

Docker

SWPP 2022

Docker container as a grading environment

- We will provide the testing environment where your codes will be graded since each student has a different development environment (e.g., OS).
- Especially, a *Docker* container will be used for grading.
- We suggest you go through the following slides so that TAs can run your codes properly in the same environment.

What is Docker? Why Docker?

- *Docker* provides an isolated environment, called a container, for each application.
- Docker enables you to separate your applications from your infrastructure.
- So, even when you are using *Windows*, you can run your program on any other environments (e.g., *Ubuntu*, *Alpine* ...).
- For us, we can share the environment through the container; a container with *node(v14.17.6)* on *Linux* will be used.
 - https://hub.docker.com/_/node

What is Docker? Why Docker?

- Start from an empty desk? 🤔



What is Docker? Why Docker?

- Start from a prepared desk for *each app*! 😎 (+ save + share)



What is Docker? Why Docker?

NOTE: This material covers only the minimum requirements for checking the assignments.

If you are interested, you can check more on the details in the following links:

- **Introductions**

- English: [Introduction to Docker containers](#)
- Korean: [Docker 컨테이너 소개](#)

- **Practice**

- English: [Build a containerized web application with Docker](#)
- Korean: [Docker를 사용하여 컨테이너화된 웹 애플리케이션 빌드](#)

Install Docker (Ubuntu)

```
$ sudo apt-get update
$ sudo apt-get install apt-transport-https ca-certificates curl software-properties-common
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
$ sudo add-apt-repository \
    "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) \
    stable"
$ sudo apt-get update
$ sudo apt-get install docker-ce
```

Recommended version for Docker Engine: 20.10.XX

Install Docker

```
$ sudo docker version
```

```
# output:
```

```
Client: Docker Engine - Community
```

```
Version:          20.10.05
```

```
API version:       1.41
```

```
Go version:        go1.13.15
```

```
Git commit:        55c4c88
```

```
Built:             Tue Mar  2 20:13:00 2021
```

```
OS/Arch:           darwin/amd64
```

```
Experimental:      true
```

```
Server: Docker Engine - Community
```

```
Engine:
```

```
Version:          20.10.05
```

```
API version:       1.41 (minimum version 1.12)
```

```
Go version:        go1.13.15
```

```
...
```


Install Docker

For Mac: <https://docs.docker.com/desktop/mac/install/>

For Windows: <https://docs.docker.com/desktop/windows/install/>

Simple Docker Tutorial

1. Check if *Docker* is installed

- Both the following two commands should show the proper messages:

```
$ docker version
```

```
dhkim ~/hw3/swpp-hw3-kdh0102 main !48 ?4 docker version
Client: Docker Engine - Community
 Cloud integration: 1.0.12
 Version:          20.10.5
 API version:      1.41
 Go version:       go1.13.15
 Git commit:       55c4c88
 Built:            Tue Mar  2 20:13:00 2021
 OS/Arch:          darwin/amd64
 Context:          default
 Experimental:     true
```

```
$ docker ps -a
```

```
dhkim ~/hw3/swpp-hw3-kdh0102 main !48 ?4 docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS   NAMES
```

1. Pull *Docker* image

- Pull image from remote => check image on local

```
$ docker pull snuspl/swpp:session1
```

```
jaewoo@maengjaeuui-MacBookAir swpp % docker pull snuspl/swpp:session1
session1: Pulling from snuspl/swpp
d2f2687beb6d: Already exists
4f1a3e831c98: Already exists
a823482a1469: Already exists
4c2c7a10f616: Already exists
24000b1aeb3: Already exists
062c64d4af79: Already exists
21754efc5e80: Already exists
1bc1a454fe2e: Already exists
440e7866d2d9: Already exists
5ae878f3db85: Already exists
d1a08593e516: Already exists
4f4fb700ef54: Already exists
Digest: sha256:4459cd59f3943d4d2e920a2c7755ae37cf91d437b7b2c2685970979aec68b97c
Status: Downloaded newer image for snuspl/swpp:session1
docker.io/snuspl/swpp:session1
```

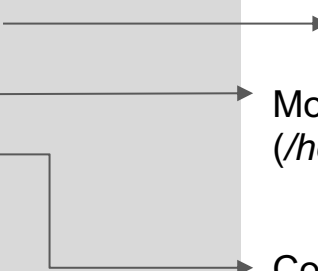
```
$ docker images
```

```
jaewoo@maengjaeuui-MacBookAir swpp % docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
snuspl/swpp         session1     e61827f20122     3 hours ago     816MB
```

2. Run *Docker* container

- Run a *docker* container with the downloaded image.
- Use `docker run` command as follows:

```
$ docker run -it \  
-p 127.0.0.1:3000:3000 \  
-v ${PWD}:/home \  
--name session1 \  
snuspl/swpp:session1 \  
/bin/bash
```



Bind host port 3000 to container port 3000.

