

## Travaux Pratiques Cloud Computing N°2 Manipulations de base

### Objectifs :

L'objectif principal de ce TP est de se familiariser avec les commandes de base Docker au niveau de la manipulation des images et des conteneurs.

### Boîte à outils :

- Les outils qui seront utilisés tout au long de ce TP sont comme suit : Ubuntu desktop.
- Disposer des droits d'administration.
- Disposer d'une connexion à Internet configurée et activée.

### 1. Trouver une image docker préexistante

Connexion au compte Docker Hub par ligne de commande:

```
$ sudo docker login
```

```
Y4Y4@smooth MINGW64 ~  
$ docker login  
Login with your Docker ID to push and pull images from Docker Hub. If you do  
n't have a Docker ID, head over to https://hub.docker.com to create one.  
Username: Y4HYA4  
Password:
```

Déconnexion du compte Docker Hub par ligne de commande:

```
$ sudo docker logout
```

```
Y4Y4@smooth MINGW64 ~  
$ docker logout  
Removing login credentials for https://index.docker.io/v1/  
Y4Y4@smooth MINGW64 ~
```

Maintenant, pour **trouver une image existante**, Docker permet de rechercher des images en utilisant la commande suivante :

```
$ sudo docker search [OPTIONS] TERM
```

Voici un exemple :

```
$ sudo docker search apache
```

Le résultat est le suivant :

```
wlded@wlded-VirtualBox:~$ sudo docker search apache
NAME          STARS     OFFICIAL   AUTOMATED  DESCRIPTION
httpd         3368      [OK]
tomcat        2948      [OK]
cassandra     1242      [OK]
maven         1159      [OK]
zookeeper     1009      [OK]
solr          815       [OK]
apache/airflow 219
apache/nifi    205       [OK]
eboraas/apache-php 144      [OK]
apache/zeppelin 143      [OK]
eboraas/apache 92       [OK]
```

Mon Essay :

```
Y4Y4@smooth MINGW64 ~
$ docker logout
Removing login credentials for https://index.docker.io/v1/

Y4Y4@smooth MINGW64 ~
$ docker search apache
NAME          STARS     OFFICIAL   AUTOMATED  DESCRIPTION
httpd         4263      [OK]
tomcat        3452      [OK]
cassandra     1436      [OK]
maven         1428      [OK]
zookeeper     1302      [OK]
solr          937       [OK]
apache/airflow 410
flink         364       [OK]
apache/nifi    266       [OK]
storm         181       [OK]
apache/superset 176
apache/zeppelin 165      [OK]
groovy        137       [OK]
apache/skywalking-oap-server 125
tomEE         125
```

Les colonnes d'informations renvoient :

- ✓ Le nom de l'image
- ✓ La description de l'image
- ✓ Le nombre d'étoiles
- ✓ Une indication du caractère officiel de l'image
- ✓ Une indication du fait que la compilation de l'image a été réalisée de manière automatique.

Pour plus de détails sur cette commande, consulter l'url suivante :

<https://docs.docker.com/engine/reference/commandline/search/>

## 2. Obtenir la liste des commandes du client docker

```
$ sudo docker help
```

```
Y4Y4@smooth MINGW64 ~
$ docker help

Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default
                       "C:\\Users\\Admin\\.docker")
  -c, --context string Name of the context to use to connect to the
                       daemon (overrides DOCKER_HOST env var and
                       default context set with "docker context use")
  -D, --debug          Enable debug mode
  -H, --host list      Daemon socket(s) to connect to
  -l, --log-level string Set the logging level
                       ("debug"|"info"|"warn"|"error"|"fatal")
                       (default "info")
  --tls               Use TLS; implied by --tlsverify
  --tlscacert string  Trust certs signed only by this CA (default
                       "C:\\Users\\Admin\\.docker\\ca.pem")
  --tlscert string    Path to TLS certificate file (default
                       "C:\\Users\\Admin\\.docker\\cert.pem")
  --tlskey string     Path to TLS key file (default
                       "C:\\Users\\Admin\\.docker\\key.pem")
  --tlsverify         Use TLS and verify the remote
  -v, --version       Print version information and quit

Management Commands:
  builder      Manage builds
  buildx*     Docker Buildx (Docker Inc., v0.9.1)
  compose*    Docker Compose (Docker Inc., v2.10.2)
  config      Manage Docker configs
  container   Manage containers
  context     Manage contexts
  extension*  Manages Docker extensions (Docker Inc., v0.2.9)
  image       Manage images
  manifest    Manage Docker image manifests and manifest lists
  network     Manage networks
```

