

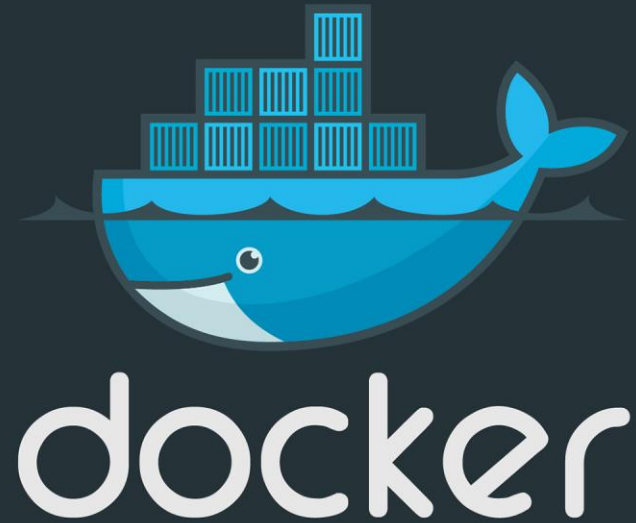
EXPLOITER LES CONTENEURS AVEC DOCKER

Pratique

Presenté par

ELJ Adel Architect JEE / Spring / Cloud / Devops

03/2020



SOMMAIRE J1

- Installation / Configuration
 - Windows
 - Linux
- Commandes simples
- docker run

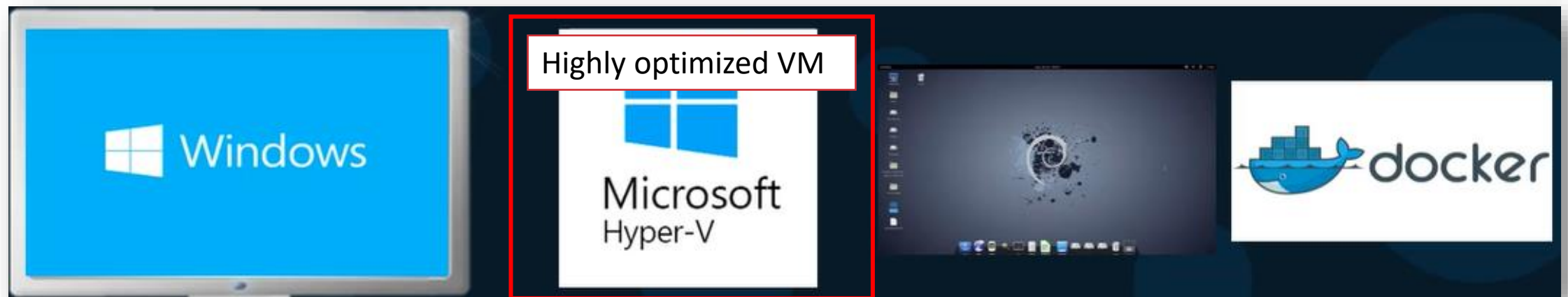
INSTALLATION / CONFIGURATION

installation Docker pour Windows

Docker ToolBox pour Windows < 10 pro



Docker Desktop pour Windows 10 pro



installation Docker Desktop pour Windows 10 Pro

docker for windows

All Videos News Images

About 83,600,000 results (0.68 seconds)

docs.docker.com › docker-for-windows ▾

Get started with Docker for Windows

Open a terminal window (Command Prompt or PowerShell) and run `docker --version` to ensure that you have a supported version.

Install Docker Desktop

The Docker Desktop Windows installer enables Hyper-V if ...

[More results from docker.com »](#)


Install Docker Desktop on Windows

Estimated reading time: 6 minutes

Docker Desktop for Windows is the [Community](#) version of Docker for Microsoft Windows. You can download Docker Desktop for Windows from Docker Hub.

[Download from Docker Hub](#)

Explore Docker Desktop for Windows



Docker Desktop for Windows

By [Docker](#)

The fastest and easiest way to get started with Docker on Windows

Edition Windows x86-64

Get Docker Desktop for Windows

Docker Desktop for Windows is available for free.

