

## TP DEV OPS

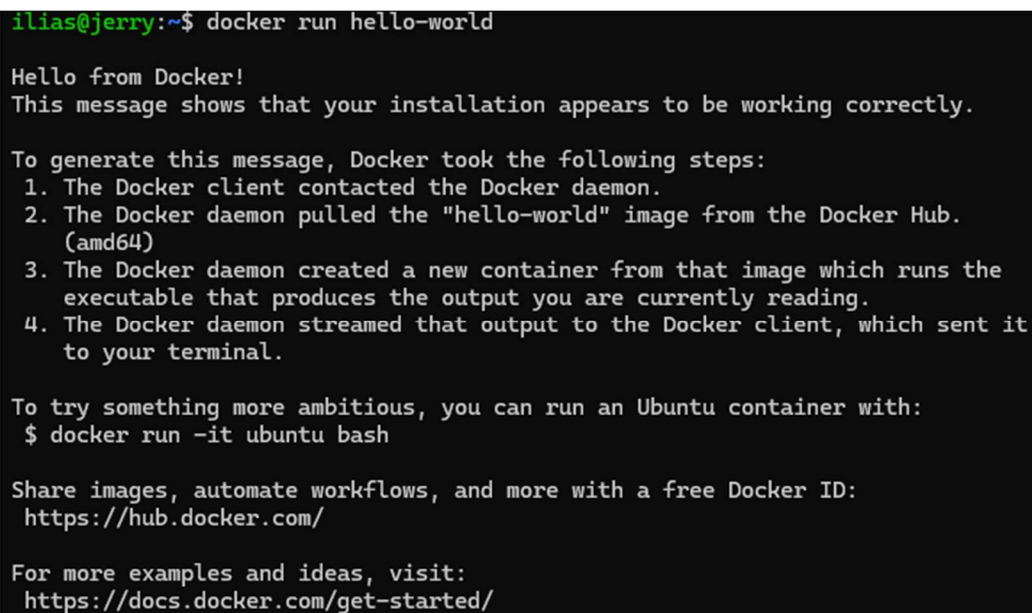
### TP1

Ceci est le compte rendu du TP1 :

Difficulté : 7/10 (ce qui était dur c'est l'installation de docker desktop, on en a parlé en cours lors de notre première séance mais j'avais un problème pour lancer et initialiser docker. J'ai dû suivre la suite du premier TP avec Thomas brunet ou Brandon Lopez comme convenu).

Quelques commandes à tester :

- **docker run hello-world**  
Hello-world d'exemple avec Docker



```
ilias@jerry:~$ 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.
   (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/
```

Apparemment tout marche !

La suite :

- **docker run -it ubuntu bash**  
Création d'un conteneur et utilisation d'un bash en interactif  
*exit ou Ctrl+D* - Pour sortir du conteneur

```

ilias@jerry:~$ docker run -it ubuntu bash
root@241ae8e9ec61:/#
exit
ilias@jerry:~$ D|

```

On a créé et quitté un conteneur !

- **docker images**

Afficher les images Docker disponibles en local

```

ilias@jerry:~$ docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
tenable/was-scanner  latest         bce3a658bef4   2 days ago     2.44GB
<none>              <none>         ef3948553143   2 weeks ago    2.44GB
ubuntu              latest         b59d21599a2b   3 weeks ago    117MB
nginx               latest         fb39280b7b9e   2 months ago   279MB
hello-world         latest         0b6a027b5cf3   5 months ago   20.4kB
ilias@jerry:~$ docker ps -a

```

Et maintenant les images qui ne sont pas exécutés :

```

ilias@jerry:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
241ae8e9ec61   ubuntu        "bash"                  About a minute ago    Exited (0) About a minute ago
a50a43ca5fd7   hello-world   "/hello"                5 minutes ago        Exited (0) 5 minutes ago
8aa6fddb8c4    nginx        "/docker-entrypoint..." 2 weeks ago         Exited (255) 2 weeks ago    0.0.0.0:8081->80/tcp
a96ab8c5cb4e   nginx        "/docker-entrypoint..." 2 weeks ago         Exited (0) 2 weeks ago
b156a79bfb92   nginx        "/docker-entrypoint..." 2 weeks ago         Exited (0) 2 weeks ago
b806302ab5d3   nginx        "/docker-entrypoint..." 2 weeks ago         Exited (0) 2 weeks ago
39b01ff444dd   nginx        "/docker-entrypoint..." 2 weeks ago         Exited (0) 2 weeks ago
ecc86f805a32   nginx        "/docker-entrypoint..." 2 weeks ago         Exited (0) 2 weeks ago
601e0b1c4a74   ubuntu        "bash"                  2 weeks ago         Exited (127) 2 weeks ago
661d071bc4d6   hello-world   "/hello"                2 weeks ago         Exited (0) 2 weeks ago
6c4424370093   hello-world   "/hello"                2 weeks ago         Exited (0) 2 weeks ago
fbf8da42fb52   hello-world   "/hello"                2 weeks ago         Exited (0) 2 weeks ago
zealous_brown
ilias@jerry:~$

