TP DEV OPS

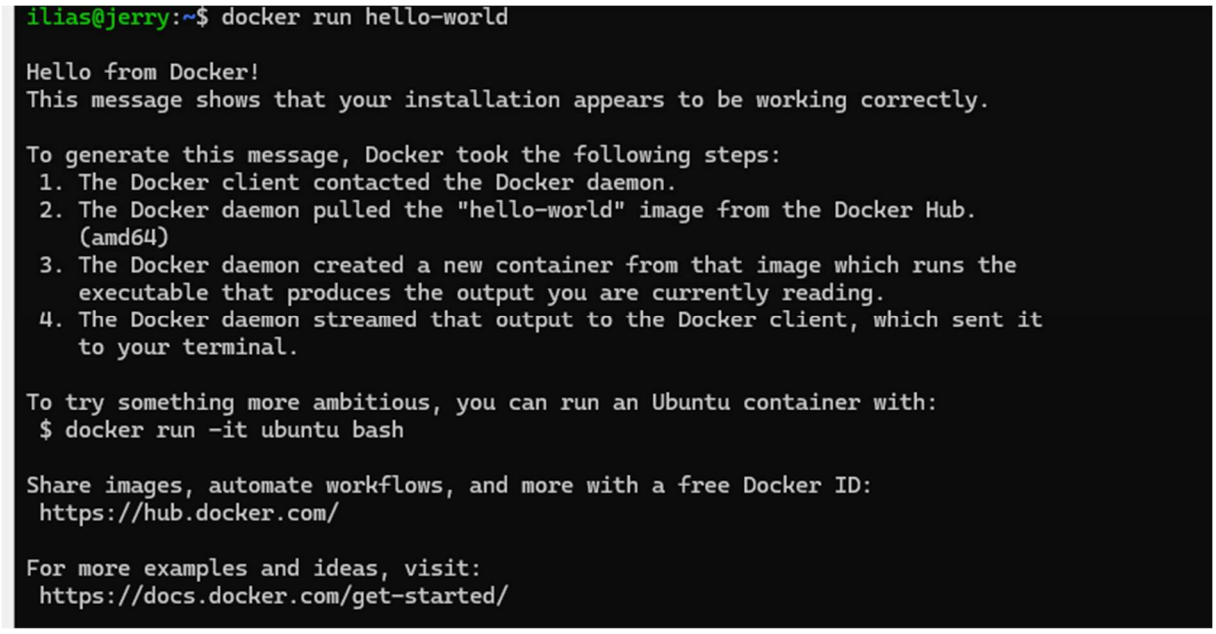
TP1

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



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 :

