

## MODUL 12

### Docker Image



#### CAPAIAN PEMBELAJARAN

---

- Mahasiswa dapat memahami dan mempraktikan penggunaan dockerhub dan docker image



#### KEBUTUHAN ALAT/BAHAN/SOFTWARE

---

- Aplikasi putty atau SSH remote
- Aplikasi git/github
- Docker
- Linux Ubuntu 20.04



#### DASAR TEORI

---

Docker adalah platform perangkat lunak yang memungkinkan Anda membuat, menguji, dan menerapkan aplikasi dengan cepat. Docker mengemas perangkat lunak ke dalam unit standar yang disebut [kontainer](#) yang memiliki semua yang diperlukan perangkat lunak agar dapat berfungsi termasuk pustaka, alat sistem, kode, dan waktu proses. Dengan menggunakan Docker, Anda dapat dengan cepat menerapkan dan menskalakan aplikasi ke lingkungan apa pun dan yakin bahwa kode Anda akan berjalan.

Menjalankan Docker di AWS memberi pengembang dan admin cara yang sangat andal dan murah untuk membuat, mengirim, dan menjalankan aplikasi terdistribusi dalam segala skala.

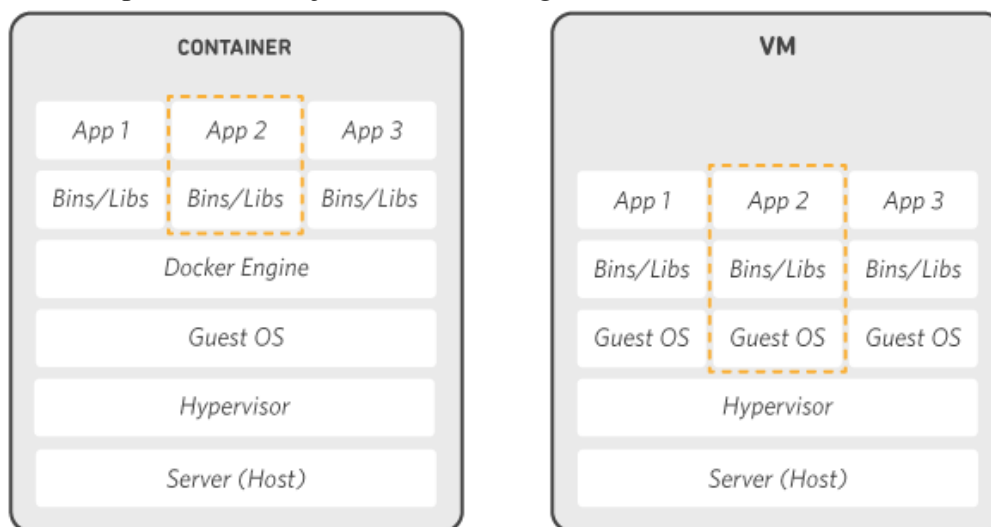
Pengumuman terbaru: Docker berkolaborasi dengan AWS untuk membantu pengembang mempercepat pengiriman aplikasi modern ke cloud. Kolaborasi ini membantu pengembang menggunakan Docker Compose dan Docker Desktop untuk memanfaatkan alur kerja yang sama dengan yang digunakan saat ini untuk menerapkan aplikasi di Amazon ECS dan AWS Fargate dengan mulus.



## Cara kerja Docker

Docker berfungsi dengan menyediakan cara standar untuk menjalankan kode Anda. Docker adalah sistem operasi untuk kontainer. Mirip dengan cara [mesin virtual](#) memvirtualisasi (menghilangkan kebutuhan untuk secara langsung mengelola) perangkat keras server, kontainer memvirtualisasi sistem operasi server. Docker diinstal di setiap server dan memberikan perintah sederhana yang dapat Anda gunakan untuk membuat, memulai, atau menghentikan kontainer.

Layanan AWS seperti [AWS Fargate](#), [Amazon ECS](#), [Amazon EKS](#), dan [AWS Batch](#) mempermudah menjalankan dan mengelola kontainer Docker dalam skala besar.



## Mengapa menggunakan Docker

Menggunakan Docker memungkinkan Anda mengirimkan kode lebih cepat, menstandarisasi operasi aplikasi, memindahkan kode dengan lancar, dan menghemat uang dengan meningkatkan pemanfaatan sumber daya. Dengan Docker, Anda mendapatkan satu objek yang dapat dijalankan di mana saja. Sintaks Docker yang sederhana dan lugas memberi Anda kontrol penuh. Adopsi yang luas berarti ada ekosistem alat yang kuat dan aplikasi off-the-shelf yang siap digunakan dengan Docker.



## PRAKTIK

---

### Docker Images

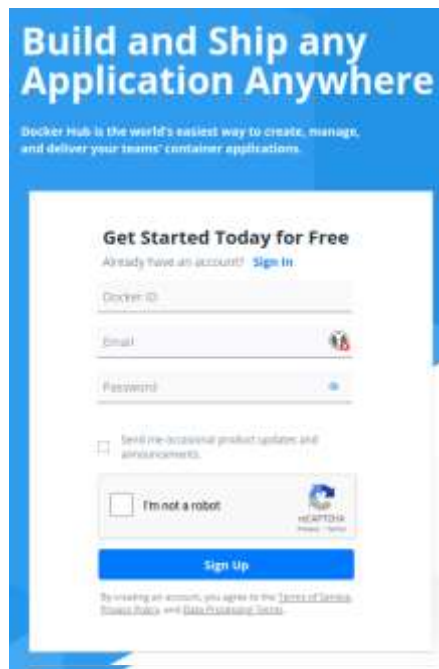
Pengertiannya secara singkat yaitu kumpulan dari beberapa commands/instructions yang terstruktur. Docker images punya 2 tingkatan, top layer dan common layer (layer yang tepat berada di bawah “top layer”). Top layer menggunakan permission Read-Write (RW) sedangkan common layer menggunakan permission Read-Only (RO). Setiap images dan containers secara default punya nickname dan ID yang selalu berbeda (unique) diambil dari Docker Hub. Docker Hub sebagai cloud service dan kumpulan images dari berbagai developer termasuk official partner dari Docker (MySQL, mongoDB, Laravel, Nginx, Apache, dll)

