Création d'un Docker Registry Privé

Créer un Docker Registry privé permet d'héberger et de gérer des images Docker de manière sécurisée, offrant ainsi un contrôle total sur les images utilisées dans votre infrastructure. Vous pouvez utiliser **Docker Registry**, une solution open-source, pour créer un registry privé. Voici les étapes pour créer et configurer un Docker Registry privé.

1. Prérequis

Avant de commencer, assurez-vous que vous avez :

- Docker installé sur la machine qui hébergera le registry privé.
- Des permissions d'administrateur sur cette machine.

2. Lancer un Docker Registry

La manière la plus simple de créer un registry Docker privé est d'utiliser l'image officielle `registry` disponible sur Docker Hub. Cette image permet de configurer un registry privé de manière rapide et efficace.

Étape 1 : Lancer le registry avec Docker

Exécutez la commande suivante pour démarrer un container Docker avec le registry privé sur votre machine locale (par défaut, il écoutera sur le port 5000) :

```
docker run -d -p 5000:5000 --name registry registry:2
```

Explication des options :

- `-d` : Lance le conteneur en mode détaché (en arrière-plan).
- `-p 5000:5000` : Expose le port 5000 du conteneur vers le port 5000 de la machine hôte.
- `--name registry` : Nom du conteneur.
- `registry:2` : Utilisation de l'image officielle de Docker Registry (version 2).

Une fois lancé, le registry sera accessible via `http://localhost:5000` ou `http://<votre-ip>:5000` (si vous souhaitez y accéder à partir d'une autre machine).

Étape 2 : Vérifier que le registry fonctionne

Pour vérifier que votre registry fonctionne correctement, vous pouvez exécuter une commande `docker push` ou `docker pull`. Essayez de pousser une image test vers votre registry :

```
docker tag nginx:latest localhost:5000/my-nginx
docker push localhost:5000/my-nginx
```

Cela poussera l'image `nginx:latest` vers le registry privé. Vous pouvez ensuite vérifier que l'image est bien présente dans votre registry avec la commande `curl` :

```
curl http://localhost:5000/v2/_catalog
```

Cela affichera un JSON contenant la liste des repositories disponibles dans votre registry.

3. Sécuriser le Docker Registry avec HTTPS

L'accès au Docker Registry privé doit être sécurisé pour éviter les risques liés à la transmission de données en clair. Pour ce faire, vous devez configurer **HTTPS** avec un certificat SSL.

Étape 1 : Créer un certificat SSL

Si vous n'avez pas encore de certificat SSL, vous pouvez en créer un pour tester avec **OpenSSL** ou obtenir un certificat valide d'une autorité de certification (CA).

1. Créez un répertoire pour stocker les certificats :

   ```
   mkdir -p /etc/docker/certs
   ```

2. Générez un certificat auto-signé (pour usage local) :

   ```
   openssl req \
     -newkey rsa:4096 -nodes -keyout /etc/docker/certs/domain.key \
     -x509 -days 365 -out /etc/docker/certs/domain.crt
   ```

   Assurez-vous de remplir les informations demandées lors de la génération du certificat.

Étape 2 : Configurer le Registry pour utiliser HTTPS

Pour que votre registry privé utilise HTTPS, vous devez spécifier les chemins des fichiers de certificat et de clé lors du démarrage du conteneur Docker Registry.

1. Lancez le registry avec les options de certificat :

   ```
   docker run -d -p 443:5000 \
     --name registry \
     -v /etc/docker/certs:/certs \
     -e REGISTRY_HTTP_SECRET=mysecretkey \
     -e REGISTRY_HTTP_HEADERS=X-Content-Type-Options=no-sniff \
     --restart=always \
     registry:2
   ```

- `-v /etc/docker/certs:/certs` : Monte le répertoire contenant les certificats dans le conteneur.
- `REGISTRY_HTTP_SECRET` : Utilisé pour signer les cookies de session pour la sécurité.
- `REGISTRY_HTTP_HEADERS` : Définit des en-têtes HTTP de sécurité.

2. **Redémarrez le conteneur** pour appliquer les changements.

3. Vérifiez si votre registry fonctionne sur `https://localhost` en vous connectant via un navigateur ou en utilisant `curl`.

4. Authentification avec un Docker Registry privé

Si vous souhaitez ajouter une couche d'authentification à votre Docker Registry privé (par exemple, pour le rendre accessible uniquement aux utilisateurs authentifiés), vous devez configurer une authentification de type `Basic Auth`.

Étape 1 : Créer un fichier d'authentification

1. Installez le paquet `htpasswd` (pour gérer les utilisateurs) :

```
sudo apt-get install apache2-utils
```

2. Créez un fichier d'authentification et ajoutez un utilisateur :

```
htpasswd -c /etc/docker/registry/htpasswd myuser
```

Il vous sera demandé de définir un mot de passe pour l'utilisateur `myuser`.

Étape 2 : Configurer le registry avec l'authentification

1. Créez un fichier de configuration `config.yml` pour le registry, où vous définissez l'authentification.
Exemple de fichier `config.yml` :

```
http:
  headers:
    X-Content-Type-Options: nosniff
auth:
  htpasswd:
    realm: basic-realm
    path: /etc/docker/registry/htpasswd
```

2. Relancez le registry Docker avec la configuration de l'authentification :

```
docker run -d -p 443:5000 \
  --name registry \
  -v /etc/docker/certs:/certs \
  -v
/etc/docker/registry/config.yml:/etc/docker/registry/config.yml \
  -v /etc/docker/registry/htpasswd:/etc/docker/registry/htpasswd \
  registry:2
```

Cette configuration active une authentification **Basic Auth** pour votre registry.

3. Vous devrez maintenant entrer les identifiants lors du **push** ou **pull** d'images vers/depuis le registry :

```
docker login localhost:5000
```

5. Utilisation du Docker Registry Privé

Une fois votre Docker Registry privé configuré et sécurisé, vous pouvez commencer à l'utiliser pour stocker et gérer vos images Docker.

- **Push d'une image** :

```
docker tag my-image localhost:5000/my-image
docker push localhost:5000/my-image
```

- **Pull d'une image** :

```
docker pull localhost:5000/my-image
```

Résumé

- Vous avez maintenant créé un **Docker Registry privé** accessible localement ou à distance.
- Vous avez sécurisé l'accès avec **HTTPS** et configuré une **authentification** pour protéger l'accès à votre registry.
- Vous pouvez utiliser ce registry pour stocker des images Docker sensibles ou propriétaires.

Ce registry privé peut être utilisé dans vos environnements de développement et de production pour gérer de manière centralisée vos images Docker.