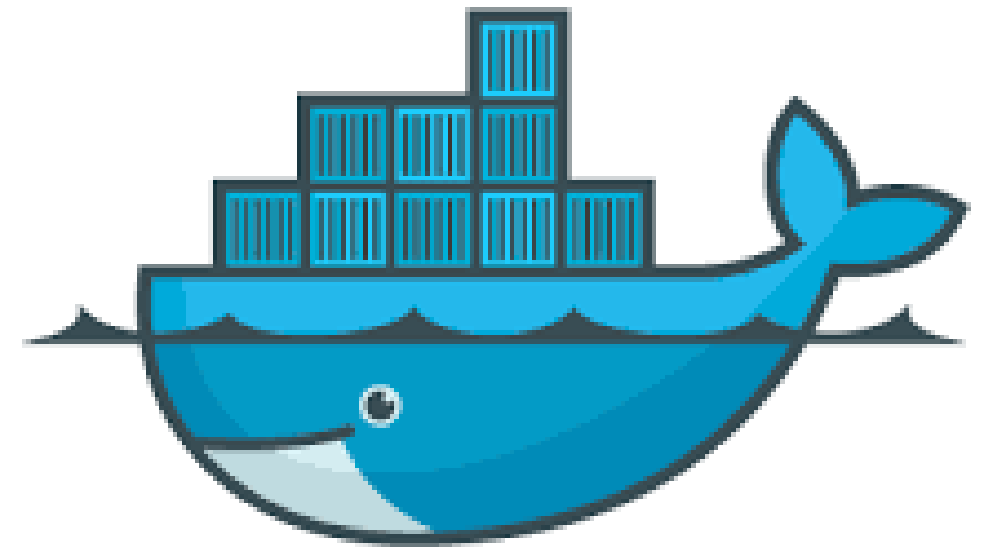


DARA FIBRIANA



docker

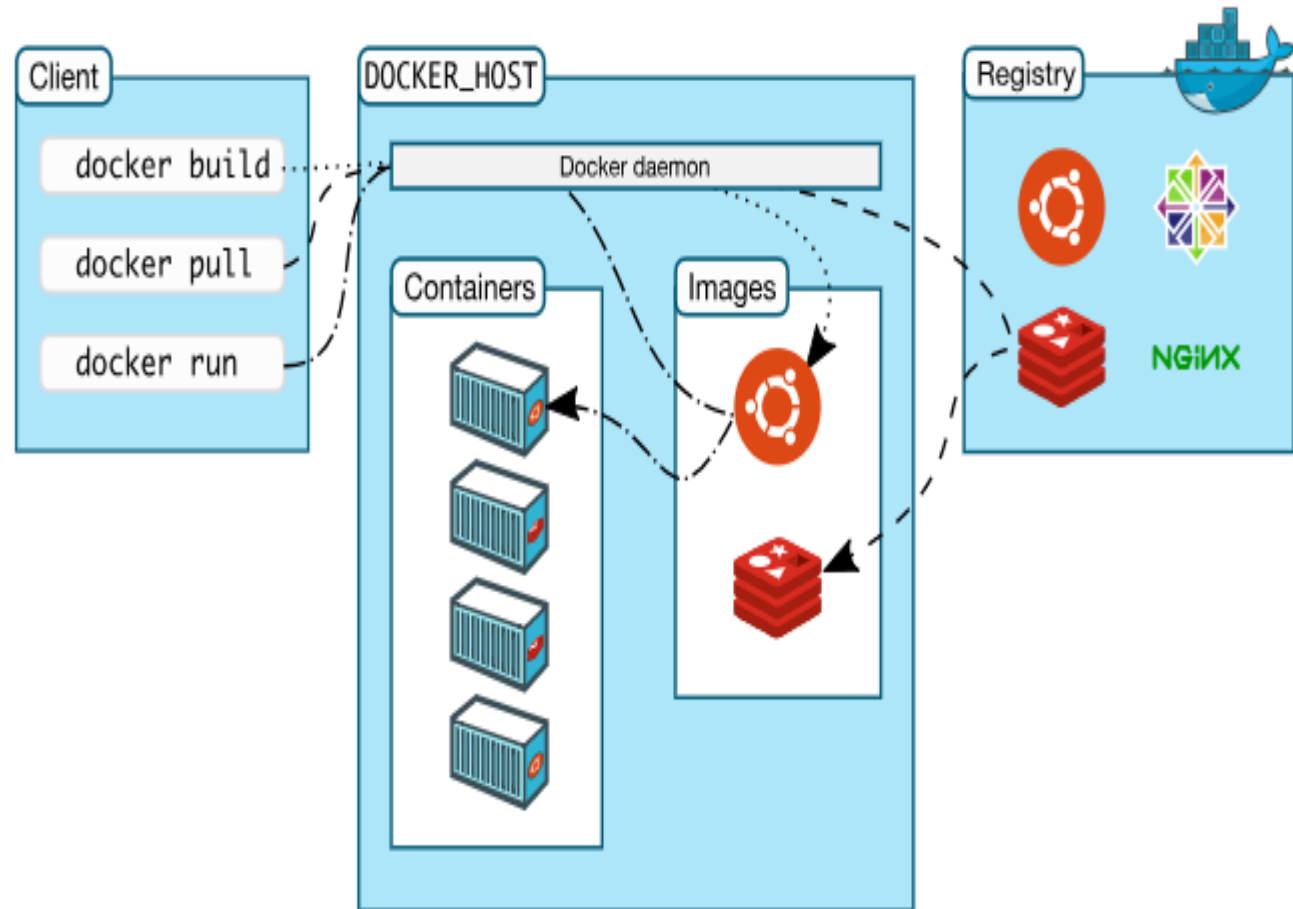
# DOCKER ITU APA ?