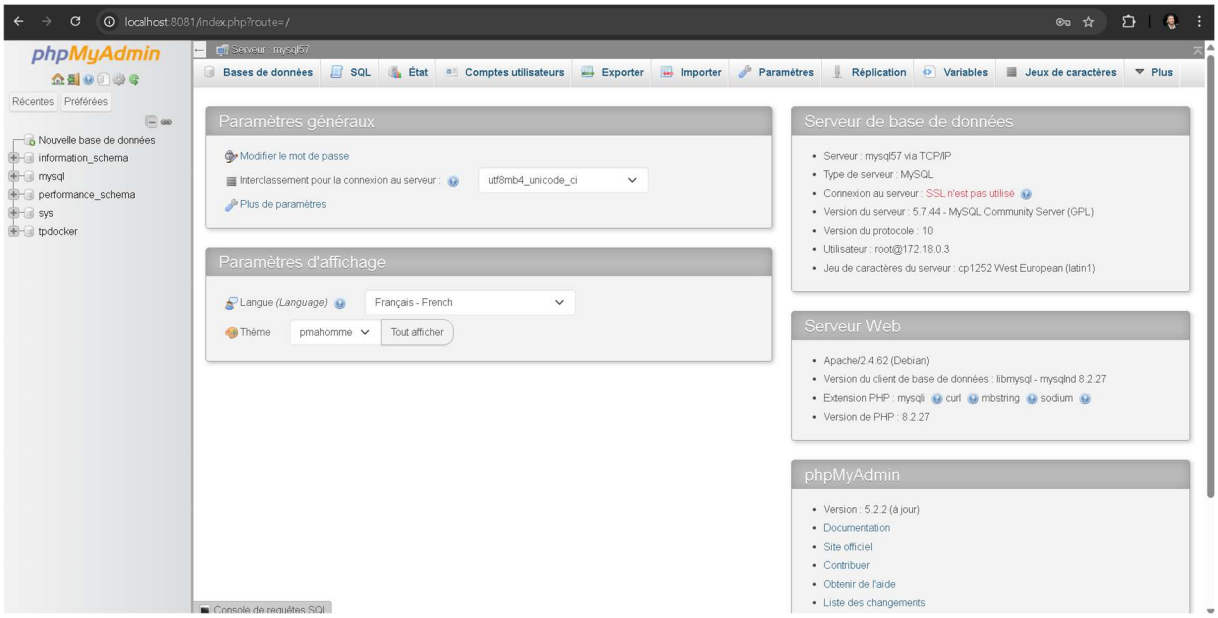
```
ilias@jerry:~$ docker ps -a
CONTAINER ID   IMAGE         COMMAND                  CREATED             STATUS                          PORTS
          NAMES
241ae8e9ec61   ubuntu        "bash"                   About a minute ago   Exited (0) About a minute ago
          confident_davinci
a50a43ca5fd7   hello-world   "/hello"                 5 minutes ago        Exited (0) 5 minutes ago
          blissful_grothendieck
8aa6fddbb8c4   nginx         "/docker-entrypoint.…"   2 weeks ago          Exited (255) 2 weeks ago        0.0.0.0:8081-
>80/tcp   test-nginx
a96ab8c5cb4e   nginx         "/docker-entrypoint.…"   2 weeks ago          Exited (0) 2 weeks ago
          infallible_poincare
b156a79bfb92   nginx         "/docker-entrypoint.…"   2 weeks ago          Exited (0) 2 weeks ago
          blissful_wiles
b806302ab5d3   nginx         "/docker-entrypoint.…"   2 weeks ago          Exited (0) 2 weeks ago
          charming_beaver
39b01ff444dd   nginx         "/docker-entrypoint.…"   2 weeks ago          Exited (0) 2 weeks ago
          relaxed_ishizaka
