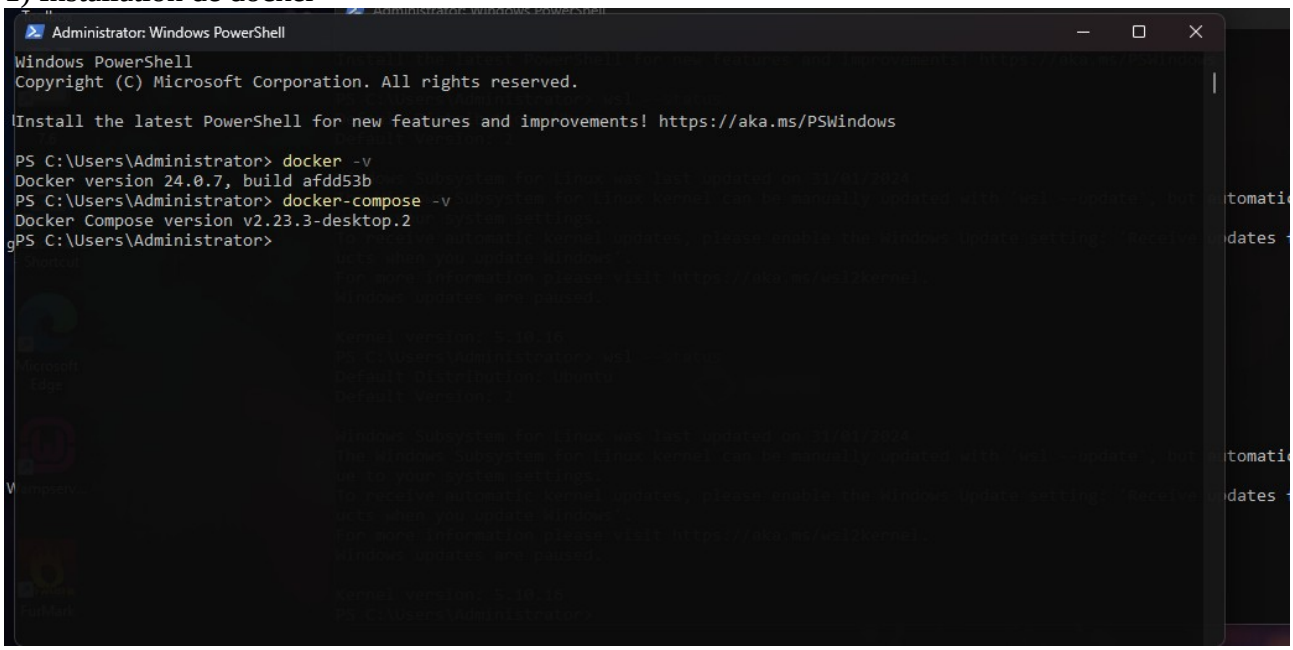
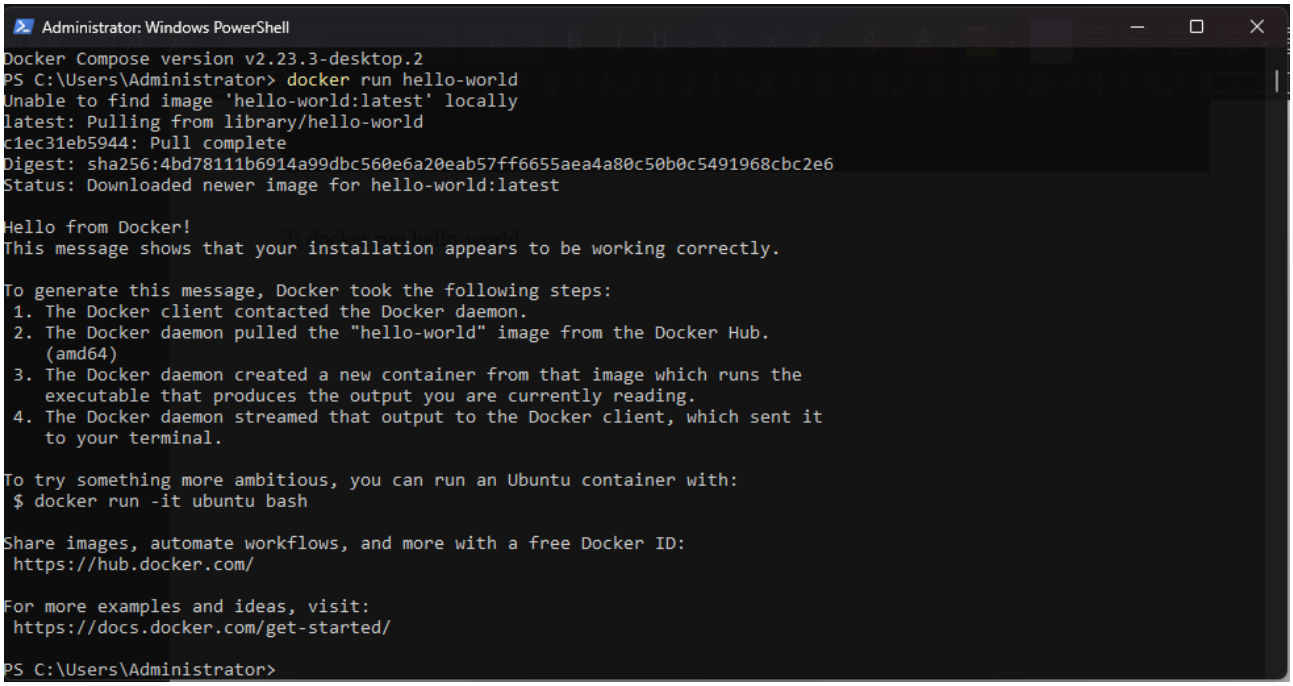
Repo git du projet : [repo tp docker]

