1st Week

수업은 이렇게 진행하려구요

▷ 일단 2주는 제가 발표하겠습니다.

▷ 계속 제가 주도하는 방식으로 할지, 나눠서 발표하는 방식으로 할지는 계속 고민해보겠습니다.

▷ 로컬 실습 환경은 꼭 구성해서 직접 다루면서 공부할 수 있으면 좋겠습니다.

▷ 기본적으로는 매주 월요일 18시 30분 ~ 20시 (1시간 30분) 정도의 스터디 시간으로 하겠습니다.

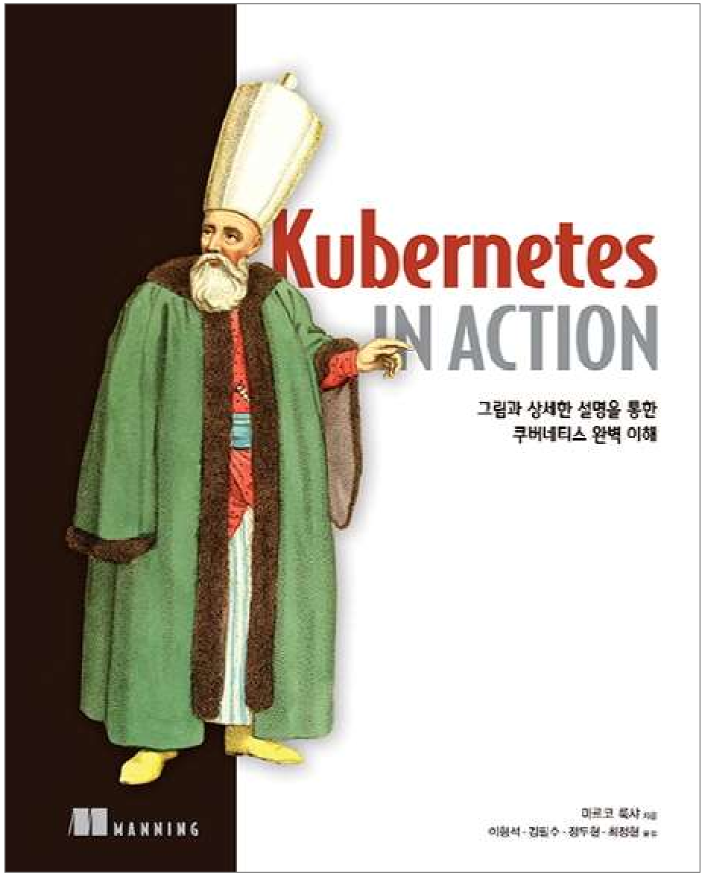
▷ Camera On, Mic On은 은 필수 !!!!

▷ 바쁘시겠지만 소수 정예이니, 빠지지 않도록 해주시면 고맙겠습니다 !!!

▷ 질문을 창피해 하지 말아요. 같이 소통하면서 알찬 수업을 만들어요 !!!

▷ 공부도 중요하지만, 우리 서로의 인맥이 되어봐요 !!!

Books


Kubernetes
IN ACTION
그림과 상세한 설명을 통한
쿠버네티스 완벽 이해
MANNING
마르코 룩샤 지음
이완석 · 김필수 · 정두현 · 최정현 옮김

but …

Our goal is not to be **~~user~~**,

to be **administrator** of **K8s**

///

Study Environment

S/W Development Environment

It's Linux !!

///

Container

A Brief History of Containers: From the 1970s Till Now

1979: Unix V7 – chroot 도입

2000: FreeBSD Jails – 서비스와 고객 서비스를 구분하기 위해 여러 개의 독립적이고 작은 시스템(jails)으로 분할

2001: Linux VServer - Jails와 유사하게, 리소스(파일 시스템, 네트워크 주소, 메모리)를 분할 할 수 있는 운영 체제 가상화를 Linux 커널 패치로 구현

2004: Solaris Containers – 첫 번째 공개 베타 출시

2005: Open VZ (Open Virtuzzo) - 가상화, 격리, 리소스 관리 및 체크 포인트를 위해 패치 된 Linux 커널을 사용하는 Linux 용 운영 체제 수준의 가상화 기술

2006: Process Containers - 2006년 Google 출시. 리소스 사용량(CPU, Mem, Disk I/O, NW)을 제한, 계산 및 격리하도록 설계. 1년 후 "cgroups"으로 이름 변경.

