

Consignes du Projet Docker

1. Dates à retenir

- **Date de distribution** : 17/01/2025
- **Date de rendu** : 07/02/2025 (3 semaines après), à 23:59:59

Assurez-vous de respecter cette échéance.

2. Livrables

2.1 Repository GitHub

1. **Code source complet** : Tous les fichiers nécessaires au projet.
2. **Fichier `Dockerfile`** : Permettant de construire l'image Docker de votre application.
3. **Fichier `docker-compose.yml`** : Permettant de lancer les différents services.

2.2 Documentation

1. **README.md** :
 - **Fonctionnement global du projet** : Expliquez l'usage de votre application (par exemple, un site web, une API, etc.).
 - **Rôle de chaque conteneur** : Présentez brièvement les services (backend, base de données, etc.).
 - **Contenu et logique du `Dockerfile`** : Décrivez les étapes principales (FROM, COPY, RUN, etc.).
 - **Logique du `docker-compose.yml`** : Présentez les services, la configuration réseau (networking), les volumes, etc.
 - **Guide de lancement** : Indiquez clairement les commandes pour :
 - Construire l'image (e.g. `docker build -t votre_image .`)
 - Lancer l'environnement complet (e.g. `docker-compose up -d`)
 2. **Brève explication du code** :
 - Donnez un aperçu de la finalité de l'application.
 - Décrivez les grandes lignes de votre logique métier (sans entrer dans tous les détails).
-

3. Points clés pour le `docker-compose`

- **Nombre de services** : Au moins 2 conteneurs distincts (exemples : un backend + une base de données, ou un frontend + un backend).
 - **Networking** : Les conteneurs doivent communiquer entre eux (réseau partagé, liens, etc.).
 - **Volumes** : Gérez au moins un volume pour la persistance de données (fichiers de configuration, base de données, etc.).
-

4. Optimisation (point bonus)

- **Optimiser le Dockerfile** :
 - Expliquer les optimisations (par exemple, multi-stage build, utilisation de versions minimales, réduction du nombre de couches, etc.).
 - Justifier pourquoi ces optimisations sont utiles (réduction de la taille de l'image, amélioration des temps de build, déploiement plus rapide, etc.).