

Louka Fauvel  
2 BTS SIO

## Exploitation de Docker sous Linux (1 sur 2)

### 1<sup>ère</sup> Partie : Installation d'une machine virtuelle Debian 11.x

Créer une machine virtuelle.

### 2<sup>ème</sup> Partie : Installation de Docker sur Linux

Installer des paquets nécessaires à l'utilisation du dépôt docker en https avec la commande  
**apt install apt-transport-https ca-certificates curl gnupg2 software-properties-common**

```
root@debian11:~# apt install apt-transport-https ca-certificates curl gnupg2 software-properties-common
```

Importer la clé du dépôt docker avec la commande :

**curl -fsSL https://download.docker.com/linux/\$(. /etc/os-release; echo "\$ID")/gpg |**

**apt-key add -**

```
root@debian11:~# curl -fsSL https://download.docker.com/linux/$(. /etc/os-release; echo "$ID")/gpg |  
apt-key add -
```

Intégrer le dépôt docker dans la *source.list* avec la commande **add-apt-repository "deb**

**[arch=amd64] https://download.docker.com/linux/\$(. /etc/os-release; echo**

**"\$ID") \$(lsb\_release -cs) stable"** et mettre à jour les dépôts avec la commande : **apt update**

```
root@debian11:~# add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/$(. /etc/os-  
release; echo "$ID") $(lsb_release -cs) stable"  
root@debian11:~# apt update  
Atteint :1 http://security.debian.org/debian-security bullseye-security InRelease  
Atteint :2 http://deb.debian.org/debian bullseye InRelease  
Atteint :3 http://deb.debian.org/debian bullseye-updates InRelease  
Réception de :4 https://download.docker.com/linux/debian bullseye InRelease [43,3 kB]  
Réception de :5 https://download.docker.com/linux/debian bullseye/stable amd64 Packages [9 412 B]  
52,8 ko réceptionnés en 1s (74,5 ko/s)  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait  
Lecture des informations d'état... Fait  
Tous les paquets sont à jour.
```

Installer Docker avec la commande : **apt install docker-ce**

```
root@debian11:~# apt install docker-ce
```

Configurer Docker pour démarrer automatiquement avec la commande : **systemctl enable docker**

```
root@debian11:~# systemctl enable docker
Synchronizing state of docker.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable docker
```

Vérifier l'installation de Docker avec la commande : **docker version**

```
root@debian11:~# docker version
Client: Docker Engine - Community
 Version:      20.10.14
 API version:  1.41
 Go version:   go1.16.15
 Git commit:   a224086
 Built:        Thu Mar 24 01:48:21 2022
 OS/Arch:     linux/amd64
 Context:      default
 Experimental: true

Server: Docker Engine - Community
 Engine:
  Version:      20.10.14
  API version:  1.41 (minimum version 1.12)
  Go version:   go1.16.15
  Git commit:   87a90dc
  Built:        Thu Mar 24 01:46:14 2022
  OS/Arch:     linux/amd64
  Experimental: false
 containerd:
  Version:      1.5.11
  GitCommit:    3df54a852345ae127d1fa3092b95168e4a88e2f8
 runc:
  Version:      1.0.3
  GitCommit:    v1.0.3-0-gf46b6ba
 docker-init:
  Version:      0.19.0
  GitCommit:    de40ad0
```

Lancer un Docker test : "Hello world" avec la commande : **docker run hello-world**

```
root@debian11:~# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:10d7d58d5ebd2a652f4d93fdd86da8f265f5318c6a73cc5b6a9798ff6d2b2e67
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.

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/
```

### 3<sup>ème</sup> Partie : Lancement d'un conteneur en interactif

Lancer un conteneur ubuntu en mode interactif avec la commande :

