

BUILDING IMAGE WITH DOCKERFILE

1. buat folder ubuntu kemudian isi dengan Dockerfile berikut

```
padli@mrX: ~/Docker/ubuntu
padli@mrX: ~/Docker/ubuntu
padli@mrX:~/Docker/ubuntu$ more Dockerfile
FROM ubuntu:latest

RUN apt-get update && \
    apt-get install -y apache2 && \
    /etc/init.d/apache2 start

EXPOSE 80
padli@mrX:~/Docker/ubuntu$
```

2. lalu build image dengan nama tes

```
padli@mrX:~/Docker/ubuntu$ docker build -t tes .
Sending build context to Docker daemon 2.048kB
Step 1/3 : FROM ubuntu:latest
----> c9d990395902
Step 2/3 : RUN apt-get update && apt-get install -y apache2 && /etc/init.d/apache2 start
----> Running in c7194c1102c8
Get:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [107 kB]
Get:2 http://archive.ubuntu.com/ubuntu xenial InRelease [247 kB]
Get:3 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu xenial-security/universe Sources [79.5 kB]
Get:5 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [612 kB]
Get:6 http://security.ubuntu.com/ubuntu xenial-security/restricted amd64 Packages [12.7 kB]
Get:7 http://archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:8 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 Packages [435 kB]
Get:9 http://archive.ubuntu.com/ubuntu xenial/universe Sources [9802 kB]
Get:10 http://archive.ubuntu.com/ubuntu xenial/main amd64 Packages [1558 kB]
Get:11 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 Packages [3489 B]
Get:12 http://archive.ubuntu.com/ubuntu xenial/restricted amd64 Packages [14.1 kB]
Get:13 http://archive.ubuntu.com/ubuntu xenial/universe amd64 Packages [9827 kB]
Get:14 http://archive.ubuntu.com/ubuntu xenial/multiverse amd64 Packages [176 kB]
Get:15 http://archive.ubuntu.com/ubuntu xenial-updates/universe Sources [254 kB]
Get:16 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [990 kB]
Get:17 http://archive.ubuntu.com/ubuntu xenial-updates/restricted amd64 Packages [13.1 kB]
Get:18 http://archive.ubuntu.com/ubuntu xenial-updates/universe amd64 Packages [803 kB]
```

```
debconf: (TERM is not set, so the dialog frontend is not usable.)
debconf: falling back to frontend: Readline
Processing triggers for libc-bin (2.23-0ubuntu10) ...
Processing triggers for systemd (229-4ubuntu21.2) ...
Processing triggers for sgml-base (1.26+nmu4ubuntu1) ...
* Starting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the '
ServerName' directive globally to suppress this message
*
Removing intermediate container c7194c1102c8
----> a5d6a2442d37
Step 3/3 : EXPOSE 80
----> Running in 4e04a27a4523
Removing intermediate container 4e04a27a4523
----> 77c0038beee3
Successfully built 77c0038beee3
Successfully tagged tes:latest
padli@mrX:~/Docker/ubuntu$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
tes                  latest             77c0038beee3       7 seconds ago      257MB
julian23/ubuntu-apache2-ssh-net-tools latest             7a60b267f465       3 days ago         301MB
ubuntu              latest             c9d990395902       2 weeks ago        113MB
hello-world         latest             e38bc07ac18e       2 weeks ago        1.85kB
padli@mrX:~/Docker/ubuntu$
```

3. jalankan container dengan image tes buka port 80 host : ke 80 container

```
padli@mrX: ~/Docker/ubuntu
padli@mrX:~/Docker/ubuntu$ sudo docker run -d --name my-2nd-container -p 80:80 tes
a08644780adcd44f74daed01bdf9c040f4e6a28a985422a8411677961c83a911
padli@mrX:~/Docker/ubuntu$
```

4. cek web server apache2 di lokal:80

```
padli@mrX:~/Docker/ubuntu$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS               NAMES
44e1a1ca7a74   tes      "/bin/bash"             About a minute Up About a minute 0.0.0.0:80->80/tcp  my-2nd-container

padli@mrX:~/Docker/ubuntu$ curl http://localhost:80
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<!--
  Modified from the Debian original for Ubuntu
  Last updated: 2014-03-19
  See: https://launchpad.net/bugs/1288690
-->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Apache2 Ubuntu Default Page: It works</title>
<style type="text/css" media="screen">
* {
  margin: 0px 0px 0px 0px;
  padding: 0px 0px 0px 0px;
}

body, html {
  padding: 3px 3px 3px 3px;
  background-color: #D8DBE2;

  font-family: Verdana, sans-serif;
  font-size: 11pt;
  text-align: center;
}

div.main_page {
  position: relative;
  display: table;

  width: 800px;

  margin-bottom: 3px;
  margin-left: auto;
  margin-right: auto;
  padding: 0px 0px 0px 0px;
}
```

5. berikut contoh Dockerfile untuk membangun image berisi node.js siap pakai

Dockerizing Node.js applications | Katacoda - Mozilla Firefox

https://katacoda.com/courses/docker/3

Example: Build & Launch

The command to build the image is

```
docker build -t my-nodejs-app .
```

The command to launch the built image is

```
docker run -d --name my-running-app -p 3000:3000 my-nodejs-app
```

Testing Container

You can test the container is accessible using curl. If the application responds then you know that everything has correctly started.

```
curl http://docker:3000
```

Dockerfile

```
1 # This is your Editor pane. Write the Dockerfile here and
2 # use the command line to execute commands
3 FROM node:7-alpine
4
5 RUN mkdir -p /src/app
6
7 WORKDIR /src/app
8
9 COPY package.json /src/app/package.json
10
11 RUN npm install
12
13 COPY . /src/app
14
15 EXPOSE 3000
16
17 CMD [ "npm", "start" ]
18
```

Terminal

```
Removing intermediate container 2d3299fc01a9
--> 8aa698e812ce
Successfully built 8aa698e812ce
Successfully tagged my-nodejs-app:latest

> docker run -d --name my-running-app -p 3000:3000 my-nodejs-app
1c06e7d9faa28f5743985e02ed1633e3fb6ecede90f592eb4cbcffbeba62a9ee

> curl http://docker:3000
<!DOCTYPE html><html><head><title>Express</title><link rel="stylesheet" href="/stylesheets/st
yle.css"></head><body><h1>Express</h1><p>Welcome to Express</p></body></html>
>
```