

SAE23 Etape 3 Compte Rendu Eren Turkoglu

Les commandes :

Actions	Commandes
Créer une image d'après un conteneur	<code>docker commit NOMCONTENEUR NOMIMAGE</code>
Créer un dépôt pour stocker vos images	
Stocker une image sur un dépôt privé	
Lister les images disponibles sur un dépôt	
Construire une image d'après un Dockerfile	<code>docker build -t NOMIMAGE NOMREPDOCKERFILE</code>
Dockerfile : bâtir d'après image xxxx	<code>FROM xxxx</code>
Dockerfile : copier un fichier	<code>COPY NOMFICHIER REPCONTENEUR/NOMFICHIER</code>
Dockerfile : Exécuter une commande	<code>RUN COMMANDE</code>
Dockerfile : Exécuter une commande après avoir crée le conteneur	<code>CMD COMMANDE</code>
Dockerfile : Expose un port	<code>EXPOSE NOPORTCONTENEUR</code>
Créer les conteneurs d'après un fichier docker-compose	
Dockercompose : même chose et revenir à la ligne de commande	
Stopper les conteneurs créé par dockercompose	

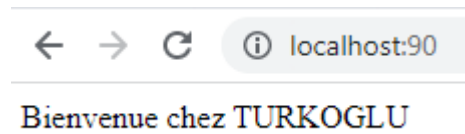
Création d'un image à partir d'un conteneur :

J'ai créé un conteneur webdevops/php-nginx en utilisant la commande `docker run -name mon_web -p 90:80 webdevops/php-nginx`.

```
C:\Users\ismai>docker run --name mon_web -p 90:80 webdevops/php-nginx
-> Executing /opt/docker/provision/entrypoint.d/05-permissions.sh
-> Executing /opt/docker/provision/entrypoint.d/20-nginx.sh
-> Executing /opt/docker/provision/entrypoint.d/20-php-fpm.sh
-> Executing /opt/docker/provision/entrypoint.d/20-php.sh
```

Puis, j'ai utilisé cette commande pour changer la page index. `docker exec -ti mon_web sh -c "echo \"Bienvenue chez TURKOGLU\" > /app/index.php"`.

```
C:\Users\ismai>docker exec -ti mon_web sh -c "echo \"Bienvenue chez TURKOGLU\" > /app/index.php"
```



Et pour finir on a transformé le conteneur en image en utilisant la commande `docker commit mon_web im_web`.

```
C:\Users\ismai>docker commit mon_web im_web
sha256:eec360a8c51c44dc10f86ef72d9ba385d3a74f3280da679b2c8e99d0cec2f27e
```

Les images :

```
C:\Users\ismai>docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
im_web              latest     eec360a8c51c  2 minutes ago  539MB
phpmyadmin          latest     b71bc1562440  9 days ago    510MB
wordpress           latest     764973ecc5df  9 days ago    609MB
mariadb             latest     ea81af801379  12 days ago   383MB
debian              latest     4eacea30377a  3 weeks ago   124MB
webdevops/php-nginx latest     4edee5cecbe7  2 years ago   538MB
```

Pour tester l'image qu'on a créé on supprime le conteneur et on le recrée en utilisant l'image `im_web`.

```
C:\Users\ismai>docker stop mon_web
mon_web

C:\Users\ismai>docker rm mon_web
mon_web

C:\Users\ismai>docker create --name mon_web2 -p 90:80 im_web
418c76765649b2e4022f6b019d6e3822c88520b5306d78d3bda445b3bc22d8d7

C:\Users\ismai>docker start mon_web2
mon_web2
```

Exemple création de Dockerfile :

Maintenant, on va faire un Dockerfile qui va créer un conteneur et changer quelque configurations.

```

C:\Users\ismai\Desktop>docker build -timg_web_turkoglu rep_web_turkoglu
[+] Building 11.9s (9/9) FINISHED
=> [internal] load build definition from Dockerfile                                0.8s
=> => transferring dockerfile: 221B                                              0.1s
=> [internal] load .dockerignore                                                  0.6s
=> => transferring context: 2B                                                    0.0s
=> [internal] load metadata for docker.io/webdevops/php-nginx:latest             0.0s
=> [internal] load build context                                                  1.4s
=> => transferring context: 17.65kB                                              0.0s
=> [1/4] FROM docker.io/webdevops/php-nginx                                     4.7s
=> [2/4] COPY photo.jpg /app/photo.jpg                                           0.2s
=> [3/4] RUN echo "<p>Coucou c'est TURKOGLU</p>" > /app/index.php               3.2s
=> [4/4] RUN echo "<img src='photo.jpg'>" >> /app/index.php                     1.6s
=> exporting to image                                                            0.8s
=> => exporting layers                                                            0.5s
=> => writing image sha256:2c688c907f5f58d08edcc1b5da9fc4c15fcb6f880d1fca7ba61e66390b36ab84 0.0s
=> => naming to docker.io/library/img_web_turkoglu                             0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

