

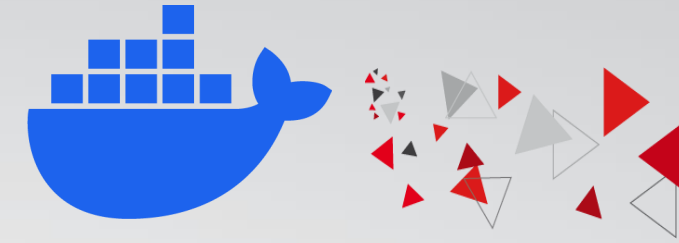


## DevOps WORKSHOP La livraison continue - Docker



ESPRIT – UP ASI (Architecture des Systèmes d'Information)  
Bureau E204

# ► Docker - Installation



- Allez dans le répertoire de votre machine virtuelle Ubuntu, démarrez la machine virtuelle (`vagrant up`) et ouvrez un terminal (`vagrant ssh`).

```
Sélection vagrant@vagrant: ~  
PS D:\Vagrant\Ubuntu> vagrant up  
Bringing machine 'default' up with 'virtualbox' provider...  
==> default: Checking if box 'bento/ubuntu-22.04' version '202309.08.0' is up to date...  
default: /vagrant => D:/Vagrant/Ubuntu  
==> default: Machine already provisioned. Run `vagrant provision` or use the `--provision`  
==> default: flag to force provisioning. Provisioners marked to run always will still run.  
PS D:\Vagrant\Ubuntu> vagrant ssh  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-83-generic x86_64)  
  
Last login: Mon Sep 11 14:26:05 2023 from 10.0.2.2  
vagrant@vagrant: $  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-83-generic x86_64)
```



# Docker - Installation

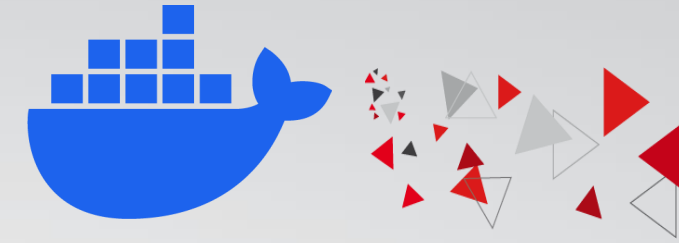


- Pour installer Docker, exécutez les commandes suivantes :

```
sudo apt-get install ca-certificates curl gnupg
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg
echo \
"deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu \
"$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-
plugin
sudo service docker start
sudo service docker enable
```



# ► Docker - Installation



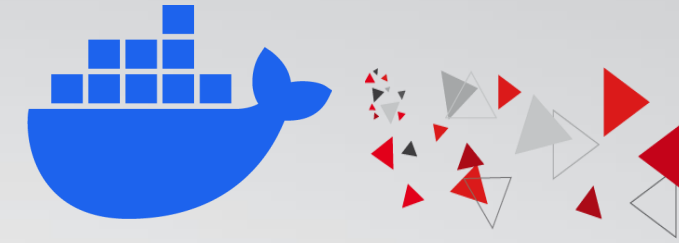
- Docker est déjà en cours d'exécution après l'installation. Pour vérifier son état, utilisez la commande suivante : `sudo systemctl status docker`

```
vagrant@vagrant: ~  
● docker.service - Docker Application Container Engine  
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)  
   Active: active (running) since Tue 2023-09-26 00:04:14 UTC; 14min ago  
TriggeredBy: ● docker.socket  
     Docs: https://docs.docker.com  
    Main PID: 8107 (dockerd)  
      Tasks: 8  
     Memory: 26.2M  
        CPU: 5.962s  
    CGroup: /system.slice/docker.service  
            └─8107 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

- **CTRL C** pour sortir de la commande de status



# ► Docker - Installation



- Pour accorder des droits de lecture et d'écriture (mais pas d'exécution) à l'utilisateur "vagrant", à son groupe et à tout autre utilisateur ( $6 = (R=1 / W=1 X=0)$ ) sur le fichier */var/run/docker.sock*, utilisez la commande suivante :