### Docker Hub

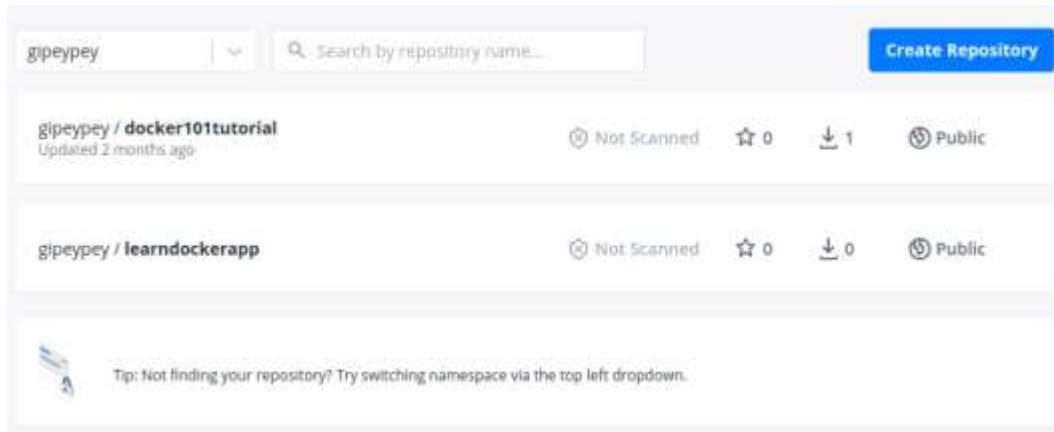
Kita bisa akses Docker Hub [disini](#). Berbagai aktifitas seperti push dan pull bisa dilakukan oleh pengguna Docker client seperti kita saat ini.

#### Overview

- Buka Docker Hub
- Daftar/masukkan email dan password

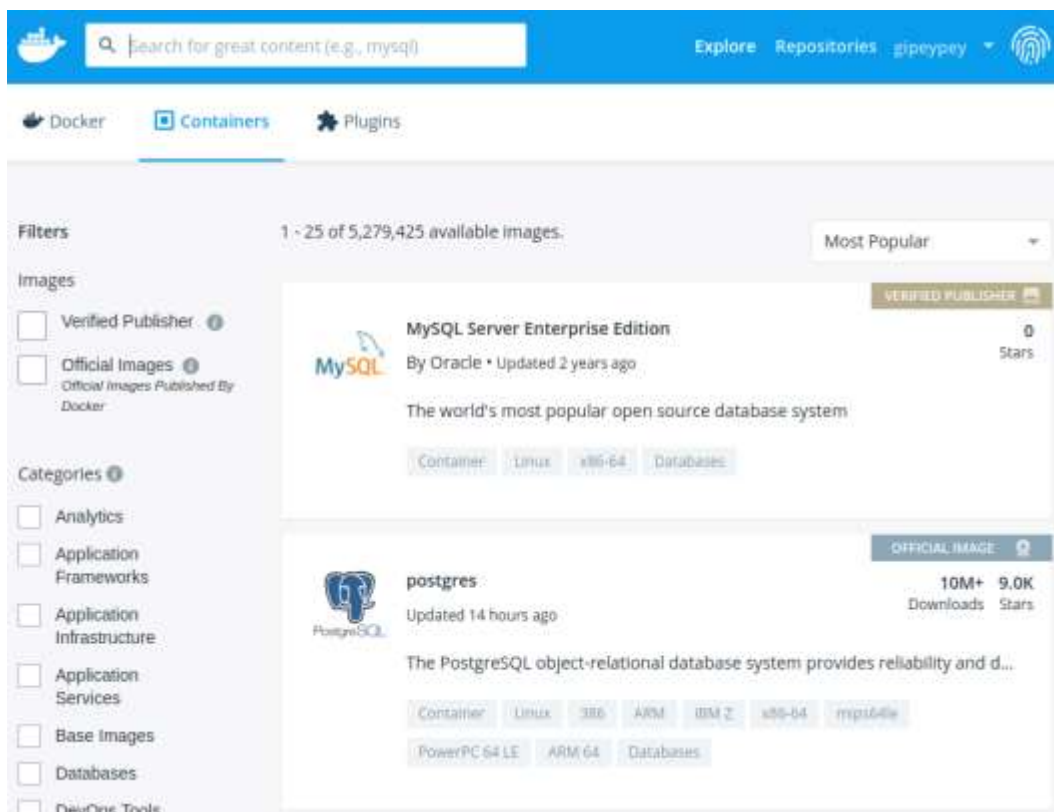


- Dashboard meliputi repository, images, organization, dan audit logs



## Search

- Kita coba mencari images dan containers dari official repository dengan klik explore di navbar



Selanjutnya, kita akan menggunakan terminal. Coba mencari images python  
\$ docker search **python** atau \$ docker search **python:3.6**

```
gipey@Think-E470: ~$ docker search python
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
python	Python is an interpreted, interactive, objec...	5879	[OK]	
django	Django is a free web application framework, ...	1048	[OK]	
pypy	PyPy is a fast, compliant alternative implem...	264	[OK]	
nikolai/python-nodejs	Python with Node.js	65		[OK]
joyzoursky/python-chromedriver	Python with Chromedriver, for running automa...	57		[OK]
arm32v7/python	Python is an interpreted, interactive, objec...	36		
circleci/python	Python is an interpreted, interactive, objec...	42		
centos/python-35-centos7	Platform for building and running Python 3.5...	39		
centos/python-36-centos7	Platform for building and running Python 3.6...	31		
hylang	Hy is a Lisp dialect that translates express...	29	[OK]	
arm64v8/python	Python is an interpreted, interactive, objec...	24		
revolutionsystems/python	Optimized Python Images	18		
centos/python-27-centos7	Platform for building and running Python 2.7...	17		
bitnami/python	Bitnami Python Docker Image	10		[OK]
publicisworldwide/python-conde	Basic Python environments with Conda	8		[OK]
dockershelf/python	Repository for docker images of Python, Test...	5		[OK]
d3fk/python_in_bottle	Simple python:alpine compiled by Bottle+Req...	5		[OK]
clearlinux/python	Python programming interpreted language with...	4		
i386/python	Python is an interpreted, interactive, objec...	3		
centos/python-34-centos7	Platform for building and running Python 3.4...	2		

**Catatan:**

- python (kalo hanya ARG python maka docker membaca versi latest)
- python=3.6 (docker membaca python versi 3.6)

**Gunakan filter untuk mencari keyword tertentu**