ecc86f805a32   nginx         "/docker-entrypoint.…"   2 weeks ago          Exited (0) 2 weeks ago
          nervous_lovelace
601e0b1c4a74   ubuntu        "bash"                   2 weeks ago          Exited (127) 2 weeks ago
          flamboyant_perlman
661d071bc4d6   hello-world   "/hello"                 2 weeks ago          Exited (0) 2 weeks ago
          eager_goldwasser
6c4424370093   hello-world   "/hello"                 2 weeks ago          Exited (0) 2 weeks ago
          affectionate_lalande
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 ficher

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

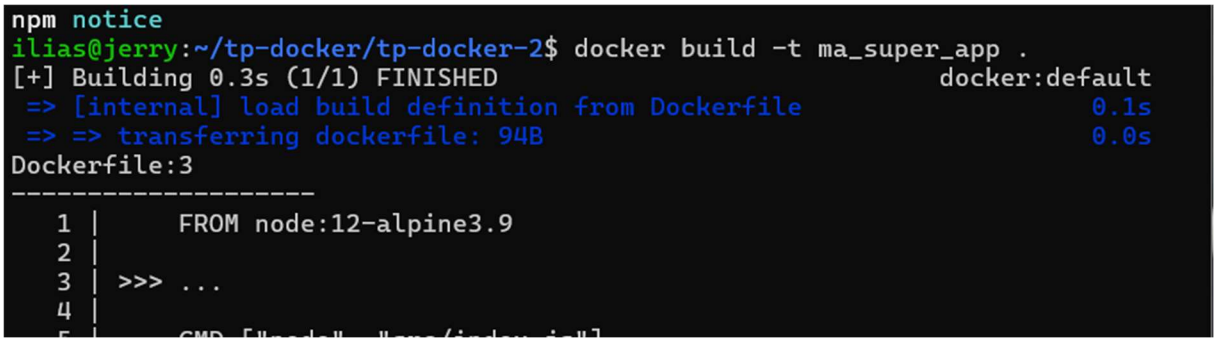
TP 2 :

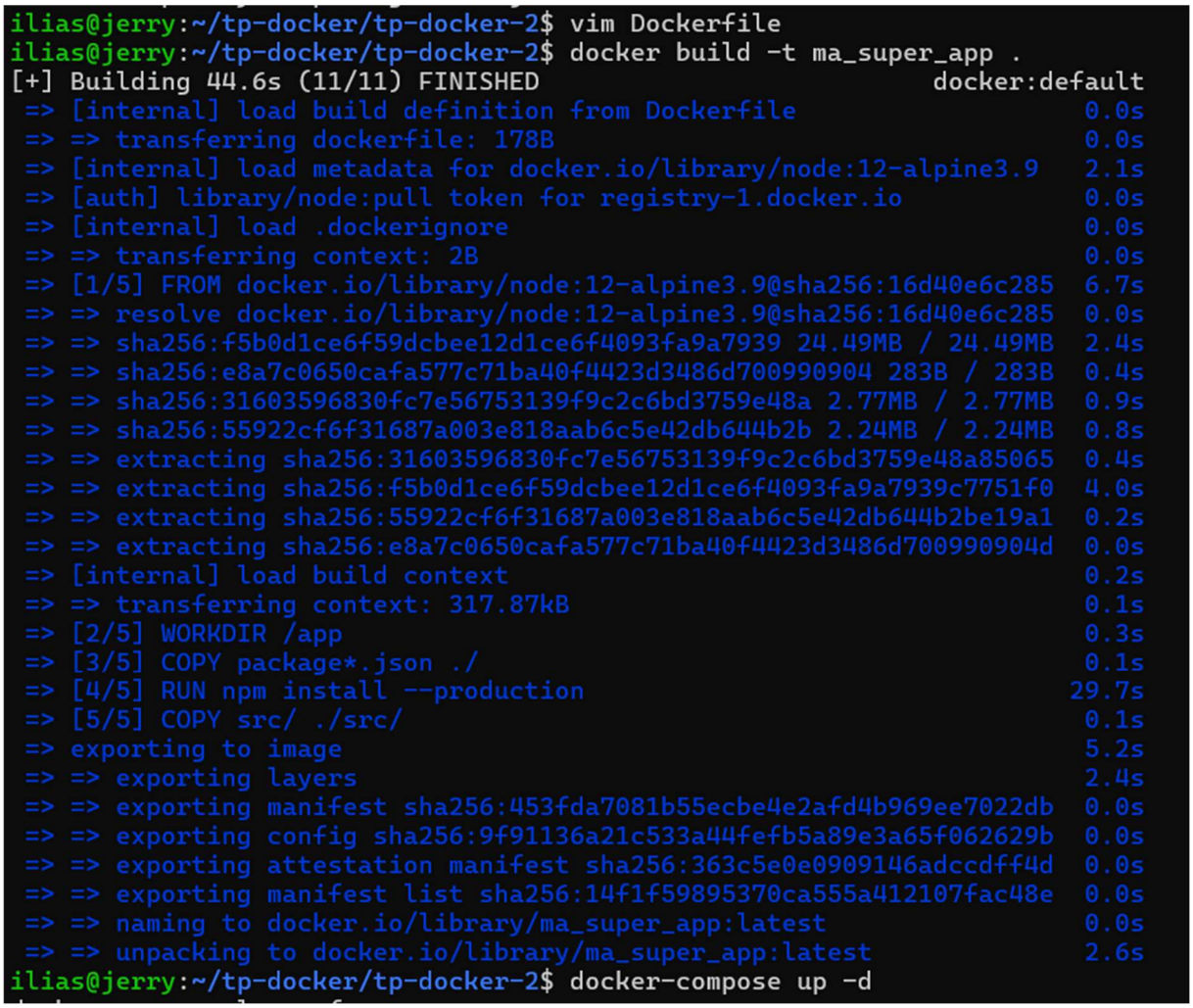
TP2 réalisé avec Clément Matias.

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 nsoli
d -y

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





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

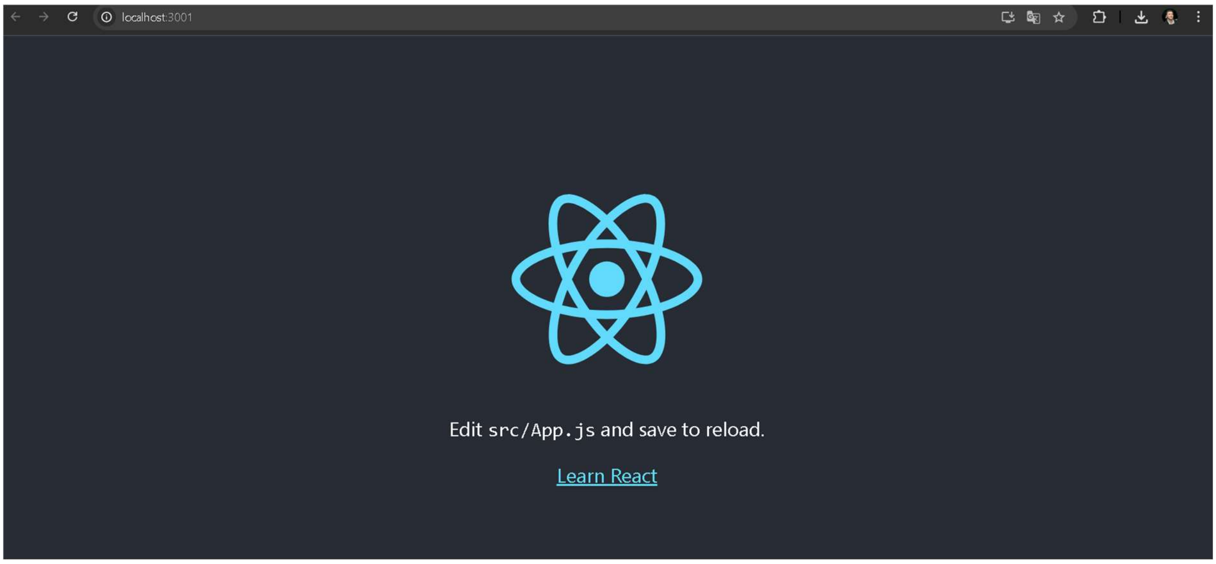
```
ilias@jerry:~/tp-docker/tp-docker-2$ npm install --production
npm warn config production Use `--omit=dev` instead.