```
sudo chmod 666 /var/run/docker.sock
```

- Pour vous assurer que tout fonctionne correctement, vous pouvez vérifier la version de Docker et exécuter l'image "hello-world" (qui sera automatiquement téléchargée depuis Docker Hub) en utilisant les commandes suivantes :
  - Pour vérifier la version de Docker : `docker -v`
  - Pour exécuter l'image "hello-world" : `docker run hello-world`





# Docker - Installation

```
vagrant@vagrant: ~  
vagrant@vagrant:~$ docker v  
Docker version 24.0.6, build ed223bc  
vagrant@vagrant:~$ docker run hello-world  
docker: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: dial unix /var/run/docker.sock: connect: permission denied.  
See 'docker run --help'.  
vagrant@vagrant:~$ sudo chmod 666 /var/run/docker.sock  
vagrant@vagrant:~$ docker run hello-world  
Unable to find image 'hello-world:latest' locally  
latest: Pulling from library/hello-world  
719385e32844: Pull complete  
Digest: sha256:4f53e2564790c8e7856ec08e384732aa38dc43c52f02952483e3f003afbf23db  
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.
```

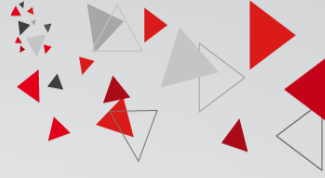
# ► Docker - Manipulation



- Pour importer une image, vous avez plusieurs options :
  - **Docker Hub** : Importer une image depuis le référentiel cloud Docker.
  - **Utiliser un fichier Dockerfile** : Créer une image en utilisant un fichier Dockerfile et les commandes associées.
  - **À partir d'un registre local** : Importer une image à partir d'un registre local, sans accéder au cloud.
  - **À partir d'un conteneur** : Créer une nouvelle image à partir d'un conteneur existant, en prenant le contenu du conteneur pour créer une nouvelle image ( Image => conteneur => image).



# Docker - Docker Hub



- **Docker Hub** est le registre officiel de Docker.
- C'est un service **SaaS** (Software As A Service ou logiciel en tant que service) qui rassemble des applications sous forme de conteneurs (images). Ces images sont créées par des développeurs et des opérationnels et sont accessibles à tous.
- Docker lui-même peut également fournir ces images.
- Vous avez la possibilité de télécharger ces images et de partager les vôtres.





# ► Docker - Docker File



- Un **Dockerfile** est un fichier texte contenant les commandes que l'on utiliserait en ligne de commande pour créer notre propre image.
- Il part d'une image de base standard à laquelle nous ajoutons les éléments spécifiques à l'application que nous souhaitons déployer. Créons le **Dockerfile**



Dockerfile

