

TP1 – Gestion des conteneurs

Brahim Hamdi

Objectifs :

- Savoir gérer les conteneurs Docker
- Savoir se connecter à un conteneur

Préparation de l'environnement

Dans cette partie, nous allons préparer l'environnement des travaux pratiques. Les outils utilisés dans cet environnement sont :

- Git
- VirtualBox 6
- Vagrant

1. Git clonez le dépôt Git suivant : <https://github.com/brahimhamdi/docker-lab>

```
brahim@Training:~$ git clone https://github.com/brahimhamdi/docker-lab
Clonage dans 'docker-lab'...
remote: Enumerating objects: 153, done.
remote: Counting objects: 100% (153/153), done.
remote: Compressing objects: 100% (63/63), done.
remote: Total 153 (delta 21), reused 152 (delta 20), pack-reused 0 (from 0)
Réception d'objets: 100% (153/153), 25.70 Kio | 8.56 Mio/s, fait.
Résolution des deltas: 100% (21/21), fait.
brahim@Training:~$
```

1. Déplacez-vous sous le dossier docker-lab et lancez le déploiement de l'environnement.

```
brahim@Training:~$ cd docker-lab/
brahim@Training:~/docker-lab$
brahim@Training:~/docker-lab$ vagrant up
Bringing machine 'Manager' up with 'virtualbox' provider...
Bringing machine 'worker1' up with 'virtualbox' provider...
Bringing machine 'worker2' up with 'virtualbox' provider...
==> Manager: Importing base box 'generic/ubuntu2204'...
```

...

Cette commande va déployer et provisionner les trois Machines Virtuelles décrites dans le tableau suivant :

Machine	Manager	Worker1	Worker2
<i>OS</i>	Ubuntu 22.04	Ubuntu 22.04	Ubuntu 22.04
<i>IP</i>	192.168.205.10	192.168.205.11	192.168.205.12
<i>CPU</i>	2 CPU	1 CPU	1 CPU
<i>Mémoire</i>	2 Go RAM	1 Go RAM	1 Go RAM

Docker-ce est installé sur les 3 VMs.

2. Une fois terminé, vérifiez que les 3 VMs sont accessibles par ssh, et que Docker est installé et son service a démarré.

```
brahim@Training:~/docker-lab$ vagrant ssh Manager
Last login: Sat Oct 19 10:07:39 2024 from 10.0.2.2
vagrant@Manager:~$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2024-10-19 09:33:18 UTC; 35min ago
 TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
    Main PID: 23868 (dockerd)
       Tasks: 16
      Memory: 246.7M
         CPU: 25.652s
    CGroup: /system.slice/docker.service
            └─23868 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
              25190 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 8000 -container-ip 172.17.0.2 -container-port 80
vagrant@Manager:~$
vagrant@Manager:~$
logout
brahim@Training:~/docker-lab$ vagrant ssh worker1
vagrant@worker1:~$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2024-10-19 09:49:51 UTC; 19min ago
 TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
    Main PID: 24736 (dockerd)
       Tasks: 8
      Memory: 24.3M
         CPU: 400ms
    CGroup: /system.slice/docker.service
            └─24736 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
vagrant@worker1:~$
```

...

3. Sur la machine **Manager**, donnez les droits à l'utilisateur courant (vagrant dans notre cas) d'exécuter les commandes docker dans les privilèges du super-utilisateur (sans sudo).

```
brahim@Training:~/docker-lab$ vagrant ssh Manager
Last login: Sat Oct 19 09:38:47 2024 from 10.0.2.2
vagrant@Manager:~$
vagrant@Manager:~$ docker ps
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get
"http://%2Fvar%2Frun%2Fdocker.sock/v1.24/containers/json": dial unix /var/run/docker.sock: connect: permis
sion denied
vagrant@Manager:~$
vagrant@Manager:~$ sudo usermod -aG docker ${USER}
vagrant@Manager:~$
vagrant@Manager:~$
logout
brahim@Training:~/docker-lab$ vagrant ssh Manager
Last login: Sat Oct 19 09:43:46 2024 from 10.0.2.2
vagrant@Manager:~$
vagrant@Manager:~$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED    STATUS    PORTS    NAMES
vagrant@Manager:~$
```

N.B. : Dans tous les TPs, sauf le dernier (Docker Swarm), nous utiliserons uniquement la VM **Manager**.

Création des conteneurs

4. Créez un nouveau conteneur avec les paramètres suivants :

- Image : serveur web nginx
- Détacher le conteneur du terminal
- Nom du conteneur : wiki_web
- Mapper le port 80 du conteneur sur le port 8000 de l'hôte

```
vagrant@Manager:~$ docker run -d --name wiki_web -p 8000:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
a480a496ba95: Pull complete
f3ace1b8ce45: Pull complete
11d6fdd0e8a7: Pull complete
f1091da6fd5c: Pull complete
40eea07b53d8: Pull complete
6476794e50f4: Pull complete
70850b3ec6b2: Pull complete
Digest: sha256:28402db69fec7c17e179ea87882667f1e054391138f77ffaf0c3eb388efc3ffb
Status: Downloaded newer image for nginx:latest
5cb28da36a5005df341457aa04b76a25b53c1519544a986d7b0d6d5b58c8ca6e
vagrant@Manager:~$
```

◦ Vérifiez que le conteneur est bien créé et qu'il tourne sur la machine.

```
vagrant@Manager:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                NAMES
5cb28da36a50   nginx    "/docker-entrypoint...." About a minute ago Up About a minute   0.0.0.0:8000->80/tcp   wiki_web
vagrant@Manager:~$
```

◦ Donnez des informations détaillées sur le conteneur (IP, Passerelle, Adresse MAC, ...)