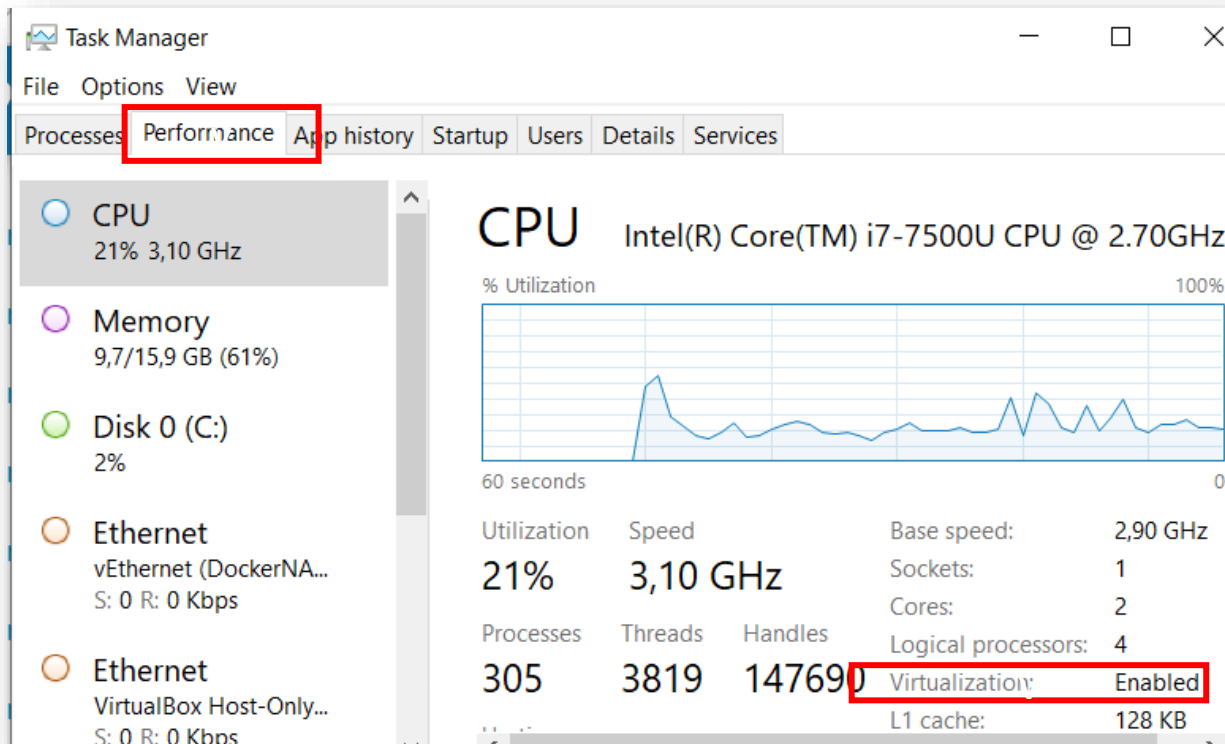
Requires Microsoft Windows 10 Professional or Enterprise 64-bit. For previous versions get [Docker Toolbox](#).

By downloading this, you agree to the terms of the [Docker Software End User License Agreement](#) and the [Docker Data Processing Agreement \(DPA\)](#).