```
$ docker search --filter "is-official=true" python
```

```
gipey@Think-E470: ~$ docker search --filter "is-official=true" python
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
python	Python is an interpreted, interactive, objec...	5879	[OK]	
django	Django is a free web application framework, ...	1048	[OK]	
pypy	PyPy is a fast, compliant alternative implem...	264	[OK]	
hylang	Hy is a Lisp dialect that translates express...	29	[OK]	

**Catatan:**

- -filter "keyword yang ingin difilter" <nama images>

is-official=true (official dari nama kolom nya. True adalah sebuah nilai seperti true/false)

**Memunculkan images**

```
$ docker images
```

```
gipey@Think-E470: ~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
img_apache    latest    dfe864991279   2 days ago    186MB
img_expose    latest    bb029fced368   3 days ago    112MB
img_run-env    latest    ba9b44394edd   3 days ago    89.1MB
img_from      latest    7f85e0e6328a   3 days ago    100MB
ubuntu        20.04    f63181f19b2f   4 weeks ago    72.9MB
hello-world    latest    bf756fb1ae65   13 months ago 13.3kB
gipey@Think-E470: ~$
```

Memunculkan images yang lebih spesifik (contohnya ubuntu)

```
$ docker images ubuntuatau$ docker images ubuntu:18.04
```

```
gipey@Think-E470: ~$ docker images ubuntu
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        20.04    f63181f19b2f   4 weeks ago    72.9MB
gipey@Think-E470: ~$
```

### Catatan:

ubuntu:18.04 (nama images beserta tag/versi nya)

### Pull

Bisa dikatakan pull mirip dengan download. Jadi, kita akan mengambil images dari Docker Hub ke Docker Host PC kita.

Pull images Nginx dengan tag latest

```
$ docker pull nginx:latest
```

```
gipey@Think-E470: -
gipey@Think-E470:~$ docker pull nginx:latest
latest: Pulling from library/nginx
45b42c59be33: Downloading [====>] 1.670MB/27.3MB
8acc495f1d91: Downloading [====>] 1.661MB/26.57MB
ec3bd7de96d7: Download complete
19e2441aeeab: Download complete
f5a38c5f8d4e: Waiting
8356d851118: Waiting
```

Pull images Nginx dengan seluruh tag yang tersedia di Docker Hub

\$ docker pull --all-tags nginx

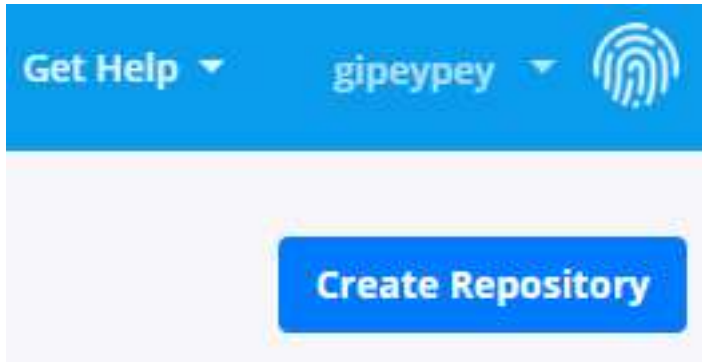
```
gipey@Think-E470: -
gipey@Think-E470:~$ docker pull --all-tags nginx
1-alpine-perl: Pulling from library/nginx
ba3557a56b15: Pull complete
ea8e1cc6ed51: Pull complete
3501a4d88348: Pull complete
0da7fe6dd994: Pull complete
b6a9bc76e363: Pull complete
6ec9bf4ae3da: Pull complete
Digest: sha256:7e0d35bdd73b2d547a4c8108c53c6e10dec4f114e8b26e9b2603579bb7d2177a
1-alpine: Pulling from library/nginx
ba3557a56b15: Already exists
1a18b9f93d41: Downloading [=====] 1.371MB/0.908MB
38ceab6c6432: Download complete
6104f3bd82cc: Verifying Checksum
750e0e12d70c: Waiting
d7c3a871210: Waiting
```

```
gipey@Think-E470:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
lrg_apache           latest              dfe6c4991279       2 days ago         186MB
lrg_expose           latest              b6629fced368       3 days ago         112MB
lrg_run-env          latest              ba9b44394edd       3 days ago         89.1MB
lrg_fron             latest              7f85e8e0328a       3 days ago         100MB
nginx                1-alpine-perl      0285ceab5cbf       5 days ago         57.2MB
nginx                1-alpine           eb9291454164       5 days ago         22.6MB
nginx                1-perl             2acb1236d13e       5 days ago         179MB
nginx                latest             35c43ace9216       5 days ago         133MB
ubuntu              20.04             f63181f19b2f       4 weeks ago        72.9MB
hello-world          latest            bf756fb1ae05       13 months ago      13.3kB
nginx                1.10-alpine       f94d6dd9b576       3 years ago         54MB
nginx                1.10.1-alpine     2cd900f340dd       4 years ago         54MB
nginx                1.10.0            10666ff3a57f       4 years ago         183MB
nginx                1.10.0-alpine     8328c2165672       4 years ago         60.0MB
gipey@Think-E470:~$
```

## Push

Kita pun bisa push/upload images dan container ke Docker Hub

- Pertama, Buat repository di Docker Hub



- Beri nickname dan description pada repository yang akan kita buat

### Create Repository

- Atur privileges, repository nya akan dimunculkan ke public atau private

### Visibility

Using 0 of 1 private repositories. [Get more](#)

☒ **Public**   
Public repositories appear in Docker Hub search results

☐ **Private**   
Only you can view private repositories

- Sekarang, Docker pun bisa terintegrasi dengan Github dan Bitbucket (opsional) dan tekan create