---

- Docker adalah salah satu platform yang dibangun berdasarkan teknologi container. Docker merupakan sebuah project open-source yang menyediakan platform terbuka untuk dapat membangun, mengemas, dan menjalankan aplikasi dimanapun sebagai sebuah wadah (container) yang ringan.
- Tools Manage Container
- Docker berfungsi sebagai virtualisasi sebuah sistem operasi atau sebuah server atau sebuah web server atau bahkan sebuah database server, dimana para developer lebih mudah untuk mengatur mengenai deployment atau menjalankan aplikasi di server production.

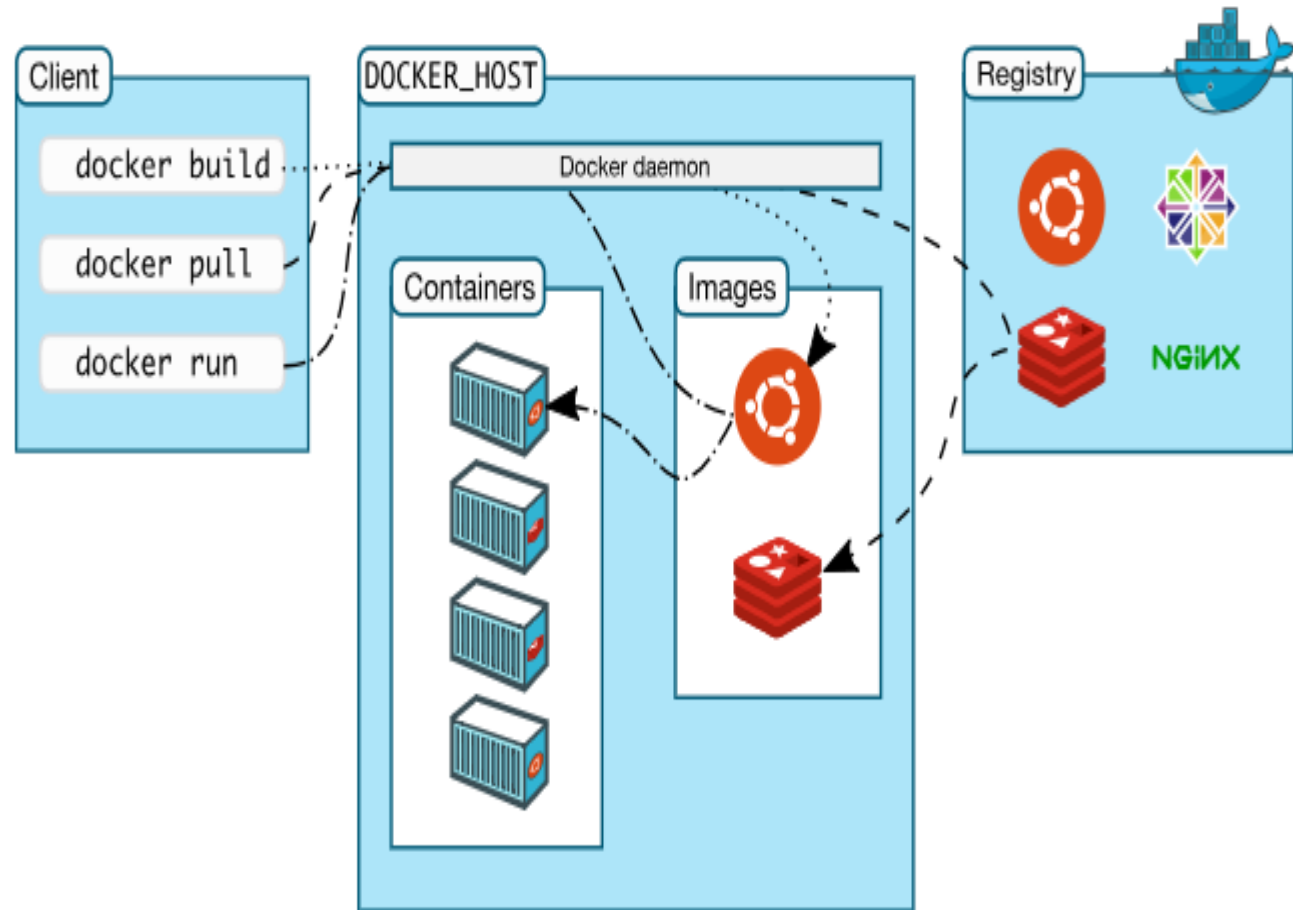
# ARSITEKTUR DOCKER

Gambar tersebut merupakan arsitektur docker, dimana docker terdiri dari beberapa element yaitu docker client, docker daemon, docker container, docker images dan docker registry. Docker menggunakan teknologi client server untuk menghubungkan antara docker client dan docker daemon.



