/94VFYVJ0leAv8W92KgdKhTcTD
G7c0tIkVEKNUq48b3aQ64NOZQW7fVjfoKwEZD0qPE72Pa45jrZzvUFxSpdiNk2tZ
XYukHjLxxEgBdC/J3cMMNRE1F4NCA3ApfV1Y7/hTeOnmDuDYwr9/obA8t016Yljj
q5rdkywPf4JF8mXUW5eCN1vAFHXeg9ZwemhBtQmGxXnw9M+z6hWwc6ahmWARAQAB
tCtEb2NrZXIguUmsZWFzZSAoQ0UgZGVhKSA8ZG9ja2Vya2Vya2Vya2Vya2Vya2Vya2
BBMBCgAhBQJYrefAAhsvBQsJCAcDBRUKCQgLBRYCAwEAAh4BAheAAAJEI2BgDwO
v82IsskP/iQzo68fLDQmNvn8X5XTd6RRaUH33kXYXquT6NkHJciS7E2gTJmqvMqd
tI4mNYHCSEYXI5qrcV5YqX9P6+Ko+vozo4nseUQLPH/ATQ4qL0Zok+1jkag3Lgk
jonyUf9bwtWxFp05HC3GMHPhhcUSexCxQLQvnFXD2sWLKivHp2ft8QbRGeZ+d3m
6fQcd5Fu7pxsqm0EUDK5NL+nPIgYhN+auTrhgzhK1CSHFgCCM/wfRleI9Utz6p9P
XRKILWnXtT4qNGZNTN0tr+NLG/6Bqd80YBaFAUcue/w1VW6JQ2VGYZHnZu9S8LMc
FYBa5I9PpxwGQ0gq6RDkDbV+PqTQT5EFMeR1mrjckk4DQJjbxemZbiNMG5kGECAB
g383P3eLhn03WGbEEa4Mnc3Z4+7c236QI3xWJfNPDubXRaAwhy/6rTSFbzwKB0Jm
ebwz0fw10Y6f55MiT/RqDcYUuPi3r3ivVRkK86p0KBAJwFHvqI9KaKXMZifVnowlh

```

```

david@david-virtual-machine:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
Warning: apt-key is deprecated
0Warning: apt-key is deprecated
0curl: (6) Could not resolve host: xn--fssl-z86a
OK

```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
			Dload	Upload	Total	Spent	Left
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0:00:24	0
100	3817	100	3817	0	0	20276	20195

3. Ajoutez le référentiel Docker aux sources APT :

```
bionic stable"
david@david-virtual-machine:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable'
Description:
Archive for codename: bionic components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
```

4. Mettez à jour la base de données de paquets

```
david@david-virtual-machine:~$ sudo apt-get update
Hit:1 http://ma.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:3 http://ma.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://ma.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:5 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:6 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:7 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:8 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:9 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy InRelease
Hit:10 https://ppa.launchpadcontent.net/gns3/ppa/ubuntu jammy InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/jammy/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
W: https://download.docker.com/linux/ubuntu/dists/bionic/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
david@david-virtual-machine:~$
```

5. Pour s'assurer que vous êtes sur le point d'installer à partir du référentiel Docker au lieu du référentiel par défaut Ubuntu, tapez :

```
david@david-virtual-machine:~$ sudo apt-cache policy docker-ce
docker-ce:
  Installed: 5:28.1.1-1~ubuntu.22.04~jammy
  Candidate: 5:28.1.1-1~ubuntu.22.04~jammy
  Version table:
 *** 5:28.1.1-1~ubuntu.22.04~jammy 500
    500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
    100 /var/lib/dpkg/status
 5:28.1.0-1~ubuntu.22.04~jammy 500
    500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 5:28.0.4-1~ubuntu.22.04~jammy 500
    500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 5:28.0.3-1~ubuntu.22.04~jammy 500
    500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 5:28.0.2-1~ubuntu.22.04~jammy 500
    500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 5:28.0.1-1~ubuntu.22.04~jammy 500
    500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 5:28.0.0-1~ubuntu.22.04~jammy 500
    500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 5:27.5.1-1~ubuntu.22.04~jammy 500
    500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 5:27.5.0-1~ubuntu.22.04~jammy 500
    500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 5:27.4.1-1~ubuntu.22.04~jammy 500
    500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 5:27.4.0-1~ubuntu.22.04~jammy 500
    500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
```

6. Maintenant, installez alors Docker (Community Edition) :

```
500 https://download.docker.com/linux/ubuntu/bionic/stable amd64 Pa
david@david-virtual-machine:~$ sudo apt install docker-ce
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker-ce is already the newest version (5:28.1.1-1~ubuntu.22.04~jammy).
0 upgraded, 0 newly installed, 0 to remove and 101 not upgraded.
david@david-virtual-machine:~$
```

7. Vérifiez que le Docker est en cours d'exécution.