2008: LXC (LinuX Containers) – 컨테이너 관리자의 가장 완벽한 최초 구현. cgroups & namespace를 사용하여 구현.

2011: Warden – CloudFoundry에서 초기는 LXC를 사용하고 나중에 자체 구현으로 대체. cgroups, namespace 및 프로세스 수명주기 관리 서비스 포함.

2013: LMCTFY (Let Me Contain That For You) – Linux 애플리케이션 컨테이너를 제공하는 Google 컨테이너 스택의 오픈 소스 버전. 2015년 중단.

2013: Docker - 컨테이너 인기 폭발. 초기 단계 LXC 사용, 추후 자체 라이브러리 libcontainer로 대체.

2014: Kubernetes (Google)

2015: Kubernetes to CNCF

2016: The Importance of Container Security Is Revealed – DevSecOps

2017: Container Tools Become Mature – 컨테이너 도구의 성숙

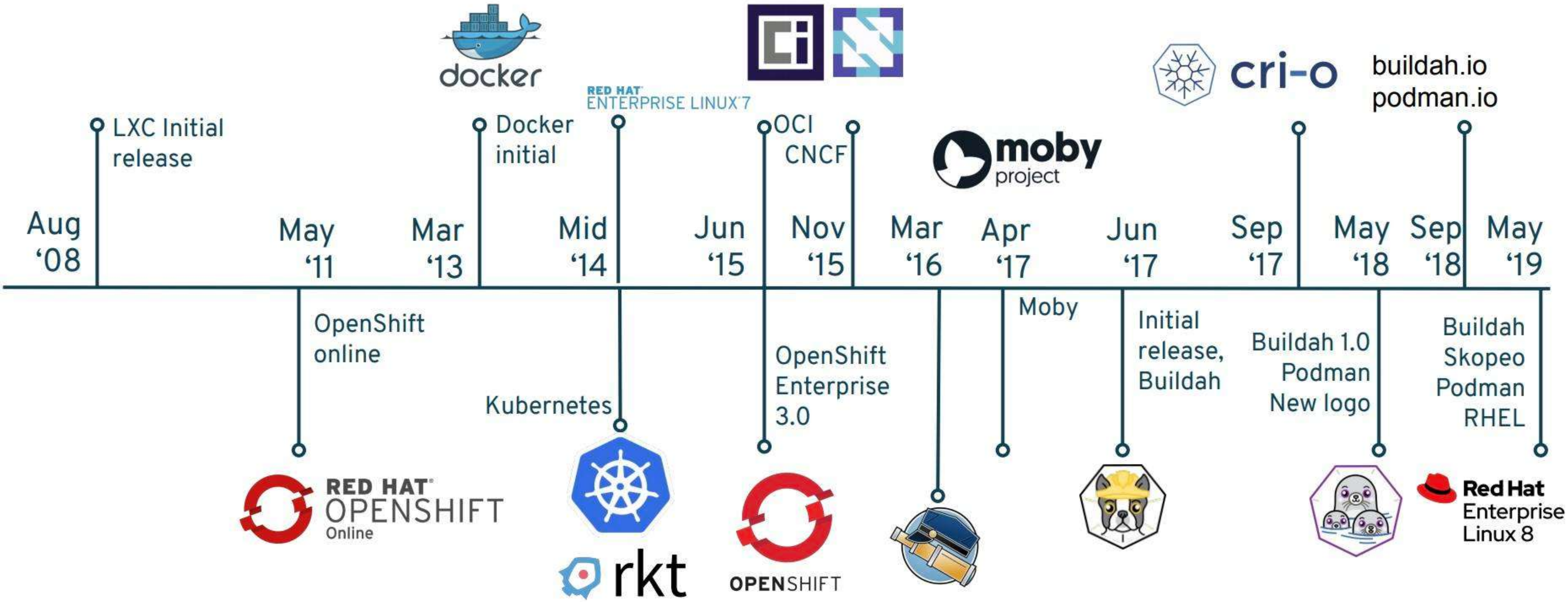
2017: containerd to CNCF (Docker)

