

Guide Installation Docker _00_ASBank2023

Afin de permettre une installation plus rapide du projet et de son environnement, nous avons choisi de dockeriser les outils du projet. <https://fr.wiktionary.org/wiki/dockeriser>

Ainsi, en partageant les Dockerfiles et le docker-compose, nous permettons à nos collaborateurs d'installer tout l'environnement tout en ayant exactement le même. Afin d'utiliser cela, il faut permettre à sa machine l'utilisation de Docker et également comprendre les commandes à utiliser dans celui-ci.

Ce document a donc pour objectif d'expliquer l'installation de Docker, des images de notre projet, mais également les commandes à utiliser pour le faire fonctionner.

Installation de Docker

Windows

Pour windows il suffit d'installer l'application [Docker Desktop](#), lancer le fichier .exe puis suivre les instructions. La création de compte n'est pas obligatoire et donne seulement quelques fonctionnalités en plus.

Afin de faire fonctionner le docker il faut également s'assurer que wsl2 est installé sur sa machine, le tuto de Microsoft [Installer WSL2 sur Windows](#) couvre très bien le sujet, nous choisissons de garder la distribution Ubuntu qui est celle de base.

Ainsi il sera possible d'utiliser des commandes docker dans les terminaux lorsque docker desktop est lancé.

Linux

Pour installer docker sur un linux, pas besoin de docker desktop, l'installation demandera l'utilisation du terminal et quelques commandes. Vous pouvez les retrouver sur le guide officiel de [DockerDocs](#) afin d'installer le docker engine.

Utilisation de Docker

Après avoir téléchargé le dossier du projet, installé docker et exécuté docker desktop si vous êtes sur windows nous pouvons construire les images et lancer les conteneurs afin de lancer l'application.

Pour ce faire nous ouvrons un premier terminal dans la racine du projet, nous utiliserons ce terminal afin de construire les images du docker compose et démarrer les conteneurs.

```
C:\Users\LFste>cd C:\Users\LFste\OneDrive\Bureau\Dossier\Cours\A3\semestre_5\QualDev\project\But3DevOpsG5\_00_ASBank2023
C:\Users\LFste\OneDrive\Bureau\Dossier\Cours\A3\semestre_5\QualDev\project\But3DevOpsG5\_00_ASBank2023>_
```

Déplacé dans la racine du projet nous pouvons lancer le docker compose avec la commande :
docker compose up --build -d

```
C:\Users\LFste\OneDrive\Bureau\Dossier\Cours\A3\semestre_5\QualDev\project\But3DevOpsG5\_00_ASBank2023>docker compose up --build
[+] Running 12/27
  phpmyadmin [-----] Pulling
    - faef57eae888 Downloading [ > ] 294.9kB/29.12MB
    - 989a1d6c052e Download complete
    - 0705c9c2f22d Downloading [ > ] 540.7kB/104.3MB
    - 621478e043ce Waiting
    - 98246dcca987 Waiting
    - bfed8c155cb6 Waiting
    - 7a7c2e988867 Waiting
    - d176994b625c Waiting
```

Ainsi les images de mysql, phpmyadmin, tomcat, et maven seront construites. A la fin de leurs constructions des conteneurs seront exécutés avec ces images.

A la fin de la compilation du projet et après le lancement du serveur tomcat, il sera accessible dans le navigateur avec le lien **localhost:8080/_00_ASBank2023/** . Le lien n'est pas le lien définitif est demandera une modification par la suite.

Afin de compiler le projet il n'est pas nécessaire de relancer la construction des images, il suffit d'utiliser une commande afin de rentrer dans le container :

Docker exec -ti asbank_maven bash

En étant dans le terminal du container de maven nous pouvons compiler en utilisant la commande :

mvn install

Dans le context du développement, la commande ci-dessous permet de générer les stats du

test coverage du projet, ceci seront installé dans le dossier du projet
.\target\site\jacoco\index.html

mvn jacoco:prepare-agent test install jacoco:report -P docker

Si vous souhaitez afficher l'aperçu directement des lignes couvertes par les tests, je vous invite à suivre :

<https://stackoverflow.com/questions/51964147/java-code-coverage-in-visual-studio-code>

Il faudra sur vs code installer les extensions :

Coverage Guuter

Live Previews

Liste des commandes :

- **docker compose up** : Lance les conteneurs du fichier docker-compose.yml sans construire les images
- **docker compose up --build** : Lance les conteneurs après avoir build les images du fichier docker-compose.yml
- **docker compose down** : supprime tout ce qui est défini dans le docker-compose.yml sauf les volumes persistants (la base de données) et les images
- **docker compose down -v** : supprime tout ce qui est défini dans le docker-compose.yml avec également les volumes persistants et les images.