```
david@david-virtual-machine:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2025-04-25 20:41:24 +01; 20min ago
 TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
   Main PID: 1533 (dockerd)
     Tasks: 9
    Memory: 95.3M
       CPU: 1.943s
   CGroup: /system.slice/docker.service
           └─1533 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

20:41:17 25 بريل | david-virtual-machine dockerd[1533]: time="2025-04-25T20:41:17.081621187+01:00">
20:41:17 25 بريل | david-virtual-machine dockerd[1533]: time="2025-04-25T20:41:17.099326024+01:00">
20:41:23 25 بريل | david-virtual-machine dockerd[1533]: time="2025-04-25T20:41:23.007587645+01:00">
20:41:23 25 بريل | david-virtual-machine dockerd[1533]: time="2025-04-25T20:41:23.721835649+01:00">
20:41:23 25 بريل | david-virtual-machine dockerd[1533]: time="2025-04-25T20:41:23.938884289+01:00">
20:41:23 25 بريل | david-virtual-machine dockerd[1533]: time="2025-04-25T20:41:23.939621170+01:00">
20:41:24 25 بريل | david-virtual-machine dockerd[1533]: time="2025-04-25T20:41:24.061120856+01:00">
20:41:24 25 بريل | david-virtual-machine dockerd[1533]: time="2025-04-25T20:41:24.100072760+01:00">
20:41:24 25 بريل | david-virtual-machine dockerd[1533]: time="2025-04-25T20:41:24.100150355+01:00">
20:41:24 25 بريل | david-virtual-machine systemd[1]: Started Docker Application Container Engine.
lines 1-22/22 (END)
```

8. Ajoutez votre nom d'utilisateur au groupe docker :

```
david@david-virtual-machine:~$ sudo usermod -aG docker $ david
Usage: usermod [options] LOGIN

Options:
  -b, --badnames          allow bad names
  -c, --comment COMMENT   new value of the GECOS field
  -d, --home HOME_DIR     new home directory for the user account
  -e, --expiredate EXPIRE_DATE
                          set account expiration date to EXPIRE_DATE
  -f, --inactive INACTIVE
                          set password inactive after expiration
                          to INACTIVE
  -g, --gid GROUP          force use GROUP as new primary group
  -G, --groups GROUPS     new list of supplementary GROUPS
  -a, --append             append the user to the supplemental GROUPS
                          mentioned by the -G option without removing
                          the user from other groups
  -h, --help              display this help message and exit
  -l, --login NEW_LOGIN   new value of the login name
  -L, --lock               lock the user account
  -m, --move-home         move contents of the home directory to the
                          new location (use only with -d)
  -o, --non-unique        allow using duplicate (non-unique) UID
  -p, --password PASSWORD use encrypted password for the new password
  -R, --root CHROOT_DIR   directory to chroot into
  -P, --prefix PREFIX_DIR prefix directory where are located the /etc/* files
  -s, --shell SHELL       new login shell for the user account
  -u, --uid UID            new UID for the user account
  -U, --unlock            unlock the user account
  -v, --add-subuids FIRST-LAST
                          add range of subordinate uids
  -V, --del-subuids FIRST-LAST
                          remove range of subordinate uids
  -w, --add-subgids FIRST-LAST
                          add range of subordinate gids
  -W, --del-subgids FIRST-LAST
                          remove range of subordinate gids
  -Z, --selinux-user SEUSER
                          new SELinux user mapping for the user account
```

9. Pour appliquer la nouvelle appartenance à un groupe et éviter la déconnexion et la reconnexion au serveur :

```
david@david-virtual-machine:~$ su $ david
su: user $ does not exist or the user entry does not contain all the required fields
david@david-virtual-machine:~$ su david
Password:
david@david-virtual-machine:~$ █
```

10. Confirmez que votre utilisateur est maintenant ajouté au groupe docker en tapant :

```
root@david-virtual-machine:/home/david# id -nG
root
root@david-virtual-machine:/home/david# █
```

III. Utilisation de la Commande DOCKER

Utiliser docker consiste à lui transmettre une chaîne d'options et de commandes suivie d'arguments.

La syntaxe prend cette forme : **docker [OPTIONS] COMMAND**

- Pour afficher toutes les sous-commandes disponibles de docker :

```
root@david-virtual-machine:/home/david# docker
Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Common Commands:
run          Create and run a new container from an image
exec        Execute a command in a running container
ps          List containers
build       Build an image from a Dockerfile
bake        Build from a file
pull        Download an image from a registry
push        Upload an image to a registry
images      List images
login       Authenticate to a registry
logout      Log out from a registry
search      Search Docker Hub for images
version     Show the Docker version information
info        Display system-wide information

Management Commands:
builder     Manage builds
buildx*     Docker Buildx
compose*    Docker Compose
container   Manage containers
context     Manage contexts
image       Manage images
```

- Pour afficher les options disponibles pour une commande spécifique, tapez :

```

root@david-virtual-machine:/home/david# docker command --help
Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Common Commands:
  run       Create and run a new container from an image
  exec     Execute a command in a running container
  ps       List containers
  build    Build an image from a Dockerfile
  bake     Build from a file
  pull    Download an image from a registry
  push    Upload an image to a registry
  images  List images
  login    Authenticate to a registry
  logout  Log out from a registry
  search  Search Docker Hub for images
  version  Show the Docker version information
  info    Display system-wide information

Management Commands:
  builder  Manage builds
  buildx*  Docker Buildx
  compose* Docker Compose
  container Manage containers
  context  Manage contexts
  image    Manage images
  manifest Manage Docker image manifests and manifest lists
  network  Manage networks
  plugin   Manage plugins
  system   Manage Docker
  trust    Manage trust on Docker images
  volume   Manage volumes