1) Installation de docker

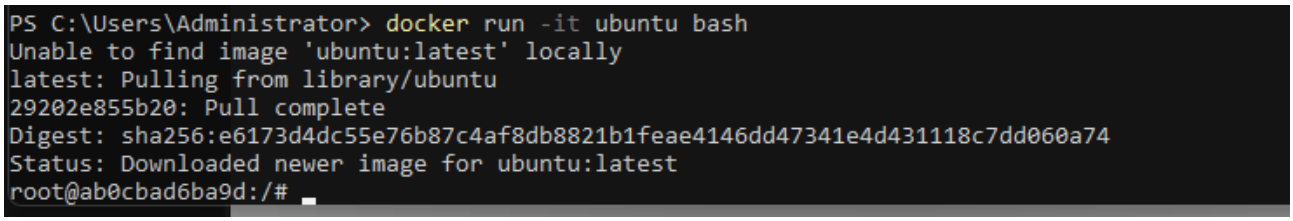


2) commandes à tester

docker run hello-world



docker run -it ubuntu bash



docker images

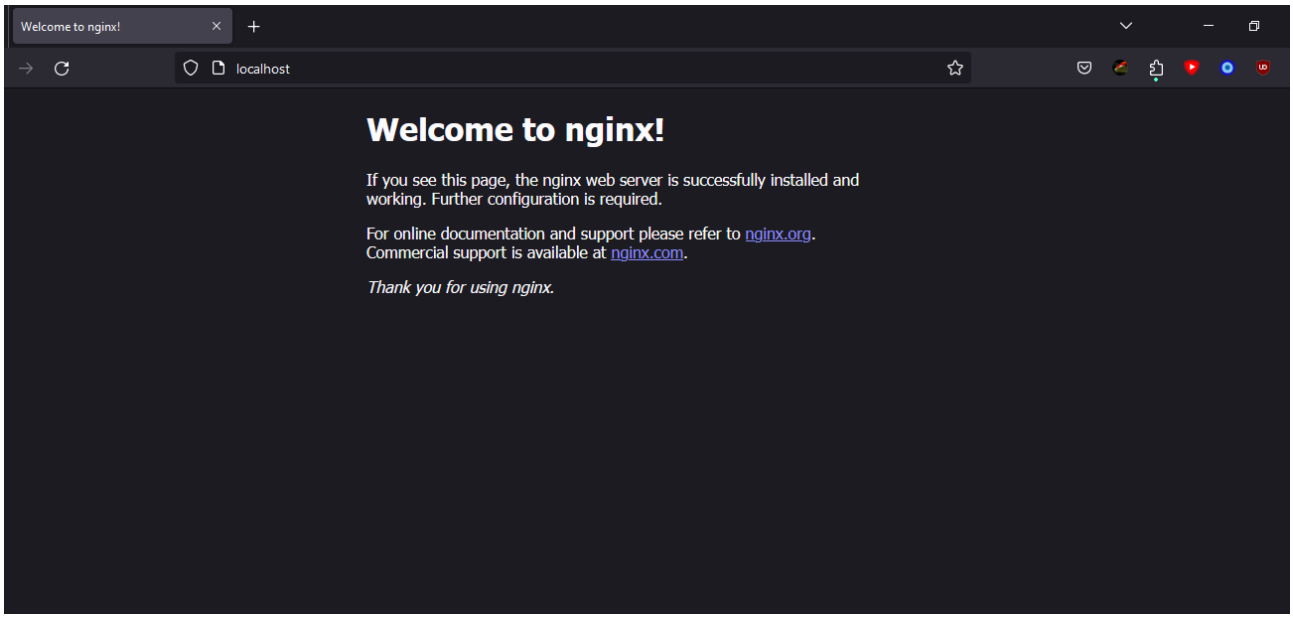
```
exit
PS C:\Users\Administrator> docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
ubuntu        latest    e34e831650c1   2 weeks ago     77.9MB
hello-world   latest    d2c94e258dcb   9 months ago    13.3kB
PS C:\Users\Administrator>
```

docker ps -a

```
PS C:\Users\Administrator> docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED         STATUS                          PORTS     NAMES
ab0cbad6ba9d   ubuntu        "bash"     2 minutes ago   Exited (0) About a minute ago             distracted_visvesvaray
a
82865f832e61   hello-world   "/hello"   4 minutes ago   Exited (0) 4 minutes ago                  wonderful_vaughan
PS C:\Users\Administrator>
```

docker run -p 80:80 nginx

```
PS C:\Users\Administrator> docker run -p 80:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
2f44b7a888fa: Pull complete
8b7dd3ed1dc3: Pull complete
35497dd96569: Pull complete
36664b6ce66b: Pull complete
2d455521f76c: Pull complete
dc9c4fdb83d6: Pull complete
8056d2bcf3b6: Pull complete
Digest: sha256:4c0fdaa8b6341bfdeca5f18f7837462c80cff90527ee35ef185571e1c327beac
Status: Downloaded newer image for nginx:latest
/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
2024/02/01 09:24:27 [notice] 1#1: using the "epoll" event method
2024/02/01 09:24:27 [notice] 1#1: nginx/1.25.3
2024/02/01 09:24:27 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2024/02/01 09:24:27 [notice] 1#1: OS: Linux 5.15.133.1-microsoft-standard-WSL2
2024/02/01 09:24:27 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/02/01 09:24:27 [notice] 1#1: start worker processes
2024/02/01 09:24:27 [notice] 1#1: start worker process 30
2024/02/01 09:24:27 [notice] 1#1: start worker process 31
2024/02/01 09:24:27 [notice] 1#1: start worker process 32
2024/02/01 09:24:27 [notice] 1#1: start worker process 33
```