**docker run --name serveurUbuntu -it ubuntu**

--name : permet de nommer le conteneur

-i : permet le mode interactif

-t : permet d'avoir un pseudo terminal pour exécuter des commandes dans le conteneur

```
root@debian11:~# docker run --name serveurUbuntu -it ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
e0b25ef51634: Pull complete
Digest: sha256:9101220a875cee98b016668342c489ff0674f247f6ca20dfc91b91c0f28581ae
Status: Downloaded newer image for ubuntu:latest
root@611488ac93ab:/# _
```

Mettre à jour le système et y installer le service ssh avec la commande : **apt update** puis **apt install openssh-server**

```
root@611488ac93ab:/# apt update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [1104 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [25.8 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [868 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [1731 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:12 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1153 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [1178 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2146 kB]
Get:16 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [30.3 kB]
Get:17 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [51.2 kB]
Get:18 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [26.0 kB]
Fetched 21.7 MB in 15s (1488 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
4 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@611488ac93ab:/# apt install openssh-server
```

```

Please select the geographic area in which you live. Subsequent configuration questions will narrow
this down by presenting a list of cities, representing the time zones in which they are located.

  1. Africa    3. Antarctica  5. Arctic    7. Atlantic  9. Indian    11. SystemV  13. Etc
  2. America  4. Australia  6. Asia      8. Europe    10. Pacific  12. US
Geographic area: 8

Please select the city or region corresponding to your time zone.

  1. Amsterdam  12. Busingen  23. Kiev      34. Moscow    45. Saratov    56. Vatican
  2. Andorra    13. Chisinau  24. Kirov     35. Nicosia    46. Simferopol  57. Vienna
  3. Astrakhan  14. Copenhagen 25. Lisbon    36. Oslo      47. Skopje     58. Vilnius
  4. Athens     15. Dublin   26. Ljubljana 37. Paris      48. Sofia      59. Volgograd
  5. Belfast    16. Gibraltar 27. London    38. Podgorica  49. Stockholm  60. Warsaw
  6. Belgrade   17. Guernsey  28. Luxembourg 39. Prague     50. Tallinn    61. Zagreb
  7. Berlin     18. Helsinki  29. Madrid    40. Riga       51. Tirane     62. Zaporozhye
  8. Bratislava 19. Isle_of_Man 30. Malta     41. Rome       52. Tiraspol   63. Zurich
  9. Brussels   20. Istanbul  31. Mariehamn 42. Samara     53. Ulyanovsk
  10. Bucharest 21. Jersey    32. Minsk     43. San_Marino 54. Uzhgorod
  11. Budapest  22. Kaliningrad 33. Monaco    44. Sarajevo   55. Vaduz
Time zone: 37
Progress: [ 68%] [#####.....]

```

Créer un utilisateur pour pouvoir se connecter ultérieurement avec la commande :

**adduser « votre nom de user »**

```

root@611488ac93ab:/# adduser louka
Adding user `louka' ...
Adding new group `louka' (1000) ...
Adding new user `louka' (1000) with group `louka' ...
Creating home directory `/home/louka' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for louka
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
root@611488ac93ab:/# _

```

Pour arrêter le conteneur, utiliser la commande suivante : **exit**

```

root@611488ac93ab:/# exit
exit
root@debian11:~# _

```

Pour réactiver et accéder de nouveau au conteneur : **docker start serveurUbuntu** puis

**docker attach serveurUbuntu**

```

root@debian11:~# docker start serveurUbuntu
serveurUbuntu
root@debian11:~# docker attach serveurUbuntu
root@611488ac93ab:/#

```

Pour voir les modifications apportées au conteneur, utiliser la commande :

**docker diff serveurUbuntu**

```
root@debian11:~# docker diff serveurUbuntu _
```

Pour voir les processus lancés dans le conteneur, utiliser la commande : **docker top**

**serveurUbuntu**

```
root@debian11:~# docker top serveurUbuntu
Error response from daemon: Container 611488ac93ab24b875f8ef4c13789213c1f64ac7e2f38863e4a7ede966c3b677 is not running
root@debian11:~# docker start serveurUbuntu
serveurUbuntu
root@debian11:~# docker top serveurUbuntu
UID                PID                PPID                C                   STIME
TTY                TIME              CMD
root               9309              9288               0                   16:33
?                  00:00:00          bash
```

Rentrer dans le conteneur **serveurUbuntu** et pour démarrer le service ssh, utiliser la commande : **service ssh start**

Appuyer sur Ctrl-P, suivi de Ctrl-Q pour sortir du conteneur sans l'arrêter.