Swarm Commands:
  swarm    Manage Swarm

```

- Pour afficher des informations sur Docker, tapez :

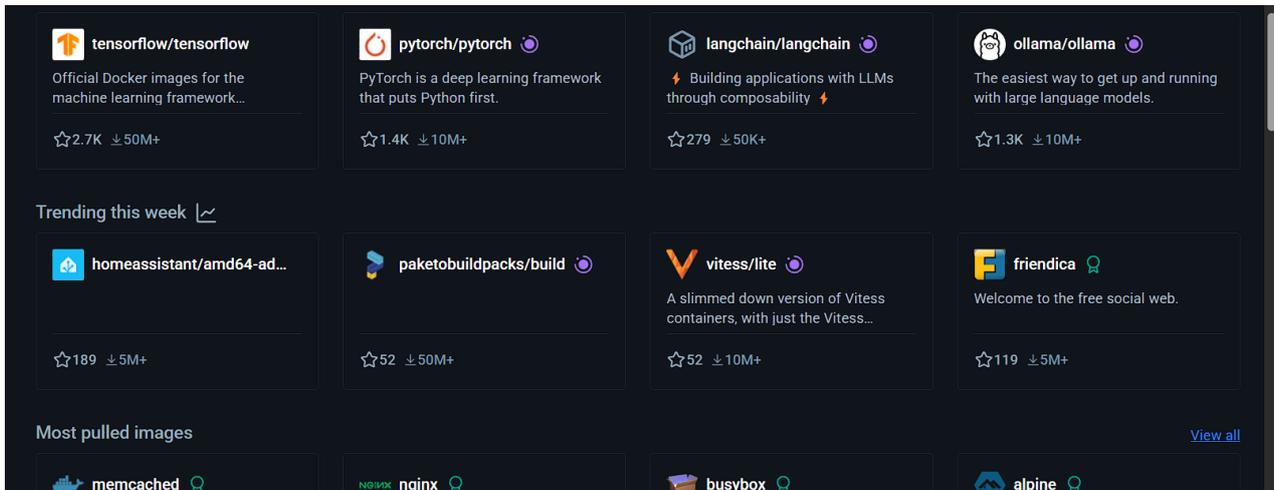
```

root@david-virtual-machine:/home/david# docker info
Client: Docker Engine - Community
Version: 28.1.1
Context: default
Debug Mode: false
Plugins:
  buildx: Docker Buildx (Docker Inc.)
    Version: v0.23.0
    Path: /usr/libexec/docker/cli-plugins/docker-buildx
  compose: Docker Compose (Docker Inc.)
    Version: v2.35.1
    Path: /usr/libexec/docker/cli-plugins/docker-compose

Server:
Containers: 1
  Running: 0
  Paused: 0
  Stopped: 1
Images: 1
Server Version: 28.1.1
Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Using metacopy: false
  Native Overlay Diff: true
  userxattr: false
Logging Driver: json-file
Cgroup Driver: systemd
Cgroup Version: 2
Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local splunk syslog
Swarm: inactive
Runtimes: io.containerd.runc.v2 runc
Default Runtime: runc
Init Binary: docker-init
containerd version: 05044ec0a9a75232cad458027ca83437aae3f4fda
runc version: v1.2.5-0-g59923ef
init version: de40ad0
Security Options:
  apparmor
  seccomp
  Profile: builtin

```

IV. Utilisation de la Commande DOCKER



- Pour accéder et télécharger l'image du conteneur « hello-world » à partir de Docker Hub, tapez :

```
root@david-virtual-machine:/home/david# docker run hello-world

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/

root@david-virtual-machine:/home/david#
```

- Pour rechercher des images disponibles sur Docker Hub.

```

root@david-virtual-machine:/home/david# docker search ubuntu
NAME                DESCRIPTION                STARS
AL
ubuntu              Ubuntu is a Debian-based Linux operating sys... 17551
ubuntu/squid        Squid is a caching proxy for the Web. Long-t... 111
ubuntu/nginx        Nginx, a high-performance reverse proxy & we... 129
ubuntu/cortex       Cortex provides storage for Prometheus. Long... 4
ubuntu/kafka        Apache Kafka, a distributed event streaming ... 53
ubuntu/prometheus   Prometheus is a systems and service monitori... 71
ubuntu/bind9        BIND 9 is a very flexible, full-featured DNS... 104
ubuntu/apache2      Apache, a secure & extensible open-source HT... 90
ubuntu/zookeeper    ZooKeeper maintains configuration informatio... 13
ubuntu/redis        Redis, an open source key-value store. Long-... 23
ubuntu/jre           Distrosless Java runtime based on Ubuntu. Lon... 20
ubuntu/mysql        MySQL open source fast, stable, multi-thread... 67
ubuntu/dotnet-aspnet Chiselled Ubuntu runtime image for ASP.NET a... 26
ubuntu/postgres     PostgreSQL is an open source object-relatio... 41
ubuntu/python       A chiselled Ubuntu rock with the Python runt... 22
ubuntu/memcached    Memcached, in-memory keyvalue store for sma... 5
ubuntu/cassandra    Cassandra, an open source NoSQL distributed ... 2
ubuntu/dotnet-runtime Chiselled Ubuntu runtime image for .NET apps... 20
ubuntu/prometheus-alertmanager Alertmanager handles client alerts from Prom... 10
ubuntu/grafana      Grafana, a feature rich metrics dashboard & ... 12
ubuntu/dotnet-deps  Chiselled Ubuntu for self-contained .NET & A... 16
ubuntu/telegraf     Telegraf collects, processes, aggregates & w... 4
ubuntu/mlflow       MLFlow: for managing the machine learning li... 5
ubuntu/loki         Grafana Loki, a log aggregation system like ... 2
ubuntu/chiselled-jre [MOVED TO ubuntu/jre] Chiselled JRE: distro... 3

```

Décrivez le résultat obtenu ?? : tous les images qui rassemblent à ubuntu , (les services : sql, apache, nginx...)

- La commande « **pull** » permet de télécharger sur votre ordinateur l'image que vous souhaitez utiliser.

