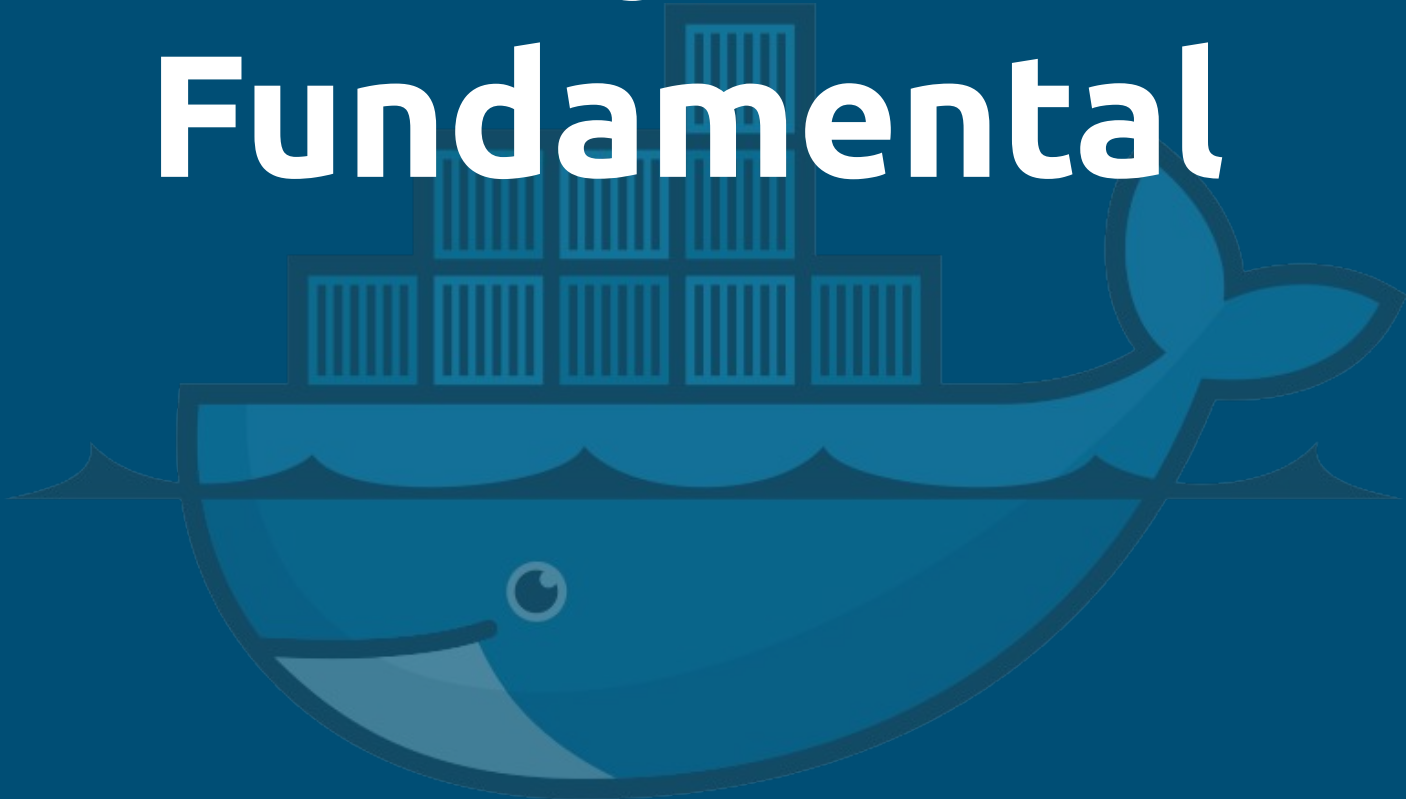




Welcome to Cangkrük'an April 2019



Lokakarya Docker Fundamental



Special Thanks to

 **CLOUDKILAT**



Me

Fadhil Yori Hibatullah

Semester 6 @ D3 IT **PENS** 2016

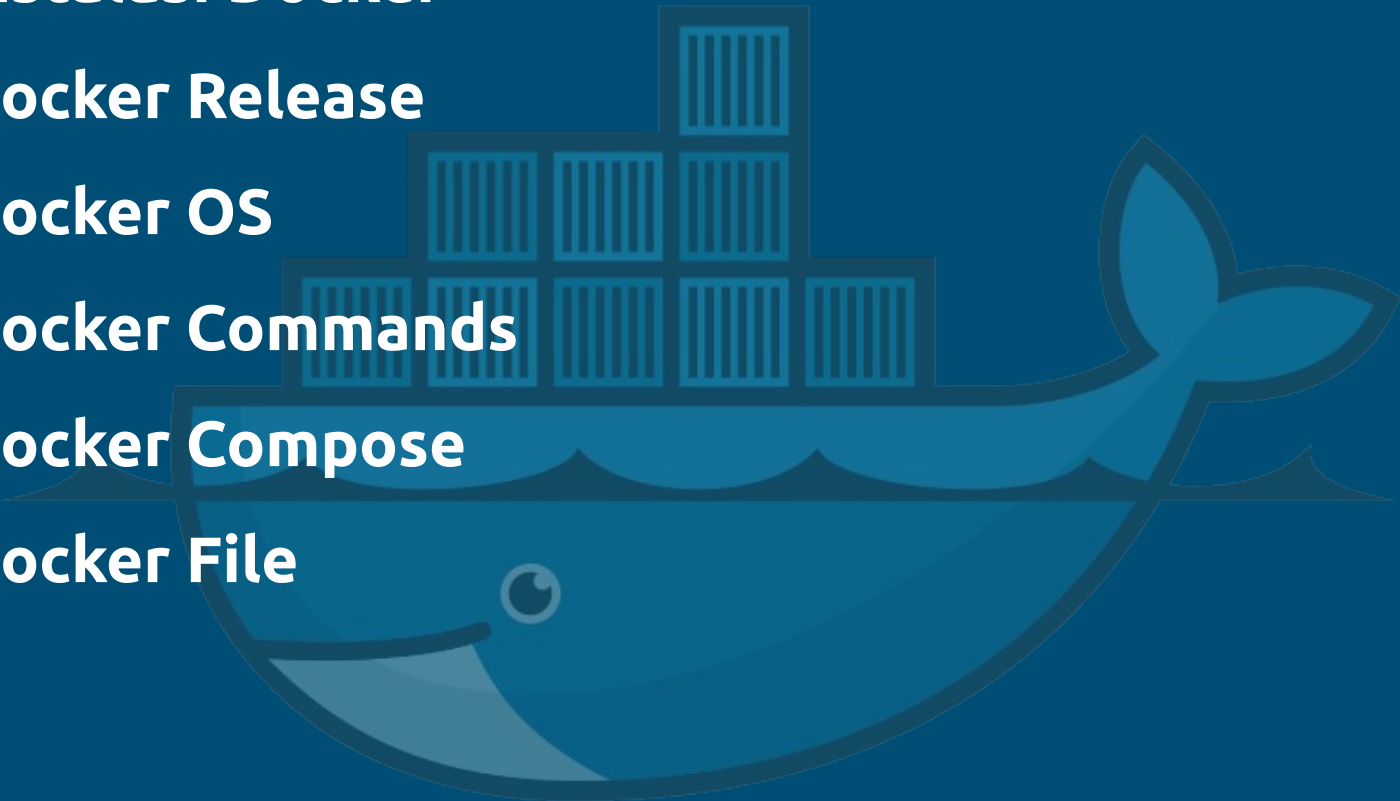
- Software Developer
- Open Source enthusiast
- Pengurus Aktif Kelompok Linux Arek Suroboyo
- Lulusan NolSatu Angkatan 08



