**Enoncé Chapitre 1**

1. Créer un repo sur GitHub et nommez le « **Formation\_DevOps** »
2. Créer un projet Maven QuickStart avec Maven artifact
3. Créer un module SpringBoot nommé **product-service** avec Spring Initializer contenant les dépendances Spring Web, Spring Data JPA et MySQL Driver.
4. Créer un serveur MYSQL soit en installant MYSQL ou en lançant un conteneur MYSQL Docker

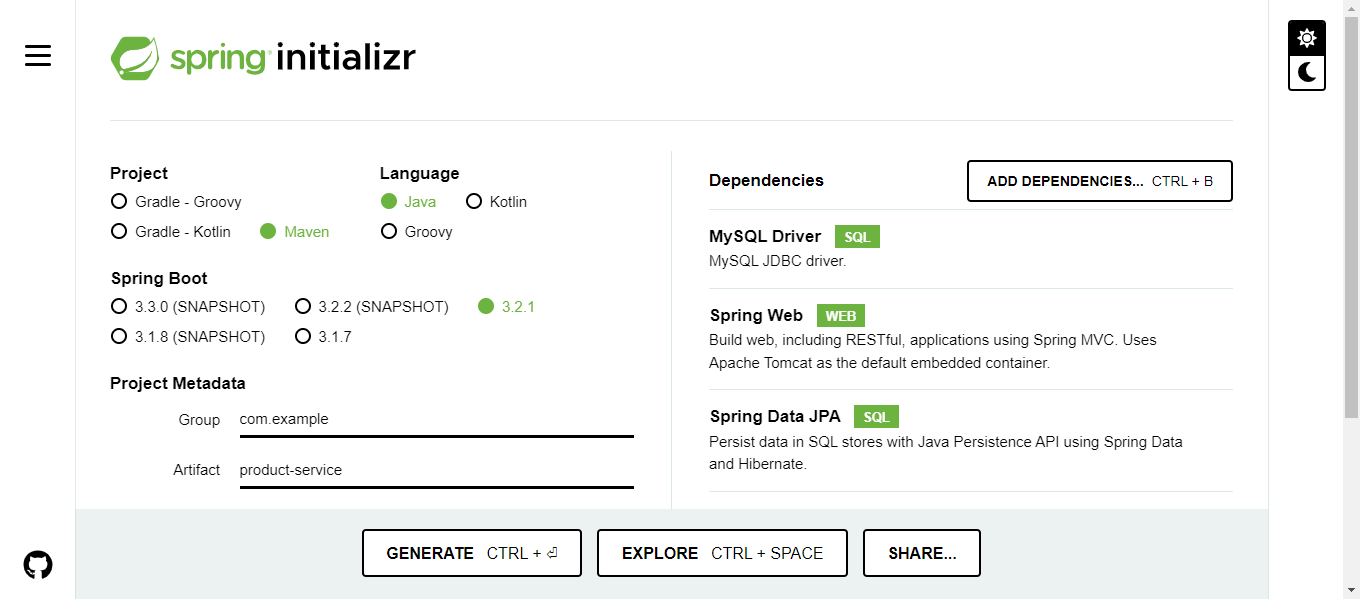
**Solution**

* [Créer](https://github.com/aymendr/formation_devOps_prise_en_mains) le repo sur Github
* Faire un git clone depuis votre machine
* Vérifier la branche sur laquelle vous êtes positionné

1. Utiliser ce lien pour la création du projet MAVEN : <https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html>
2. Aller sur <https://start.spring.io/>

Vous pouvez directement suivre ce lien pour télécharger le projet :

<https://shorturl.at/bmqQZ>



* Ouvrir le projet avec votre IDE préféré (moi j’utilise Intellij Idea)

1. Aller sur <https://hub.docker.com/_/mysql>

Prérequis : Docker installé

* Lancer mysql avec Docker

$ docker run -p 3306:3306 --name mysql -e MYSQL\_ROOT\_PASSWORD=root -d mysql:latest

* Se connecter au container mysql et créer la base de donnée

$ docker exec -it mysql mysql -p

* Créer la base de donnée

create database product\_db;

* Créer l’utilisateur avec lequel on va lire/ecrire dans la base depuis l’application springboot et lui attribuer les privilèges nécessaires

mysql> create user 'springuser'@'%' identified by 'springuser';

mysql> grant all on product\_db.\* to 'springuser'@'%';

* Suivre les étapes spring jpa pour mettre en place les operations crud au sein de l’application: <https://spring.io/guides/gs/accessing-data-mysql/>