



Liste de commandes

Sources : <https://docs.docker.com/engine/reference/commandline/cli/>
 <https://docs.docker.com/engine/installation/linux/centos/>
 <http://linuxfr.org/news/docker-tutoriel-pour-manipuler-les-conteneurs>

1 Qu'est-ce que Docker ?

Application client/Serveur

Système qui regroupe ou enveloppe codes sources, applications et dépendances dans un paquet ou objet unique Conteneur (Container en Anglais)

Sert à exécuter des applications

Ce conteneur peut être exécuté sur n'importe quel serveur (principalement des Linux)

On peut utiliser plusieurs conteneurs dans une machine Hôte et chaque conteneur s'exécute dans un environnement isolé.

A compléter....

2 Installer et vérifier Docker sous Centos

Pré-requis:

Pour utiliser Docker, Il faut connaître les commandes de base Linux.

Pour l'installer il faut un système **64 bits** avec un noyau minimum **3.10...**

Vérifier la version du noyau :

```
# uname -r ou # uname -a
```

Mettre à jour votre système :

```
# yum update
```

Installer Docker :

```
# yum -y install docker
```

Vérifier son état :

```
# systemctl status docker
```

Démarrer docker :

```
# systemctl start docker
```

Activer le démarrage automatique :

```
# systemctl enable docker
```

Afficher les informations sur docker :

```
# docker info
```

Afficher sa version :

```
# docker version
```

Obtenir de l'aide :

```
# docker --help ou # docker
```

Tester et vérifier l'installation de docker en exécutant l'image test hello-world dans un conteneur :

N'oubliez pas de se connecter à internet

```
# docker run hello-world
```

```
[root@SrvCentOS7-2 ~]# 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.
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 Hub account:
https://hub.docker.com

For more examples and ideas, visit:
https://docs.docker.com/engine/userguide/

Usage of loopback devices is strongly discouraged for production use. Either use `--storage-opt dm.thin|
oop_devices=true` to suppress this warning.
```

Ce résultat vérifie l'installation, explique les étapes (Client /Serveur) de création et d'exécution de l'image hello-world dans un conteneur et à partir de quel site (Docker Hub) l'image a été récupérée.

Désinstaller docker :

```
# yum -y remove docker
```

3 Utiliser Docker sous Centos

Une fois Docker est installé sous votre Centos (**Hôte Docker**), on peut exécuter des commandes à partir d'un terminal (**Client Docker**) pour gérer les images et exécuter les conteneurs.

Conteneur = Image en cours d'exécution

On peut, à tout moment, avoir des informations détaillées sur une commande :

```
# docker <commande> --help
```

Chaque commande peut avoir des options supplémentaires

Voici les commandes fréquemment utilisées :

Rechercher une image à partir d'un Docker Hub :

```
# docker search <nom image>
```

Exemple :

```
# docker search centos :7 → Recherche une image centos 7 dans Docker Hub.
```

```
[root@SrvCentOS7-2 ~]# docker search centos:7
```

INDEX	NAME	DESCRIPTION
D		
docker.io	docker.io/million12/varnish	Varnish 4.x on top of CentOS-7, highly cus...
docker.io	docker.io/chef/centos-7	A CentOS 7 Docker Image with Chef Containe...
docker.io	docker.io/mikayel/centos-6-7	update centos 6.7
docker.io	docker.io/pacur/centos-7	Pacur CentOS 7
docker.io	docker.io/alanfranz/fwd-centos-7	
docker.io	docker.io/clusterhq/fpm-centos-7	
docker.io	docker.io/dhoppe/chef-centos-7	
docker.io	docker.io/ezamriy/centos-7-mongodb	A standard CentOS 7 container with Chef 12...
docker.io	docker.io/ezamriy/centos-7-vagrant-supervisor-ssh	CentOS 7 container with MongoDB 3.2 instal...
docker.io	docker.io/imofftoseethewizard/centos-7-julia-0.3	CentOS 7 vagrant friendly container with s...
docker.io	docker.io/jego/centos-7	
docker.io	docker.io/lloydbenson/docker-centos-7-node	centos 7 node container
docker.io	docker.io/lloydbenson/docker-centos-7-sshd	docker centos 7 sshd

Télécharger une image depuis Docker Hub :

```
# docker pull <nom image>
```

Déposer une image dans Docker Hub :

```
# docker push <nom image>
```

Afficher la liste des images (Il y a plusieurs options)

```
# docker images
```

```
[root@SrvCentOS7-2 ~]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE
docker.io/hello-world	latest	4c4abd6d4278	2 weeks ago	967 B
docker.io/centos	latest	28e524afdd05	5 weeks ago	196.7 MB
docker.io/tutum/wordpress	latest	20c52a34e232	7 weeks ago	476.4 MB

Exécuter une commande dans un conteneur :

