TP4 – Les volumes Docker

Brahim HAMDI

1. Arrêtez et supprimez tous les conteneurs crées précédemment.

```
vagrant@Manager:~/demoapp$ docker stop $(docker ps -a -q)
914b637f7eb8
vagrant@Manager:~/demoapp$
vagrant@Manager:~/demoapp$ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
914b637f7eb81dc658acee894345577db021083805b3a6fc6b3c56152aa4c64a

Total reclaimed space: 0B
vagrant@Manager:~/demoapp$ _
```

2. Créez un volume nommé « vol1 », puis affichez des informations détaillées sur ce volume.

```
vagrant@Manager:~/demoapp$ docker volume ls
DRIVER      VOLUME NAME
vagrant@Manager:~/demoapp$
vagrant@Manager:~/demoapp$ docker volume create vol1
vol1
vagrant@Manager:~/demoapp$
vagrant@Manager:~/demoapp$ docker volume ls
DRIVER      VOLUME NAME
local       vol1
vagrant@Manager:~/demoapp$
vagrant@Manager:~/demoapp$ docker volume inspect vol1
[
    {
        "CreatedAt": "2024-10-19T15:18:05Z",
        "Driver": "local",
        "Labels": null,
        "Mountpoint": "/var/lib/docker/volumes/vol1/_data",
        "Name": "vol1",
        "Options": null,
        "Scope": "local"
    }
]
vagrant@Manager:~/demoapp$
```

Quel est le chemin du point de montage de vol1 ? Est-il vide ?

```
vagrant@Manager:~/demoapp$ sudo ls -l /var/lib/docker/volumes/vol1/_data
total 0
vagrant@Manager:~/demoapp$
```

3. Lancez un nouveau conteneur *web* qui exécute l'image *nginx* et attache le volume *vol1* au chemin */usr/share/nginx/html*.

```
vagrant@Manager:~/demoapp$ docker run -d --name web -v vol1:/usr/share/nginx/html -p 8000:80 nginx
dba902da3cc8528bac37943d3c8015f09229ba6b797dc91e26ea885c0034785b
vagrant@Manager:~/demoapp$
vagrant@Manager:~/demoapp$ docker ps
CONTAINER ID   IMAGE   COMMAND              CREATED        STATUS         PORTS                  NAMES
dba902da3cc8   nginx   "/docker-entrypoint.…"   4 seconds ago  Up 2 seconds   0.0.0.0:8000->80/tcp   web
vagrant@Manager:~/demoapp$
vagrant@Manager:~/demoapp$ docker inspect web | grep -A5 vol1
                "vol1:/usr/share/nginx/html"
            ],
            "ContainerIDFile": "",
            "LogConfig": {
                "Type": "json-file",
                "Config": {}
--
                "Name": "vol1",
                "Source": "/var/lib/docker/volumes/vol1/_data",
                "Destination": "/usr/share/nginx/html",
                "Driver": "local",
                "Mode": "z",
                "RW": true,
                "Propagation": ""
vagrant@Manager:~/demoapp$
```

Le volume *vol1* est-il vide ?

```
vagrant@Manager:~/demoapp$ sudo ls -l /var/lib/docker/volumes/vol1/_data
total 8
-rw-r--r-- 1 root root 497 Oct  2 15:13 50x.html
-rw-r--r-- 1 root root 615 Oct  2 15:13 index.html
vagrant@Manager:~/demoapp$
```

4. À l'aide de *vol1*, modifiez le fichier *index.html* afin d'afficher le contenu suivant :
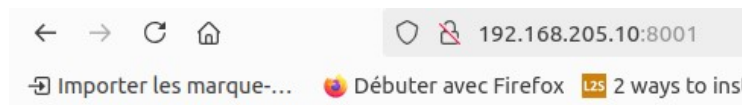*Formation Docker*
[Votre Nom]

```
vagrant@Manager:~/demoapp$ sudo nano /var/lib/docker/volumes/vol1/_data/index.html
vagrant@Manager:~/demoapp$
vagrant@Manager:~/demoapp$ sudo cat /var/lib/docker/volumes/vol1/_data/index.html
<h1> Formation Docker </h1> <br>
<b>Brahim HAMDI</b>
vagrant@Manager:~/demoapp$
```