5) executer un serveur web
a) récupérer l'image :
l'image récupérée précédemment est utilisée
b) vérifier que l'image est présente en local

```
PS C:\Users\Administrator> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        latest    e34e831650c1   2 weeks ago    77.9MB
nginx         latest    a8758716bb6a   3 months ago   187MB
hello-world   latest    d2c94e258dcb   9 months ago   13.3kB
PS C:\Users\Administrator>
```

c) créer un fichier index.html simple

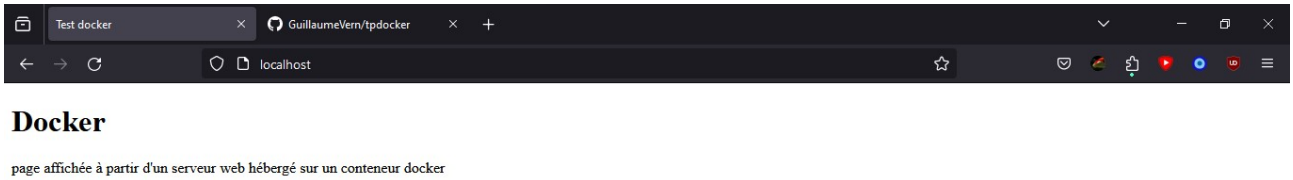
index.html U

index.html > html > body > p

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Test docker</title>
</head>
<body>
    <h1>Docker</h1>
    <p>page affichée à partir d'un serveur web hébergé sur un conteneur docker</p>
</body>
</html>
```

d) démarrer un conteneur avec une page html custom

```
PS C:\Users\Administrator> docker run -p 80:80 -v //c/Docker/web:/usr/share/nginx/html nginx
/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
2024/02/01 10:08:27 [notice] 1#1: using the "epoll" event method
2024/02/01 10:08:27 [notice] 1#1: nginx/1.25.3
2024/02/01 10:08:27 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2024/02/01 10:08:27 [notice] 1#1: OS: Linux 5.15.133.1-microsoft-standard-WSL2
2024/02/01 10:08:27 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/02/01 10:08:27 [notice] 1#1: start worker processes
2024/02/01 10:08:27 [notice] 1#1: start worker process 29
2024/02/01 10:08:27 [notice] 1#1: start worker process 30
2024/02/01 10:08:27 [notice] 1#1: start worker process 31
2024/02/01 10:08:27 [notice] 1#1: start worker process 32
```



e) suppression des conteneurs

```
PS C:\Users\Administrator> docker container ls -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
PS C:\Users\Administrator>
```

création du nouveau conteneur vierge

```
PS C:\Users\Administrator> docker create -p 80:80 nginx
d558c28b56c9c1da76eace2f70b1d48792dfa552780e774a74d3e78b6aa22646
PS C:\Users\Administrator> docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS    PORTS     NAMES
d558c28b56c9   nginx     "/docker-entrypoint.…"   5 seconds ago   Created             objective_banzai
PS C:\Users\Administrator>
```

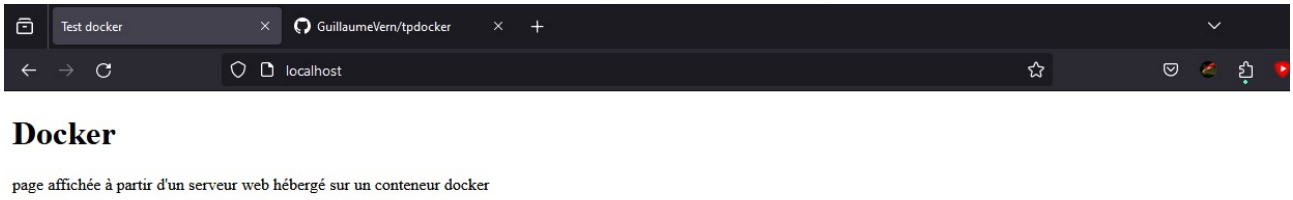
copie des fichiers dans le conteneur

```
PS C:\Users\Administrator> docker cp C:\Docker\web\index.html objective_banzai:/usr/share/nginx/html
Successfully copied 2.05kB to objective_banzai:/usr/share/nginx/html
```

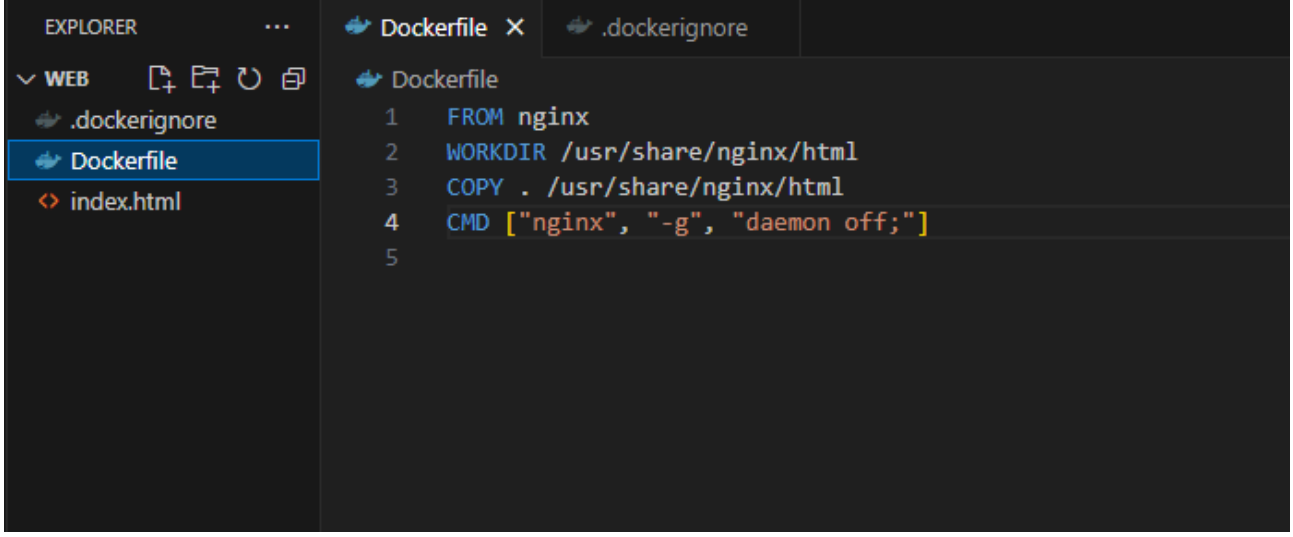
démarrage du conteneur

