# node-default

```
# base image
FROM node:18.12.1

# Create and change to the app directory.
WORKDIR /usr/app

# Copy application dependency manifests to the container image.
# A wildcard is used to ensure copying both package.json AND package-lock.json (when available).
# Copying this first prevents re-running npm install on every code change.
COPY . .

# Install production dependencies.
# If you add a package-lock.json, speed your build by switching to 'npm ci'.
RUN npm ci --only=production

RUN npm run build

CMD ["npm", "start"]
```

```
❯ docker image ls
REPOSITORY   TAG            IMAGE ID       CREATED          SIZE
compare      node-default   ed5c3e445816   36 seconds ago   1.2GB
```

## node-default-slim

```
# base image
FROM node:18.12.1-slim

# Create and change to the app directory.
WORKDIR /usr/app

# Copy application dependency manifests to the container image.
# A wildcard is used to ensure copying both package.json AND package-lock.json (when available).
# Copying this first prevents re-running npm install on every code change.
COPY . .

# Install production dependencies.
# If you add a package-lock.json, speed your build by switching to 'npm ci'.
RUN npm ci --only=production

RUN npm run build

CMD ["npm", "start"]
```

```
❯ docker image ls
REPOSITORY   TAG                 IMAGE ID       CREATED         SIZE
compare      node-default-slim    c26d3356e15f   8 seconds ago   490MB
```

# node-buster

```
# base image
FROM node:buster

# Create and change to the app directory.
WORKDIR /usr/app

# Copy application dependency manifests to the container image.
# A wildcard is used to ensure copying both package.json AND package-lock.json (when available).
# Copying this first prevents re-running npm install on every code change.
COPY . .

# Install production dependencies.
# If you add a package-lock.json, speed your build by switching to 'npm ci'.
RUN npm ci --only=production

RUN npm run build

CMD ["npm", "start"]
```

```
❯ docker image ls
REPOSITORY   TAG                 IMAGE ID       CREATED         SIZE
compare      node-buster          60669e6c673d   4 seconds ago   1.17GB
```

# node-bullseye

```
# base image
FROM node:bullseye

# Create and change to the app directory.
WORKDIR /usr/app

# Copy application dependency manifests to the container image.
# A wildcard is used to ensure copying both package.json AND package-lock.json (when available).
# Copying this first prevents re-running npm install on every code change.
COPY . .

# Install production dependencies.
# If you add a package-lock.json, speed your build by switching to 'npm ci'.
RUN npm ci --only=production

RUN npm run build

CMD ["npm", "start"]
```

```
❯ docker image ls
REPOSITORY   TAG              IMAGE ID       CREATED         SIZE
compare      node-bullseye    38d4b480e0f7   4 seconds ago   1.2GB
```

# node-bullseye-slim

```
# base image
FROM node:bullseye-slim

# Create and change to the app directory.
WORKDIR /usr/app

# Copy application dependency manifests to the container image.
# A wildcard is used to ensure copying both package.json AND package-lock.json (when available).
# Copying this first prevents re-running npm install on every code change.
COPY . .

# Install production dependencies.
# If you add a package-lock.json, speed your build by switching to 'npm ci'.
RUN npm ci --only=production

RUN npm run build

CMD ["npm", "start"]
```

```
❯ docker image ls
REPOSITORY   TAG                 IMAGE ID       CREATED         SIZE
compare      node-bullseye-slim   ca20541aa6b4   4 seconds ago   493MB
```

# node-bullseye-lts

```
# base image
FROM node:lts-bullseye-slim

# Create and change to the app directory.
WORKDIR /usr/app

# Copy application dependency manifests to the container image.
# A wildcard is used to ensure copying both package.json AND package-lock.json (when available).
# Copying this first prevents re-running npm install on every code change.
COPY . .

# Install production dependencies.
# If you add a package-lock.json, speed your build by switching to 'npm ci'.
RUN npm ci --only=production

RUN npm run build

CMD ["npm", "start"]
```

```
❯ docker image ls
REPOSITORY   TAG                IMAGE ID       CREATED         SIZE
compare      node-bullseye-lts   5c5b0fe89c79   5 seconds ago   490MB
```

# node-alpine

```
# base image
FROM node:alpine

# Create and change to the app directory.
WORKDIR /usr/app

# Copy application dependency manifests to the container image.
# A wildcard is used to ensure copying both package.json AND package-lock.json (when available).
# Copying this first prevents re-running npm install on every code change.
COPY . .

# Install production dependencies.
# If you add a package-lock.json, speed your build by switching to 'npm ci'.
RUN npm ci --only=production

RUN npm run build

CMD ["npm", "start"]
```

```
❯ docker image ls
REPOSITORY   TAG                 IMAGE ID       CREATED         SIZE
compare      node-alpine          1a7aae2e02bb   7 seconds ago   424MB
```

# node-distroless

```
# Build image
FROM node:18-bullseye-slim AS build

# Create and change to the app directory.
WORKDIR /usr/src/app
COPY . /usr/src/app
RUN npm install
RUN npm run build

# Copy to distroless image
FROM gcr.io/distroless/nodejs:18
COPY --from=build /usr/src/app /usr/src/app
WORKDIR /usr/src/app
CMD ["./node_modules/next/dist/bin/next","start"]
```

```
❯ docker image ls
REPOSITORY   TAG                 IMAGE ID       CREATED              SIZE
compare      node-distroless      f8080d714410   About a minute ago   448MB
```

# node-distroless-smaller

## DockerFile

```
# Build image
FROM node:18-bullseye-slim AS build

# Create and change to the app directory.
WORKDIR /usr/src/app
COPY . /usr/src/app
RUN npm install
RUN npm run build

# Copy to distroless image
FROM gcr.io/distroless/nodejs:18
COPY --from=build /usr/src/app/.next/standalone /usr/src/app
WORKDIR /usr/src/app
CMD ["server.js"]
```

## next.config.js

```
/** @type {import('next').NextConfig} */
const nextConfig = {
 reactStrictMode: true,
 swcMinify: true,
 output: "standalone",
};

module.exports = nextConfig;
```

```
❯ docker image ls
REPOSITORY   TAG                       IMAGE ID       CREATED          SIZE
compare      node-distroless-smaller   c4c55655b844   30 seconds ago   172MB
```

# Summary

```
❯ docker image ls
REPOSITORY   TAG                          IMAGE ID       CREATED             SIZE
compare      node-distroless-smaller      c4c55655b844   37 minutes ago      172MB
compare      node-distroless              f8080d714410   40 minutes ago      448MB
compare      node-alpine                  1a7aae2e02bb   44 minutes ago      424MB
compare      node-bullseye-lts            5c5b0fe89c79   48 minutes ago      490MB
compare      node-bullseye-slim           ca20541aa6b4   49 minutes ago      493MB
compare      node-bullseye                38d4b480e0f7   51 minutes ago      1.2GB
compare      node-buster                  60669e6c673d   53 minutes ago      1.17GB
compare      node-default-slim            c26d3356e15f   55 minutes ago      490MB
compare      node-default                 ed5c3e445816   About an hour ago   1.2GB
```