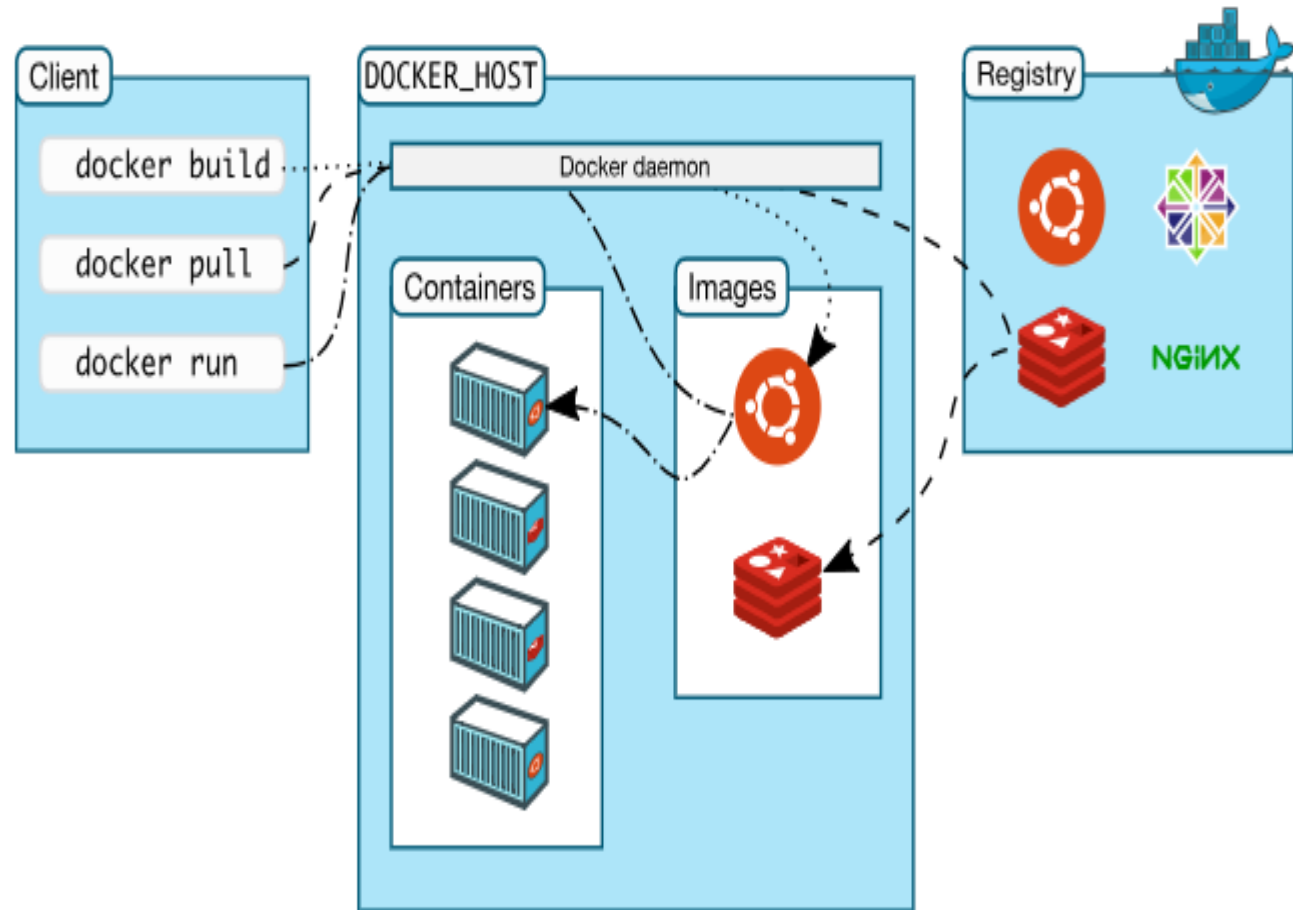
# ARSITEKTUR DOCKER

Docker daemon berfungsi untuk membangun, mendistribusikan dan menjalankan container docker. User tidak dapat langsung menggunakan docker daemon, akan tetapi untuk menggunakan docker daemon maka user menggunakan docker client sebagai perantara atau cli.



