

# 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:2.6`, lancez un container en précisant qu'il doit tourner en background et exposer son port 5000 sur le port 5000 de la machine hôte.

```
$ docker container run -d -p 5000:5000 registry:2.6
```

## 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.

```
$ 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.

```
$ docker image pull nginx
Using default tag: latest
latest: Pulling from library/nginx
ff3d52d8f55f: Pull complete
226f4ec56ba3: Pull complete
53d7dd52b97d: Pull complete
Digest: sha256:41ad9967ea448d7c2b203c699b429abe1ed5af331cd92533900c6d77490e0268
Status: Downloaded newer image for 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`.

```
$ docker image tag nginx localhost:5000/w3:1.0
```

Listez les images disponibles localement. Vous pourrez voir l'image `registry:2.6`, l'image `nginx:latest` téléchargée manuellement, ainsi que l'image `localhost:5000/w3:1.0` que vous venez de créer.

```
$ docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
localhost:5000/w3	1.0	958a7ae9e569	2 weeks ago	109MB
nginx	latest	958a7ae9e569	2 weeks ago	109MB
registry	2.6	9d0c4eabab4d	5 weeks ago	33.2MB

## Upload de l'image dans le registry

---

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

```
$ docker image push localhost:5000/w3:1.0
The push refers to a repository [localhost:5000/w3]
a552ca691e49: Pushed
7487bf0353a7: Pushed
8781ec54ba04: Pushed
1.0: digest:
sha256:41ad9967ea448d7c2b203c699b429abe1ed5af331cd92533900c6d77490e0268 size:
948
```

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.

---

```
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`