Mount the current directory to the container path (*/home*).

Container name.

3. Basic Container Management

- check local containers
(-a option: see even the stopped containers)

```
$ docker ps -a
```

- stop a running container

```
$ docker stop {container_name}
```

- start a stopped container

```
$ docker start {container_name}
```

3. Basic Container Management

- run commands in a running container
(usually used to attach to running containers)

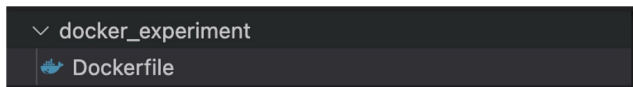
```
$ docker exec -it {container_name} /bin/bash
```

- remove a container

```
$ docker rm {container_name}
```

4. Docker Build

- Make a file named Dockerfile
 - **FROM**: basic docker image to start from
 - **RUN**: shell commands you want to run in the image
 - **WORKDIR**: note the starting directory in the image
 - more syntax (<https://docs.docker.com/engine/reference/builder/>)



```
1  FROM snuspl/swpp:session1
2
3  RUN apt-get install npm -y
4
5  WORKDIR /home
```


4. Docker Build

- Build an image with Dockerfile
 - navigate to the directory with Dockerfile

```
$ docker build -t {user_name}/swpp:session1_with_npm .
```

