Résumé du travail réaliser avec MR YAO

Utilisation de docker

Déployer l'application du prof grâce aux dockers

1/ récupérer les fichier du prof grâce a GIT

- copier le lien

- "git clone (coller le lien)" dans la racine

- Déplacer init-db.sql dans ~/mypostgres/

- Déplacer dbproject.war dans ~/mytomcatt/

2/ récupérer l'image de tomcat:8-jre8

- docker run -d --mytomcat tomcat:8-jre8

3/ récupérer l'image de postgres:9.5

- docker run -d --mypostgres postgres:9.5

4/ Créer le Dockerfile de tomcat dans ~/mytomcatt/

- Vi Dockerfile :

======================================

FROM tomcat:8-jre8

RUN apt-get update && apt-get install -y postgresql-9.5

COPY dbproject.war /usr/local/tomcat/webapps/dbproject.war

======================================

5/ Lancer le build du Dockerfile

- docker build -t mytomcatt .

6/ Créer le Dockerfilede postgres dans ~/mypostgress/

- vi Dockerfile

=======================================

FROM postgres:9.5

RUN apt-get update

COPY init-db.sql /docker-entrypoint-initdb.d/init-db.sql

========================================

7/ Lancer le build du Dockerfile:

- docker build -t mypostgress .

8/ Lancer les deux containers avec le mapping de ports:

- docker run -d --name db -p 8889:5432 -v dbdata:/var/lib/postgresql/data mypostgres

- docker run -d --name mytomcat --link db -p 9293:8080 mytomcat

9/ A partir d'un navigateur internet, taper : 192.168.99.100:9293/dbproject/accueil.jsp

-----------------------------------------

=======================================

-----------------------------------------

Mettre les images créées dans le regisetry hub

1/ Se créer un compte sur github.com

2/ Se connecter à partir du docker:

- docker login -u mamadt

3/ Créer un tag d'une image :

- docker tag postgres1 mamadt/mypostgres:9.5

- docker tag tomcat1 mamadt/mytomcat:8-jre8

4/ Mettre l'image sur le docker hub:

- docker push mamadt/mypostgres:9.5

- docker push mamadt/mytomcat:8-jre8

Mettre des fichiers sur git hub

1récupérer le fichier sur mon compte git à 'laide du lien

$ git clone https://github.com/MamadT/mahamadou.git

Entrer dans le repertoir mahamadou (Master)

$ cd mahamadou/

crée un fichier test

touch fichier.txt

Ajouter le fichier.txt et rajouter un commentaire

$ git add fichier.txt

$ git commit -m "1 fichier ra"

$ git config --global --edit

$ git commit -m "1 fichier ra"

Faire un git push afin que le fichier.txt soit publié dans le hub

$ git push

Pour automatiser les commandes de container à l’aide d’un script

Il faut :

1 Récupérer le docker-compose.yml sur le git hub du prof

2 Copier le docker-compose sur ma racine avec les autres dossiers avec lequel de je travail (mypostgress et mytomcatt)

3 Modifier le fichier docker-compose.yml

$ vi docker-compose.yml

version: '3'

services:

mydbapp:

build: ./mytomcatt/

ports:

- "9293:8080"

depends\_on:

- db

db:

build: ./mypostgress/

ports:

- "8889:5432"

volumes:

- dbdata:/var/lib/postgresql/data

container\_name: db

volumes:

dbdata:

4 les modifications ont étai effectué

5 Ensuite s'assurer que tous mes container sont suprimmer

6 et lancer la commande :

docker-compose --build -d

7 Dockerfile se lance automatiquement

Docker network

1 Pour afficher les réseaux : faire

$ docker network ls

(il Ya plusieurs réseau dont bridge celui qui est le standard ou le réseau autorise la connexion

2 Cree son réseau

$ docker network create monreseau

$docker inspect monreseau (pour vérifier)

3 Cree un container et le rattacher a un réseau

$ docker run -d --net monreseau1 --name container1 busybox:1.27 sleep 400

$ docker run -d --net monreseau2 --name container2 busybox:1.27 sleep 400

4 Entrer dans le container

$ docker exec -it container1 bin/ash

ifconfig pour afficher les infos réseau du container puis ping 8.8.8.8

5 pour connecter un container a un autre reseaux

$ docker network connect monreseau container2

Cette commande permet de faire en sorte que 2 container dans de sous réseau différent peuvent communiquer

6 Pour que le container a les mêmes infos que la carte réseau de la machine :

Choisir host