Docker Overview

Simple tutorial for Docker

Junyoung KIM

Docker Overview.

Why docker?

Multiple versions of software

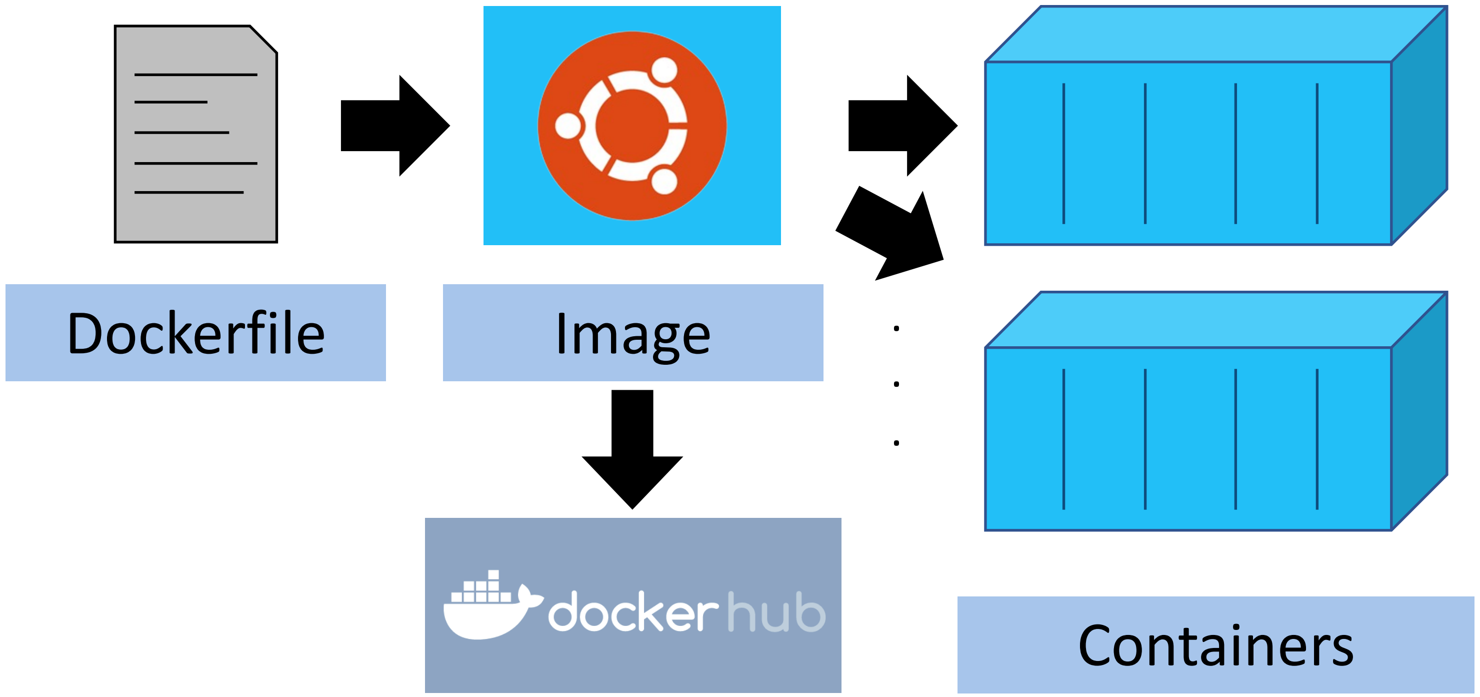
Complicated dependencies

Setting many systems into certain environments

⋮

Docker Overview.

Isolate our environment.



