

Docker : Maîtrisez la conteneurisation applicative de A à Z

Table des matières

Docker pour une application Front End	2
Installation et utilisation d'un conteneur NGINX	2

Docker pour une application Front End

Pour une application "classique" Front End, contenant au minimum des fichiers HTML, CSS et JavaScript, nous allons créer un conteneur NGINX, via un Dockerfile.

Installation et utilisation d'un conteneur NGINX

Voici un exemple de Dockerfile pour une application front-end classique utilisant HTML, CSS et JavaScript.

Ce Dockerfile suppose que vous avez déjà vos fichiers HTML, CSS et JavaScript prêts dans un répertoire nommé **app** à côté du Dockerfile.

```
# Utilisez une image légère basée sur un serveur web pour servir
l'application front-end
FROM nginx:alpine

# Copiez les fichiers de l'application dans le répertoire de travail de
l'image nginx
COPY ./app /usr/share/nginx/html

# Le serveur web nginx écoute le port 80 par défaut, donc pas besoin de
spécifier EXPOSE

# CMD n'est généralement pas nécessaire pour une image web statique,
car le serveur nginx est déjà démarré par défaut
# Si vous souhaitez ajouter un CMD personnalisé, il démarrera
automatiquement après le démarrage du serveur nginx
# Par exemple : CMD ["nginx", "-g", "daemon off;"]
```

Assurez-vous de placer ce Dockerfile dans le même répertoire que le répertoire **app** contenant vos fichiers HTML, CSS et JavaScript.

Pour construire et exécuter l'image Docker, exécutez les commandes suivantes dans le terminal :

1 - Placez-vous dans le répertoire contenant le Dockerfile et le répertoire **app**.

2 - Construisez l'image :

```
docker build -t nom_de_votre_image .
```

Assurez-vous de remplacer **nom_de_votre_image** par le nom que vous souhaitez donner à votre image.

3 - Exécutez le conteneur à l'aide de l'image que vous avez créée :

```
docker run -d -p 8080:80 nom_de_votre_image
```

Cela va démarrer un conteneur Docker exécutant un serveur NGINX pour servir vos fichiers HTML, CSS et JavaScript. Vous pourrez accéder à votre application en ouvrant un navigateur et en accédant à <http://localhost:8080>.

Assurez-vous que les chemins et la structure des fichiers dans le répertoire **app** sont corrects pour que l'application puisse être servie correctement par le serveur web NGINX.