Yann Emmanuel Brou
Sir William Ngoma

TP3 Devops

4. Application

a) Fait
b) On a choisi Redis
c) On a lancé la commande *docker pull redis* pour installer cette image depuis dockerhub
d) Sudo docker images

```
root@Ubuntu:/home/yannebrou# docker images
REPOSITORY          TAG       IMAGE ID       CREATED         SIZE
redis               latest    e579380d4317   7 days ago      138MB
hello-world         latest    9c7a54a9a43c   5 months ago    13.3kB
docker/whalesay     latest    6b362a9f73eb   8 years ago     247MB
root@Ubuntu:/home/yannebrou#
```

Nous avons 3 images de docker sur la machine (redis, whalesay et hello-world)

e) docker run redis
f) sudo docker ps ; Aucun conteneur en cours d'exécution car nous avons arrêté redis pour lancer cette commande
g) docker run -d redis
h) On va d'abord faire sudo docker ps-a pour lister tous les conteneurs. On va ensuite copier l'ID du conteneur actif avec sudo docker stop idConteneur

```
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS                      PORTS      NAMES
97ed88611501   redis     "docker-entrypoint.s…"   7 minutes ago    Up 7 minutes                6379/tcp   cranky_hypatia
277be125c1ae   redis     "docker-entrypoint.s…"   8 minutes ago    Exited (1) 8 minutes ago               elastic_euler
dfad75107d42   redis     "docker-entrypoint.s…"   27 minutes ago   Up 27 minutes               6379/tcp   sleepy_ellis
3117197cfbcb   redis     "docker-entrypoint.s…"   37 minutes ago   Exited (0) 34 minutes ago              adoring_dubinsky
258d4f6d41c3   redis     "docker-entrypoint.s…"   58 minutes ago   Exited (0) 45 minutes ago              intelligent_gould
root@UbuntuL:/home/user# docker stop 97ed88611501
97ed88611501
root@UbuntuL:/home/user#
```

i) sudo docker start 97ed88611501
j) répondu à la question h)
k) on va lancer la commande : docker pull redis/redis-stack-server
l) On va lancer le 2e conteneur avec la commande sudo docker run -d redis/redis-stack-server. Puis on affiche l'historique des conteneurs et on voit que les 2 sont en cours d'exécution.
m) On remarque que les conteneurs sont des processus indépendants et que le fait d'en lancer plusieurs n'affecte pas l'état des autres conteneurs déjà actifs;

Yann Emmanuel Brou
Sir William Ngoma

n) docker run -d -p 8000:80 redis. Puis on affiche les différents conteneurs en cours et cela donne :

```
CONTAINER ID   IMAGE                      COMMAND                  CREATED          STATUS                       PORTS
                    NAMES
ecf9fa093f81   redis                      "docker-entrypoint.s…"   15 seconds ago   Up 13 seconds                6379/tcp, 0.0.0
000->80/tcp, :::8000->80/tcp   frosty_rubin
595cf1ff663a   redis/redis-stack-server   "/entrypoint.sh"         6 hours ago      Exited (255) 8 minutes ago   6379/tcp
                    thirsty_fermi
37ec9857622b   redis/redis-stack-server   "/entrypoint.sh"         6 hours ago      Exited (130) 6 hours ago
                    boring_elion
97ed88611501   redis                      "docker-entrypoint.s…"   23 hours ago     Up 39 seconds                6379/tcp
                    cranky_hypatia
277be125c1ae   redis                      "docker-entrypoint.s…"   23 hours ago     Exited (1) 23 hours ago
                    elastic_euler
dfad75107d42   redis                      "docker-entrypoint.s…"   24 hours ago     Exited (255) 23 hours ago    6379/tcp
                    sleepy_ellis
8117197cfbcb   redis                      "docker-entrypoint.s…"   24 hours ago     Exited (0) 24 hours ago
                    adoring_dubinsky
258d4f6d41c3   redis                      "docker-entrypoint.s…"   24 hours ago     Exited (0) 24 hours ago
                    intelligent_gould
```

5. Initiation à Dockerfile

a) docker pull alpine

```
root@UbuntuL:/home/user# docker run -ti --rm alpine:latest /bin/sh
/ # git --version
/bin/sh: git: not found
/ # ^C

/ # ^C

/ # su root
/ # ^C

/ # ocker run -ti --rm alpine:latest
ash: ocker: not found
/ # docker run -ti --rm alpine:latest
ash: docker: not found
/ # wget
BusyBox v1.36.1 (2023-07-27 17:12:24 UTC) multi-call binary.

Usage: wget [-cqS] [--spider] [-O FILE] [-o LOGFILE] [--header STR]
        [--post-data STR | --post-file FILE] [-Y on/off]
        [-P DIR] [-U AGENT] [-T SEC] URL...

Retrieve files via HTTP or FTP

        --spider        Only check URL existence: $? is 0 if exists
        --header STR    Add STR (of form 'header: value') to headers
        --post-data STR Send STR using POST method
        --post-file FILE        Send FILE using POST method
        -c              Continue retrieval of aborted transfer
        -q              Quiet
        -P DIR          Save to DIR (default .)
        -S              Show server response
        -T SEC          Network read timeout is SEC seconds
        -O FILE         Save to FILE ('-' for stdout)
        -o LOGFILE      Log messages to FILE
        -U STR          Use STR for User-Agent header
        -Y on/off       Use proxy
/ #
```

b)