1. Write Dockerfile
2. Build Docker image from Dockerfile
3. Run Docker container from Docker image

Isolation

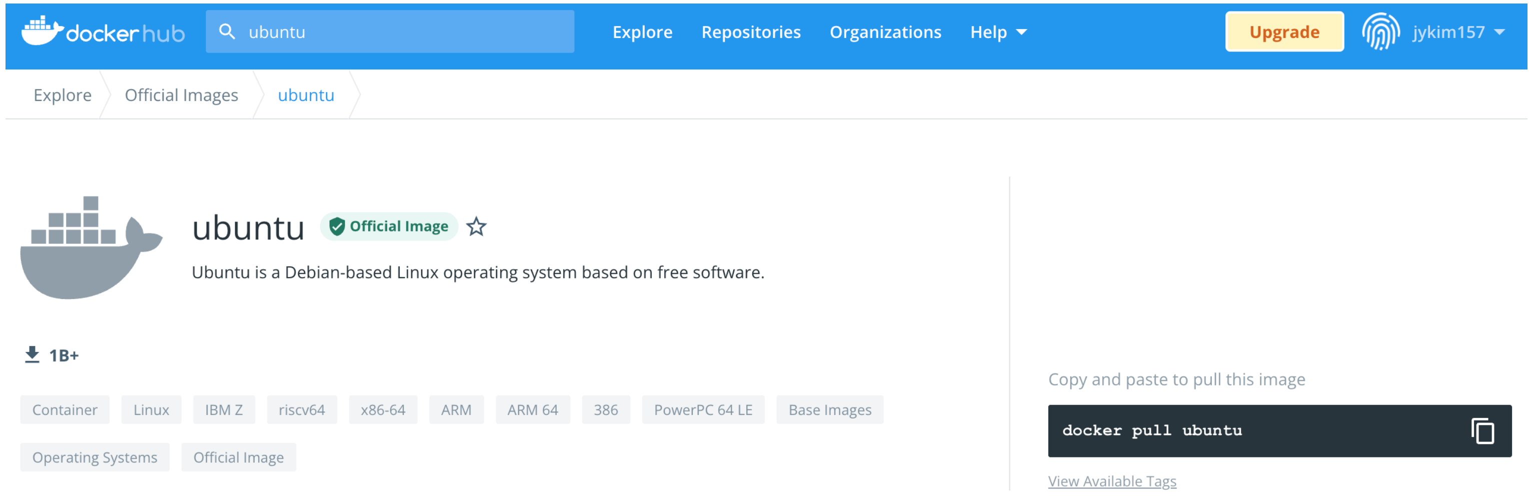
Portability

Performance

Write a Dockerfile

Find the base image in dockerhub.

https://hub.docker.com/



ubuntu

Ubuntu is a Debian-based Linux operating system based on free software.

Copy and paste to pull this image

```
docker pull ubuntu
```

Name format : <image name>:<tag name>

- ubuntu:20.04 [Example would be based on Ubuntu:20.04]
- nvidia/cuda:11.4.0-devel-ubuntu20.04
- alpine:3.15.2
- <image name>:latest will bind the latest version.

Toy example

Example situation [*Not practical just for example*]

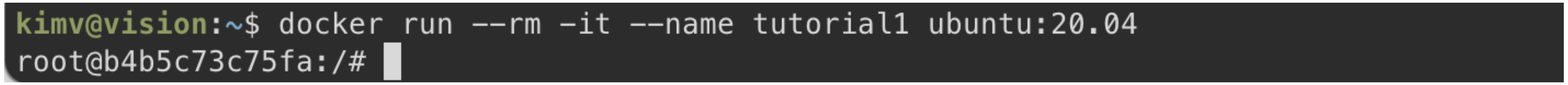
> : for command in the server
# : for command in the container

--rm : for remove container after exit.
-it : for terminal input(interactive)
--name <name> : for specify container name
--net=host : for giving the same internet config.

