

```
#
=====

==
# Dockerfile Examples for Different Puppeteer Scenarios
# Choose the one that best fits your needs
#
```

```
=====

# -----
# OPTION 1: Official Puppeteer Image (Recommended for simplicity)
# -----
FROM ghcr.io/puppeteer/puppeteer:23.0.0
```

```
ENV PUPPETEER_SKIP_CHROMIUM_DOWNLOAD=true \
    PUPPETEER_EXECUTABLE_PATH=/usr/bin/google-chrome-stable
```

```
WORKDIR /app
```

```
# Copy package files
COPY package*.json ./
```

```
# Install dependencies
RUN npm ci
```

```
# Copy application files
COPY . .
```

```
# Non-root user (already configured in base image)
USER pptruser
```

```
CMD ["node", "index.js"]
```

```
# -----
# OPTION 2: Debian-based with full control (Good for customization)
# -----
FROM node:20-bullseye
```

Install Chrome dependencies

RUN apt-get update && apt-get install -y \

ca-certificates \

fonts-liberation \

libasound2 \

libatk-bridge2.0-0 \

libatk1.0-0 \

libc6 \

libcairo2 \

libcups2 \

libdbus-1-3 \

libexpat1 \

libfontconfig1 \

libgbm1 \

libgcc1 \

libglib2.0-0 \

libgtk-3-0 \

libnspr4 \

libnss3 \

libpango-1.0-0 \

libpangocairo-1.0-0 \

libstdc++6 \

libx11-6 \

libx11-xcb1 \

libxcb1 \

libxcomposite1 \

libxcursor1 \

libxdamage1 \

libxext6 \

libxfixed3 \

libxi6 \

libxrandr2 \

libxrender1 \

libxss1 \

libxtst6 \

lsb-release \

wget \

```
xdg-utils \
libdrm2 \
libxkbcommon0 \
fonts-ipafont-gothic \
fonts-wqy-zenhei \
fonts-thai-tlwg \
fonts-kacst \
fonts-freefont-ttf \
--no-install-recommends \
&& rm -rf /var/lib/apt/lists/*
```

WORKDIR /app

Create non-root user

```
RUN groupadd -r pptruser && useradd -r -g pptruser -G audio,video pptruser \
    && mkdir -p /home/pptruser/Downloads \
    && chown -R pptruser:pptruser /home/pptruser \
    && chown -R pptruser:pptruser /app
```

Copy package files

```
COPY --chown=pptruser:pptruser package*.json ./
```

Switch to non-root user

```
USER pptruser
```

Install dependencies

```
RUN npm install
```

Copy application files

```
COPY --chown=pptruser:pptruser . .
```

```
CMD ["node", "index.js"]
```

OPTION 3: Alpine-based (Smallest image size)

FROM node:20-alpine

Install Chromium and dependencies

```
RUN apk add --no-cache \  
    chromium \  
    nss \  
    freetype \  
    harfbuzz \  
    ca-certificates \  
    ttf-freefont \  
    font-noto-emoji \  
&& rm -rf /var/cache/apk/*
```

Tell Puppeteer to skip downloading Chrome and use installed version

```
ENV PUPPETEER_SKIP_CHROMIUM_DOWNLOAD=true \  
    PUPPETEER_EXECUTABLE_PATH=/usr/bin/chromium-browser
```

WORKDIR /app

Create non-root user

```
RUN addgroup -g 1001 -S pptruser && adduser -S -u 1001 -G pptruser pptruser \  
    && mkdir -p /home/pptruser/Downloads \  
    && chown -R pptruser:pptruser /home/pptruser \  
    && chown -R pptruser:pptruser /app
```

