

Dokumentasi Deployment Aplikasi Web Profil Pribadi

Nama: Taukhid Aji Nurwijayadi

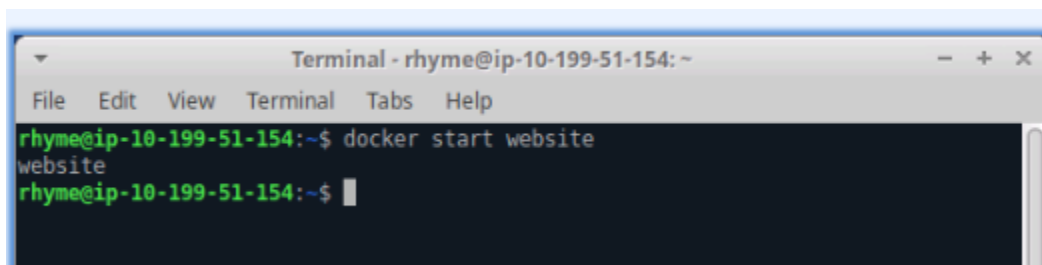
Media Pembelajaran: Coursera

Teknologi: html, css & js

Tool: Docker

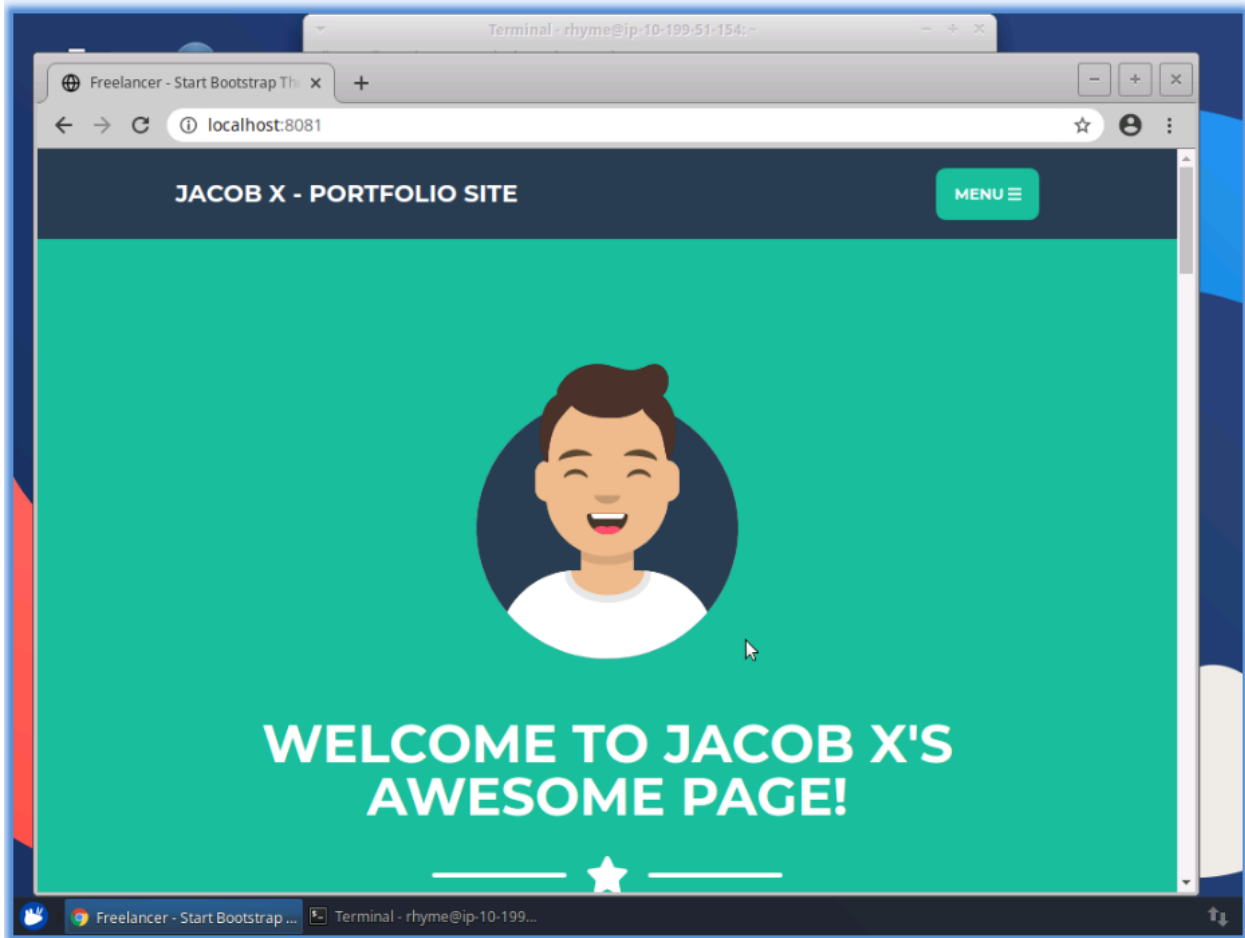
Server: Nginx

No telp/wa: 089656889236

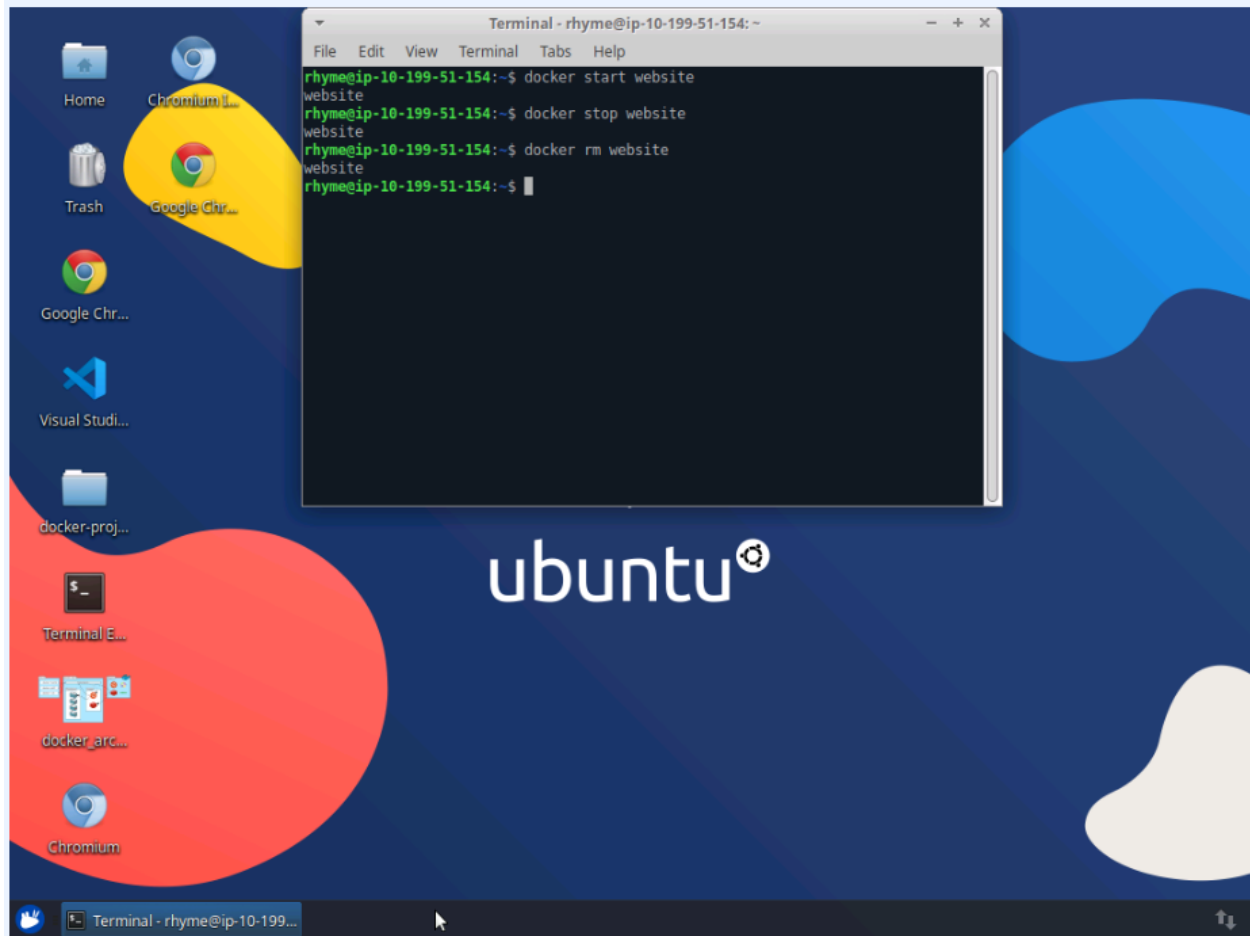
A screenshot of a terminal window titled "Terminal - rhyme@ip-10-199-51-154: ~". The terminal has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The prompt is "rhyme@ip-10-199-51-154:~\$". The command "docker start website" has been entered, and the output "website" is displayed on the next line. The prompt "rhyme@ip-10-199-51-154:~\$" is shown again with a cursor at the end.

```
rhyme@ip-10-199-51-154:~$ docker start website
website
rhyme@ip-10-199-51-154:~$
```

