

* Environnement

- WSL 2
- Docker Desktop ①

* Dépôt Github de travail ②

https://github.com/Florianspk0/toulouse_bd_13

Récupérer le code

NE PAS OUBLIER DE RAMENER LES PACKAGES

`$ npm i`

ou `$ npm install`

* Renommer .env.example → .env

* Verification BDD

→ en créer pour le projet. ③
version mariadb sur linux

⚠ Le start donne POUF

↳ Passer sur docker pour simplifier la vie ④

* Premier start de l'API

✓ .env

✓ les bons params dans .env

↳ voir "scripts" dans package.json

`$ npm run dev`

* Mise en place des tests.

⚠ Def une stratégie de test

1 test → router principal

2 test → route user (CRUD)

→ /test à créer

↳ main-route.test.js

↳ user-route.test.js.

* Qualité & Sécurité.

i) SOUVERAINETÉ. ⑤

Docker Desktop + WSL 2

→ Settings > Resources > WSL Integration

↳ Activer la "distros" désirée

↳ En bas à gauche "Apply & Restart"

↳ Test: - Dans une distro linux
- Taper la commande

\$ docker version

Projet venant de github.

(A) `git clone [url]`

↳ créer un dossier (nom du dépôt)

⚠ Ramène aussi la config du dépôt.

(B) `git clone [url] [chemin]`

↑ habituel ". "
↳ là où on est

Plus pratique car pas de nouveau dossier **MAIS** fig la config git.

(C) La Méthode de git

`$ degit [url]`

Pas de dossier

Pas de config git

→ Install

`$ sudo npm i -g degit`

Base de Données

* Vérifier si mariadb tourne

```
$ sudo service mariadb status
```

si off →

```
$ sudo service mariadb start
```

* BDD & user pour le projet

```
$ mysql -u admin -p
```

```
$ create database [nom];
```

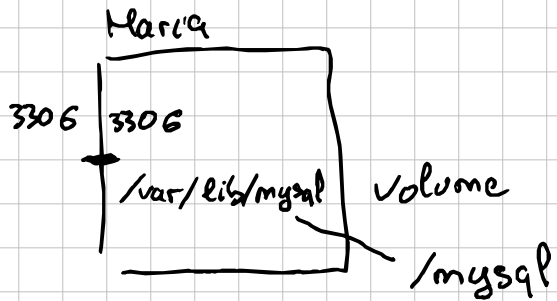
```
$ grant all on [bdd].* to 'user'@'localhost'  
identified by 'password';
```

```
$ exit
```

Base de donnée en docker.

* Image mariadb.

Idée →



→ Faire un docker-compose.yml

version: '3.1'

services:

db:

image: mariadb:latest

restart: always

volumes:

- /mysql:/var/lib/mysql

ports:

- 3306:3306

environment:

MARIADB_ROOT_PASSWORD:

MARIADB_DATABASE:

" - USER:

" - PASSWORD:

Start: \$ docker-compose up -d

Normalement

- dossier mysql

① - un conteneur en ligne

② - un service sur le port 3306

① \$ docker ps // liste de conteneurs

② \$ sudo netstat -tlnp

⚠ Attention si une base tourne en local style mariadb, penser à la stopper.

\$ sudo service mariadb stop

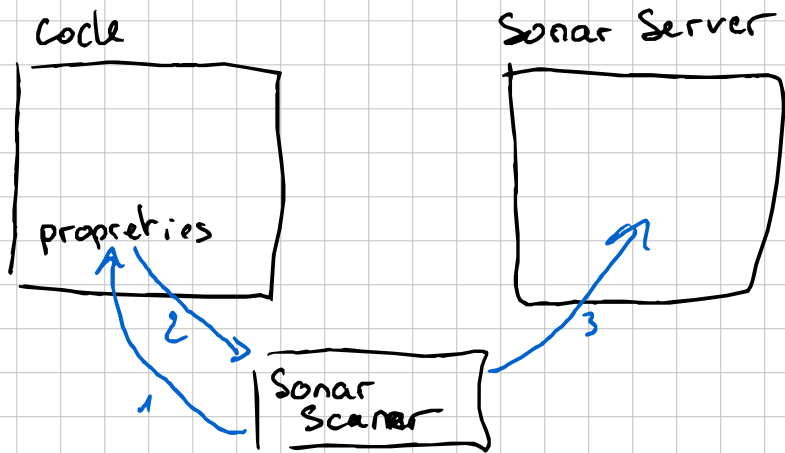
Relancer le compose histoire d'être propre

\$ docker-compose down

\$ docker-compose up -d

SONAR QUBE.

Rappel



- 1- Lancer le serveur → docker
- 2 → Préparer le fichier de propriétés
- 3 → lancer le scanner → docker.

1) Lancement serveur (de pref hors projet)

Fichier : sonar-server.sh

```
$ chmod u+x sonar-server.sh  
$ ./sonar-server.sh
```

Compte par défaut admin/admin

2) Créer un projet.

ProjectKey
ProjectName) Ecrire le nom du projet

nommer le token avec le m même nom que le projet (plus pratique)

Copier/Coler le token dans scan.sh

→ SONAR-TOKEN

3) Mettre le scan.sh au même niveau que le fichier sonar-project.properties

4) Lancer le scanner

```
$ chmod u+x scan.sh  
$ ./scan.sh
```

5) Regarder ou pleurer le résultat sur le serveur sonar.