1. Make a first container based on the base image.

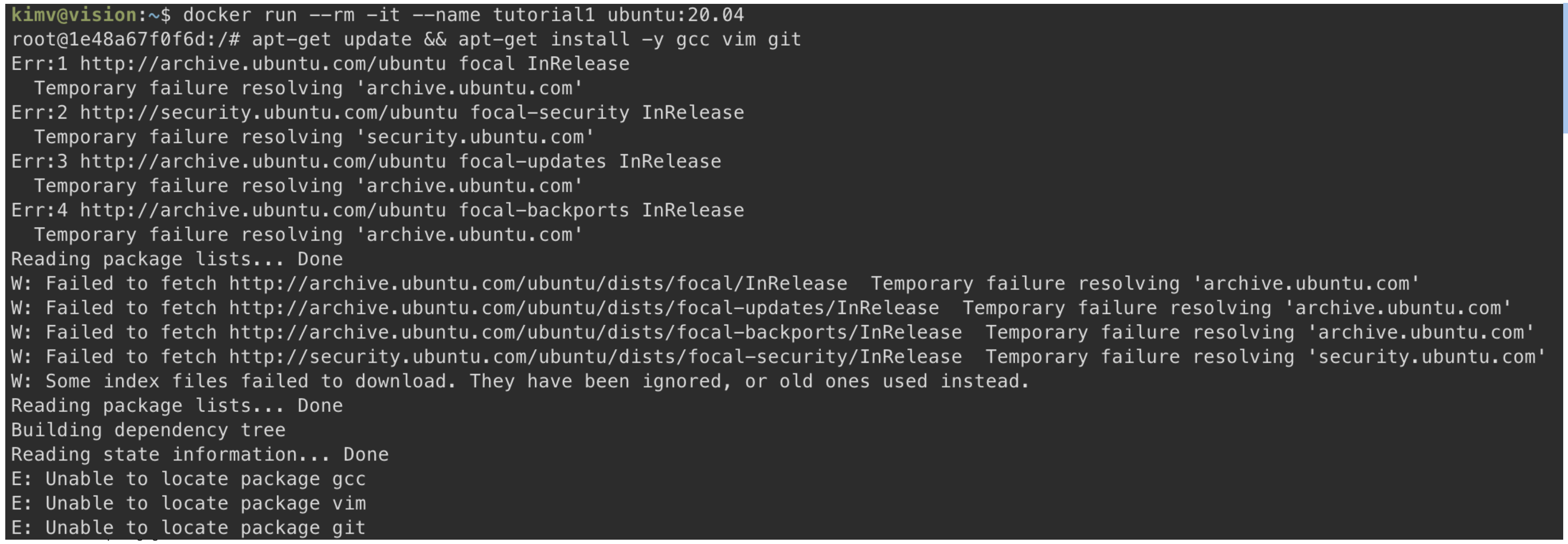
```
> docker run --rm -it --net=host --name tutorial1 ubuntu:20.04
```

Then, you will be in the container.



```
kimv@vision:~$ docker run --rm -it --name tutorial1 ubuntu:20.04
root@b4b5c73c75fa:/#
```

Without "--net=host" option, you **might** encounter the following error message due to the internet connection :



```
kimv@vision:~$ docker run --rm -it --name tutorial1 ubuntu:20.04
root@1e48a67f0f6d:/# apt-get update && apt-get install -y gcc vim git
Err:1 http://archive.ubuntu.com/ubuntu focal InRelease
  Temporary failure resolving 'archive.ubuntu.com'
Err:2 http://security.ubuntu.com/ubuntu focal-security InRelease
  Temporary failure resolving 'security.ubuntu.com'
Err:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease
  Temporary failure resolving 'archive.ubuntu.com'
Err:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease
  Temporary failure resolving 'archive.ubuntu.com'
Reading package lists... Done
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/focal/InRelease  Temporary failure resolving 'archive.ubuntu.com'
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/focal-updates/InRelease  Temporary failure resolving 'archive.ubuntu.com'
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/focal-backports/InRelease  Temporary failure resolving 'archive.ubuntu.com'
W: Failed to fetch http://security.ubuntu.com/ubuntu/dists/focal-security/InRelease  Temporary failure resolving 'security.ubuntu.com'
W: Some index files failed to download. They have been ignored, or old ones used instead.
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package gcc
E: Unable to locate package vim
E: Unable to locate package git
```

'--net=host' for docker run
'--network=host' for docker build

Toy example

2. Make the environment what you want [As if just in the usual linux environment].

```
# apt-get update && apt-get install -y <Required Packages>
# git clone <Git Repo Link>
# cd.. mkdir.. gcc .. make..
```

⋮

3. Make sure none of the commands require the user input during (2.)

If some package requires some user input during apt-get install, the following lines will be helpful.

```
ENV DEBIAN_FRONTEND noninteractive
ENV TZ Asia/Seoul
RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone
```

Alternatively, you can squash the multiple lines of ENV command into one layer.

```
ENV DEBIAN_FRONTEND=noninteractive \
        TZ=Asia/Seoul
RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone
```

