

# Projet Containers

MSI 4-26 DO ISI RENNES

## Contexte et définition des besoins

Pour égayer la journée de ses étudiants, l'école Sup de Vinci a missionné un groupe d'étudiants à développer l'application **Joke of the day**.

Après de longues recherches, les étudiants ont choisi de mettre en place une API et une application Front-End. Ils se sont répartis en deux équipes indépendantes pour implémenter ces deux applications.

L'équipe API a choisi d'utiliser le langage Rust (v1.82.0) pour construire une application performante et peu consommatrice en ressources. Ils ont utilisé le framework web [Rocket](#).

L'équipe Front-End a quant à elle choisi de développer en Javascript à l'aide du framework Next.JS (v14) propulsé par Node.JS (v23.0.0).

Les 2 applications communiqueront entre elles à l'aide de requêtes HTTP.

Le code source de leurs applications est disponible dans ce dépôt Git :  
<https://github.com/baiello/sdv-m1do-containers-project>.

Leur développement avance et les applications sont fonctionnelles, mais les développeurs sont confrontés à des problèmes d'environnement, tant en développement qu'en production. Ces conflits les ralentissent dans leur travail et ils aimeraient disposer d'environnements standardisés. Ils ont donc fait appel à vous, expert DevOps, pour mettre en place une solution utilisant Docker.

Pour l'environnement de développement, chaque équipe aimerait pouvoir lancer facilement la stack applicative complète (API + Front-End) en un claquement de doigts. Les 2 applications devront être capable de se recharger automatiquement à chaque modification de leur code source.

Pour l'environnement de production, ils attendent de vous que vous mettiez en place une automatisation de création de conteneurs optimisés, qui se déclenchera à chaque fois qu'ils mettront à jour la branche principal de leur dépôt. Les conteneurs ainsi créés (un pour l'API et un pour l'application Front-End), devront ensuite être mis à disposition sur [Docker Hub](#).

# Consignes et livrables

Ce projet sert d'évaluation au module **Containers**. Il devra être réalisé de façon **individuelle**.

Il vous est demandé de :

- Créer un fork (une copie) du [dépôt de base](#) ;
- Le mettre à jour avec la stack Docker de développement ;
- Mettre à jour le fichier README.md avec les explications permettant de lancer le projet ;
- Configurer une pipeline d'intégration continue qui construira les images Docker de production, optimisées (une image pour l'API et une image pour l'application Next.JS)
- Ajouter un lien vers votre registre [Docker Hub](#) dans le fichier README.md

**Date limite de rendu :** Lundi 21 Octobre 2024 à 23h59.

L'accès au dépôt devra être fourni à votre formateur avant la date limite de rendu.

Tout retard sera pénalisé ainsi que tout commit ajouté entre la deadline et la remise des notes.

# Barème

Critère	Barème
Mise en place de la stack de développement	/ 5
Mise en place de l'intégration continue	/ 5
Optimisation des images de production	/ 5
Cohérence et organisation des fichiers de configuration	/ 5