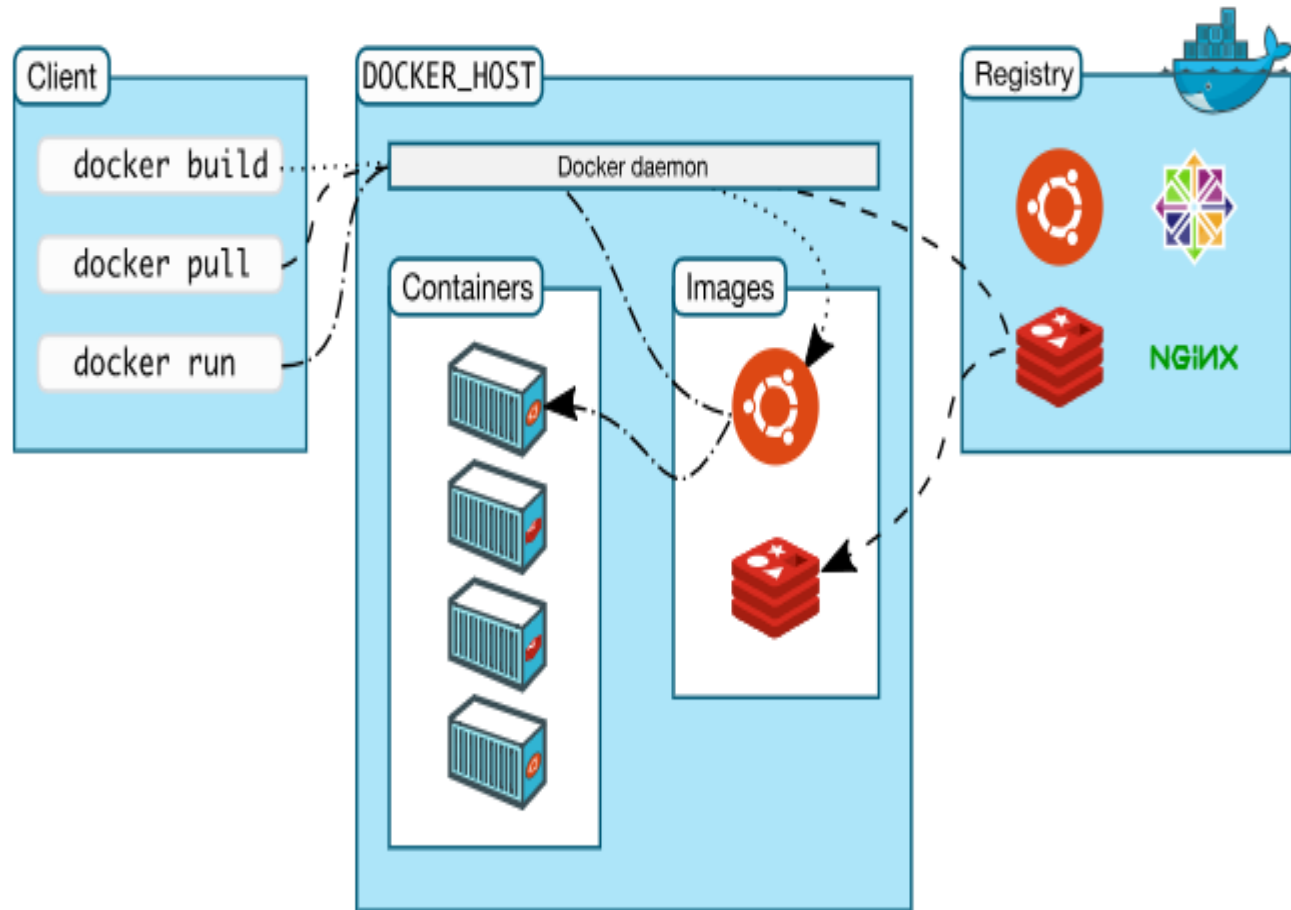
# ARSITEKTUR DOCKER

Docker images adalah sebuah template yang bersifat read only. Template ini sebenarnya adalah sebuah OS atau OS yang telah diinstall berbagai aplikasi. Docker images berfungsi untuk membuat docker container, dengan hanya 1 docker images kita dapat membuat banyak docker container.