Expliquer l'utilisation des commandes suivantes:

- ✓ run
- ✓ images
- ✓ image
- ✓ pull
- ✓ ps -a
- ✓ rm
- ✓ stats
- ✓ stop

en utilisant la commande :

```
$ sudo docker help nom_commande
```

```
Y4Y4@smooth MINGW64 ~
$ docker help run

Usage:  docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run a command in a new container

Options:
  --add-host list          Add a custom host-to-IP mapping
                           (host:ip)
  -a, --attach list       Attach to STDIN, STDOUT or STDERR
  --blkio-weight uint16   Block IO (relative weight),
                           between 10 and 1000, or 0 to
                           disable (default 0)
  --blkio-weight-device list
                           Block IO weight (relative device
                           weight) (default [])
  --cap-add list          Add Linux capabilities
  --cap-drop list        Drop Linux capabilities
  --cgroup-parent string  Optional parent cgroup for the
                           container
  --cgroupns string       Cgroup namespace to use
                           (host|private)
                           'host':    Run the container in
                           the Docker host's cgroup namespace
                           'private': Run the container in
                           its own private cgroup namespace
                           ':':       Use the cgroup
                           namespace as configured by the
                           default-cgroupns-mode
                           option on the daemon (default)
  --cidfile string       Write the container ID to the file
  --cpu-period int       Limit CPU CFS (Completely Fair
                           Scheduler) period
  --cpu-quota int        Limit CPU CFS (Completely Fair
                           Scheduler) quota
```

### 3. Récupération (téléchargement) d'une image dans sa dernière version

```
$ sudo docker pull ubuntu :latest
```

```
latest: Pulling from library/ubuntu
ae79f2514705: Pulling fs layer
c59d01a7e4ca: Pulling fs layer
41ba73a9054d: Pulling fs layer
f1bbfd495cc1: Waiting
0c346f7223e2: Waiting
```

Mon Essai :

```
Y4Y4@smooth MINGW64 ~
$ docker pull ubuntu:latest
latest: Pulling from library/ubuntu
6e3729cf69e0: Pull complete
Digest: sha256:27cb6e6ccef575a4698b66f5de06c7ecd61589132d5a91d098f7f3f9285415a9
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

On peut aussi télécharger la version 18.04 :

```
wided@wided-VirtualBox:~$ sudo docker pull ubuntu:18.04
18.04: Pulling from library/ubuntu
d519e2592276: Pull complete
d22d2dfcfa9c: Pull complete
b3afe92c540b: Pull complete
Digest: sha256:ea188fdc5be9b25ca048f1e882b33f1bc763fb976a8a4fea446b38ed0efcbeba
Status: Downloaded newer image for ubuntu:18.04
docker.io/library/ubuntu:18.04
```

Mon essai :

```
Y4Y4@smooth MINGW64 ~
$ docker pull ubuntu:18.04
18.04: Pulling from library/ubuntu
fb668870d8a7: Pull complete
Digest: sha256:daf3e62183e8aa9a56878a685ed26f3af3dd8c08c8fd11ef1c167a1
aa9bd66a3
Status: Downloaded newer image for ubuntu:18.04
docker.io/library/ubuntu:18.04
```

#### 4. Afficher les images Docker présentes sur la machine

```
$ sudo docker images
```

Le résultat sera un affichage comme cela :