[Get Docker](#)

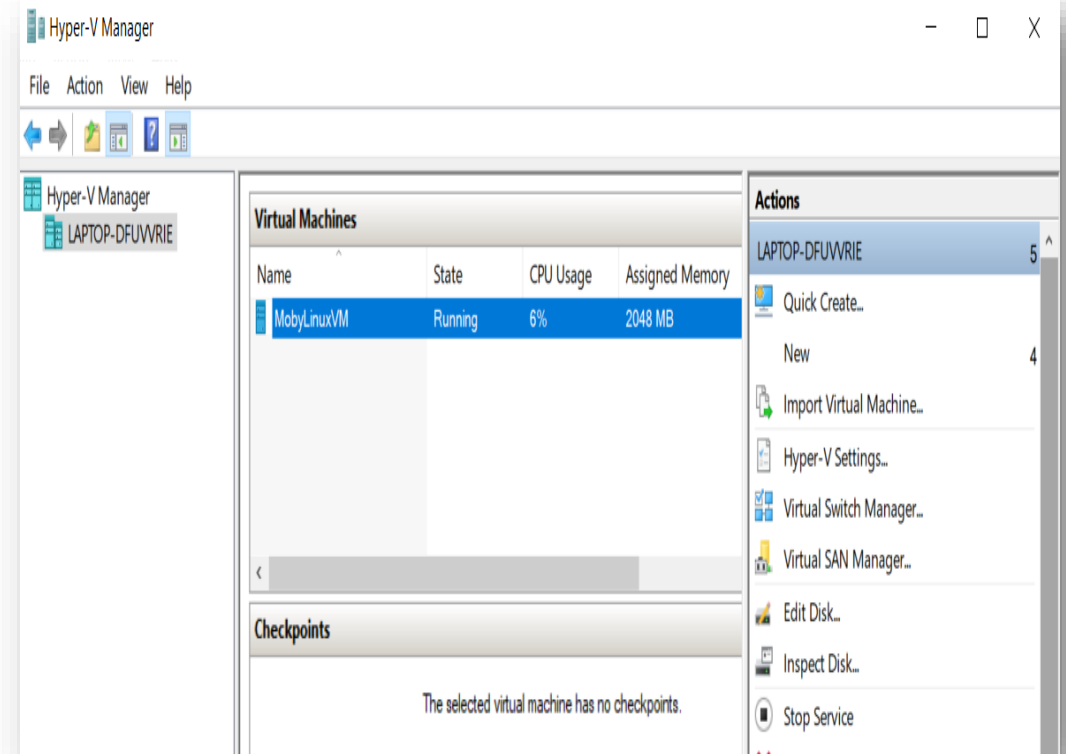
installation Docker Desktop pour Windows 10 Pro

Si problème voir : <https://docs.docker.com/docker-for-windows/troubleshoot/#virtualization-must-be-enabled>

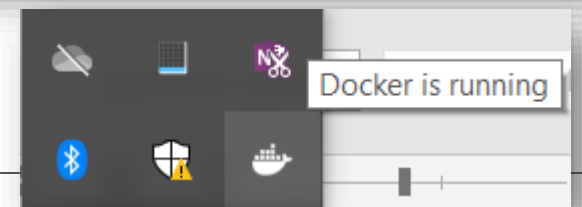


The screenshot shows the Windows Task Manager Performance tab. The 'Performance' tab is highlighted with a red box. The CPU section is selected, showing 'CPU Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz' with 21% utilization. A graph shows the utilization over 60 seconds. Below the graph, a table lists system specifications:

Utilization	Speed	Base speed:	2,90 GHz	
21%	3,10 GHz	Sockets:	1	
Processes	Threads	Handles	Cores:	2
305	3819	14769	Logical processors:	4
			Virtualization:	Enabled
			L1 cache:	128 KB



The screenshot shows the Hyper-V Manager window. The 'Virtual Machines' table lists a VM named 'MobyLinuxVM' which is in a 'Running' state with 6% CPU usage and 2048 MB of assigned memory. The 'Actions' pane on the right shows various options for the selected VM, including 'Quick Create...', 'New', 'Import Virtual Machine...', 'Hyper-V Settings...', 'Virtual Switch Manager...', 'Virtual SAN Manager...', 'Edit Disk...', 'Inspect Disk...', and 'Stop Service'.



The screenshot shows the Windows system tray. The Docker Desktop icon is visible, and a notification bubble next to it says 'Docker is running'.