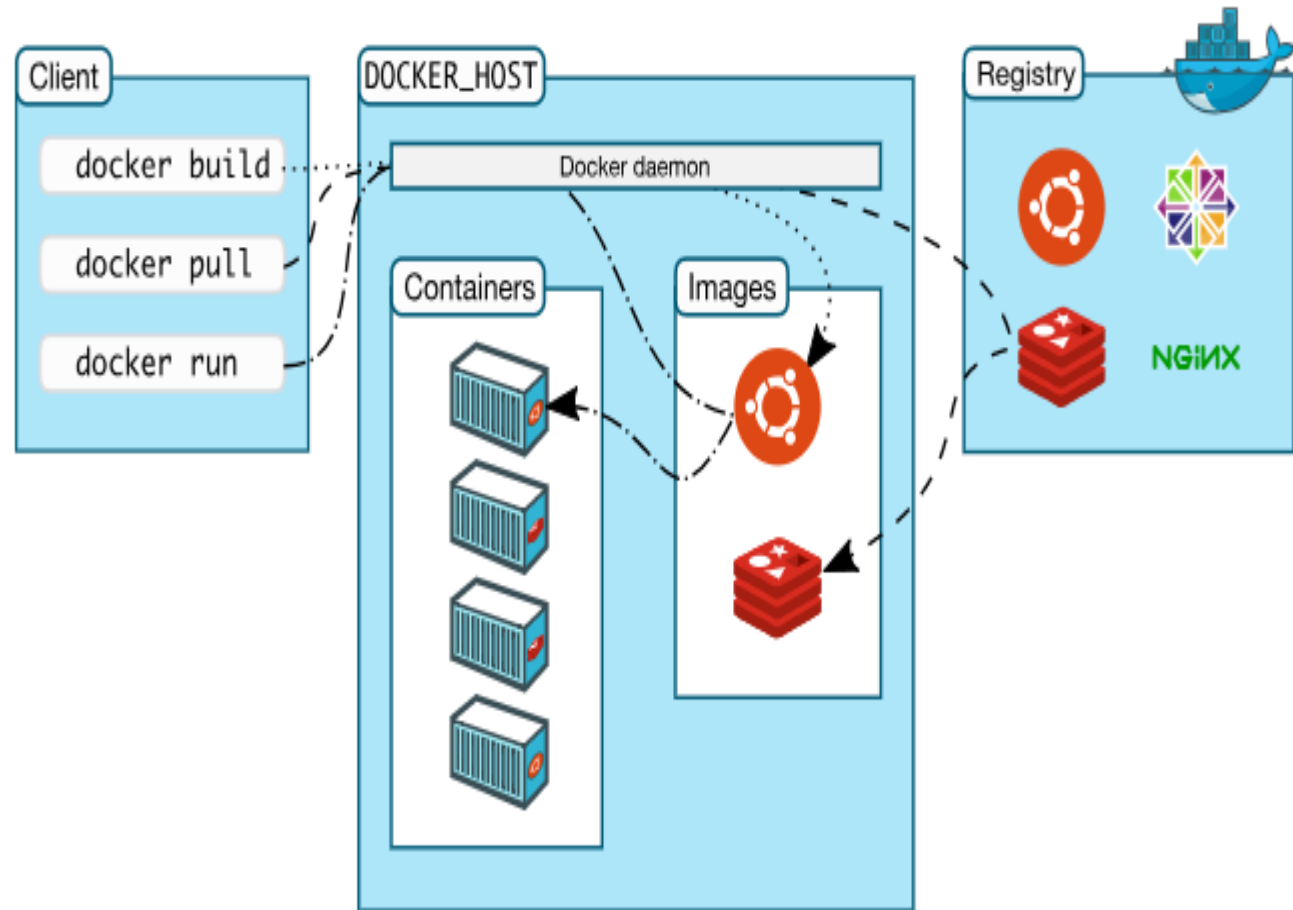
# ARSITEKTUR DOCKER

Docker container bisa dikatakan sebagai sebuah folder, dimana docker container ini dibuat dengan menggunakan docker daemon. Setiap docker container disimpan maka akan terbentuk layer baru tepat diatas docker images atau base image diatasnya. Docker container ini nantinya dapat dibuild sehingga akan menghasilkan sebuah docker images, dan docker images yang dihasilkan dari docker container ini dapat kita gunakan kembali untuk membuat docker container yang baru.



# ARSITEKTUR DOCKER

Docker registry adalah kumpulan docker image yang bersifat private maupun public yang dapat anda akses di docker hub. Manfaat Docker registry yaitu dapat menggunakan docker image yang telah dibuat oleh developer yang lain, sehingga mempermudah dalam pengembangan aplikasi.



# MANFAAT DARI DOCKER

---

- Dapat melakukan Pengujian dan Distribusi Aplikasi secara Terus-menerus
- Platform Multi-Cloud (Salah satu manfaat terbesar Docker adalah portabilitas).
- Standarisasi Lingkungan dan Kontrol Versi (kontainer Docker memastikan konsistensi di beberapa siklus pengembangan dan rilis, standarisasi lingkungan. Di atas semua itu, wadah Docker bekerja seperti repositori GIT, yang memungkinkan untuk melakukan perubahan ke Image Docker dan mengendalikan versi.)
- Isolasi (Docker memastikan aplikasi dan sumber daya yang terisolasi serta terpisah.)
- Docker dapat meng-optimalisasi infrastruktur yang mana seluruh pekerjaan pembuatan aplikasi, modul, monitoring, serta manajemen infrastruktur dapat lebih cepat, efisien dan lebih aman.



# MEMBUAT DOCKER

MINGW64:/c/Program Files/Docker Toolbox

```
Starting "default"...
(default) Check network to re-create if needed...
(default) Windows might ask for the permission to configure a dhcp server. Sometimes, such confirmation window is minimized in the taskbar.
(default) Waiting for an IP...
Machine "default" was started.
Waiting for SSH to be available...
Detecting the provisioner...
Started machines may have new IP addresses. You may need to re-run the `docker-machine env` command.
Regenerate TLS machine certs? Warning: this is irreversible. (y/n): Regenerating TLS certificates
Waiting for SSH to be available...
Detecting the provisioner...
Copying certs to the local machine directory...
Copying certs to the remote machine...
Setting Docker configuration on the remote daemon...
```

```

      ##
    ## ##
  ## ## ## ##
/*****/
NNN { NN NNNN NNN NNNN NNN N / ****
    \____/
      O
    \____/

