

# Dockerize Everything! - 2023/2024

---

## Pré-requis

---

### Identifier les développeurs du projet

Vous devez mettre un fichier AUTHORS à la racine de votre projet. Celui-ci contiendra le nom complet et l'adresse e-mail MyGES de chaque membre de l'équipe. Le format du fichier est décrit dans la section Données du projet. Ce fichier permet à l'intervenant de corriger plus facilement les projets.

Si le fichier n'est pas présent au moment du rendu, vous perdrez 2 points sur la note finale de tous les membres du groupe.

### ⚠ Ne pas tricher ⚠

Tricher, tout comme partager du code source ou des tests logiciels, est strictement interdit et est passible de sanctions et/ou de signalements au directeur pédagogique.

### Rendre un dépôt Git propre

Votre dépôt Git doit être propre au moment de la soutenance. Un dépôt de code "sale" contient notamment (liste non exhaustive) :

- Des fichiers binaires ;
- Des fichiers avec des droits incorrects (pas de lecture, écriture ou exécution autorisées) ;
- Des fichiers interdits : `*~`, `*.swp`, `*.DS_Store`, `.idea`, `*.class`, `*.log`, `*.core`, etc.

Présenter un dépôt "sale" fera perdre 2 points sur la note finale de tous les membres du groupe.

## Conseils

---

- Lisez entièrement le sujet avant de commencer à coder.
- Au moindre problème lié au projet (format du sujet, etc), vous pouvez contacter vos intervenants.  
Envoyez un e-mail à vos intervenants (voir Données de projet) pour toute question à propos du projet ou du sujet. Ne pas oublier d'utiliser ce préfixe pour vos e-mails, cela facilite la communication : `[ESGI][3AL<numéro de promotion>][DevTools]` `<objet>`.  
Ex : `[ESGI][3AL1][DevTools]` Quelle version de docker-compose à utiliser ?
- Démarrez le projet le plus tôt possible

## Données du projet

---

## Membres par équipe

Minimum : 2

Maximum : 3

## Fichiers **AUTHORS**

Le fichier `./AUTHORS` (**obligatoire !**) liste les membres de l'équipe projet avec la convention suivante : Prénom, un espace, Nom de famille, une ou plusieurs tabulations, un chevron gauche, l'adresse e-mail, un chevron droit et un saut de ligne. Cela donne :

```
cat AUTHORS
```

```
Hana Song <hana.song@example.com>
Jesse McCree <jesse.mccree@example.com>
Genji Shimada <genji.shimada@example.com>
Olivia Colomar <olivia.colomar@example.com>
```

## Objectif

Industrialiser les trois applications présentes dans [ce dépôt Git](#).

Industrialiser une application signifie que vous devez installer, paramétrer et optimiser des outils afin qu'une équipe de développement de toute taille puisse installer, développer et déployer l'application le plus facilement et le plus rapidement possible.

## Consigne

Pour réussir, vous allez devoir créer un environnement de production partageable avec tous les développeurs de votre équipe avec docker-compose, et une image de production, pour chacune des trois applications, avec Docker.

Tous les détails sont présents dans le README de chacune des trois applications.

## Critères d'évaluation

- Vous devrez exécuter les applications correctement via l'environnement de production conteneurisé que vous aurez créé ;
- Vous devrez être capables d'expliquer le processus de conteneurisation lors de la soutenance ;
- Vous devrez savoir justifier l'utilisation d'une commande ou d'une instruction Dockerfile et/ou docker-compose lors de la soutenance.

## Conseils pour le projet

- N'hésitez pas à vous appuyer sur le [dépôt flaskr-solution](#) pour apprendre à conteneuriser une

application.

- Séparez les différents services dans des conteneurs adaptés : par exemple, votre application tournera dans un conteneur dédié, et votre base de données tournera dans un conteneur séparé ; ces deux conteneurs devront évidemment pouvoir communiquer entre eux.

Bon courage !

- Thomas Domingues