```
vagrant@vagrant: ~/docker
vagrant@vagrant:~$ mkdir docker
vagrant@vagrant:~$ cd docker
vagrant@vagrant:~/docker$ vi Dockerfile
vagrant@vagrant:~/docker$ cat Dockerfile
FROM alpine
RUN apk add openjdk11
EXPOSE 80
CMD "java"
vagrant@vagrant:~/docker$
```

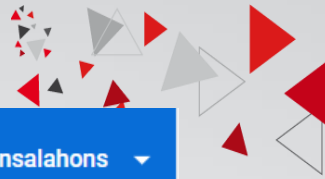
```
vagrant@vagrant: ~/d
FROM alpine
RUN apk add openjdk11
EXPOSE 80
CMD "java"
```


-- INSERT --

4,11

All

# Docker - Docker File




 **dockerhub**

alpine

Explore Repositories Organizations Help

Upgrade

 bensalahons

**Filters**

1 - 25 of 10 000 results for **alpine**.

Best Match


Products


☐ Images


☐ Extensions

☐ Plugins

Trusted Content

☐  Docker Official Image

☐  Verified Publisher

☐  Sponsored OSS





Operating Systems

☐ Linux

☐ Windows


Architectures

☐ ARM





 **alpine**  Docker Official Image •  1B+ •  10K+  
Updated 24 days ago  
A minimal Docker image based on Alpine Linux with a complete package index and only 5 ...  

Linux IBM Z riscv64 x86-64 ARM ARM 64 386 PowerPC 64 LE

Pulls: 11,710,561  
Last week




[Learn more](#)





 **alpinelinux/docker-cli**  Sponsored OSS •  5M+ •  9  
By Alpine Linux • Updated 2 days ago  
Simple and lightweight Alpine Linux image with docker-cli  

Linux 386 x86-64 arm arm64 ppc64le riscv64 IBM Z

Pulls: 30,919  
Last week



[Learn more](#)

 **alpinelinux/alpine-gitlab-ci**  Sponsored OSS •  100K+ •  3  
By Alpine Linux • Updated 2 days ago  
Build Alpine Linux packages with Gitlab CI



# ► Docker - Docker File



Les instructions de base que peut contenir un DockerFile sont les suivantes :

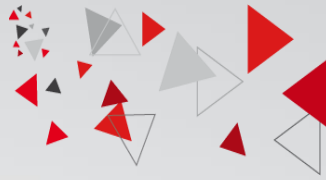
- **FROM** permet de définir depuis quelle base votre image va être créée
- **LABEL** permet de définir l'auteur de l'image
- **RUN** permet de lancer une commande
- **ADD** permet de copier un fichier depuis la machine hôte ou depuis une URL
- **EXPOSE** permet d'exposer un port du container vers l'extérieur
- **CMD** détermine la commande qui sera exécutée lorsque le container démarrera
- **ENTRYPOINT** permet d'ajouter une commande qui sera exécutée par défaut
- **WORKDIR** permet de définir le dossier de travail pour toutes les autres commandes
- **ENV** permet de définir des variables d'environnements
- **VOLUMES** permet de créer un point de montage qui permettra de persister les données



Docker File

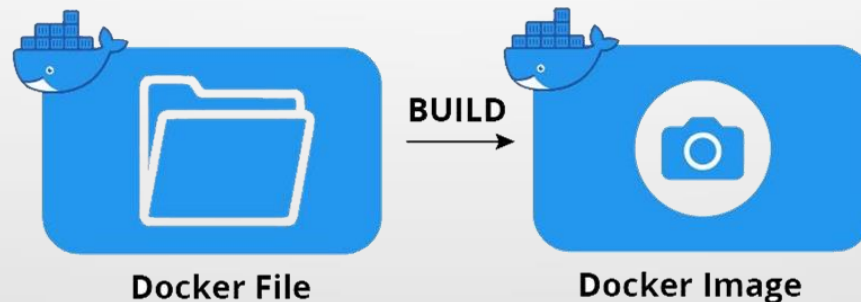


# ► Création d'une image à partir d'un Dockerfile



- Pour construire l'image Docker définie dans le fichier Dockerfile ci-dessus  

```
docker build -t "Nom_de_l'image_à_créer" "Chemin_vers_le_fichier"
```
- Vous pouvez spécifier "." (un point) si le fichier Dockerfile se trouve dans le répertoire courant.
- Si vous n'utilisez pas le nom de fichier Dockerfile par défaut, l'option **-f** est requise.
- Dans le nom de l'image à créer, vous devez spécifier votre nom d'utilisateur sur Docker Hub pour pouvoir envoyer votre image sur Docker Hub (par exemple, "bensalahons/alpine").



# Création d'une image à partir d'un Dockerfile



- A partir de l'image officielle d'Alpine (sur Docker Hub), nous allons créer notre propre image personnalisée "alpine:1.0.0" avec Java 11.

