

Université de Picardie Jules Verne

Informatique – Master CCM

INSSET – Saint-Quentin

Conteneurs Applicatifs et Micro-Services

M2

C. Drocourt

cyril.drocourt@u-picardie.fr

Cours 4 : Registry

V2023.01

Table des matières

Cours 4 : Registry	2
1 - Registry Docker.....	4
2 - Registry personnel.....	6
3 - Registry Sécurisé.....	12

1 - Registry Docker

Il est possible de sauvegarder une image sur un serveur distant offrant un service de « registry ». Si aucun serveur n'est précisé celui de « docker hub » est utilisé, à condition d'avoir créé un compte auparavant, et de s'être connecté :

```
[root@drocourt ~]# docker login -u rahan  
Password:  
Email: rahan@u13.org
```

On consulte les images possibles :

```
[root@drocourt ~]# docker images  
REPOSITORY          TAG          IMAGE ID          CREATED          VIRTUAL SIZE  
debian              8.1         91f98c48edd6    5 days ago     123 MB
```

Il est nécessaire de réaliser un tag en fonction de son nom de login, par exemple :

```
[root@drocourt ~]# docker tag debian:8.1 rahan/debian:8.1
```

On consulte les images possibles :

```
[root@drocourt ~]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE
debian	8.1	91f98c48edd6	5 days ago	123 MB
rahan/debian	8.1	91f98c48edd6	5 days ago	123 MB

On utilise la commande « push » pour sauvegarder :

```
[root@drocourt ~]# docker push rahan/debian:8.1
```

The push refers to a repository [rahan/debian] (len: 1)
91f98c48edd6: Image already exists
432ac41e2bbf: Image successfully pushed
a29f2b1e7978: Pushing [=====>] 23.59 MB/51.36 MB

On peu retrouver facilement ses images avec :

```
[root@drocourt ~]# docker search rahan/
```

2 - Registry personnel

2.1 - Pré-requis

On change le nom de la machine, ce qui nous servira dans la suite.

```
root@skel:~# hostname master
root@skel:~# hostname
master
```

Egalement de manière définitive :

```
root@skel:~# hostname > /etc/hostname
root@skel:~# hostname -I
10.3.134.190
```

On l'ajoute également dans le système :

```
root@skel:~# echo "10.3.134.190 master" >> /etc/hosts
```

On se déconnecte et on se reconnecte ...

ATTENTION : L'adresse IP est à adapter à votre cas !!!

2.2 - Installation

Il est possible de mettre en place son propre « registry », soit en installant l'ensemble des services associés, soit en utilisant un conteneur de registry pré-existant.

On récupère le conteneur :

```
[root@drocourt ~]# docker pull registry:2
```

On l'exécute :

```
[root@drocourt ~]# docker run -d -p 5000:5000 --restart=always --name registry registry:2
```

2.3 - Utilisation

Il est nécessaire ensuite d'effectuer un « tag » spécifique pour notre serveur sur l'image :

```
[root@drocourt ~]# docker tag cos7 master:5000/mycos7
```

Nous pouvons maintenant faire un « push » :

```
[root@drocourt ~]# docker push master:5000/mycos7
```

Et enfin supprimer toute trace :

```
[root@drocourt ~]# docker image rm cos7  
[root@drocourt ~]# docker image rm master:5000/mycos7
```

Maintenant, il est possible de l'installer à partir de notre « registry » :

```
[root@drocourt ~]# docker pull master:5000/mycos7
[root@drocourt ~]# docker images
```

REPOSITORY	TAG	IMAGE ID
master:5000/mycos7	latest	4220126e97b3
46 minutes ago	300 MB	

Il n'est pas possible a priori d'effectuer une recherche classique sur le « registry » local, mais on peut récupérer la liste par une simple commande « curl » :

```
[root@master ~]# curl master:5000/v2/_catalog
```

2.4 - Intégration

Pour intégrer ce « Registry » à l'ensemble des service de Docker, il faut modifier la configuration pour forcer le mode non sécurisé (c'est à dire non TLS) sur ce dernier. Il faut créer/modifier le fichier « /etc/docker/daemon.json » pour qu'il contienne les lignes suivantes (10.3.134.130 étant l'adresse IP du nœud maitre) :

