

How to create Dockerfile

A dockerfile is just a text file with a set of instructions telling docker how to build image (like a recipe for your app's environment).

Ex → In your react or vite or backend where you want.

Step 1 Base image

FROM node:20

Step 2 Working directory inside contains WORKDIR /app

Copy all source code

COPY ..

Copy & install dependencies
RUN npm install

Expose port for app
Expose 5173

Step 6: command to start app

CMD ["npm", "run", "dev", "--", "--host"]

* Building the docker image

Once you have your Dockerfile in the same directory as your app:

In terminal

docker build .

or

docker build -t my-node-app.

- = -t my-node-app tags the image with a name
- → means "build from current directory"

* Running the Container

Now run your image inside a container.

In terminal

docker run -ym -d -p 5173:5173 --name myapp myapp my-node-app

or
docker run -d -p 5173:5173 --name myapp imageid
(if name not set).

- = -d → detached mode
(run in background)
- = -p 5173:5173 → map your local port 5173
to container port 5173
- = --name my-container → name the container
- = my-node-app → image name.

Visit `http://localhost:5173` to see your app.

* After adding detached mode (background process).

= You can check logs later:

`docker logs my-app-container`

= Stop the container

`docker stop myapp-container`

check `docker ps` (if name not set)
a random name at last seen.
then

`docker stop random-name.`

* Remove the container

docker rm myapp-container

or

docker rm random-name

if on making container if add --rm then when stops it automatically delete the container.

* How it works internally

Think of it like packing your entire app into a mini-computer.

1. dockerfile: recipe of environment (base OS, language & dependencies).
2. docker build: creates an image (a read-only template).
3. docker run: launches a container (a running copy of that image).
4. Container: isolated from your system but shares the kernel → so it's light weight.

5. Port mapping + let your machine communicate with containerized app.

Visual flowchart

Your PC (host machine)

| Docker Engine (manages images/containers).

| Image (Template)

| | FROM node:20 → Install node.js

| | Copy files

| | RUN npm install

| | CMD npm start

| Container (running app)

| Isolated filesystem, processes, network.