```
PS C:\Users\Administrator> docker start objective_banzai
objective_banzai
PS C:\Users\Administrator> docker container ls
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                NAMES
d558c28b56c9   nginx     "/docker-entrypoint.…"   2 minutes ago   Up 11 seconds  0.0.0.0:80->80/tcp   objective_banzai
PS C:\Users\Administrator>
```

6) builder une image
a)



dockerfile :



build de l’image

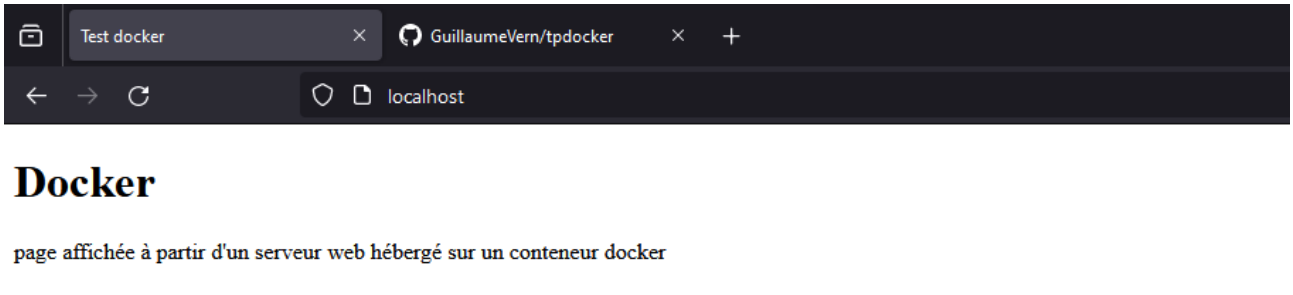
```
PS C:\docker\web> docker build ./ --tag my-nginx
[+] Building 0.3s (8/8) FINISHED                                                                                  docker:default
 => [internal] load build definition from Dockerfile                                                                        0.0s
 => => transferring dockerfile: 146B                                                                                        0.0s
 => [internal] load .dockerignore                                                                                           0.1s
 => => transferring context: 50B                                                                                            0.0s
 => [internal] load metadata for docker.io/library/nginx:latest                                                             0.0s
 => [1/3] FROM docker.io/library/nginx                                                                                      0.0s
 => [internal] load build context                                                                                           0.0s
 => => transferring context: 64B                                                                                            0.0s
 => CACHED [2/3] WORKDIR /usr/share/nginx/html                                                                              0.0s
 => CACHED [3/3] COPY . /usr/share/nginx/html                                                                               0.0s
 => exporting to image                                                                                                      0.0s
 => => exporting layers                                                                                                     0.0s
 => => writing image sha256:cfcb2918bf8d12a7f22e8cd6a957037e2dedb83d5f33f74497905523d542524b                                0.0s
 => => naming to docker.io/library/my-nginx                                                                                 0.0s

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
```

b)
démarrage du conteneurs