Formation Docker

Brahim HAMDI

5. Lancez un autre conteneur *web2* qui exécute l'image *nginx* et partage le même volume *vol1*.

```
vagrant@Manager:~/demoapp$ docker run -d --name web2 -v vol1:/usr/share/nginx/html -p 8001:80 nginx
48ace45a7e330a853aeb55bb49aa7a8f8b1047f0d0f03112004b636b5f07701c
vagrant@Manager:~/demoapp$
vagrant@Manager:~/demoapp$ docker ps
CONTAINER ID   IMAGE   COMMAND              CREATED        STATUS         PORTS                  NAMES
48ace45a7e33   nginx   "/docker-entrypoint.…"   4 seconds ago  Up 3 seconds   0.0.0.0:8001->80/tcp   web2
dba902da3cc8   nginx   "/docker-entrypoint.…"   12 minutes ago Up 12 minutes  0.0.0.0:8000->80/tcp   web
vagrant@Manager:~/demoapp$ _
```

Qu'affiche le conteneur *web2* sur sa page Web ? Est-il le même contenu affiché par le conteneur *web* ?

Formation Docker

Brahim HAMDI

6. Lancez un conteneur nommé *db* qui exécute l'image *mariadb* ,puis listez les volumes.

```
vagrant@Manager:~/demoapp$ docker volume ls
DRIVER    VOLUME NAME
local     vol1
vagrant@Manager:~/demoapp$
vagrant@Manager:~/demoapp$ docker run -d --name wiki_db -e MARIADB_ROOT_PASSWORD="secret" mariadb
2c3dc2ff4ed952c0d6e4a42e3c64f69681c4cb3758b418e475833bd4af9da029
vagrant@Manager:~/demoapp$
vagrant@Manager:~/demoapp$ docker ps
CONTAINER ID   IMAGE     COMMAND                 CREATED          STATUS          PORTS                    NAMES
2c3dc2ff4ed9   mariadb   "docker-entrypoint.s…"  7 seconds ago    Up 5 seconds    3306/tcp                 wiki_db
48ace45a7e33   nginx     "/docker-entrypoint.…"  5 minutes ago    Up 5 minutes    0.0.0.0:8001->80/tcp     web2
dba902da3cc8   nginx     "/docker-entrypoint.…"  18 minutes ago   Up 18 minutes   0.0.0.0:8000->80/tcp     web
vagrant@Manager:~/demoapp$
vagrant@Manager:~/demoapp$ docker volume ls
DRIVER    VOLUME NAME
local     d355dbf14fa6c4e46e6867494c87536835a641513027da099ca69d31d661d3ec
local     vol1
vagrant@Manager:~/demoapp$
```

○ Inspectez le nouveau volume. D'où vient-il ?

```
vagrant@Manager:~/demoapp$ docker volume inspect d355dbf14fa6c4e46e6867494c87536835a641513027da099ca69d31d661d3ec
[
    {
        "CreatedAt": "2024-10-19T15:41:08Z",
        "Driver": "local",
        "Labels": {
            "com.docker.volume.anonymous": ""
        },
        "Mountpoint": "/var/lib/docker/volumes/d355dbf14fa6c4e46e6867494c87536835a641513027da099ca69d31d661d3ec/_data",
        "Name": "d355dbf14fa6c4e46e6867494c87536835a641513027da099ca69d31d661d3ec",
        "Options": null,
        "Scope": "local"
    }
]
vagrant@Manager:~/demoapp$
```