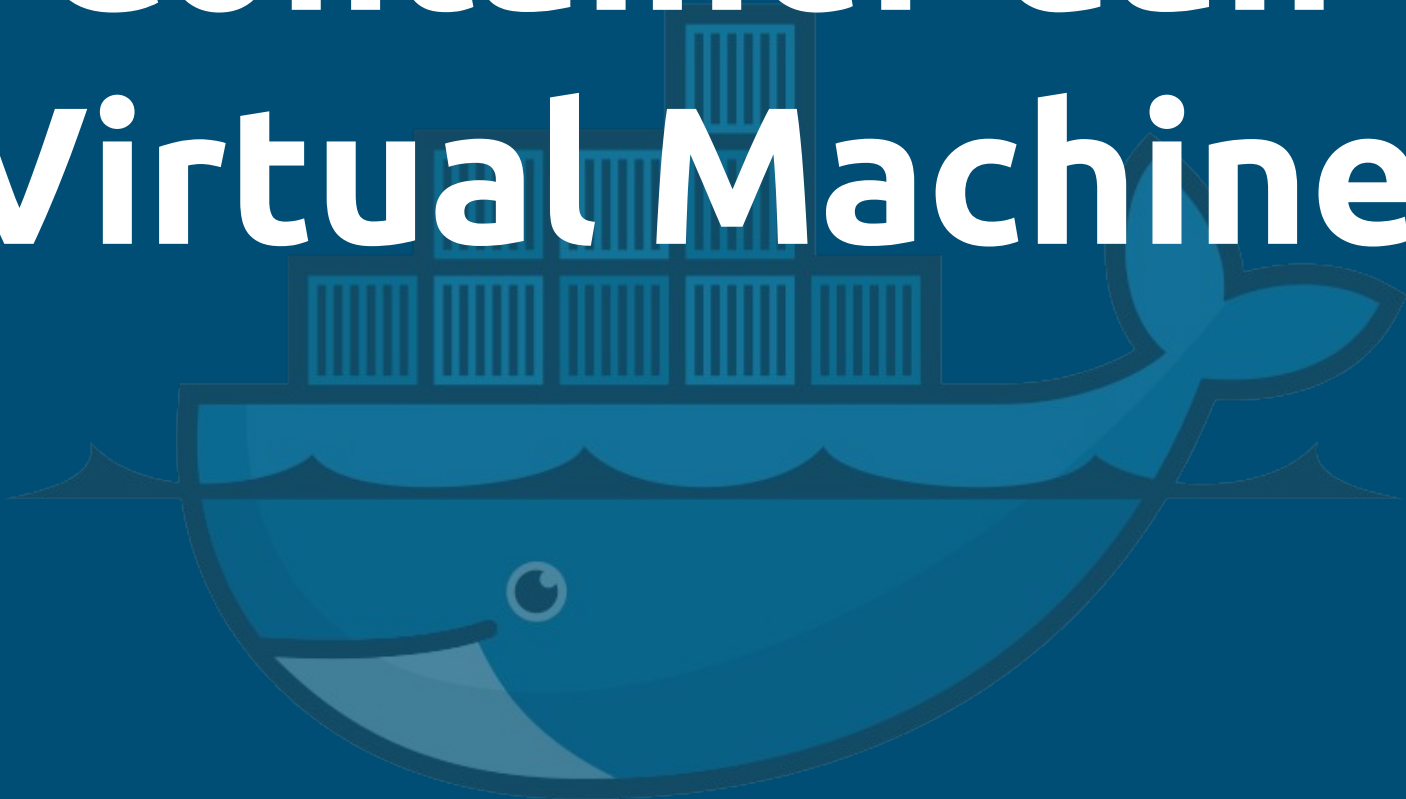
Contact :

Telegram : @fadhilyori
Facebook : <https://www.facebook.com/fadhilyori.h>
Surel : fadhil4812@gmail.com

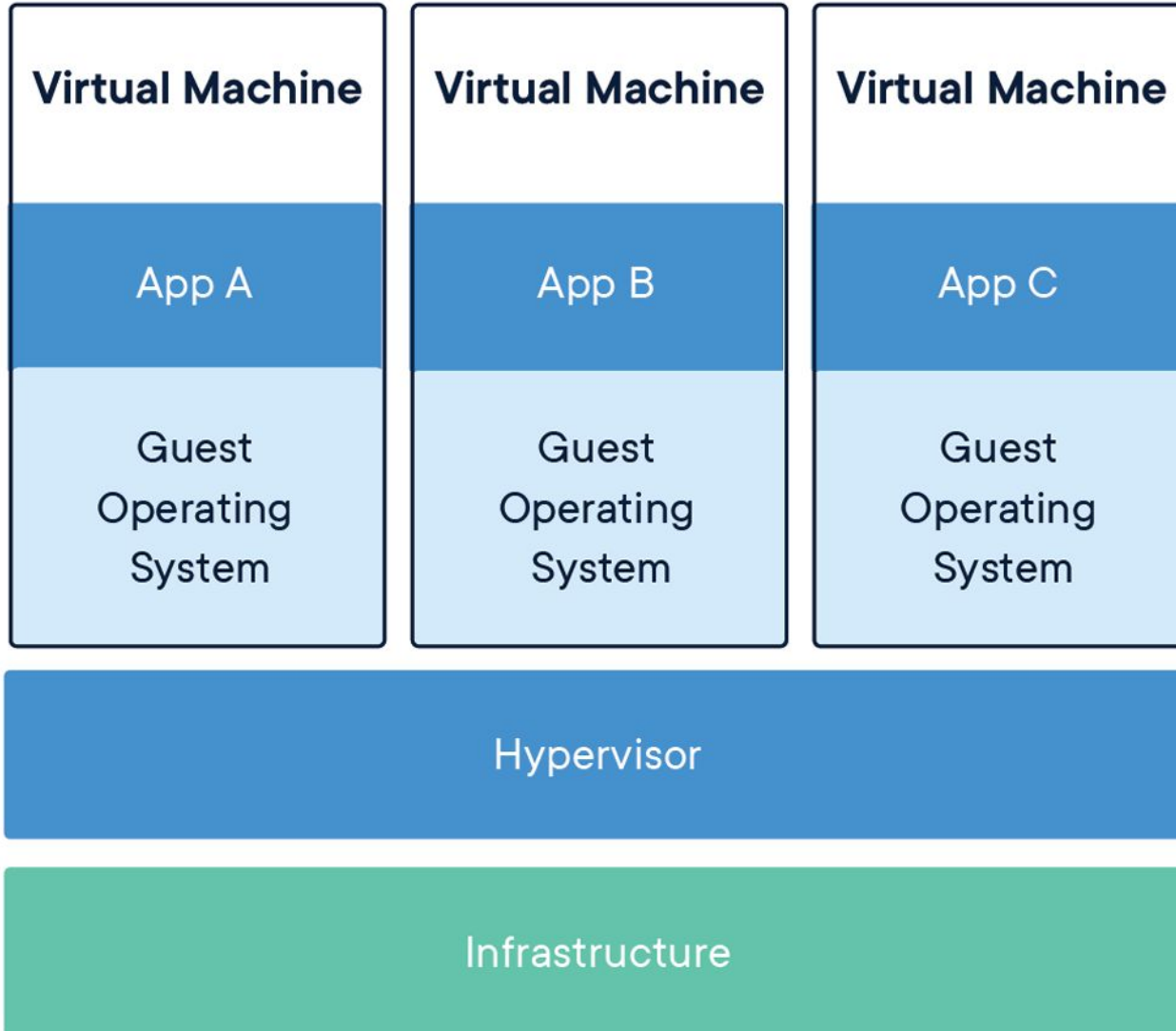
- **Container dan Virtual Machine**
- **Apa itu Docker?**
- **Keuntungan dari Docker**
- **Komponen dari Docker**
- **Instalasi Docker**
- **Docker Release**
- **Docker OS**
- **Docker Commands**
- **Docker Compose**
- **Docker File**



