

Démo 10 : Créer, Build & Pusher une Image Docker avec GitHub Actions

Objectif

- ☒ Créer une image Docker à partir de son propre projet.
 - ☒ Automatiser le **build et le push** de cette image vers Docker Hub.
 - ☒ Mettre en place un pipeline **CI/CD avec GitHub Actions**.
-

Ce que vous allez apprendre

- ✓ Écrire un **Dockerfile** pour créer une image personnalisée.
 - ✓ Lancer un build local pour tester l'image.
 - ✓ Configurer un **workflow GitHub Actions**.
 - ✓ Utiliser des **secrets GitHub** pour sécuriser l'accès à Docker Hub.
-

◇ **1** Créer un Dockerfile personnalisé

Créez un fichier appelé **Dockerfile** à la racine de votre projet.

Exemple : Application Web basique en HTML

```
# Utiliser une image de base légère
FROM nginx:alpine

# Copier le contenu du dossier ./site dans le dossier web de nginx
COPY ./site /usr/share/nginx/html

# Exposer le port 80
EXPOSE 80

# Lancer nginx en mode foreground
CMD ["nginx", "-g", "daemon off;"]
```

 Structure du projet :

```
.
├── Dockerfile
├── site
│   └── index.html
```

 Dans **index.html**, ajoutez par exemple :

```
<h1>Hello Docker depuis GitHub Actions ! 🚀</h1>
```

◇ **2 Build & Push local de l'image (optionnel mais conseillé)**

🔗 Étape 1 : Build local

```
docker build -t votrenomdockerhub/monimage:latest .
```

🔗 Étape 2 : Tester localement

```
docker run -d -p 8080:80 votrenomdockerhub/monimage
```

🔗 Accéder à : <http://localhost:8080>

🔗 Étape 3 : Push manuel (test)

```
docker login
docker push votrenomdockerhub/monimage:latest
```

◇ **3 Prérequis GitHub Actions**

Avant de passer à l'automatisation, ajoutez vos **identifiants Docker Hub** en tant que **secrets GitHub**.

🔗 **Ajoutez les secrets dans votre repo GitHub :**

1. Allez dans **Settings > Secrets and variables > Actions**
2. Cliquez sur **New repository secret** et ajoutez :
 - **DOCKERHUB_USERNAME** = votre nom d'utilisateur Docker Hub
 - **DOCKERHUB_TOKEN** = un [token d'accès personnel](#)

◇ **4 Création du Workflow GitHub Actions**

📁 Fichier à créer :

`.github/workflows/docker-build.yml`

📄 Contenu :

```
name: Build and Push Docker Image

on:
  push:
    branches:
      - main
    tags:
      - "v*"

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout Repository
        uses: actions/checkout@v3

      - name: Login to Docker Hub
        uses: docker/login-action@v2
        with:
          username: ${ secrets.DOCKERHUB_USERNAME }
          password: ${ secrets.DOCKERHUB_TOKEN }

      - name: Build Docker Image
        run: docker build -t ${ secrets.DOCKERHUB_USERNAME }/monimage:latest .

      - name: Push Docker Image to Docker Hub
        run: docker push ${ secrets.DOCKERHUB_USERNAME }/monimage:latest
```

◇ 5 Vérifications après exécution

☑ Sur GitHub :

- Menu **Actions** > **Build and Push Docker Image**
- Voir les étapes "green" et sans erreur

☑ Sur Docker Hub :

- L'image `monimage:latest` apparaît dans votre dépôt public ou privé

☑ Localement :

```
docker pull votrenomdockerhub/monimage:latest
docker run -d -p 8080:80 votrenomdockerhub/monimage
```