```

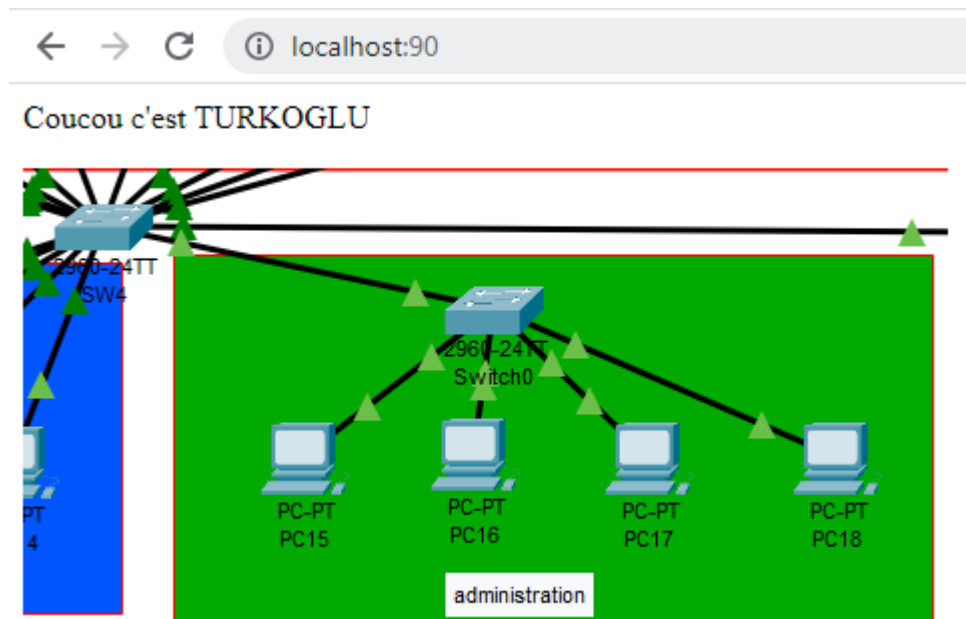
Maintenant, on va créer un conteneur a partir de cette image.

```

C:\Users\ismai\Desktop>docker run --name mon_web -p 90:80 img_web_turkoglu

```

La page :



Publication d'une image docker sur un dépôt privé :

Premièrement, il faut installer l'image registry pour créer un dépôt privé.
Après l'installation, il faut créer le conteneur.

```
C:\Users\ismai>docker run -d -p 5000:5000 --restart always --name registry registry:2
Unable to find image 'registry:2' locally
2: Pulling from library/registry
Digest: sha256:bedef0f1d248508fe0a16d2cacea1d2e68e899b2220e2258f1b604e1f327d475
Status: Downloaded newer image for registry:2
ec22f653f20ad78a22e28170a1dd25cf63606c8d882436977347f4557640feb5
```

Puis pour mettre une image dans ce dépôt il faut premièrement faire un tag. Comme ceci :

```
C:\Users\ismai>docker tag img_web_turkoglu localhost:5000/img_web_turkoglu
```

Après qu'on a fait le tag on peut le mettre sur le dépôt.

```
C:\Users\ismai>docker push localhost:5000/img_web_turkoglu
Using default tag: latest
The push refers to repository [localhost:5000/img_web_turkoglu]
b2357f5d2f71: Pushed
fd02a3826b0f: Pushed
327c33919514: Pushed
c9021a0df109: Pushed
b8167a4f532b: Pushed
780f768e11e9: Pushed
06eda94af7f8: Pushed
5a8729e90d94: Pushed
3daf0cc5c641: Pushed
b32f54152b11: Pushed
6be7fa578c62: Pushed
9bab65932166: Pushed
015de4fe565b: Pushed
16542a8fc3be: Pushed
6597da2e2e52: Pushed
977183d4e999: Pushed
c8be1b8f4d60: Pushed
latest: digest: sha256:b072126aec66161cbe91ccb6ee22d7d7a201b51d1e8869a55f0ac0c38f6b2e41 size: 3876
```

Après on peut aussi récupérer l'image de notre voisin en changeant l'adresse dans les paramètres.

```
{
  "insecure-registries" : ["192.168.103.130:5000"]
  "builder": {
    "gc": {
      "defaultKeepStorage": "20GB",
      "enabled": true
    }
  },
  "experimental": false,
  "features": {
    "buildkit": true
  }
}
```

Et puis on peut prendre les images comme on fait pour docker hub

Création du site portfolio :

Pour créer le site du portfolio j'ai fait un script. Comme ceci :

```
docker pull debian
docker network create --subnet=192.168.23.0/24 sae23-turkoglu
docker run --name eren --network sae23-turkoglu -p 89:80 debian
docker start eren
docker exec eren apt update -y
docker exec eren apt install apache2 -y
docker exec eren apt install php -y
docker exec eren apt install php-xml php-mbstring php-intl php-mysql
-y
docker exec eren apt install zip -y
docker exec eren php -r "copy('https://getcomposer.org/installer',
'/root/composer-setup.php');"
docker exec eren php /root/composer-setup.php
--install-dir="/usr/bin"
docker exec eren apt install git -y
docker exec eren git clone
https://github.com/egenroanne/sae14-erenn.git /var/www/html/site
docker exec eren apt update && apt install composer -y
docker exec -w /var/www/html/site eren composer.phar -n update &&
composer.phar -n install
docker exec eren apt install lynx -y
docker exec eren apt install nano -y
docker cp C:/Users/Administrateur/Desktop/dockerfile-eren/site.conf
eren:/etc/apache2/sites-available/
docker exec -w /etc/apache2/sites-available/ eren a2ensite site.conf
docker exec -w /etc/apache2/sites-available/ eren a2dissite
000-default.conf
docker exec -w /etc/apache2/sites-available/ eren a2enmod rewrite
docker exec eren /usr/sbin/service apache2 start
```

dfgdfgfdg