Toy example

4. Write a Dockerfile based on what you execute in the base container.

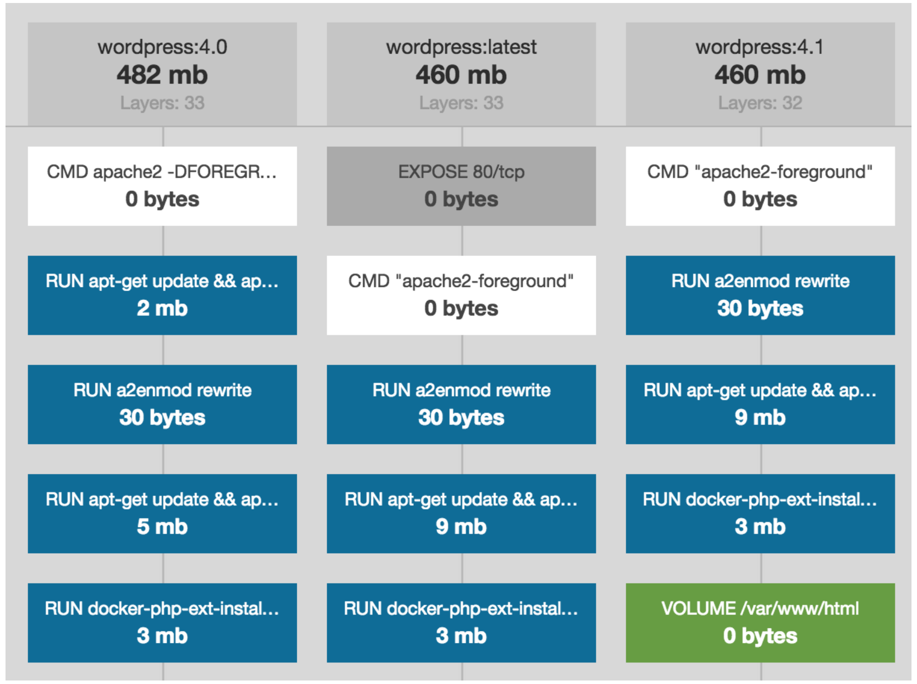
```
FROM ubuntu:20.04

RUN apt-get update && \
   apt-get install -y wget vim git gcc && \
   mkdir /workspace
COPY ./test.c /workspace

ENV PATH $PATH:/workspace
```

Execute the commands as you did in the base container.

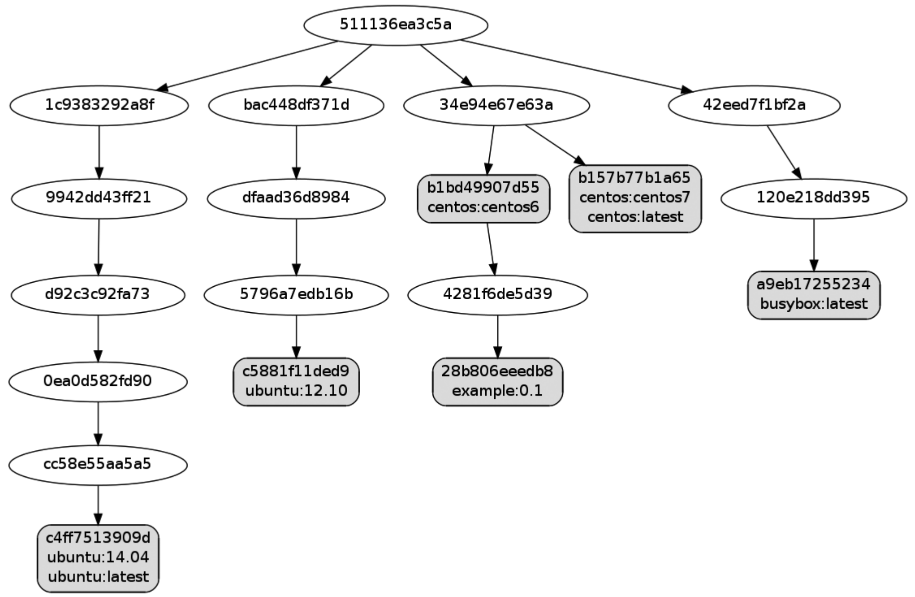
Copy local files into the image.

Set environment variables.

Toy example



wordpress:4.0 **482 mb** Layers: 33	wordpress:latest **460 mb** Layers: 33	wordpress:4.1 **460 mb** Layers: 32
CMD apache2 -DFOREGR... **0 bytes**	EXPOSE 80/tcp **0 bytes**	CMD "apache2-foreground" **0 bytes**
RUN apt-get update && ap... **2 mb**	CMD "apache2-foreground" **0 bytes**	RUN a2enmod rewrite **30 bytes**
RUN a2enmod rewrite **30 bytes**	RUN a2enmod rewrite **30 bytes**	RUN apt-get update && ap... **9 mb**
RUN apt-get update && ap... **5 mb**	RUN apt-get update && ap... **9 mb**	RUN docker-php-ext-instal... **3 mb**
RUN docker-php-ext-instal... **3 mb**	RUN docker-php-ext-instal... **3 mb**	VOLUME /var/www/html **0 bytes**



Each Command makes a layer.