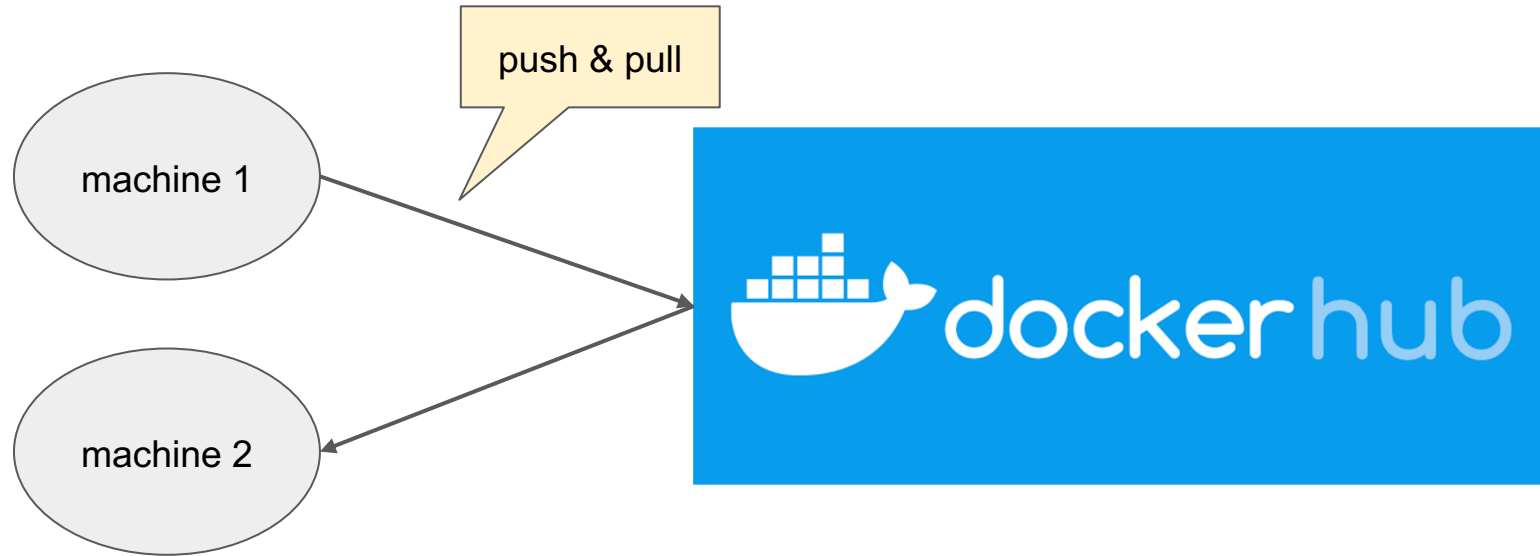
```
jaewoo@maengjaeuui-MacBookAir docker_experiment % docker build -t snuspl/swpp:session1_with_npm .
[+] Building 11.9s (4/6)
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 110B                                                0.0s
=> [internal] load .dockerignore                                                    0.0s
=> => transferring context: 2B                                                      0.0s
=> [internal] load metadata for docker.io/snuspl/swpp:session1                    0.0s
=> [1/3] FROM docker.io/snuspl/swpp:session1                                     0.1s
=> [2/3] RUN apt-get install npm -y                                              11.7s
=> => # Get:31 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 libubsan0 arm64 7.5.0-3ubuntu1~18.04 [117 kB]
=> => # Get:32 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 libgcc-7-dev arm64 7.5.0-3ubuntu1~18.04 [833 kB]
=> => # Get:33 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 gcc-7 arm64 7.5.0-3ubuntu1~18.04 [7772 kB]
=> => # Get:34 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 gcc arm64 4:7.4.0-1ubuntu2.3 [5208 B]
=> => # Get:35 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 libstdc++-7-dev arm64 7.5.0-3ubuntu1~18.04 [1471 kB]
=> => # Get:36 http://ports.ubuntu.com/ubuntu-ports bionic-updates/main arm64 g++-7 arm64 7.5.0-3ubuntu1~18.04 [8077 kB]
```

```
jaewoo@maengjaeuui-MacBookAir docker_experiment % docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
snuspl/swpp	session1_with_npm	5d049e73a2eb	28 seconds ago	1.04GB
snuspl/swpp	session1	e61827f20122	4 hours ago	816MB

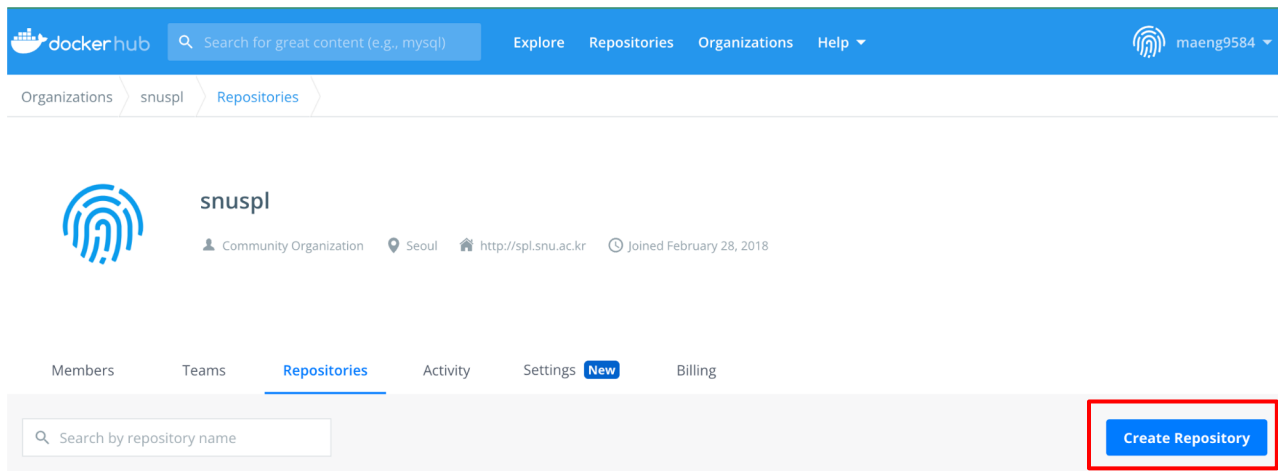
4. Docker Hub

- Remote storage that enables saving and sharing docker images!



4. Docker Hub

1. make an account at [docker hub](https://hub.docker.com/)
2. create a repository



4. Docker Hub

3. login with docker hub account in terminal

```
$ docker login
```

4. push local image

```
$ docker push {docker_account}/{repository_name}:{tag}
```

```
# e.g.
```

```
$ docker push jaewoo/swpp:session1_with_npm
```