

VS Code interface showing the Dockerfile editor. The Explorer sidebar on the left shows the project structure: **EXPLORER**, **OPEN EDITORS** (Dockerfile, .dockerignore), **WEBSITE** (backend, JS data.js, JS server.js, node_modules, public, src), and **OUTLINE** (package-lock.json, package.json, README.md). The Dockerfile editor shows the following content:

```
1 FROM node:12
2
3 WORKDIR /
4
5 COPY package*.json ./
6
7 RUN npm install
8
9 COPY . .
10
```

The Dockerignore editor shows:

```
1 node_modules
```

The Terminal panel at the bottom shows the output of the `docker build` command:

```
PS F:\TE\Website> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
PS F:\TE\Website> docker build -t docker/app:1.0 .
[+] Building 2.0s (9/9) FINISHED
=> [internal] load .dockerignore
=> => transferring context: 52B
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 183B
=> [internal] load metadata for docker.io/library/node:12
=> [1/5] FROM docker.io/library/node:12@sha256:01627afeb110b3054ba4a1405541ca095c8bfca1cb6f2be9479c767a2711879e
=> [internal] load build context
=> => transferring context: 7.16kB
=> CACHED [2/5] COPY package*.json ./
=> CACHED [3/5] RUN npm install
=> CACHED [4/5] COPY . .
=> exporting to image
=> => exporting layers
=> => writing image sha256:55288c433a8000709b5db9aa461c7192b9b3b958e9fa5f99c9c6b95129d358ec
=> naming to docker.io/docker/app:1.0
PS F:\TE\Website>
```

VS Code interface showing the Dockerfile editor. The Explorer sidebar on the left shows the project structure: **EXPLORER**, **OPEN EDITORS** (Dockerfile, .dockerignore), **WEBSITE** (backend, JS data.js, JS server.js, node_modules, public, src), and **OUTLINE** (package-lock.json, package.json, README.md). The Dockerfile editor shows the following content:

```
1 FROM node:12
2
3 WORKDIR /
4
5 COPY package*.json ./
6
7 RUN npm install
8
9 COPY . .
10
11 ENV PORT=8080
12
```

The Dockerignore editor shows:

```
1 node_modules
```

The Terminal panel at the bottom shows the output of the `docker run` command:

```
PS F:\TE\Website> docker run -p 5000:8080 55288c433a8000709b5db9aa461c7192b9b3b958e9fa5f99c9c6b95129d358ec
> chat-app@0.1.0 start /
> react-scripts start

(node:26) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please
use the 'setupMiddlewares' option.
(node:26) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Pleas
e use the 'setupMiddlewares' option.
Starting the development server...

One of your dependencies, babel-preset-react-app, is importing the
"@babel/plugin-proposal-private-property-in-object" package without
declaring it in its dependencies. This is currently working because
"@babel/plugin-proposal-private-property-in-object" is already in your
node_modules folder for unrelated reasons, but it may break at any time.

babel-preset-react-app is part of the create-react-app project, which
```

Docker Desktop interface showing the Containers tab. The top bar includes the Docker logo, "Update to latest", a search bar, and a "Sign in" button. The left sidebar shows the navigation menu: **Containers**, **Images**, **Volumes**, **Dev Environments** (BETA), **Docker Scout** (EARLY ACCESS), **Learning Center**, **Extensions**, and **Add Extensions**. The main area displays the "Containers" tab with a search bar and a toggle for "Only show running containers".

Container CPU usage: 0.01% / 800% (8 cores allocated). Container memory usage: 332MB / 7.46GB. [Show charts](#)

Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
vigorously_raman 30dcf545a2f9	55288c433a8000709b5db9aa461c7192b	Running	0.01%	5000:8080	3 minutes ago	

Showing 1 item

RAM 4.41 GB CPU 0.13% Not connected to Hub v4.20.0