```

- **docker run -p 80:80 nginx** et **docker run -p -d 80:80 nginx**

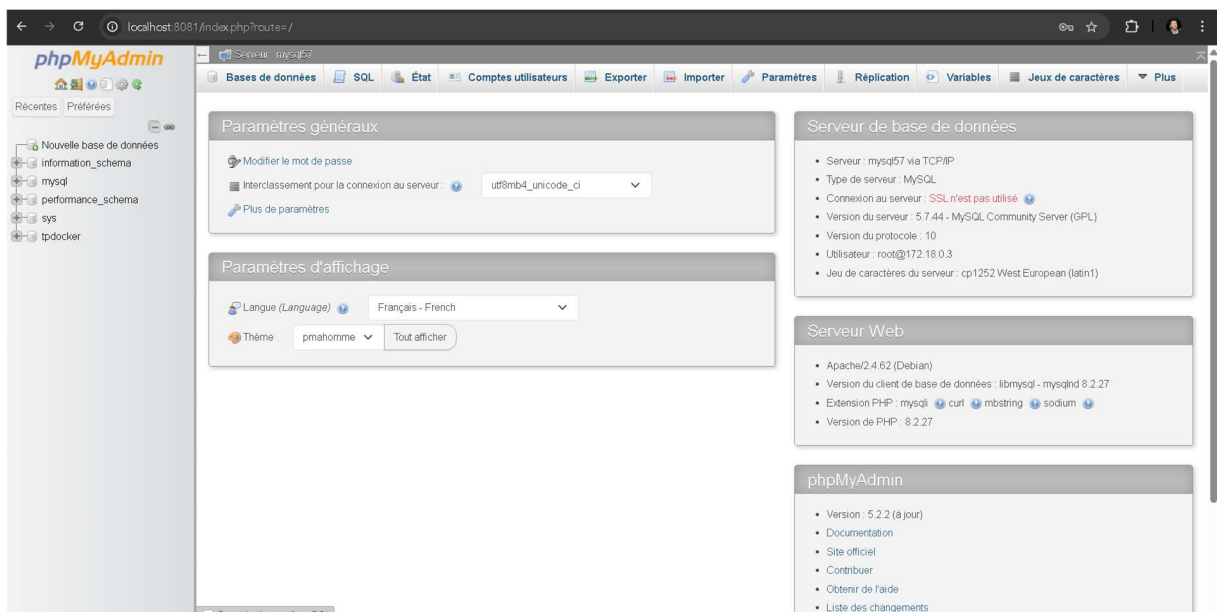
Démarre un serveur web disponible sur votre navigateur à l'adresse  
localhost:80



Maintenant le début actif du tp :

- c. Quelles différences observez-vous entre les procédures 5. et 6. ? Avantages et inconvénients de l'une et de l'autre méthode ? (Mettre en relation ce qui est observé avec ce qui a été présenté pendant le cours)

La procédure 5 est rapide et simple, elle ne m'a pas fait galérer et comparé à la méthode 6 à chaque modification on doit reconstruire l'image.



- a. Qu'apporte le fichier docker-compose par rapport aux commandes docker run ? Pourquoi est-il intéressant ? (cf. ce qui a été présenté pendant le cours)
- b. Quel moyen permet de configurer (premier utilisateur, première base de données, mot de passe root, ...) facilement le conteneur mysql au lancement ?

A : toutes les images sont dispo dans un seul fichier

B : des variables d'environnement

- b. Quelles lignes du résultat de la commande `docker inspect` justifient ce comportement ?
- c. Dans quelle situation réelles (avec quelles images) pourrait-on avoir cette configuration réseau ? Dans quel but ?

B : `docker inspect web`

TP 2 :

TP2 réalisé avec Clément Matias.