```
vagrant@vagrant:~/docker$ cat Dockerfile
```

```
FROM alpine
```

```
RUN apk add openjdk11
```

```
EXPOSE 80
```

```
CMD "java"
```

```
vagrant@vagrant:~/docker$ docker build -t bensalahons/alpine:1.0.0 .
```

```
[+] Building 158.9s (6/6) FINISHED
```

docker:default

```
=> [internal] load .dockerignore
```

0.2s

```
=> => transferring context: 2B
```

0.1s

```
=> [internal] load build definition from Dockerfile
```

0.3s

```
=> => transferring dockerfile: 92B
```

0.0s

```
=> [internal] load metadata for docker.io/library/alpine:latest
```

3.7s

```
=> [1/2] FROM docker.io/library/alpine@sha256:7144f7bab3d4c2648d7e59409f15ec52a18006a128c733fcff20d3a4a54ba44a
```

5.0s

```
=> => resolve docker.io/library/alpine@sha256:7144f7bab3d4c2648d7e59409f15ec52a18006a128c733fcff20d3a4a54ba44a
```

0.1s

```
=> => sha256:7144f7bab3d4c2648d7e59409f15ec52a18006a128c733fcff20d3a4a54ba44a 1.64kB / 1.64kB
```

0.0s

```
=> => sha256:c5c5fda71656f28e49ac9c5416b3643eaa6a108a8093151d6d1afc9463be8e33 528B / 528B
```

0.0s

```
=> => sha256:7e01a0d0a1dcd9e539f8e9bbd80106d59efbdf97293b3d38f5d7a34501526cdb 1.47kB / 1.47kB
```

0.0s

```
=> => sha256:7264a8db6415046d36d16ba98b79778e18accee6ffa71850405994cffa9be7de 3.40MB / 3.40MB
```

3.5s

```
=> => extracting sha256:7264a8db6415046d36d16ba98b79778e18accee6ffa71850405994cffa9be7de
```

0.9s

```
=> [2/2] RUN apk add openjdk11
```

143.3s

```
=> exporting to image
```

6.1s

```
=> => exporting layers
```

6.0s

```
=> => writing image sha256:63ca191ca9cf3d0ec8668e45ff3e670891422944b91ce5fe4784bd79731f4958
```

0.0s

```
=> => naming to docker.io/bensalahons/alpine:1.0.0
```

0.0s

```
vagrant@vagrant:~/docker$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
bensalahons/alpine	1.0.0	63ca191ca9cf	35 seconds ago	273MB
hello-world	latest	9c7a54a9a43c	4 months ago	13.3kB



# ► Docker - Commandes de base



- Pour extraire une image ou un référentiel d'un registre Docker, tapez la commande suivante : `docker pull "nom_de_l'image":"version"`
  - Si vous ne spécifiez pas la version, Docker télécharge automatiquement la dernière version.

```
vagrant@vagrant: ~/docker
vagrant@vagrant:~/docker$ docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
7264a8db6415: Already exists
Digest: sha256:7144f7bab3d4c2648d7e59409f15ec52a18006a128c733fcff20d3a4a54ba44a
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest
vagrant@vagrant:~/docker$
```





# ► Docker - Commandes de base



Pour créer un conteneur, utilisez la commande suivante : `docker run "nom_de_l'image"`

