Master II BDCC- Big Data et Cloud Computing

TP 4: Docker (Network, Storage, Sécurité, Compose, Swarm, Kubernetes)

PARTIE 1 : Network, Storage, Sécurité, Compose

1. Lister les réseaux installer par défaut avec Dockerengine.

```
vm10@vm1:~$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:d9:b8:d6:70  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.33  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fd80::a00:27ff:fed9:ca0c  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::a00:27ff:fed9:ca0c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:d9:ca:0c  txqueuelen 1000  (Ethernet)
        RX packets 28601  bytes 40221921 (40.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13715  bytes 933827 (933.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 116  bytes 9552 (9.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 116  bytes 9552 (9.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
C:\Windows\system32>docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
3cc0cf910bb3        bridge              bridge              local
425a3ccdf4b9        host                host                local
782a77eb33e4        none                null                local
```

2. Inspecter le réseau Bridge.

```
C:\Windows\system32>docker inspect bridge
[
    {
        "Name": "bridge",
        "Id": "3cc0cf910bb3ecfde93a9feee08eb41d9180361045e43d77a601d9af9b1e8541",
        "Created": "2021-11-28T09:10:00.919025199Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16",
                    "Gateway": "172.17.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {},
        "Options": {
```

3. Déconnecter le conteneur 1 (TP2.PARTIE3.2) du Bridge.

```
C:\Windows\system32>docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                      PORTS
                    NAMES
5d0c3a372b44        webapptp2           "/docker-entrypoint.…"   6 days ago          Exited (255) 21 minutes ago   0.0.
.0:8081->80/tcp, :::8081->80/tcp   webapptp2
1e10432a21ab        webapp              "/docker-entrypoint.…"   13 days ago         Exited (0) 13 days ago
                    elated_ganguly
fc399c1c410ba       webapp              "/docker-entrypoint.…"   3 weeks ago         Exited (255) 13 days ago      0.0.
.0:8081->80/tcp, :::8081->80/tcp   exciting_poincare
f5b79c27c0c8        adminer             "entrypoint.sh docke…"   3 weeks ago         Exited (0) 13 days ago
                    tp2-adminer
b91c7c325b5c        postgres            "docker-entrypoint.s…"   3 weeks ago         Exited (255) 7 days ago       5432
tcp                 tp2bddpostgres
7821065cb701        phpmyadmin          "/docker-entrypoint.…"   3 weeks ago         Exited (0) 3 weeks ago
                    tp2-phpmyadmin
7dcb591b12dd        phpmyadmin          "/docker-entrypoint.…"   3 weeks ago         Exited (0) 3 weeks ago
                    tp2myadmin
093e4e514029        mysql:latest        "docker-entrypoint.s…"   3 weeks ago         Exited (0) 3 weeks ago
                    tp2bdd
982b2d7d45b3        phpmyadmin          "/docker-entrypoint.…"   4 weeks ago         Exited (255) 3 weeks ago      0.0.
.0:8080->80/tcp, :::8080->80/tcp   tp2phpmyadmin
adc4a5cb6721        mysql:latest        "docker-entrypoint.s…"   4 weeks ago         Exited (255) 3 weeks ago      3306
tcp, 33060/tcp      tp2mysql

C:\Windows\system32>docker start fc39
```

```
C:\Windows\system32>docker network disconnect bridge fc39
```

4. Connecter à nouveau ce conteneur au réseau Bridge.

```
C:\Windows\system32>docker network connect bridge fc39
```

5. Créer un nouveau réseau du type bridge (nom : my_bridge); affecter ce réseau au conteneur 1.

```
C:\Windows\system32>docker network create -d bridge my_bridge
9bf63b1126551d3d3497965741c0d9f480816b62feb5e8654c8d20ceb80161e2
```

```
C:\Windows\system32>docker network disconnect bridge fc39

C:\Windows\system32>docker network connect my_bridge fc39
```

6. Afficher l'adresse IP de votre conteneur 1.

```
C:\Windows\system32>docker inspect fc39
```

7. Créer et exécuter une image (TP2.PARTIE3.2), dans un conteneur 2 avec un port différent du conteneur 1.

```
C:\Windows\system32>docker run -d -p 8082:80 webapp
fb6f2eeabf490bf8c797166a1fd4ddae038081a090e331e2b0b9704e8f3e25bf
```

8. Afficher l'adresse IP de votre conteneur 2.

```
C:\Windows\system32>docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
                    NAMES
fb6f2eeabf49        webapp              "/docker-entrypoint.…"   10 seconds ago      Up 8 seconds        0.0.0.0:8082->8
0/tcp, :::8082->80/tcp   condescending_mahavira
fc3991c410ba        webapp              "/docker-entrypoint.…"   3 weeks ago         Up 22 minutes       0.0.0.0:8081->8
0/tcp, :::8081->80/tcp   exciting_poincare

C:\Windows\system32>docker inspect fb6f
```

