

TP2 – Gestion des conteneurs Docker

Objectifs :

- Savoir gérer les conteneurs Docker
- Savoir se connecter à un conteneur exécutant un serveur web
- Savoir se connecter à un conteneur exécutant un serveur de base de données

Installation de Docker

1. Suivez les étapes pour installer docker sous ubuntu 20.04 :

- Tout d'abord, mettez à jour votre liste de packages existante :

sudo apt update

- Ensuite, installez quelques paquets pré-requis qui permettent à apt d'utiliser les paquets sur HTTPS :

sudo apt install apt-transport-https ca-certificates curl software-properties-common

- Ensuite, ajoutez la clé GPG du dépôt officiel de Docker à votre système :

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

- Ajoutez le référentiel Docker aux sources APT :

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"

- Ensuite, mettez à jour la base de données des paquets avec les paquets Docker à partir du référentiel qui vient d'être ajouté :

sudo apt update

- Enfin, installez Docker :

sudo apt install docker-ce

- Le Docker devrait maintenant être installé, le démon démarré, et le processus autorisé à démarrer au boot. Vérifiez qu'il tourne :

sudo systemctl status docker

- Si vous voulez éviter de taper sudo chaque fois que vous exécutez la commande docker, ajoutez votre nom d'utilisateur au groupe docker :

sudo usermod -aG docker \${USER}

Création des conteneurs

2. 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

```
ubuntu@formation1:~$ docker run -d --name wiki_web -p 8000:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
faef57eae888: Pull complete
76579e9ed380: Pull complete
cf707e233955: Pull complete
91bb7937700d: Pull complete
4b962717ba55: Pull complete
f46d7b05649a: Pull complete
103501419a0a: Pull complete
Digest: sha256:08bc36ad52474e528cc1ea3426b5e3f4bad8a130318e3140d6cfe29c8892c7ef
Status: Downloaded newer image for nginx:latest
64bf3727fa5c381b2fdac4f915b956d6d59dc082633290a53ff308390d6f16c6
ubuntu@formation1:~$
```

○ Vérifiez que le conteneur est bien créé et il tourne

```
ubuntu@formation1:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
64bf3727fa5c	nginx	"/docker-entrypoint. ..."	About a minute ago	Up About a minute	0.0.0.0:8000->80/tcp, :::8000->80/tcp	wiki_web

```
ubuntu@formation1:~$
```

○ Donnez des informations détaillées sur le conteneur

```
ubuntu@formation1:~$ docker inspect wiki_web
```

```
[
  {
    "Id": "64bf3727fa5c381b2fdac4f915b956d6d59dc082633290a53ff308390d6f16c6",
    "Created": "2023-07-15T11:55:45.910262568Z",
    "Path": "/docker-entrypoint.sh",
    "Args": [
      "nginx",
      "-g",
      "daemon off;"
    ],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 79523,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2023-07-15T11:55:46.601647537Z",
      "FinishedAt": "0001-01-01T00:00:00Z"
    },
    "Image": "sha256:021283c8eb95be02b23db0de7f609d603553c6714785e7a673c6594a624ffbd",
    "ResolvConfPath": "/var/lib/docker/containers/64bf3727fa5c381b2fdac4f915b956d6d59dc082633290a53ff308390d6f16c6/resolv.conf"
  }
]
```

...

```

"MacAddress": "02:42:ac:11:00:02",
"Networks": {
  "bridge": {
    "IPAMConfig": null,
    "Links": null,
    "Aliases": null,
    "NetworkID": "15c26d892c9b3a3c98ee8618579b9e1e6dc180e0f8f3ede1b1910f3632fbdf3a",
    "EndpointID": "11c60fb5b5290be14526e2e7d0631f616eb3b8b1c22331da31e910af322d3f78",
    "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
  }
}
}
}
]
ubuntu@formation1:~$

```

3. Créez un autre conteneur avec les paramètres suivants :

- Image : Base de donnée MariaDB
- Détacher le conteneur du terminal
- Nom du conteneur : wiki_db
- Mot de passe de la base de données (MYSQL_ROOT_PASSWORD) : « *secret* »