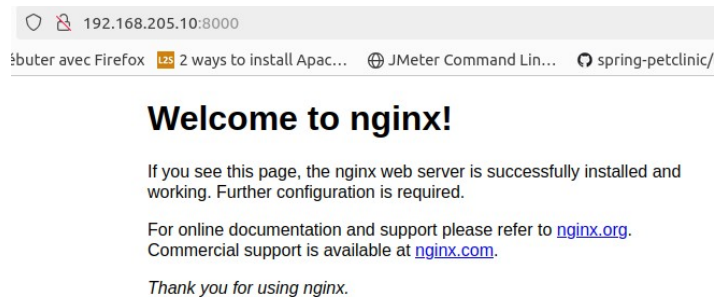
```
vagrant@Manager:~$ docker inspect wiki_web
[
  {
    "Id": "5cb28da36a5005df341457aa04b76a25b53c1519544a986d7b0d6d5b58c8ca6e",
    "Created": "2024-10-19T09:49:36.238492404Z",
    "Path": "/docker-entrypoint.sh",
    "Args": [
      "nginx",
      "-g",
      "daemon off;"
    ],
    "Networks": {
      "bridge": {
        "IPAMConfig": null,
        "Links": null,
        "Aliases": null,
        "NetworkID": "36dfd7307966a7610c7292dfabbe862e49452a37a27214295263b4feb7ef0df",
        "EndpointID": "d99914af10c5f2fc7cbb6352182f66cb58ee57c94fd2349aecca17704130f50",
        "Gateway": "172.17.0.1",
        "IPAddress": "172.17.0.2",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "MacAddress": "02:42:ac:11:00:02",
        "DriverOpts": null
      }
    }
  }
]
vagrant@Manager:~$
```

- ```
vagrant@Manager:~$ docker run -d --name wiki_db -e MARIADB_ROOT_PASSWORD="secret" mariadb
Unable to find image 'mariadb:latest' locally
latest: Pulling from library/mariadb
ff65ddf9395b: Pull complete
60bf5e4921eb: Pull complete
d4d47b8d6ea0: Pull complete
e38fc7e50f91: Pull complete
725acd03a3bc: Pull complete
db106334738c: Pull complete
234171a3de7d: Pull complete
25efdea61d9c: Pull complete
Digest: sha256:4alde8fa2a929944373d7421105500ff6f889ce90dcb883fbb2fdb070e4d427e
Status: Downloaded newer image for mariadb:latest
a78c57a7c942d7842b63aecff8e5e85d6477d5bafcd6c009f3e8aa126cb274cc6
vagrant@Manager:~$
```

```
vagrant@Manager:~$ docker inspect wiki_db
[
 {
 "Id": "a78c57a7c942d7842b63aecff8e5e85d6477d5bafd6c009f3e8aa126cb274cc6",
 "Created": "2024-10-19T09:59:42.084028833Z",
 "Path": "docker-entrypoint.sh",
 "Args": [
 "mariadb"
],
 "State": {
 "Status": "running",
 "Running": true,
 "Paused": false,
 "Restarting": false,
 "OOMKilled": false,
 "IPV6Gateway": "",
 "MacAddress": "02:42:ac:11:00:03",
 "Networks": {
 "bridge": {
 "IPAMConfig": null,
 "Links": null,
 "Aliases": null,
 "NetworkID": "36dfd7307966a7610c7292dfabbeff862e49452a37a27214295263b4feb7ef0df",
 "EndpointID": "7eb7b33d953d9bb50bef45ffb264a6bb7a235145480941dbb3237dd78eb758d8",
 "Gateway": "172.17.0.1",
 "IPAddress": "172.17.0.3",
 "IPPrefixLen": 16,
 "IPv6Gateway": "",
 "GlobalIPv6Address": "",
 "GlobalIPv6PrefixLen": 0,
 "MacAddress": "02:42:ac:11:00:03",
 "DriverOpts": null
 }
 }
 }
 }
]
vagrant@Manager:~$
```

## Se connecter aux conteneurs

- Le conteneur *wiki\_web* exécute l'image *nginx* qu'est un serveur web. Affichez l'interface web de *wiki\_web*.

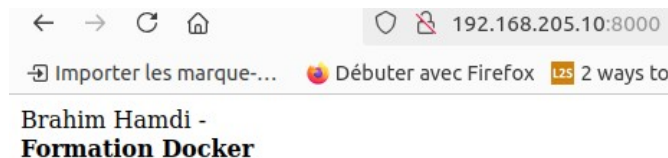


- Connectez-vous au conteneur *wiki\_web*, et modifiez l'interface web par défaut (*/usr/share/nginx/html/index.html*).

