

Nama : Putri Mauliza

NIM : 1608107010062

A. Your First Linux Container

1.0 Running your first container

```
[node1] (local) root@192.168.0.13 ~
$ docker container run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:4df8ca8a7e309c256d60d7971ea14c27672fc0d10c5f303856d7
bc48f8cc17ff
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working cor
rectly.

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 Docke
r Hub.
```

- docker container run hello-world -> pada saat docker dijalankan, ia akan mencari gambar bernama hello-world di lokal komputer. Jika tidak ada, maka akan dicari dan mendownload image Hello-World dari registry, secara default yang ada docker store.

1.1 Docker Images

```
[node1] (local) root@192.168.0.13 ~
docker image pull alpine
Using default tag: latest
latest: Pulling from library/alpine
89d9c30c1d48: Pull complete
Digest: sha256:c19173c5ada610a598915111163d28a67368362762534d8a812
1ce95cf2bd5a
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest
```

- Docker image pull alpine -> ketika perintah ini dijalankan maka secara otomatis akan mendownload image dari Docker registry dan menyimpan image di komputer lokal.

```

$ docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
alpine              latest             965ea09ff2eb       6 weeks
ago                5.55MB
hello-world        latest             fce289e99eb9       11 mont
hs ago             1.84kB
[node1] (local) root@192.168.0.13 ~
$

```

- Docker image ls -> berfungsi untuk menampilkan image yang ada sudah terdownload

1.1 Docker Container Run

```

$ docker container run alpine ls -l
total 8
drwxr-xr-x  2 root  root    4096 Oct 21 13:39 bin
drwxr-xr-x  5 root  root    340  Dec  4 08:12 dev
drwxr-xr-x  1 root  root     66  Dec  4 08:12 etc
drwxr-xr-x  2 root  root     6  Oct 21 13:39 home
drwxr-xr-x  5 root  root   185  Oct 21 13:39 lib
drwxr-xr-x  5 root  root    44  Oct 21 13:39 media
drwxr-xr-x  2 root  root     6  Oct 21 13:39 mnt
drwxr-xr-x  2 root  root     6  Oct 21 13:39 opt
dr-xr-xr-x 978 root  root     0  Dec  4 08:12 proc
drwx----- 2 root  root     6  Oct 21 13:39 root
drwxr-xr-x  2 root  root     6  Oct 21 13:39 run
drwxr-xr-x  2 root  root  4096  Oct 21 13:39 sbin
drwxr-xr-x  2 root  root     6  Oct 21 13:39 srv
dr-xr-xr-x 13 root  root     0  Dec  4 08:12 sys
drwxrwxrwt  2 root  root     6  Oct 21 13:39 tmp

```

- Docker container run alpine ls -l -> pada saat perintah ini dijalankan, docker akan menjalankan image Alpine yang sudah dan mengakses perintah ls -l pada Alpine. Setelah menghasilkan output berupa list direktori pada container, docker akan mematikan container yang sedang dijalankan tadi.

```

[node1] (local) root@192.168.0.13 ~
$ docker container run alpine echo "hello from alpine"
hello from alpine

```

- Docker Fontaine run alpine echo "hello from alpine" -> perintah ini digunakan untuk membuat dan menampilkan nama file yang sudah dibuat yaitu hello from alpine.

```

[node1] (local) root@192.168.0.13 ~
$ docker container run alpine /bin/sh
[node1] (local) root@192.168.0.13 ~
$

```

- Docker container alpine /bin/sh -> perintah /bin/ sh akan menjalankan shell pada alpine.

```
$ docker container run -it alpine /bin/sh
/ #
```

- Perintah docker container run -it alpine /bin/sh -> untuk membuat container menjadi interaktif.

```
[node1] (local) root@192.168.0.13 ~
$ docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
[node1] (local) root@192.168.0.13 ~
```

- Docker container ls -> fungsinya untuk melihat container yang sedang berjalan

```
[node1] (local) root@192.168.0.13 ~
$ docker container ls -a
CONTAINER ID        IMAGE               COMMAND             CR
EATED              STATUS              PORTS              NAMES
66958b741787       alpine             "/bin/sh"         10
minutes ago       Exited (127) About a minute ago
sharp_hopper
1f11b1175cb6       alpine             "/bin/sh"         19
minutes ago       Exited (0) 19 minutes ago
mystifying_bohr
b88227498a36       alpine             "/bin/hs"         19
minutes ago       Created
practical_mestorf
b9973cc101c0       alpine             "echo 'hello from al..." 24
minutes ago       Exited (0) 24 minutes ago
dazzling_wright
94ac9ba99a84       alpine             "ls -l"           33
minutes ago       Exited (0) 33 minutes ago
laughing_snyder
```

- Docker container ls -a -> fungsinya untuk mengetahui apa saja yang telah dilakukan dengan container.