2018: The Gold Standard – 시장 표준

2019: A Shifting Landscape - 변화

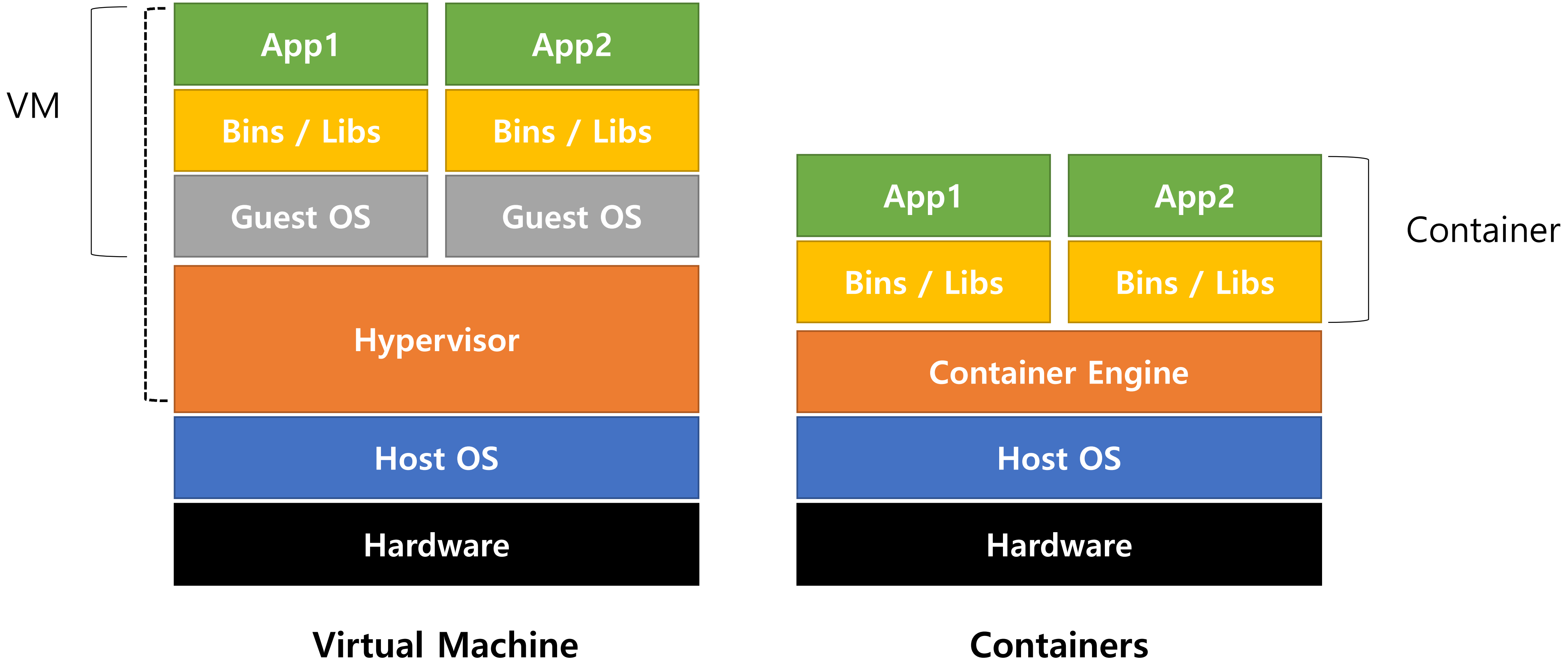
※ 참고 : https://blog.aquasec.com/a-brief-history-of-containers-from-1970s-chroot-to-docker-2016