Vérification installation / config

```
$ docker --version
```

```
C:\Users\adelej>docker --version  
Docker version 18.06.1-ce, build e68fc7a
```

```
$ docker info
```

```
C:\Users\adelej>docker info  
Containers: 88  
  Running: 36  
  Paused: 0  
  Stopped: 52  
Images: 35  
Server Version: 18.06.1-ce
```

```
Operating System: Docker for Windows  
OSType: linux  
Architecture: x86_64  
CPUs: 2  
Total Memory: 1.952GiB  
Name: linuxkit-00155d00c101
```

Vérification installation / config

```
$ docker pull hello-world
```

```
Using default tag: latest
latest: Pulling from library/hello-world
Digest: sha256:fc6a51919cfef2e6763f62b6d9e8815acbf7cd2e476ea353743570610737b752
Status: Image is up to date for hello-world:latest
```

```
$ 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/
```

installation Docker pour linux

docs.docker.com/install/linux/docker-ce/ubuntu/

docker docs Search the docs Guides Product manuals Glossary Reference Samples

Docker Engine

- Overview
- Linux
 - CentOS
 - Debian
 - Fedora
 - Ubuntu**
 - Binaries
 - Optional Linux post-installation steps
- Release notes

Docker Assemble

Docker App

Get Docker Engine - Community for Ubuntu

Estimated reading time: 12 minutes

To get started with Docker Engine - Community on Ubuntu, make sure you meet the prerequisites, then install Docker.

Prerequisites

Docker EE customers

To install Docker Enterprise Edition (Docker EE), go to [Get Docker EE for Ubuntu](#) instead of this topic.

To learn more about Docker EE, see [Docker Enterprise Edition](#).

OS requirements

To install Docker Engine - Community, you need the 64-bit version of one of these Ubuntu versions:

- Eoan 19.10
- Bionic 18.04 (LTS)
- Xenial 16.04 (LTS)

```
vagrant@ubuntu-bionic:~$ cat /etc/*release*
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=18.04
DISTRIB_CODENAME=bionic
DISTRIB_DESCRIPTION="Ubuntu 18.04.2 LTS"
NAME="Ubuntu"
VERSION="18.04.2 LTS (Bionic Beaver)"
ID=ubuntu
```

Uninstall old versions

Older versions of Docker were called `docker`, `docker.io`, or `docker-engine`.

```
$ sudo apt-get remove docker docker-engine docker.io containerd runc
```

```
vagrant@ubuntu-bionic:~$ sudo apt-get remove docker docker-engine docker.io containerd runc
Reading package lists... Done
Building dependency tree
Reading state information... Done

No apt package "docker", but there is a snap with that name.
Try "snap install docker"

E: Unable to locate package docker
E: Unable to locate package docker-engine
E: Unable to locate package docker.io
E: Couldn't find any package by glob 'docker.io'
E: Couldn't find any package by regex 'docker.io'
E: Unable to locate package containerd
E: Unable to locate package runc
```

installation Docker pour linux

Install using the convenience script

Docker provides convenience scripts at get.docker.com and test Docker Engine - Community into development environments quickly and easily. These scripts are available in the [docker-install repository](#). Using these scripts is not recommended without understanding the potential risks before you use them:

- The scripts require `root` or `sudo` privileges to run. They run as root when running them.
- The scripts attempt to detect your Linux distribution and install the appropriate version. In addition, the scripts do not allow you to customize any configuration, either from Docker's point of view or from your system's point of view.
- The scripts install all dependencies and recommendations. This means they may install a large number of packages, depending on the system.
- The script does not provide options to specify which version to install. The version released in the "edge" channel.
- Do not use the convenience script if Docker has already been installed.

This example uses the script at get.docker.com to install the latest testing version, use test.docker.com instead. In each of the following examples, the `get-docker.sh` script is used to install Docker.