9. Accéder en mode interactive (bash) au conteneur 1, tester un ping vers le conteneur 2.

```
C:\Windows\system32>docker container exec -it fc39 bash
root@fc3991c410ba:/#
```

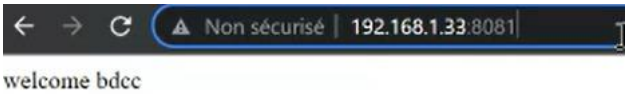
```
bash: ping: command not found
root@fc3991c410ba:/# apt-get update
```

```
root@fc3991c410ba:/# apt install net-tools
```

```
root@fc3991c410ba:/# apt-get update && apt-get install -y iputils-ping
```

Ping @ conteneur 2

10. Accéder depuis la machine hôte au URL de la machine virtuel avec les ports des conteneurs 1 et 2.



welcome bdcc

11. Affecter le réseau my_bridge au conteneur 2. Afficher l'adresse IP de votre conteneur 2.

    Docker network disconnect bridge fb69

    Docker network connect my-bridge fb69

    Docker inspect fb69

12. Accéder en mode interactive (bash) au conteneur 1, tester à nouveau un ping vers le conteneur 2.

```
C:\Windows\system32>docker container exec -it fc39 bash
root@fc3991c410ba:/#
```

    Ping @ conteneur 2

13. Créer un nouveau volume de stockage my-vol.

```
C:\Windows\system32>docker volume create my-vol
my-vol
```

14. Lister les volumes disponibles.

```
C:\Windows\system32>docker volume ls
```

15. Inspecter le volume my-vol.

```
C:\Windows\system32>docker volume inspect my-vol
[
    {
        "CreatedAt": "2021-11-28T11:03:27Z",
        "Driver": "local",
        "Labels": {},
        "Mountpoint": "/var/lib/docker/volumes/my-vol/_data",
        "Name": "my-vol",
        "Options": {},
        "Scope": "local"
    }
]
```

16. Associer ce volume avec un conteneur 3 exécutant l'image (TP2.PARTIE3.2) en précisant comme source de stockage my-vol et comme destination /usr/share/nginx/html.

```
C:\Windows\system32>docker run -d -v my-vol:/usr/share/nginx/html webapp
a5d0f2bb9bbe44c1f9a18ad39022fdd02824a97816d6397894c8c31c7d2658f6
```

17. Inspecter le conteneur 3 afin d'afficher le volume utilisé.

```
C:\Windows\system32>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS
               NAMES
a5d0f2bb9bbe   webapp    "/docker-entrypoint.…"   About a minute ago   Up About a minute   80/tcp
               nifty_austin
e84ccfdf6c95   webapp    "/docker-entrypoint.…"   About an hour ago    Up About an hour
               dreamy_bhabha
fb6f2eeabf49   webapp    "/docker-entrypoint.…"   About an hour ago    Up About an hour    0.0.0.0:8082->
80/tcp, :::8082->80/tcp   condescending_mahavira
fc3991c410ba   webapp    "/docker-entrypoint.…"   3 weeks ago          Up 2 hours          0.0.0.0:8081->
80/tcp, :::8081->80/tcp   exciting_poincare

C:\Windows\system32>docker inspect a5d0
```

```
"Mounts": [
    {
        "Type": "volume",
        "Name": "my-vol",
        "Source": "/var/lib/docker/volumes/my-vol/_data",
        "Destination": "/usr/share/nginx/html",
        "Driver": "local",
        "Mode": "z",
        "RW": true,
        "Propagation": ""
    }
```

