Création d'un Docker Registry Authentifié

Lorsqu'un Docker Registry privé est utilisé pour stocker des images sensibles ou d'entreprise, il est souvent nécessaire de mettre en place un système d'**authentification** pour restreindre l'accès aux images. Une des méthodes d'authentification courantes est l'**authentification basique** (Basic Authentication), qui utilise un nom d'utilisateur et un mot de passe pour sécuriser l'accès.

Voici comment configurer un Docker Registry privé avec une **authentification basique** en utilisant un fichier `htpasswd`.

---

1. Ajouter une Authentification Basique

Qu'est-ce que l'authentification basique ?

L'authentification basique repose sur un mécanisme simple où le client Docker doit fournir un nom d'utilisateur et un mot de passe lors de l'accès au Docker Registry. Ces informations sont envoyées au serveur sous forme de texte brut, mais dans une requête HTTP chiffrée (via HTTPS), elles sont sécurisées.

Dans cette étape, nous allons configurer un Docker Registry privé pour utiliser l'authentification basique en associant un fichier `htpasswd` pour gérer les utilisateurs et les mots de passe.

Étapes pour configurer l'authentification basique

1. **Créer un fichier `htpasswd` pour gérer les utilisateurs**

   Le fichier `htpasswd` contient les noms d'utilisateur et les mots de passe cryptés qui seront utilisés pour l'authentification. Vous pouvez créer ce fichier en utilisant l'outil `htpasswd`, qui est fourni par le paquet `apache2-utils`.

   ##### Étape 1 : Installer `htpasswd`

   Si vous n'avez pas encore installé `htpasswd`, installez-le avec la commande suivante (sur un système basé sur Debian/Ubuntu) :

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

   ##### Étape 2 : Créer un fichier `htpasswd`

   Créez un fichier `htpasswd` et ajoutez un utilisateur avec un mot de passe. L'option `-c` crée le fichier, et vous serez invité à entrer un mot de passe pour l'utilisateur.

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

L'option `-c` crée un nouveau fichier `htpasswd`. Si vous souhaitez ajouter d'autres utilisateurs, ne réutilisez pas l'option `-c` (cela écraserait le fichier existant) :

```
sudo htpasswd /etc/docker/registry/htpasswd anotheruser
```

Le mot de passe sera crypté et ajouté au fichier `htpasswd`.

Vérifiez que le fichier `htpasswd` contient les utilisateurs et leurs mots de passe cryptés :

```
cat /etc/docker/registry/htpasswd
```

2. **Configurer le Docker Registry pour utiliser l'authentification basique**

   Une fois que vous avez créé votre fichier `htpasswd`, vous devez configurer Docker Registry pour l'utiliser pour l'authentification des utilisateurs.

   ##### Étape 1 : Créer ou modifier le fichier de configuration `config.yml`

   Le fichier `config.yml` est utilisé pour configurer les paramètres du Docker Registry. Vous devez ajouter la configuration d'authentification basique dans ce fichier.

   Créez ou modifiez le fichier `config.yml` de votre registry Docker pour inclure la section `auth` avec le chemin vers votre fichier `htpasswd`.

   Exemple de `config.yml` :

   ```
   http:
     secret: a_random_secret_key
     addr: :5000
     headers:
       X-Content-Type-Options: nosniff
   auth:
     htpasswd:
       realm: basic-realm
       path: /etc/docker/registry/htpasswd
   ```

   - **`realm`** : Spécifie le nom du domaine pour l'authentification, utilisé dans les messages d'invite de connexion.
   - **`path`** : Chemin du fichier `htpasswd` contenant les utilisateurs et les mots de passe.

   ##### Étape 2 : Lancer le Docker Registry avec l'authentification basique

   Maintenant que vous avez configuré l'authentification, vous devez lancer ou redémarrer votre Docker Registry en utilisant le fichier de configuration `config.yml` et le fichier `htpasswd`.

```
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 \
  --restart=always \
  registry:2
```

Ce conteneur exécute le Docker Registry avec l'authentification basique activée.

3. **Vérification de l'authentification**

   Pour vérifier que l'authentification fonctionne correctement, vous pouvez tenter de pousser ou de tirer une image depuis votre registry privé. Vous devrez fournir un nom d'utilisateur et un mot de passe chaque fois que vous interagissez avec le registry.

   Exemple de commande `docker login` :

   Pour vous connecter au registry avec le nom d'utilisateur et le mot de passe définis dans le fichier `htpasswd`, utilisez la commande `docker login` :

   ```
   docker login localhost:5000
   ```

   Docker vous demandera un nom d'utilisateur et un mot de passe. Si les informations sont correctes, l'authentification sera réussie et vous pourrez pousser ou tirer des images.

   Pousser une image dans le registry sécurisé :

   Pour pousser une image vers votre registry, taguez l'image avec le nom du registry et poussez-la :

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

   Tirer une image depuis le registry sécurisé :

   Pour tirer une image depuis votre registry sécurisé, utilisez la commande `docker pull` :

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

Résumé

- **Ajout d'une authentification basique** : L'authentification basique permet de restreindre l'accès à un Docker Registry privé en exigeant un nom d'utilisateur et un mot de passe.
- **Fichier `htpasswd`** : Ce fichier contient les utilisateurs et leurs mots de passe cryptés. Il est utilisé par le Docker Registry pour valider les connexions des utilisateurs.
- **Lancement du Docker Registry sécurisé** : Une fois le fichier `htpasswd` créé et configuré dans le fichier `config.yml`, vous pouvez lancer le registry avec l'authentification basique active.

Cette configuration permet d'ajouter une couche de sécurité supplémentaire à votre Docker Registry privé et de protéger les images sensibles ou propriétaires.