diginamic FORMATION
« Votre passeport pour l'emploi numérique »

☑ Bonnes pratiques avec Docker

Sommaire

Environnements séparés

- Utiliser un fichier `docker-compose.override.yml` pour le développement local
- Créer une image de production plus légère avec un **multi-stage build**
- Séparer les variables (`.env.dev`, `.env.prod`)

Sécurité

- Ne jamais inclure de **secrets** dans le code ou dans les images
  - Utiliser des variables d'environnement injectées au runtime
- Ajouter un fichier `.dockerignore` pour éviter d'inclure des fichiers sensibles ou inutiles (ex: `.env`, `.git`, `__pycache__`)

Optimisations

- Partir d'**images officielles** ou maintenues (ex: `python:3.11-slim`)
- Supprimer le cache lors de l'installation des dépendances :

```
RUN pip install --no-cache-dir -r requirements.txt
```

- Minimiser le nombre d'instructions dans le `Dockerfile` pour accélérer les builds

- Ne copier que ce qui est nécessaire (`COPY requirements.txt .` avant `COPY . .`)