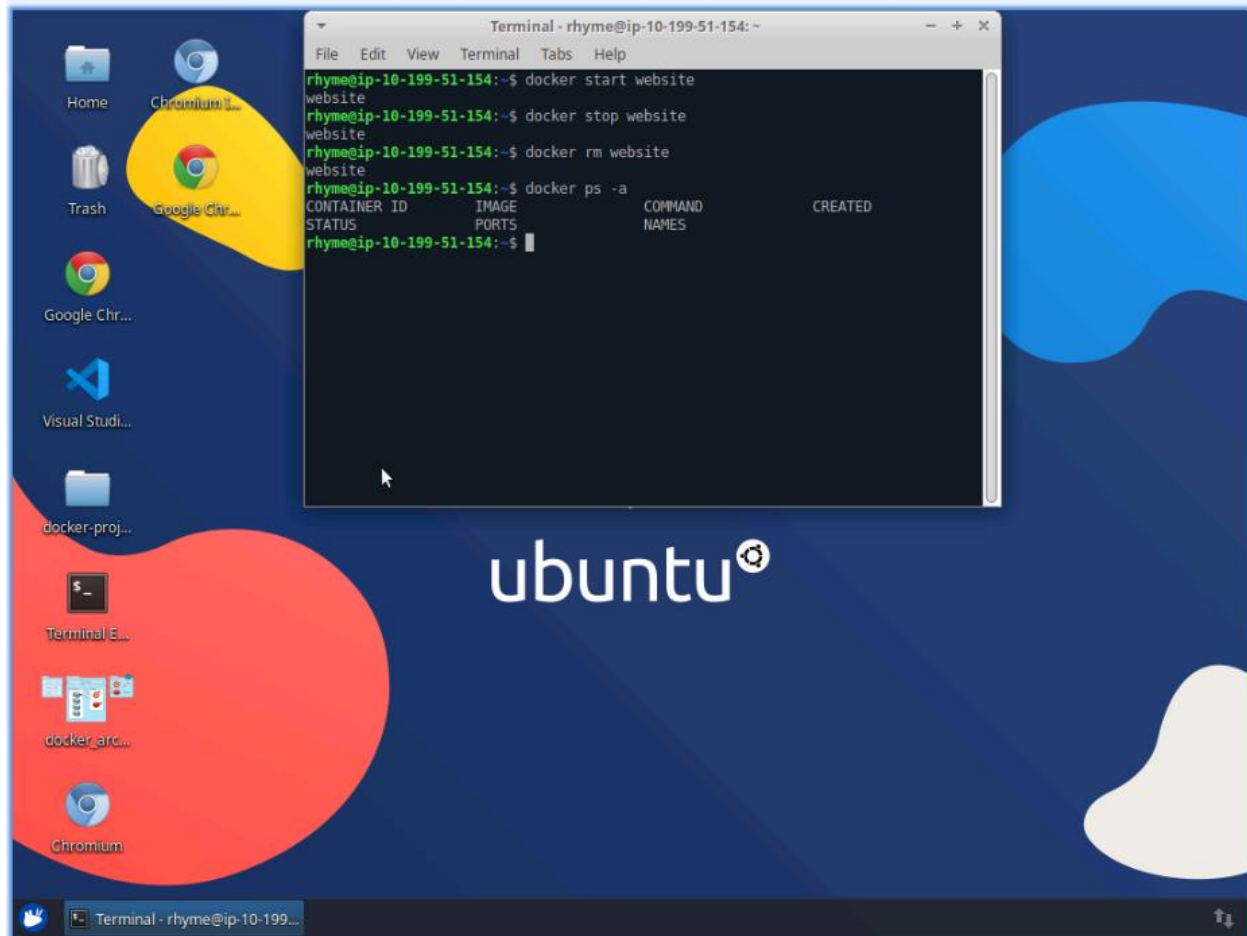
Start Dockernya terlebih dahulu dengan mengetikan “**docker start nama_container**”



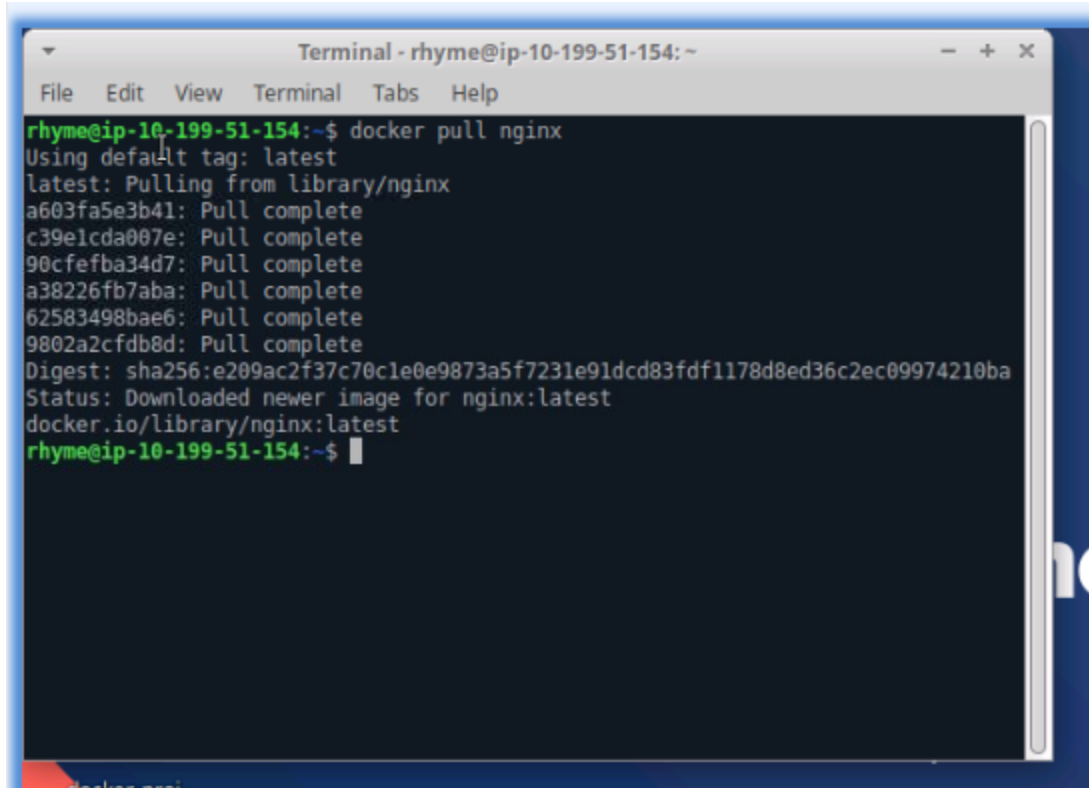
Setelah itu masuk chrome dan ketikkan “**localhost:8081**”, maka aplikasi web akan tampil



Setelah itu hentikan dan hapus aplikasi websitenya dengan mengetikan “**docker rm nama_container**”



Setelah itu lakukan lakukan “**docker ps -a**” yang berarti untuk mengecek proses status untuk semua list Container yang tersedia

A screenshot of a terminal window titled "Terminal - rhyme@ip-10-199-51-154: ~". The window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The terminal content shows the command "docker pull nginx" being executed. The output indicates that the latest tag is used, the image is pulled from the library/nginx repository, and several layers are pulled successfully. The digest is shown as sha256:e209ac2f37c70c1e0e9873a5f7231e91dcd83fdf1178d8ed36c2ec09974210ba. The status is "Downloaded newer image for nginx:latest" and the source is "docker.io/library/nginx:latest". The prompt "rhyme@ip-10-199-51-154:~\$" is shown at the bottom.

```
rhyme@ip-10-199-51-154:~$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
a603fa5e3b41: Pull complete
c39e1cda007e: Pull complete
90cfefba34d7: Pull complete
a38226fb7aba: Pull complete
62583498bae6: Pull complete
9802a2cfdb8d: Pull complete
Digest: sha256:e209ac2f37c70c1e0e9873a5f7231e91dcd83fdf1178d8ed36c2ec09974210ba
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
rhyme@ip-10-199-51-154:~$
```

Lalu lakukan “**docker pull nginx**”

```
Terminal - rhyme@ip-10-199-51-154:~  
File Edit View Terminal Tabs Help  
rhyme@ip-10-199-51-154:~$ docker pull nginx  
Using default tag: latest  
latest: Pulling from library/nginx  
a603fa5e3b41: Pull complete  
c39e1cda007e: Pull complete  
90cfefba34d7: Pull complete  
a38226fb7aba: Pull complete  
62583498bae6: Pull complete  
9802a2cfdb8d: Pull complete  
Digest: sha256:e209ac2f37c70c1e0e9873a5f7231e91dcd83fdf1178d8ed36c2ec09974210ba  
Status: Downloaded newer image for nginx:latest  
docker.io/library/nginx:latest  
rhyme@ip-10-199-51-154:~$ docker images  
REPOSITORY          TAG                 IMAGE ID            CREATED  
SIZE  
nginx                latest             88736fe82739       2 days ago  
142MB  
nginx                <none>             2622e6cca7eb       2 years ago  
132MB  
hello-world          latest             bf756fb1ae65       2 years ago  
13.3kB  
rhyme@ip-10-199-51-154:~$
```

Setelah itu lakukan perintah “**docker images**” untuk mengecek environment docker


```
Terminal - rhyme@ip-10-199-51-154: ~
rhyme@ip-10-199-51-154: ~
rhyme@ip-10-199-51-154:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
486c4bd98baa        nginx              "/docker-entrypoint..." 3 minutes ago       Up 3 minutes       80/tcp             hopeful_stonebrake
rhyme@ip-10-199-51-154:~$
```

Setelah itu buka tab baru dan jalankan perintah “**docker ps**”, untuk mengecek proses yang berjalan saat ini. Maka kita akan tahu berbagai informasi proses yang sedang berjalan apa saja. Dan berbagaimacam keterangannya.

```
Terminal - rhyme@ip-10-199-51-154: ~
rhyme@ip-10-199-51-154: ~
rhyme@ip-10-199-51-154:~$ docker run -d nginx
163016b364b074b10e285f4dea1123badd6d7657dd1e6471da19ae190df58e1e
rhyme@ip-10-199-51-154:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
163016b364b0        nginx              "/docker-entrypoint..." 3 minutes ago       Up 3 minutes       80/tcp             condescending_cartwright
```

Setelah itu pindah tab lagi ke tab awal, dan lakukan “**clear**” untuk membersihkan terminal. Lalu jalan docker di background dengan mengetikan “**docker run -d nginx**”. Lalu ketikan “**docker ps**” untuk melihat proses yang barusan kita jalankan. Dapat dibandingkan dengan proses sebelumnya yangmana bsia dicek dari ContainerID-nya saja sudah berbeda.