On initialise git :

```
ilias@jerry:/mnt/c/Users/ilias/Downloads$ cp TP-2-Docker.zip /home/ilias/tp-docker/
ilias@jerry:/mnt/c/Users/ilias/Downloads$ cd /home/ilias/tp-docker
unzip TP-2-Docker.zip
Archive: TP-2-Docker.zip
  creating: tp-docker-2/
  inflating: tp-docker-2/docker-compose.yml
  creating: tp-docker-2/src/
  inflating: tp-docker-2/src/index.js
  extracting: tp-docker-2/Dockerfile
  inflating: tp-docker-2/package.json
ilias@jerry:~/tp-docker$ cd tp-docker-2
```

```
2025-06-23 09:03:12 - Repository configured successfully.
2025-06-23 09:03:12 - To install Node.js, run: apt-get install nodejs -y
2025-06-23 09:03:12 - You can use N|solid Runtime as a node.js alternative
2025-06-23 09:03:12 - To install N|solid Runtime, run: apt-get install nsolid -y
```

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required
```



```

npm notice
ilias@jerry:~/tp-docker/tp-docker-2$ docker build -t ma_super_app .
[+] Building 0.3s (1/1) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile            0.1s
=> => transferring dockerfile: 94B                             0.0s
Dockerfile:3
-----
1 | FROM node:12-alpine3.9
2 |
3 | >>> ...
4 |
5 | CMD ["node", "-x", "index.js"]

ilias@jerry:~/tp-docker/tp-docker-2$ vim Dockerfile
ilias@jerry:~/tp-docker/tp-docker-2$ docker build -t ma_super_app .
[+] Building 44.6s (11/11) FINISHED                            docker:default
=> [internal] load build definition from Dockerfile            0.0s
=> => transferring dockerfile: 178B                             0.0s
=> [internal] load metadata for docker.io/library/node:12-alpine3.9 2.1s
=> [auth] library/node:pull token for registry-1.docker.io    0.0s
=> [internal] load .dockerignore                               0.0s
=> => transferring context: 2B                                    0.0s
=> [1/5] FROM docker.io/library/node:12-alpine3.9@sha256:16d40e6c285 6.7s
=> => resolve docker.io/library/node:12-alpine3.9@sha256:16d40e6c285 0.0s
=> => sha256:f5b0dlce6f59dcbee12d1ce6f4093fa9a7939 24.49MB / 24.49MB 2.4s
=> => sha256:e8a7c0650cafa577c71ba40f4423d3486d700990904 283B / 283B 0.4s
=> => sha256:31603596830fc7e56753139f9c2c6bd3759e48a 2.77MB / 2.77MB 0.9s
=> => sha256:55922cf6f31687a003e818aab6c5e42db644b2b 2.24MB / 2.24MB 0.8s
=> => extracting sha256:31603596830fc7e56753139f9c2c6bd3759e48a85065 0.4s
=> => extracting sha256:f5b0dlce6f59dcbee12d1ce6f4093fa9a7939c7751f0 4.0s
=> => extracting sha256:55922cf6f31687a003e818aab6c5e42db644b2be19a1 0.2s
=> => extracting sha256:e8a7c0650cafa577c71ba40f4423d3486d700990904d 0.0s
=> [internal] load build context                               0.2s
=> => transferring context: 317.87kB                             0.1s
=> [2/5] WORKDIR /app                                         0.3s
=> [3/5] COPY package*.json ./                                0.1s
=> [4/5] RUN npm install --production                         29.7s
=> [5/5] COPY src/ ./src/                                     0.1s
=> exporting to image                                         5.2s
=> => exporting layers                                           2.4s
=> => exporting manifest sha256:453fda7081b55ecbe4e2afd4b969ee7022db 0.0s
=> => exporting config sha256:9f91136a21c533a44fefb5a89e3a65f062629b 0.0s
=> => exporting attestation manifest sha256:363c5e0e0909146adccdf4d 0.0s
=> => exporting manifest list sha256:14f1f59895370ca555a412107fac48e 0.0s
=> => naming to docker.io/library/ma_super_app:latest          0.0s
=> => unpacking to docker.io/library/ma_super_app:latest       2.6s
ilias@jerry:~/tp-docker/tp-docker-2$ docker-compose up -d

```

Une option de npm vous permet de n'installer que ce qui est nécessaire. Quelle est cette option ? Quelle bonne pratique Docker permet-elle de respecter ?

L'option de npm qui permet d'installer seulement les dépendances nécessaires en production est

npm install --production

```
ilias@jerry:~/tp-docker/tp-docker-2$ npm install --production
npm warn config production Use '--omit=dev' instead.

added 87 packages, and audited 88 packages in 19s

14 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New major version of npm available! 10.8.2 -> 11.4.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.4.2
npm notice To update run: npm install -g npm@11.4.2
npm notice
ilias@jerry:~/tp-docker/tp-docker-2$ docker build -t ma_super_app .
[+] Building 0.3s (1/1) FINISHED                                docker:default
Step 1/1: FROM node:10.16.0-alpine
```