Refaire **docker top serveurUbuntu** pour vérifier que le service ssh est activé dans le conteneur.

```
root@debian11:~# docker start serveurUbuntu
serveurUbuntu
root@debian11:~# docker attach serveurUbuntu
root@611488ac93ab:/# service ssh start
* Starting OpenBSD Secure Shell server sshd [ OK ]
root@611488ac93ab:/# read escape sequence
root@debian11:~# docker top serveurUbuntu
UID                PID                PPID                C                   STIME
TTY                TIME              CMD
root               9871              9851               0                   16:49
?                  00:00:00          bash
root               9924              9871               0                   16:50
?                  00:00:00          sshd: /usr/sbin/sshd [listener] 0 of 10-100 startups
```

## 4<sup>ème</sup> Partie : Création d'une nouvelle image à partir d'un conteneur

Pour créer une nouvelle image ubuntu qui contient le service ssh, utiliser la commande :

**docker commit serveurUbuntu ubuntu:ssh**

Cela permet d'identifier de manière unique la nouvelle image grâce au hash produit :

**docker images**

```
root@debian11:~# docker commit serveurUbuntu ubuntu:ssh
sha256:b64a1d95c3ed575ebab0bd301d9019d370328e1b40f4b76a99615e62f8d0c2d0
root@debian11:~# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	ssh	b64a1d95c3ed	34 seconds ago	234MB
ubuntu	latest	825d55fb6340	9 days ago	72.8MB
hello-world	latest	feb5d9fea6a5	6 months ago	13.3kB

## 5<sup>ème</sup> Partie : Rendre accessible un service (conteneur en arrière-plan)

Pour créer un nouveau conteneur exécutant le service ssh, utiliser la commande :

**docker run -d -p @IPMachineHôte:22222:22 --name serveurSSH ubuntu:ssh /usr/sbin/sshd -D**

```
root@debian11:~# docker run -d -p 192.168.1.10:22222:22 --name serveurSSH ubuntu:ssh /usr/sbin/sshd -D
eed4a87d6edbb63b61d2a093c52c0c5ee1f0e2187a984902d8dc269d79507352
root@debian11:~# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
eed4a87d6edb	ubuntu:ssh	"/usr/sbin/sshd -D"	6 seconds ago	Up 6 seconds	192.168.1.10:22222->22/tcp

serveurSSH

Pour accéder au conteneur en ssh à partir de n'importe quel poste, utiliser la commande : **ssh user@IP -p 22222** .

```
root@debian11:~# ssh louka@192.168.1.10 -p 22222
The authenticity of host '[192.168.1.10]:22222 ([192.168.1.10]:22222)' can't be established.
ECDSA key fingerprint is SHA256:V3IOjrnft8e8wyLZbZz+Yh5AMapnEm3FQhL5Tkpnax4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.1.10]:22222' (ECDSA) to the list of known hosts.
louka@192.168.1.10's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.10.0-13-amd64 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

louka@eed4a87d6edb:~$
```

Pour se déconnecter, utiliser la commande : **exit** .

```
louka@eed4a87d6edb:~$ exit
logout
Connection to 192.168.1.10 closed.
root@debian11:~# _
```

Pour afficher les logs du conteneur, utiliser la commande : **docker logs serveurSSH** .

Si rien ne s'affiche alors tout va bien.

```
root@debian11:~# docker logs serveurSSH
root@debian11:~#
```