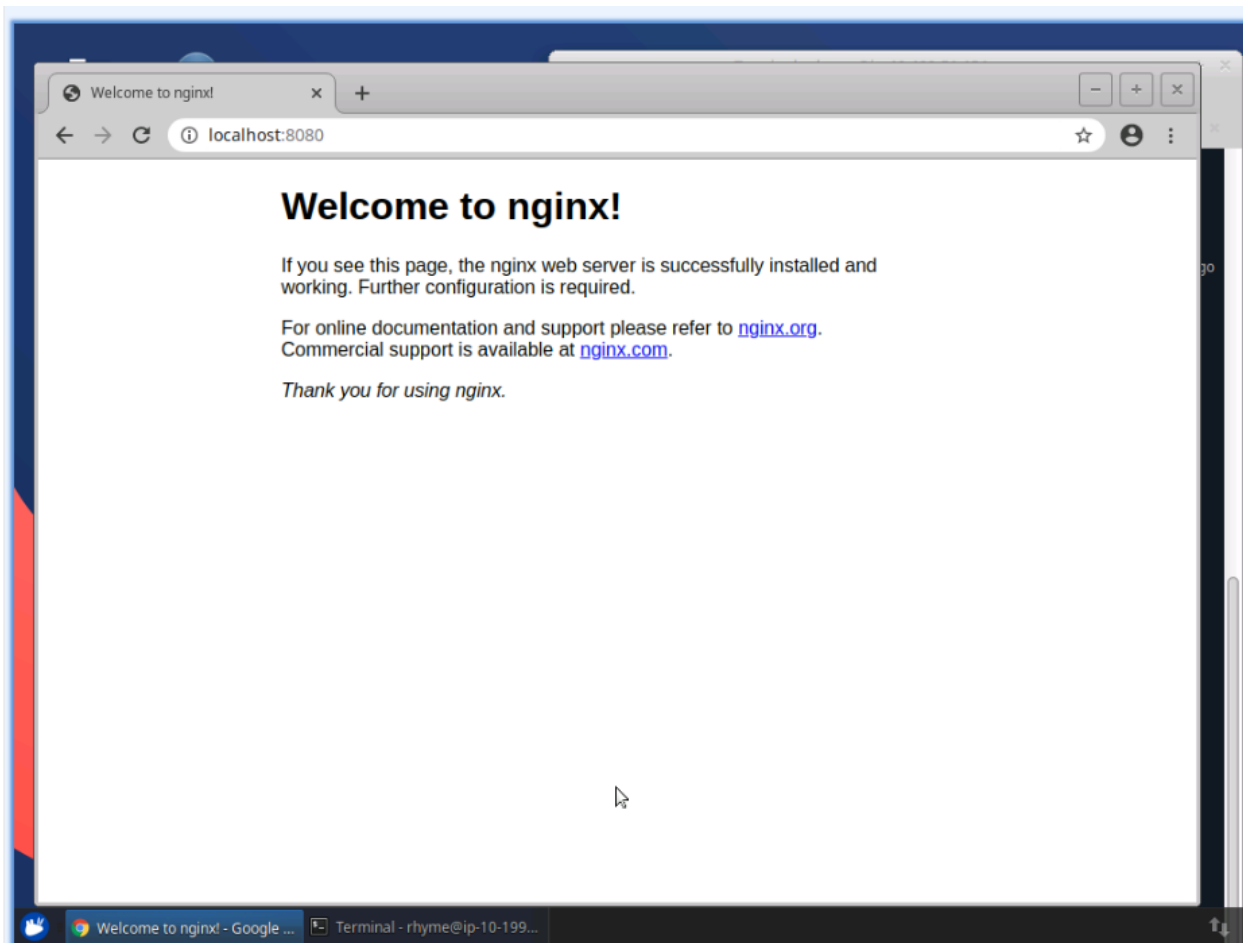
```
rhyme@ip-10-199-51-154:~$ docker exec -
--detach --env --interactive --tty --workdir -e -t -w
--detach-keys --help --privileged --user -d -i -u
rhyme@ip-10-199-51-154:~$ docker exec -
--detach --env --interactive --tty --workdir -e -t -w
--detach-keys --help --privileged --user -d -i -u
rhyme@ip-10-199-51-154:~$ docker exec -it 163016b364b0 bash
root@163016b364b0:/# ls
bin dev docker-entrypoint.sh home lib64 mnt proc run srv tmp var
boot docker-entrypoint.d etc lib media opt root sbin sys usr
root@163016b364b0:/# ^C
root@163016b364b0:/# ^C
root@163016b364b0:/# exit
exit
rhyme@ip-10-199-51-154:~$
```

Setelah itu kita masuk ke container dengan mengcopy bagian ContainerID dari “**docker ps**” tadi. Setelah itu jalankan perintah “**docker exec -it 163016b364b0 bash**”, maka anda akan masuk kedalam container. Setelah itu anda bisa mengetikan “**ls**” untuk melihat isi dari container, setelah kita sudah mengetahui isi dari container maka ketikan “**exit**”.


```
Terminal - rhyme@ip-10-199-51-154: ~
File Edit View Terminal Tabs Help

rhyme@ip-10-199-51-154: ~
rhyme@ip-10-199-51-154:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
163016b364b0   nginx    "/docker-entrypoint..." 16 minutes ago Up 16 minutes  80/tcp         condescending_cartwright
rhyme@ip-10-199-51-154:~$ docker run -d -p 8080:80 nginx
a8989456b5666e576fe3f82f86c8fe705131e81489a395ac0db15232d6a1fd9d
rhyme@ip-10-199-51-154:~$
```

Setelah itu anda bisa mengecek port bisa mencoba untuk menjalankan nginx, tapi sebelumnya anda perlu mengkonfigurasi portnya terlebih dahulu, Maka dari itu anda perlu melakukan perintah “**docker ps**” setelah itu anda bisa mengecek port nginx. Lalu anda bisa menjalankan nginx server dengan menggunakan perintah “**docker run -d -p 8080:80 nginx**”. Sebelumnya maksud dari 8080:80 adalah mapping port dari port 80 nginx yang sedang berjalan ke port 8080.



Setelah itu anda bisa mengeceknya dnegan membuka chrome dan mengisikan di url dengan “**localhost:8080**” lalu tekan enter, maka akan tampil seperti pada gambar diatas.

```
Terminal - rhyme@ip-10-199-51-154: ~
File Edit View Terminal Tabs Help
rhyme@ip-10-199-51-154:~$ docker pull mongo
Using default tag: latest
latest: Pulling from library/mongo
eaead16dc43b: Pull complete
8a00eb9f68a0: Pull complete
f683956749c5: Pull complete
b33b2f05ea20: Pull complete
3a342bea915a: Pull complete
fa956ab1c2f0: Pull complete
138a8542a624: Pull complete
0a5a5d2ec822: Pull complete
37200fef7cf6: Pull complete
Digest: sha256:8bed0be3e86595283d67836e8d4f3f08916184ea6f2aac7440bda496083ab0c8
Status: Downloaded newer image for mongo:latest
docker.io/library/mongo:latest
rhyme@ip-10-199-51-154:~$
```

Setelah anda menginstall nginx server, maka Langkah selanjutnya anda perlu menginstall database. Disini Databasenya menggunakan Mongpo DB. Maka kita jalankan perintah “**docker pull mongo**”

```
docker.io/library/mongo:latest
rhyme@ip-10-199-51-154:~$ docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
mongo                latest          2dd27bb6d3e6   20 hours ago   695MB
nginx                latest          88736fe82739   2 days ago     142MB
nginx                <none>          2622e6cca7eb   2 years ago    132MB
hello-world          latest          bf756fb1ae65   2 years ago    13.3kB
rhyme@ip-10-199-51-154:~$ docker run -d 2dd27bb6d3e6
f64e94cb14bfb6baf68db1de7e1ca2f2051056c312463c580e31b9534a57cca
```

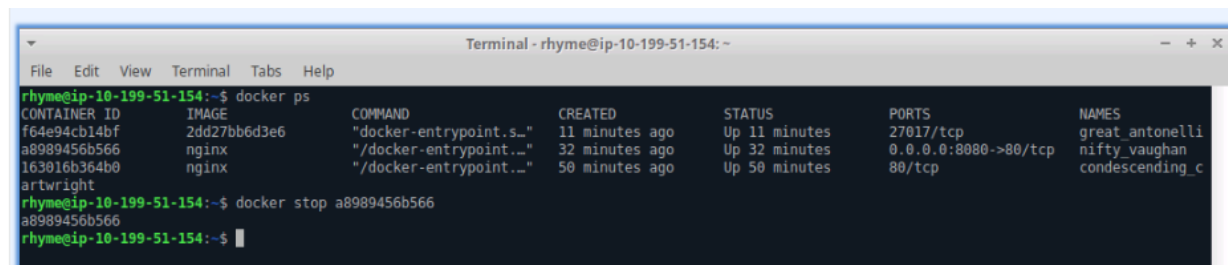
Setelah itu kita perlu menjalankan perintah “**docker images**” Maka kita bisa melihat bahwa mongo sudah terpasang. Anda bisa membandingkannya dengan report docker images sebelumnya. Lalu anda bisa langsung menjalankan mongo db dibackground dengan menggunakan perintah “**docker run -d image_id**”

```

exec export
rhyme@ip-10-199-51-154:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
f64e94cb14bf        2dd27bb6d3e6       "/docker-entrypoint.s..." 6 minutes ago       Up 6 minutes        27017/tcp           great_antonelli
a8989456b566        nginx              "/docker-entrypoint..." 27 minutes ago      Up 26 minutes        0.0.0.0:8080->80/tcp nifty_vaughan
163016b364b0        nginx              "/docker-entrypoint..." 44 minutes ago      Up 44 minutes        80/tcp              condensing_c
artwright
rhyme@ip-10-199-51-154:~$ docker exec -it f64e94cb14bf bash
root@f64e94cb14bf:/# ls
bin  data  docker-entrypoint-initdb.d  home  lib  lib64  media  opt  root  sbin  sys  usr
boot dev  etc                js-yaml.js  lib32  libx32  mnt    proc  run  srv  tmp  var
root@f64e94cb14bf:/#

```

Setelah itu anda bisa mengecek proses yang sedang berjalan dibackground dengan mengetikan perintah **“docker ps”**. Bisa dipastikan terlebih dahulu bahwa Image_id yang tadi anda copy sudah tampil didaftar proses yangn sedang berjalan, dan tampil dibagian kolom IMAGE. Setelah itu anda bisa cek isi dari container dengan mengetikan perintah **“docker exec -it container_id bash”**. Lalu ketika anda sudah memasuki bash , anda hanya perlu mengetikan perintah **“ls”** untuk mengetahui isi dari container mongo, apabila sudah selesai, anda hanya perlu mengetikan perintah **“exit”** untuk keluar dari container mongo



