

Optimized Dockerfile Documentation (Next.js Multi-Stage Build)

This document includes optimized multi-stage Dockerfiles and a comparison table to help choose the best version.

Which Version Should You Use?

Feature	No Supervisor (Recommended)	With Supervisor
Image Size	Smallest	Medium
Build Speed	Fastest	Moderate
Runtime Stability	Very Stable	Very Stable
Automatic Restart Inside Container	No	Yes
Best For	AWS ECS, EKS, App Runner, Cloud Run, Bare-metal, Docker standalone, PM2-like environments	

Recommended Optimized Multi-Stage Dockerfile (No Supervisor)

```
FROM node:18-alpine AS builder
WORKDIR /app

RUN apk add --no-cache python3 make g++ libc6-compat

COPY ./akshayapatra/package*.json ./
RUN npm install

COPY ./akshayapatra .

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

RUN npm run build

FROM node:18-alpine AS runner
WORKDIR /app

ENV NODE_ENV=production
ENV NEXT_TELEMETRY_DISABLED=1

COPY --from=builder /app/package*.json ./
COPY --from=builder /app/node_modules ./node_modules
COPY --from=builder /app/.next ./next
COPY --from=builder /app/public ./public

EXPOSE 3000

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

Optimized Multi-Stage Dockerfile With Supervisor

```
FROM node:18-alpine AS builder
WORKDIR /app
RUN apk add --no-cache python3 make g++ libc6-compat

COPY ./akshayapatra/package*.json ./
RUN npm install
COPY ./akshayapatra .

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

RUN npm run build

FROM node:18-alpine AS runner
WORKDIR /app

RUN apk add --no-cache supervisor wget libc6-compat

ENV NODE_ENV=production
ENV NEXT_TELEMETRY_DISABLED=1

COPY --from=builder /app/package*.json ./
COPY --from=builder /app/node_modules ./node_modules
COPY --from=builder /app/.next ./next
COPY --from=builder /app/public ./public

RUN mkdir -p /etc/supervisor/conf.d /var/log/supervisor /var/run

COPY <<EOF /etc/supervisor/conf.d/supervisord.conf
[supervisord]
nodaemon=true

[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

EXPOSE 3000

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

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