Try to make target environment in the container of base image.
Then, write the histories of command into the dockerfile.

https://www.ctl.io/developers/blog/post/imagelayers-io-docker-visualization-and-badges
http://pyrasis.com/book/DockerForTheReallyImpatient/Chapter01/02

Toy example

4. Write a Dockerfile based on what you execute in the base container.

```
FROM ubuntu:20.04

RUN apt-get update && \
   apt-get install -y wget vim git gcc && \
   mkdir /workspace
COPY ./test.c /workspace

ENV PATH $PATH:/workspace
```

To minimize the number of layers.

“-y” required to avoid the user-input

Copy local files into the image.

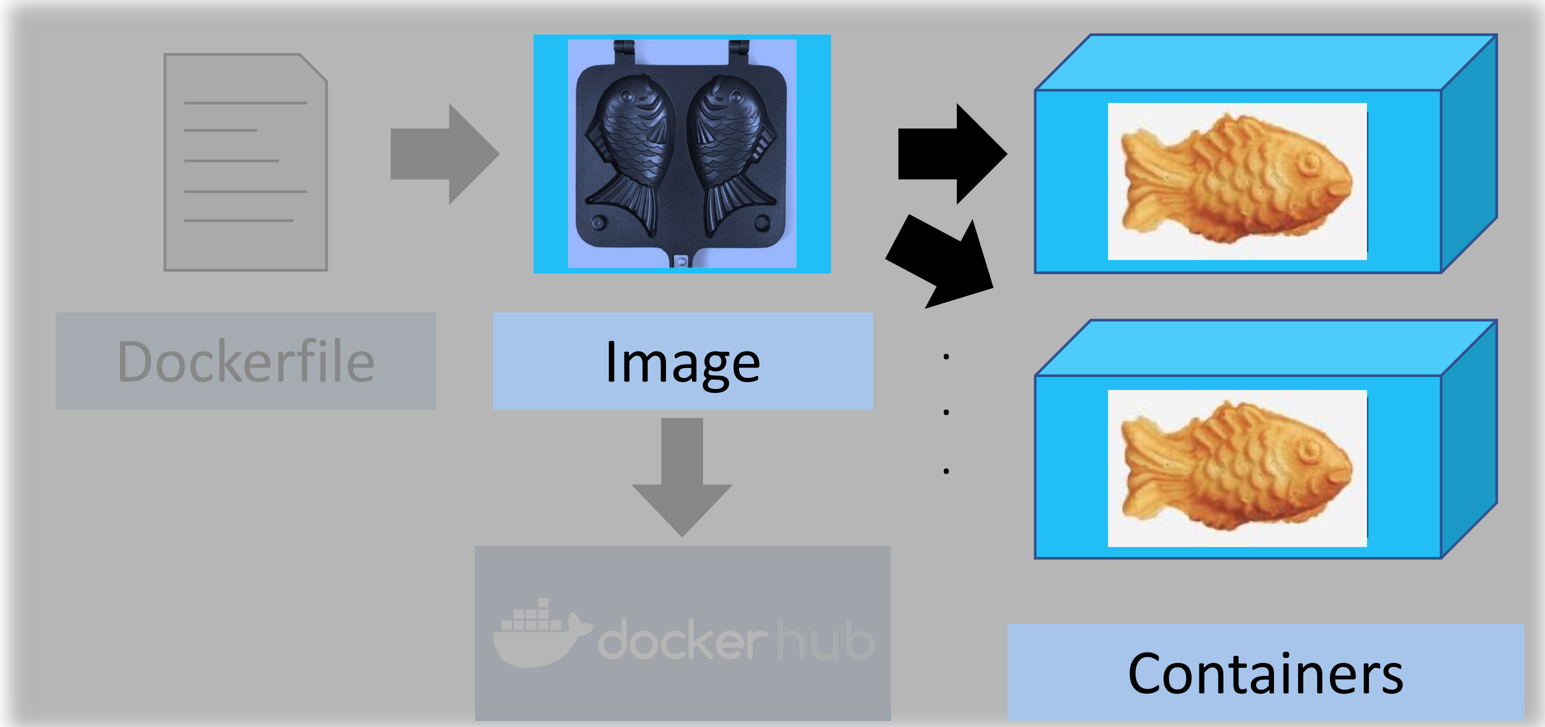
Set environment variables.