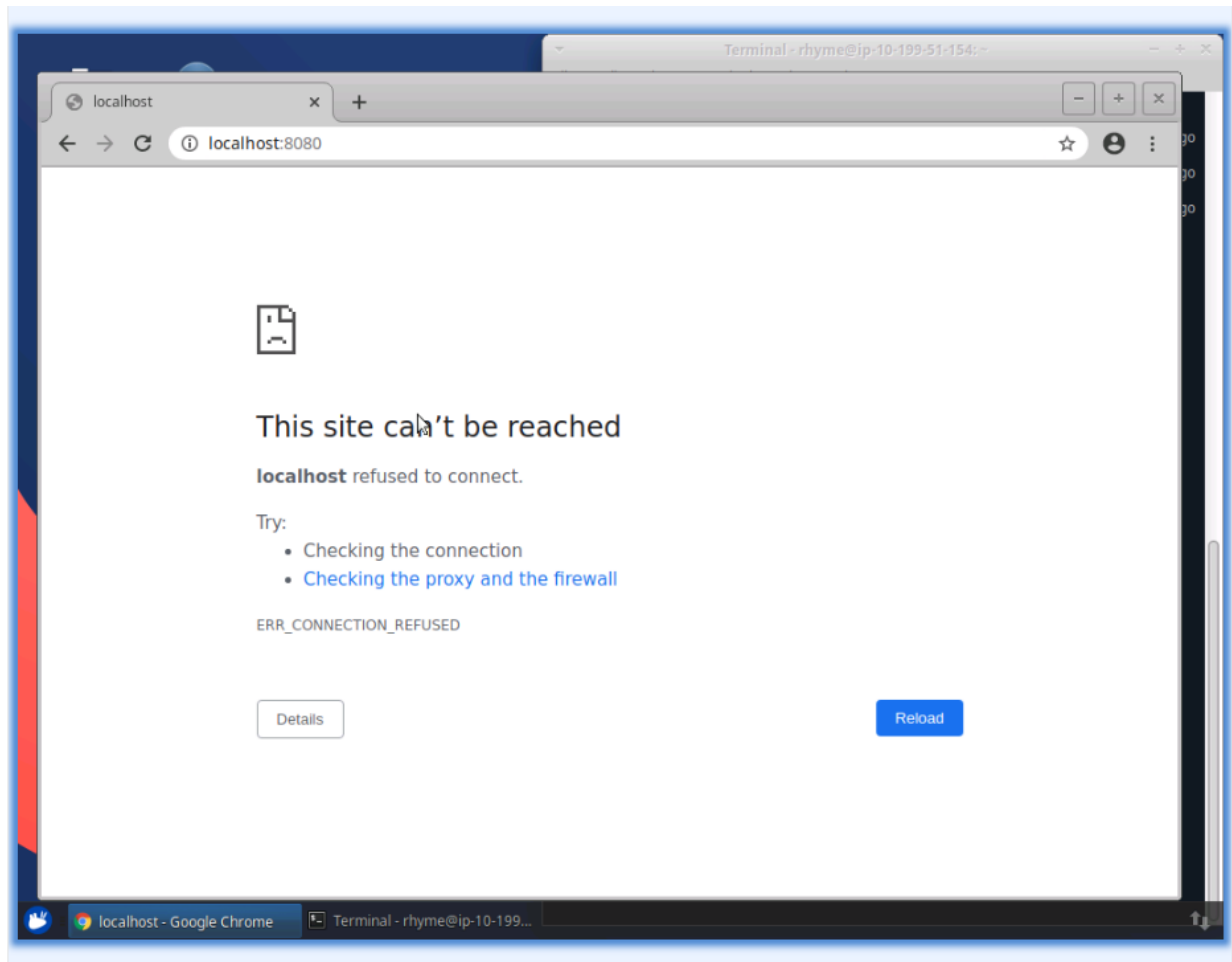
```

Terminal - rhyme@ip-10-199-51-154: ~
File Edit View Terminal Tabs Help
rhyme@ip-10-199-51-154:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
f64e94cb14bf        2dd27bb6d3e6       "/docker-entrypoint.s..." 11 minutes ago      Up 11 minutes        27017/tcp           great_antonelli
a8989456b566        nginx              "/docker-entrypoint..." 32 minutes ago      Up 32 minutes        0.0.0.0:8080->80/tcp nifty_vaughan
163016b364b0        nginx              "/docker-entrypoint..." 50 minutes ago      Up 50 minutes        80/tcp              condensing_c
artwright
rhyme@ip-10-199-51-154:~$ docker stop a8989456b566
a8989456b566
rhyme@ip-10-199-51-154:~$

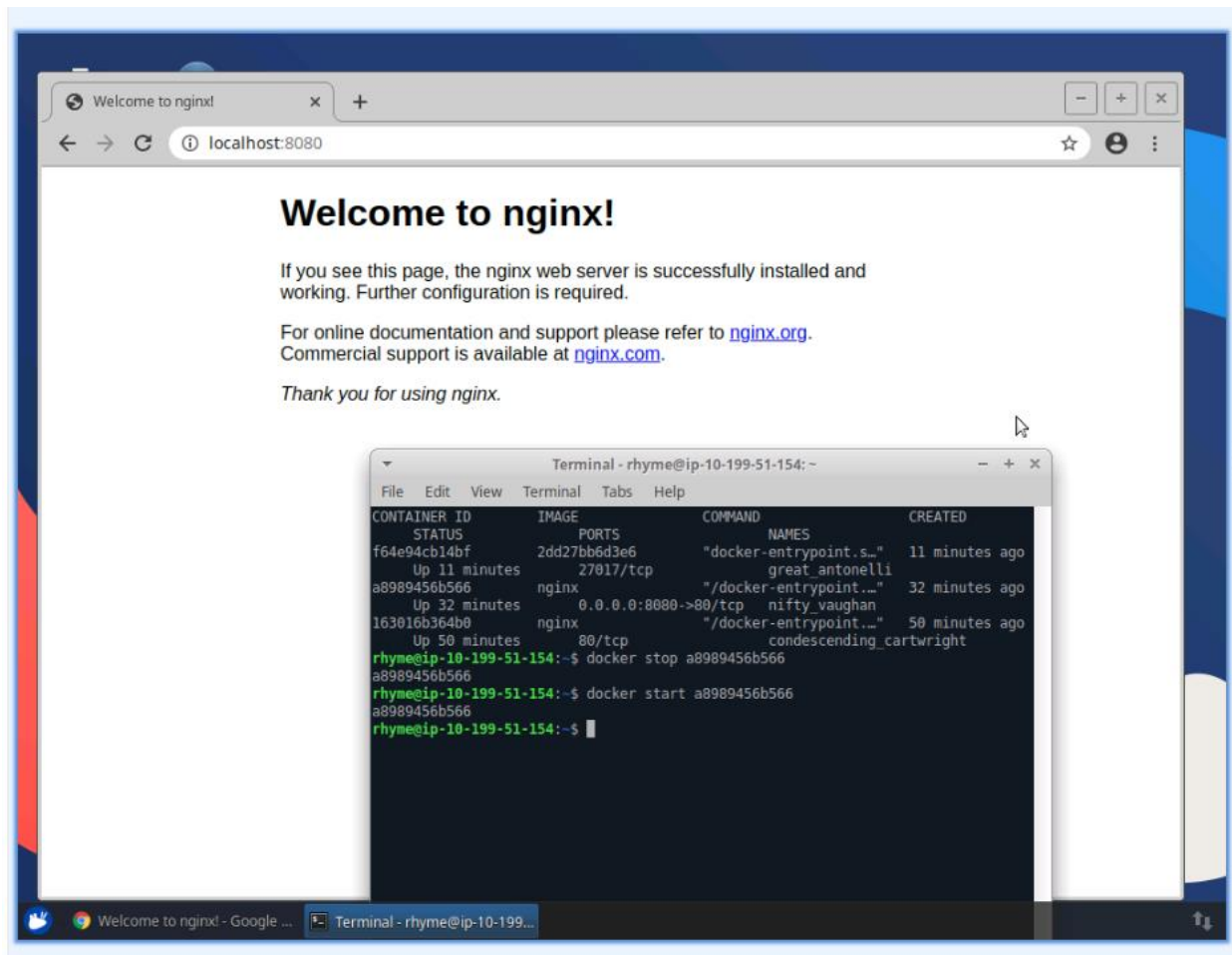
```

Langkah selanjutnya anda perlu menghentikan nginx, dengan mengetikan perintah **“docker ps”**, lalu anda perlu mencari nginx yang berjalan dengan menggunakan port yang dimapping dari 80 ke 8080.

Maka dari itu anda hanya perlu mengcopy container_id dari nginx tersebut, dan menjalankan perintah **“docker stop container_id”**



Setelah itu anda perlu membuka chrome lagi dan mererefresh halaman nginx yang tadi berjalan, maka akan tampil tampilan seperti digamabr diatas ini.



Lalu anda bisa menjalankan ulang lagi nginx dengan menjalankan perintah “**docker start container_id**” dan anda buka lagi chrome dan refresh halaman tersebut. Maka Nginx akan berjalan seperti tampilan pada gambar diatas.

```
Terminal - rhyme@ip-10-199-51-154: ~
File Edit View Terminal Tabs Help

rhyme@ip-10-199-51-154:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
f64e94cb14bf        2dd27bb6d3e6       "/docker-entrypoint.s..." 11 minutes ago      Up 11 minutes       27017/tcp           great_antonelli
a8989456b566        nginx              "/docker-entrypoint..." 32 minutes ago      Up 32 minutes       0.0.0.0:8080->80/tcp nifty_vaughan
163016b364b0        nginx              "/docker-entrypoint..." 50 minutes ago      Up 50 minutes       80/tcp              condensing_cartwright

rhyme@ip-10-199-51-154:~$ docker stop a8989456b566
a8989456b566

rhyme@ip-10-199-51-154:~$ docker start a8989456b566
a8989456b566

rhyme@ip-10-199-51-154:~$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAME
5                   S                  "/docker-entrypoint..." 22 minutes ago      Up 22 minutes       27017/tcp           grea
t_antonelli
a8989456b566        nginx              "/docker-entrypoint..." 42 minutes ago      Up 2 minutes       0.0.0.0:8080->80/tcp nift
y_vaughan
163016b364b0        nginx              "/docker-entrypoint..." About an hour ago    Up About an hour    80/tcp              cond
escending_cartwright
486c4bd98baa        nginx              "/docker-entrypoint..." About an hour ago    Exited (0) About an hour ago hope
ful_stonebraker

rhyme@ip-10-199-51-154:~$
```

Anda bisa cek status nginx dan mongo db yang berjalan dengan menjalankan perintah **“docker ps -a”**, maka anda bisa mengecek semua proses mana yang berjalan dibelakang layar anda.

```
ful_stonebraker
rhyme@ip-10-199-51-154:~$ docker rm a8989456b566
Error response from daemon: You cannot remove a running container a8989456b566e576fe3f82f86c8fe705131e81489a395ac0db15232d6a1fd9d. Stop the co
ntainer before attempting removal or force remove
rhyme@ip-10-199-51-154:~$
```