```
{ "insecure-registries": ["10.3.134.190:5000"] }
```

On redémarre

```
root@master:~# systemctl restart docker
```

2.5 - Changement de port

Il est possible de demander un autre port au service :

```
docker run -d -e REGISTRY_HTTP_ADDR=0.0.0.0:5001 -p 5001:5001 --name registry-test registry:2
```

2.6 - Stockage et Volume

Il est préférable d'utiliser un volume externe pour stocker les différents dépôts :

```
root@master:~# docker volume create registry
root@master:~# docker run -d -p 5000:5000 --name registry --restart=always -v registry:/var/lib/registry registry:2
```

3 - Registry Sécurisé

3.1 - Configuration

Il est préférable d'utiliser un Registry sécurisé en TLS, pour cela nous allons passer par un autre volume dans lequel seront stocker les certificats.

```
root@master:~# docker volume create registry_certs
root@master:~# cd /var/lib/docker/volumes/registry_certs/
_data/
```

Dans ce répertoire, il suffit de recopier la clé privée et le certificat, dans notre cas nous allons utiliser un certificat auto-signé :

```
root@master:/var/lib/docker/volumes/registry_certs/_data#
openssl req \
  -newkey rsa:4096 -nodes -sha256 -keyout registry.pem \
  -addext "subjectAltName = DNS:master" \
  -x509 -days 365 -out registry.crt
```

Cette commande va poser un certain nombre de questions, la plus important est le nom de la machine qui héberge le Registry :

```
Country Name (2 letter code) [AU]:FR  
State or Province Name (full name) [Some-State]:Hauts de  
France  
Locality Name (eg, city) []:Saint-Quentin  
Organization Name (eg, company) [Internet Widgits Pty  
Ltd]:UPJV  
Organizational Unit Name (eg, section) []:INSSET  
Common Name (e.g. server FQDN or YOUR name) []:master  
Email Address []:cyril@drocourt.fr
```

On vérifie :

```
root@master:/var/lib/docker/volumes/registry_cert/_data# ls  
-l  
total 8  
-rw-r--r-- 1 root root 1509 sept. 13 13:23 registry.crt  
-rw----- 1 root root 1704 sept. 13 13:22 registry.pem
```

Comme le certificat est auto-signé, le service Docker va émettre des erreurs, il faut donc ajouter une configuration supplémentaire :

```
mkdir -p /etc/docker/certs.d/10.3.134.130:5443/  
mkdir -p /etc/docker/certs.d/master:5443/  
cp registry.crt /etc/docker/certs.d/10.3.134.130:5443/ca.crt  
cp registry.crt /etc/docker/certs.d/master:5443/ca.crt  
cp -a registry.crt /usr/local/share/ca-certificates/ca.crt  
update-ca-certificates
```

Il faut maintenant redémarrer le service Docker :

```
root@master:~# systemctl restart docker
```

Pour créer maintenant notre Registry :

```
docker run -d \  
  --restart=always \  
  --name registry \  
  -v registry:/var/lib/registry \  
  -v registry_certs:/certs \  
  -e REGISTRY_HTTP_ADDR=0.0.0.0:443 \  
  -e REGISTRY_HTTP_TLS_CERTIFICATE=/certs/registry.crt \  
  -e REGISTRY_HTTP_TLS_KEY=/certs/registry.pem \  
  -p 5443:443 \  
  registry:2
```

3.2 - Utilisation

De la même manière que pour un conteneur personnel il est nécessaire d'abord de réaliser un tag sur l'image souhaitée :

```
root@master:~# docker tag nginx master:5443/nginx
```

On vérifie :

```
root@master:~# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
master:5443/nginx	latest	2b7d6430f78d	3 weeks ago	142MB
nginx	latest	2b7d6430f78d	3 weeks ago	142MB
registry	2	3a0f7b0a13ef	4 weeks ago	24.1MB

Il suffit maintenant de réaliser un « push » :

```
root@master:~# docker push master:5443/nginx  
Using default tag: latest  
The push refers to repository [master:5443/nginx]
```