

Dockerfile Documentation (Line by Line Explanation + Notes + Caution Points)

1. Base Image

FROM node:18-alpine AS base

- Uses Node 18 Alpine, a lightweight Linux distribution.
- Smaller image size → faster download & deployment.
- Suitable for production environments.

2. Install Required System Packages

```
RUN apk add --no-cache \
    python3 \
    make \
    g++ \
    supervisor \
    wget \
    libc6-compat \
    && rm -rf /var/cache/apk/*
```

Purpose:

- python3 / make / g++ — Needed for native npm module compilation (bcrypt, sharp, prisma).
- supervisor — Runs and restarts Next.js automatically.
- wget — Used in container healthcheck.
- libc6-compat — Provides compatibility libraries for Node modules.
- --no-cache — Reduces final image size.

Common mistakes:

- Missing build tools → npm install failures.
- Unremoved tools increase final image weight.

3. Build Time Environment Variables

```
ARG SKIP_LINT_AND_TYPECHECK=1
ENV SKIP_LINT_AND_TYPECHECK=$SKIP_LINT_AND_TYPECHECK
ENV NEXT_DISABLE_ESLINT=1
ENV NEXT_DISABLE_TYPECHECK=1
ENV NEXT_TELEMETRY_DISABLED=1
ENV NODE_OPTIONS=--max-old-space-size=4096
```

Why these exist:

- Skips linting & typechecking → faster builds.
- Prevents telemetry sending.
- Prevents Node.js out of memory crashes.

Caution:

- Skipping TS checks means errors won't be detected during build.

4. Set Working Directory

`WORKDIR /app`

- All following commands execute inside /app.
- Helps organize the container filesystem.

5. Copy Application Code

`COPY ./akshayapatra .`

- Copies complete project folder into container.

Common issues:

- Wrong folder path → package.json not found.
- Ensure .dockerignore excludes node_modules and .git.

6. Install Dependencies

`RUN npm install`

- Installs all required npm modules.

Possible problems:

- Missing package-lock.json → inconsistent versions.
- Private packages require .npmrc.

7. Build Next.js Application

`RUN npm run build`

- Creates production build inside .next.
- Uses previously defined environment optimizations.

Build may fail if:

- Required environment variables are missing.
- Native modules incompatible with Alpine Linux.

8. Add Supervisor Configuration

`RUN mkdir -p /etc/supervisor/conf.d`

`COPY <<EOF /etc/supervisor/conf.d/supervisord.conf`

```
[supervisord]
nodaemon=true
logfile=/var/log/supervisor/supervisord.log

[program:nextjs]
command=npm start
directory=/app
autostart=true
autorestart=true
stdout_logfile=/var/log/supervisor/nextjs.out.log
stderr_logfile=/var/log/supervisor/nextjs.err.log
environment=NODE_ENV=production
EOF
```

Why supervisor:

- Ensures continuous uptime.
- Automatically restarts Next.js if it crashes.
- Stores stdout/stderr logs.

Note:

- PM2 not required — supervisor is lightweight and Alpine friendly.

9. Create Supervisor Log Directories

```
RUN mkdir -p /var/log/supervisor /var/run
```

- Prevents runtime errors involving missing log directories.

10. Expose Port

```
EXPOSE 3000
```

- Indicates the app listens on port 3000.
- Port mapping still requires: `docker run -p 3000:3000`.

11. Health Check

```
HEALTHCHECK --interval=30s --timeout=3s --start-period=5s --retries=3 \
  CMD wget --no-verbose --tries=1 --spider http://localhost:3000/api/health || exit 1
```

Purpose:

- Ensures container is operational.
- AWS ECS/EKS restarts unhealthy containers.

Requirement:

- App must include `/api/health` endpoint.

Mistake to avoid:

- Missing health endpoint → AWS marks container unhealthy, causing restart loops.

12. Final Command (Start Container)

CMD ["/usr/bin/supervisord", "-c", "/etc/supervisor/conf.d/supervisord.conf"]

What happens on start:

1. Supervisor launches.
2. Supervisor runs npm start.
3. App stays alive with auto restart.

Important Notes & Common Mistakes Summary

- Wrong COPY path → project not copied correctly.
- Missing .dockerignore → slower, heavier builds.
- Missing system libraries → sharp/prisma build errors.
- Incorrect healthcheck path → container marked unhealthy.
- Missing environment variables during build → build failure.
- Wrong supervisor config path → app will not start.
- Exposing wrong port → app unreachable.