Container dan Virtual Machine

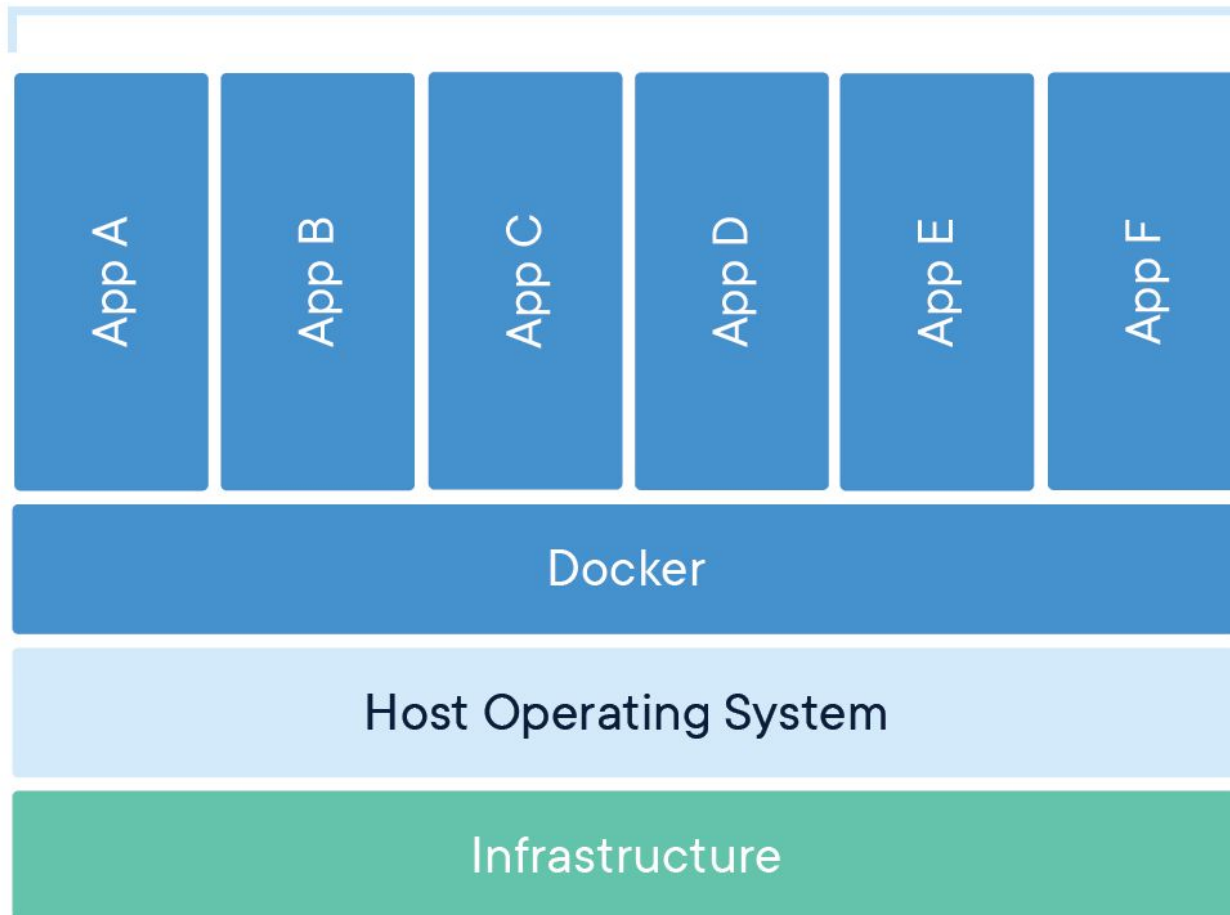


Virtual Machine



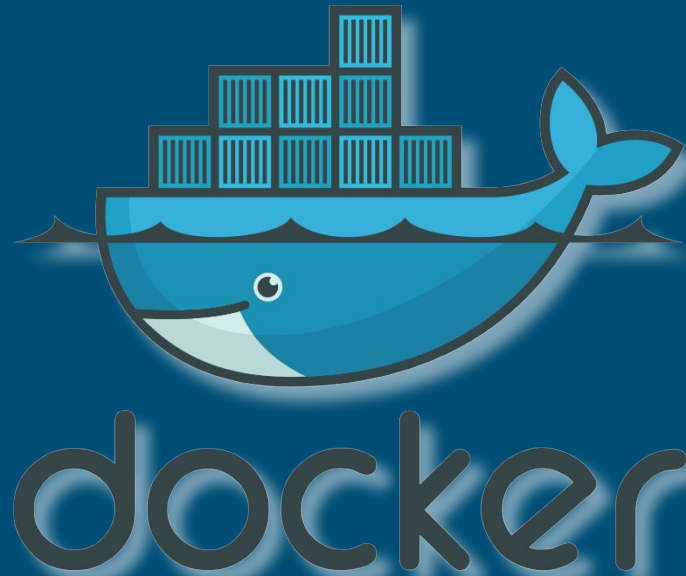
Container

Containerized Applications



Apa itu Docker?





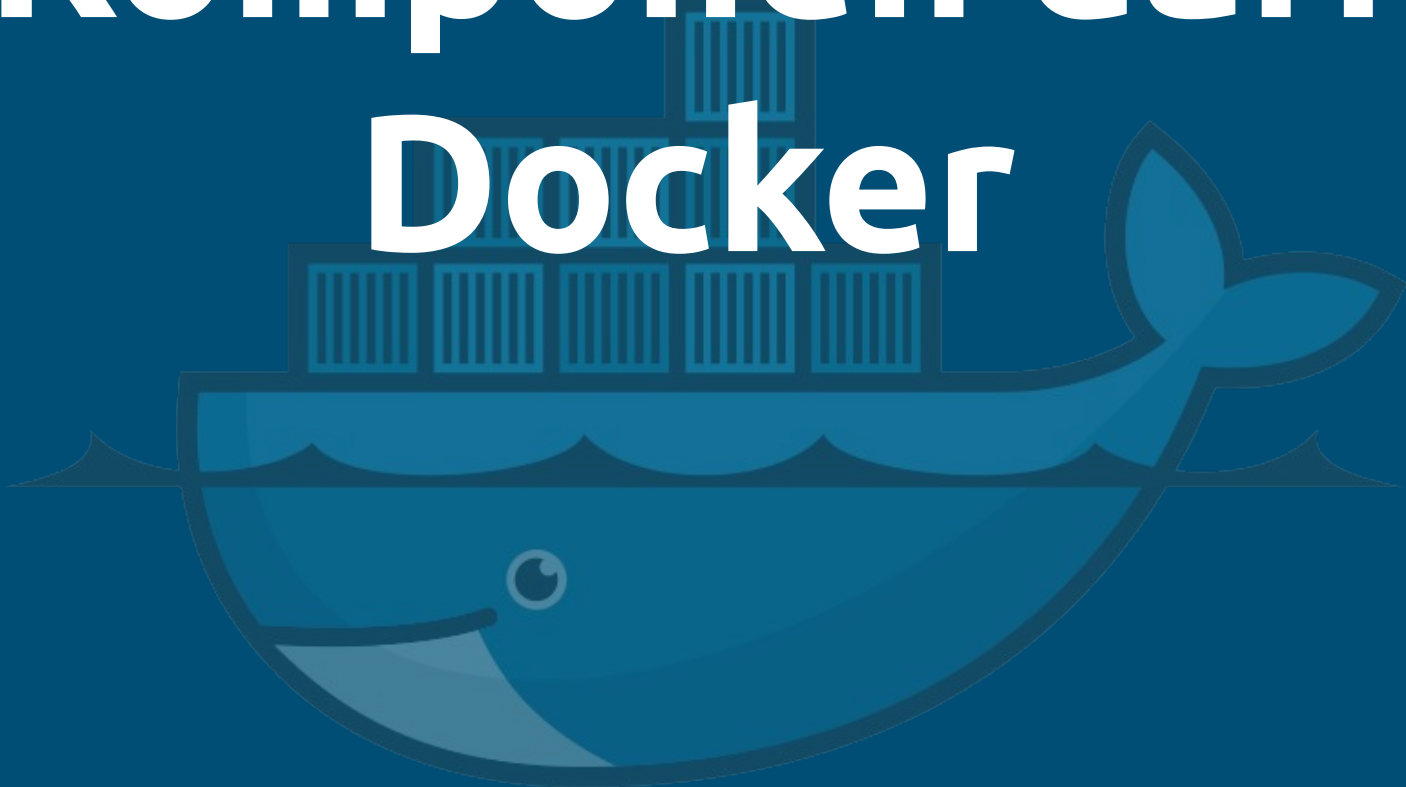
Docker merupakan sebuah alat yang digunakan untuk memudahkan dalam proses deployment aplikasi dalam envirointment (container) yang ringan sehingga aplikasi dapat bekerja secara efisien di lingkungan yang berbeda

Keuntungan dari Docker



Virtual Machine		Docker
High	Memory Usage	Low
Poor	Performance	Good
Bad	Portability	Good
Slow	Boot-up time	Very Fast