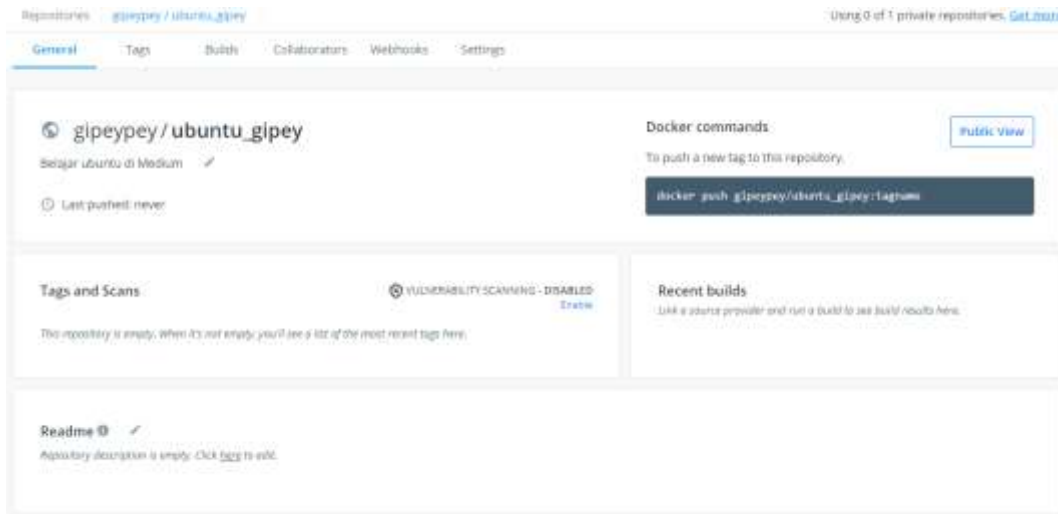


### Build Settings *(optional)*

Autobuild triggers a new build with every **git push** to your source code repository. [Learn More.](#)



- Berikut tampilan dari admin yaitu kita sendiri, akan ada Docker push command untuk upload dari Docker Host/Machine yang kita pake sekarang ke Docker Hub

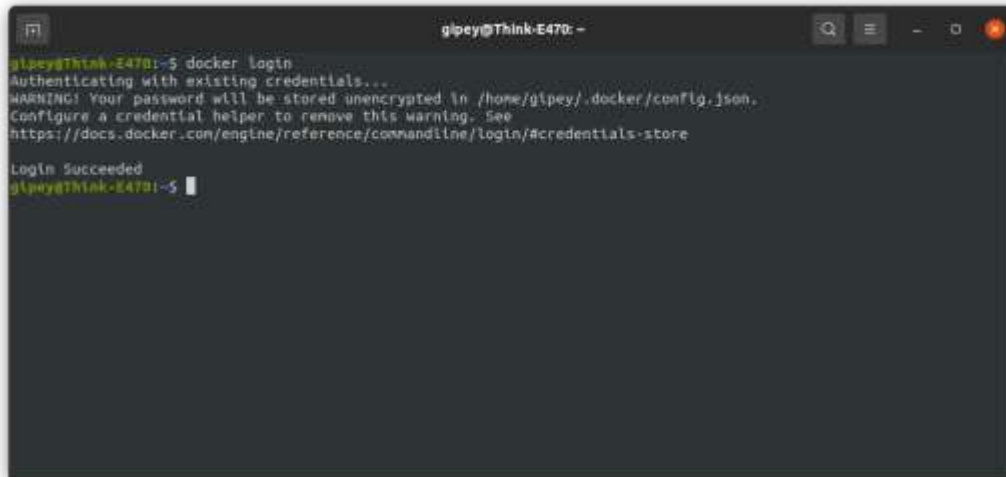


- Berikut tampilan dari publik. Publik bisa pull/download repository sesuka hati ke Docker Host nya



Saatnya kita push suatu images ke Docker Hub. Buka CLI dan login ke Docker Hub

```
$ docker login
```

A terminal window titled 'gipey@Think-E470: -' showing the output of the 'docker login' command. The output indicates successful authentication with existing credentials, a warning about unencrypted passwords, and a link to Docker's documentation. The prompt returns to the user.

```
gipey@Think-E470:~$ docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/gipey/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
gipey@Think-E470:~$
```

Kita membuat images baru terlebih dahulu

```
$ docker tag nginx:latest gipeypey/ubuntu_gipey:cc-nginx
```

#### Catatan:

- gipeypey (nama akun Docker Hub)
- ubuntu\_gipey (nama repository)
- cc-nginx (custom tag/versi untuk images kita yang baru)

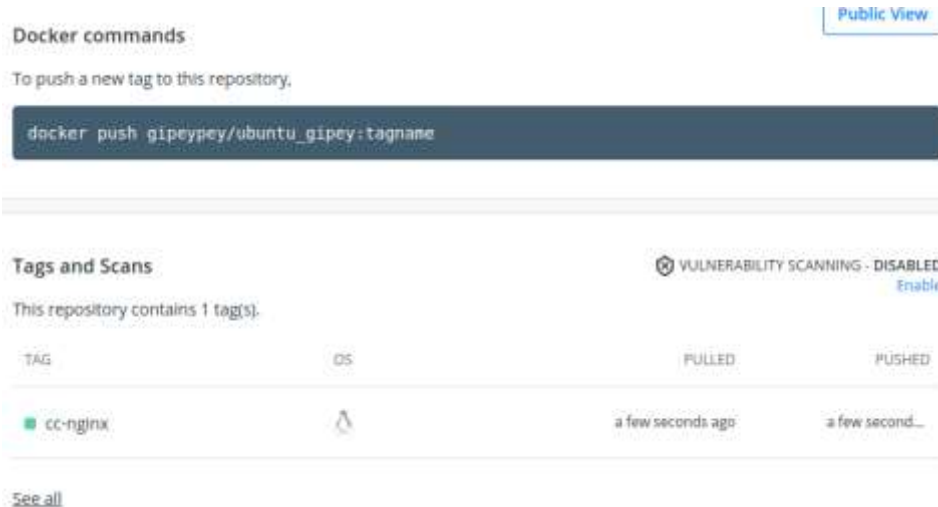
Push images ke Docker Hub

```
$ docker push gipeypey/ubuntu_gipey:cc-nginx
```