Evolution of the open-source container

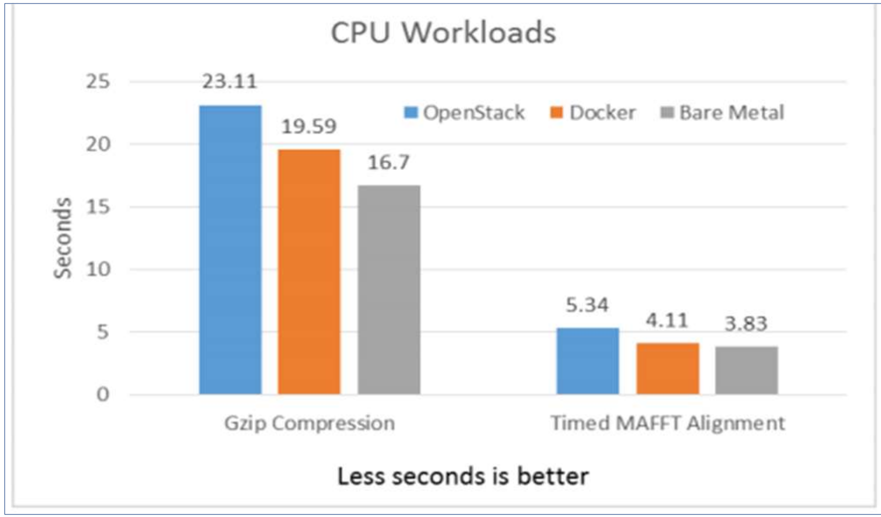


※ 참고 : https://developer.ibm.com/tutorials/multi-architecture-cri-o-container-images-for-red-hat-openshift/

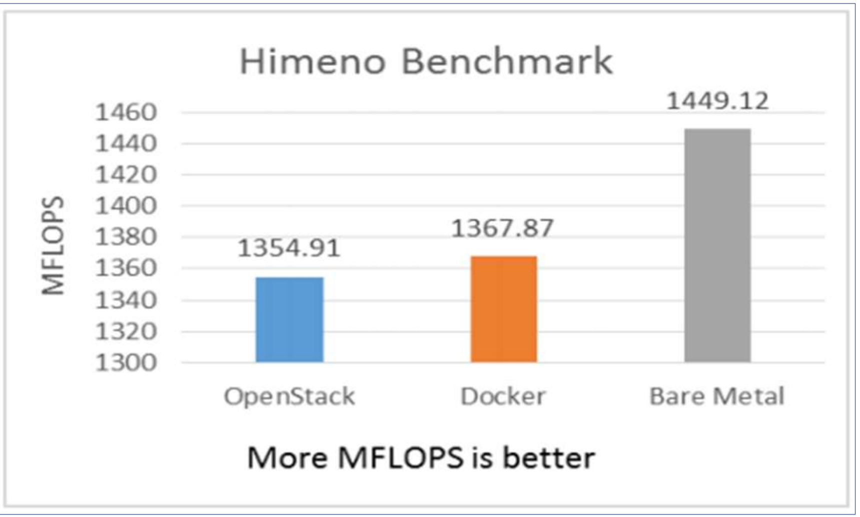
Virtual Machine vs. Containers



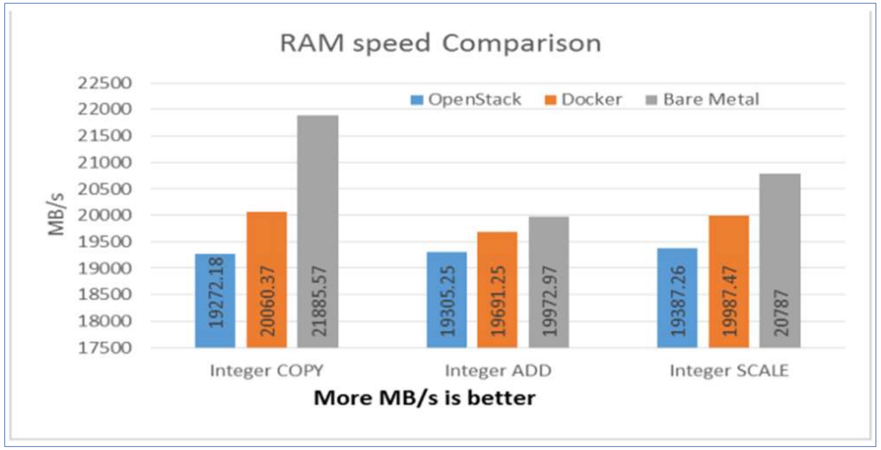
Docker Container vs. Openstack VM vs. Bare Metal Server