```

ubuntu@formation1:~$ docker run -d --name wiki_db -e MARIADB_ROOT_PASSWORD=secret mariadb
Unable to find image 'mariadb:latest' locally
latest: Pulling from library/mariadb
9d19ee268e0d: Pull complete
718e898a86ff: Pull complete
43bd7a143a6c: Pull complete
80cdf483b70a: Pull complete
8c13b197eea7: Pull complete
fe76c18bf258: Pull complete
67fa5c829e7f: Pull complete
a5cb79f31ff6: Pull complete
Digest: sha256:f94bb4868d953fed5220c9d3cdc8449f4c314efb07d3a18eefa6010b383f2ab8
Status: Downloaded newer image for mariadb:latest
1c856841c9f633d8053dff1ed0d33a71448e0e3035c66b7c0d9796a48e12b946
ubuntu@formation1:~$
ubuntu@formation1:~$ docker ps

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
1c856841c9f6	mariadb	"docker-entrypoint.s..."	3 seconds ago	Up 2 seconds	3306/tcp	wiki_db
64bf3727fa5c	nginx	"/docker-entrypoint..."	30 minutes ago	Up 30 minutes	0.0.0.0:8000->80/tcp, :::8000->80/tcp	wiki_web

```

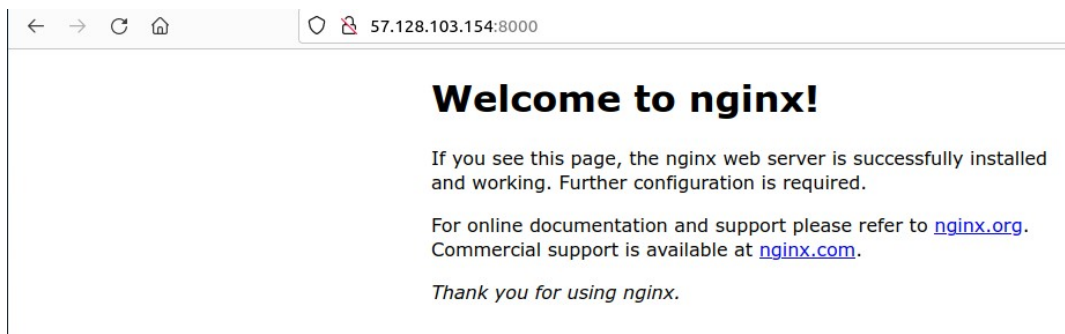
ubuntu@formation1:~$

```

◦ Donnez les informations détaillées sur ce conteneur.

[illegible]

4. Affichez l'interface web du conteneur `wiki_web`. (remplacez l'IP par la votre)

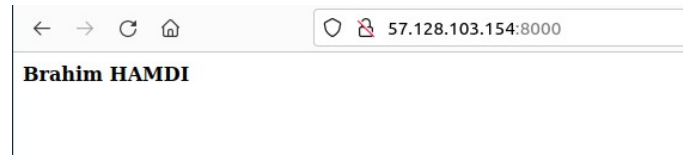


- Connectez-vous au conteneur `wiki_web`, et modifiez l'interface web par défaut (`/usr/share/nginx/html/index.html`).

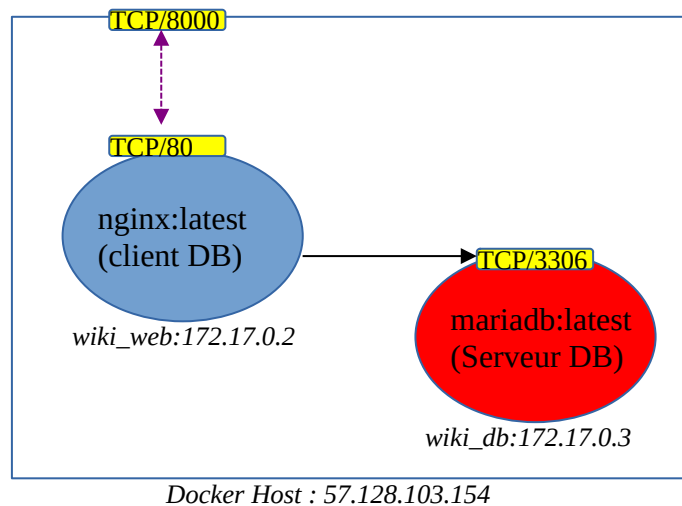
```
ubuntu@formation1:~$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS          PORTS          NAMES
1c856841c9f6   mariadb   "docker-entrypoint.s..." About a minute ago Up About a minute 3306/tcp      wiki_db
64bf3727fa5c   nginx     "/docker-entrypoint..." 31 minutes ago   Up 31 minutes   0.0.0.0:8000->80/tcp, :::8000->80/tcp   wiki_web

ubuntu@formation1:~$
ubuntu@formation1:~$ docker exec -it wiki_web bash
root@64bf3727fa5c:/#
root@64bf3727fa5c:/# echo "<b> Brahm HAMDI</b>" > /usr/share/nginx/html/index.html
root@64bf3727fa5c:/#
root@64bf3727fa5c:/# exit
exit
ubuntu@formation1:~$
```