A terminal window titled 'gipey@Think-E470: -' showing the output of the 'docker push' command. It lists the images being pushed, their digests, and their sizes. The output shows that the images were successfully pushed to the repository 'docker.io/gipeypey/ubuntu\_gipey'.

```
gipey@Think-E470:~$ docker push gipeypey/ubuntu_gipey:cc-nginx
The push refers to repository [docker.io/gipeypey/ubuntu_gipey]
2ac782836f38: Mounted from library/nginx
9f65d1d4c869: Mounted from library/nginx
8f884d36244d: Mounted from library/nginx
9b23c8e1e0f9: Mounted from library/nginx
ffd36313c9b: Mounted from library/nginx
9eb02f04c782: Mounted from library/nginx
cc-nginx: digest: sha256:b88ecc9f7997452ef24358f3e43b9c66888fadbb31f3e5de22fec922975caa75a size: 1578
gipey@Think-E470:~$
```

Cek Docker Hub



## Inspect

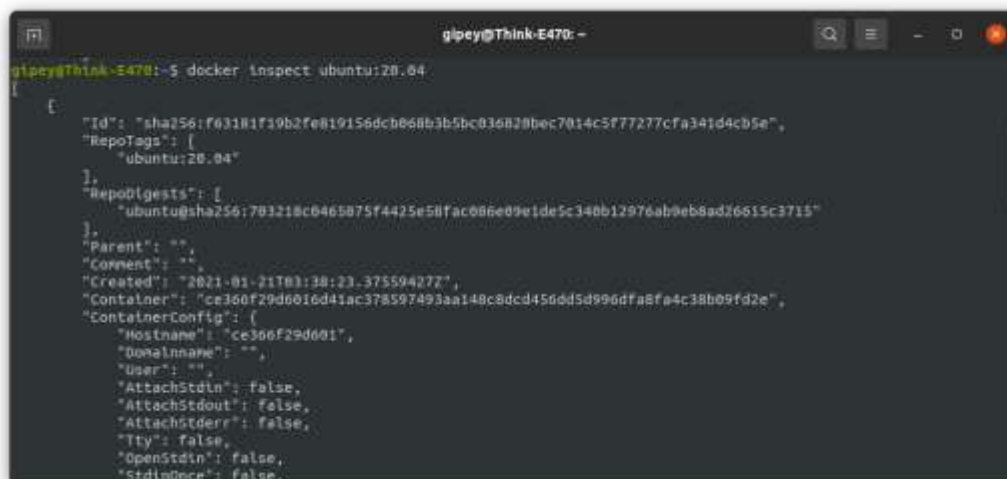
Command inspect akan memunculkan informasi yang detail dari setiap object dalam images yang terdaftar di Docker Engine milik kita. inspect akan lebih berguna kalau digunakan untuk debugging

Melihat images dengan nama yang spesifik

```
$ docker images ubuntu
```

Gunakan command inspect untuk melihat informasi detail dari image tersebut dalam bentuk JSON array

```
$ docker inspect ubuntu:20.04
```



## Catatan:

- id (image id dari ubuntu:latest)
- RepoTags (versi dari image ubuntu)
- RepoDigests (64 bit hex number)
- Container (container identifier)
- Container config (configuration detail tentang image tersebut)

- Architecture (arsitektur dari images tersebut)
- OS (ubuntu keluarga dari Linux)

Karena informasi tersebut sangat banyak, maka kita bisa filter dengan cara

```
$ docker inspect --format "{{.RepoTags}} : {{.RepoDigests}}"
ubuntu:20.04
```

```
gipey@Think-E470: -
gipey@Think-E470:~$ docker inspect --format "{{.RepoTags}} : {{.RepoDigests}}" ubuntu:20.04
[ubuntu:20.04] : [ubuntu@sha256:783218cd465875f4425e38fac08de09e1de5c346b12978ab9eb8ad20615c3715]
gipey@Think-E470:~$
```

Catatan:

`{{.RepoTags}}` (salah satu dari variable kunci dalam informasi images)

atau convert format tersebut menjadi sebuah file .txt dengan cara

```
$ docker inspect --format "{{json .Config}}" ubuntu >
inspect_report_ubuntu.txt
```

```
gipey@Think-E470: /mnt/01D6EB79A3B34E70/Networking/Docker/Section 2/Docker Essentials - 10. Architecture and Components
of Docker/CC_Docker/53/04$ ls
inspect_report_ubuntu.txt
gipey@Think-E470: /mnt/01D6EB79A3B34E70/Networking/Docker/Section 2/Docker Essentials - 10. Architecture and Components
of Docker/CC_Docker/53/04$ more inspect_report_ubuntu.txt
{"Hostname":"","Domainname":"","User":"","AttachStdin":false,"AttachStdout":false,"AttachStderr":false,"Tty":false,"Open
Stdin":false,"StdinOnce":false,"Env":["PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"],"Cmd":["/b
in/bash"],"Image":"sha256:87be921e3d9d70f133d5bd1e25324a6c4ab803dc3c1b7e3b29c7bdd873ba1","Volumes":{"WorkingDir":
""},"Entrypoint":null,"OnBuild":null,"Labels":null}
gipey@Think-E470: /mnt/01D6EB79A3B34E70/Networking/Docker/Section 2/Docker Essentials - 10. Architecture and Components
of Docker/CC_Docker/53/04$
```

Catatan:

- ubuntu (images yang sesuai di command docker images)
- inspect\_report\_ubuntu.txt (nama dan ekstensi file)

## History

docker history akan menampilkan log (created by) dalam sebuah images.