Komponen dari Docker



**Docker
Client and
Server**

**Docker
Images**

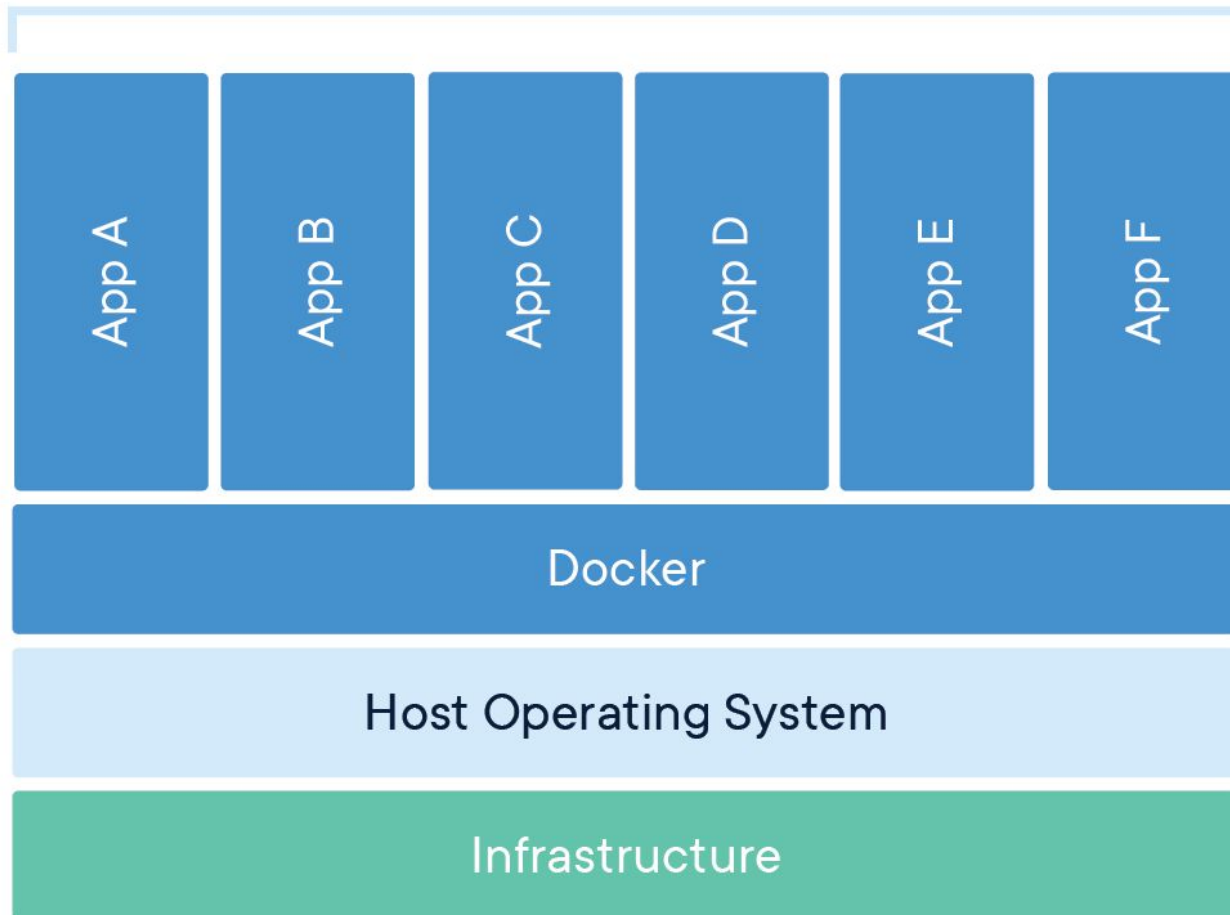
**Docker
Containers**

**Docker
Registry**



Container

Containerized Applications



Instalasi Docker



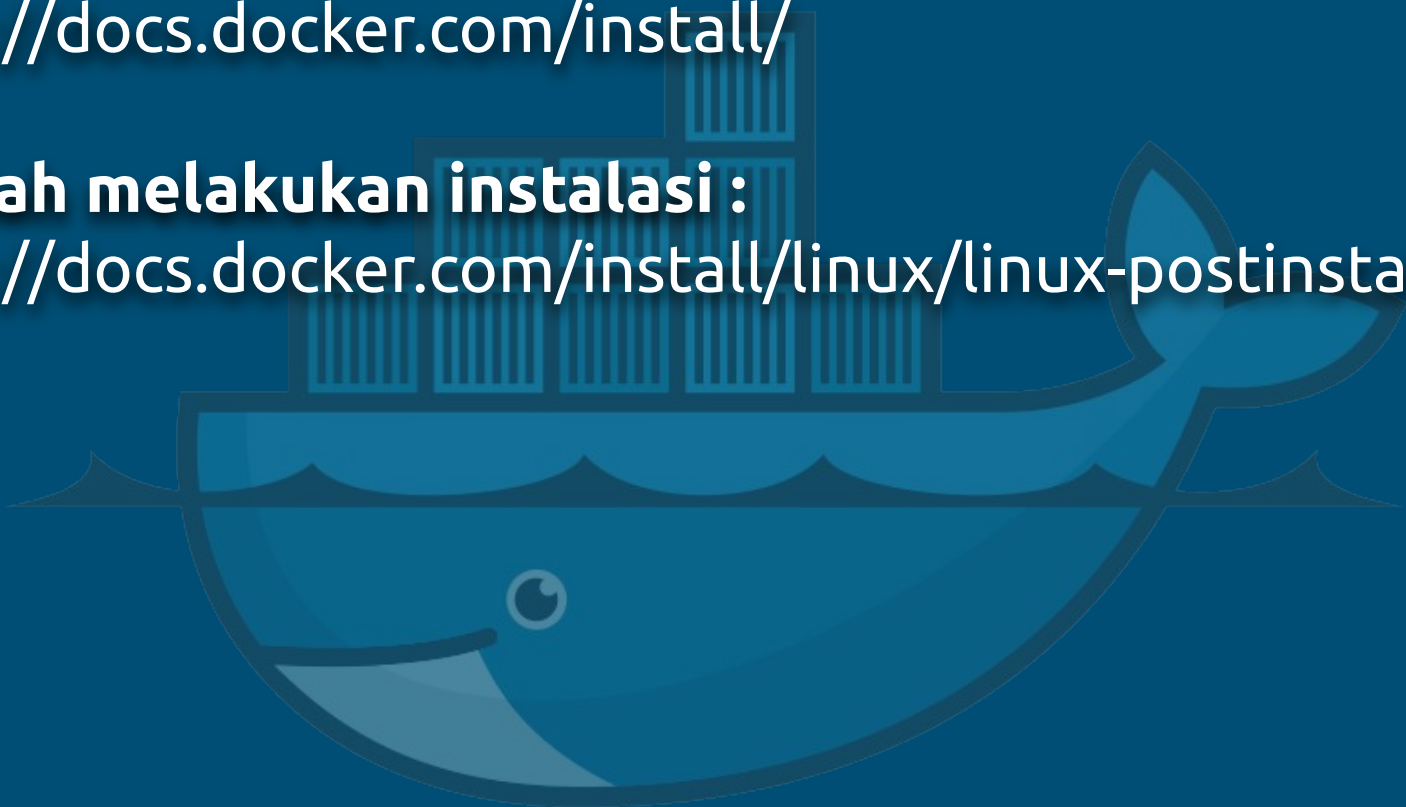
Dokumentasi Docker

Tutorial instalasi Docker :

<https://docs.docker.com/install/>

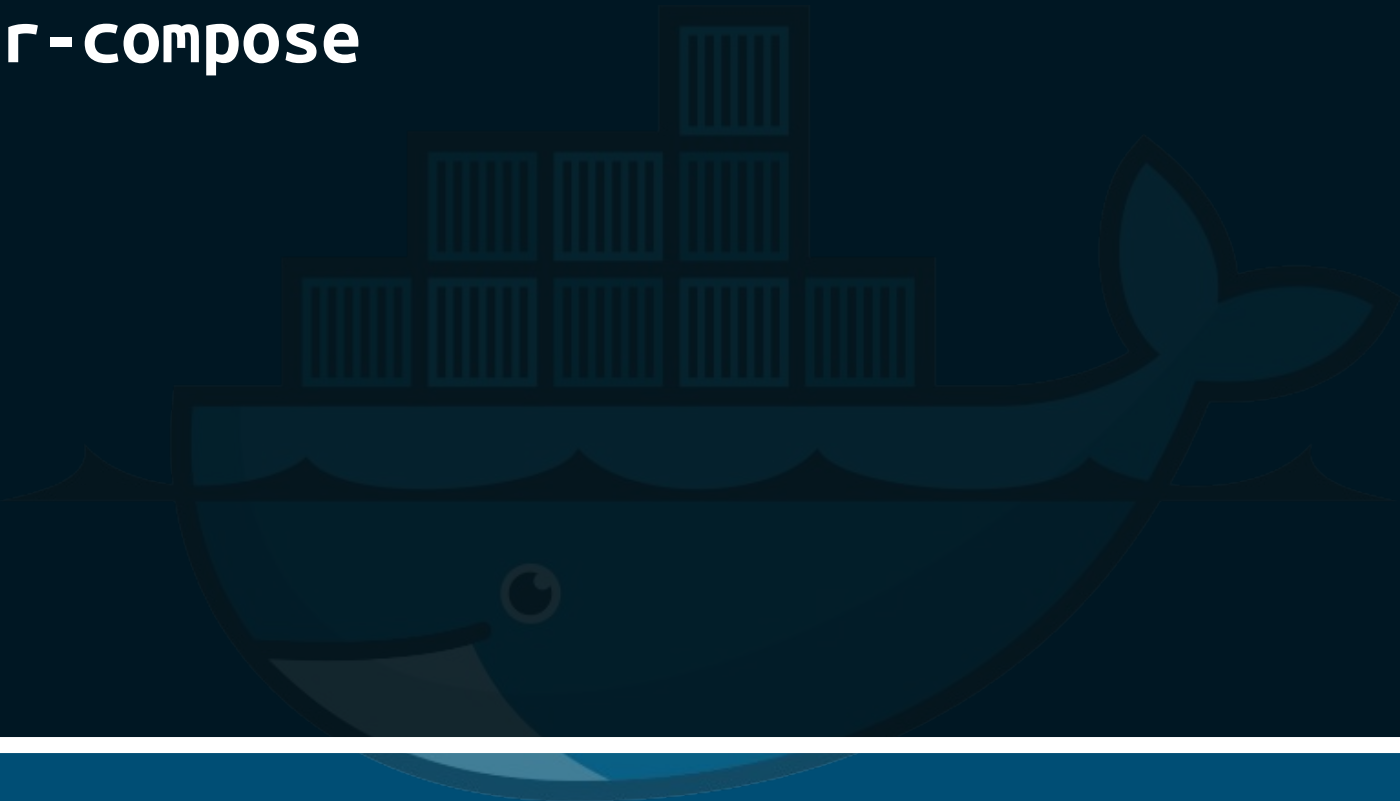
Setelah melakukan instalasi :