CPU Workloads

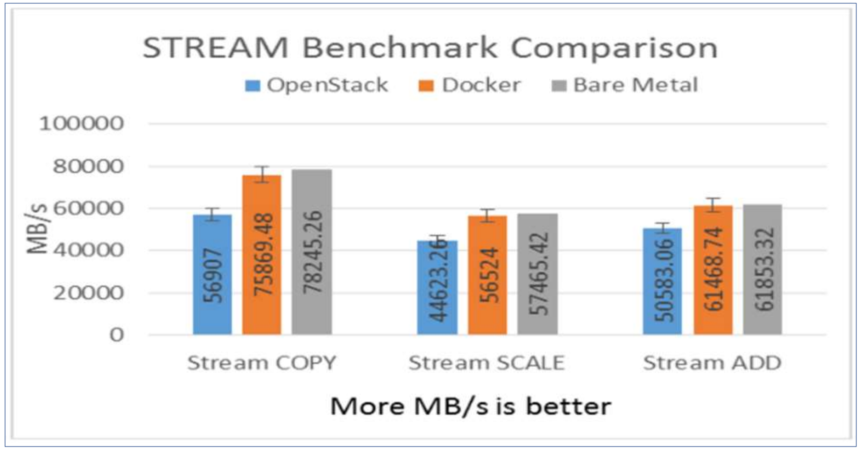
Less seconds is better


Himeno Benchmark

More MFLOPS is better


RAM speed Comparison

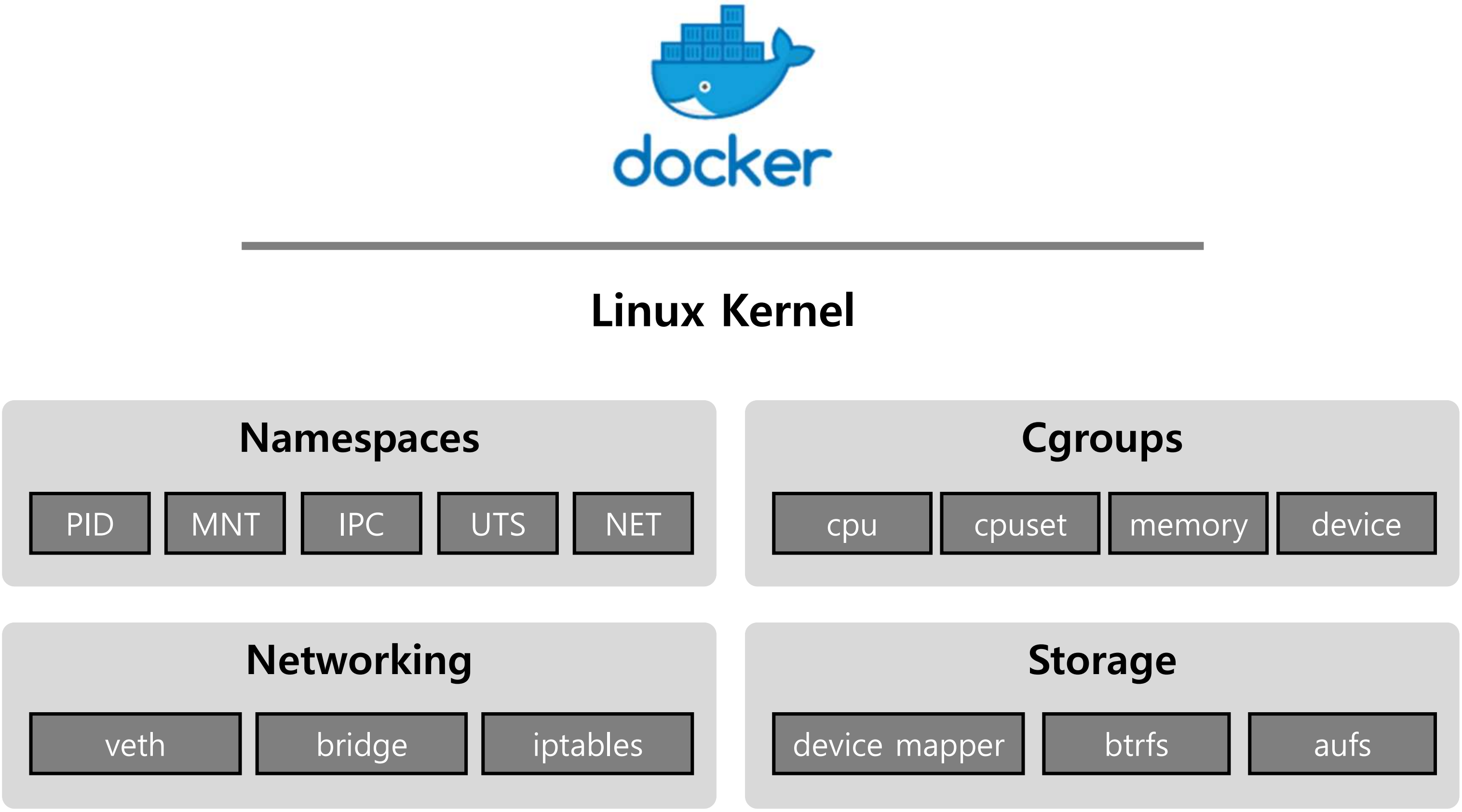
More MB/s is better


STREAM Benchmark Comparison

More MB/s is better

※ 참고 : http://ijeecs.iaescore.com/index.php/IJEECS/article/view/7925

Docker with Kernel



docker

Linux Kernel

Namespaces

PID

MNT

IPC

UTS

NET

Cgroups

cpu

cpuset

memory

device

Networking

veth

bridge

iptables

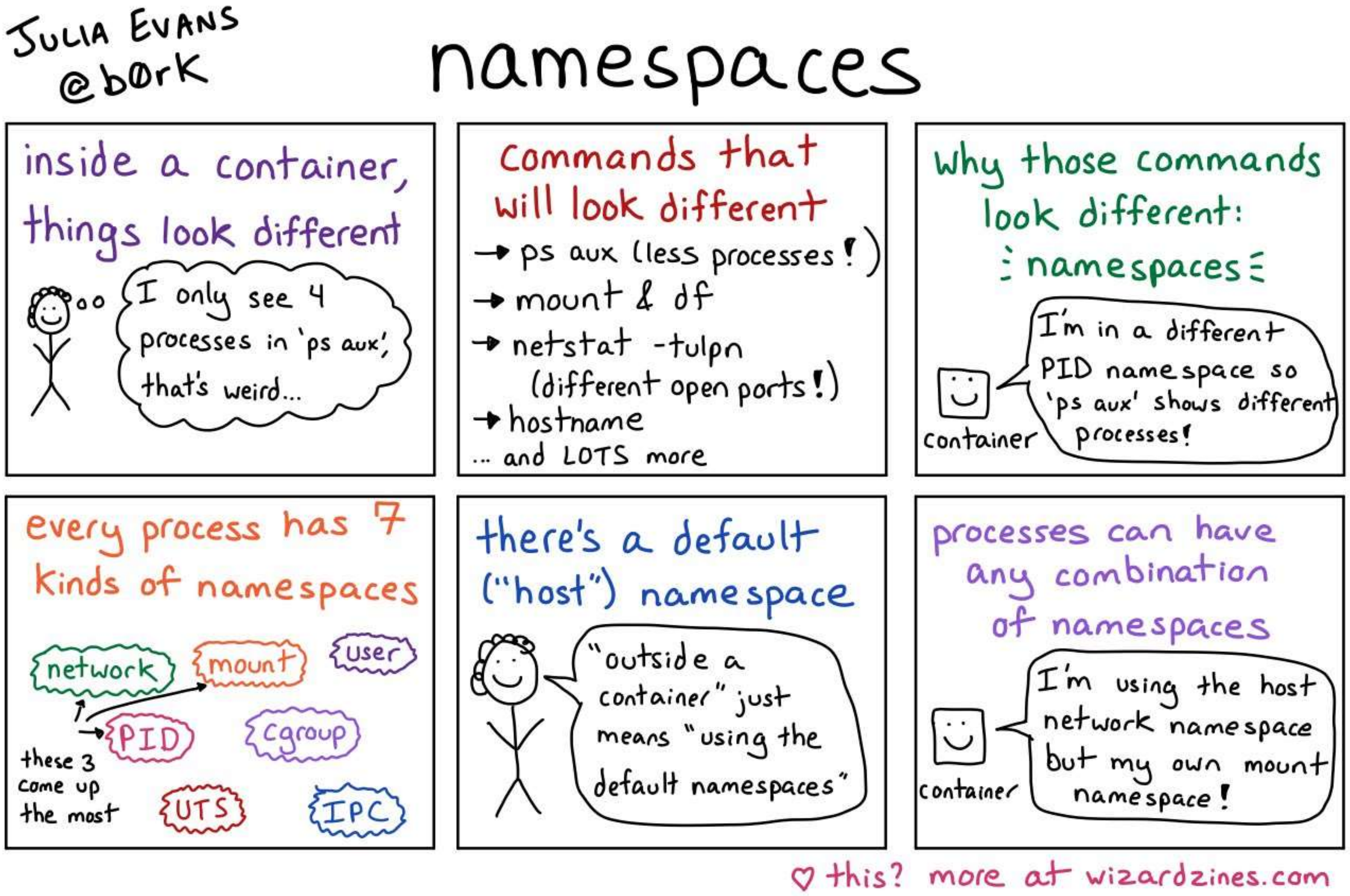
Storage

device mapper