Kita coba menggunakan history dengan `img_apache` yang sudah kita install beberapa waktu lalu

```
$ docker image history img_apache
```

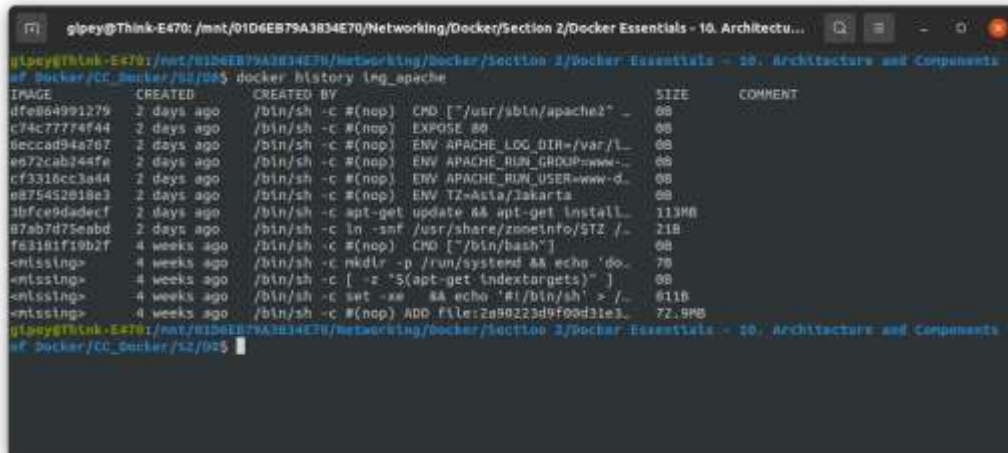


IMAGE	CREATED	CREATED BY	CMD	SIZE	COMMENT
dfe864991279	2 days ago	/bin/sh -c #(nop)	CMD ["/usr/sbin/apache2" -...	0B	
c74c7777f44	2 days ago	/bin/sh -c #(nop)	EXPOSE 80	0B	
6eccad94a767	2 days ago	/bin/sh -c #(nop)	ENV APACHE_LOG_DIR=/var/l...	0B	
w672cab344fe	2 days ago	/bin/sh -c #(nop)	ENV APACHE_RUN_GROUP=www-...	0B	
cf3318cc3a44	2 days ago	/bin/sh -c #(nop)	ENV APACHE_RUN_USER=www-d...	0B	
w875452018e3	2 days ago	/bin/sh -c #(nop)	ENV TZ=Asia/Jakarta	0B	
3bfce9d4decf	2 days ago	/bin/sh -c	apt-get update && apt-get install...	113MB	
87ab7d75eabd	2 days ago	/bin/sh -c ln -sfn	/usr/share/zoneinfo/STZ /...	21B	
f63191f19b2f	4 weeks ago	/bin/sh -c #(nop)	CMD ["/bin/bash"]	0B	
<missing>	4 weeks ago	/bin/sh -c rmdir -p	/run/systemd && echo 'do...	7B	
<missing>	4 weeks ago	/bin/sh -c [ -z '\$(apt-get indextargets)' ]		0B	
<missing>	4 weeks ago	/bin/sh -c set -xe && echo '#!/bin/sh' > /...		611B	
<missing>	4 weeks ago	/bin/sh -c #(nop)	ADD file:2a90223d9f00d31e...	72.9MB	

## Remove

Hapus image menggunakan `rm`

```
$ docker rm nginx:latest
```

Hapus image menggunakan `rmi` untuk image ID

```
$ docker rmi 35c43ace9216
```

nginx	1-alpine	336262580e12
nginx	alpine	336262580e12

*Jika kalian menggunakan `--all-tags` akan terhapus image yang memiliki ID yang sama*

## Docker Container

### Container Run V.S Create

Membuat container sekaligus terdapat images dengan nama custom untuk images busybox

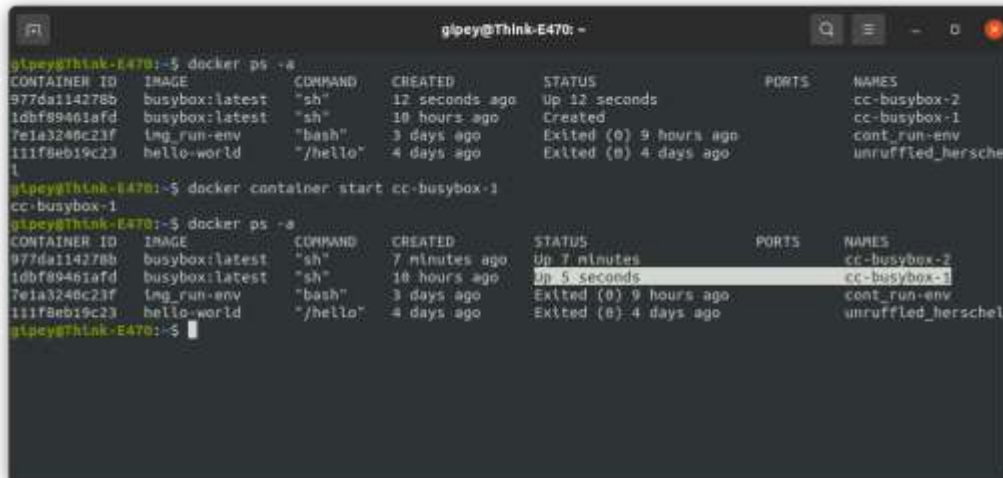
```
$ docker container create -it --name cc-busybox-1 busybox:latest
```



```
$ docker container start cc-busybox-1
```

Cek kembali

```
$ docker ps -a
```



The terminal window shows the output of the `docker ps -a` command. It lists four containers: `cc-busybox-2` (Up 12 seconds), `cc-busybox-1` (Created), `cont_run-env` (Exited 9 hours ago), and `unruffled_herschel` (Exited 4 days ago). Below this, the command `docker container start cc-busybox-1` is executed. A second `docker ps -a` command is then run, showing that `cc-busybox-1` is now 'Up 5 seconds' and `cc-busybox-2` is 'Up 7 minutes'.