Warning:

Always examine scripts downloaded from the internet before running them.

```
$ curl -fsSL https://get.docker.com -o get-docker.sh
$ sudo sh get-docker.sh
```

```
vagrant@ubuntu-bionic:~$ curl -fsSL https://get.docker.com -o get-docker.sh
vagrant@ubuntu-bionic:~$
vagrant@ubuntu-bionic:~$ ls
get-docker.sh
vagrant@ubuntu-bionic:~$
vagrant@ubuntu-bionic:~$ sudo sh get-docker.sh
# Executing docker install script, commit: 6bf300318ebaab958c4adc341a8c7bb9f3a54a1a
+ sh -c apt-get update -qq >/dev/null
+ sh -c apt-get install -y -qq apt-transport-https ca-certificates curl >/dev/null
+ sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" | apt-key add -qq - >/dev/null
```

```
vagrant@ubuntu-bionic:~$ sudo docker version
Client: Docker Engine - Community
Version:          19.03.1
API version:      1.40
Go version:       go1.12.5
Git commit:       74b1e89
Built:            Thu Jul 25 21:21:05 2019
OS/Arch:          linux/amd64
Experimental:     false
```

How to use this image

```
$ docker run docker/whalesay cowsay boo
```

hub.docker.com/search?q=whalesay&type=image

The screenshot shows the Docker Hub search interface. The search bar contains the text 'whalesay'. Below the search bar, there are several search results listed. The first result is 'Community (1646)'. The second result is 'docker/whalesay'. The third result is 'ox0spy/whalesay-fortune'. The Docker Hub logo is visible in the top left corner of the search results area.

```
$ sudo docker run docker/whalesay cowsay Hello-World!
```

installation Docker pour linux

The screenshot shows the Docker documentation website at `docs.docker.com/install/linux/docker-ce/ubuntu/`. The page title is "Install Docker Engine - Community". The main content area lists three installation methods:

- Most users [set up Docker's repositories](#) and install from them, for ease of installation and the recommended approach.
- Some users download the DEB package and [install it manually](#) and manage upgrades complex situations such as installing Docker on air-gapped systems with no access to the internet.
- In testing and development environments, some users choose to use automated [convenience scripts](#).

The section "Install using the repository" is highlighted with a red box. Below it, the text reads: "Before you install Docker Engine - Community for the first time on a new host machine, you need to set up the repository. Afterward, you can install and update Docker from the repository."

SET UP THE REPOSITORY

1. Update the `apt` package index:

COMMANDES SIMPLES

docker run

```
$ docker run nginx
```

```
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
bc51dd8edc1b: Pull complete
66ba67045f57: Pull complete
bf317aa10aa5: Pull complete
Digest: sha256:ad5552c786f128e389a0263104ae39f3d3c7895579d45ae716f528185b36bc6f
Status: Downloaded newer image for nginx:latest
```

```
$ docker run redis
```

```
Unable to find image 'redis:latest' locally
latest: Pulling from library/redis
bc51dd8edc1b: Already exists
37d80eb324ee: Pull complete
392b7748dfaf: Pull complete
48df82c3534d: Pull complete
2ec2bb0b4b0e: Pull complete
1302bce0b2cb: Pull complete
Digest: sha256:7b84b346c01e5a8d204a5bb30d4521bcc3a8535bbf90c660b8595fad248eae82
Status: Downloaded newer image for redis:latest
1:C 25 Feb 2020 20:55:13.871 # o000o000o000o Redis is starting o000o000o000o
1:C 25 Feb 2020 20:55:13.871 # Redis version=5.0.7, bits=64, commit=00000000, modified=0, pid=1, just started
1:C 25 Feb 2020 20:55:13.871 # Warning: no config file specified, using the default config. In order to specify a config file use redis-server /path/to/redis.conf
1:M 25 Feb 2020 20:55:13.873 * Running mode=standalone, port=6379.
1:M 25 Feb 2020 20:55:13.873 # WARNING: The TCP backlog setting of 511 cannot be enforced because /proc/sys/net/core/somaxconn is set to the lower value of 128.
1:M 25 Feb 2020 20:55:13.873 # Server initialized
1:M 25 Feb 2020 20:55:13.873 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will create latency and memory usage issues with Redis. To fix this issue run the command 'echo never > /sys/kernel/mm/transparent_hugepage/enabled' as root, and add it to your /etc/rc.local in order to retain the setting after a reboot. Redis must be restarted after THP is disabled.
```

