

# Node.js: how to use Docker and Docker Compose to deploy your app

Building an application in Node.js can be a fun and rewarding experience, but managing the development environment can be a bit frustrating. Docker and Docker Compose are two powerful tools that can greatly simplify the process of building and deploying your app.

To get started, make sure you have Docker and Docker Compose installed on your computer. If you don't have them yet, you can download them from the official Docker site.

Once installed, you can create a `Dockerfile` file in the root of your project. This file will define the Docker image that will be used to run your application. Here is an example of a Dockerfile:

```
FROM node:18
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 3000
CMD ["npm", "start"]
```

This Dockerfile uses the `node:18` image as a base and sets the working directory to the `/app` folder. Then, it copies the `package.json` and `package-lock.json` files into your working directory and installs the

dependencies with `npm install`. Next, it copies the entire project to your working directory, exposes port 3000, and starts the app with `npm start`.

Now, you can create a `docker-compose.yml` file in the root of your project. This file will define the service that will run when you use the `docker compose up` command. Here is an example of `docker-compose.yml` file:

```
version: "3"
services:
  app:
    build: .
    ports:
      - "3000:3000"
    volumes:
      - .:/app
```

This file defines an app service which is built from the Docker image we defined in the Dockerfile. It exposes port 3000 and mounts the current directory in the `/app` directory inside the container.

Finally, you can start the application by running the `docker compose up` command. This command will build the Docker image, start the container and give you access to the application at `http://localhost:3000`.

In conclusion, using Docker and Docker Compose to build an application in Node.js is an efficient and easy way to manage your development environment. With a little practice, you'll be able to build and deploy your applications quickly and easily.