Yann Emmanuel Brou
Sir William Ngoma

Git n'est pas dans alpine, c'est également la case de vim et wget

c) On va choisir un emplacement sur le pc pour lui écrire les infos suivantes de la question. On a choisi d'appeler le fichier Dockerfile
d) sudo docker build -t Dockerfile .

```
root@UbuntuL:/home/user# ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
root@UbuntuL:/home/user# cd Desktop
root@UbuntuL:/home/user/Desktop# ls
root@UbuntuL:/home/user/Desktop# ls
root@UbuntuL:/home/user/Desktop# mkdir Tp Docker
root@UbuntuL:/home/user/Desktop# ls
Docker  Tp
root@UbuntuL:/home/user/Desktop# cd Docker Tp
bash: cd: too many arguments
root@UbuntuL:/home/user/Desktop# cd Docker
root@UbuntuL:/home/user/Desktop/Docker# ls
root@UbuntuL:/home/user/Desktop/Docker# nano Dockerfile
```

A la fin de l'exécution de la commande on a le message suivant :

```
Executing busybox-1.36.1-r2.trigger
OK: 49 MiB in 29 packages
Removing intermediate container 05a4f2149686
 ---> cb7b31032d36
Successfully built cb7b31032d36
Successfully tagged dockerfile:latest
```

e) docker run -ti –rm Dockerfile.

```
root@UbuntuL:/home/user/Desktop/Docker# docker run -ti --rm dockerfile
/ # docker ps -a
/bin/sh: docker: not found
/ # ls
bin   dev   etc   home  lib   media mnt   opt   proc  root  run   sbin  srv   sys   tmp   usr   var
```

f) Une fois la commande lancée on remarque que des sous fichiers ont été créés pour le dockerfile et que le document texte s'est transformé en fichier !
g) Fait. Pour cela on va faire un nano Dockerfile pour pouvoir modifier le contenu du fichier
h) Il n'y a aucune différence
i) On remarque qu'à l'étape 4 Run est lancé et la fonction echo permets d'afficher Hello RUN. Puis l'étape 5 CMD est lancée et on affiche une message à l'aide d'echo à ce moment.

```
FROM alpine:latest
MAINTAINER Badre BOUSALEM b.bousalem@enpc.fr

#installer git et vim
RUN apk update&&  \
    apk add git&&  \
    apk add vim

RUN echo "Hello RUN"
CMD echo "Hello CMD"
COPY ./Dockerfile ../Dockerfile
```

Yann Emmanuel Brou
Sir William Ngoma

```
Step 4/5 : RUN echo "Hello RUN"
 ---> Running in 6005b3f3afae
Hello RUN
Removing intermediate container 6005b3f3afae
 ---> 94154f181440
Step 5/5 : CMD echo "Hello CMD"
 ---> Running in 9c2cf0f9af57
Removing intermediate container 9c2cf0f9af57
 ---> 0b7ac9fa6d88
Successfully built 0b7ac9fa6d88
Successfully tagged dockerfile:latest
root@UbuntuL:/home/user/Desktop/Docker# docker run -ti --rm dockerfile
Hello CMD
```

6. Cas Dockerfile pour programme Java

a) Alpine ne contient pas java et jdk

```
root@UbuntuL:/home/user# docker run -ti --rm alpine
/ # java
/bin/sh: java: not found
/ # java/jdk
/bin/sh: java/jdk: not found
/ # ls
bin    etc    lib    mnt    proc   run    srv    tmp    var
dev    home   media  opt    root   sbin   sys    usr
/ #
```

Je lance docker build -t javadocker et c'est le fichier dockerfile qui se lance par conséquent je n'arrive pas à exécuter le fichier Helloword.java

```
user@UbuntuL:~$ su root
Password:
root@UbuntuL:/home/user# cd Desktop
root@UbuntuL:/home/user/Desktop# cdDocker
cdDocker: command not found
root@UbuntuL:/home/user/Desktop# cd Docker
root@UbuntuL:/home/user/Desktop/Docker# ls
Dockerfile  HelloWord.java  javadocker
root@UbuntuL:/home/user/Desktop/Docker#
```

à l'intérieur de javadocker nous avons écrit :

```
RUN apt-get install -y openjdk-17-jdk && apt-get install -y wget
RUN apt-get clean

RUN java -version
RUN javac - version

RUN javac HelloWord.java
CMD ["java","HelloWord"]
```

b) pas fait

Yann Emmanuel Brou
Sir William Ngoma

7. Cas Dockerfile pour programme Java

a) Compte créé

On se connecte avec : docker login → On rentre ensuite le nom d'utilisateur et le mot de passe

Pour eviter les problème de push on va créer un tag avec la syntaxe suivante : docker tag dockername dockerhub_username/dockername

puis on fera docker push dockerhub_username/dockername
pour charger javadocker et dockerfile dans l'espace concerné !