ps – list containers

```
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
59402985aedd	redis	"docker-entrypoint.s..."	About a minute ago	Up About a minute	6379/tcp	sharp_davinci
d46d396f1b48	nginx	"nginx -g 'daemon of..."	7 minutes ago	Up 7 minutes	80/tcp	sharp_curie

```
$ docker run ubuntu
```

```
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
59402985aedd	redis	"docker-entrypoint.s..."	12 minutes ago	Up 12 minutes	6379/tcp	sharp_davinci
d46d396f1b48	nginx	"nginx -g 'daemon of..."	18 minutes ago	Up 18 minutes	80/tcp	sharp_curie

```
$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
bb0a8daedd1b	ubuntu	"/bin/bash"	7 minutes ago	Exited (0) 7 minutes ago		determined_hawking
59402985aedd	redis	"docker-entrypoint.s..."	13 minutes ago	Up 13 minutes	6379/tcp	sharp_davinci
d46d396f1b48	nginx	"nginx -g 'daemon of..."	19 minutes ago	Up 19 minutes	80/tcp	sharp_curie

Option -it

```
$ docker run --name ubuntu-sans-it ubuntu
```



Le conteneur est inaccessible et en état inactif vu que la CMD dans l'image est "CMD bin/bash". Cette commande n'associe pas des STDIN/STDOUT aux canaux d'E/S du conteneur.

Pour pouvoir y accéder, il faut utiliser l'option -it qui établie cette association (plus de detail après dans le cour)

```
$ docker ps -a
```

```
C:\Users\adelelj>docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
fadd7ee1af13	ubuntu	"/bin/bash"	6 seconds ago	Exited (0) 5 seconds ago		ubuntu-sans-it

```
$ docker run --name ubuntu-avec-it -it ubuntu
```

```
C:\Users\adelelj>docker run --name ubuntu-avec-it -it ubuntu
root@e74e7f81e530:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt
```

STOP – stoper un container

```
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
59402985aedd	redis	"docker-entrypoint.s..."	14 minutes ago	Up 14 minutes	6379/tcp	sharp_davinci
d46d396f1b48	nginx	"nginx -g 'daemon of..."	20 minutes ago	Up 20 minutes	80/tcp	sharp_curie

```
$ docker stop sharp_davinci
```

```
sharp_davinci
```

```
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
d46d396f1b48	nginx	"nginx -g 'daemon of..."	23 minutes ago	Up 23 minutes	80/tcp	sharp_curie

```
$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
bb0a8daedd1b	ubuntu	"/bin/bash"	12 minutes ago	Exited (0) 12 minutes ago		determined_hawking
59402985aedd	redis	"docker-entrypoint.s..."	19 minutes ago	Exited (0) 2 minutes ago		sharp_davinci
d46d396f1b48	nginx	"nginx -g 'daemon of..."	25 minutes ago	Up 24 minutes	80/tcp	sharp_curie

Rm – supprimer (Remove) un container