<https://docs.docker.com/install/linux/linux-postinstall/>



Install

```
$ sudo apt-get update  
$ sudo apt-get install -y docker.io  
docker-compose
```



Test

```
$ sudo docker run hello-world
```



Post Installation (Linux)

Menjalankan perintah docker tanpa harus menggunakan perintah sudo :

```
$ sudo usermod -aG docker $USER
```

Membuat layanan Docker berjalan saat komputer dihidupkan :

```
$ sudo systemctl enable docker
```

```
$ sudo systemctl start docker
```

Docker Release



Latest Stable Version

Release versi stable terakhir :
18.09.4-ce (28 Maret 2019)

Memiliki skema versi x.y.z-. Memiliki arti :

x = Major Releases

y = Minor Releases

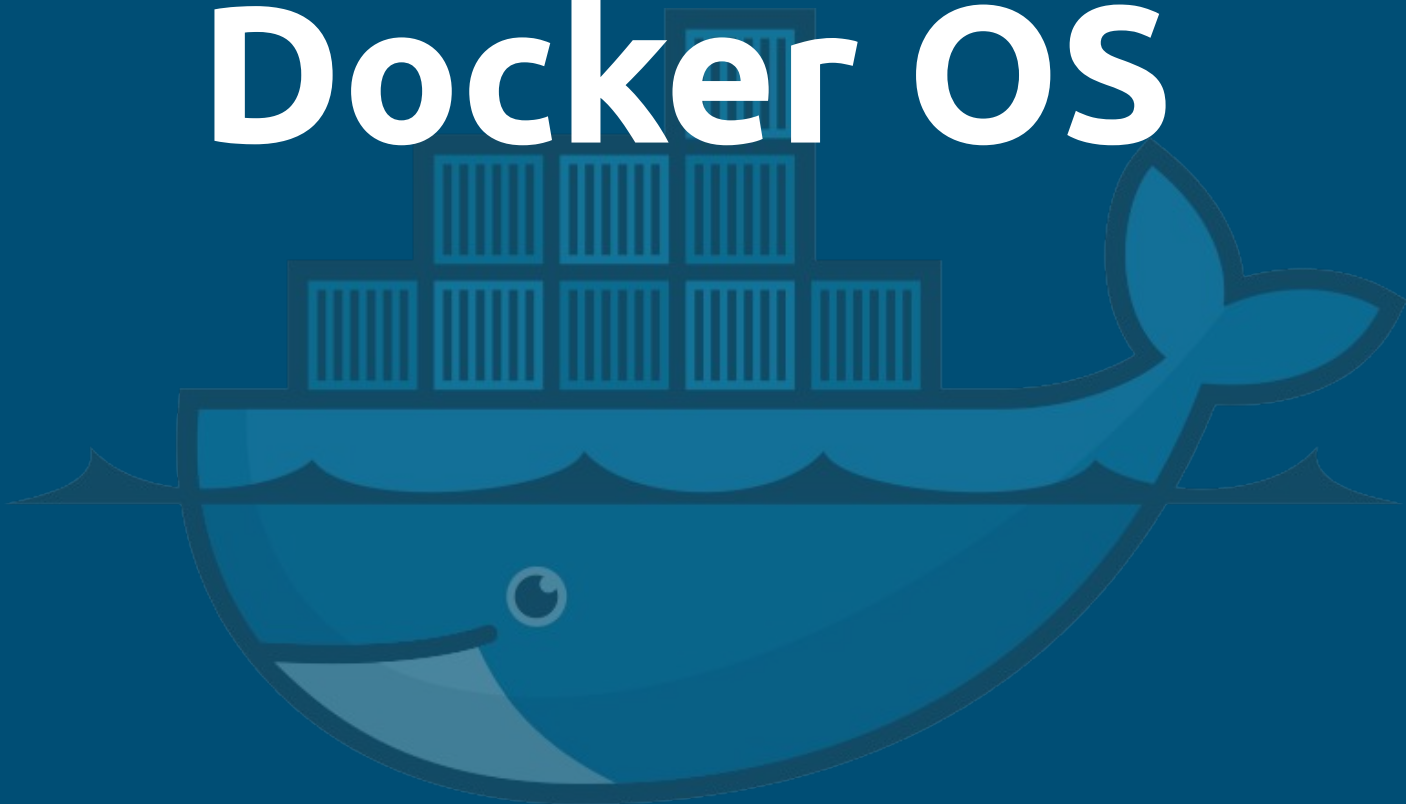
z = Maintenance Releases

Melihat Versi Docker

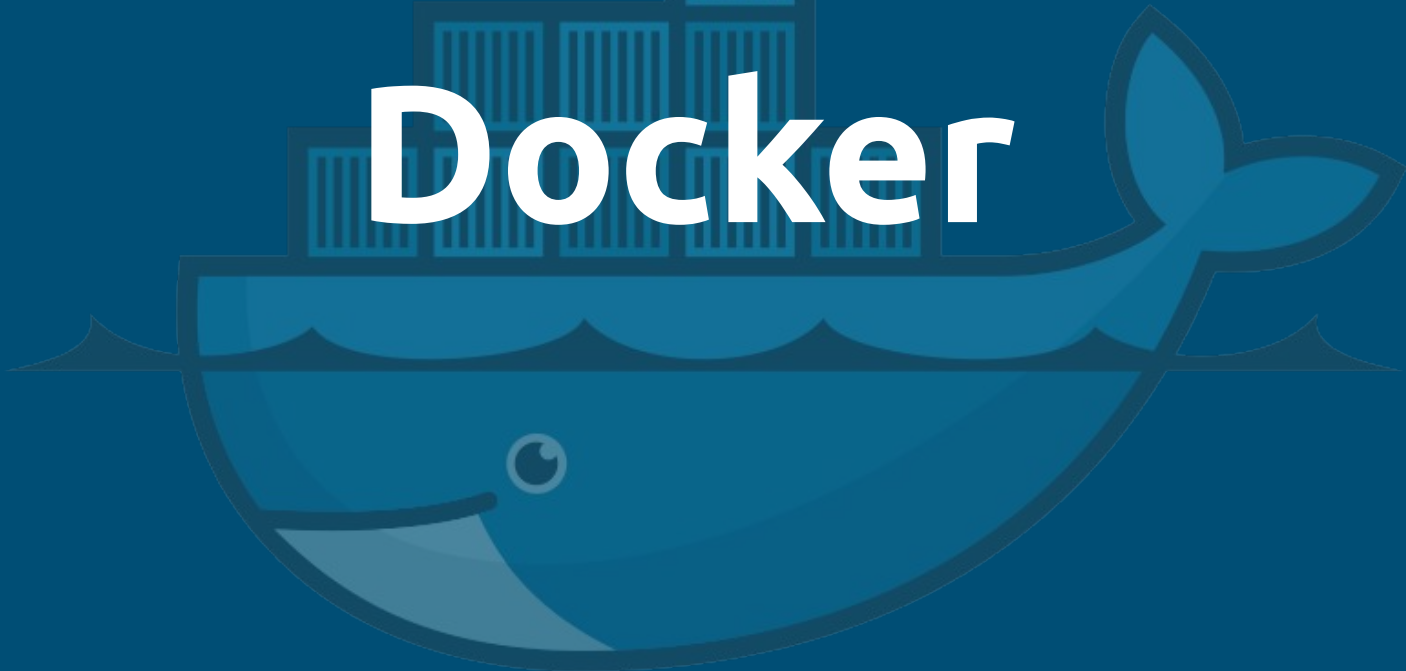
```
$ docker -v
```



Docker OS

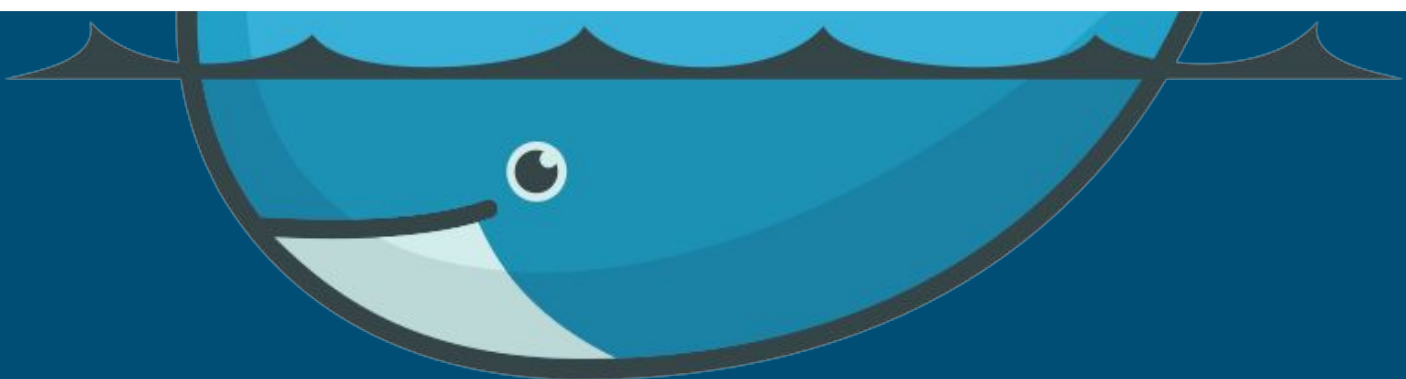


**It's not a brand
new OS called
Docker**



Docker OS (Community Edition)