1.2 Container Isolation

```
[node1] (local) root@192.168.0.13 ~
$ docker container run -it alpine /bin/ash
/ #
```

- docker container run -it alpine /bin/ash -> merupakan salah satu jenis shell lain yang tersedia pada alpine image

```
[node1] (local) root@192.168.0.13 ~
$ docker container run -it alpine /bin/ash
/ # echo "hello world"> hello.txt
/ # ls
bin      hello.txt  media     proc      sbin      tmp
dev      home       mnt       root      srv       usr
etc      lib        opt       run       sys       var
/ #
```

- Perintah diatas merupakan perintah untuk membuat file dengan nama hello.txt, kemudian menampilkan lis file yang telah dibuat.

```

$ docker container run alpine ls
bin
dev
etc
home
lib
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
[nodel1] (local) root@192.168.0.13 ~
$

```

- Docker container run alpine ls -> saat perintah tersebut dijalankan, akan ditampilkan daftar-daftar alpine.

```

[nodel1] (local) root@192.168.0.13 ~
$ docker container ls -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
fab6d00d985a       alpine             "ls"                36 seconds ago     Exited (0)          36 seconds ago    angry_davinci
bd99082602cc       alpine             "/bin/ash"          4 minutes ago      Exited (0)          About a minute ago sharp_hopper
66958b741787       alpine             "/bin/sh"           9 minutes ago      Exited (127)        9 minutes ago     bold_mendel
1f11b1175cb6       alpine             "/bin/sh"           27 minutes ago     Exited (0)          27 minutes ago    mystifying_bohr
b88227498a36       alpine             "/bin/hs"           28 minutes ago     Exited (0)          28 minutes ago    sharp_hopper

```

- Docker container ls -a -> saat perintah ini dijalankan maka akan ditampilkan informasi container di, image yang dijalankan, perintah yang sudah dijalankan, waktu dijalankan dan nama container.

```

[nodel1] (local) root@192.168.0.13 ~
$ docker container start bd99082602cc
bd99082602cc

```

- Perintah diatas merupakan perintah untuk mengakses dan menjalankan ulang container dengan ID bd99082602cc.

```
[node1] (local) root@192.168.0.13 ~
$ docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
bd99082602cc       alpine             "/bin/ash"         22 minu
tes ago           Up 11 seconds     bold_mendel
```

- Docker container ls -> perintah ini berfungsi untuk melihat container yang sedang berjalan.

```
[node1] (local) root@192.168.0.13 ~
$ docker container exec bd99082602cc ls
bin
dev
etc
hello.txt
home
lib
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
```

- Perintah diatas menampilkan daftar direktori dan mengakses container yang sedang berjalan.

A. Customizing Docker Images

```
[node1] (local) root@192.168.0.48 ~
$ docker container run -ti ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
7ddbc47eeb70: Pull complete
c1bbdc448b72: Pull complete
8c3b70e39044: Pull complete
45d437916d57: Pull complete
Digest: sha256:6e9f67fa63b0323e9a1e587fd71c561ba48a034504fb804fd26fd8800039835d
Status: Downloaded newer image for ubuntu:latest
root@698f00431aac:/#
```

- docker container run -ti ubuntu bah -> perintah diatas merupakan perintah untuk memindahkan Insurance, karena image buntu belum ada dalam lokal maka secara otomatis akan di ambil dari library/buntu.

```
[node1] (local) root@192.168.0.48 ~
$ docker container ls -a
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
698f00431aac       ubuntu            "bash"             7 minut
es ago           Exited (0) 11 seconds ago      thirst
y darwin
```

- Perintah diatas berfungsi untuk menampilkan image yang ada pada doker dan menampilkan informasi yang sedang di jalankan.

```
[node1] (local) root@192.168.0.48 ~
$ docker container commit 698f00431aac
sha256:ba0b7fbbec66e96b542cdd49d8b4fd49e2fdb41b7bf35f6bbb0611e648ec
01ad
```

- Perintah diatas berfungsi untuk commit image dengan ID 698f00431aac

```
$ docker image ls
REPOSITORY          TAG                IMAGE ID           CREATED
SIZE
<none>              <none>            ba0b7fbbec66      3 minut
es ago             93.3MB
ubuntu              latest            775349758637      4 weeks
ago                64.2MB
[node1] (local) root@192.168.0.48 ~
$ docker image tag ba0b7fbbec66 ourfiglet
```