```
# docker run <nom image>
```

Exemple 1:

```
# docker run centos echo "BONJOUR"
```

```
→ [root@SrvCentOS7-2 ~]# docker run centos echo "BONJOUR"
Usage of loopback devices is strongly discouraged for production use.
  oop_devices=true` to suppress this warning.
→ BONJOUR
→ [root@SrvCentOS7-2 ~]# █
```

Exemple 2: Exécution Interactive (-i) depuis un terminal (-t) de l'interpréteur de commande bash

Exemple 1:

```
# docker run -it centos bash
```

```
[root@SrvCentOS7-2 ~]# docker run -it centos bash
Usage of loopback devices is strongly discouraged for production use.
oop_devices=true` to suppress this warning.
[root@74c403a71acf /]#
```

On peut vérifier que le conteneur est en cours d'exécution :

```
[root@SrvCentOS7-2 ~]# docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
74c403a71acf   centos     "bash"                  16 seconds ago Up 15 seconds 0.0.0.0:80->80/t
471eb3ea655d   tutum/wordpress "/run.sh"              3 days ago    Up 3 days    0.0.0.0:80->80/t
[root@SrvCentOS7-2 ~]#
```

L'exécution du conteneur s'arrête si on tape la commande **exit**

```
[root@74c403a71acf /]#
[root@74c403a71acf /]# exit
exit
[root@SrvCentOS7-2 ~]#
```

Vérification :

```
[root@SrvCentOS7-2 ~]# docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
471eb3ea655d   tutum/wordpress "/run.sh"              3 days ago    Up 3 days    0.0.0.0:80->80/tcr
[root@SrvCentOS7-2 ~]#
```

On peut exécuter un conteneur en arrière-plan avec l'option **-d** :

```
[root@SrvCentOS7-2 ~]# docker run -itd centos bash
cadbe32008bfa3454ddf898dd9cefb157c72c8aaffc22074214b430b99dfafa4
Usage of loopback devices is strongly discouraged for production use.
oop_devices=true` to suppress this warning.
[root@SrvCentOS7-2 ~]# docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
cadbe32008bf   centos     "bash"                  About a minute ago Up About a minute 0.0.0.0:80->80/
471eb3ea655d   tutum/wordpress "/run.sh"              3 days ago    Up 3 days    0.0.0.0:80->80/
[root@SrvCentOS7-2 ~]#
```

On peut arrêter l'exécution, dans ce cas, avec la commande :

```
# docker kill <Identificateur du Conteneur>
```

```
[root@SrvCentOS7-2 ~]# docker kill ca
ca
[root@SrvCentOS7-2 ~]#
```

Vérification :

```
[root@SrvCentOS7-2 ~]# docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
471eb3ea655d   tutum/wordpress "/run.sh"              3 days ago    Up 3 days    0.0.0.0:80->80/t
[root@SrvCentOS7-2 ~]#
```

Il est possible de faire des modifications dans un conteneur et de voir les différences avec l'image de base par la commande suivante :

```
# docker diff <Identificateur du Conteneur>
```

Exemple :

Créer un dossier dans le conteneur centos :

```
[root@86defd9bf815 ~]# mkdir MonDossierTest
[root@86defd9bf815 ~]#
```

```
[root@SrvCentOS7-2 ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
86defd9bf815	centos	"bash"	7 minutes ago	Up 7 minutes
471eb3ea655d	tutum/wordpress	"/run.sh"	3 days ago	Up 3 days

```
[root@SrvCentOS7-2 ~]#
```

Voir la différence par rapport à l'image de base :

```
# docker diff 86 (Pas besoin de mettre tout l'ID s'il est unique)
```

```
[root@SrvCentOS7-2 ~]# docker diff 86
C /run
A /run/secrets
→A /MonDossierTest
[root@SrvCentOS7-2 ~]#
```

Maintenant, si on arrête l'exécution de notre conteneur, les modifications ne seront pas enregistrées dans l'image. Pour que les modifications soient effectives et persistantes, il faut reconstruire une autre image avec la commande :

```
# docker commit <Identificateur du Conteneur> <Nom de l'image>
```