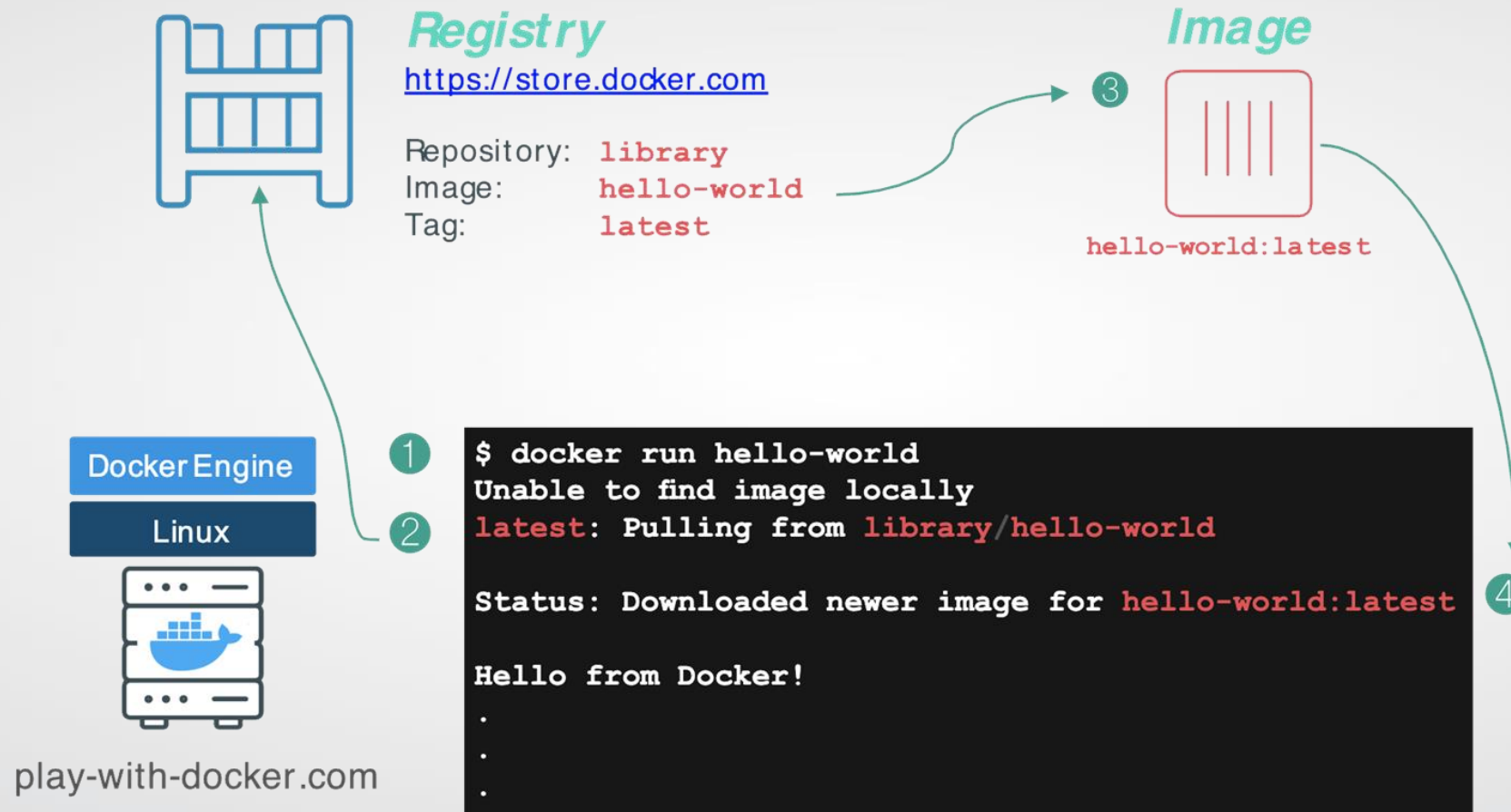
→ Si Docker ne trouve pas l'image localement, il la télécharge automatiquement depuis Docker Hub, puis crée le conteneur.

```
vagrant@vagrant:~/docker$ docker run mysql
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
5262579e8e45: Pulling fs layer
741b767e25b7: Pulling fs layer
06e0c37837cf: Pull complete
c6f5d3670db7: Pull complete
d5c567b29c3e: Pull complete
6effcc6561c9: Pull complete
1e1493d45d9c: Pull complete
7101609fa7d9: Pull complete
432a1261dc2a: Pull complete
865a24d6d1f2: Pull complete
Digest: sha256:566007208a3f1cc8f9df6b767665b5c9b800fc4fb5f863d17aa1df362880ed04
Status: Downloaded newer image for mysql:latest
2023-09-26 03:06:42+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.1.0-1.el8 started.
2023-09-26 03:06:48+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2023-09-26 03:06:49+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.1.0-1.el8 started.
2023-09-26 03:06:54+00:00 [ERROR] [Entrypoint]: Database is uninitialized and password option is not specified
```