- Setelah di commit kemudian ditambah gambar baru.

```
$ docker container run ourfiglet figlet hello

 _
| |__  _  | | | |__
| ' \ / _ \ | | / _ \
| | | | _ / | | ( ) |
|_| | | \___|_| | \___/

[node1] (local) root@192.168.0.48 ~
$
```

- Perintah diatas untuk menampilkan output hello

Image Creation Using a Dockerfile

B. Deploy and Managing Multiple Containers

```
[node1] (local) root@192.168.0.12 ~
$ docker swarm init --advertise-addr $(hostname -i)
Swarm initialized: current node (pkg1goinwsdq97ef11c5o8puc) is now
a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-5ayn9fbrm2e47r20swul4n54kx8
70e28cyl757etu7dg49pex-b7pccrpy12cvelmhcmy8uzpwy-192.168.0.12:2377
```

- Create manager, docker dapat bekerja dengan dua mode , pada stage ini mode yang berjalan **Docker Swarm Mode** , swarm mode dapat berjalan pada single atau pun multi node, pada tutorial berjalan pada single node. Swarm mode adalah mode docker yang berjalan dengan sistem cluster.

```
[node1] (local) root@192.168.0.12 ~
$ docker node ls
ID                                HOSTNAME          STATUS
AVAILABILITY                      MANAGER STATUS   ENGINE VERSION
pkg1lgoinwsdq97ef11c5o8puc *    node1            Ready
Active                             Leader           19.03.4
```

- Perintah diatas menampilkan output list dari node yang ada, pada perintah diatas terdapat output Leader untuk menandakan node mana yang bertindak sebagai node manager, dan sisanya adalah node worker.

```
[node1] (local) root@192.168.0.12 ~
$ git clone https://github.com/docker/example-voting-app
Cloning into 'example-voting-app'...
remote: Enumerating objects: 832, done.
remote: Total 832 (delta 0), reused 0 (delta 0), pack-reused 832
Receiving objects: 100% (832/832), 950.35 KiB | 14.85 MiB/s, done.
Resolving deltas: 100% (293/293), done.
[node1] (local) root@192.168.0.12 ~
```

- Clone voting App dengan menggunakan github

```
[node1] (local) root@192.168.0.12 ~/example-voting-app
$ cat docker-stack.yml
version: "3"
services:

  redis:
    image: redis:alpine
    networks:
      - frontend
```

- File docker-stack.yml mendefinisikan seluruh tumpukan arsitektur layanan, jumlah instance, bagaimana semuanya disatukan, bagaimana menangani pembaruan untuk setiap layanan. Ini adalah kode sumber untuk desain aplikasi kita.

```
[node1] (local) root@192.168.0.12 ~/example-voting-app
$ docker stack deploy --compose-file=docker-stack.yml voting_stack
Creating network voting_stack_frontend
Creating network voting_stack_default
Creating network voting_stack_backend
Creating service voting_stack_result
Creating service voting_stack_worker
Creating service voting_stack_visualizer
```

```

$ docker stack ls
NAME                SERVICES            ORCHESTRATOR
voting_stack        6                   Swarm
[nod1] (local) root@192.168.0.12 ~/example-voting-app
$

```

- Perintah diatas untuk melihat lis stack atau tumpukan

```

ORTS
6k9rlwxldrog       voting_stack_visualizer   replicated
/1                 dockersamples/visualizer:stable
:8080->8080/tcp
6o5b9iq9vqtq       voting_stack_vote         replicated
/2                 dockersamples/examplevotingapp_vote:before
:5000->80/tcp
cp28tvizhhcf       voting_stack_worker       replicated
/1                 dockersamples/examplevotingapp_worker:latest

```

- Perintah diatas menampilkan detail pada setiap Service dengan stack.

```

[nod1] (local) root@192.168.0.12 ~/example-voting-app
$ docker service ps voting_stack_vote
ID                NAME                IMAGE
NODE              DESIRED STATE      CURRENT STATE      PORTS
STATE             ERROR               PORTS
10dm05r3keot     voting_stack_vote.1 dockersamples/examplevotingapp_vote:before
node1             Running            Running            3 minutes ago

```

Scaling The Application

```

[nod1] (local) root@192.168.0.12 ~/example-voting-app
$ docker service scale voting_stack_vote=5
voting_stack_vote scaled to 5
overall progress: 5 out of 5 tasks
1/5: running
2/5: running
3/5: running
4/5: running

```

- Perintah diatas untuk mengukur layanan docker yang sedang bekerja