18. Créer le dossier test sur conteneur 3, vérifier sa présence sur le volume my-vol ; créer le dossier test 2 sur volume my-vol vérifier est ce qu'il y a un changement sur le conteneur.

```
C:\Windows\system32>docker container exec -it a5d0 bash
root@a5d0f2bb9bbe:/# ls
bin   dev                  docker-entrypoint.sh  home  lib64  mnt  proc  run   srv  tmp  var
boot  docker-entrypoint.d  etc                   lib   media  opt  root  sbin  sys  usr
root@a5d0f2bb9bbe:/# ls usr
bin  games  include  lib  local  sbin  share  src
root@a5d0f2bb9bbe:/# ls share
ls: cannot access 'share': No such file or directory
root@a5d0f2bb9bbe:/# cd usr
root@a5d0f2bb9bbe:/usr# ls
bin  games  include  lib  local  sbin  share  src
root@a5d0f2bb9bbe:/usr# cd share
root@a5d0f2bb9bbe:/usr/share# ls
X11          bash-completion  debconf      doc-base    gcc-8  keyrings  man    pam          polkit-1        terminfo
adduser      bug              debianutils  dpkg        gdb    libc-bin  menu   pam-configs  readline        xml
base-files   ca-certificates  dict         fontconfig  info   lintian   misc   perl5        sensible-utils  zoneinfo
base-passwd  common-licenses  doc          fonts       java   locale    nginx  pixmaps      tabset          zsh
root@a5d0f2bb9bbe:/usr/share# cd nginx
root@a5d0f2bb9bbe:/usr/share/nginx# ls
html
root@a5d0f2bb9bbe:/usr/share/nginx# cd html
root@a5d0f2bb9bbe:/usr/share/nginx/html# ls
50x.html  index.html
root@a5d0f2bb9bbe:/usr/share/nginx/html# nano index.html
bash: nano: command not found
root@a5d0f2bb9bbe:/usr/share/nginx/html# mkdir test
root@a5d0f2bb9bbe:/usr/share/nginx/html# ls
50x.html  index.html  test
```

```
root@vm1:/home/vm1# cd /var/lib/docker/volumes/my-vol/_data
root@vm1:/var/lib/docker/volumes/my-vol/_data# ls
50x.html  index.html  test
root@vm1:/var/lib/docker/volumes/my-vol/_data# mkdir test2
root@vm1:/var/lib/docker/volumes/my-vol/_data# ls
50x.html  index.html  test  test2
root@vm1:/var/lib/docker/volumes/my-vol/_data# ls
50x.html  index.html  test  test2
root@vm1:/var/lib/docker/volumes/my-vol/_data# cd..
cd..: command not found
root@vm1:/var/lib/docker/volumes/my-vol/_data# cd /var/lib/docker/volumes/my-vol/
root@vm1:/var/lib/docker/volumes/my-vol# ls
_data
root@vm1:/var/lib/docker/volumes/my-vol# cd /var/lib/docker/volumes/
root@vm1:/var/lib/docker/volumes# ls
backingFsBlockDev
metadata.db
my-vol
root@vm1:/var/lib/docker/volumes#
```

```
root@a5d0f2bb9bbe:/usr/share/nginx/html# ls
50x.html  index.html  test  test2
root@a5d0f2bb9bbe:/usr/share/nginx/html# exit
exit
```

19. Supprimer le conteneur 3 ainsi que le volume my-vol.

```
C:\Windows\system32>docker stop a5d0
```

```
C:\Windows\system32>docker rm a5d0
a5d0

C:\Windows\system32>docker volume rm my-vol
```

20. Ajouter quelques fonctionnalités de sécurité au conteneur 3 (Seccomp, Apparmor). Tester l'efficacité de ce conteneur.

```
$ docker run --rm \
             -it \
             --security-opt seccomp=/path/to/seccomp/profile.json \
             hello-world
```

Understand the policies

The `docker-default` profile is the default for running containers. It is moderately protective while providing wide application compatibility. The profile is generated from the following template.

When you run a container, it uses the `docker-default` policy unless you override it with the `security-opt` option. For example, the following explicitly specifies the default policy:

```
$ docker run --rm -it --security-opt apparmor=docker-default hello-world
```

Load and unload profiles

To load a new profile into AppArmor for use with containers:

```
$ apparmor_parser -r -W /path/to/your_profile
```

Then, run the custom profile with `--security-opt` like so:

```
$ docker run --rm -it --security-opt apparmor=your_profile hello-world
```