Copy package files

```
COPY --chown=pptruser:pptruser package*.json ./
```

Switch to non-root user

```
USER pptruser
```

Install dependencies

```
RUN npm install
```

Copy application files

```
COPY --chown=pptruser:pptruser . .
```

```
CMD ["node", "index.js"]
```

```
# -----
# OPTION 4: Ubuntu-based with system Chrome (For specific Chrome version needs)
# -----

FROM ubuntu:22.04

# Avoid interactive prompts
ENV DEBIAN_FRONTEND=noninteractive

# Install Node.js and Chrome
RUN apt-get update && apt-get install -y \
    curl \
    gnupg \
    ca-certificates \
    && curl -fsSL https://deb.nodesource.com/setup_20.x | bash - \
    && apt-get install -y nodejs \
    && wget -q -O - https://dl-ssl.google.com/linux/linux_signing_key.pub | apt-key add - \
    && sh -c 'echo "deb [arch=amd64] http://dl.google.com/linux/chrome/deb/ stable main" >> /etc/
apt/sources.list.d/google.list' \
    && apt-get update \
    && apt-get install -y google-chrome-stable \
    fonts-ipafont-gothic \
    fonts-wqy-zenhei \
    fonts-thai-tlwg \
    fonts-kacst \
    fonts-freefont-ttf \
    libxss1 \
    --no-install-recommends \
    && rm -rf /var/lib/apt/lists/*

ENV PUPPETEER_SKIP_CHROMIUM_DOWNLOAD=true \
    PUPPETEER_EXECUTABLE_PATH=/usr/bin/google-chrome-stable

WORKDIR /app

# Create non-root user
RUN groupadd -r pptruser && useradd -r -g pptruser -G audio,video pptruser \
    && mkdir -p /home/pptruser/Downloads \
```

```
&& chown -R pptruser:pptruser /home/pptruser \  
&& chown -R pptruser:pptruser /app
```

```
# Copy package files
```

```
COPY --chown=pptruser:pptruser package*.json ./
```

```
USER pptruser
```

```
RUN npm install
```

```
COPY --chown=pptruser:pptruser . .
```

```
CMD ["node", "index.js"]
```

```
# -----
```

```
# OPTION 5: Multi-stage build (Optimized for production)
```

```
# -----
```

```
FROM node:20-bullseye AS builder
```

```
WORKDIR /app
```

```
# Copy package files
```

```
COPY package*.json ./
```

```
# Install ALL dependencies (including dev)
```

```
RUN npm ci
```

```
# Copy source
```

```
COPY . .
```

```
# If you have a build step (TypeScript, etc)
```

```
# RUN npm run build
```

```
# Production stage
```

```
FROM node:20-bullseye-slim
```

```
# Install Chrome dependencies
```

```
RUN apt-get update && apt-get install -y \  
  ca-certificates \  
  fonts-liberation \  
  libatk-bridge2.0-0 \  
  libatk1.0-0 \  
  libcups2 \  
  libdbus-1-3 \  
  libgbm1 \  
  libglib2.0-0 \  
  libgtk-3-0 \  
  libnspr4 \  
  libnss3 \  
  libxcomposite1 \  
  libxdamage1 \  
  libxrandr2 \  
  xdg-utils \  
  wget \  
  --no-install-recommends \  
&& rm -rf /var/lib/apt/lists/*
```

WORKDIR /app

Create non-root user

```
RUN groupadd -r pptruser && useradd -r -g pptruser -G audio,video pptruser \  
  && mkdir -p /home/pptruser/Downloads \  
  && chown -R pptruser:pptruser /home/pptruser \  
  && chown -R pptruser:pptruser /app
```

Copy from builder

```
COPY --from=builder --chown=pptruser:pptruser /app/node_modules ./node_modules  
COPY --chown=pptruser:pptruser package*.json ./  
COPY --chown=pptruser:pptruser . .
```

USER pptruser

CMD ["node", "index.js"]

```
# -----  
# OPTION 6: With docker-compose for development  
# Save this as docker-compose.yml  
# -----  
# version: '3.8'  
#  
# services:  
#   puppeteer-app:  
#     build: .  
#     volumes:  
#       - ./app  
#       - /app/node_modules  
#     environment:  
#       - NODE_ENV=development  
#     shm_size: '2gb' # Important: Increase shared memory  
#     security_opt:  
#       - seccomp:unconfined # Only for development  
#     cap_add:  
#       - SYS_ADMIN # Only if absolutely necessary  
#     ports:  
#       - "3000:3000"
```