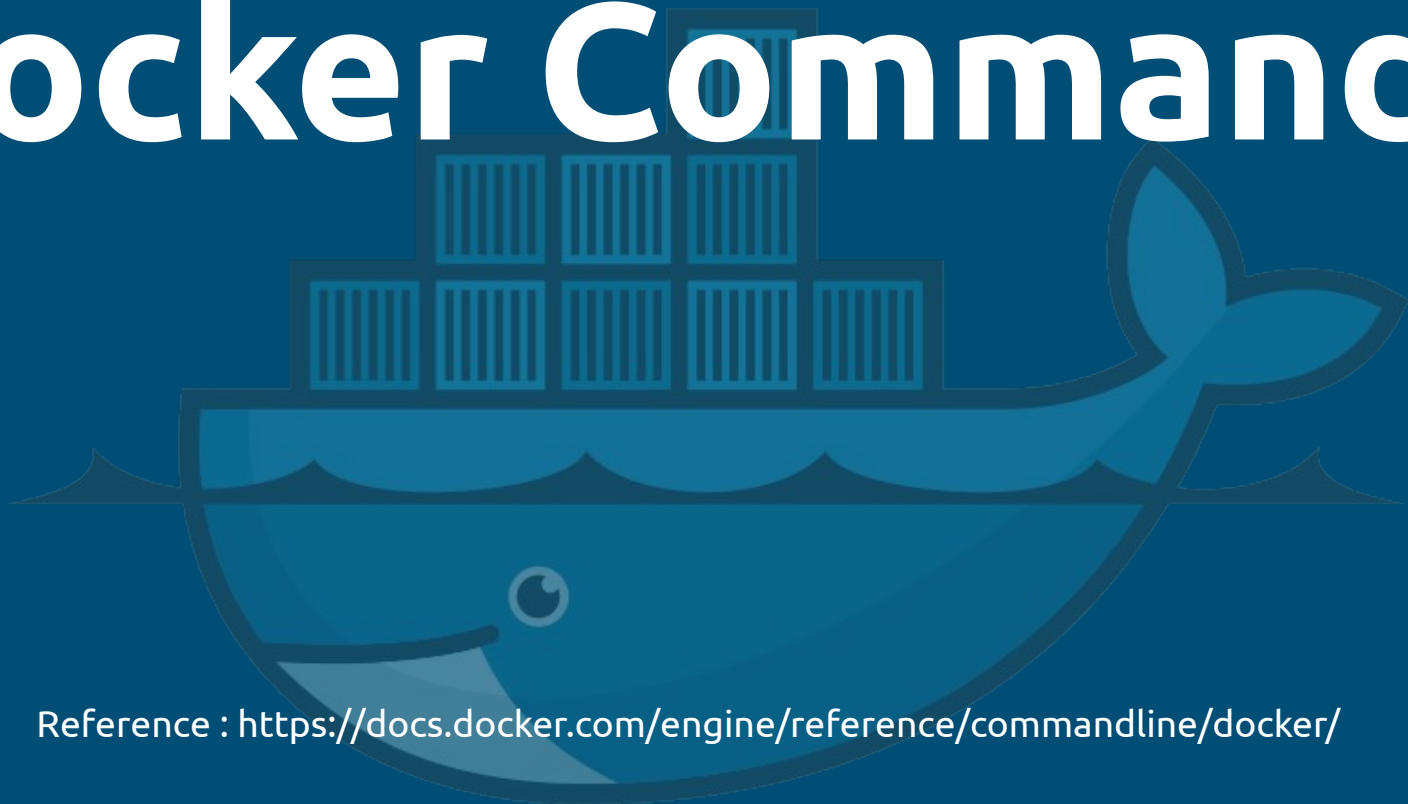
- Ubuntu : https://hub.docker.com/_/ubuntu
- CentOS : https://hub.docker.com/_/centos
- Fedora : https://hub.docker.com/_/fedora
- openSUSE : https://hub.docker.com/_/opensuse
- Debian : https://hub.docker.com/_/debian
- Alpine Linux : https://hub.docker.com/_/alpine



**Visit
hub.docker.com
for more list**



Docker Commands



Reference : <https://docs.docker.com/engine/reference/commandline/docker/>

Manage Images

```
$ docker image COMMAND
```

Commands:

inspect	Melihat secara lengkap informasi dari satu atau lebih image
ls	Menampilkan daftar image
prune	Menghapus image yang tidak dipakai
pull	Menarik image/repository dari registry
push	Mendorong image/repository ke registry
rm	Menghapus satu atau lebih image
tag	Membuat sebuah tag baru dari sebuah image

Melihat Daftar Image

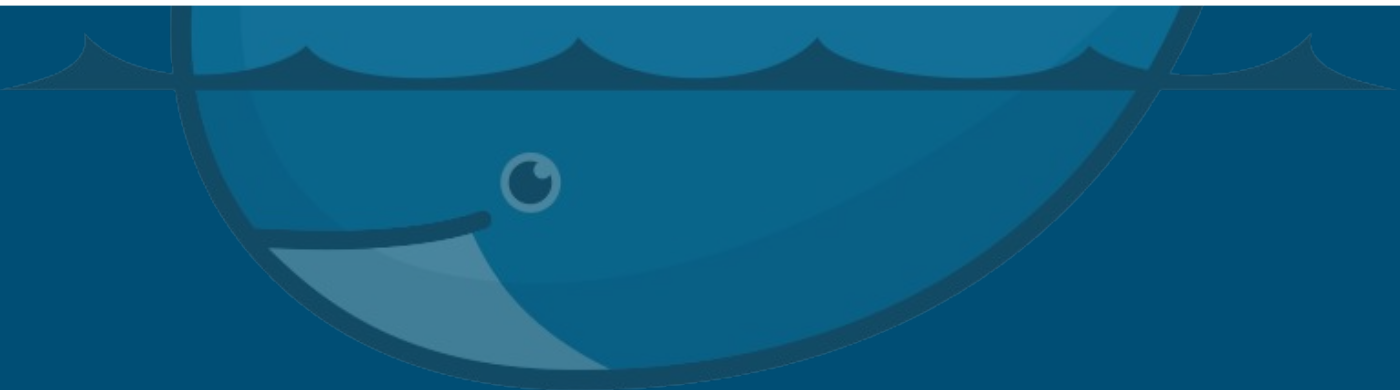
```
$ sudo docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED
centos	latest	9f38484d220f	3 weeks ago
hello-world	latest	fce289e99eb9	3 months ago

SIZE

202MB

1.84kB



Manage Containers

```
$ docker container COMMAND
```

Commands:

exec	Menjalankan sebuah perintah dalam container yang sedang berjalan
inspect	Melihat secara lengkap informasi dari satu atau lebih container
kill	Menghentikan satu atau lebih container yang sedang berjalan
logs	Melihat log dari sebuah container
prune	Menghapus semua container yang berhenti
rename	Mengganti nama container
rm	Menghapus satu atau lebih container
run	Menjalankan perintah dari container baru

Melihat Daftar Container