```
wided@wided-VirtualBox:~$ sudo docker images
REPOSITORY          TAG          IMAGE ID          CREATED           SIZE
busybox             latest       491198851f0c     4 days ago       1.23MB
ubuntu              latest       f63181f19b2f     4 weeks ago      72.9MB
ubuntu              18.04       c090eaba6b94     4 weeks ago      63.3MB
hello-world         latest       bf756fb1ae65     13 months ago    13.3kB
```

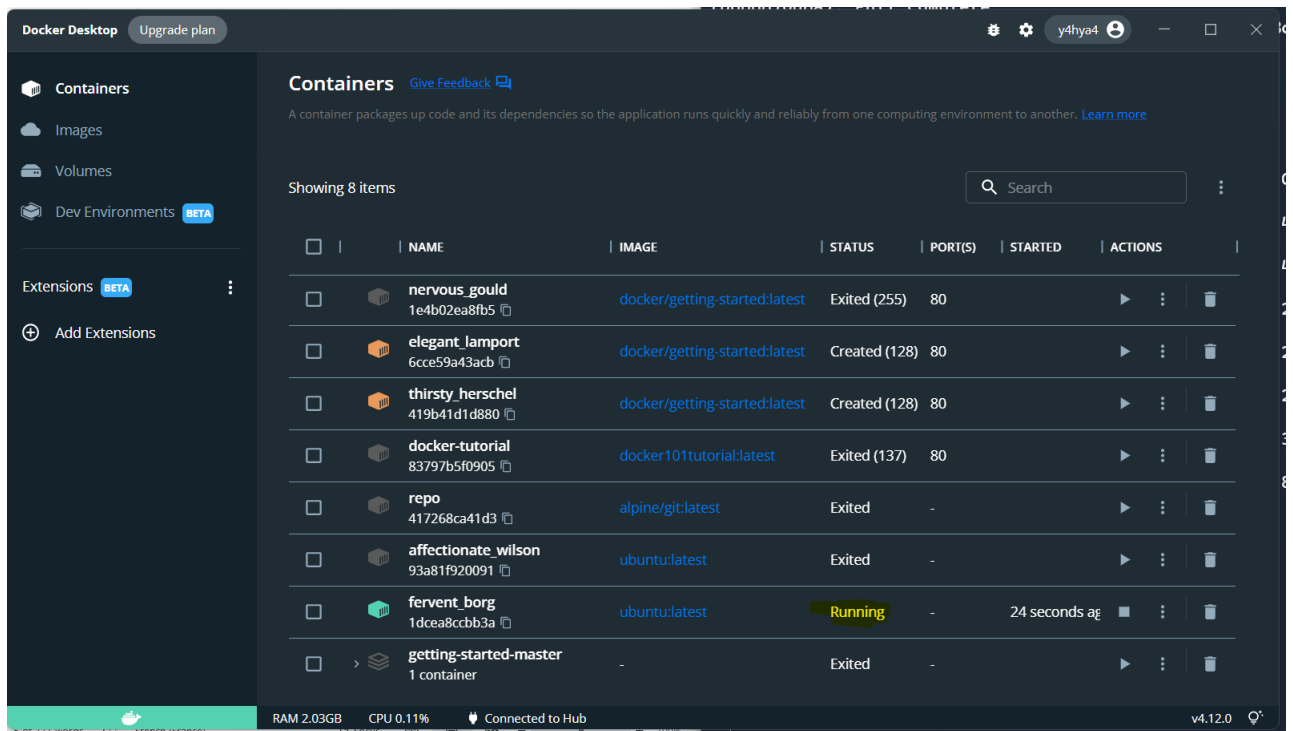
Mon Essai :

```
Y4Y4@smooth MINGW64 ~
$ docker images
REPOSITORY          TAG          IMAGE ID          CREATED           SIZE
ubuntu              latest       6b7dfa7e8fdb     4 days ago       77.8MB
ubuntu              18.04       251b86c83674     4 days ago       63.1MB
petstorepetservice latest       4e117363cc89     2 months ago    131MB
docker101tutorial  latest       44f5875ec963     2 months ago    28.9MB
getting-started-master_docs latest       e53550cc5330     2 months ago    90.7MB
alpine/git          latest       692618a0d74d     3 months ago    43.4MB
docker/getting-started latest       cb90f98fd791     8 months ago    28.8MB
```

## 5. Lancement sans option d'un conteneur :

```
$ sudo docker run ubuntu
```

```
Y4Y4@smooth MINGW64 ~  
$ docker run ubuntu
```



The screenshot shows the Docker Desktop interface. On the left, there is a sidebar with navigation options: Containers, Images, Volumes, Dev Environments (BETA), Extensions (BETA), and Add Extensions. The main area displays a list of containers under the heading "Containers". A search bar is present above the list. The list contains 8 items, with the following details:

	NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
<input type="checkbox"/>	nervous_gould 1e4b02ea8fb5	docker/getting-started:latest	Exited (255)	80		▶ ⋮ 🗑️
<input type="checkbox"/>	elegant_lampport 6cce59a43acb	docker/getting-started:latest	Created (128)	80		▶ ⋮ 🗑️
<input type="checkbox"/>	thirsty_herschel 419b41d1d880	docker/getting-started:latest	Created (128)	80		▶ ⋮ 🗑️
<input type="checkbox"/>	docker-tutorial 83797b5f0905	docker101tutorial:latest	Exited (137)	80		▶ ⋮ 🗑️
<input type="checkbox"/>	repo 417268ca41d3	alpine/git:latest	Exited	-		▶ ⋮ 🗑️
<input type="checkbox"/>	affectionate_wilson 93a81f920091	ubuntu:latest	Exited	-		▶ ⋮ 🗑️
<input type="checkbox"/>	fervent_borg 1dcea8ccb3a	ubuntu:latest	Running	-	24 seconds ago	▶ ⋮ 🗑️
<input type="checkbox"/>	getting-started-master 1 container	-	Exited	-		▶ ⋮ 🗑️

At the bottom of the interface, system statistics are shown: RAM 2.03GB, CPU 0.11%, and Connected to Hub. The version number v4.12.0 is also visible in the bottom right corner.

## 6. Lister tous les conteneurs (y compris les inactifs) :

