

TRAVAUX DIRIGÉS

DOCKER : INSTALLATION ET PREMIER PAS

I. INTRODUCTION

le but de ce TD est de procéder à l'installation de Docker dans votre environnement et à lancer les premières commandes

II. INSTALLATION DE DOCKER

tout d'abord, afin de partir d'une installation propre, nous allons supprimer les anciennes installations de l'éco-système docker (cli, runtime, ...)

```
1 ## suppression de l'ancienne installation
2 sudo apt remove docker docker-engine docker.io containerd runc
```

ensuite il faut préparer l'installation proprement dit

```
1 ## Mise en place du depot Docker
2 sudo apt install ca-certificates curl gnupg lsb-release
3 sudo mkdir -m 0755 -p /etc/apt/keyrings
4 curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /etc/apt
  /keyrings/docker.gpg
5
6 echo \
7 "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https
  ://download.docker.com/linux/debian \
8 $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

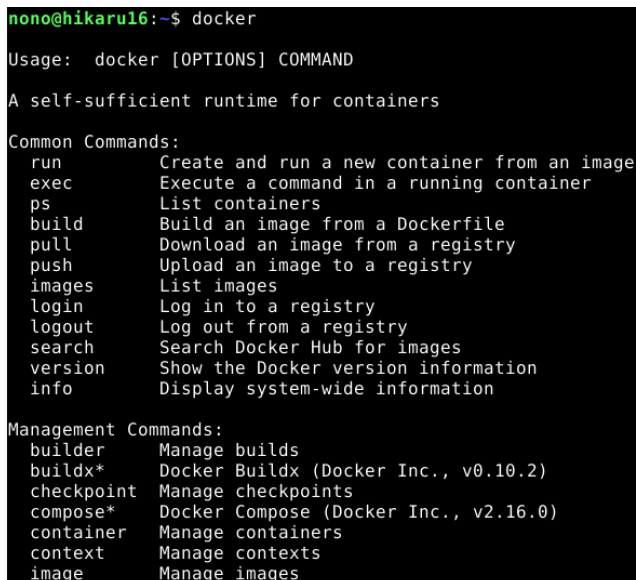
Le téléchargement se fait de façon sécurisée avec une clé de certificat.

```
1 ## installation de l'eco-systeme docker
2 sudo apt install apt-transport-https
3 sudo apt update # pour mettre a jour la liste des depots
4 sudo apt install docker-ce docker-ce-cli containerd
```

à présent vous pouvez tester la commande dans un terminal

```
1 sudo docker
```

vous devriez obtenir le résultat présenté par la capture 1



```
nono@hikaru16:~$ 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
pull     Download an image from a registry
push     Upload an image to a registry
images   List images
login    Log in 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 (Docker Inc., v0.10.2)
checkpoint Manage checkpoints
compose* Docker Compose (Docker Inc., v2.16.0)
container Manage containers
context  Manage contexts
image    Manage images
```

FIGURE 1. résultat de l'exécution de la commande docker.

nous allons exécuter notre premier conteneur. il faut au préalable s'assurer que le démon docker est en fonctionnement.

```
1 sudo systemctl status docker
```

qui doit nous retourner les informations illustrées par 2

```
root@hikaru16:/home/nono# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-02-16 11:18:11 CET; 3min 42s ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 12042 (dockerd)
      Tasks: 9
     Memory: 63.7M
        CPU: 3.602s
    CGroup: /system.slice/docker.service
            └─12042 /usr/sbin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

févr. 16 11:18:10 hikaru16 dockerd[12042]: time="2023-02-16T11:18:10.943574930+01:00" level=info
févr. 16 11:18:11 hikaru16 dockerd[12042]: time="2023-02-16T11:18:11.003184795+01:00" level=info
févr. 16 11:18:11 hikaru16 dockerd[12042]: time="2023-02-16T11:18:11.046897198+01:00" level=info
févr. 16 11:18:11 hikaru16 dockerd[12042]: time="2023-02-16T11:18:11.450254162+01:00" level=info
févr. 16 11:18:11 hikaru16 dockerd[12042]: time="2023-02-16T11:18:11.593781804+01:00" level=info
févr. 16 11:18:11 hikaru16 dockerd[12042]: time="2023-02-16T11:18:11.677816653+01:00" level=info
févr. 16 11:18:11 hikaru16 dockerd[12042]: time="2023-02-16T11:18:11.679959812+01:00" level=info
févr. 16 11:18:11 hikaru16 systemd[1]: Started Docker Application Container Engine.
févr. 16 11:18:11 hikaru16 dockerd[12042]: time="2023-02-16T11:18:11.738757889+01:00" level=info
févr. 16 11:18:25 hikaru16 dockerd[12042]: time="2023-02-16T11:18:25.721756615+01:00" level=info
lines 1-22/22 (END)
```

FIGURE 2. status du service docker en fonctionnement.

si ce n'est pas le cas, un simple appel à la commande

```
1 sudo systemctl start docker
```

doit régler le souci.

III. PREMIÈRES COMMANDES

III.1. exercice 1.

- (1) Pour démarrer notre premier conteneur nous allons utiliser la commande suivante

```
1 sudo docker run alpine:latest
```

- (2) lancer la commande suivante

```
1 sudo docker ps
```

Que fait cette commande ? Que constatez vous ?

- (3) Quelles sont les étapes effectuées par la commande docker run ?
- (4) Que se passe t'il lorsque nous ajoutons une commande à docker run

```
1 sudo docker run alpine:latest echo hello
```

III.2. exercice 2 : shell interactif.

- (1) lancez le conteneur basé sur alpine en mode interactif non détaché et sans commande.
- (2) Que constatez vous ?
- (3) quelle commande est lancée par défaut ?

III.3. exercice 3 : interactif ou détaché.

- (1) lancer un conteneur alpine avec la commande ping www.google.fr. Que se passe t'il ?
- (2) arrêtez le conteneur avec Ctrl+C. Que pouvez vous dire à propos du conteneur
- (3) relancez le container en mode interactif avec la même commande, puis arrêtez le à l'aide de Ctrl+P Ctrl+Q ? Que pouvez dire sur le conteneur ?
- (4) lancer le conteneur en mode détaché. Que pouvez vous dire sur l'état du conteneur ?

III.4. exercice 4 : exécution de commande dans un conteneur.

- (1) lancer un conteneur alpine en mode interactif détaché. Quel est son état ?
- (2) à l'aide de la commande suivante, lister les éléments du path courant

```
1 sudo docker exec id_conteneur ls
```

- (3) lancer un shell interactif dans le conteneur et afficher les processus