```
$ sudo docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

```
$ sudo docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
b7766651b752	hello-world	"/hello"	3 minutes ago	Exited (0)		objective_wright

Manage Network

\$ docker network COMMAND

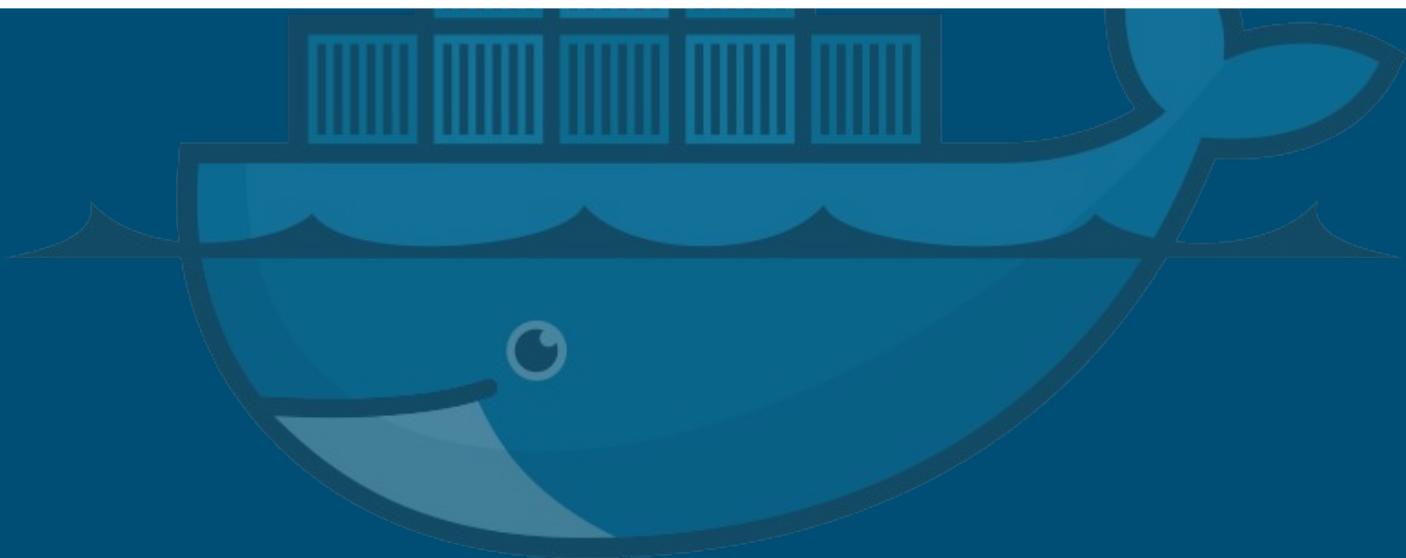
Commands:

connect	Menghubungkan container dengan network
create	Membuat sebuah network baru
disconnect	Memutuskan container dengan network
inspect	Menampilkan informasi secara detail
tentang satu atau lebih network	
ls	Melihat daftar network
prune	Menghapus network yang tidak digunakan
rm	Menghapus satu atau lebih network

Melihat Daftar Network

```
$ sudo docker network ls
```

NETWORK ID	NAME	DRIVER	SCOPE
33303e713dea	bridge	bridge	local
ceb30651bbd9	host	host	local
9a33f6b221ee	none	null	local



Let's Try



Pasang SSH-Key



Pasang SSH-Key

```
$ ssh-copy-id <username>@<ip-vps> -p <port>
```

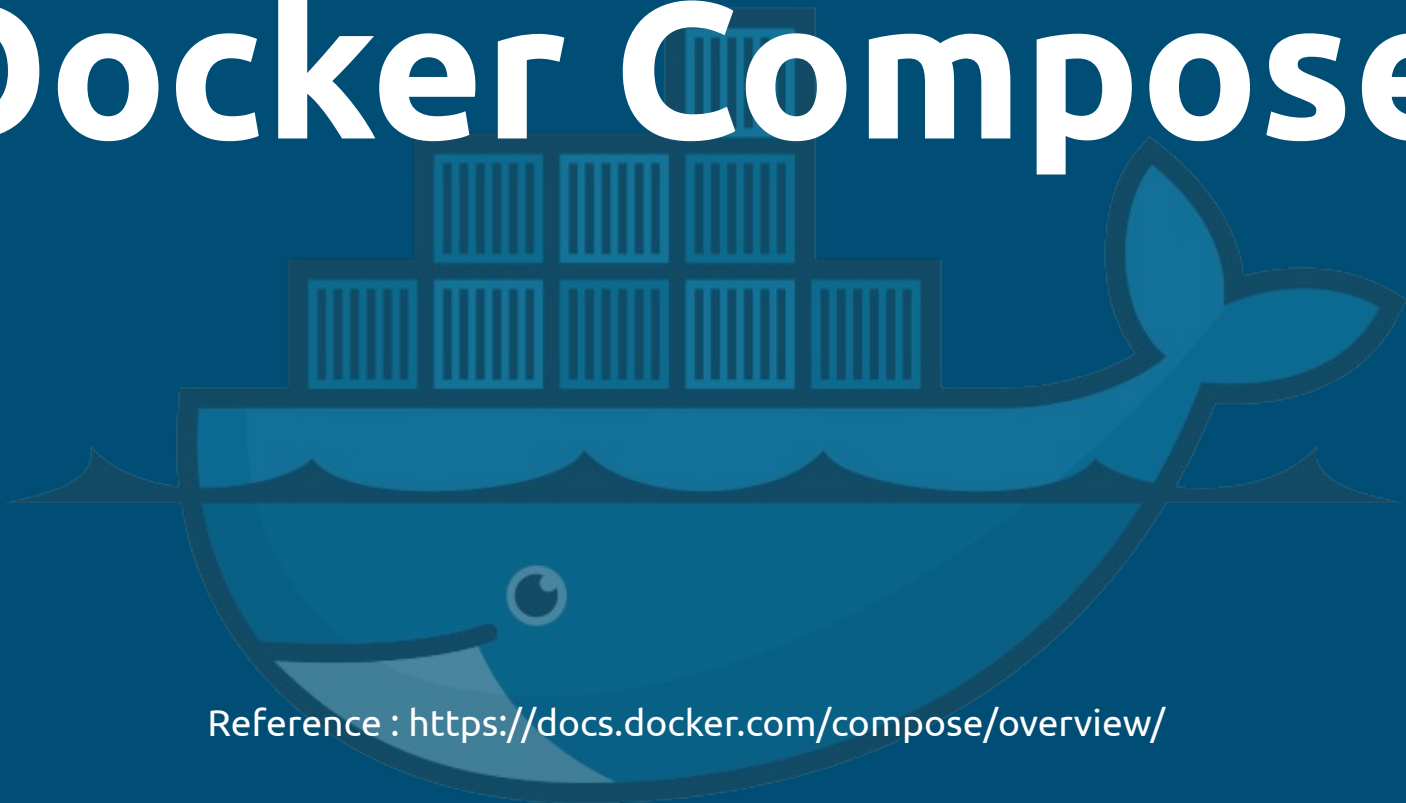


Connect and try this

