

Infrastructure

This document provides an overview and documentation for the API implemented using Docker containers and the provided configuration files (Dockerfile, docker-compose.yml, and .env).

- [Docker Container Setup](#)
- [Docker Compose Setup](#)

Docker Container Setup

The API is designed to run inside a Docker container. The container includes the necessary dependencies and configurations for the API to function properly. The Docker container is built using the provided `Dockerfile`.

Dockerfile

```
FROM node:20-alpine

LABEL maintainer="Hxtninfosec | MasterBigD | Ezeqielle"

# Create App directory for the project
RUN mkdir /app

# Change directory to /app
WORKDIR /app

# Copy all necessary files
COPY . .

RUN npm install --save
# If you are building your code for production
# RUN npm ci --omit=dev

# Set timezone
ENV TZ="Europe/Paris"

EXPOSE 5000

CMD [ "node", "app.js" ]
```

The `Dockerfile` sets up the Docker container by:

1. Using the `node:20-alpine` base image.
2. Creating an `/app` directory inside the container.
3. Changing the working directory to `/app`.

4. Copying all files from the current directory (the API source code) into the container's `/app` directory.
5. Installing the required dependencies using `npm install`.
6. Setting the timezone to `Europe/Paris` (change as needed).
7. Exposing port `5000` for the API to listen on.
8. Specifying the command to run the API using `CMD ["node", "app.js"]`.

Docker Compose Setup

To simplify the deployment and management of the API and its dependencies, Docker Compose is used. The `docker-compose.yml` file defines the services and their configurations.

docker-compose.yml

```
version: '3.1'

services:
  mariadb:
    image: mariadb
    restart: always
    environment:
      MARIADB_ROOT_PASSWORD: ${DB_ROOT_PASS}
    volumes:
      - ./main.sql:/docker-entrypoint-initdb.d/dump.sql
    networks:
      - backend

  adminer:
    image: adminer
    restart: always
    environment:
      ADMINER_DEFAULT_SERVER: mariadb
    depends_on:
      - mariadb
    networks:
      - backend
      - frontend
    ports:
      - 8080:${ADMINER_PORT}

  backend:
    build: ./backend
    depends_on:
```

```
- mariadb

volumes:
  - ./backend/app.js:/app/app.js
  - cthulhu-backend:/CTHULHU

environment:
  DB_HOST: mariadb
  DB_USER: ${DB_USER}
  DB_MAIN: ${DB_MAIN}
  DB_PASS: ${DB_ROOT_PASS}

networks:
  - backend
  - frontend

ports:
  - 5000:${BACKEND_PORT}
```

```
frontend:
  restart: always
  build: ./frontend
  ports:
    - 443:443
  networks:
    - frontend
```

```
networks:
  backend:
  frontend:
```

```
volumes:
  cthulhu-backend:
```

The `docker-compose.yml` file defines three services: `mariadb`, `adminer`, and `backend`.

mariadb Service

- Image: `mariadb`
- Restart: `always`
- Environment Variables:
 - `MARIADB_ROOT_PASSWORD`: The root password for the MariaDB instance (specified in `.env` file).
- Networks: `backend`

The `mariadb` service runs a MariaDB database server. It uses the specified root password and is connected to the `backend` network.

adminer Service

- Image: `adminer`
- Restart: `always`
- Environment Variables:
 - `ADMINER_DEFAULT_SERVER`: The hostname of the

MariaDB service (`mariadb`).

- Depends On: `mariadb`
- Networks: `backend`, `frontend`
- Ports:
 - `8080:${ADMINER_PORT}`: Maps the container's port 8080 to the specified `ADMINER_PORT` (specified in `.env` file).

The `adminer` service provides a web-based database management interface. It depends on the `mariadb` service and is connected to both the `backend` and `frontend` networks. The specified port is exposed for accessing the Adminer interface.

backend Service

- Build: `./backend`
- Depends On: `mariadb`
- Volumes:
 - `./backend/app.js:/app/app.js`: Mounts the local `app.js` file into the container's `/app/app.js`.
 - `cthulhu-backend:/CTHULHU`: Mounts the named volume `cthulhu-backend` to `/CTHULHU`.
- Environment Variables:
 - `DB_HOST`: The hostname of the MariaDB service (`mariadb`).
 - `DB_USER`: The database user (`root`).
 - `DB_MAIN`: The main database name (`CTHULHU`).
 - `DB_PASS`: The database user's password (specified in `.env` file).
- Networks: `backend`, `frontend`
- Ports:
 - `5000:${BACKEND_PORT}`: Maps the container's port 5000 to the specified `BACKEND_PORT` (specified in `.env` file).

The `backend` service builds the API using the source code located in the `./backend` directory. It depends on the `mariadb` service and mounts the local `app.js` file and the `cthulhu-backend` named volume. The environment variables define the database connection details. It is connected to both the `backend` and `frontend` networks, and the specified port is exposed for accessing the API.

.env File

```
DB_ROOT_PASS=toor
DB_USER=root
DB_MAIN=CTHULHU
BACKEND_PORT=5000
ADMINER_PORT=8080
```

The `.env` file contains environment variable definitions used by the Docker Compose configuration. It includes the following variables:

- `DB_ROOT_PASS`: The root password for the MariaDB instance.
- `DB_USER`: The database user (default: `root`).
- `DB_MAIN`: The main database name (default: `CTHULHU`).
- `BACKEND_PORT`: The port on which the API will listen (default: `5000`).
- `ADMINER_PORT`: The port on which Adminer will be accessible (default: `8080`).

Make sure to adjust these values as needed for your specific deployment.