```
PS C:\docker\web> docker run -p 80:80 my-nginx
/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
2024/02/01 10:50:43 [notice] 1#1: using the "epoll" event method
2024/02/01 10:50:43 [notice] 1#1: nginx/1.25.3
2024/02/01 10:50:43 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2024/02/01 10:50:43 [notice] 1#1: OS: Linux 5.15.133.1-microsoft-standard-WSL2
2024/02/01 10:50:43 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/02/01 10:50:43 [notice] 1#1: start worker processes
2024/02/01 10:50:43 [notice] 1#1: start worker process 29
2024/02/01 10:50:43 [notice] 1#1: start worker process 30
2024/02/01 10:50:43 [notice] 1#1: start worker process 31
2024/02/01 10:50:43 [notice] 1#1: start worker process 32
172.17.0.1 - - [01/Feb/2024:10:50:50 +0000] "GET / HTTP/1.1" 200 321 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:122.0) Gecko/20
100101 Firefox/122.0" "-"
2024/02/01 10:50:50 [error] 29#29: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.17
.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "localhost", referrer: "http://localhost/"
172.17.0.1 - - [01/Feb/2024:10:50:50 +0000] "GET /favicon.ico HTTP/1.1" 404 153 "http://localhost/" "Mozilla/5.0 (Windows NT 10.0; Win
64; x64; rv:122.0) Gecko/20100101 Firefox/122.0" "-"
```

test



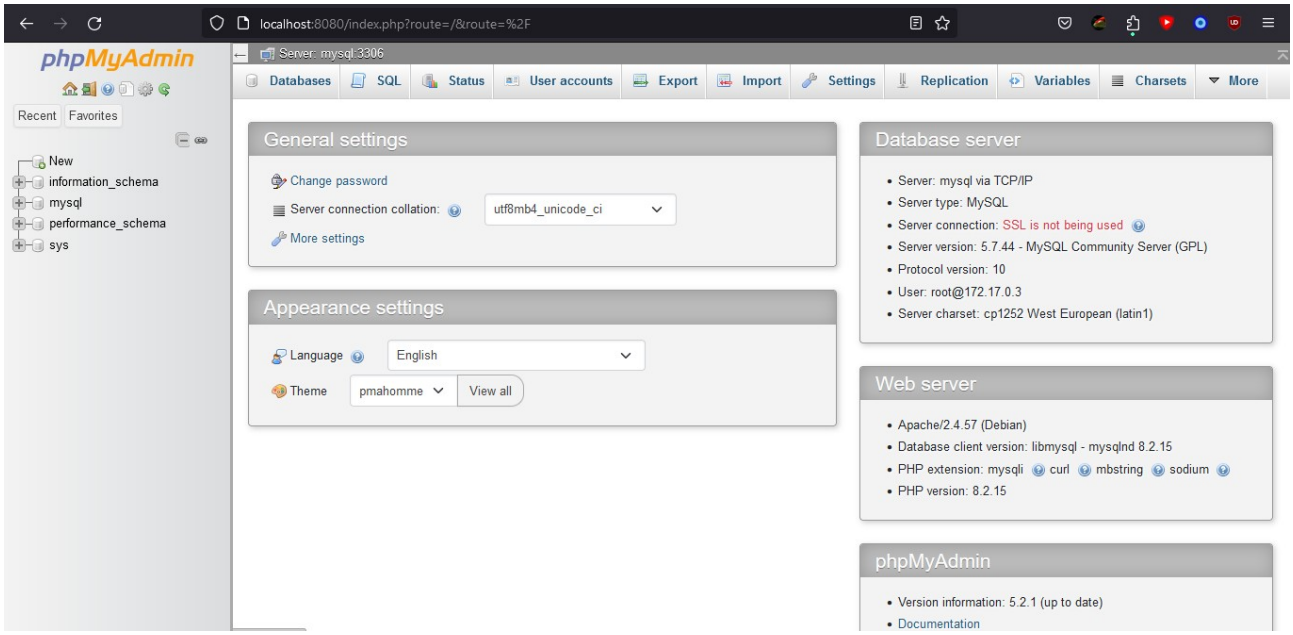
c)
une différence observée est que la méthode 5) est plus rapide pour faire des tests rapides mais la méthode 6) est automatisée après l’avoir fait une fois il n’y a plus besoin d’y toucher

la méthode 6) correspond plus a une pratique devops car il est préférable de réduire au maximum les tâches répétitives, il faut donc chercher a automatiser au maximum.

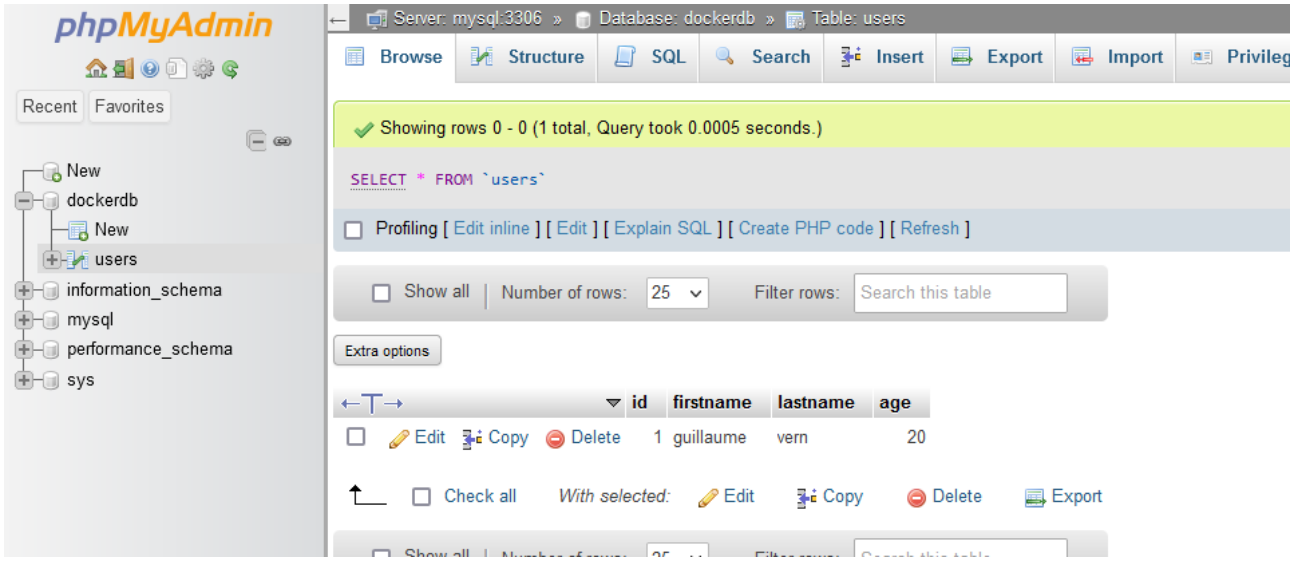
7) utiliser une base de données dans un conteneur docker

```
PS C:\docker\web> docker run --name mysql -e MYSQL_ROOT_PASSWORD=azerty2qwerty -d mysql:5.7
476afde42116ce15fde5f3590618d05004f7a2f57ff72a53504a28dddc527260
PS C:\docker\web>
```