```
$ docker run -it ubuntu:18.04  
root@aaf7cd26fe18:/#
```

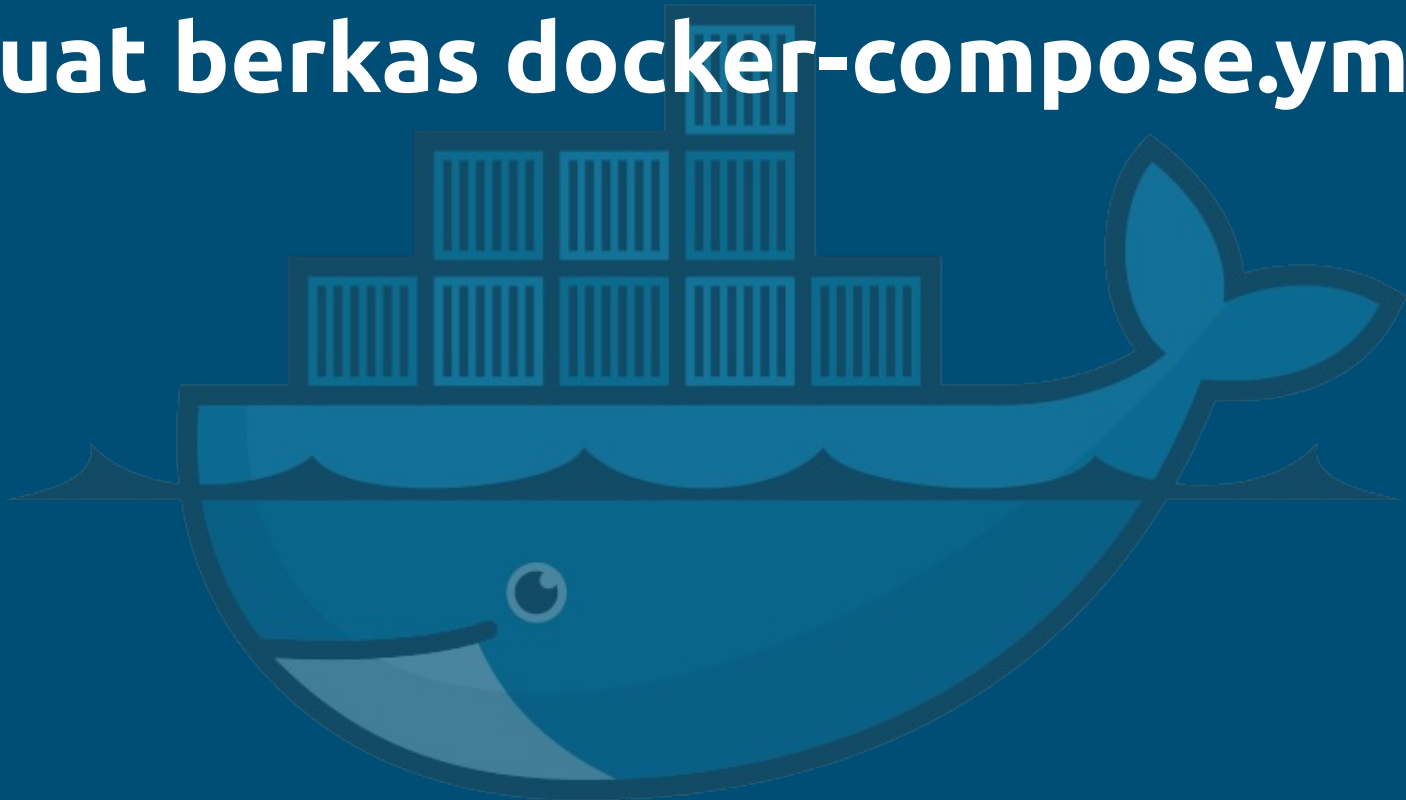


Docker Compose



Reference : <https://docs.docker.com/compose/overview/>

Buat berkas docker-compose.yml



version: '3.2'

services:

db:

image: mysql:5.7

volumes:

- dbdata:/var/lib/mysql

restart: always

environment:

MYSQL_ROOT_PASSWORD: somewordpress

MYSQL_DATABASE: wordpress

MYSQL_USER: [username]

MYSQL_PASSWORD: [password]

wordpress:

depends_on:

- db

image: wordpress:latest

ports:

- "8000:80"

restart: always

environment:

WORDPRESS_DB_HOST: db:3306

WORDPRESS_DB_USER: [username]

WORDPRESS_DB_PASSWORD: [password]

volumes:

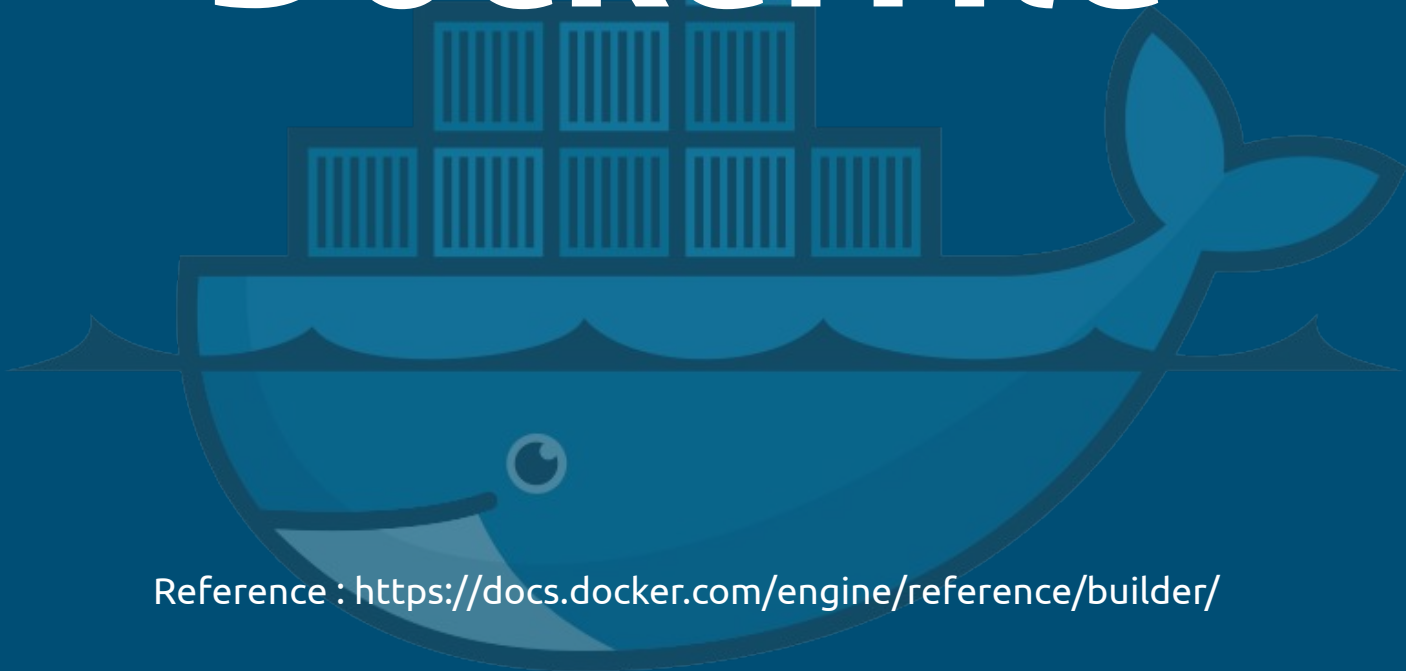
dbdata:



**Jalankan compose
sudo docker-compose up**



Dockerfile



Reference : <https://docs.docker.com/engine/reference/builder/>

Dockerfile merupakan sebuah dokumen teks yang berisi semua perintah yang dapat dipanggil user pada baris perintah untuk merakit image



Reference : <https://docs.docker.com/engine/reference/builder/>

Struktur Direktori Project

- ❏ **my_app**

- ❏ **app.py**

- ❏ **Dockerfile**

- ❏ **docker-compose.yml**

- ❏ **requirements.py**

Reference : <https://docs.docker.com/engine/reference/builder/>

app.py

```
import time
import redis
from flask import Flask

app = Flask(__name__)
cache = redis.Redis(host='redis', port=6379)

def get_hit_count():
    retries = 5
    while True:
        try:
            return cache.incr('hits')
        except redis.exceptions.ConnectionError as exc:
            if retries == 0:
                raise exc
            retries -= 1
            time.sleep(0.5)

@app.route('/')
def hello():
    count = get_hit_count()
    return 'Hello World! I have been seen {} times.\n'.format(count)

if __name__ == "__main__":
    app.run(host="0.0.0.0", debug=True)
```



requirements.txt

```
flask  
redis
```



Dockerfile

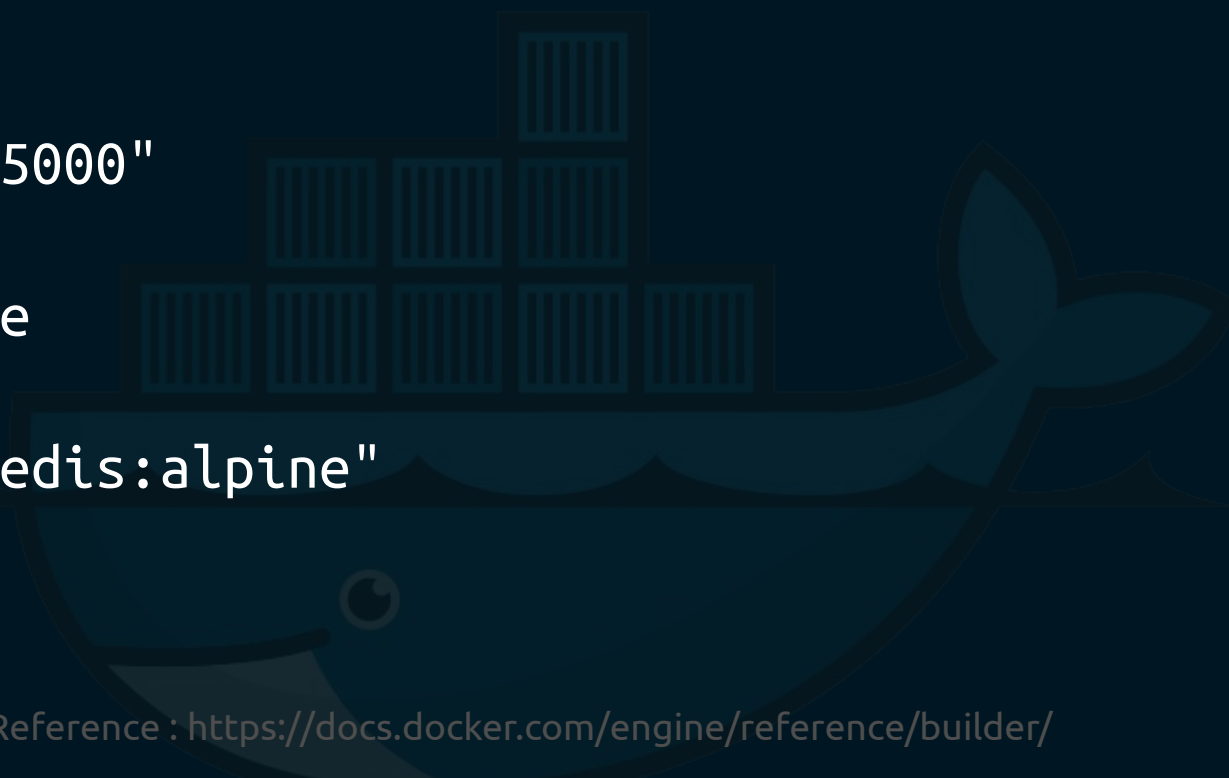
```
FROM python:3.4-alpine
ADD . /code
WORKDIR /code
RUN pip install -r requirements.txt
CMD ["python", "app.py"]
```



Reference : <https://docs.docker.com/engine/reference/builder/>

docker-compose.yml

```
version: '3.2'
services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - .:/code
  redis:
    image: "redis:alpine"
```



Reference : <https://docs.docker.com/engine/reference/builder/>

Thank You