Exemple :

```
[root@SrvCentOS7-2 ~]# docker commit 3a msabir/centos_modif
5923bcd902b2778ce73c94a4081a69825b9ca9db51ff17bf46d01a04b7a729c1
[root@SrvCentOS7-2 ~]#
```

Vérification :

```
[root@SrvCentOS7-2 ~]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE
msabir/centos_modif	latest	5923bcd902b2	2 minutes ago	196.7 MB
docker.io/hello-world	latest	4c4abd6d4278	2 weeks ago	967 B
docker.io/centos	latest	28e524afdd05	5 weeks ago	196.7 MB
docker.io/tutum/wordpress	latest	20c52a34e232	8 weeks ago	476.4 MB

```
[root@SrvCentOS7-2 ~]#
```

D'autres commandes existent pour manipuler les images.

4Utiliser Dockerfile

Jusqu'à maintenant, on a exécuté des commandes manuellement pour manipuler des images. Ces commandes peuvent être regroupées dans un fichier Docker (Dockerfile) en vue de les exécuter automatiquement.

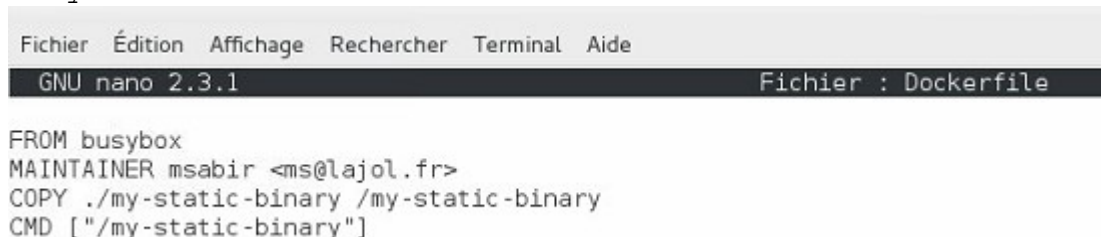
Voici les principales instructions ou directives qu'on peut utiliser dans un Dockerfile :

- **MAINTAINER** : Nom et courriel de mainteneur du conteneur;
- **FROM** : Image de base (Ubuntu, Debian, CentOS, etc...);
- **RUN** : Commande à exécuter pour installer le conteneur;
- **ENTRYPOINT** : Commande qui s'exécute au démarrage du conteneur (une seule sera exécutée);
- **CMD** : Commande qui s'exécute au démarrage du conteneur;
- **VOLUME** : Point de montage;
- **ADD** : Copier un fichier du répertoire courant dans le système de fichiers du conteneur;
- **USER** : Utilisateur qui exécute les commandes dans le conteneur;
- **EXPOSE** : Port(s) à exposer à l'extérieur.

Vidéo à regarder :

<https://www.grafikart.fr/tutoriels/docker/dockerfile-636>

Exemple 1: Construire une image d'un tout petit système Linux BusyBox.



```
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
GNU nano 2.3.1                                Fichier : Dockerfile

FROM busybox
MAINTAINER msabir <ms@lajol.fr>
COPY ./my-static-binary /my-static-binary
CMD ["/my-static-binary"]
```



```
[root@SrvCentOS7-2 ~]# docker build -t msabir/testDockerfile .
repository name component must match "[a-z0-9](?:-[a-z0-9])*(?:[._][a-z0-9](?:-[a-z0-9])*)*"
[root@SrvCentOS7-2 ~]# docker build -t msabir/test .
Sending build context to Docker daemon 13.51 MB
Step 1 : FROM busybox
Trying to pull repository docker.io/library/busybox ... latest: Pulling from library/busybox
4b51ded9aed1: Pull complete
307ac631f1b5: Pull complete
Digest: sha256:4a887a2326ec9e0fa90cce7b4764b0e627b5d6afcb81a3f73c85dc29cea00048
Status: Downloaded newer image for docker.io/busybox:latest

---> 307ac631f1b5
Step 2 : MAINTAINER msabir <ms@lajol.fr>
---> Running in 73ea2ee49cdf
---> c42a694b33a5
Removing intermediate container 73ea2ee49cdf
Step 3 : COPY ./my-static-binary /my-static-binary
lsstat my-static-binary: no such file or directory

[root@SrvCentOS7-2 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             VIRTUAL SIZE
<none>              <none>             c42a694b33a5       15 minutes ago     1.113 MB
msabir/centos_modif latest             5923bcd902b2       About an hour ago  196.7 MB
docker.io/hello-world latest             4c4abd6d4278       2 weeks ago        967 B
docker.io/centos     latest            28e524afdd05       5 weeks ago        196.7 MB
docker.io/busybox    latest            307ac631f1b5       7 weeks ago        1.113 MB
docker.io/tutum/wordpress latest            20c52a34e232       8 weeks ago        476.4 MB
[root@SrvCentOS7-2 ~]#
```

Exemple 2 : Créer un Dockerfile pour construire une image d'un serveur Web sous CentOS 7.

1. Créer un fichier Dockerfile avec un éditeur, non par exemple.

