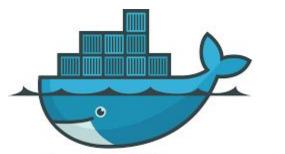
Application Deployment menggunakan Docker, Nginx sebagai Load Balancer dan MariaDB sebagai Database



# dockerNGINX



## 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 <a href="https://github.com/suhindra/php\_mariadb\_lb.git">https://github.com/suhindra/php\_mariadb\_lb.git</a>

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/

```
FROM mariadb

ADD tcc.sql /docker-entrypoint-initdb.d
```

https://hub.docker.com/r/suhindra/php/app/

```
FROM tutum/apache-php

MAINTAINER Suhindra <suhindrasuhindra@gmail.com>

RUN apt-get update && apt-get install -yq git && rm -rf /var/lib/apt/lists/*

RUN rm -fr /app

ADD . /app
```

https://hub.docker.com/r/suhindra/nginx\_php/

```
FROM nginx

MAINTAINER suhindra <suhindrasuhindra@gmail.com>

RUN rm -f /etc/nginx/nginx.conf

ADD nginx.conf /etc/nginx/nginx.conf

RUN service nginx restart

EXPOSE 80
```

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

```
version:
services:
                      myadmin:
   image: phpmyadmin/phpmyadmin
   container_name: phpmyadmin
   ports:
        - "8080:80"
   environment:
        - PMA_HOST=mysql
   restart: always
   depends_on:
        - db
image: mariadb
container name: mysql
restart: always
environment:
                                   environment:
    - MYSQL_DATABASE=tcc
    - MYSQL_ROOT_PASSWORD=root
    - MYSQL_USER=admin
    - MYSQL_PASSWORD=admin
ports:
    - "3306:3306"
volumes:
    - "./data/db/mysql:/var/lib/mysql"
                         app_server1:
    image: suhindra/php_app
    ports:
                                    - 8081:80
depends_on:
- db
                         nginx:
image: suhindra/nginx_php
ports:
                                               - 85:80
nds_on:
- db
                                                 - app_server1
- 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: htttp://localhost:8081 atau http://"machine\_ip":8081 dan htttp://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.