FROM	- Specify the base image.
RUN	- Execute the command onto the image.
COPY	- Copy local files into the image.
ENV	- Set environment variable in the image.

There are many other commands, although RUN, COPY, ENV are sufficient to deal with general setting.

https://docs.docker.com/develop/develop-images/dockerfile_best-practices/
http://pyrasis.com/docker.html (Korean 😅)
https://cultivo-hy.github.io/docker/image/usage/2019/03/14/Docker%EC%A0%95%EB%A6%AC/ (Korean 😅)

Template(Image) & Instances(Container) Metaphor



Robustness / Parallelization

https://m.bakingmon.com/product/detail.html?product_no=1560
https://shopping.interpark.com/product/productInfo.do?prdNo=7000809496&uaTp=1&
https://www.chosun.com/economy/industry-company/2021/11/30/KUWQY2WAJZETJJCJBO5HWWHTRA/

Toy example : After the Dockerfile writing.

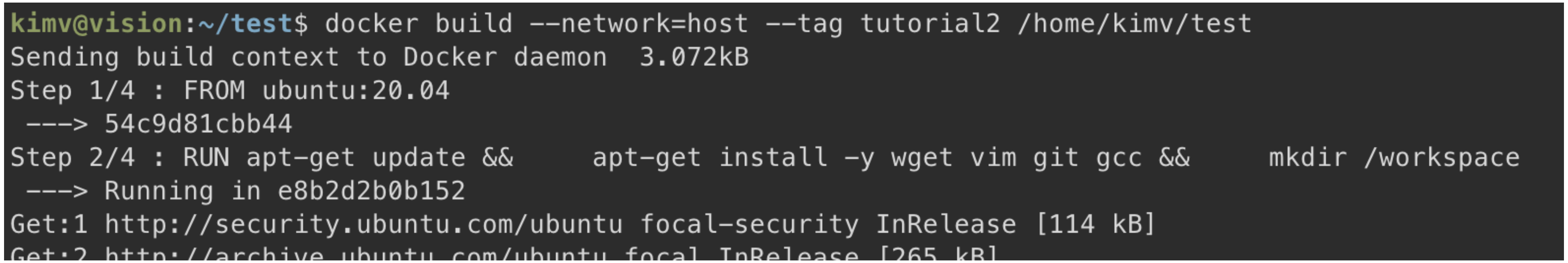
5. Build the Docker image based on your Dockerfile.

> docker build --network=host --tag tutorial2 ~/test

(format) docker build <options> <Directory of Dockerfile>

Relative path also fine

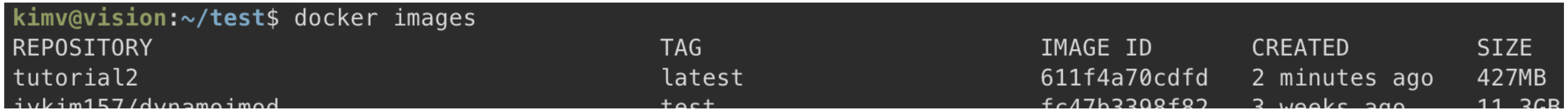
--tag <Image Name> : for the image name



```
kimv@vision:~/test$ docker build --network=host --tag tutorial2 /home/kimv/test
Sending build context to Docker daemon  3.072kB
Step 1/4 : FROM ubuntu:20.04
 ---> 54c9d81cbb44
Step 2/4 : RUN apt-get update &&     apt-get install -y wget vim git gcc &&     mkdir /workspace
 ---> Running in e8b2d2b0b152
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
```

6. Check your image

> docker images



```
kimv@vision:~/test$ docker images
REPOSITORY                                TAG                      IMAGE ID       CREATED          SIZE
tutorial2                                 latest                   611f4a70cdfd   2 minutes ago    427MB
jvkim157/dynamoimod                       test                     fc47b3398f82   3 weeks ago      11.3GB
```