```
$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
bb0a8daedd1b	ubuntu	"/bin/bash"	45 minutes ago	Exited (0) 45 minutes ago		determined_hawking
59402985aedd	redis	"docker-entrypoint.s..."	About an hour ago	Exited (0) 34 minutes ago		sharp_davinci
d46d396f1b48	nginx	"nginx -g 'daemon of..."	About an hour ago	Up About an hour	80/tcp	sharp_curie

```
$ docker rm sharp_davinci
```

```
sharp_davinci
```

```
$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
bb0a8daedd1b	ubuntu	"/bin/bash"	About an hour ago	Exited (0) About an hour ago		determined_hawking
d46d396f1b48	nginx	"nginx -g 'daemon of..."	About an hour ago	Up About an hour	80/tcp	sharp_curie

images – List images

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
nginx	latest	2073e0bcb60e	3 weeks ago	127MB
redis	latest	44d36d2c2374	3 weeks ago	98.2MB
aelj/bonjourdocker	1.0	9174da1b2242	4 weeks ago	189MB
ubuntu	18.04	ccc6e87d482b	5 weeks ago	64.2MB
ubuntu	latest	ccc6e87d482b	5 weeks ago	64.2MB
alpine	3.5	f80194ae2e0c	13 months ago	4MB
hello-world	latest	fce289e99eb9	14 months ago	1.84kB
docker/kube-compose-controller	v0.3.9	16260a912a7b	20 months ago	29.9MB
docker/kube-compose-api-server	v0.3.9	8757267f7b06	20 months ago	42.5MB
k8s.gcr.io/kube-proxy-amd64	v1.10.3	4261d315109d	21 months ago	97.1MB
k8s.gcr.io/kube-scheduler-amd64	v1.10.3	353b8f1d102e	21 months ago	50.4MB
k8s.gcr.io/kube-apiserver-amd64	v1.10.3	e027465e22e2	21 months ago	225MB

rmi – Remove images

```
$ docker rmi redis
```

```
Untagged: redis:latest
Untagged: redis@sha256:7b84b346c01e5a8d204a5bb30d4521bcc3a8535bbf90c660b8595fad248eae82
Deleted: sha256:44d36d2c2374b240abcf5da2130abf49938b8fb49446df6eec028718520332ef
Deleted: sha256:98085db137f7660420590b6a2cbf6214835a5ce192baa72c733de82302d717e4
Deleted: sha256:fb9fc76ceac433a87846595e2a45a3cca3e1dc281f9fc3e64c939e6fb7b440a1
Deleted: sha256:a89092719dcd0771b5e1c434f0ccea803852a5d2f3eac5522301ad5a00f5a10
Deleted: sha256:f4cdcef89dce64d58cfb9a0cb544e5ada843c8b6d90d92f625675f1960e46d7b
Deleted: sha256:31a81e216e79c54ea97523674895bf12d76e1b8cbb20e2ca1ced60925744a161
```

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
nginx	latest	2073e0bcb60e	3 weeks ago	127MB
aelj/bonjourdocker	1.0	9174da1b2242	4 weeks ago	189MB
ubuntu	18.04	ccc6e87d482b	5 weeks ago	64.2MB
ubuntu	latest	ccc6e87d482b	5 weeks ago	64.2MB
alpine	3.5	f80194ae2e0c	13 months ago	4MB
hello-world	latest	fce289e99eb9	14 months ago	1.84kB
docker/kube-compose-controller	v0.3.9	16260a912a7b	20 months ago	29.9MB
docker/kube-compose-api-server	v0.3.9	8757267f7b06	20 months ago	42.5MB
k8s.gcr.io/kube-proxy-amd64	v1.10.3	4261d315109d	21 months ago	97.1MB
k8s.gcr.io/kube-controller-manager-amd64	v1.10.3	40c8d10b2d11	21 months ago	148MB
k8s.gcr.io/kube-apiserver-amd64	v1.10.3	e03746fe22c3	21 months ago	225MB

Pull – télécharger une image (toupdate)

```
$ docker run nginx
```

```
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
fc7181108d40: Already exists
d2e987ca2267: Pull complete
0b760b431b11: Pull complete
Digest:
sha256:96fb261b66270b900ea5a2c17a26abbfabe95506e73c3a3c65869a6dbe83223a
Status: Downloaded newer image for nginx:latest
```

