

Prerequisites

- Install Docker & Docker compose.

Directory & File

```
tiotobing18@narasi-fe:~/frontend$ tree
.
├── Dockerfile
├── docker-compose.yml
├── nginx
│   ├── Dockerfile
│   └── default.conf
├── package.json
├── process.yml
└── server.js
```

Step

- Buat sebuah Proyek Node sederhana yang akan di dockerize. (Referensi: <https://nodejs.org/de/docs/guides/nodejs-docker-webapp/>)
- Buat file bernama server.js dan package.json

server.js

```
'use strict';

const express = require('express');

// Constants
const PORT = 3000;
const HOST = '0.0.0.0';

// App
const app = express();
app.get('/', (req, res) => {
  res.send('Hello world\n Web-App-02');
});

app.listen(PORT, HOST);
console.log(`Running on http://${HOST}:${PORT}`);
```

package.json

```
{
  "name": "frontend_app",
  "version": "1.0.0",
  "description": "Node.js on Docker",
  "author": "tiotobing18",
  "main": "server.js",
  "scripts": {
    "start": "node server.js"
  },
  "dependencies": {
    "express": "^4.16.1"
  }
}
```

- Untuk Docker, buat sebuah file bernama Dockerfile dan docker-compose.yml

Dockerfile

```
FROM node:10

# Set working directory
WORKDIR /usr/app

# Install PM2 globally
RUN npm install --global pm2

# Monitoring PM2 Plus – Real Time
ENV PM2_PUBLIC_KEY avckauyyzg3cbpm
ENV PM2_SECRET_KEY 118s40nr4280ouq

# Copy "package.json" and "package-lock.json" before other files
# Utilise Docker cache to save re-installing dependencies if unchanged
COPY ./package*.json ./

# Install dependencies
RUN npm install

# Copy all files
COPY . /

# Build app
# RUN npm run build

# Expose the listening port
EXPOSE 3000

# Launch app with PM2
CMD ["pm2-runtime", "start", "process.yml"]
```

- docker-compose.yml

```
version: '3'
services:
  nodejs:
    build: ./
    image: tiotobing/nodejs    #For naming docker image
    container_name: apps      #For naming docker container
  nginx:
    build: ./nginx
    image: tiotobing/nginx
    container_name: nginx
  ports:
    - 80:80
```

- NGINX config. Untuk konfigurasi nginx, buat sebuah folder bernama nginx kemudian didalamnya tambahkan 2 file bernama Dockerfile & default.conf.

Dockerfile (on /nginx)

```
FROM nginx:alpine

# Remove any existing config files
RUN rm /etc/nginx/conf.d/*

# Copy config files
# *.conf files in "conf.d/" dir get included in main config
COPY ./default.conf /etc/nginx/conf.d/

# Expose the listening port
EXPOSE 80

# Launch NGINX
CMD [ "nginx", "-g", "daemon off;" ]
```

default.conf

```
upstream nodejs {
    server nodejs:3000;
}

server {
    listen 80 default_server;
    server_name _;    # isi dengan domain/IP
    server_tokens off;

    gzip on;
    gzip_proxied any;
    gzip_comp_level 4;
    gzip_types text/css application/javascript image/svg+xml;

    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;

    # DYNAMIC ASSETS - NO CACHE
    location / {
        proxy_pass http://nodejs;
    }
}
```

- Tambahkan sebuah file konfigurasi PM2 bernama process.yml.

process.yml

```
apps:
- script : ./server.js
  name : 'frontend'  #your app_name
```

- Run Docker Compose

```
Docker-compose up --build
```

- Result

```
Starting nginx ... done
Starting apps ... done
Attaching to apps, nginx
apps | 2019-10-08T03:50:14: PM2 log: Launching in no daemon mode
apps | 2019-10-08T03:50:14: PM2 log: [PM2 I/O] Using: Public key: avckauyyzg3cbpm | Private key: 118s40nr4280ouq | Machine name: c65efd99ec2f
apps | 2019-10-08T03:50:14: PM2 log: App [frontend:0] starting in -fork mode-
apps | 2019-10-08T03:50:14: PM2 log: App [frontend:0] online
apps | Running on http://0.0.0.0:3000
```

docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
c65efd99ec2f	tiotobing/nodejs	"docker-entrypoint.s..."	17 hours ago	Up 5 minutes	3000/tcp	apps
74fcc605dc41	tiotobing/nginx	"nginx -g 'daemon of..."	17 hours ago	Up 5 minutes	0.0.0.0:80->80/tcp	nginx

Test web app & PM2 Plus di browser

