

LAB 5 - Registry Open Source

Dans ce lab, nous allons utiliser le registry open source de Docker.

Lancement

Nous allons utiliser l'image officielle du registry disponible dans le Docker Hub.

A partir de l'image registry:2.7.1, lancez un container en précisant qu'il doit tourner en background et exposer son port 5000 sur le port 5000 de la machine hôte.

```
[root@docker ~]# docker container run -d -p 5000:5000 registry:2.7.1
```

Liste des images du registry

Le registry expose un API HTTP Rest sur le port 5000.

En utilisant une requête GET sur le endpoint /v2/_catalog nous obtenons la liste des images présentes.

Cette liste est vide car nous n'avons pas encore envoyé d'image dans le registry.

```
[root@docker ~]# curl localhost:5000/v2/_catalog  
{"repositories": []}
```

Récupération d'une image depuis le Docker Hub

Téléchargez l'image nginx depuis le Docker Hub.

Nous utiliserons cette image par la suite pour l'envoyer dans le registry lancé précédemment.

```
[root@docker ~]# docker image pull nginx  
Using default tag: latest  
latest: Pulling from library/nginx  
69692152171a: Already exists  
49f7d34d62c1: Pull complete  
5f97dc5d71ab: Pull complete  
cfcd0711b93a: Pull complete  
be6172d7651b: Pull complete  
de9813870342: Pull complete  
Digest:  
sha256:df13abe416e37eb3db4722840dd479b00ba193ac6606e7902331dcea50f4f1f2  
Status: Downloaded newer image for nginx:latest  
docker.io/library/nginx:latest
```

Tag de l'image au format du registry

Afin que l'image puisse être uploadée dans le registry, il faut qu'elle suive un format spécifique.

Le nom de l'image doit être préfixée par le FQDN et le port du registry.

```
localhost:5000/NAME:VERSION
```

Utilisez la commande tag afin d'ajouter un tag à l'image nginx que nous avons téléchargée ci-dessus. Le tag commencera donc par **localhost:5000**, nous définissons également le nom de l'image, w3 et la version, 1.0.

```
[root@docker ~]# docker image tag nginx localhost:5000/w3:1.0
```

Listez les images disponibles localement. Vous pourrez voir l'image **registry:2.7.1**, l'image **nginx:latest** téléchargée manuellement, ainsi que l'image **localhost:5000/w3:1.0** que vous venez de créer.

```
[root@docker ~]# docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
nginx	latest	f0b8a9a54136	9 days ago	133MB
localhost:5000/w3	1.0	f0b8a9a54136	9 days ago	133MB
registry	2.7.1	1fd8e1b0bb7e	5 weeks ago	26.2MB

Upload de l'image dans le registry

Maintenant que le tag de l'image est bien formaté, uploadez celle-ci dans votre registry.

```
[root@docker ~]# docker image push localhost:5000/w3:1.0
```

```
The push refers to repository [localhost:5000/w3]
```

```
f0f30197ccf9: Pushed
```

```
eeb14ff930d4: Pushed
```

```
c9732df61184: Pushed
```

```
4b8db2d7f35a: Pushed
```

```
431f409d4c5a: Pushed
```

```
1.0: digest: sha256:eba373a0620f68ffdc3f217041ad25ef084475... size: 1570
```

Si vous listez une nouvelle fois les images présentes dans le registry, vous pourrez voir que l'image taggée localhost:5000/w3:1.0 est présente.

```
[root@docker ~]# curl localhost:5000/v2/_catalog
{"repositories":["w3"]}
```

En résumé

Cet exemple simple illustre comment le registry open source fonctionne. Il y a cependant plusieurs points à noter:

Par défaut, le Docker daemon ne peut pas communiquer avec un registry qui n'est pas sécurisé (c'est à dire sans TLS), ce qui est le cas du registry open source. Il y a cependant 3 options possibles:

- changer la configuration du daemon et le redémarrer avec le flag `--insecure-registry`
- utiliser la localhost exception qui permet la communication via localhost même sans le flag `--insecure-registry`. C'est l'approche que l'on utilise dans cet exemple.
- créer un certificat et des couples de clés publique/privée, signées par ce certificat, pour le daemon et le registry.

Quand un registry autre que le Docker Hub est utilisé, les tags des images de ce registry doivent être préfixés par `REGISTRY_HOSTNAME:REGISTRY_PORT`

Exécutez le registre Docker local avec SSL

Personnaliser le nom de domaine de votre daemon docker

```
[root@docker ~]# vi /etc/hosts
```

Entrez l'ip et le nom de domaine de votre registre

```
192.168.208.169 worker-node2 myregistrydomain.com
```

Créez un répertoire et placez des certificats sur l'hôte:

```
[root@docker ~]# mkdir /certs
```

Générez votre propre certificat:

```
[root@docker ~]# openssl req -newkey rsa:4096 -nodes -sha256 -keyout /certs/domain.key -x509 -days 365 -out /certs/domain.crt
```

Démarrez un conteneur de registre en SSL

```
[root@docker ~]# docker run -d --name docker-registry --restart=always \
  -p 5000:5000 -e REGISTRY_HTTP_ADDR=0.0.0.0:5000 \
  -e REGISTRY_HTTP_TLS_CERTIFICATE=/certs/domain.crt \
  -e REGISTRY_HTTP_TLS_KEY=/certs/domain.key \
  -v /certs:/certs \
  -v /var/lib/docker/registry:/var/lib/registry \
  --name myregistre registry:2.7.1
```

Demandez à chaque démon Docker de faire confiance à ce certificat. Copiez le fichier domain.crt dans le dossier /etc/docker/certs.d/myregistrydomain.com:5000/

```
[root@docker ~]# cp certs/domain.crt /etc/docker/certs.d/myregistrydomain.com:5000/
```

Pour tester depuis un autre serveur docker (ici sur l'IP 192.168.208.168)

```
[root@docker ~]# scp domain.crt
root@192.168.208.168:/etc/docker/certs.d/myregistrydomain.com:5000/
```

Ajoutez le nom domaine de serveur contenant le registre dans toutes les machines clientes

```
vi /etc/hosts
```

Entrez l'ip et le nom de domaine de votre registre

```
192.168.208.169 worker-node2 myregistrydomain.com
```

Test :

Tagger une image

```
docker image tag nginx myregistrydomain.com:5000/w3:2.0
```

Uploader l'image dans le registre myregistrydomain.com

```
docker image push myregistrydomain.com:5000/w3:2.0
The push refers to repository [myregistrydomain.com:5000/w3]
f0f30197ccf9: Layer already exists
eeb14ff930d4: Layer already exists
c9732df61184: Layer already exists
4b8db2d7f35a: Layer already exists
431f409d4c5a: Layer already exists
02c055ef67f5: Layer already exists
2.0: digest: sha256:eba373a0620f68ffdc3f217041ad25ef084475b8feb35b992574cd83698e9e3c size:
1570
```

Supprimez l'image depuis la machine locale

```
docker rmi myregistrydomain.com:5000/w3:2.0
```

Télécharger l'image depuis le registre myregistrydomain.com

```
docker image pull myregistrydomain.com:5000/w3:2.0
```

Vérifiez la disponibilité de l'image localement

```
docker images
```