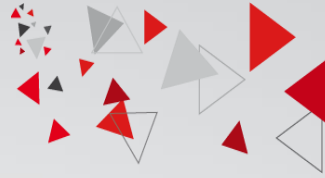
# ► Docker - Commandes de base

## Hello World: What Happened?



Source : Run a single task "Hello World"

# ► Docker - Commandes de base

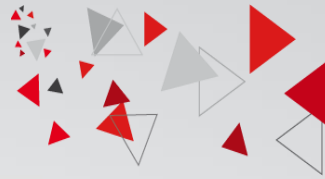


- Pour récupérer la liste des images, tapez la commande : `docker images`

```
vagrant@vagrant: ~/docker
vagrant@vagrant:~/docker$ docker images
REPOSITORY          TAG          IMAGE ID      CREATED        SIZE
bensalahons/alpine  1.0.0       63ca191ca9cf  About an hour ago  273MB
mysql               latest      b2013ac99101  4 days ago     577MB
alpine              latest      7e01a0d0a1dc  7 weeks ago    7.34MB
hello-world         latest      9c7a54a9a43c  4 months ago   13.3kB
vagrant@vagrant:~/docker$ |
```



# ► Docker - Commandes de base



- Pour obtenir plus de détails sur une image Docker, exécutez la commande suivante :

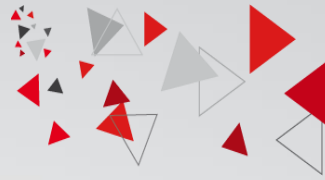
`docker image inspect "les_5_premières_lettres_de_l'ID_de_l'image"`

```
vagrant@vagrant:~/docker$ docker images
REPOSITORY          TAG          IMAGE ID      CREATED        SIZE
bensalahons/alpine  1.0.0       63ca191ca9cf  About an hour ago  273MB
mysql               latest      b2013ac99101  4 days ago     577MB
alpine              latest      7e01a0d0a1dc  7 weeks ago    7.34MB
hello-world         latest      9c7a54a9a43c  4 months ago   13.3kB
vagrant@vagrant:~/docker$ docker image inspect 63ca19
[
  {
    "Id": "sha256:63ca191ca9cf3d0ec8668e45ff3e670891422944b91ce5fe4784bd79731f4958",
    "RepoTags": [
      "bensalahons/alpine:1.0.0"
    ],
    "RepoDigests": [],
    "Parent": "",
    "Comment": "buildkit.dockerfile.v0",
    "Created": "2023-09-26T01:55:28.677139506Z",
```





# ► Docker - Commandes de base



- Pour récupérer la liste des conteneurs (y compris ceux arrêtés), vous pouvez utiliser les commandes suivantes :

`docker container ls -a` ou bien `docker ps -a`

- Pour obtenir la liste des conteneurs actifs (ceux qui sont en cours d'exécution) :

`docker ps`

```
vagrant@vagrant:~/docker$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
e9be20da1ec3   mysql         "docker-entrypoint.s..." 35 minutes ago Exited (1)    34 minutes ago nifty_borg
070cf909a42c   63ca191ca9cf  "/bin/sh -c \"java\""     59 minutes ago Exited (1)    59 minutes ago eager_cray
b6f5a99c1efa   hello-world   "/hello"                 3 hours ago   Exited (0)    3 hours ago   jovial_ritchie
vagrant@vagrant:~/docker$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
vagrant@vagrant:~/docker$
```

→ Pourquoi on ne peut pas observer le conteneur associé à l'image alpine:1.0.0 ?