7. Next…

"Instantiate" the image into the container

Share the image into the dockerhub

Toy example : “Instantiate” the image

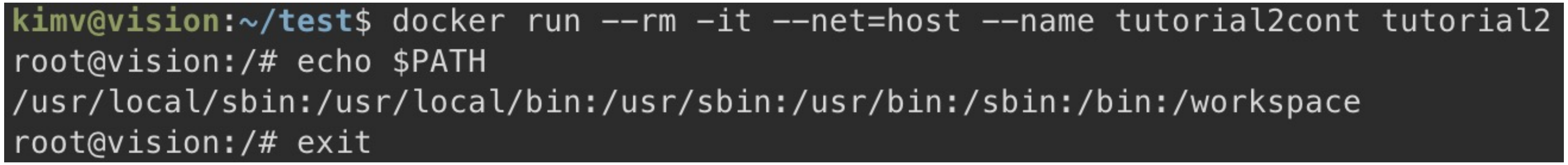
8. Run the Docker container based on your Docker image.

> docker run --rm -it --net=host --name tutorial2cont tutorial2
(format) docker run <options> <image name>

--rm : for remove container after exit.
-it : for terminal input(interactive)
--name <name> : for specify container name
--net=host : for giving the same internet config.

Other useful options

-v <Directory>:<Container Dir> : mount the <Dir> into the <Container Dir> in the container.
ex. -v ~/testData:/data : you can access ~/testData directory through /data directory in the container.
-gpus all : enable all of the gpus in the container [*Require Nvidia-docker installed in the server*]



```
kimv@vision:~/test$ docker run --rm -it --net=host --name tutorial2cont tutorial2
root@vision:/# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/workspace
root@vision:/# exit
```

- You can see the $PATH modified as we write in the Dockerfile.
- You can exit from the container with “exit” command.
- If you want to detach(while not stop the container) from the container, type “Ctrl P + Q”

Ctrl P + Q : shortcut to “Detaching without Stopping” from container

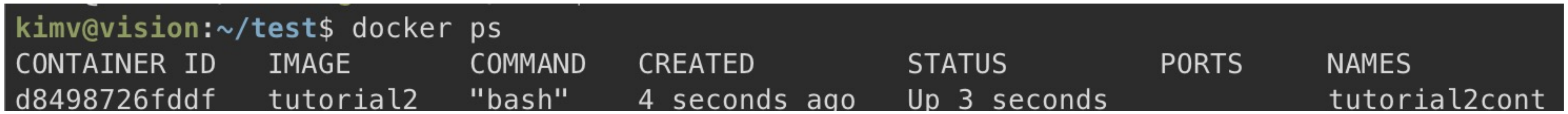
Toy example : “Instantiate” the image

8. Run the Docker container based on your Docker image.

> docker run --rm -it --net=host --name tutorial2cont tutorial2
(format) docker run <options> <image name>

Ctrl P + Q : shortcut to “Detaching without Stopping” from container

After detached with Ctrl P + Q, you can check your detached container with “docker ps” command.
> docker ps

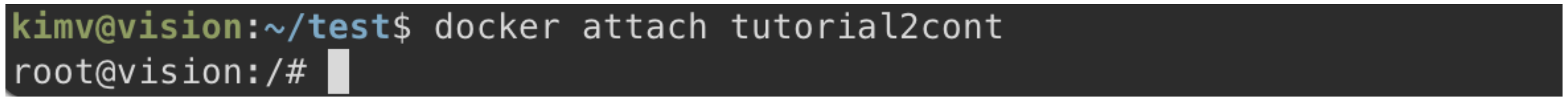


```
kimv@vision:~/test$ docker ps
CONTAINER ID   IMAGE       COMMAND   CREATED          STATUS         PORTS     NAMES
d8498726fddf   tutorial2   "bash"    4 seconds ago    Up 3 seconds             tutorial2cont
```

With this shortcut, we can efficiently execute jobs in background(like tmux).