```

**docker** is configured to use the **default** machine with IP **192.168.99.100**  
For help getting started, check out the docs at <https://docs.docker.com>

Start interactive shell

User@DESKTOP-CVF9FRJ MINGW64 /c/Program Files/Docker Toolbox

\$ docker --version

Docker version 18.03.0-ce, build 0520e24302

# MEMBUAT DOCKER

MINGW64:/c/Program Files/Docker Toolbox

User@DESKTOP-CVF9FRJ MINGW64 /c/Program Files/Docker Toolbox

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

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID:

<https://hub.docker.com/>

For more examples and ideas, visit:

<https://docs.docker.com/get-started/>

User@DESKTOP-CVF9FRJ MINGW64 /c/Program Files/Docker Toolbox

```
$ docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hello-world	latest	4ab4c602aa5e	4 weeks ago	1.84kB


User@DESKTOP-CVF9FRJ MINGW64 /c/Program Files/Docker Toolbox

```
$ docker container ls --all
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
01991dfdf3b	hello-world	"/hello"	23 seconds ago	Exited (0) 24 seconds ago		zealous_borg
740bc0a0d67d	hello-world	"/hello"	About a minute ago	Exited (0) About a minute ago		loving_ritchie
23127a6a50b2	hello-world	"/hello"	3 hours ago	Exited (0) 3 hours ago		xenodochial_proskuriakova

User@DESKTOP-CVF9FRJ MINGW64 /c/Program Files/Docker Toolbox

# MEMBUAT DOCKER

 MINGW64:/c/Program Files/Docker Toolbox

User@DESKTOP-CVF9FRJ MINGW64 /c/Program Files/Docker Toolbox

```
$ docker run --interactive --tty ubuntu bash
```

```
Unable to find image 'ubuntu:latest' locally
```

```
latest: Pulling from library/ubuntu
```

```
124c757242f8: Pull complete
```

```
9d866f8bde2a: Pull complete
```

```
fa3f2f277e67: Pull complete
```

```
398d32b153e8: Pull complete
```

```
afde35469481: Pull complete
```

```
Digest: sha256:de774a3145f7ca4f0bd144c7d4ffb2931e06634f11529653b23eba85aef8e378
```

```
Status: Downloaded newer image for ubuntu:latest
```

```
jjroot@89dc77bbb877:/#
```

```
root@89dc77bbb877:/# hostname
```

```
89dc77bbb877
```

```
root@89dc77bbb877:/#
```

# BUILD IMAGE DOCKER

---

```
User@DESKTOP-CVF9FRJ MINGW64 /c/Program Files/Docker Toolbox
$ docker run --detach --publish 80:80 --name webserver nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
802b00ed6f79: Pull complete
5291925314b3: Pull complete
bd9f53b2c2de: Pull complete
Digest: sha256:9ad0746d8f2ea6df3a17ba89eca40b48c47066dfab55a75e08e2b70fc80d929e
Status: Downloaded newer image for nginx:latest
fb4d16bc922f2c4884c6d76ff3d6eabb1c1756c71bc9910681506bd82af7587d
```

```
User@DESKTOP-CVF9FRJ MINGW64 /c/Program Files/Docker Toolbox
$
```

# BUILD IMAGE DOCKER

```
User@DESKTOP-CVF9FRJ MINGW64 /c/Program Files/Docker Toolbox
```

```
$ docker container ls --all
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
NAMES					
89dc77bbb877	ubuntu	"bash"	About an hour ago	Exited (0) 16 seconds ago	
compassionate_archimedes					
01991dfdf3b	hello-world	"/hello"	About an hour ago	Exited (0) About an hour ago	
zealous_borg					
740bc0a0d67d	hello-world	"/hello"	About an hour ago	Exited (0) About an hour ago	
loving_ritchie					
23127a6a50b2	hello-world	"/hello"	4 hours ago	Exited (0) 4 hours ago	
xenodochial_proskuriakova					

# BUILD IMAGE DOCKER

