

Troubleshooting Neo Feeder versi Docker

Update file docker-compose.yml dan Patch Nginx Error 502 504 Bad Gateway

A. Update file konfigurasi docker-compose.yml

1. Pastikan console linux sudah pada lokasi instalasi Neo Feeder

```
#ls -lah
total 40K
drwxrwxr-x 7 neofeeder neofeeder 4.0K Mar 17 18:20 .
drwxr-xr-x 11 neofeeder neofeeder 4.0K Mar 17 18:48 ..
-rwxrwxr-x 1 neofeeder neofeeder 698 Mar 15 17:04 Dockerfile
drwxrwxr-x 4 neofeeder neofeeder 4.0K Mar 17 18:09 app
-rwxrwxr-x 1 neofeeder neofeeder 1.7K Mar 15 17:04 carapenggunaan.txt
-rwxrwxr-x 1 neofeeder neofeeder 1.1K Mar 17 18:20 docker-compose.yml
drwxrwxr-x 3 neofeeder neofeeder 4.0K Mar 15 17:04 etc
drwxrwxr-x 3 neofeeder neofeeder 4.0K Mar 17 18:54 nginx
drwxrwxr-x 3 neofeeder neofeeder 4.0K Mar 15 17:04 pgsql
drwxrwxr-x 2 neofeeder neofeeder 4.0K Mar 17 18:09 prefill_pddikti
```

2. Matikan service docker, dengan perintah berikut:

```
#docker-compose down
```

3. Copy/Replace/Edit file docker-compose.yml di folder instalasi Neo Feeder dengan file docker-compose.yml yang sudah disediakan
4. Lalu buat folder dan file access.log dan error.log dengan perintah berikut:

```
#mkdir -p nginx/logs          (perintah untuk membuat folder logs)
#touch nginx/logs/access.log  (perintah untuk membuat file access.log)
#touch nginx/logs/error.log   (perintah untuk membuat file error.log)
```

5. Nyalakan service docker Neo Feeder

```
#docker-compose up -d
```

6. Pastikan service docker menyala

```
#docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
PORTS
e98dcd11d320  pddikti/new_feeder  "/usr/bin/supervisord"  About an hour ago  Up
About an hour  0.0.0.0:3003->3003/tcp, :::3003->3003/tcp, 0.0.0.0:8100-
>8100/tcp, :::8100->8100/tcp  app-pddikti
c2e0f3342ccb  postgres:12      "docker-entrypoint.s..."  About an hour ago  Up
About an hour  5432/tcp, 0.0.0.0:54333->54333/tcp, :::54333->54333/tcp
db-pddikti
```

B. Update/Patch/Replace file service nginx.conf

1. Pastikan console linux sudah pada lokasi instalasi Neo Feeder

```
#ls -lah
total 40K
drwxrwxr-x 7 neofeeder neofeeder 4.0K Mar 17 18:20 .
drwxr-xr-x 11 neofeeder neofeeder 4.0K Mar 17 18:48 ..
-rwxrwxr-x 1 neofeeder neofeeder 698 Mar 15 17:04 Dockerfile
drwxrwxr-x 4 neofeeder neofeeder 4.0K Mar 17 18:09 app
-rwxrwxr-x 1 neofeeder neofeeder 1.7K Mar 15 17:04 carapenggunaan.txt
-rwxrwxr-x 1 neofeeder neofeeder 1.1K Mar 17 18:20 docker-compose.yml
drwxrwxr-x 3 neofeeder neofeeder 4.0K Mar 15 17:04 etc
drwxrwxr-x 3 neofeeder neofeeder 4.0K Mar 17 18:54 nginx
drwxrwxr-x 3 neofeeder neofeeder 4.0K Mar 15 17:04 pgsq1
drwxrwxr-x 2 neofeeder neofeeder 4.0K Mar 17 18:09 prefill_pddikti
```

2. Matikan service docker, dengan perintah berikut:

```
#docker-compose down
```

3. Copy file nginx.conf ke folder instalasi Neo Feeder
4. Replace file nginx.conf pada folder installer Neo Feeder ./nginx/nginx.conf dengan file nginx.conf yang sudah disediakan
5. Nyalakan service docker Neo Feeder

```
#docker-compose up -d
```

6. Pastikan service docker menyala

```
#docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
PORTS
e98dcd11d320  pddikti/new_feeder  "/usr/bin/supervisord"  About an hour ago  Up
About an hour  0.0.0.0:3003->3003/tcp, :::3003->3003/tcp, 0.0.0.0:8100-
>8100/tcp, :::8100->8100/tcp  app-pddikti
c2e0f3342ccb  postgres:12     "docker-entrypoint.s..."  About an hour ago  Up
About an hour  5432/tcp, 0.0.0.0:54333->54333/tcp, :::54333->54333/tcp
db-pddikti
```

Isi file docker-compose.yml

```
version: '3'
services:
  #Web Service
  app-pddikti:
    build:
      context: .
      dockerfile: Dockerfile
    image: pddikti/new_feeder
    container_name: app-pddikti
    restart: unless-stopped
    tty: true
    environment:
      TZ: "Asia/Jakarta"
      SERVICE_NAME: "app"
      SERVICE_TAGS: "1.0.3"
      PGHOST: "localhost"
      PGPORT: 54333
    volumes:
      - './app:/app'
      - './prefill_pddikti:/prefill_pddikti'
      - './nginx/nginx.conf:/etc/nginx/nginx.conf'
      - './nginx/logs/access.log:/var/log/nginx/access.log'
      - './nginx/logs/error.log:/var/log/nginx/error.log'
    networks:
      - app-network
    ports:
      - "8100:8100"
      - "3003:3003"
    depends_on:
      - db-pddikti

  #PgSQL Service
  db-pddikti:
    image: postgres:12
    container_name: db-pddikti
    restart: unless-stopped
    tty: true
    ports:
      - "54333:54333"
    volumes:
      - './pgsql/data:/var/lib/postgresql/data'
    networks:
      - app-network

#Docker Networks
networks:
  app-network:
    driver: bridge
    ipam:
      driver: default
      config:
        - subnet: 192.167.150.0/24
```

Isi file config nginx.conf

```
events {
    worker_connections 1024;
    multi_accept on;
}

http {
    keepalive_timeout 65;
    types_hash_max_size 2048;
    include mime.types;
    default_type application/octet-stream;

    server {
        listen 8100;
        server_name localhost;

        location ~ /\.css {
            add_header Content-Type text/css;
        }

        location ~ /\.js {
            add_header Content-Type application/x-javascript;
        }

        location /ws/ {
            proxy_buffering on;
            proxy_buffer_size 1M;
            proxy_busy_buffers_size 1M;
            proxy_buffers 64 1M;
            proxy_connect_timeout 60000;
            proxy_send_timeout 60000;
            proxy_read_timeout 60000;
            send_timeout 60000;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-NginX-Proxy true;
            proxy_pass http://127.0.0.1:3003;
            proxy_ssl_session_reuse off;
            proxy_set_header Host $http_host;
            proxy_cache_bypass $http_upgrade;
            proxy_redirect off;
            add_header 'Access-Control-Allow-Origin' '*';
            add_header 'Access-Control-Allow-Credentials' 'true';
            add_header 'Access-Control-Allow-Headers' 'Authorization,Accept,Origin,DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Content-Range,Range';
            add_header 'Access-Control-Allow-Methods' 'GET,POST,OPTIONS,PUT,DELETE,PATCH';
        }

        access_log /var/log/nginx/access.log;
        error_log /var/log/nginx/error.log;

        index index.html;

        charset utf-8;

        root "/app/public";
    }
}
```