docker attach <Container Name>

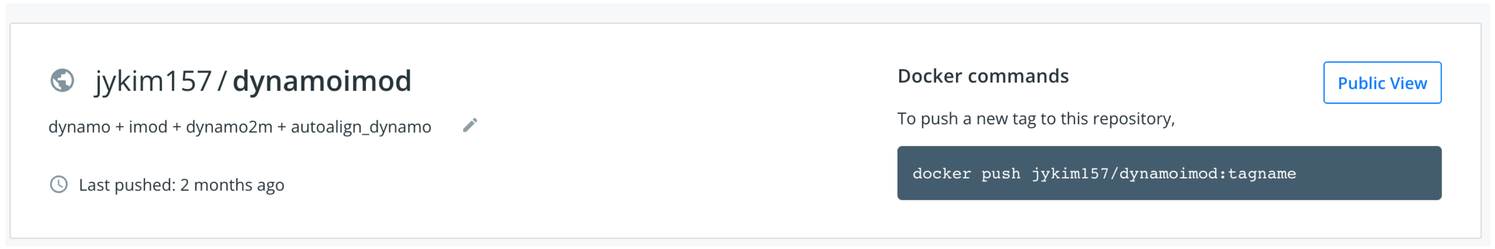
You can attach the detached container again with “docker attach” command. If you forget the name, use “docker ps”
> docker attach <Container Name>



```
kimv@vision:~/test$ docker attach tutorial2cont
root@vision:/#
```

Toy example : Share the image

Make a repository of your image.



jykim157 / **dynamoimod**

dynamo + imod + dynamo2m + autoalign_dynamo

Last pushed: 2 months ago

Docker commands

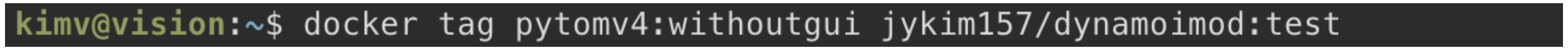
Public View

To push a new tag to this repository,

```
docker push jykim157/dynamoimod:tagname
```

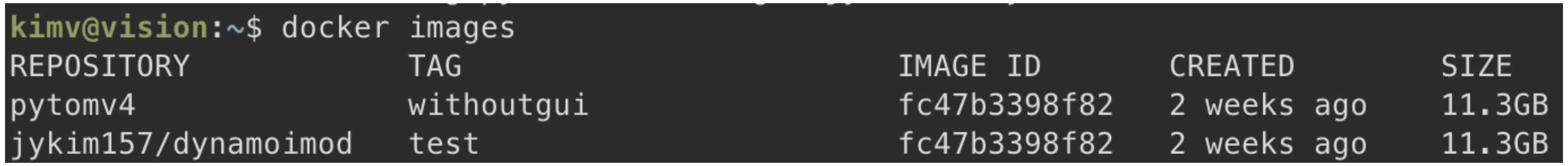
Tag the image.

> docker tag <server image name> <dockerhub image name>



```
kimv@vision:~$ docker tag pytomv4:withoutgui jykim157/dynamoimod:test
```

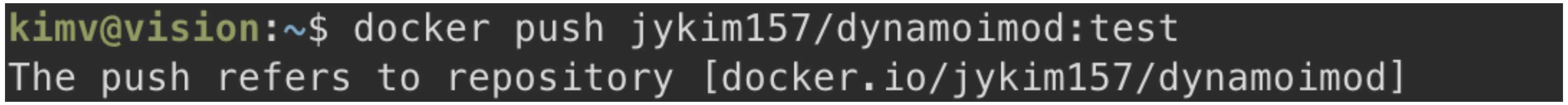
Then, you can check your tagged image with ”docker images”



```
kimv@vision:~$ docker images
REPOSITORY            TAG                      IMAGE ID       CREATED        SIZE
pytomv4               withoutgui               fc47b3398f82   2 weeks ago    11.3GB
jykim157/dynamoimod   test                     fc47b3398f82   2 weeks ago    11.3GB
```

Push the image.

> docker push <dockerhub image name>



```
kimv@vision:~$ docker push jykim157/dynamoimod:test
The push refers to repository [docker.io/jykim157/dynamoimod]
```

Login(Authentication) problem can occur! docker login would help!

Docker GUI

Overview [Assume already forward GUI config through server]

You need to access the server with -X option(-Y might okay) ex. ssh -p 7777 -X <User>@<Server>

```
> xauth list
```

Then, copy the appropriate line. If there are too many lines, removing all lines with "xauth remove" and re-access server worked for me.

```
# xauth add <copied line>
```

Obviously, you need to install xauth in your container(or image).

About GUI

https://github.com/KJYoung/DockerFilesForCryoET > README.md > Docker GUI section would help.

For the Java-based GUI application(including Matlab), there is an GUI error with xquartz in Mac
https://github.com/XQuartz/XQuartz/issues/31