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
```

TP 3 :

On a initialisé un projet vierge.

```
Compiled successfully!

You can now view app in the browser.

  Local:            http://localhost:3001
  On Your Network:  http://172.31.248.83:3001

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully


^C
ilias@jerry:~/tp-3-DevOps/app$ ls
README.md  node_modules  package-lock.json  package.json  public  src
```

L’appli marche correctement en local :



4. A

```
ilias@jerry:~/tp-3-DevOps$ nano Dockerfile
ilias@jerry:~/tp-3-DevOps$ ls
Dockerfile  README.md  app
ilias@jerry:~/tp-3-DevOps$ vim Dockerfile
ilias@jerry:~/tp-3-DevOps$ docker build -t react-app .
docker run -p 8080:80 react-app
[+] Building 124.9s (17/17) FINISHED                                                              docker:default
 => [internal] load build definition from Dockerfile                                                        0.1s
 => => transferring dockerfile: 331B                                                                        0.0s
 => [internal] load metadata for docker.io/library/nginx:alpine                                             2.3s
 => [internal] load metadata for docker.io/library/node:18-alpine                                           2.3s
 => [auth] library/node:pull token for registry-1.docker.io                                                 0.0s
 => [auth] library/nginx:pull token for registry-1.docker.io                                                0.0s
 => [internal] load .dockerignore                                                                           0.0s
 => => transferring context: 2B                                                                             0.0s
 => [build 1/6] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9  9.9s
 => => resolve docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8c  0.0s
 => => sha256:1e5a4c89cee5c0826c540ab06d4b6b491c96eda01837f430bd47f0d26702d6e3 1.26MB / 1.26MB              0.4s
 => => sha256:25ff2da83641908f65c3a74d80409d6b1b62ccfaab220b9ea70b80df5a2e0549 446B / 446B                  0.4s
 => => sha256:dd71dde834b5c203d162902e6b8994cb2309ae049a0eabc4efea161b2b5a3d0e 40.01MB / 40.01MB            4.1s
 => => sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870 3.64MB / 3.64MB              1.0s
 => => extracting sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870                   0.6s
```

4.b

```
ilias@jerry:~/tp-3-DevOps$ vim Dockerfile
ilias@jerry:~/tp-3-DevOps$ docker build -t react-app .
[+] Building 7.7s (17/17) FINISHED                                                                docker:default
 => [internal] load build definition from Dockerfile                                                        0.0s
 => => transferring dockerfile: 740B                                                                        0.0s
 => [internal] load metadata for docker.io/library/node:18-alpine                                           1.6s
 => [internal] load metadata for docker.io/library/nginx:alpine                                             1.6s
 => [auth] library/nginx:pull token for registry-1.docker.io                                                0.0s
 => [auth] library/node:pull token for registry-1.docker.io                                                 0.0s
 => [internal] load .dockerignore                                                                           0.0s
 => => transferring context: 2B                                                                             0.0s
 => [build 1/6] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9  0.0s
 => => resolve docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8c  0.0s
```

5.

```
ilias@jerry:~/tp-3-DevOps$ docker run -p 8080:80 react-app
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/06/23 12:54:58 [notice] 1#1: using the "epoll" event method
2025/06/23 12:54:58 [notice] 1#1: nginx/1.27.5
2025/06/23 12:54:58 [notice] 1#1: built by gcc 14.2.0 (Alpine 14.2.0)
2025/06/23 12:54:58 [notice] 1#1: OS: Linux 6.6.87.1-microsoft-standard-WSL2
2025/06/23 12:54:58 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/06/23 12:54:58 [notice] 1#1: start worker processes
2025/06/23 12:54:58 [notice] 1#1: start worker process 30
2025/06/23 12:54:58 [notice] 1#1: start worker process 31
2025/06/23 12:54:58 [notice] 1#1: start worker process 32
2025/06/23 12:54:58 [notice] 1#1: start worker process 33
2025/06/23 12:54:58 [notice] 1#1: start worker process 34
```