

Projet Docker

AIRBUS

Etape : 3

Liste des commandes :

Partie 1 :

téléchargement de apache alpine

The screenshot shows the Docker Desktop configuration page for a container named 'web1'. Under the 'Image configuration' section, the 'Registry' is set to 'DockerHub' and the 'Image' is 'docker.io/httpd:alpine'. There is a link for 'Advanced mode' and a toggle switch for 'Always pull the image' which is currently turned on.

redirection des ports 80 : 80

The screenshot shows the 'Network ports configuration' section for the 'web1' container. The toggle for 'Publish all exposed network ports to random host ports' is turned on. Below, the 'Manual network port publishing' section shows a mapping from host port 80 to container port 80. The protocol is set to 'TCP'.

en commandes il faut ajouter -p pour publier le conteneur notamment avec une translation de ports depuis le serveur

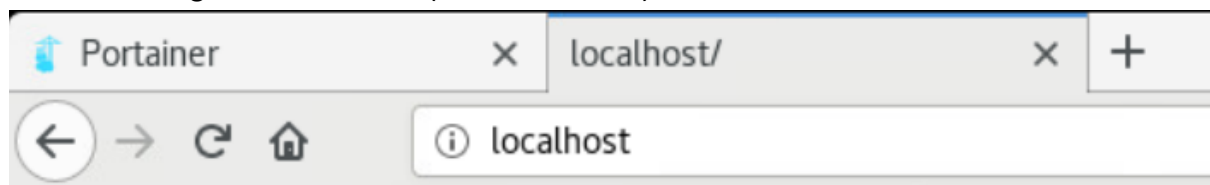
Deploy the container

pour la mise en tache de fond portainer le fais d'emblée mais sinon nous aurions dû ajouter -d pour qu'elle le soit
exemple "docker run -d

voici la machine créée

web1	running	80:80	httpd:alpine
2021-04-26 05:55:01	80:80	administrators	

test via navigateur en local (localhost:80)

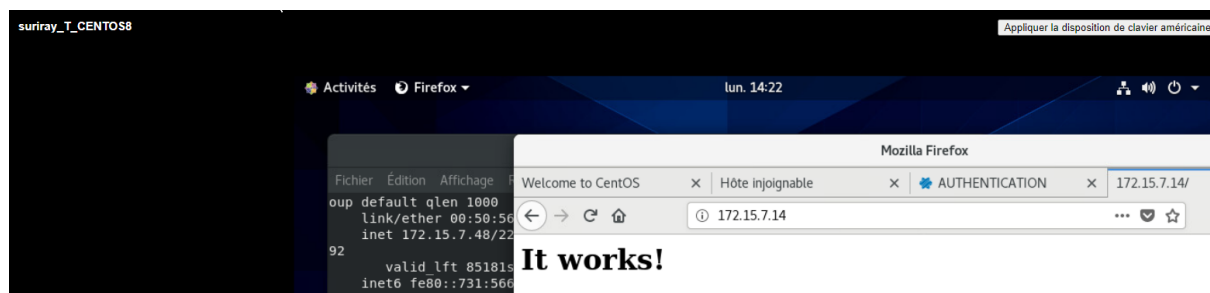


It works!

voici un test réalisé depuis une machine CentOS 8 notamment celle de Donovan car les machines windows sont lentes

ip de donovan :

172.15.7.48 /22 BTS PEDAGO



Partie 2 :

<input type="checkbox"/> web3	running	-	httpd:alpine	2021-04-26 08:25:26	80:80	ad
<input type="checkbox"/> web2	running	-	httpd:alpine	2021-04-26 08:24:30	8080:80	ad

voici les deux machines avec les mêmes images

```
/usr/local/apache2/htdocs # ls
index.html
```

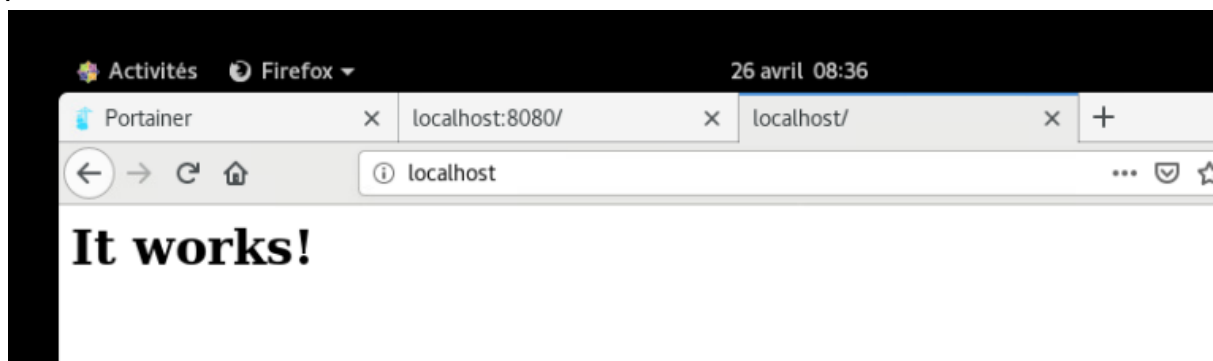
modification du fichier

```
<html><body><h1>Welcome to web 2!!!</h1></body></html>
```

puis test de web 2



puis de web 3



ils sont biens différenciés mais le web 3 était modifié mais pas pris en compte

faire un commit du web 2 ce qui revient à faire une création d'image via portainer