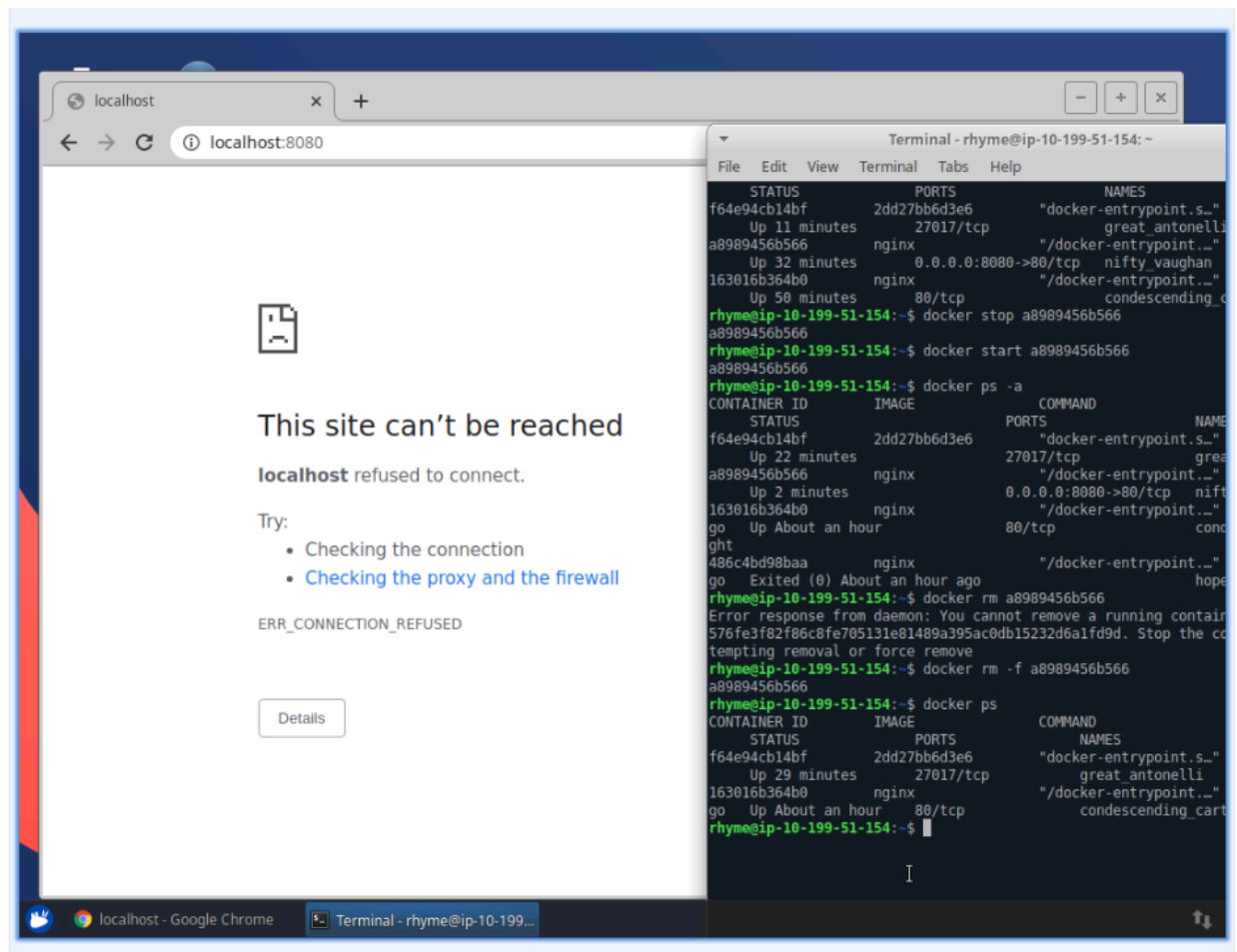
Disini saya langsung mencoba untuk menghapus proses nginx yang sedang berjalan, alhasil tidak bisa. Disana tertera bahwa anda harus menghentikan terlebih dahulu, lalu anda bisa menghapus ulang proses tersebut, atau anda bisa menggunakan alternatif force remove dengan menggunakan perintah **“docker rm -f container_id”**.

```

ntainer before attempting removal or force remove
rhyme@ip-10-199-51-154:~$ docker rm -f a8989456b566
a8989456b566
rhyme@ip-10-199-51-154:~$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED            STATUS              PORTS               NAMES
f64e94cb14bf        2dd27bb6d3e6       "/docker-entrypoint.s..." 29 minutes ago    Up 29 minutes      27017/tcp           great_antonelli
163016b364b0        nginx              "/docker-entrypoint.s..." About an hour ago  Up About an hour   80/tcp              condescending_cartwright
rhyme@ip-10-199-51-154:~$

```

Disini anda bisa melihat saya menggunakan force remove, maka anda hanya perlu memastikan apakah proses yang anda tuju sudah benar2 berhenti dengan menjalankan perintah **"docker ps"**. Disini anda bisa membandingkan dengan report docker ps sebelumnya, bahwa nginx yang terdapat port 80 yang dimapping ke 8080 sudah benar-benar berhenti.



Selanjutnya anda bisa mengecek di chrome apakah nginx masih berjalan atau sudah berhenti, apabila nginx berhenti maka akan menampilkan tampilan chrome seperti pada gambar diatas

```
rhyme@ip-10-199-51-154:~$ docker ps -aq
f64e94cb14bf
163016b364b0
486c4bd98baa
rhyme@ip-10-199-51-154:~$
```

Disini anda bisa mencoba perintah “**docker ps -aq**” untuk melihat semua proses yang sedang berjalan. Cuma hanya ditampilkan container_id nya saja.

```
rhyme@ip-10-199-51-154:~$ docker rm $(docker ps -aq)
486c4bd98baa
Error response from daemon: You cannot remove a running container f64e94cb14bf6baf68db1de7e1ca2f2051056c312463c580e31b9534a57cca. Stop the container before attempting removal or force remove
Error response from daemon: You cannot remove a running container 163016b364b074b10e285f4dea1123badd6d7657dd1e6471da19ae190df58e1e. Stop the container before attempting removal or force remove
rhyme@ip-10-199-51-154:~$
```

Setelah itu saya mencoba untuk menghapus semua proses yang berjalan dengan menggunakan perintah “**docker rm \$(docker ps -aq)**”. Maka akan didapati bahwa ada beberapa yang tidak bisa dihapus dan kasusnya sama dengan kasus sebelumnya. Dan untuk penanganannya hampir sama.

```
rhyme@ip-10-199-51-154:~$ docker rm -f $(docker ps -aq)
f64e94cb14bf
163016b364b0
```

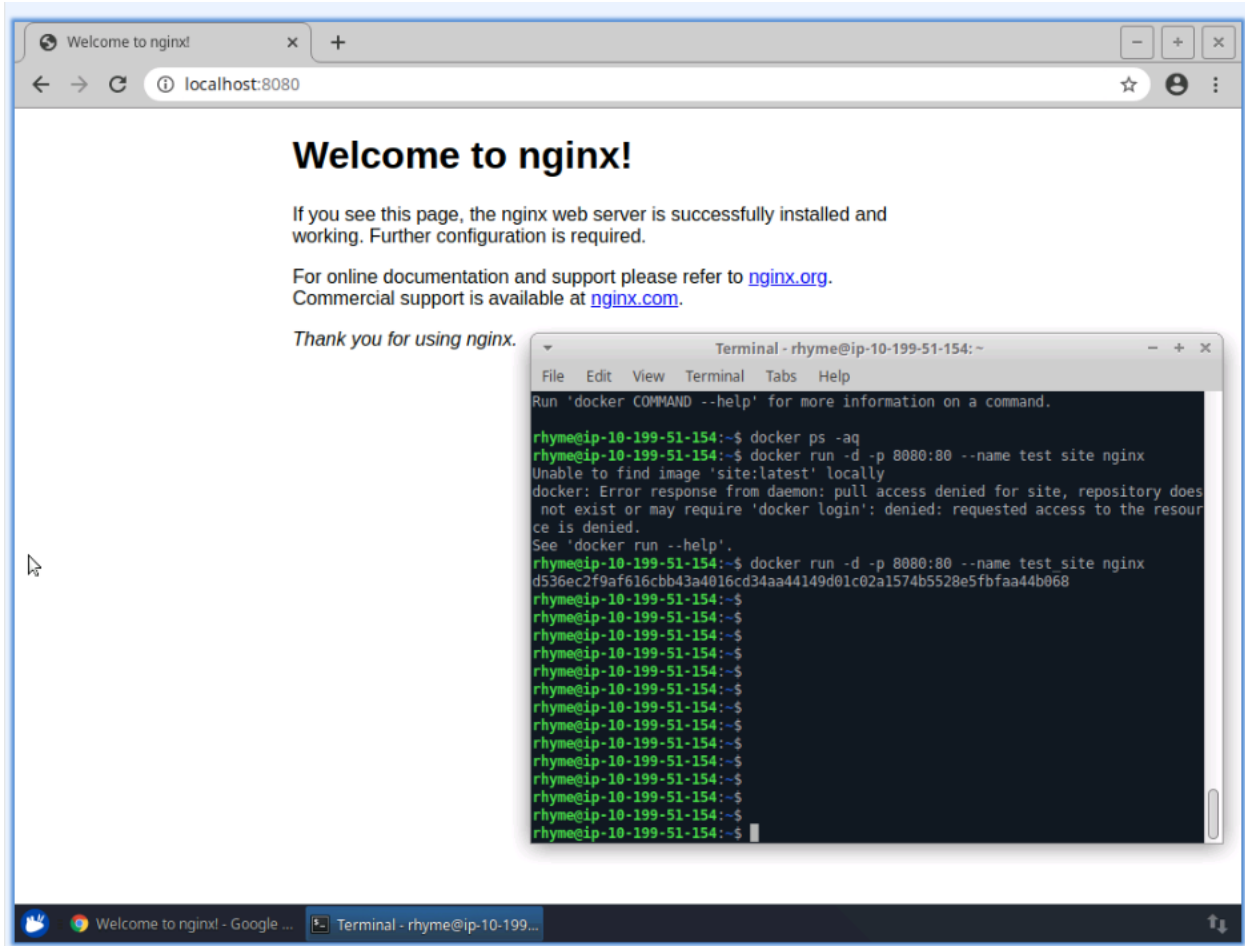
Hanya dengan menggunakan perintah “**docker rm -f \$(docker ps -aq)**” maka semua proses yang tadi berjalan dan tidak bisa dihapus, maka akan hilang.

```
rhyme@ip-10-199-51-154:~$ docker ps -aq
rhyme@ip-10-199-51-154:~$
```

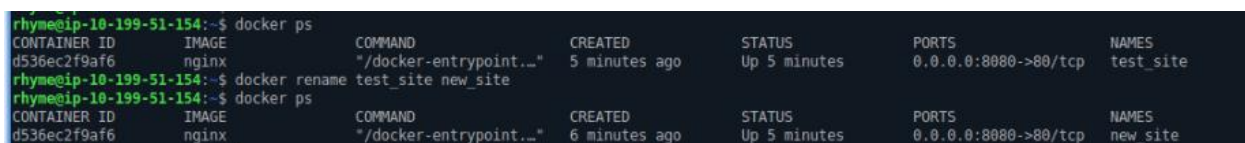
Hanya dari sini anda bisa memastikan bahwa semua proses yang tadi ada 3 yang berjalan, sudah hilang seketika.

```
rhyme@ip-10-199-51-154:~$ docker run -d -p 8080:80 --name test_site nginx
d536ec2f9af616cbb43a4016cd34aa44149d01c02a1574b5528e5fbfaa44b068
```

Setelah itu kita akan membuat container nginx baru lagi, yang nantinya port nginx 80 dimapping ke 8080 dengan nama container test_site, maka kita hanya perlu menjalankan perintah “**docker run -d -p 8080:80 --name test_site nginx**”. Maka setelah itu kita bisa mengecek dengan membuka chrome dan merefresh ulang halaman chrome tersebut.



