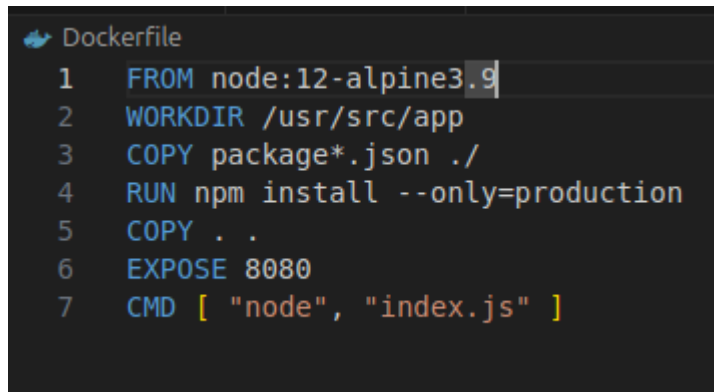


## Mise en pratique - Docker Application Express JS

1. Lien du git : [https://github.com/ACAPO21/TP2\\_Docker](https://github.com/ACAPO21/TP2_Docker)

2. Compléter le Dockerfile afin de builder correctement l'application contenu dans src/



```
Dockerfile
1 FROM node:12-alpine3.9
2 WORKDIR /usr/src/app
3 COPY package*.json ./
4 RUN npm install --only=production
5 COPY . .
6 EXPOSE 8080
7 CMD [ "node", "index.js" ]
```

a. Une option de npm vous permet de n'installer que ce qui est nécessaire. Quelle est cette option ? Quelle bonne pratique Docker permet t-elle de respecter ?

- L'option de npm pour n'installer que ce qui est nécessaire est --only=production. Cette option permet d'installer uniquement les dépendances nécessaires pour l'exécution de l'application en production, en excluant les dépendances de développement.
- Bonne pratique Docker : En utilisant cette option, nous réduisons la taille de l'image Docker et minimisons les vulnérabilités potentielles en n'incluant pas les outils de développement inutiles.

3. A l'aide de la commande docker build, créer l'image ma\_super\_app

```

ubuntu@ubuntu:~/docker_js$ sudo docker build -t ma_super_app .
[+] Building 7.4s (10/10) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 186B                               0.0s
=> [internal] load metadata for docker.io/library/node:12-alpine3.9 1.5s
=> [internal] load .dockerignore                                  0.1s
=> => transferring context: 2B                                     0.0s
=> [1/5] FROM docker.io/library/node:12-alpine3.9@sha256:16d40e6c2858ee41cc7e19bb36f8a92718ad935ceae036e88dcfffb6 3.7s
=> => resolve docker.io/library/node:12-alpine3.9@sha256:16d40e6c2858ee41cc7e19bb36f8a92718ad935ceae036e88dcfffb6 0.0s
=> => sha256:16d40e6c2858ee41cc7e19bb36f8a92718ad935ceae036e88dcfffb68041dea6c 1.43kB / 1.43kB 0.0s
=> => sha256:ed9251aca55330890ef48a274c6ce03052e5438e87b6101b0bab5362ac79b5e5 1.16kB / 1.16kB 0.0s
=> => sha256:a7d6e4c06dd4f80d3ba52db58f3c7421f6c29497ddf084928103d71b6635314c 6.73kB / 6.73kB 0.0s
=> => sha256:31603596830fc7e56753139f9c2c6bd3759e48a850659506ebfb885d1cf3aef5 2.77MB / 2.77MB 0.5s
=> => sha256:f5b0d1ce6f59dcbee12d1ce6f4093fa9a7939c7751f00ff2137b4651dcf2de8f 24.49MB / 24.49MB 1.6s
=> => sha256:55922cf6f31687a003e818aab6c5e42db644b2be19a1c74040d750db3d63354c 2.24MB / 2.24MB 0.3s
=> => sha256:e8a7c0650cafa577c71ba40f4423d3486d700990904de11327a69bdf6f61d1a92 283B / 283B 0.5s
=> => extracting sha256:31603596830fc7e56753139f9c2c6bd3759e48a850659506ebfb885d1cf3aef5 0.2s
=> => extracting sha256:f5b0d1ce6f59dcbee12d1ce6f4093fa9a7939c7751f00ff2137b4651dcf2de8f 1.8s
=> => extracting sha256:55922cf6f31687a003e818aab6c5e42db644b2be19a1c74040d750db3d63354c 0.1s
=> => extracting sha256:e8a7c0650cafa577c71ba40f4423d3486d700990904de11327a69bdf6f61d1a92 0.0s
=> [internal] load build context                                0.0s
=> => transferring context: 1.68kB                                0.0s
=> [2/5] WORKDIR /usr/src/app                                0.3s
=> [3/5] COPY package*.json ./                                0.0s
=> [4/5] RUN npm install --only=production                    1.5s
=> [5/5] COPY . .                                             0.1s
=> exporting to image                                          0.1s
=> => exporting layers                                          0.1s
=> => writing image sha256:ecccd5d18de1273cf59a80f3fdf02b6fc4b488a0f0d89dbbd9871b41fc9ca22cb 0.0s
=> => naming to docker.io/library/ma_super_app                0.0s
ubuntu@ubuntu:~/docker_js$

```

4. Compléter le fichier docker-compose.yml afin d'exécuter ma\_super\_app avec sa base de données. /\ Utiliser correctement les variables d'environnement afin de configurer la base de données et l'application /\

```

1  version: '3.8'
2
3  services:
4    app:
5      image: ma_super_app
6      container_name: app_container
7      environment:
8        - DB_HOST=db
9        - DB_USER=root
10       - DB_PASSWORD=root
11       - DB_NAME=mydatabase
12      ports:
13        - "8080:8080"
14      depends_on:
15        - db
16
17     db:
18       image: mysql:5.7
19       container_name: db_container
20       environment:
21         MYSQL_ROOT_PASSWORD: root
22         MYSQL_DATABASE: mydatabase
23       volumes:
24         - db_data:/var/lib/mysql
25
26   volumes:
27     db_data:

```

```

ubuntu@ubuntu:~/docker_js$ sudo docker-compose up -d
WARN[0000] /home/ubuntu/docker_js/docker-compose.yml: `version` is obsolete
[+] Running 2/2
 ✓ Container app_container Started
 ✓ Container db_container Started

```