

Dockerfile from node 16.19.1

1-dockerfile :

A-description :

use an officiel node as a parent image to :

- set working directory
- install nginx
- Remove default Nginx configuration
- copy the application code inside container
- Copy package.json
- Install app dependencies
- copy nginx cofiguration file
- change the owner of the var/www/html directory to node

B-code:

```
FROM node:16.19.1
```

```
USER root
```

```
# Create a directory for the app
```

```
WORKDIR /var/www/html
```

```
# Install Nginx
```

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

```
# Remove default Nginx configuration
```

```
RUN rm /etc/nginx/sites-enabled/default
```

```
# Copy app file
```

```
ADD ./ /var/www/html
```

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

```
COPY ./package*.json /var/www/html
```

```
RUN rm -rf node_modules
```

```
# Install app dependencies
```

```
RUN npm install
```

```
RUN chmod -R 777 /var/www/html
```

```
# Copy custom Nginx configuration
```

COPY ./nginx.conf /etc/nginx/conf.d

RUN chown -R node:node /var/www/html
EXPOSE 80

Start Nginx and the app
CMD service nginx start && npm start

2-nginx configuration file :

A-Description :

- **nginx** : is open-source web server software :
performance and scalability
can be configured as reverse proxy server , A load balancer , API gateway , firewall for web applications ,
- **This is an Nginx configuration file** to listen on port 80:
must listen for HTTP connections. and responds the requests by server name: localhost
- NGINX's location setting help for NGINX responds to requests for resources inside the server

B-code :

```
server {  
    listen 80;  
    server_name localhost;  
  
    root /var/www/html/out;  
    index index.html index.htm;  
    try_files $uri $uri/ /index.html;  
  
    location /_next/static {  
        alias /var/www/html/out/_next/static;  
        add_header Cache-Control "public, max-age=3600, immutable";  
    }  
  
    location / {  
        # reverse proxy for next server  
        proxy_pass http://localhost:3000;  
        proxy_http_version 1.1;  
        proxy_set_header Upgrade $http_upgrade;  
        proxy_set_header Connection 'upgrade';  
        proxy_set_header Host $host;  
        proxy_cache_bypass $http_upgrade;  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        proxy_set_header X-Forwarded-Proto $scheme;  
    }  
}
```

```
location ~ /.well-known {  
    allow all;  
}  
}
```