Maka chrome akan tampil seperti pada gambar diatas



Setelah anda berhasil menjalankan nginx dengan container baru di chrome, maka anda perlu mencoba untuk mengganti nama container dari site_new ke new_site dengan menggunakan perintah “**docker rename test_site new_site**”. Setelah itu anda bisa mengecek docker ps sebelumnya dengan yang baru anda ketikan, anda hanya perlu focus ke bagian name container saja. Maka disana akan ada sebuah perbedaan dari yang sebelumnya test_site menjadi new_site.

```
d536ec2f9af6 nginx /docker-entrypoint.sh 13 minutes ago Up 13 minutes 0.0.0.0:8080->80/tcp new_site
rhyme@ip-10-199-51-154:~$ docker exec -it new_site bash
root@d536ec2f9af6:/# ls
bin dev docker-entrypoint.sh home lib64 mnt proc run srv tmp var
boot docker-entrypoint.d etc lib media opt root sbin sys usr
root@d536ec2f9af6:/# mkdir test
root@d536ec2f9af6:/# ls
bin dev docker-entrypoint.sh home lib64 mnt proc run srv test usr
boot docker-entrypoint.d etc lib media opt root sbin sys tmp var
root@d536ec2f9af6:/# ls -la
total 92
drwxr-xr-x 1 root root 4096 Nov 17 16:38 .
drwxr-xr-x 1 root root 4096 Nov 17 16:38 ..
-rwxr-xr-x 1 root root 0 Nov 17 16:23 .dockerenv
drwxr-xr-x 2 root root 4096 Nov 14 00:00 bin
drwxr-xr-x 2 root root 4096 Sep 3 12:10 boot
drwxr-xr-x 5 root root 340 Nov 17 16:23 dev
drwxr-xr-x 1 root root 4096 Nov 15 13:14 docker-entrypoint.d
-rwxr-xr-x 1 root root 1616 Nov 15 13:14 docker-entrypoint.sh
drwxr-xr-x 1 root root 4096 Nov 17 16:23 etc
drwxr-xr-x 2 root root 4096 Sep 3 12:10 home
drwxr-xr-x 1 root root 4096 Nov 14 00:00 lib
drwxr-xr-x 2 root root 4096 Nov 14 00:00 lib64
drwxr-xr-x 2 root root 4096 Nov 14 00:00 media
drwxr-xr-x 2 root root 4096 Nov 14 00:00 mnt
drwxr-xr-x 2 root root 4096 Nov 14 00:00 opt
dr-xr-xr-x 189 root root 0 Nov 17 16:23 proc
drwx----- 2 root root 4096 Nov 14 00:00 root
drwxr-xr-x 1 root root 4096 Nov 17 16:23 run
drwxr-xr-x 2 root root 4096 Nov 14 00:00 sbin
drwxr-xr-x 2 root root 4096 Nov 14 00:00 srv
dr-xr-xr-x 13 root root 0 Nov 17 16:23 sys
drwxr-xr-x 2 root root 4096 Nov 17 16:38 test
drwxrwxrwt 1 root root 4096 Nov 15 13:14 tmp
drwxr-xr-x 1 root root 4096 Nov 14 00:00 usr
drwxr-xr-x 1 root root 4096 Nov 14 00:00 var
root@d536ec2f9af6:/# touch test/test.txt
root@d536ec2f9af6:/# cd test/
root@d536ec2f9af6:/test# ls -la
total 8
drwxr-xr-x 2 root root 4096 Nov 17 16:39 .
drwxr-xr-x 1 root root 4096 Nov 17 16:38 ..
-rw-r--r-- 1 root root 0 Nov 17 16:39 test.txt
root@d536ec2f9af6:/test#
```

Disini saya memasuki container baru yang tadinya sudah dibuat, maka dari itu anda bisa mencoba untuk membuat file baru. Disini saya membuat folder dan file baru, dengan nama folder test, dan nama file test.txt, didalam container nginx dengan nama new_site

```
root@d536ec2f9af6:/test# echo 'hello world' >> test.txt
root@d536ec2f9af6:/test# cat test.txt
hello world
root@d536ec2f9af6:/test# exit
exit
```

Disini saya memasukan text “hello world” ke dalam file test.txt, dengan menggunakan perintah “**echo ‘hello world’ >> test.txt**”, setelah itu anda bisa menampilkan isi dari test.txt dengan menggunakan perintah cat, maka adakn tampil seperti pada gambar diatas. Setelah itu saya keluar dari container tersebut.

```
exit
rhyme@ip-10-199-51-154:~$ ls
Desktop Documents Downloads Music Pictures Public R Templates Videos snap
rhyme@ip-10-199-51-154:~$ mkdir website
rhyme@ip-10-199-51-154:~$ cd website/
rhyme@ip-10-199-51-154:~/website$ touch test.txt
```

Setelah keluar dari container new_site, saya membuat direktori baru dengan nama website, yang nantinya akan dijadikan volume dari container baru saya ayng akan saya buat lagi

```

rhymer@ip-10-199-51-154:~/website$ docker run -d -p 8081:80 --name website2 -v $(pwd):/website nginx
4523ce16be4991418cb3de18a5a659b9aa6c8f7959ffd8e0439548a2725a47fa
rhymer@ip-10-199-51-154:~/website$ docker ps -a

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
4523ce16be49	nginx	"/docker-entrypoint..."	7 seconds ago	Up 6 seconds	0.0.0.0:8081->80/tcp	website2
d536ec2f9af6	nginx	"/docker-entrypoint..."	38 minutes ago	Up 38 minutes	0.0.0.0:8080->80/tcp	new_site

Disini saya membuat container baru dengan nama website2 dengan server nginx yang mana portnya dimapping dari 80 ke 8081, saya menggunakan 8081, karena port 8080 sudah digunakan oleh container new_site. Untuk bind volume dari saya ambilkan dari direktori yang tadi dibuat dengan nama website, dengan menggunakan perintah **"docker run -d -p 8081:80 --name website2 -v \$(pwd):/website nginx"** Setelah itu saya lakukan cek proses yang berjalan dengan menggunakan **"docker ps -a"** maka Ketika proses itu benar2 berjalan akan tampil, anda bisa mencarinya dengan focus ke kolom nama, anda hanya perlu mencari nama container baru yang tadi anda buat. Disini saya menggunakan nama "website2"

```

rhymer@ip-10-199-51-154:~/website$ docker exec -it website2 bash
root@4523ce16be49:/# ls
bin dev docker-entrypoint.sh home lib64 mnt proc run srv tmp var
boot docker-entrypoint.d etc lib media opt root sbin sys usr website
root@4523ce16be49:/# cd website/
root@4523ce16be49:/website# ls -la
total 8
drwxrwxr-x 2 1001 1001 4096 Nov 17 16:54 .
drwxr-xr-x 1 root root 4096 Nov 17 17:01 ..
-rw-rw-r-- 1 1001 1001 0 Nov 17 16:54 test.txt
root@4523ce16be49:/website# cd ..

```

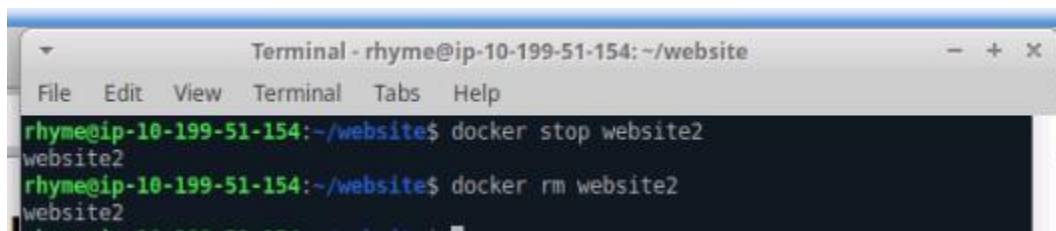
Disini saya masuk ke container baru saya dengan nama website2, setelah itu saya masuk ke direktori website yang tadi saya buat, dan menampilkan isi dari direktori tersebut. Apakah file test.txt yang tadi saya buat apa masih ada, atau tidak.