```
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
GNU nano 2.3.1                                Fichier : Dockerfile

FROM centos
MAINTAINER msabir <ms@lajol.fr>
RUN yum -y install httpd
RUN echo "BONJOUR ! Mon premier DockerFile" > /var/www/html/index.html
EXPOSE 80
CMD ["-D", "FOREGROUND"]
ENTRYPOINT ["/usr/sbin/httpd"]
```

Construire votre image avec la commande docker build:

```
# docker build -t <Nom de l'image> : tag
```

Dans notre cas :

```
# docker build -t centweb :latest .
```

Note commande construit une image nommée centweb avec la dernière version de centos à partir du fichier Dockerfile qui se trouve dans le répertoire courant (représenté par le point)

```
[root@SrvCentOS7-2 DockerWeb]# docker build -t centweb:latest .
Sending build context to Docker daemon 2.048 kB
Step 1 : FROM centos
--> 28e524afdd05
Step 2 : MAINTAINER msabir <ms@lajol.fr>
--> Running in c425a0378d13
--> e09a9a444a79
Removing intermediate container c425a0378d13
Step 3 : RUN yum -y install httpd
--> Running in 9180cd73ff3b
Loaded plugins: fastestmirror, ovl
Determining fastest mirrors
 * base: mirror.w2k.ch
 * extras: mirrors.atosworldline.com
 * updates: distrib-coffee.ipsl.jussieu.fr
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.6-40.el7.centos.1 will be installed
--> Processing Dependency: httpd-tools = 2.4.6-40.el7.centos.1 for package: httpd-2.4.6-40.el7.centos.1.x86_64
--> Processing Dependency: system-logos >= 7.92.1-1 for package: httpd-2.4.6-40.el7.centos.1.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.6-40.el7.centos.1.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.6-40.el7.centos.1.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.6-40.el7.centos.1.x86_64
--> Running transaction check
--> Package apr.x86_64 0:1.4.8-3.el7 will be installed
--> Package apr-util.x86_64 0:1.5.2-6.el7 will be installed
--> Package centos-logos.noarch 0:70.0.6-3.el7.centos will be installed
--> Package httpd-tools.x86_64 0:2.4.6-40.el7.centos.1 will be installed
--> Package mailcap.noarch 0:2.1.41-2.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
httpd x86_64 2.4.6-40.el7.centos.1 updates 2.7 M
Installing for dependencies:
apr x86_64 1.4.8-3.el7 base 103 k
apr-util x86_64 1.5.2-6.el7 base 92 k
centos-logos noarch 70.0.6-3.el7.centos base 21 M
httpd-tools x86_64 2.4.6-40.el7.centos.1 updates 82 k
mailcap noarch 2.1.41-2.el7 base 31 k
=====

Transaction Summary
=====
Install 1 Package (+5 Dependent packages)

Total download size: 24 M
Installed size: 31 M
Downloading packages:
warning: /var/cache/yum/x86_64/7/base/packages/apr-util-1.5.2-6.el7.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID f
4a80eb5: NOKEY
Public key for apr-util-1.5.2-6.el7.x86_64.rpm is not installed
Public key for httpd-tools-2.4.6-40.el7.centos.1.x86_64.rpm is not installed
http://distrib-coffee.ipsl.jussieu.fr/pub/linux/centos/7.2.1511/updates/x86_64/Packages/httpd-2.4.6-40.el7.centos.1.x86_
64.rpm: [Errno -1] Package does not match intended download. Suggestion: run yum --enablerepo=updates clean metadata
Trying other mirror.
http://centos.quelquesmots.fr/7.2.1511/updates/x86_64/Packages/httpd-2.4.6-40.el7.centos.1.x86_64.rpm: [Errno -1] Packag
e does not match intended download. Suggestion: run yum --enablerepo=updates clean metadata
Trying other mirror.
-----
Total 666 kB/s | 24 MB 00:37
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Importing GPG key 0xF4A80EB5:
Userid : "CentOS-7 Key (CentOS 7 Official Signing Key) <security@centos.org>"
Fingerprint: 6341 ab27 53d7 8a78 a7c2 7bb1 24c6 a8a7 f4a8 0eb5
Package : centos-release-7-2.1511.el7.centos.2.10.x86_64 (@CentOS)
From : /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : apr-1.4.8-3.el7.x86_64 1/6
Installing : apr-util-1.5.2-6.el7.x86_64 2/6
Installing : httpd-tools-2.4.6-40.el7.centos.1.x86_64 3/6
Installing : centos-logos-70.0.6-3.el7.centos.noarch 4/6
Installing : mailcap-2.1.41-2.el7.noarch 5/6
```