```
PS C:\docker\web> docker run --name phpmyadmin -p 8080:80 --link mysql -e PMA_HOST=mysql -e PMA_PORT=3306 -d phpmyadmin
d7500be9b6d67a0ac5939cce2bacadfe01a9de5d8233923ea1174acf7dfe60e6
PS C:\docker\web>
```



test creation db + table + insertion données

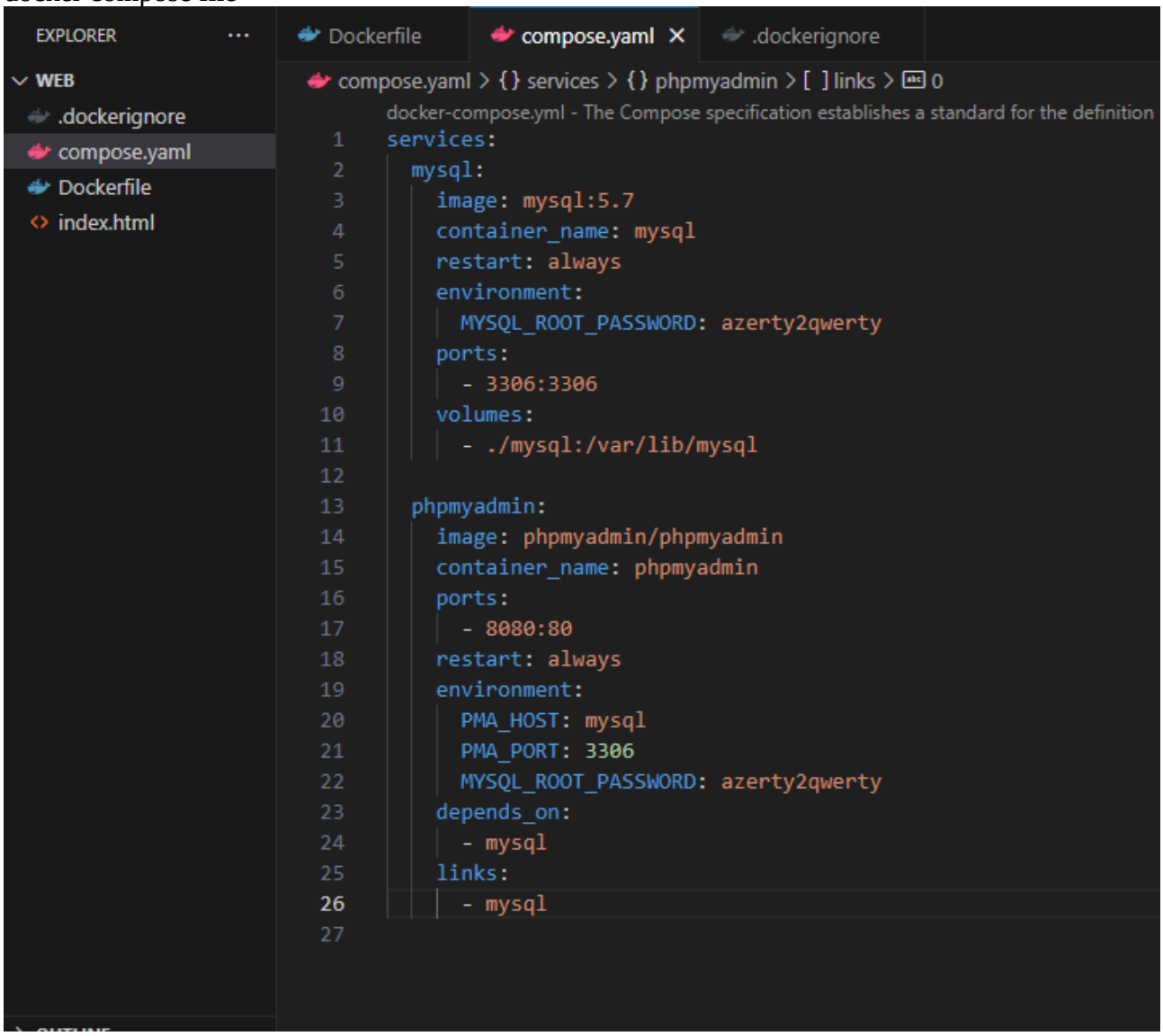


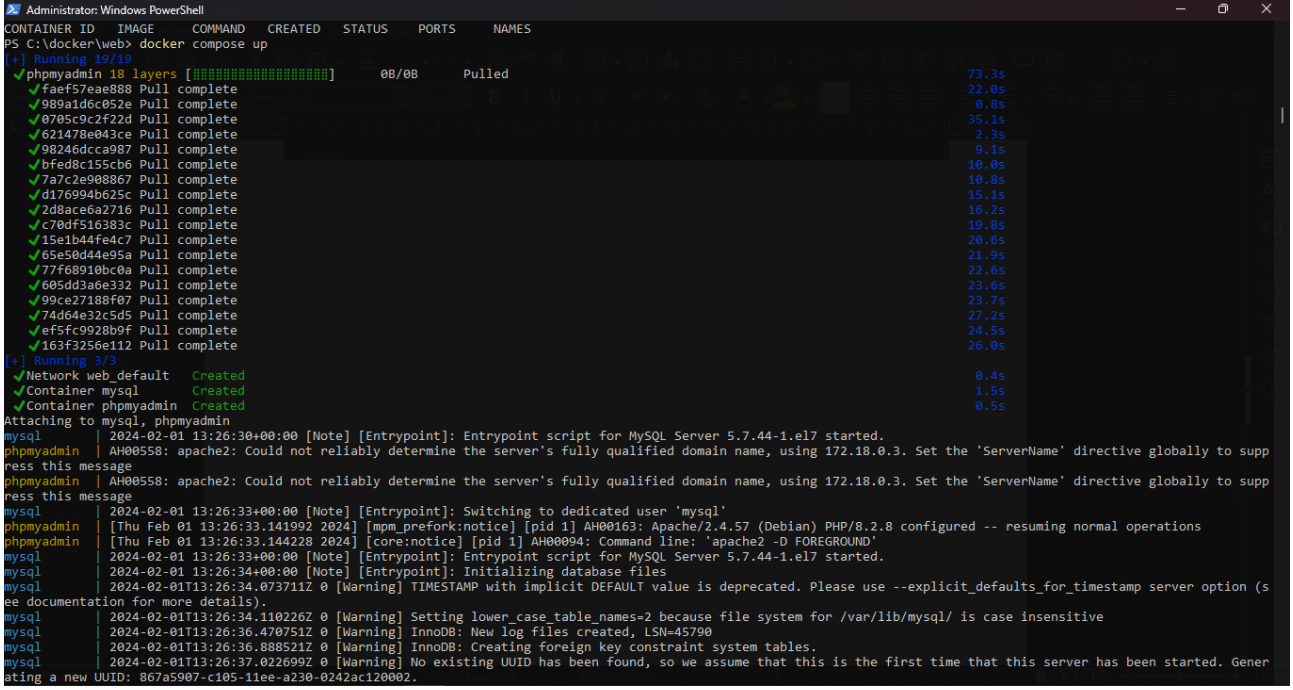
8) pareil avec docker compose

suppression des conteneurs

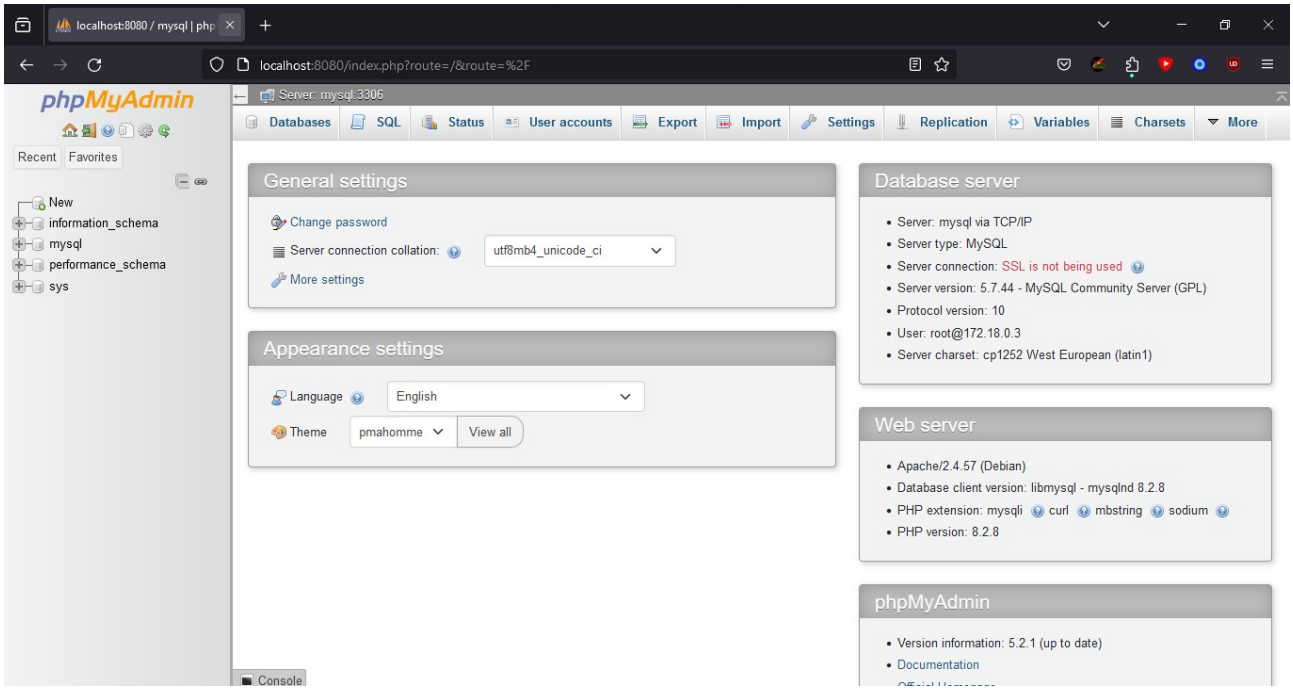
```
PS C:\docker\web> docker container ls -a
CONTAINER ID   IMAGE        COMMAND                  CREATED          STATUS          PORTS                    NAMES
d7500be9b6d6   phpmyadmin   "/docker-entrypoint.…"   17 minutes ago   Up 17 minutes   0.0.0.0:8080->80/tcp   phpmyadmin
476afde42116   mysql:5.7    "docker-entrypoint.s…"   25 minutes ago   Up 25 minutes   3306/tcp, 33060/tcp    mysql
PS C:\docker\web> docker stop d75 476
d75
476
PS C:\docker\web> docker rm d75 476
d75
476
PS C:\docker\web> docker container ls -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
PS C:\docker\web>
```

docker compose file





tout fonctionne



a) docker compose est intéressant car il permet de facilement sauvegarder et déployer toute une infrastructure plutôt qu'un seul conteneur unique.

b) les variables d'environnement permettent de configurer l'utilisateur (MYSQL_USER), la première base (MYSQL_DATABASE), le mot de passe root (MYSQL_ROOT_PASSWORD), etc...

9) observation de l'isolation reseau entre 3 conteneurs

a)
fichier docker compose :