# ► Docker - Commandes de base



- Lancer un conteneur en arrière-plan (en utilisant l'option -d : `docker run -itd`) pour y accéder ultérieurement :
- La commande ci-dessous démarre un conteneur et exécute un client SSH à l'intérieur.

```
vagrant@vagrant: ~/docker
vagrant@vagrant:~/docker$ docker run -itd bensalahons/alpine:1.0.0 /bin/sh
fa1552cadeaf81c1f7c7b5dbff53f7388f7a1aa1bdfadac8be38470165782ea6
vagrant@vagrant:~/docker$ docker ps
```

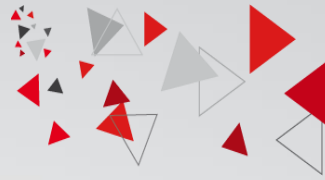
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
fa1552cadeaf	bensalahons/alpine:1.0.0	"/bin/sh"	14 seconds ago	Up 8 seconds	80/tcp	flamboyant_heisenberg

```
vagrant@vagrant:~/docker$
```





# ► Docker - Commandes de base



- Pour accéder à un conteneur déjà en cours d'exécution (en utilisant `docker exec`), vous pouvez interagir avec le conteneur comme si vous étiez dans une machine virtuelle ou un serveur Linux :

```
vagrant@vagrant: ~/docker
vagrant@vagrant:~/docker$ docker run -itd bensalahons/alpine:1.0.0 /bin/sh
fa1552cadeaf81c1f7c7b5dbff53f7388f7a1aa1bdfadac8be38470165782ea6
vagrant@vagrant:~/docker$ docker ps
CONTAINER ID   IMAGE               COMMAND             CREATED             STATUS              PORTS             NAMES
fa1552cadeaf   bensalahons/alpine:1.0.0  "/bin/sh"          14 seconds ago     Up 8 seconds       80/tcp            flamboyant_heisenberg

vagrant@vagrant:~/docker$ docker exec -it fa155 /bin/sh
/ # pwd
/
/ # ls
/ # ls
bin      etc      lib      mnt      proc     run      srv      tmp      var
dev      home    media    opt      root     sbin     sys      usr
```



# ► Docker - Commandes de base



- Pour supprimer une image, vous pouvez utiliser l'une des deux commandes suivantes :

```
docker image rm "nom_image" --force
```

ou 

```
docker rmi "nom_image" --force
```

```
vagrant@vagrant:~/docker$ docker images
REPOSITORY          TAG          IMAGE ID      CREATED        SIZE
bensalahons/alpine  1.0.0       63ca191ca9cf  2 hours ago   273MB
mysql               latest      b2013ac99101  4 days ago    577MB
alpine              latest      7e01a0d0a1dc  7 weeks ago   7.34MB
hello-world         latest      9c7a54a9a43c  4 months ago  13.3kB
vagrant@vagrant:~/docker$ docker rmi hello-world --force
Untagged: hello-world:latest
Untagged: hello-world@sha256:4f53e2564790c8e7856ec08e384732aa38dc43c52f02952483e3f003afbf23db
Deleted: sha256:9c7a54a9a43cca047013b82af109fe963fde787f63f9e016fdc3384500c2823d
vagrant@vagrant:~/docker$ docker images
REPOSITORY          TAG          IMAGE ID      CREATED        SIZE
bensalahons/alpine  1.0.0       63ca191ca9cf  2 hours ago   273MB
mysql               latest      b2013ac99101  4 days ago    577MB
alpine              latest      7e01a0d0a1dc  7 weeks ago   7.34MB
vagrant@vagrant:~/docker$
```