7. Supprimez tous les volumes. Y a-t-il une erreur ? Pourquoi ?

```
vagrant@Manager:~/demoapp$ docker volume ls
DRIVER    VOLUME NAME
local     d355dbf14fa6c4e46e6867494c87536835a641513027da099ca69d31d661d3ec
local     vol1
vagrant@Manager:~/demoapp$
vagrant@Manager:~/demoapp$ docker volume rm d355dbf14fa6c4e46e6867494c87536835a641513027da099ca69d31d661d3ec vol1
Error response from daemon: remove d355dbf14fa6c4e46e6867494c87536835a641513027da099ca69d31d661d3ec: volume is in use - [2c3dc2f
f4ed952c0d6e4a42e3c64f69681c4cb3758b418e475833bd4af9da029]
Error response from daemon: remove vol1: volume is in use - [dba902da3cc8528bac37943d3c8015f09229ba6b797dc91e26ea885c0034785b, 4
8ace45a7e330a853aeb55bb49aa7a8f8b1047f0d0f03112004b636b5f07701c]
vagrant@Manager:~/demoapp$
```

○ Arrêtez puis supprimez tous les conteneurs. Les volumes sont-ils supprimés aussi ?

```
vagrant@Manager:~/demoapp$ docker ps
CONTAINER ID   IMAGE     COMMAND               CREATED          STATUS          PORTS                    NAMES
2c3dc2ff4ed9   mariadb   "docker-entrypoint.s…" 7 minutes ago    Up 7 minutes    3306/tcp                 wiki_db
48ace45a7e33   nginx     "/docker-entrypoint.…" 13 minutes ago   Up 13 minutes   0.0.0.0:8001->80/tcp     web2
dba902da3cc8   nginx     "/docker-entrypoint.…" 25 minutes ago   Up 25 minutes   0.0.0.0:8000->80/tcp     web
vagrant@Manager:~/demoapp$
vagrant@Manager:~/demoapp$ docker stop $(docker ps -a -q)
2c3dc2ff4ed9
48ace45a7e33
dba902da3cc8
vagrant@Manager:~/demoapp$
vagrant@Manager:~/demoapp$ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
2c3dc2ff4ed952c0d6e4a42e3c64f69681c4cb3758b418e475833bd4af9da029
48ace45a7e330a853aeb55bb49aa7a8f8b1047f0d0f03112004b636b5f07701c
dba902da3cc8528bac37943d3c8015f09229ba6b797dc91e26ea885c0034785b

Total reclaimed space: 2.186kB
vagrant@Manager:~/demoapp$
vagrant@Manager:~/demoapp$ docker ps
CONTAINER ID   IMAGE     COMMAND     CREATED     STATUS     PORTS     NAMES
vagrant@Manager:~/demoapp$
vagrant@Manager:~/demoapp$ docker ps -a
CONTAINER ID   IMAGE     COMMAND     CREATED     STATUS     PORTS     NAMES
vagrant@Manager:~/demoapp$
```

```
vagrant@Manager:~/demoapp$ docker volume ls
DRIVER     VOLUME NAME
local      d355dbf14fa6c4e46e6867494c87536835a641513027da099ca69d31d661d3ec
local      vol1
vagrant@Manager:~/demoapp$
```

○ Supprimez tous les volumes non utilisés.

```
vagrant@Manager:~/demoapp$ docker volume ls
DRIVER     VOLUME NAME
local      d355dbf14fa6c4e46e6867494c87536835a641513027da099ca69d31d661d3ec
local      vol1
vagrant@Manager:~/demoapp$
vagrant@Manager:~/demoapp$ docker volume prune -a
WARNING! This will remove all local volumes not used by at least one container.
Are you sure you want to continue? [y/N] y
Deleted Volumes:
vol1
d355dbf14fa6c4e46e6867494c87536835a641513027da099ca69d31d661d3ec

Total reclaimed space: 155.9MB
vagrant@Manager:~/demoapp$
vagrant@Manager:~/demoapp$ docker volume ls
DRIVER     VOLUME NAME
vagrant@Manager:~/demoapp$
```