```

ubuntu/chiselled-jre [MOVED TO ubuntu/jre] Chiselled JRE: distro...
root@david-virtual-machine:/home/david# docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
2726e237d1a3: Downloading  5.561MB/29.72MB

```

```

root@david-virtual-machine:/home/david# docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
2726e237d1a3: Pull complete
Digest: sha256:1e622c5f073b4f6bfad6632f2616c7f59ef256e96fe78bf6a595d
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
root@david-virtual-machine:/home/david#

```

- Pour voir les images téléchargées sur votre ordinateur.

```
~$ docker images
```

```

root@david-virtual-machine:/home/david# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        latest   602eb6fb314b   2 weeks ago   78.1MB
hello-world   latest   74cc54e27dc4   3 months ago  10.1kB
root@david-virtual-machine:/home/david#

```

V. Exécuter un Conteneur DOCKER

Pour exécuter un conteneur, on utilise la commande « run » :

⇒ La combinaison des options `-i` et `-t` vous donne un accès interactif au Shell dans le conteneur :

```
root@david-virtual-machine:/home/david# docker run -it --name ubuntu-nginx ubu
root@851dc8b94e3d:/#
```

- Mettre à jour la base de données de paquets à l'intérieur du conteneur.

```
root@851dc8b94e3d:/# apt update
Get:1 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [21.9 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1077 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [1178 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [985 kB]
64% [8 Packages 12.5 MB/19.3 MB 65%]
```

- Installez le serveur web nginx :

```
root@851dc8b94e3d:/# apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  iproute2 krb5-locales libatm1t64 libbpf1 libcap2-bin libelf1t64 libgssapi-krb5-2 libk5crypto
  libkeyutils1 libkrb5-3 libkrb5support0 libmnl0 libpam-cap libtirpc-common libtirpc3t64
  libxtables12 nginx-common
Suggested packages:
  iproute2-doc python3:any krb5-doc krb5-user fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  iproute2 krb5-locales libatm1t64 libbpf1 libcap2-bin libelf1t64 libgssapi-krb5-2 libk5crypto
  libkeyutils1 libkrb5-3 libkrb5support0 libmnl0 libpam-cap libtirpc-common libtirpc3t64
  libxtables12 nginx nginx-common
0 upgraded, 18 newly installed, 0 to remove and 1 not upgraded.
Need to get 2733 kB of archives.
After this operation, 8075 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

⇒ Notez l'adresse `HY` du conteneur.

```

root@851dc8b94e3d:/# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0@if7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 2e:a4:ca:2b:66:32 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
        valid_lft forever preferred_lft forever
root@851dc8b94e3d:/#

```

⇒ Démarrez nginx.

```

root@851dc8b94e3d:/# service nginx start
* Starting nginx nginx
root@851dc8b94e3d:/# ss -anltp
State  Recv-Q  Send-Q  Local Address:Port  Peer Address:Port  Process
LISTEN  0        511          0.0.0.0:80      0.0.0.0:*          users:(("nginx
LISTEN  0        511          [::]:80        [::]:*            users:(("nginx
root@851dc8b94e3d:/#

```

Définir ss -anltp ??

This command lists all listening TCP sockets, without resolving port names, and shows which process (PID/program) is using them

VI. Publier votre Première Image DOCKER sur Docker Hub

- Démarrer et exécuter le Shell Bash à nouveau du conteneur « ubuntu-nginx »

```

root@david-virtual-machine:/home/david# docker start ubuntu-nginx
ubuntu-nginx
root@david-virtual-machine:/home/david#

```

```

root@david-virtual-machine:/home/david# docker exec -it ubuntu-nginx /bin/bash
root@851dc8b94e3d:/#

```

- Quittez le conteneur « exit ».
- Les modifications apportées au fichier de conteneur seront **validées** (commande `commit`) dans une nouvelle image, qu'on va par la suite déposer dans le docker hub. Utilisez votre Docker ID :

```
Username: ^C
root@david-virtual-machine:/home/david# docker commit ubuntu-nginx dave534/ubuntu-nginx:1.0
sha256:c947ced87ce41bc7445a9bc08c9ca36936f196ff389e95dc24331ed4dab0350e
root@david-virtual-machine:/home/david#
```