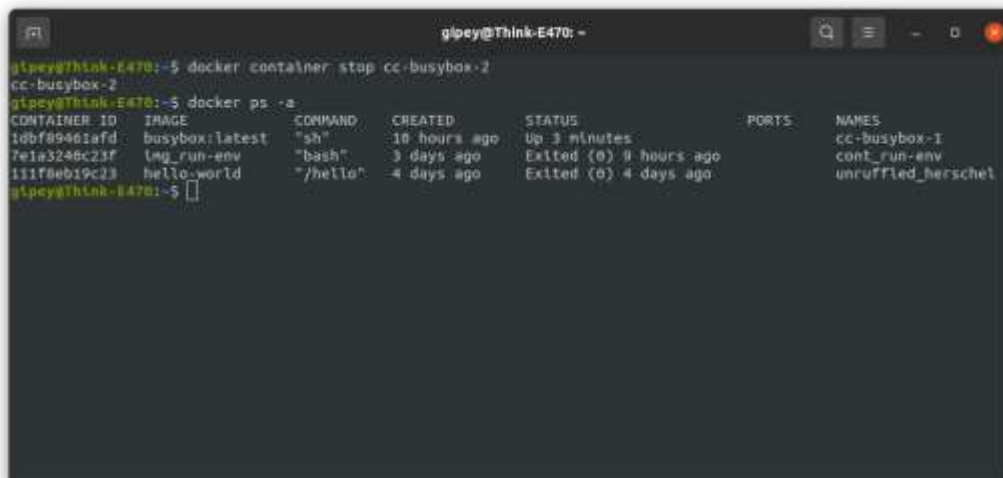
```
gipey@Think-E470: ~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
977da114278b   busybox:latest "sh"                    12 seconds ago Up 12 seconds          cc-busybox-2
1dbf89461afd   busybox:latest "sh"                    10 hours ago   Created                                cc-busybox-1
7e1a3246c23f   lmg_run-env    "bash"                  3 days ago     Exited (0) 9 hours ago          cont_run-env
111f8eb19c23   hello-world    "/hello"                4 days ago     Exited (0) 4 days ago          unruffled_herschel

gipey@Think-E470: ~$ docker container start cc-busybox-1
cc-busybox-1
gipey@Think-E470: ~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
977da114278b   busybox:latest "sh"                    7 minutes ago   Up 7 minutes          cc-busybox-2
1dbf89461afd   busybox:latest "sh"                    10 hours ago   Up 5 seconds          cc-busybox-1
7e1a3246c23f   lmg_run-env    "bash"                  3 days ago     Exited (0) 9 hours ago          cont_run-env
111f8eb19c23   hello-world    "/hello"                4 days ago     Exited (0) 4 days ago          unruffled_herschel
gipey@Think-E470: ~$
```

Stop

Menghentikan container cc-busybox-2

```
$ docker container stop cc-busybox-2
```



The terminal window shows the output of the `docker container stop cc-busybox-2` command. Below this, the `docker ps -a` command is run, showing that `cc-busybox-2` is now 'Exited (0) 3 minutes ago'. The other containers remain in their previous states.

```
gipey@Think-E470: ~$ docker container stop cc-busybox-2
cc-busybox-2
gipey@Think-E470: ~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
1dbf89461afd   busybox:latest "sh"                    10 hours ago   Exited (0) 3 minutes          cc-busybox-1
7e1a3246c23f   lmg_run-env    "bash"                  3 days ago     Exited (0) 9 hours ago          cont_run-env
111f8eb19c23   hello-world    "/hello"                4 days ago     Exited (0) 4 days ago          unruffled_herschel
gipey@Think-E470: ~$
```

Restart

Gunakan restart saat terjadi masalah di container tersebut

```
$ docker container restart cc-busybox-a
```

atau

```
$ docker container restart --time 5 cc-busybox-a
```

```
gipey@Think-E470: -
gipey@Think-E470:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
1dbf89461afd   busybox:latest "sh"                   10 hours ago   Up 10 minutes                cc-busybox-1
7e1a3240c23f   img_run-env   "bash"                 3 days ago    Exited (0) 10 hours ago      cont_run-env
111f8eb19c23   hello-world    "/hello"               4 days ago    Exited (0) 4 days ago       unruffled_herschel
gipey@Think-E470:~$ docker container restart cc-busybox-1
cc-busybox-1
gipey@Think-E470:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
1dbf89461afd   busybox:latest "sh"                   10 hours ago   Up 0 seconds                cc-busybox-1
7e1a3240c23f   img_run-env   "bash"                 3 days ago    Exited (0) 10 hours ago      cont_run-env
111f8eb19c23   hello-world    "/hello"               4 days ago    Exited (0) 4 days ago       unruffled_herschel
gipey@Think-E470:~$
```

## Rename

Kalo kalian pengen ubah nama container karena suatu hal, gunakan command dibawah ini  
\$ docker container rename **cc-busybox-1** **busybox1**

```
gipey@Think-E470: -
gipey@Think-E470:~$ docker container rename cc-busybox-1 busybox1
gipey@Think-E470:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
1dbf89461afd   busybox:latest "sh"                   10 hours ago   Up 8 minutes                busybox1
7e1a3240c23f   img_run-env   "bash"                 3 days ago    Exited (0) 10 hours ago      cont_run-env
111f8eb19c23   hello-world    "/hello"               4 days ago    Exited (0) 4 days ago       unruffled_herschel
gipey@Think-E470:~$
```

## Catatan:

- cc-busybox-1 (nama container yang saat ini)
- busybox1 (nama container yang baru)

