# Builder - Install all dependencies (including devDependencies) and build the application

FROM node:18-alpine AS builder

WORKDIR /usr/src/app

# Copy package.json and package-lock.json

# package-lock.json is crucial for npm to ensure consistent installs

COPY package.json package-lock.json ./

# Install ALL dependencies (dev included, for build tools) using npm ci

# npm ci is generally recommended for CI environments as it's faster and stricter than npm install

RUN npm ci

# Copy the rest of the application source code

COPY . .

# Run the build script defined in package.json (e.g., to compile TypeScript into ./dist)

RUN npm run build

# Production - Setup the final production image

FROM node:18-alpine AS production

WORKDIR /usr/src/app

ENV NODE\_ENV=production

# Copy package.json and package-lock.json again

COPY package.json package-lock.json ./

# Install ONLY production dependencies using npm ci

RUN npm ci --only=production --ignore-scripts

# Copy the built application artifacts from the 'builder' stage

# Ensure your build process outputs to a 'dist' folder, or adjust this path

COPY --from=builder /usr/src/app/dist ./dist

# Copy package.json (some apps might need it for runtime context, and it's small)

# This is useful if your CMD relies on scripts in package.json or if the app reads it.

COPY --from=builder /usr/src/app/package.json ./

# Expose the port the ActivePieces application runs on (default is often 3000)

# Adjust if ActivePieces uses a different port by default

EXPOSE 3000

# Command to run the application

# Ensure 'dist/main.js' is the correct entry point after your build process

CMD ["node", "dist/main.js"]