```

-rw-rw-r-- 1 1001 1001 0 Nov 17 16:54 test.txt
root@4523cel6be49:/website# cd ..
root@4523cel6be49:/# ls -la
total 92
drwxr-xr-x 1 root root 4096 Nov 17 17:01 .
drwxr-xr-x 1 root root 4096 Nov 17 17:01 ..
-rwxr-xr-x 1 root root 0 Nov 17 17:01 .dockerenv
drwxr-xr-x 2 root root 4096 Nov 14 00:00 bin
drwxr-xr-x 2 root root 4096 Sep  3 12:10 boot
drwxr-xr-x 5 root root 340 Nov 17 17:01 dev
drwxr-xr-x 1 root root 4096 Nov 15 13:14 docker-entrypoint.d
-rwxrwxr-x 1 root root 1616 Nov 15 13:14 docker-entrypoint.sh
drwxr-xr-x 1 root root 4096 Nov 17 17:01 etc
drwxr-xr-x 2 root root 4096 Sep  3 12:10 home
drwxr-xr-x 1 root root 4096 Nov 14 00:00 lib
drwxr-xr-x 2 root root 4096 Nov 14 00:00 lib64
drwxr-xr-x 2 root root 4096 Nov 14 00:00 media
drwxr-xr-x 2 root root 4096 Nov 14 00:00 mnt
drwxr-xr-x 2 root root 4096 Nov 14 00:00 opt
dr-xr-xr-x 195 root root 0 Nov 17 17:01 proc
drwx----- 2 root root 4096 Nov 14 00:00 root
drwxr-xr-x 1 root root 4096 Nov 17 17:01 run
drwxr-xr-x 2 root root 4096 Nov 14 00:00 sbin
drwxr-xr-x 2 root root 4096 Nov 14 00:00 srv
dr-xr-xr-x 13 root root 0 Nov 17 17:01 sys
drwxrwxrwt 1 root root 4096 Nov 15 13:14 tmp
drwxr-xr-x 1 root root 4096 Nov 14 00:00 usr
drwxr-xr-x 1 root root 4096 Nov 14 00:00 var
drwxrwxr-x 2 1001 1001 4096 Nov 17 16:54 website
root@4523cel6be49:/# cd website/
root@4523cel6be49:/website# ls
test.txt
root@4523cel6be49:/website# touch test2.txt
root@4523cel6be49:/website# ls -la
total 8
drwxrwxr-x 2 1001 1001 4096 Nov 17 17:12 .
drwxr-xr-x 1 root root 4096 Nov 17 17:01 ..
-rw-rw-r-- 1 1001 1001 0 Nov 17 16:54 test.txt
-rw-r--r-- 1 root root 0 Nov 17 17:12 test2.txt
root@4523cel6be49:/website#

```

Setelah itu saya mencoba untuk menampilkan isi dari container dan menampilkannya. Lalu saya membuat file baru dengan nama test2.txt



```

Terminal - rhyme@ip-10-199-51-154: ~/website
rhyme@ip-10-199-51-154:~/website$ docker stop website2
website2
rhyme@ip-10-199-51-154:~/website$ docker rm website2
website2

```

Disini saya menghentikan container website2 terlebih dahulu, dan setelah itu saya akan menghapus container tersebut

```
rhyme@ip-10-199-51-154:~/website$ cd ~/Desktop/docker-project/website/  
rhyme@ip-10-199-51-154:~/Desktop/docker-project/website$ ls  
assets  css  index.html  js
```

Setelah itu saya akan masuk ke directory dimana project docker yang paling awal tadi berada, dan menampilkan isi dari direktori tersebut

```
rhyme@ip-10-199-51-154:~/Desktop/docker-project/website$ docker rm -f $(docker p  
s -aq)  
d536ec2f9af6  
rhyme@ip-10-199-51-154:~/Desktop/docker-project/website$
```

Selanjutnya saya akan menghapus semua proses yang sedang berjalan dibelakang layer dengan perintah **"docker rm -f \$(docker ps -aq)"**

```
rhyme@ip-10-199-51-154:~/Desktop/docker-project/website$ docker run -d -p 8080:8  
0 --name website2 -v $(pwd):/usr/share/nginx/html nginx  
9af6b1ff0db943c0a460005c9bad7c35720a0e5ceca646a7af21b738ebe73b5d
```

Hampir sama seperti project sebelumnya, disini saya membuat container baru lagi dengan nama website2 dengan bind volume berbeda dengan konfigurasi sebelumnya.

```

rhyme@ip-10-199-51-154:~/Desktop/docker-project/website$ docker exec -it website
2 bash
root@9af6b1ff0db9:/# ls -la
total 88
drwxr-xr-x  1 root root 4096 Nov 17 17:38 .
drwxr-xr-x  1 root root 4096 Nov 17 17:38 ..
-rwxr-xr-x  1 root root   0 Nov 17 17:38 .dockerenv
drwxr-xr-x  2 root root 4096 Nov 14 00:00 bin
drwxr-xr-x  2 root root 4096 Sep  3 12:10 boot
drwxr-xr-x  5 root root  340 Nov 17 17:38 dev
drwxr-xr-x  1 root root 4096 Nov 15 13:14 docker-entrypoint.d
-rwxrwxr-x  1 root root 1616 Nov 15 13:14 docker-entrypoint.sh
drwxr-xr-x  1 root root 4096 Nov 17 17:38 etc
drwxr-xr-x  2 root root 4096 Sep  3 12:10 home
drwxr-xr-x  1 root root 4096 Nov 14 00:00 lib
drwxr-xr-x  2 root root 4096 Nov 14 00:00 lib64
drwxr-xr-x  2 root root 4096 Nov 14 00:00 media
drwxr-xr-x  2 root root 4096 Nov 14 00:00 mnt
drwxr-xr-x  2 root root 4096 Nov 14 00:00 opt
dr-xr-xr-x 189 root root   0 Nov 17 17:38 proc
drwx----- 1 root root 4096 Nov 17 17:43 root
drwxr-xr-x  1 root root 4096 Nov 17 17:38 run
drwxr-xr-x  2 root root 4096 Nov 14 00:00 sbin
drwxr-xr-x  2 root root 4096 Nov 14 00:00 srv
dr-xr-xr-x 13 root root   0 Nov 17 17:38 sys
drwxrwxrwt  1 root root 4096 Nov 15 13:14 tmp
drwxr-xr-x  1 root root 4096 Nov 14 00:00 usr
drwxr-xr-x  1 root root 4096 Nov 14 00:00 var
root@9af6b1ff0db9:/# ls /usr/

```

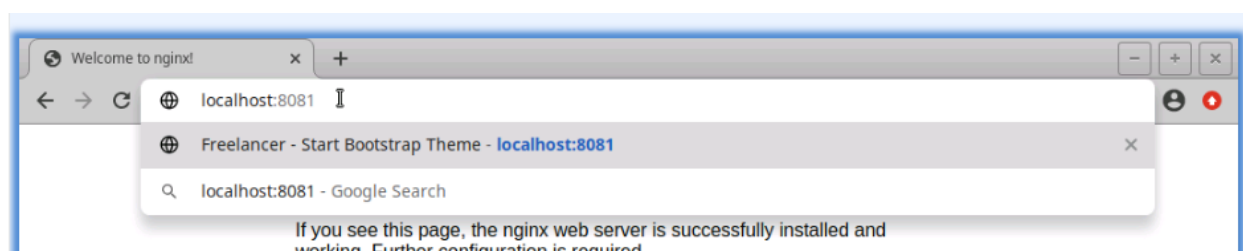
Disini saya telah memasuki container yang baru saya buat tadi dengan nama website2, dan saya akan masuk ke directory usr/share/nginx/html untuk melihat isi dari project docker apakah masih ada.

```

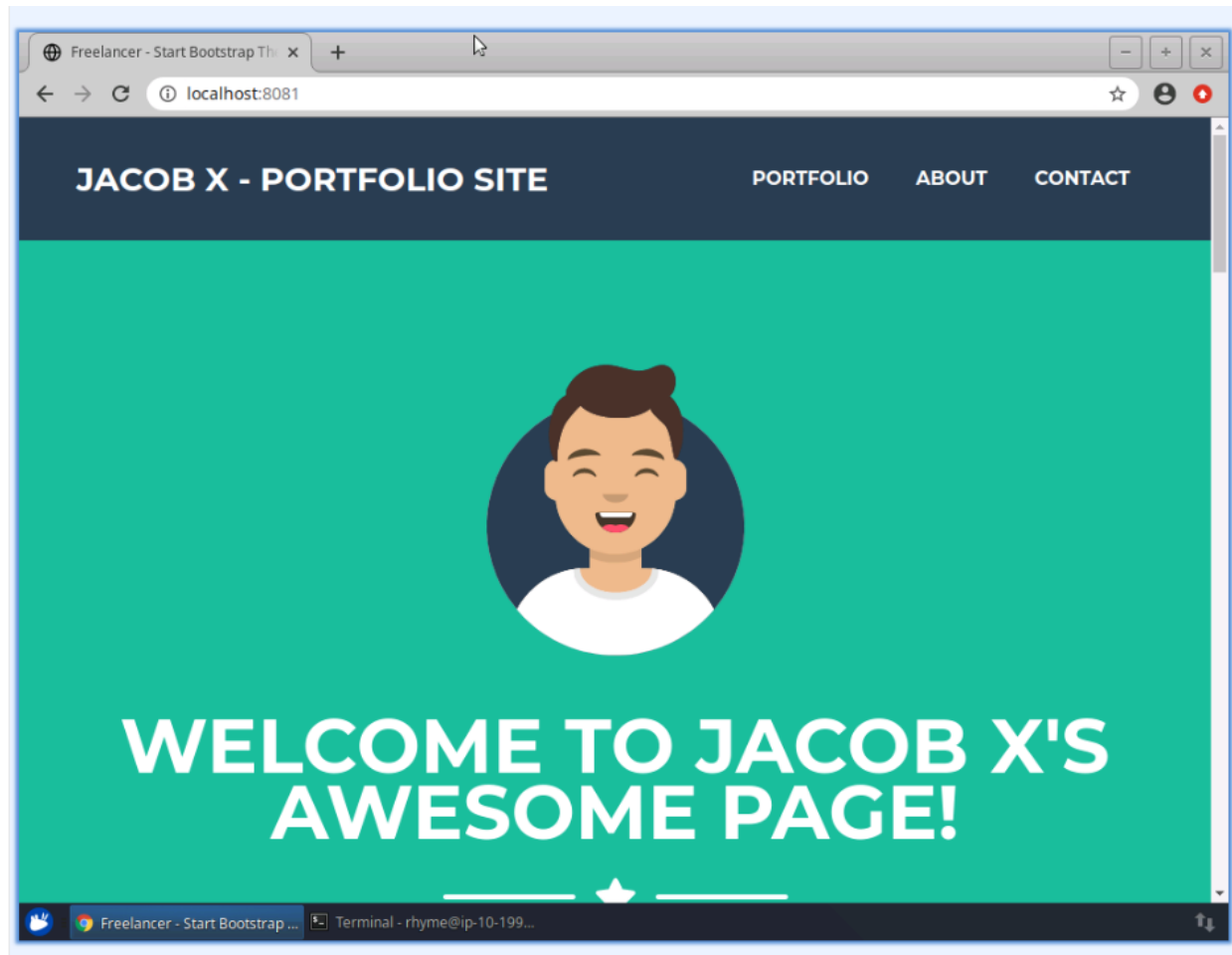
root@9af6b1ff0db9:/# ls /usr/share/nginx/html/
assets  css  index.html  js

```

Setelah saya cek ternyata masih ada, dan masih aman



OK, maka kita akan lanjut untuk pengecekan apakah aplikasi sudah selesai dideployment dnegan baik atau belum. Jangan lupa untuk mengganti port dari 8080 ke 8081 untuk akses ke container website2



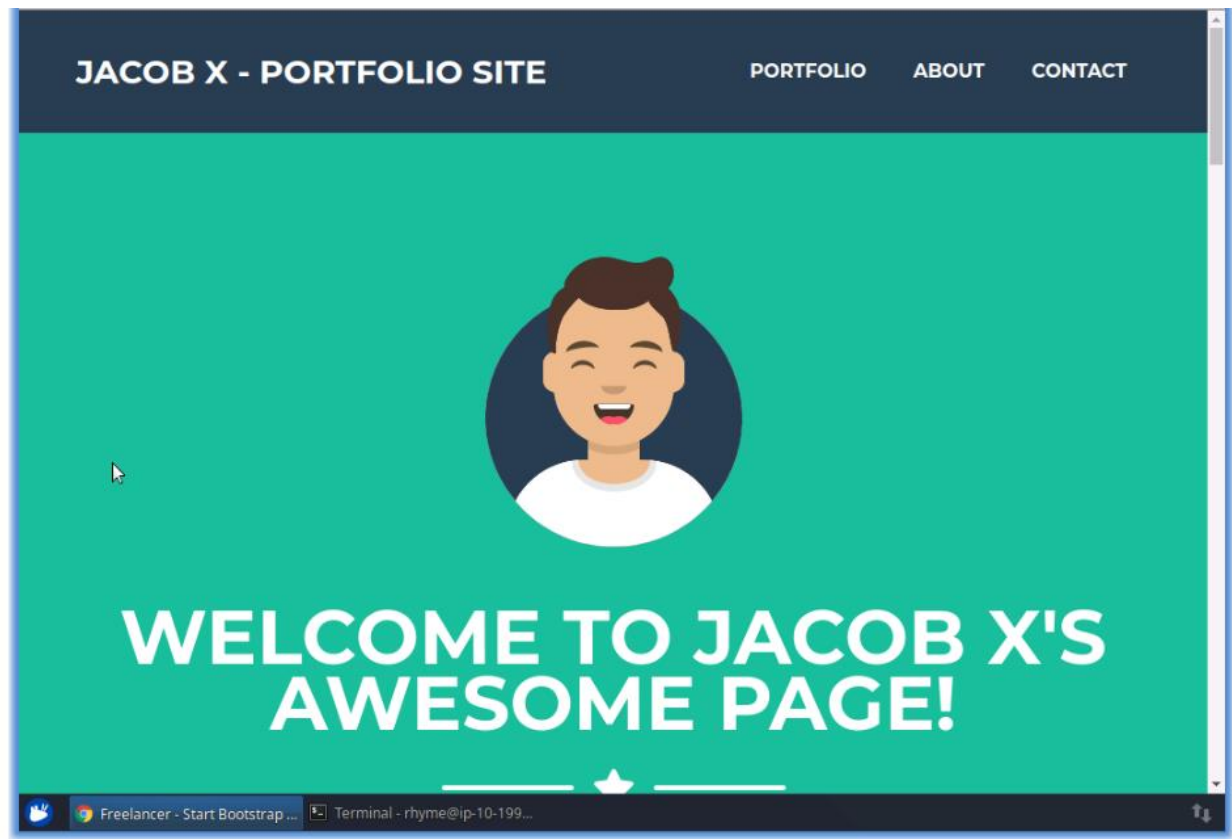
Yaps, aplikasi sukses dijalankan dengan baik sebelumnya jangan lupa untuk mengecek semua fitur2 yang ada, guna untuk memastikan aplikasi berjalan dengan baik dan aman