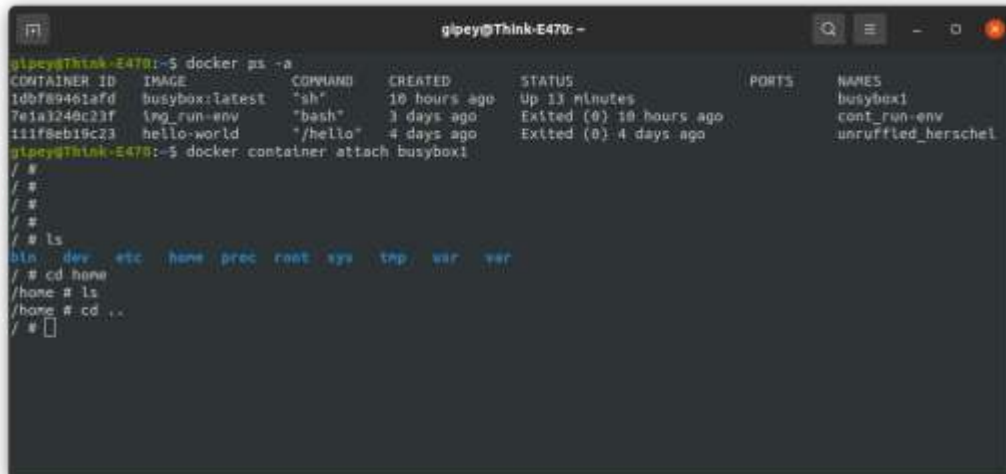


## Attach

Attach berguna untuk melihat standard I/O dan standard error di Docker client

Masuk ke environment pada container **busybox1**

\$ docker container attach **busybox1**



```
gipey@Think-E470: ~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
1dbf89461afd   busybox:latest "sh"                    10 hours ago  Up 11 minutes                busybox1
7e1a1246c23f   img_run-env    "bash"                  3 days ago    Exited (0) 10 hours ago      cont_run-env
111f8eb19c23   hello-world    "/hello"                4 days ago    Exited (0) 4 days ago        unruffled_herschel

gipey@Think-E470: ~$ docker container attach busybox1
/ #
/ #
/ #
/ #
/ # ls
bin  dev  etc  home  proc  root  sys  tmp  var  /
/ # cd /home
/home # ls
/home # cd ..
/ #
```

Saat environment ditutup menggunakan command **exit**, maka container tersebut akan berhenti process nya



```
gipey@Think-E470: ~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
1dbf89461afd   busybox:latest "sh"                    10 hours ago  Exited (0) 8 seconds ago      busybox1
7e1a1246c23f   img_run-env    "bash"                  3 days ago    Exited (0) 10 hours ago      cont_run-env
111f8eb19c23   hello-world    "/hello"                4 days ago    Exited (0) 4 days ago        unruffled_herschel

gipey@Think-E470: ~$
```

## Execute

Menggunakan command **exec** harus dalam keadaan container sedang running

\$ docker container start busybox1

**exec** berguna saat kita hanya mengeksekusi satu command saja dan ngga akan memberhentikan process dari container

\$ docker exec -it busybox1 pwd

```
gipey@Think-E470: -
gipey@Think-E470:~$ docker exec -it busybox1 pwd
/
gipey@Think-E470:~$
```

Catatan:

- pwd (salah satu command linux, kalian bisa menggunakan command lainnya)

## Port Mapping

Pada materi sebelumnya, kita sudah mencoba port mapping pada Expose Configuration. Port mapping bertujuan untuk mengubah port pada container.

### Port mapping pada Expose Configuration

```
$ docker container run -itd --name cont_nginx -p 8080:80/tcp
img_expose
```

```
gipey@Think-E470: -
gipey@Think-E470:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
bc1dc7f6597	img_expose	"nginx -g 'daemon of..."	12 seconds ago	Up 14 seconds	0.0.0.0:8080->80/tcp	cont_expose
1d0f09e1a1a2	busybox:latest	"sh"	12 hours ago	Exited (127) 9 hours ago		busybox1
7e1a328b11f	img_run-env	"bash"	1 days ago	Exited (0) 12 hours ago		cont_run-env
111fbeb19c23	hello-world	"/hello"	4 days ago	Exited (0) 4 days ago		unraffled_herschel

```
gipey@Think-E470:~$
```

Catatan:

- cont\_nginx (nama custom untuk container)
- -p 8080:80/tcp (port default container yaitu 8080 dan 80 adalah port untuk HTTP)
- Apa jadinya jika port mapping command hanya -P ? maka, docker akan memberikan random port container secara otomatis

Cara melihat port yang digunakan oleh container

```
$ docker container port cont_expose
80/tcp -> 0.0.0.0:8080
```

## Clean Up

Menghapus container dengan rm

```
$ docker container rm cont_expose cont_run-env hello-world busybox1
```

Atau

```
$ docker container rm belc6c7fe697
```

## Catatan:

- Saat menghapus container, harus dalam keadaan berhenti/stop process



```
gpcy@Think-E470:~$ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                    NAMES
belc6c7fe697   img_expose "nginx -g 'daemon of...'" 11 minutes ago Up 11 minutes      80.0.0.0:8080->80/tcp    cont_expose
bdf8b6e1a7e    busybox1   "ls"                    13 hours ago   Exited (137) 3 hours ago                               busybox1
7a13224c11f    img_run-env "bash"                 3 days ago    Exited (0) 13 hours ago                               cont_run-env
111f6b13c13    hello-world "/hello"                4 days ago    Exited (0) 4 days ago                               unruffled_herschel

gpcy@Think-E470:~$ docker container rm cont_expose
Error response from daemon: You cannot remove a running container belc6c7fe69727c8f73b5c73af5774ac78403874d1bc2b61dc13a783d1bdc88. Stop the container before attempting removal or force remove
gpcy@Think-E470:~$
```

Kalau masih mau menghapus container saat keadaan running/up gunakan --force

```
$ docker container rm cont_expose --force
```

Gunakan banner/notification saat container tersebut dihapus

```
$ docker container kill --signal=SIGTERM cont_expose
```

Menghapus seluruh container yang ngga terpakai (exited & dangling)

```
$ docker container prune
```



## LATIHAN

---

Diskusikan dengan rekan anda terkait persoalan apa saja yang dihadapi ketika proses praktikum dan ikuti instruksi dosen atau asisten praktikum.



## TUGAS

---

Dokumentasikan persoalan yang anda hadapi selama proses praktikum, sertakan screenshot dan pesan error yang ada, ikuti instruksi dosen atau asisten praktikum



## REFERENSI

---

Amazon Web Services. (2020). *Apa itu Docker ?* Amazon. Retrieved March 14, 2023, from <https://aws.amazon.com/id/docker/>

Poulton, N. (2017). *Docker deep dive*. Nigel Poulton.

Moilanen, M. (2018). Deploying an application using Docker and Kubernetes.

Miell, I., & Sayers, A. (2019). *Docker in practice*. Simon and Schuster.