- **Login nécessaire pour accéder votre compte docker :**

```
root@david-virtual-machine:/home/david# docker login
Failed to start web-based login - falling back to command line login...

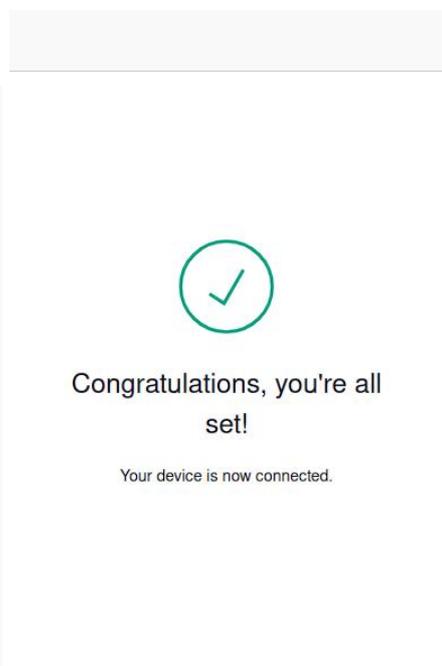
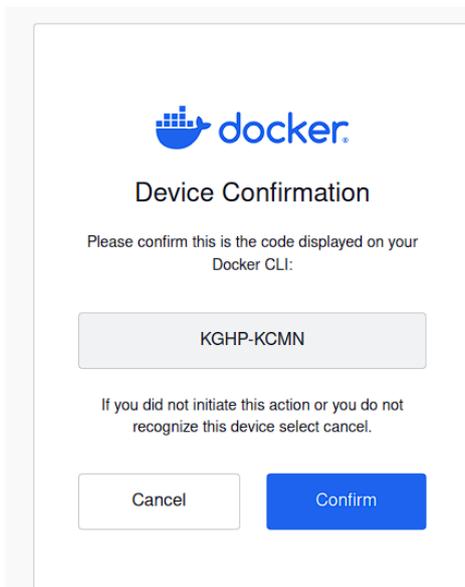
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.

You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/

Username: dave534

Info → A Personal Access Token (PAT) can be used instead.
To create a PAT, visit https://app.docker.com/settings

Password: █
```



```
root@david-virtual-machine:/home/david# docker login -u dave534

Info → A Personal Access Token (PAT) can be used instead.
      To create a PAT, visit https://app.docker.com/settings

Password:

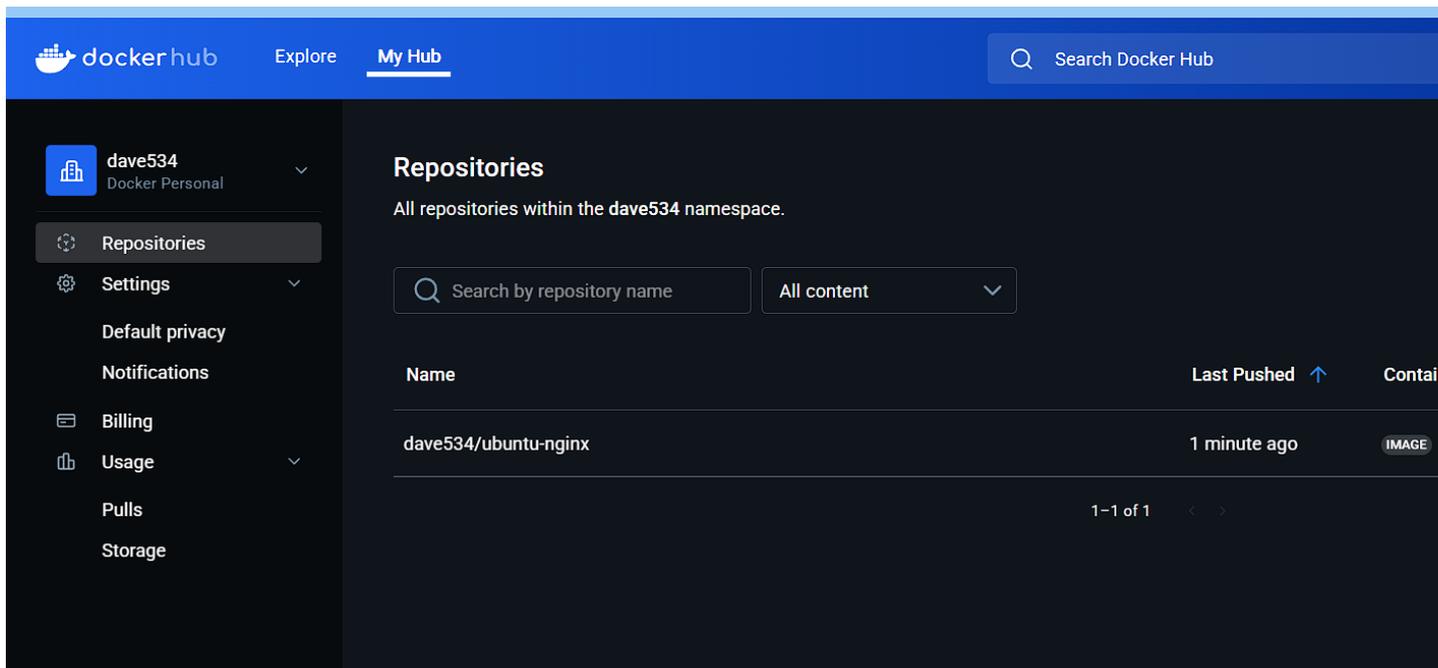
WARNING! Your credentials are stored unencrypted in '/root/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
root@david-virtual-machine:/home/david#
```

- Publiez l'image à l'aide de la commande « **push** » :

```
root@david-virtual-machine:/home/david# docker push iric2324/ubuntu-nginx:1.0
The push refers to repository [docker.io/iric2324/ubuntu-nginx]
An image does not exist locally with the tag: iric2324/ubuntu-nginx
root@david-virtual-machine:/home/david# docker push dave534/ubuntu-nginx:1.0
The push refers to repository [docker.io/dave534/ubuntu-nginx]
db63020a7d0c: Pushed
3abdd8a5e7a8: Pushed
1.0: digest: sha256:179f4103bb8ff297855d553b04732e182b8fb9f9f85029a95c8fd9bc7301a764 size: 741
root@david-virtual-machine:/home/david#
```

- Vérifiez votre compte sur Docker Hub. Vous trouverez votre nouvelle image :



Question : Le référentiel Docker Hub contient déjà une image nginx prête, comment vous pouvez alors l'utiliser directement ??