```
rhyme@ip-10-199-51-154:~/Desktop/docker-project/website$ chmod -R o+rX ~/Desktop/  
/docker-project/dockerarchitectur1.webp  
rhyme@ip-10-199-51-154:~/Desktop/docker-project/website$
```

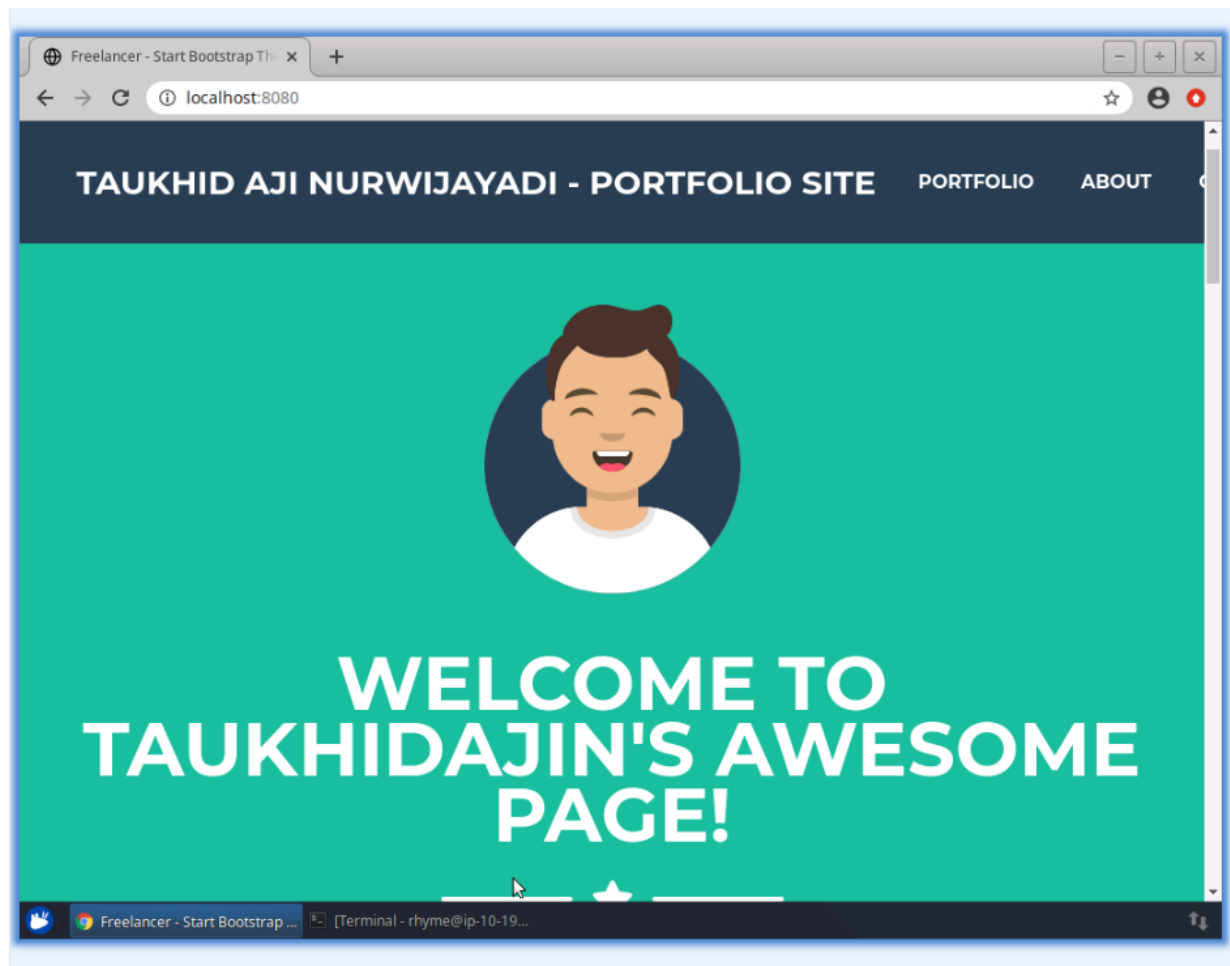
Setelah itu akan saya ganti untuk “other” bisa “membaca” dan “mengeksekusi” file docker.....webp, Lalu setelah anda menjalankan perintah tadi, anda bisa mengetikan perintah “code.” untuk mengeksekusi file “index.html”.

```
index.html x
index.html > html > body#page-top
4 <meta charset="utf-8" />
5 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
6 <meta name="description" content="" />
7 <meta name="author" content="" />
8 <title>Freelancer - Start Bootstrap Theme</title>
9 <!-- Favicon-->
10 <link rel="icon" type="image/x-icon" href="assets/img/favicon.ico" />
11 <!-- Font Awesome icons (free version)-->
12 <script src="https://use.fontawesome.com/releases/v5.13.0/js/all.js" crossorigin="anonymous"></script>
13 <!-- Google fonts-->
14 <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700" rel="stylesheet" type="text/css">
15 <link href="https://fonts.googleapis.com/css?family=Lato:400,700,400italic,700italic" rel="stylesheet" type="text/css">
16 <!-- Core theme CSS (includes Bootstrap)-->
17 <link href="css/styles.css" rel="stylesheet" />
18 </head>
19 <body id="page-top">
20 <!-- Navigation-->
21 <nav class="navbar navbar-expand-lg bg-secondary text-uppercase fixed-top" id="mainNav">
22 <div class="container">
23 <a class="navbar-brand js-scroll-trigger" href="#page-top">Taukhid Aji Nurwijayadi - portfolio</a>
24 <button class="navbar-toggler navbar-toggler-right text-uppercase font-weight-bold bg-secondary text-white" type="button" data-toggle="collapse" data-target="#navbar" data-ri...>
25 <div class="collapse navbar-collapse">
26 <ul class="navbar-nav">
27 <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px-0 px-lg-3 rounded" href="#page-top">Home</a>
28 <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px-0 px-lg-3 rounded" href="#about">About</a>
29 <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px-0 px-lg-3 rounded" href="#services">Services</a>
30 <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px-0 px-lg-3 rounded" href="#portfolio">Portfolio</a>
31 <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px-0 px-lg-3 rounded" href="#contact">Contact</a>
32 </ul>
33 </div>
34 </div>
35 </div>
36 </nav>
37 <!-- Masthead-->
38 <header class="masthead bg-primary text-white text-center">
39 <div class="container d-flex align-items-center flex-column">
40 <!-- Masthead Avatar Image-->
41 
42 <!-- Masthead Heading-->
43 <h1 class="masthead-heading text-uppercase mb-0">Welcome to Taukhidajin's Awesome Page!</h1>
44 <!-- Icon Divider-->
45 <div class="divider-custom divider-light">
46 <div class="divider-custom-line"></div>
47 </div>
48 </div>
49 </header>
```

Untuk beberapa yang saya tandai adalah yang saya ganti dari isi file index.html



Sebelum diedit



Sesudah diedit, terimakasih sudah menyimak 😊