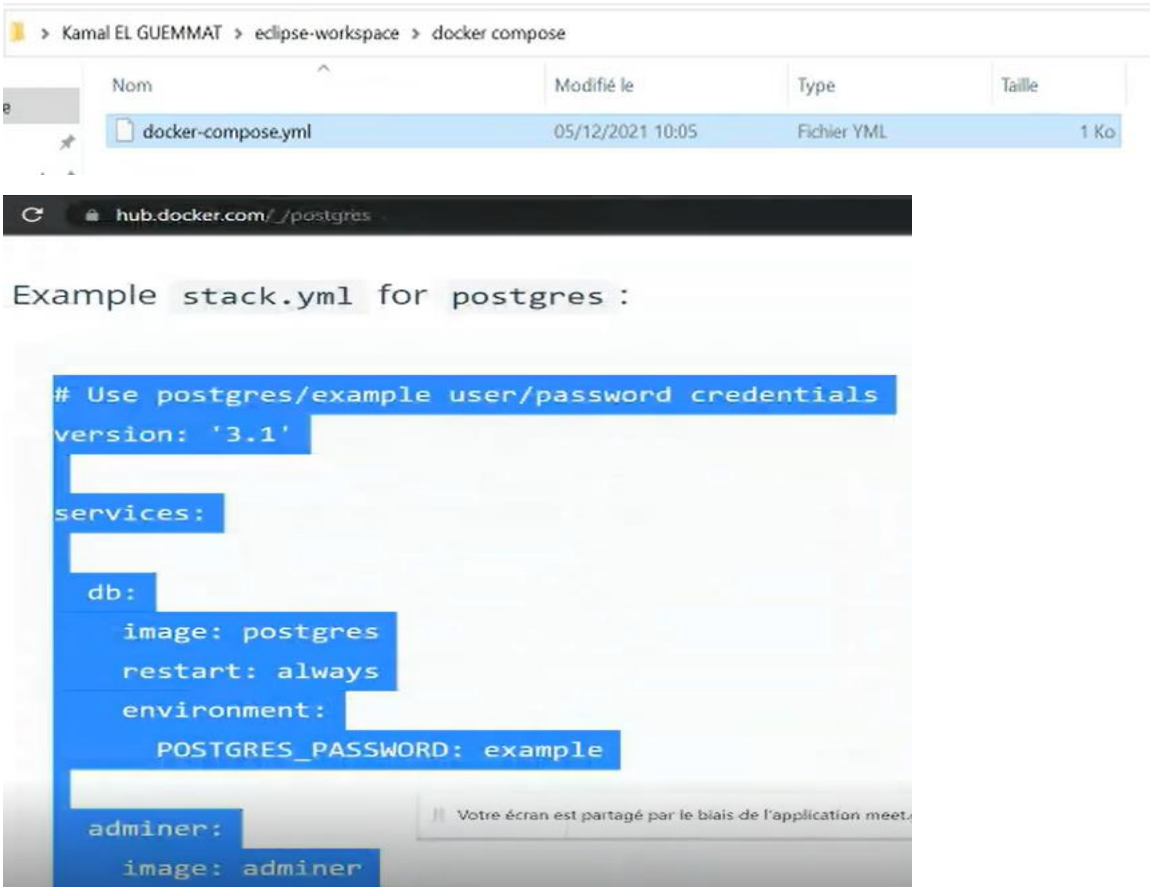
21. Installer docker compose.

```
root@vm1:/home/vm1# docker-compose --version
Command 'docker-compose' not found, but can be installed with:
snap install docker          # version 20.10.8, or
apt  install docker-compose  # version 1.25.0-1
See 'snap info docker' for additional versions.
root@vm1:/home/vm1# apt install docker-compose
```

```
PS C:\Windows\system32> choco install docker-compose
```

```
PS C:\Windows\system32> docker-compose --version
docker-compose version 1.29.2, build 5beceaac
```

22. Créer un nouveau fichier docker-compose.yml ; exécuter, à la fois, une image de BDD et une image d'administration de cette dernière sous forme de deux services.



*stack.yml - Bloc-notes

Fichier Edition Format Affichage Aide

```
# Use postgres/example user/password credentials
version: '3.1'

services:

  db:
    image: postgres
    restart: always
    environment:
      POSTGRES_PASSWORD: example

  adminer:
    image: adminer
    restart: always
    ports:
      - 8080:8080
```

```
C:\Users\Kamal EL GUEMMAT\eclipse-workspace\docker compose>docker-compose up -d
Creating network "dockercompose_default" with the default driver
Creating dockercompose_adminer_1 ... done
Creating dockercompose_db_1      ... done
```

23. Lister les conteneurs exécutés par le docker compose.

```
C:\Users\Kamal EL GUEMMAT\eclipse-workspace\docker compose>docker-compose ps
          Name                        Command               State                Ports
-------------------------------------------------------------------------------------------------------
dockercompose_adminer_1   entrypoint.sh docker-php-e ...   Up      0.0.0.0:8080->8080/tcp,:::8080->8080/tcp
dockercompose_db_1        docker-entrypoint.sh postgres    Up      5432/tcp
```

24. Lancer le système de gestion de la BDD.

25. Afficher l'état des services en cours d'exécution grâce au Docker compose.

```
C:\Users\Kamal EL GUEMMAT\eclipse-workspace\docker compose>docker-compose events
```

26. Diffuser la sortie du journal des services en cours d'exécution grâce au Docker compose.

```
C:\Users\Kamal EL GUEMMAT\eclipse-workspace\docker compose>docker-compose logs
```

27. Exécuter une commande ponctuelle sur un service grâce au Docker compose (par exemple : afficher les variables d'environnement du service BDD).

```
C:\Users\Kamal EL GUEMMAT\eclipse-workspace\docker compose>docker-compose run db env
```

28. Arrêter les services avec Docker compose.

```
C:\Users\Kamal EL GUEMMAT\eclipse-workspace\docker compose>docker-compose down
```

Ressources :

https://docs.docker.com/network/, https://docs.docker.com/storage/,
https://docs.docker.com/engine/security/, https://docs.docker.com/engine/security/seccomp/,
https://docs.docker.com/engine/security/apparmor/, https://docs.docker.com/compose/