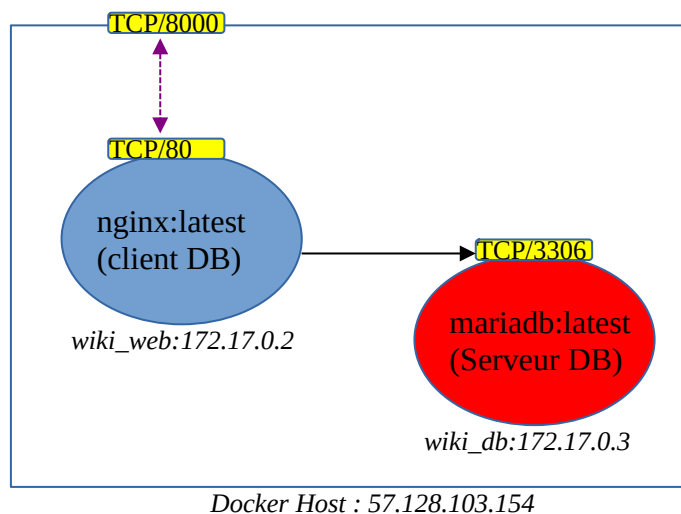
```
vagrant@Manager:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
a78c57a7c942 mariadb "docker-entrypoint.s..." 16 minutes ago Up 15 minutes 3306/tcp wiki_db
5cb28da36a50 nginx "/docker-entrypoint. ..." 26 minutes ago Up 26 minutes 0.0.0.0:8000->80/tcp wiki_web

vagrant@Manager:~$ docker exec -ti wiki_web bash
root@5cb28da36a50:/#
root@5cb28da36a50:/# echo "Brahim Hamdi -
Formation Docker" > /usr/share/nginx/html/index.html
root@5cb28da36a50:/#
root@5cb28da36a50:/# exit
exit
vagrant@Manager:~$
```

- Affichez la page web après la modification.



- Le conteneur *wiki\_web* est le client de la base de données *mariadb* déployée sur le conteneur *wiki\_db*. La figure suivante explique l'architecture de notre solution déployée sur les conteneurs.



- Connectez vous au conteneur *wiki\_web* et installez le paquet *apt mariadb-client*.

```
vagrant@Manager:~$ docker exec -ti wiki_web bash
root@5cb28da36a50:/#
root@5cb28da36a50:/# apt update && apt install mariadb-client -y
Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB]
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [55.4 kB]
Get:3 http://deb.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 Packages [8787 kB]
Get:5 http://deb.debian.org/debian bookworm-updates/main amd64 Packages [2468 B]
Get:6 http://deb.debian.org/debian-security bookworm-security/main amd64 Packages [188 kB]
Fetched 9232 kB in 9s (1007 kB/s)
Reading package lists... Done
```

...

- Testez la connexion du client de la base de données (conteneur *wiki\_web*) sur le serveur de base de données (conteneur *wiki\_db*).

```
vagrant@Manager:~$ docker exec -ti wiki_web bash
root@5cb28da36a50:/#
root@5cb28da36a50:/# mysql -h 172.17.0.3 -u root -psecret
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 11.5.2-MariaDB-ubu2404 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.001 sec)

MariaDB [(none)]>
MariaDB [(none)]> quit;
Bye
root@5cb28da36a50:/#
root@5cb28da36a50:/# exit
exit
vagrant@Manager:~$
```

## Suppression des conteneurs

8. Essayez de supprimer les conteneurs *wiki\_web* et *wiki\_db*. Y a-t-il une erreur ? Pourquoi ?

```
vagrant@Manager:~$ docker rm wiki_web wiki_db
Error response from daemon: You cannot remove a running container 5cb28da36a5005df341457aa04b76a25b53c1519544a986d7b0d6d5b58c8ca6e. Stop the container before attempting removal or force remove
Error response from daemon: You cannot remove a running container a78c57a7c942d7842b63aefcf8e5e85d6477d5bafd6c009f3e8aa126cb274c6. Stop the container before attempting removal or force remove
vagrant@Manager:~$
```

## 9. Arrêtez puis supprimez les deux conteneurs.

```
vagrant@Manager:~$ docker stop wiki_web wiki_db
wiki_web
wiki_db
vagrant@Manager:~$
vagrant@Manager:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
vagrant@Manager:~$
vagrant@Manager:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
a78c57a7c942 mariadb "docker-entrypoint.s..." 35 minutes ago Exited (0) 15 seconds ago wiki_db
5cb28da36a50 nginx "/docker-entrypoint..." 45 minutes ago Exited (0) 14 seconds ago wiki_web
vagrant@Manager:~$
vagrant@Manager:~$ docker rm wiki_web wiki_db
wiki_web
wiki_db
vagrant@Manager:~$
vagrant@Manager:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
vagrant@Manager:~$
```