```yaml
services:
  web:
    image: wbitt/network-multitool
    networks:
      - frontend
    command: sleep 3600
  app:
    image: wbitt/network-multitool
    networks:
      - frontend
      - backend
    command: sleep 3600
  db:
    image: wbitt/network-multitool
    networks:
      - backend
    command: sleep 3600
networks:
  frontend:
  backend:
```

tests des différents réseaux :

```
Administrator: Windows PowerShell
PS C:\docker\web> docker-compose exec web ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
16: eth0@if17: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:12:00:03 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.18.0.3/16 brd 172.18.255.255 scope global eth0
       valid_lft forever preferred_lft forever
PS C:\docker\web> docker-compose exec app ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
12: eth1@if13: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:12:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.18.0.2/16 brd 172.18.255.255 scope global eth1
       valid_lft forever preferred_lft forever
18: eth0@if19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:13:00:03 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.19.0.3/16 brd 172.19.255.255 scope global eth0
       valid_lft forever preferred_lft forever
PS C:\docker\web> docker-compose exec db ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
14: eth0@if15: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:13:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.19.0.2/16 brd 172.19.255.255 scope global eth0
       valid_lft forever preferred_lft forever
PS C:\docker\web>
```

web sur 172.18
app sur 172.18 et 172.19
db sur 172.19

```
PS C:\docker\web> docker-compose exec web ping app
PING app (172.18.0.2) 56(84) bytes of data.
64 bytes from web-app-1.web_frontend (172.18.0.2): icmp_seq=1 ttl=64 time=0.193 ms
^C
--- app ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.193/0.193/0.193/0.000 ms
PS C:\docker\web> docker-compose exec web ping db
ping: db: Try again
PS C:\docker\web> docker-compose exec db ping app
PING app (172.19.0.3) 56(84) bytes of data.
64 bytes from web-app-1.web_backend (172.19.0.3): icmp_seq=1 ttl=64 time=2.81 ms
^C
--- app ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 3ms
rtt min/avg/max/mdev = 2.814/2.814/2.814/0.000 ms
PS C:\docker\web>
```

web ne peut pas ping db mais db et web peuvent ping app

b)
la ligne NetworkID dans NetworkSettings justifie ce comportement

NetworkID de web : (réseau frontend)


```
    },
    "NetworkSettings": {
        "Bridge": "",
        "SandboxID": "d598b82616913ae7db271c5191b72c4ab0c8ad6bed06d05c52811360c4bfa529",
        "HairpinMode": false,
        "LinkLocalIPv6Address": "",
        "LinkLocalIPv6PrefixLen": 0,
        "Ports": {},
        "SandboxKey": "/var/run/docker/netns/d598b8261691",
        "SecondaryIPAddresses": null,
        "SecondaryIPv6Addresses": null,
        "EndpointID": "",
        "Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "IPAddress": "",
        "IPPrefixLen": 0,
        "IPv6Gateway": "",
        "MacAddress": "",
        "Networks": {
            "web_frontend": {
                "IPAMConfig": null,
                "Links": null,
                "Aliases": [
                    "web-web-1",
                    "web",
                    "6d93b69fc295"
                ],
                "MacAddress": "",
                "NetworkID": "64f247568bc40420973976220287f61dc13f2b6ff9e6dbd6e1c636a6d4a4489f",
                "EndpointID": "",
                "Gateway": "",
                "IPAddress": "",
                "IPPrefixLen": 0,
                "IPv6Gateway": "",
                "GlobalIPv6Address": "",
                "GlobalIPv6PrefixLen": 0,
                "DriverOpts": null
            }
        }
    }
}
```


NetworkID de app (2 network id qui correspondent au réseau frontend et backend)

```
        "Networks": {
            "web_backend": {
                "IPAMConfig": null,
                "Links": null,
                "Aliases": [
                    "web-app-1",
                    "app",
                    "0b3354b9ebf5"
                ],
                "MacAddress": "",
                "NetworkID": "157d3b83747ec3bf5db97a7141367963ee2ec5bbb8db2a338dfd64c677fbab53",
                "EndpointID": "",
                "Gateway": "",
                "IPAddress": "",
                "IPPrefixLen": 0,
                "IPv6Gateway": "",
                "GlobalIPv6Address": "",
                "GlobalIPv6PrefixLen": 0,
                "DriverOpts": null
            },
            "web_frontend": {
                "IPAMConfig": null,
                "Links": null,
                "Aliases": [
                    "web-app-1",
                    "app",
                    "0b3354b9ebf5"
                ],
                "MacAddress": "",
                "NetworkID": "64f247568bc40420973976220287f61dc13f2b6ff9e6dbd6e1c636a6d4a4489f",
                "EndpointID": "",
                "Gateway": "",
                "IPAddress": "",
                "IPPrefixLen": 0,
                "IPv6Gateway": "",
                "GlobalIPv6Address": "",
                "GlobalIPv6PrefixLen": 0,
                "DriverOpts": null
            }
        }
    }
}
```


NetworkID de db : (réseau backend)


```
},
"NetworkSettings": {
    "Bridge": "",
    "SandboxID": "22becd2ba7c9eadc9fdbb9d58e45727aee1203dd53811a2d01db7bb135b804af",
    "HairpinMode": false,
    "LinkLocalIPv6Address": "",
    "LinkLocalIPv6PrefixLen": 0,
    "Ports": {},
    "SandboxKey": "/var/run/docker/netns/22becd2ba7c9",
    "SecondaryIPAddresses": null,
    "SecondaryIPv6Addresses": null,
    "EndpointID": "",
    "Gateway": "",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "IPAddress": "",
    "IPPrefixLen": 0,
    "IPv6Gateway": "",
    "MacAddress": "",
    "Networks": {
        "web_backend": {
            "IPAMConfig": null,
            "Links": null,
            "Aliases": [
                "web-db-1",
                "db",
                "af6ba8588650"
            ],
            "MacAddress": "",
            "NetworkID": "157d3b83747ec3bf5db97a7141367963ee2ec5bbb8db2a338dfd64c677fbab53",
            "EndpointID": "",
            "Gateway": "",
            "IPAddress": "",
            "IPPrefixLen": 0,
            "IPv6Gateway": "",
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "DriverOpts": null
        }
    }
}
```

c)
on pourrait vouloir cette configuration réseau pour installer un pare feu sur app afin de protéger les données de la base de données

on pourrait utiliser nginx pour le conteneur web, node pour le conteneur app et mysql pour le conteneur db.