```
$ docker pull nginx
```

```
Using default tag: latest
latest: Pulling from library/nginx
fc7181108d40: Pull complete
d2e987ca2267: Pull complete
0b760b431b11: Pull complete
Digest:
sha256:96fb261b66270b900ea5a2c17a26abbfabe95506e73c3a3c65869a6dbe83223a
Status: Downloaded newer image for nginx:latest
```

Ajouter une commande

```
$ docker run ubuntu sleep 5
```



```
$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
5f40387da806	ubuntu	"sleep 5"	9 seconds ago	Exited (0) 3 seconds ago		elastic_banach
f2b198773c7d	ubuntu	"/bin/bash"	32 minutes ago	Exited (0) 32 minutes ago		kind_noether

Run – attach and detach

```
$ docker run -d --name topdemo ubuntu /usr/bin/top -b
```

```
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
306d207d6dfb	ubuntu	"/usr/bin/top -b"	21 seconds ago	Up 20 seconds		topdemo

```
$ docker attach topdemo
```

```
top - 15:01:59 up 15:56, 0 users, load average: 0.77, 0.58, 0.56
Tasks: 1 total, 1 running, 0 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.7 us, 2.5 sy, 0.0 ni, 93.2 id, 1.0 wa, 0.0 hi, 0.5 si, 0.0 st
KiB Mem : 2046812 total, 638084 free, 856304 used, 552424 buff/cache
KiB Swap: 1048572 total, 900740 free, 147832 used. 1037088 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	36480	3056	2700	R	0.0	0.1	0:00.06	top

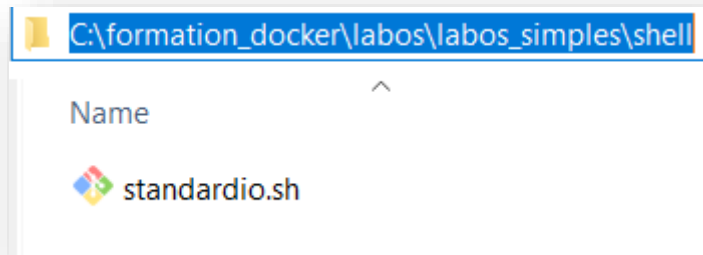
cp – copie de fichiers

HOST → Container

```
$ docker cp PATH_IN_HOST CONTAINER_NAME:PATH_IN_CONTAINER
```

```
$ docker cp c:/formation_docker/labos/labos_simples/shell/standardio.sh shell_simple_demo:/shell/standardio.sh
```

Host



Container

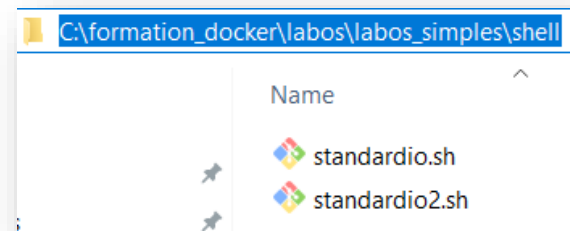
```
root@d2fc5a06da2a:/shell# ls .
standardio.sh
```

Container → Host

```
$ docker cp CONTAINER_NAME:PATH_IN_CONTAINER PATH_IN_HOST
```

```
$ docker cp shell_simple_demo:/shell/standardio.sh c:/formation_docker/labos/labos_simples/shell/standardio2.sh
```

Host



Exec – executer une commande

```
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
d2fc5a06da2a	ubuntu	"bash"	37 minutes ago	Up 37 minutes		ubuntu_bash

```
$ docker exec ubuntu_bash cat /etc/hosts
```

```
127.0.0.1      localhost
::1           localhost ip6-localhost ip6-loopback
fe00::0       ip6-localnet
ff00::0       ip6-mcastprefix
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
172.17.0.2    d2fc5a06da2a
```

```
$ docker exec --workdir /usr/src ubuntu_last mkdir adel
```

```
$ docker exec ubuntu_bash rm -rf /shell/standardio.sh
```

docker run

RAPPEL : Workflow pour docker run

