

# Sujet du Projet :

## Conteneurisation d'une Application Full Stack avec Docker

### Contexte

Dans le cadre de ce projet, vous allez conteneuriser une application full stack existante que vous avez développée pour votre projet annuel (vous pouvez utiliser une autre application, mais il faut nécessairement un front-end, un back-end et un DB). L'objectif est de mettre en pratique vos compétences en Docker et Docker Compose pour déployer chaque composant de l'application (front-end, back-end, et base de données) dans des conteneurs Docker distincts. Vous orchestrerez ces services à l'aide de Docker Compose, en mettant l'accent sur la configuration des volumes pour la persistance des données et la mise en place de réseaux personnalisés pour une communication sécurisée entre les services.

### Objectifs

- Appliquer la conteneurisation à une application full stack existante.
- Maîtriser l'utilisation de Docker et Docker Compose pour l'orchestration des services.
- Configurer la persistance des données et la communication sécurisée entre les services à l'aide des réseaux Docker.

### Exigences Techniques

- **Conteneurisation** : Chaque composant de l'application (front-end, back-end, base de données) doit être conteneurisé.
- **Docker Compose** : Utilisation de Docker Compose pour orchestrer le déploiement des services, y compris la configuration des volumes et des réseaux.
- **Persistance des Données** : Utilisation de volumes Docker pour garantir la persistance des données de la base de données.
- **Réseau Docker** : Configuration d'un réseau Docker pour sécuriser la communication entre les conteneurs du back-end et de la base de données.
- **Images Docker** : Vos images doivent être poussées sur un repository Docker HUB.

### Documentation

Votre projet doit inclure une documentation comprenant :

- Instructions détaillées pour construire et démarrer l'application avec Docker et Docker Compose.
- Explication de l'architecture Docker de l'application, y compris la description des services, volumes, et réseaux configurés dans Docker Compose.

- Guide sur la manière de tester la communication entre les conteneurs et la persistance des données.

## Livrables

- Lien vers votre repository GIT Public contenant le code source de l'application ainsi que les fichiers docker (Dockerfiles et docker-compose.yml, etc..).
- Lien vers votre repository Docker HUB, contenant vos images.
- Documentation du setup de l'application.

## Notez bien :

Votre évaluation ne portera pas sur le code de votre application ni sur les fonctionnalités qu'elle propose.

## Barème (Total : 100 points)

- 1. Conteneurisation (30 points)**
  - **Front-end conteneurisé (10 points):** Correctement conteneurisé et fonctionnel.
  - **Back-end conteneurisé (10 points):** Correctement conteneurisé et fonctionnel.
  - **Base de données conteneurisée (10 points):** Correctement conteneurisée et fonctionnelle.
- 2. Configuration Docker Compose (40 points)**
  - **Services (15 points):** Configuration adéquate des services dans docker-compose.yml, démontrant une compréhension de l'orchestration des conteneurs.
  - **Volumes (10 points):** Mise en place efficace des volumes pour assurer la persistance des données de la base de données.
  - **Réseaux (15 points):** Configuration d'un ou plusieurs réseaux Docker pour sécuriser la communication entre les conteneurs, avec une explication claire de la stratégie de réseau choisie.
- 3. Gestion des images (15 points)**
  - **Mise en œuvre (15 points):** Nommages des images et de leurs versions. Images "latest" fonctionnelles sur Docker HUB.
- 4. Documentation (15 points)**
  - **Complétude et Clarté (15 points):** Documentation détaillée incluant les instructions pour construire, démarrer, et interagir avec l'application; explications sur l'architecture Docker mise en œuvre; description des choix de volumes et réseaux.