`docker run -d -p 80:80 nginx`

VII. Gestion du Réseau sous DOCKER

Mes types des réseaux

Lors de l'installation de Docker, trois réseaux sont créés automatiquement :

```
root@david-virtual-machine:/home/david# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
fd65863f4c84       bridge             bridge              local
8cf2698abd6e       host               host                local
49bc2222dc63       none              null                local
root@david-virtual-machine:/home/david#
```

N.B : Lors de la création d'un conteneur, si l'on ne spécifie pas un réseau particulier, les conteneurs sont connectés par défaut au **Bridge docker0**.

```
root@david-virtual-machine:/home/david# ip addr show docker0
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
    link/ether 6e:af:0a:5d:9e:e6 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
root@david-virtual-machine:/home/david#
```

- Testez la commande suivante, et dites ce qu'elle retourne :

```
root@david-virtual-machine:/home/david# docker network inspect bridge
[
  {
    "Name": "bridge",
    "Id": "fd65863f4c84b905a50cb18f55635797daf74e4d7ad6ed8264c6b71cc71f66e7",
    "Created": "2025-04-27T00:34:48.828175317+01:00",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv4": true,
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.17.0.0/16",
          "Gateway": "172.17.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {},
    "Options": {
      "com.docker.network.bridge.default_bridge": "true",
      "com.docker.network.bridge.enable_icc": "true",
      "com.docker.network.bridge.enable_ip_masquerade": "true",
      "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
      "com.docker.network.bridge.name": "docker0",
      "com.docker.network.driver.mtu": "1500"
    },
    "Labels": {}
  }
]
root@david-virtual-machine:/home/david#
```

- Créez deux conteneurs avec l'image ubuntu :

```
root@david-virtual-machine:/home/david# docker run -itd --name containerOne ubuntu
9b0dc3bcf2fe0770d3106170a07c2d261811f94fc1eeaff516949a9e7ca64d0a
root@david-virtual-machine:/home/david# docker run -itd --name containertwo ubuntu
6d8e366ce093dc8c49cbc60ff76b42e8d671d765d1dd1a557b6fbbebf4d7583e
root@david-virtual-machine:/home/david#
```

- Visualisez les infos du réseau avec la commande `inspect`

```
},
"ConfigOnly": false,
"Containers": {
  "6d8e366ce093dc8c49cbc60ff76b42e8d671d765d1dd1a557b6fbbebf4d7583e": {
    "Name": "containertwo",
    "EndpointID": "f868d0b0726931ac0f3ff96b7fb24bee29c6deb246ae437fae9bc7bf7aa4fc1f",
    "MacAddress": "62:f9:09:e8:13:28",
    "IPv4Address": "172.17.0.3/16",
    "IPv6Address": ""
  },
  "9b0dc3bcf2fe0770d3106170a07c2d261811f94fc1eeaff516949a9e7ca64d0a": {
    "Name": "containerOne",
    "EndpointID": "2eeba1752421e3dd56ad8be2860fb1cbbd6846fb38e3bfe8b0b4be320dd7d766",
    "MacAddress": "ea:81:3c:e9:a1:2c",
    "IPv4Address": "172.17.0.2/16",
    "IPv6Address": ""
  }
}
```

- Notez les adresses IP des deux conteneurs et effectuez un ping.

```
root@david-virtual-machine:/home/david# ping 172.17.0.3
PING 172.17.0.3 (172.17.0.3) 56(84) bytes of data.
64 bytes from 172.17.0.3: icmp_seq=1 ttl=64 time=0.236 ms
64 bytes from 172.17.0.3: icmp_seq=2 ttl=64 time=0.105 ms
64 bytes from 172.17.0.3: icmp_seq=3 ttl=64 time=0.116 ms
^C
--- 172.17.0.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2035ms
rtt min/avg/max/mdev = 0.105/0.152/0.236/0.059 ms
root@david-virtual-machine:/home/david# ping 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.160 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.101 ms
64 bytes from 172.17.0.2: icmp_seq=3 ttl=64 time=0.076 ms
^C
--- 172.17.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2030ms
rtt min/avg/max/mdev = 0.076/0.112/0.160/0.035 ms
root@david-virtual-machine:/home/david#
```

Réseaux Définis par l'Utilisateur

- Créez un réseau Bridge « monbridge » :

```
root@david-virtual-machine:/home/david# docker network create --driver bridge monbridge
12e10647f93ff48647ee64380a66c54b39d666adfa4d3d0ec445e1bfa7eaf690
root@david-virtual-machine:/home/david#
```

