

# Application Deployment menggunakan Docker, Nginx sebagai Load Balancer dan MariaDB sebagai Database



docker **NGINX**



Kelompok :

Suhindra	165410166
Rochmad Adhi Y	165410167
Hidayatullah	165410160

Oposite

## Overview

Disini kami telah menggunakan Docker untuk menyebarkan aplikasi, MySQL sebagai database dan Nginx sebagai load balancer. Dua kontainer Docker akan berkomunikasi dengan satu kontainer MySQL dengan bantuan bantuan Nginx. Kami telah menyiapkan setup lengkap di repositori github kami [https://github.com/suhindra/php\\_mariadb\\_lb.git](https://github.com/suhindra/php_mariadb_lb.git)

Silakan lihat repositori yang disebutkan di atas. Ada Dockerfile untuk aplikasi, database dan load balancer.

## Teknologi dan Tools yang digunakan:

1. Php-apache
2. MariaDB
3. Docker containerization
4. Docker Compose
5. Nginx sebagai Load Balancer.

## Komponen:

1. Dockerfile: Dockerfile untuk konfigurasi aplikasi php, MySQL dan Nginx dengan semua dependensinya.

- [https://hub.docker.com/r/suhindra/php\\_mysql/](https://hub.docker.com/r/suhindra/php_mysql/)

```
1 FROM mariadb
2
3 ADD tcc.sql /docker-entrypoint-initdb.d
```

- [https://hub.docker.com/r/suhindra/php\\_app/](https://hub.docker.com/r/suhindra/php_app/)

```
1 FROM tutum/apache-php
2
3 MAINTAINER Suhindra <suhindrasuhindra@gmail.com>
4
5 RUN apt-get update && apt-get install -yq git && rm -rf /var/lib/apt/lists/*
6
7 RUN rm -fr /app
8
9 ADD . /app
```

- [https://hub.docker.com/r/suhindra/nginx\\_php/](https://hub.docker.com/r/suhindra/nginx_php/)

```

1 FROM nginx
2
3 MAINTAINER suhindra <suhindrasuhindra@gmail.com>
4
5 RUN rm -f /etc/nginx/nginx.conf
6
7 ADD nginx.conf /etc/nginx/nginx.conf
8
9 RUN service nginx restart
10
11 EXPOSE 80

```

2. Docker-compose.yml: Digunakan untuk memulai beberapa containers sekaligus dengan satu perintah sederhana.

```

1  version: '2'
2  services:
3
4      myadmin:
5          image: phpmyadmin/phpmyadmin
6          container_name: phpmyadmin
7          ports:
8              - "8080:80"
9          environment:
10             - PMA_ARBITRARY=1
11             - PMA_HOST=mysql
12          restart: always
13          depends_on:
14             - db
15
16      db:
17          image: mariadb
18          container_name: mysql
19          restart: always
20          environment:
21             - MYSQL_DATABASE=tcc
22             - MYSQL_ROOT_PASSWORD=root
23             - MYSQL_USER=admin
24             - MYSQL_PASSWORD=admin
25          ports:
26             - "3306:3306"
27          volumes:
28             - "./data/db/mysql:/var/lib/mysql"
29
30      app_server1:
31          image: suhindra/php_app
32          ports:
33             - 8081:80
34          depends_on:
35             - db
36
37      app_server2:
38          image: suhindra/php_app
39          ports:
40             - 8082:80
41          depends_on:
42             - db
43
44      nginx:
45          image: suhindra/nginx_php
46          ports:
47             - 85:80
48          depends_on:
49             - db
50             - app_server1
51             - app_server2

```

3. Folder: Berisi semua file aplikasi dan file konfigurasi.

## Deployment:

1. Aplikasi php di deploy di atas Docker Container.
2. Image Docker dibuat dengan semua paket dan dependensinya.
3. Image itu tersedia di Docker Hub.
4. Image itu mampu dijalankan / ditempatkan di platform Linux manapun.
5. Dua contoh aplikasi php sedang berjalan, yaitu dua Docker Containers yang berjalan pada mesin dengan port yang berbeda.
6. Hanya satu MariaDB instance yang berjalan.
7. Nginx digunakan sebagai Load Balancer.
8. Nginx juga berjalan di Docker Container.
9. Nginx berhubungan dengan apache instance misal aplikasi instance dan semua permintaan diteruskan / dibagi ke apache via nginx load balancer.
10. Nginx akan mendistribusikan beban antara dua apache instance karena kedua instance tersebut akan berkomunikasi dengan MySQL Container.

Dengan cara ini kita bisa mencapai load balancing antara dua instance apache dan mysql.

## Konfigurasi:

1. Dockerfile: Silakan lihat Dockerfile di semua folder, berisi semua konfigurasi dan dependensi yang dibutuhkan oleh aplikasi.
2. Nginx.conf: Path file Nginx.conf adalah /etc/nginx/nginx.conf, dimana semua konfigurasi dilakukan.
3. Disediakan blok "server" dan "upstream" di bawah blok "http".
4. Di bawah server: berikan semua rincian endpoint nginx.
5. Di bawah upstream: berikan semua rincian server apache.
6. Silakan lihat perubahan yang dilakukan pada: folder nginx / nginx.conf

## Langkah untuk menerapkan proyek ini:

1. Clone repo ke mesin Anda
2. Pindah ke folder dimana file docker-compose.yml berada.
3. Jalankan perintah: `$docker-compose -up -d`
4. Yang akan menjalankan semua kontainer secara berurutan seperti yang di tuliskan dalam file docker-compose.yml.
5. Jika image tidak ada di local mesin, docker akan mencoba menarik(pull) image dari public docker hub, setelah gambar berhasil pull, image akan berjalan sebagai Docker Container.
6. Setelah eksekusi berhasil, periksa status semua Container yang sedang berjalan dengan perintah: `$ docker id`
7. Ini akan menunjukkan empat kontainer yang ada akan rinciannya.
8. Aplikasi Php di deploy di atas dua kontainer dengan port: 8081, 8082
9. Port MariaDB diteruskan ke 3306
10. Port Nginx diteruskan ke 85
11. Jadi sesuai perincian yang disebutkan di atas,
12. Kita dapat mengakses UI aplikasi di: `http://localhost:8081` atau `http://"machine_ip":8081` dan `http://localhost:8082` atau `http://"machine_ip":8082`
13. Kita bisa mengakses UI Nginx di: `http://localhost:85` atau `http://"machine_ip":85`
14. Di sini kita bisa mengecek dengan menghentikan satu instance aplikasi kita dan request tetap diproses oleh instance lainnya.