

Rapport de TP : Déploiement Web avec Docker et Nginx

Fanambimiadantsoa Khejia

2 décembre 2025

1 Contexte et Objectif

Dans le cadre de notre apprentissage sur la conteneurisation et le déploiement web, nous avons travaillé sur le serveur web **Nginx** via Docker.

L'objectif principal de cette manipulation était de résoudre une problématique courante : **comment modifier le contenu d'un site web hébergé dans un conteneur sans avoir à reconstruire l'image à chaque changement ?**

2 La Solution Technique : Les Volumes

Par défaut, un conteneur Docker est un environnement isolé et éphémère. Si l'on modifie un fichier à l'intérieur, ces modifications sont perdues dès que le conteneur est supprimé.

Pour pallier cela, nous avons utilisé un mécanisme appelé **Volume** (ou *bind mount*). Cela permet de créer un pont direct entre un dossier de notre ordinateur physique (l'hôte) et un dossier à l'intérieur du conteneur.

3 Mise en œuvre

3.1 1. Préparation

J'ai créé localement un fichier `index.html` contenant le code source de ma page d'accueil personnalisée.

3.2 2. Exécution du conteneur

Nous avons lancé le conteneur Nginx avec la commande suivante :

```
docker run -d -p 8080:80 -v $(pwd)/index.html:/usr/share/nginx/html/index.html nginx
```

3.3 3. Analyse détaillée de la commande

Chaque paramètre a un rôle précis dans l'architecture mise en place :

- `-d` : Lance le conteneur en arrière-plan (mode *détaché*), libérant ainsi le terminal.
- `-p 8080:80` : Configure la redirection de port.
 - Le port **80** est le port d'écoute standard du serveur Nginx dans le conteneur.
 - Le port **8080** est celui que j'utilise sur ma machine pour accéder au site (via `http://localhost:8080`)

- **-v** ... : Configure le volume (le point de montage).
 - **Source** : \$(pwd)/index.html (le chemin vers mon fichier sur mon ordinateur).
 - **Destination** : /usr/share/nginx/html/index.html (l'emplacement où Nginx s'attend à trouver la page d'accueil).

4 Conclusion

Cette manipulation démontre la puissance des volumes Docker. Grâce à cette configuration, le conteneur ne sert plus sa page par défaut, mais "lit" directement mon fichier local.

Résultat obtenu : Toute modification que j'apporte à mon fichier `index.html` est immédiatement visible sur le navigateur après un simple rafraîchissement, ce qui offre un flux de travail fluide et efficace pour le développement web.