- Révérifiez maintenant vos réseaux sous Docker :

```
root@david-virtual-machine:/home/david# docker network inspect monbridge
[
  {
    "Name": "monbridge",
    "Id": "12e10647f93ff48647ee64380a66c54b39d666adfa4d3d0ec445e1bfa7eaf690",
    "Created": "2025-04-27T00:57:53.796615367+01:00",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv4": true,
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.18.0.0/16",
          "Gateway": "172.18.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {},
    "Options": {},
    "Labels": {}
  }
]
root@david-virtual-machine:/home/david#
```

```
root@david-virtual-machine:/home/david# docker network ls
NETWORK ID        NAME          DRIVER        SCOPE
fd65863f4c84     bridge       bridge        local
8cf2698abd6e     host         host          local
12e10647f93f     monbridge    bridge        local
49bc2222dc63     none         null          local
root@david-virtual-machine:/home/david#
```

- Connectez deux conteneurs différents au réseau monbridge :

```
root@david-virtual-machine:/home/david# docker run --network monbridge -itd --rm
ec3f79d771e29a751237c3138de93b5720bfe3c010024537d02ab2ac2e14ea6d
root@david-virtual-machine:/home/david# docker run --network monbridge -itd --rm
7800a2ac294b95ec6ab41c4a16c2f986f7a5eabe6e1fe0c53a70bf1b97b9b57b
root@david-virtual-machine:/home/david#
```

-

```

root@david-virtual-machine:/home/david# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS          NAMES
7800a2ac294b   ubuntu   "/bin/bash"             33 seconds ago  Up 32 seconds  -             contn2
ec3f79d771e2   ubuntu   "/bin/bash"             49 seconds ago  Up 48 seconds  -             contn1
6d8e366ce093   ubuntu   "/bin/bash"             7 minutes ago   Up 7 minutes   -             containertwo
9b0dc3bcf2fe   ubuntu   "/bin/bash"             7 minutes ago   Up 7 minutes   -             containerOne
root@david-virtual-machine:/home/david#

```

- Exécutez le premier conteneur :

```

root@david-virtual-machine:/home/david# docker exec -it ec3f79d771e2 bash
root@ec3f79d771e2:/#

```

- Testons le serveur DNS intégré : Docker daemon exécute un serveur DNS intégré qui fournit une résolution de noms aux conteneurs connectés au réseau créé par les utilisateurs, de sorte que ces conteneurs peuvent résoudre les noms de d'hôtes en adresses IP.

```

root@ec3f79d771e2:/# 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://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [21.9 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1077 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://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [985 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [1178 kB]
75% [10 Packages 15.7 MB/19.3 MB 81%] 2254 kB/s 3s

```

- o Interrogez le DNS à travers la commande « nslookup » :

```

/# nslookup contn2

```

```

root@ec3f79d771e2:/# nslookup contn2
Server:          127.0.0.11
Address:         127.0.0.11#53

Non-authoritative answer:
Name:   contn2
Address: 172.18.0.3

root@ec3f79d771e2:/# █

```

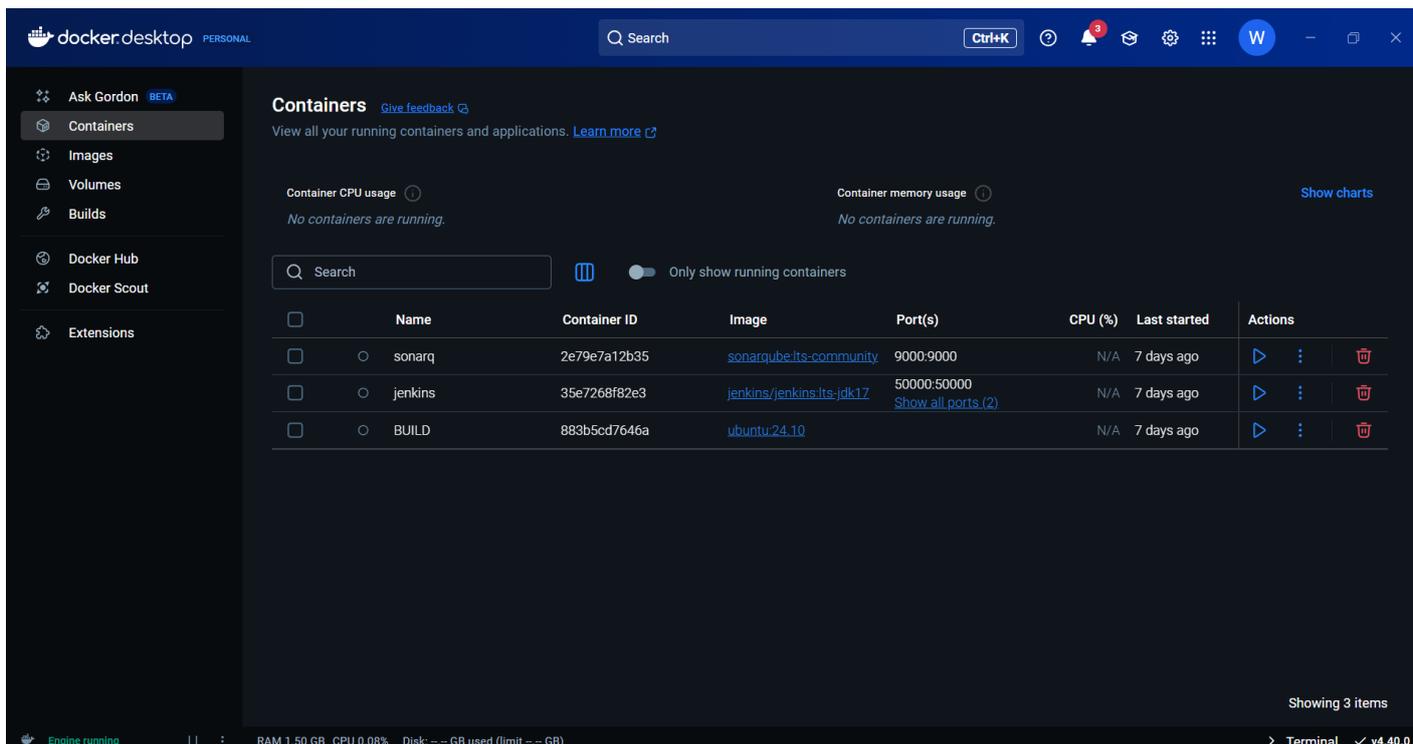
Le serveur DNS intégré de Docker a correctement traduit le nom `contn2` en son adresse IP Docker `172.18.0.3`, en utilisant le service DNS spécial sur `127.0.0.11`.