```
$ sudo docker ps -a
```

```
Y4Y4@smooth MINGW64 ~
$ docker ps -a
CONTAINER ID   IMAGE          STATUS          COMMAND
CREATED              STATUS          PORTS
NAMES
1dcea8ccbb3a   ubuntu:latest  Up About a minute  "bash"
About a minute ago
fervent_borg
93a81f920091   ubuntu        Exited (0) 2 minutes ago  "bash"
2 minutes ago
affectionate_wilson
05a82f422855   getting-started-master_docs  "mkdocs serve -a 0.0..."
2 months ago      Exited (1) 2 months ago
getting-started-master_docs_1
1e4b02ea8fb5   docker/getting-started        "/docker-entrypoint..."
2 months ago      Exited (255) 2 months ago  0.0.0.0:80->80/tcp
nervous_gould
6cce59a43acb   docker/getting-started        "/docker-entrypoint..."
2 months ago      Created
elegant_lamport
419b41d1d880   docker/getting-started        "/docker-entrypoint..."
2 months ago      Created
thirsty_herschel
83797b5f0905   docker101tutorial            "/docker-entrypoint..."
2 months ago      Exited (137) 2 months ago
docker-tutorial
417268ca41d3   alpine/git                    "git clone https://g..."
2 months ago      Exited (0) 2 months ago
repo
```



Le résultat est :

```
wided@wided-VirtualBox:~$ sudo docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS
PORTS         NAMES
fefce1756fb8   ubuntu    "/bin/bash"            11 seconds ago Exited (0) 8 second
s ago
great_mendel
e90046054c07   ubuntu    "/bin/bash"            25 minutes ago Exited (0) 25 minut
es ago
dazzling_golick
1a8de480eef0   hello-world "/hello"                11 days ago    Exited (0) 11 days
ago
cool_babbage
b15165390d76   hello-world "/hello"                12 days ago    Exited (0) 12 days
ago
priceless_mclaren
726058c97e86   ubuntu    "bash"                  13 days ago    Exited (127) 12 day
s ago
gallant_williamson
6e1945da730e   hello-world "/hello"                13 days ago    Exited (0) 13 days
ago
youthful_wiles
```

Voici l'explication des différentes colonnes :

- ✓ **CONTAINER ID** : L'identificateur du conteneur
- ✓ **IMAGE** : L'image sur laquelle est basé le conteneur
- ✓ **COMMAND** : Dernière commande lancée lors de l'exécution de votre image
- ✓ **CREATED** : date de création de votre conteneur
- ✓ **STATUS** : statut de votre conteneur, voici une liste des **différents états d'un conteneur** :
  - **created** : conteneur créé mais non démarré (cet état est possible avec la commande docker create)
  - **restarting** : conteneur en cours de redémarrage
  - **running** : conteneur en cours d'exécution
  - **paused** : conteneur stoppé manuellement (cet état est possible avec la commande docker pause)
  - **exited** : conteneur qui a été exécuté puis terminé
  - **dead** : conteneur que le service docker n'a pas réussi à arrêter correctement (généralement en raison d'un périphérique occupé ou d'une ressource utilisée par le conteneur)
- ✓ **PORTS** : les ports utilisés par votre conteneur
- ✓ **NAMES** : nom de votre conteneur

## 7. Lancement d'un conteneur en mode interactif

```
$ sudo docker run -it ubuntu
```

```
wided@wided-VirtualBox:~$ sudo docker run -it ubuntu
[sudo] Mot de passe de wided :
root@48b6f65d2623:/#
```

Mon Essai : I did log into the ubuntu ssh.

```
root@a7758337e39d:/#  
exit
```

```
Y4Y4@smooth MINGW64 ~
```

```
wided@wided-VirtualBox:~$ sudo docker run -it ubuntu  
[sudo] Mot de passe de wided :  
root@48b6f65d2623:/# ls  
bin  dev  home  lib32  libx32  mnt  proc  run  srv  tmp  var  
boot  etc  lib  lib64  media  opt  root  sbin  sys  usr
```

En lançant le conteneur en mode interactif, nous ne sommes plus identifiés comme l'utilisateur courant sur la machine hôte, mais comme root sur la machine nommée ou tout autre