# ► Docker - Commandes de base



- Pour supprimer un conteneur, utilisez la commande : `docker rm "ID_du_conteneur"`

```
vagrant@vagrant:~/docker$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS
f             bensalahons/alpine:1.0.0   "/bin/sh"               23 minutes ago   Up 23 minutes   80/tcp
e9be20da1ec3   mysql         "docker-entrypoint.s..." About an hour ago Exited (1) About an hour ago
070cf909a42c   63ca191ca9cf  "/bin/sh -c \"java\""    2 hours ago      Exited (1) 2 hours ago
b6f5a99c1efa   9c7a54a9a43c  "/hello"                4 hours ago      Exited (0) 4 hours ago

vagrant@vagrant:~/docker$ docker remove fa1552cadeaf
Error response from daemon: You cannot remove a running container fa1552cadeaf81c1f7c7b5dbff53f7388f7a1aa1bdfadac8b6
tainer before attempting removal or force remove

vagrant@vagrant:~/docker$ docker stop fa1552cadeaf
fa1552cadeaf

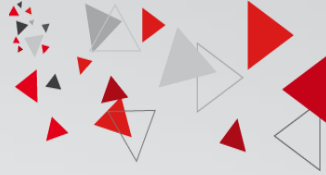
vagrant@vagrant:~/docker$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS          NAMES
vagrant@vagrant:~/docker$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS          NAMES
f             bensalahons/alpine:1.0.0   "/bin/sh"               24 minutes ago   Exited (137) 23 seconds ago
e9be20da1ec3   mysql         "docker-entrypoint.s..." About an hour ago Exited (1) About an hour ago
070cf909a42c   63ca191ca9cf  "/bin/sh -c \"java\""    2 hours ago      Exited (1) 2 hours ago
b6f5a99c1efa   9c7a54a9a43c  "/hello"                4 hours ago      Exited (0) 4 hours ago

vagrant@vagrant:~/docker$ docker rm fa1552cadeaf
fa1552cadeaf

vagrant@vagrant:~/docker$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS
e9be20da1ec3   mysql         "docker-entrypoint.s..." About an hour ago Exited (1) About an hour ago
070cf909a42c   63ca191ca9cf  "/bin/sh -c \"java\""    2 hours ago      Exited (1) 2 hours ago
vagrant@vagrant:~/docker$
```



# ► Docker - Commandes de base



- Si ce n'est pas déjà fait, créez un compte sur Docker Hub à l'adresse <https://hub.docker.com>, puis connectez-vous en utilisant la commande :  
`docker login`

```
vagrant@vagrant:~/docker$ docker 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: bensalahons
```

```
Password:
```

```
WARNING! Your password will be stored unencrypted in /home/vagrant/.docker/config.json.
```

```
Configure a credential helper to remove this warning. See
```

```
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
```

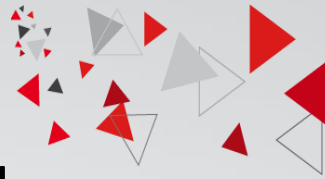
```
Login Succeeded
```

```
vagrant@vagrant:~/docker$ |
```





# ► Docker - Commandes de base



- Pour envoyer une image que vous avez créée vers Docker Hub, utilisez la commande : `docker push`

```
vagrant@vagrant: ~/docker
vagrant@vagrant:~/docker$ docker images
REPOSITORY          TAG          IMAGE ID        CREATED         SIZE
bensalahons/alpine  1.0.0        63ca191ca9cf    3 hours ago    273MB
mysql                latest       b2013ac99101    4 days ago     577MB
alpine               latest       7e01a0d0a1dc    7 weeks ago    7.34MB
vagrant@vagrant:~/docker$ docker push bensalahons/alpine:1.0.0
The push refers to repository [docker.io/bensalahons/alpine]
764e3bdab165: Pushed
4693057ce236: Mounted from library/alpine
1.0.0: digest: sha256:7fad49bab9b99cb0bff6012d0925b91f3d35e5a28ad597e52fb7ac1624d502f8 size: 741
vagrant@vagrant:~/docker$
```




# ► Docker - Commandes de base



The screenshot shows the Docker Hub web interface. The browser address bar displays 'hub.docker.com'. The navigation bar includes a search bar, links for 'Explore', 'Repositories', 'Organizations', and 'Help', an 'Upgrade' button, and a user profile dropdown for 'bensalahons'. Below the navigation bar, there are filters for 'bensalahons' and 'All Content', a search bar for repository names, and a 'Create repository' button. A list of repositories is shown, with 'bensalahons / alpine' highlighted. This repository is described as 'Contains: Image | Last pushed: 8 minutes ago' and has 'Inactive' status, 0 stars, 0 downloads, and is 'Public'. To the right, a promotional banner encourages creating an organization to manage repositories with a team.







► *"Apprendre par le projet, c'est découvrir  
par l'action, créer par la compréhension, et  
réussir par la persévérance."*