VIII. Docker sous Windows

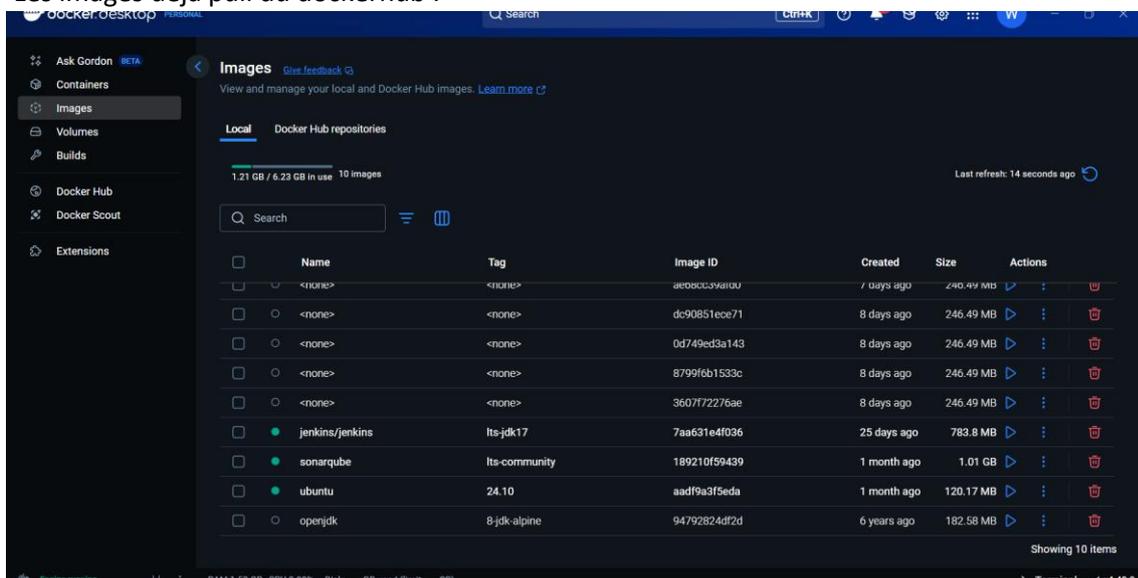
Docker Desktop

Question : installez Docker Desktop sous Windows et explorez l'environnement ??

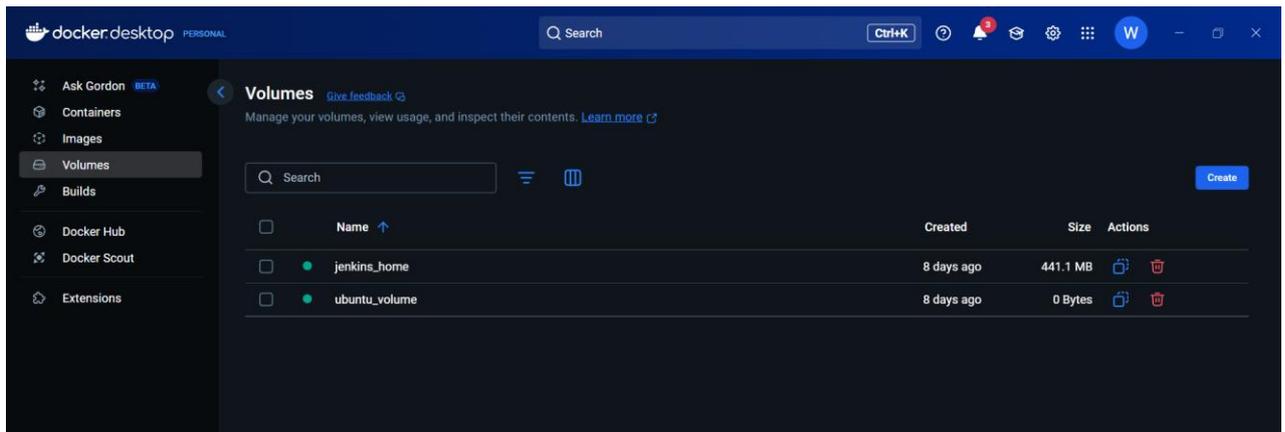
Nous avons plusieurs pages , comme les conteneurs :



Les images déjà pull du dockerhub :



Aussi les volumes associé à ces conteneurs :



Et ainsi docker hub pour exploiter les différents images :

