**Projet DevOps**

**Partie Backend**

1. Télécharger le dossier : **nodejs-express-sequelize-mysql-master** a partir de ce lien

<https://drive.google.com/file/d/1eoj5IjXONB_sFQu7kXwA-yr2O6ny94We/view?usp=sharing>

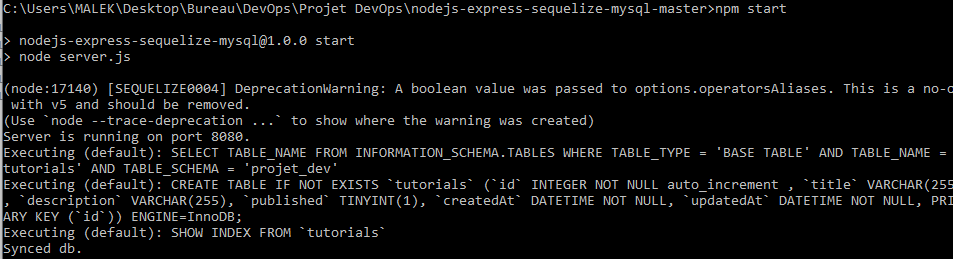
1. Démarrer MySQL et créer la base de **données projet\_dev**



1. Démarrer l’invite des commandes taper l’instruction suivante :



1. Une fois vous avez installé les dépendances nécessaires veuillez taper la commande suivante :



**Partie Frontend**

1. Télécharger le dossier react-crud-web-api-master a partir de ce lien

<https://drive.google.com/file/d/1Wxm_mYJxDRVUuapXoc-kn3JHyQQ14f_A/view?usp=sharing>

1. Dans l’invite des commandes taper l’instruction suivante :



1. Dans la ligne des commandes taper la commande suivante :

**Partie DevOps**

**1. Pusher les applications dans Github**

**2. Conteneurisation avec Docker**

* Écrivez un Dockerfile pour conteneuriser l'application backend Node.js.
* Écrivez un Dockerfile pour conteneuriser l'application frontend React Js.

**3. Configuration avec Docker Compose**

* Créez un fichier docker-compose.yml pour exécuter les conteneurs suivants :
  + Frontend React.
  + Backend Node.js.
  + Base de données MySQL.

**4. Pipeline CI/CD avec Jenkins**

* Écrivez un pipeline Jenkins (fichier Jenkinsfile) pour :
  + Construire les images Docker du frontend et du backend.

**5. Gestion des artefacts avec Nexus**

* Décrivez comment configurer Jenkins pour pousser les images Docker dans Nexus.

**6. Analyse de la qualité du code avec SonarQube**

* Comment intégrer SonarQube dans un pipeline Jenkins pour analyser le code React et Node.js ?

**7. Déploiement dans Kubernetes**

* Déployer l’application sur un Cluster Kuberntes

**8. Surveillance et gestion**

* Configurer prometheus et grafana pour surveiller l'utilisation des ressources des conteneurs.