- Affichez la page web après la modification (remplacez l'IP par la votre).



- On va considérer *wiki_web* comme client de la base de données qui tourne sur le serveur (conteneur) *wiki_db*. La figure suivante explique l'architecture de notre solution jusqu'à maintenant.



- Connectez vous au conteneur *wiki_web* et installez *mariadb-client*.

```

ubuntu@formation1:~$ docker exec -it wiki_web bash
root@64bf3727fa5c:/# apt update && apt install mariadb-client -y
Get:1 http://deb.debian.org/debian bookworm InRelease [147 kB]
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [52.1 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 [8904 kB]
Get:5 http://deb.debian.org/debian-security bookworm-security/main amd64 Packages [47.3 kB]
Fetched 9199 kB in 2s (4559 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libconfig-inifiles-perl libdbd-mariadb-perl libdbi-perl libedit2 libgdbm-compat4 libgdbm6 libgpm2 libmariadb3 libncurses6 libperl5.36
  libterm-readkey-perl mariadb-client-core mariadb-common mysql-common netbase perl perl-modules-5.36
Suggested packages:
  libmariadb3 perl libdbd-driver-perl libdbi-driver-perl libdbi-statement-perl edb libgdbm-compat4 libgdbm6 libgpm2 libmariadb3 libncurses6 libperl5.36
  libterm-readkey-perl mariadb-client-core mariadb-common mysql-common netbase perl perl-modules-5.36

```

- Connectez-vous au serveur de base de données (conteneur wiki_db) à partir du client (conteneur wiki_web).

```
ubuntu@formation1:~$ docker exec -it wiki_web bash
root@64bf3727fa5c:/# mysql -h 172.17.0.3 -u root -psecret
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 11.0.2-MariaDB-1:11.0.2+maria-ubu2204 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)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.001 sec)

MariaDB [(none)]> q
-> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'q' at line 1
MariaDB [(none)]> exit
Bye
root@64bf3727fa5c:/# exit
exit
ubuntu@formation1:~$
ubuntu@formation1:~$
```

6. Essayez de supprimer les conteneurs wiki_web et wiki_db. Y a-t-il une erreur ? Pourquoi ?

```
ubuntu@formation1:~$ docker rm wiki_web wiki_db
Error response from daemon: You cannot remove a running container 64bf3727fa5c381b2fdac4f915b956d6d59dc082633290a53ff308390d6f16c6. Stop the container before attempting removal or force remove
Error response from daemon: You cannot remove a running container 1c856841c9f633d8053dff1ed0d33a71448e0e3035c66b7c0d9796a48e12b946. Stop the container before attempting removal or force remove
ubuntu@formation1:~$
```

7. Arrêtez puis supprimez les deux conteneurs.

```
ubuntu@formation1:~$ docker stop wiki_web wiki_db
wiki_web
wiki_db
ubuntu@formation1:~$
ubuntu@formation1:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS              PORTS          NAMES
1c856841c9f6   mariadb   "docker-entrypoint.s..." 15 minutes ago Exited (0) 4 seconds ago      wiki_db
64bf3727fa5c   nginx     "/docker-entrypoint...." 45 minutes ago Exited (0) 4 seconds ago      wiki_web
ubuntu@formation1:~$
ubuntu@formation1:~$ docker rm wiki_web wiki_db
wiki_web
wiki_db
ubuntu@formation1:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS              PORTS          NAMES
ubuntu@formation1:~$
```