

# Guide d'utilisation

03-04-2023



## SAE S4 – Groupe B1

CANALE Enzo – GANASSI Alexandre  
GONZALES Lenny – KADDOURI Mohammed – SAUVA Mathieu

Tout d'abord, notre projet est presque entièrement virtualisé grâce à des conteneurs Docker. En effet, le serveur de notre application, notre site web et notre base de données sont conteneurisés.

# Introduction à Docker et Docker Compose

Pour comprendre notre manière de penser pour vous donner les meilleures conditions d'utilisations, voici une brève explication de Docker et Docker Compose.

Docker est une plateforme de conteneurisation qui permet de créer, de déployer et de gérer des applications dans des conteneurs.

Chaque conteneur Docker est un processus isolé qui exécute une application, de sorte qu'elle peut fonctionner sur n'importe quel système (Windows, MacOS, Linux, ...) compatible Docker.

Supposons que vous avez produit un site web et que vous devez le déployer sur différents serveurs pour que les utilisateurs puissent y accéder.

Donc, sur votre ordinateur, vous devrez installer Apache, PHP, ..., puis, configurer le serveur, choisir la version de php utilisé, ...

Cela fonctionne enfin et les utilisateurs peuvent accéder à votre site web.

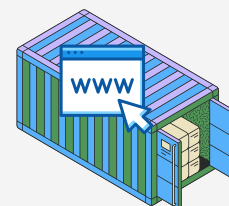
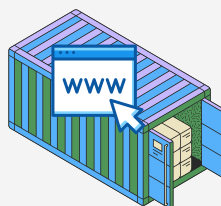
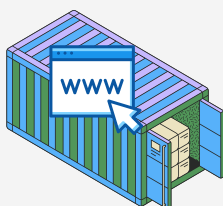
Cependant, un jour, vous changez d'ordinateur, vous devrez donc re-installer tous les logiciels, configurer le serveur, choisir la bonne version de php, ... sur votre nouvel ordinateur.

Cela peut être fastidieux et prendre beaucoup de temps, sans compter le risque d'erreurs humaines.

C'est ici que Docker intervient.

Docker permettra d'automatiser cela à l'aide d'un fichier appelé Dockerfile contenant un nombre d'instructions (transfère de fichier, commandes à exécuter dans un terminal, ...).

Donc, lorsque vous changerez d'ordinateur, vous n'aurez plus qu'à exécuter ce fichier pour déployer votre site web.



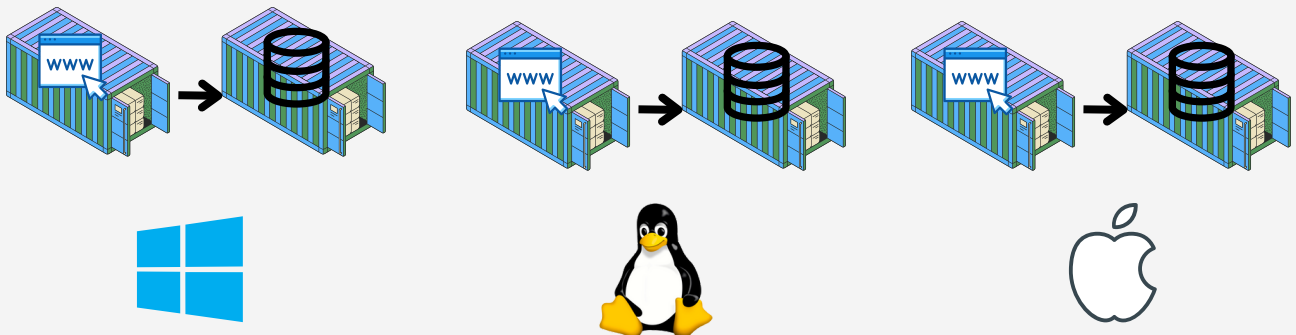
Supposons maintenant que vous avez ajouté une base de données à votre projet et que votre site web utilise cette base de données (inscription et connexion de l'utilisateur), donc vous vous dites naturellement que vous allez créer 2 Dockerfile, un pour votre site web et un autre pour votre base de données.

Mais cela n'est pas la meilleure solution.

En effet, vous pouvez utiliser Docker Compose qui permet de décrire les différents services (site web, base de données) de votre projet, mais aussi les différentes dépendances entre les services (le site web dépend de la base de données), les réseaux permettant de connecter les conteneurs de manière efficace, ...

Docker Compose permet d'exécuter tous vos services en une seule commande à partir d'un fichier nommé `docker-compose.yml`.

Comme vous l'avez compris, Docker Compose facilite la création et la gestion de réseaux de conteneurs complexes, tout en assurant une cohérence et une portabilité maximale des environnements de développement, de test et de production, ...



# Guide d'utilisation

Notre projet contient exactement 3 services :

- Site web
- Serveur de l'application
- Base de données

Nous avons donc produit pour chaque service, un Dockerfile. Puis, pour exécuter tous ses services en une seule commande et pour spécifier les dépendances et le réseau, nous avons utilisé un docker-compose.yml.

Maintenant, passons à l'exécution du projet.

Tout d'abord, veuillez installer Git à partir de ce lien :

<https://git-scm.com/downloads>

Pour vous permettre d'exécuter ses services, nous avons produit un répertoire Github énumérant toutes les étapes à suivre :

<https://github.com/LennyGonzales/SAES4-Dockers>

Sinon, vous pouvez directement suivre les étapes ci-dessous :

## Étape 1 : Installation de Docker

Suivez ce lien pour installer Docker :

<https://docs.docker.com/get-docker/>

## Étape 2 : Installation de Docker Compose

Tout d'abord, installez pip à partir de ce lien :

<https://pip.pypa.io/en/stable/installation/>

Puis, installez Docker-Compose en exécutant une de ces commandes :

```
pip install docker-compose
ou
pip3 install docker-compose
```

## Étape 3 : Téléchargement ou Clonage du projet Github

Vous pouvez soit télécharger notre répertoire Github en téléchargeant le zip se trouvant à ce lien :

<https://github.com/LennyGonzales/SAES4-Dockers>

Sinon, vous pouvez cloner notre répertoire Github en exécutant la commande :

```
git clone https://github.com/LennyGonzales/SAES4-Dockers.git
```

Étape 4 : Téléchargement de nos autres services (site web et serveur de l'application).

Téléchargez notre serveur et notre site web en exécutant la commande suivante :

```
bash script.sh
```

Étape 5 : Exécutez le docker-compose.yml

```
docker-compose up --build -d
```