```

Verifying : httpd-2.4.6-40.el7.centos.1.x86_64 3/6
Verifying : apr-util-1.5.2-6.el7.x86_64 4/6
Verifying : httpd-tools-2.4.6-40.el7.centos.1.x86_64 5/6
Verifying : centos-logos-70.0.6-3.el7.centos.noarch 6/6

```

```

Installed:
  httpd.x86_64 0:2.4.6-40.el7.centos.1

```

```

Dependency Installed:
  apr.x86_64 0:1.4.8-3.el7
  apr-util.x86_64 0:1.5.2-6.el7
  centos-logos.noarch 0:70.0.6-3.el7.centos
  httpd-tools.x86_64 0:2.4.6-40.el7.centos.1
  mailcap.noarch 0:2.1.41-2.el7

```

```

Complete!
---> 4c48e5c8534f
Removing intermediate container 9180cd73ff3b
Step 4 : RUN echo "BONJOUR ! Mon premier DockerFile" > /var/www/html/index.html
---> Running in 164f59bc76ba
---> 9faea92a1733
Removing intermediate container 164f59bc76ba
Step 5 : EXPOSE 80:8081
---> Running in 99eadf2975b6
---> ffc940be582c
Removing intermediate container 99eadf2975b6
Step 6 : CMD -D FOREGROUND
---> Running in 8ead11431edc
---> 8dc57aa83644
Removing intermediate container 8ead11431edc

```

```

Step 7 : ENTRYPOINT /usr/sbin/httpd
---> Running in d76f1980a044
---> 22745597c048
Removing intermediate container d76f1980a044
Successfully built 22745597c048
[root@SrvCentOS7-2 DockerWeb]#

```

On remarque que la construction de l'image se fait par étape. Autant d'étapes que d'instructions dans le fichier Dockerfile.

Vérification (en affichant la liste des images):

```

[root@SrvCentOS7-2 DockerWeb]# docker images

```

REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE
centweb	latest	22745597c048	20 minutes ago	310.6 MB
<none>	<none>	c42a694b33a5	14 hours ago	1.113 MB
msabir/centos_modif	latest	5923bcd902b2	16 hours ago	196.7 MB
docker.io/hello-world	latest	4c4abd6d4278	2 weeks ago	967 B
docker.io/centos	latest	28e524afdd05	5 weeks ago	196.7 MB
docker.io/busybox	latest	307ac631f1b5	7 weeks ago	1.113 MB
docker.io/tutum/wordpress	latest	20c52a34e232	8 weeks ago	476.4 MB

```

[root@SrvCentOS7-2 DockerWeb]#

```

On peut exécuter notre conteneur :

```
# docker run -d -p 80:80 centweb
```

```

[root@SrvCentOS7-2 DockerWeb]#
[root@SrvCentOS7-2 DockerWeb]# docker run -d -p 80:80 centweb
ad0e177c1484106acea3ad2ad7b37b370934214338515d115730d408c4946476
Usage of loopback devices is strongly discouraged for production use. Either use `--storage-opt dm.thinpooldev` or use `
--storage-opt dm.no_warn_on_loop_devices=true` to suppress this warning.
[root@SrvCentOS7-2 DockerWeb]#

```

Vérification (état du conteneur)

```

[root@SrvCentOS7-2 DockerWeb]# docker ps

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
ad0e177c1484	centweb	"/usr/sbin/httpd -D F"	2 minutes ago	Up 2 minutes	0.0.0.0:80->80/tcp, 8081/tcp
cc1b2fae5c83	centos	"bash"	16 hours ago	Up 16 hours	

```


[root@SrvCentOS7-2 DockerWeb]#

```

Le conteneur centweb est en cours d'exécution.

Test du fonctionnement du service web du conteneur :

```
[root@SrvCentOS7-2 DockerWeb]# curl http://localhost  
BONJOUR ! Mon premier DockerFile  
[root@SrvCentOS7-2 DockerWeb]#
```



Ou avec un navigateur :

