# Node.js Dockerfile Explanation

This Dockerfile is for a Node.js application. Below is a detailed breakdown of each instruction and its purpose.  
  
------------------------------------------------------------  
Base Image  
------------------------------------------------------------  
FROM node:18-slim  
- Uses the official lightweight Node.js 18 slim image.  
- Slim variant keeps the image size smaller and build faster.  
  
------------------------------------------------------------  
Install Required Packages  
------------------------------------------------------------  
RUN apt-get update && apt-get install -y --no-install-recommends curl tini && rm -rf /var/lib/apt/lists/\*  
- Installs curl (for health checks and debugging).  
- Installs tini (a minimal init system) to handle zombie processes and signals, ensuring stable container behavior.  
- Cleans apt cache to reduce image size.  
  
------------------------------------------------------------  
Application Directory  
------------------------------------------------------------  
WORKDIR /usr/local/app  
- Sets the working directory inside the container.  
  
------------------------------------------------------------  
Install Nodemon (for development use)  
------------------------------------------------------------  
RUN npm install -g nodemon  
- Installs nodemon globally.  
- Nodemon automatically restarts the server when file changes are detected.  
- Mostly useful for development; not typically required in production.  
  
------------------------------------------------------------  
Copy Package Files  
------------------------------------------------------------  
COPY package\*.json ./  
- Copies package.json and package-lock.json into the container.  
- Allows Docker caching so dependencies are only reinstalled when these files change.  
  
------------------------------------------------------------  
Install Dependencies  
------------------------------------------------------------  
RUN npm ci && npm cache clean --force && mv /usr/local/app/node\_modules /node\_modules  
- npm ci installs dependencies from package-lock.json, ensuring reproducible builds and faster installs.  
- Cleans npm cache to reduce image size.  
- Moves node\_modules outside the app directory to /node\_modules.  
 This prevents overwriting dependencies when mounting volumes during development.  
  
------------------------------------------------------------  
Copy Application Code  
------------------------------------------------------------  
COPY . .  
- Copies the application source code into the container.  
  
------------------------------------------------------------  
Expose Port  
------------------------------------------------------------  
ENV PORT 80  
EXPOSE 80  
- Sets PORT environment variable to 80.  
- Declares port 80 for the application.  
  
------------------------------------------------------------  
Entrypoint and Command  
------------------------------------------------------------  
ENTRYPOINT ["/usr/bin/tini", "--"]  
CMD ["node", "server.js"]  
- ENTRYPOINT runs tini as the container's init process (PID 1).  
 This ensures proper signal handling and process cleanup.  
- CMD starts the Node.js application by running server.js.  
  
------------------------------------------------------------  
Summary  
------------------------------------------------------------  
- Uses Node.js 18 slim base image for efficiency.  
- Installs curl for debugging and tini for stable container behavior.  
- Installs nodemon for development hot reloading.  
- Efficiently installs dependencies using npm ci and cache cleanup.  
- Runs the application on port 80 using server.js with tini as the init process.