Create image

You can create an image from this container, this allows you to backup important data or save helpful configurations. You'll be able to spin up another container based on this image afterward.

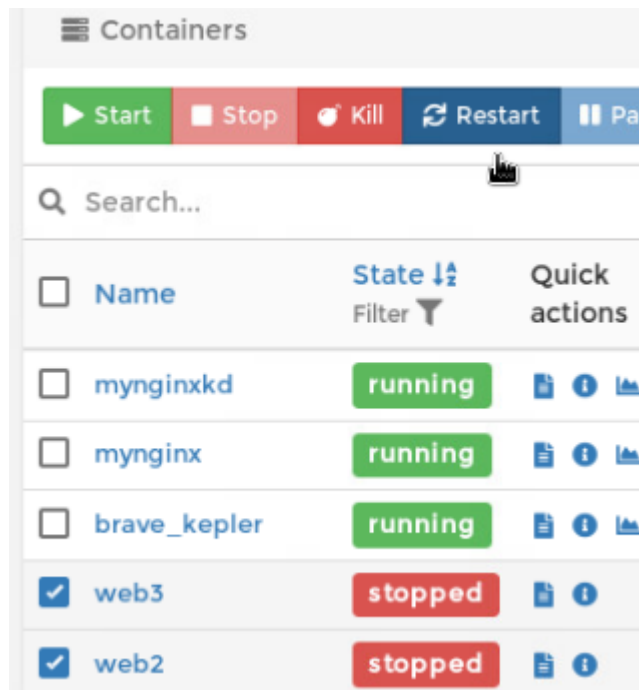
Registry

DockerHub

Image

docker.io

httpd:web3



changements avec les tests:

nous avons eu les pages web afficher “**Welcome to Web 2 et 3**”

conclusion:

après redémarrages des conteneurs, ceci permet une actualisation de toutes les modifications

Partie 3 (avec commandes) :

Récapitulatif des commandes :

docker run itd 80:80 httpd:web3 sh

docker rename (id) web4

#docker volume create mesdata

#docker volume inspect mesdata

nous avons fait le lien avec web4 via portainer

#docker restart web4

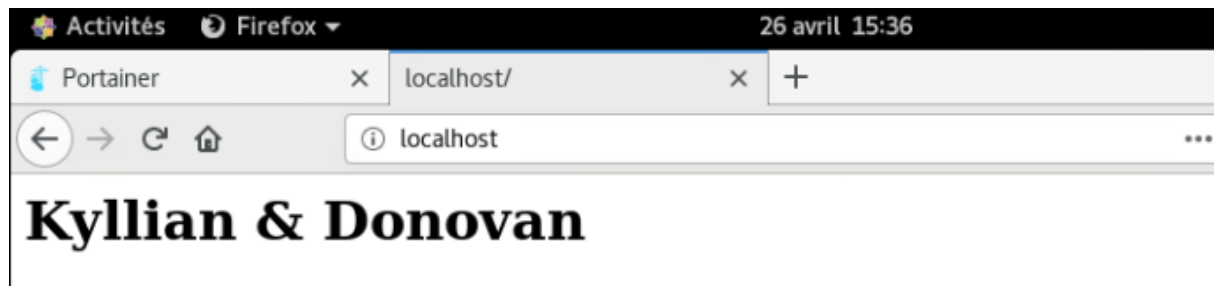
#cd /volume1

#vi index.html

```
<html><head></head><body><h1>Kyllian &
Donovan</h1></body></html>
```

Voici le résultat:

localhost:80 ou 127.0.0.1:80

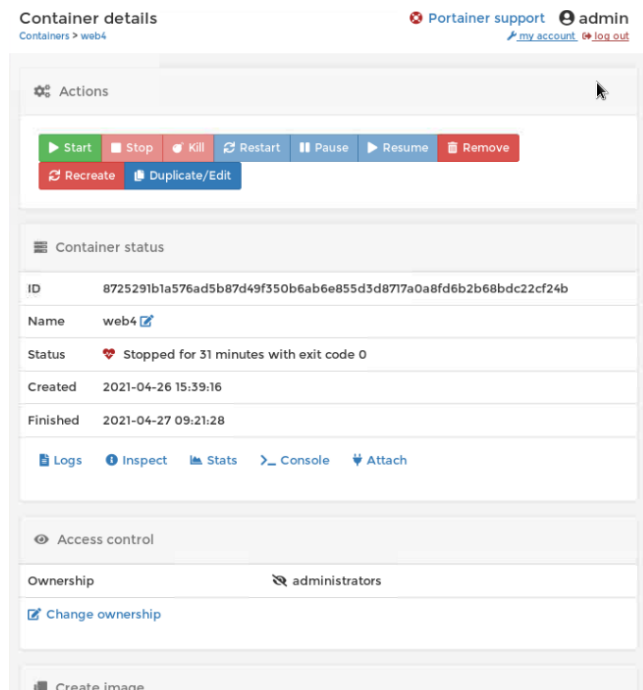


#docker stop web4 ou (id)

Partie 3 avec (portainer) :



création du conteneur web4



voici le volume associé au conteneur, équivalent en commande est -v (options)

Volumes	
Host/volume	Path in container
mesdata	/usr/local/apache2/htdocs

utilisation de la console sur portainer pour la gestion et modification du fichier index.html

>_ Execute

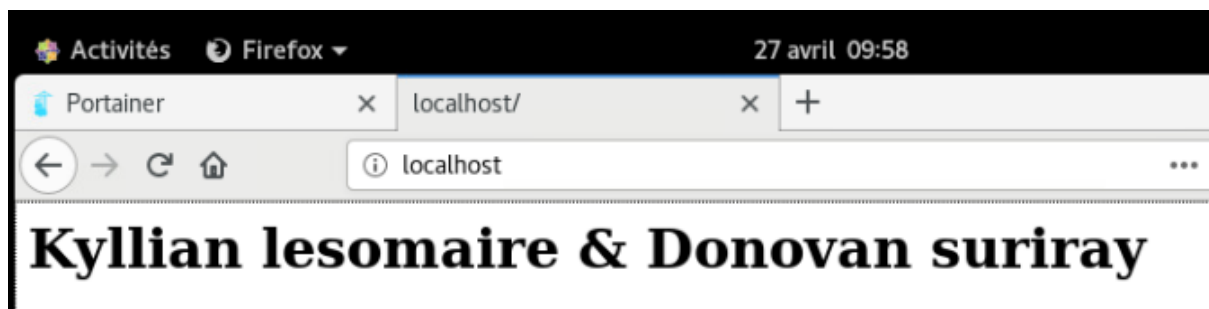
Exec into container as **default user** using command **sh**

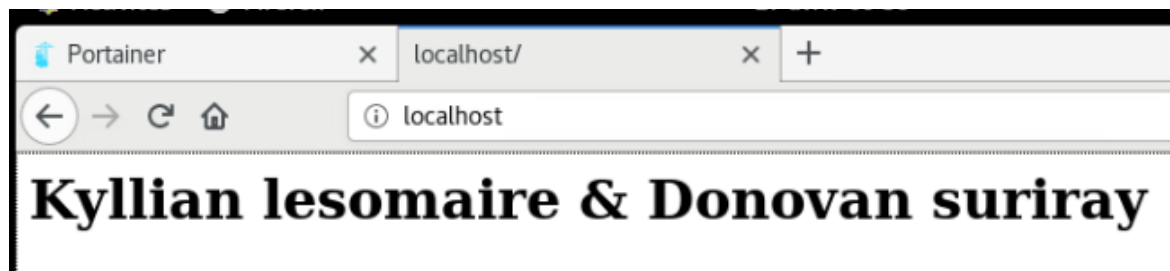
Disconnect

```
<html><body><h1>Kyllian lesomaire & Donovan suriray</h1></body></html>
```

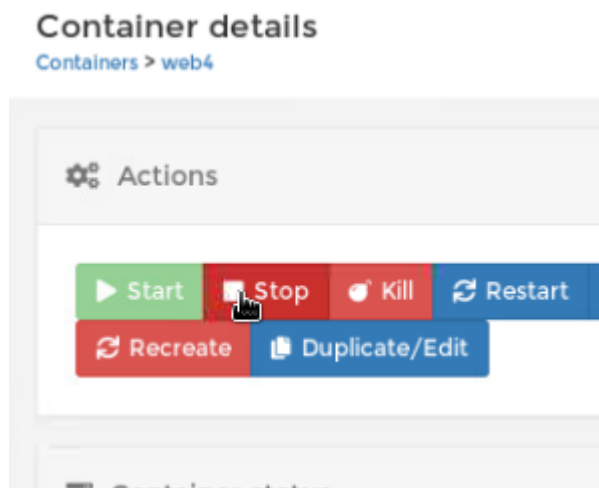
```
- index.html 1/1 100%
```

voici le résultat





tests ok après redémarrage, suppression du processus



c'est bon tout est ok