btrfs

aufs

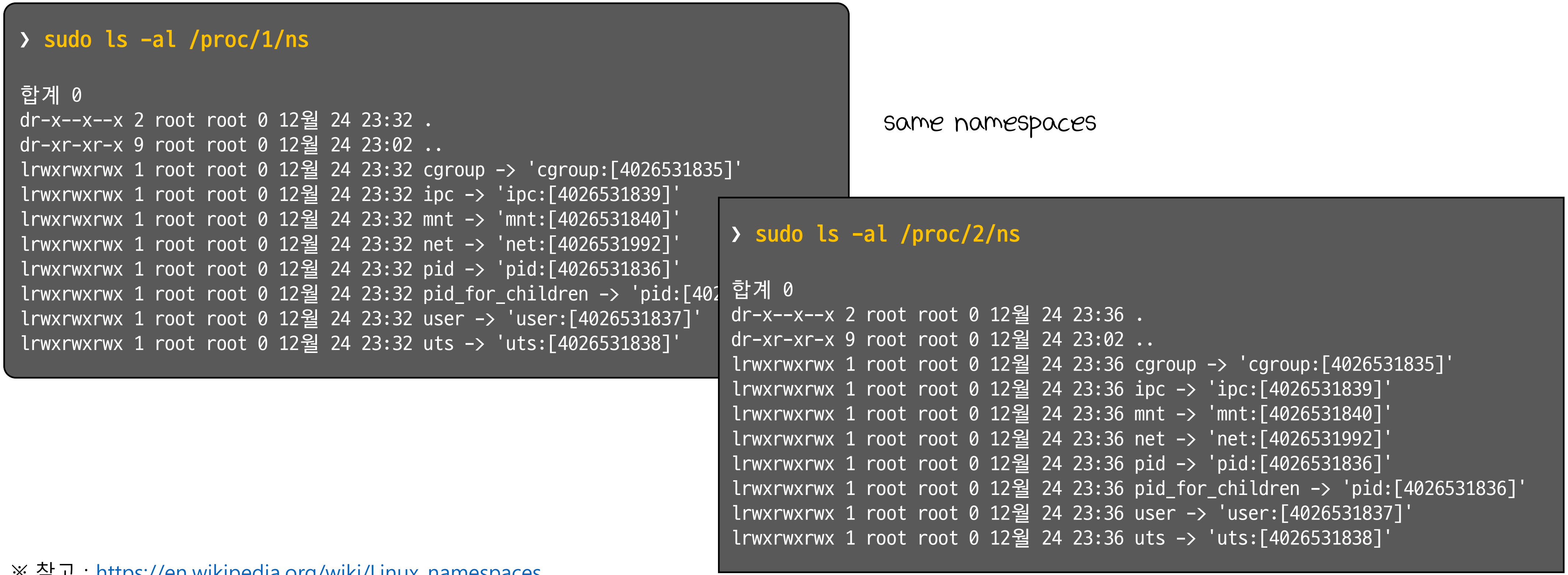
namespaces


JULIA EVANS
@b0rk
namespaces
inside a container, things look different
I only see 4 processes in 'ps aux', that's weird...
Commands that will look different
→ ps aux (less processes!)
→ mount & df
→ netstat -tulpn (different open ports!)
→ hostname
... and LOTS more
Why those commands look different: namespaces
I'm in a different PID namespace so 'ps aux' shows different processes!
container
every process has 7 kinds of namespaces
network
mount
user
PID
cgroup
UTS
IPC
these 3 come up the most
there's a default ("host") namespace
"outside a container" just means "using the default namespaces"
processes can have any combination of namespaces
I'm using the host network namespace but my own mount namespace!
container
♡ this? more at wizardzines.com

※ 참고 : https://twitter.com/b0rk/status/1240364585766576128

namespaces

▷ a feature of the Linux kernel that **partitions kernel resources** such that one set of processes sees one set of resources while another set of processes sees a different set of resources.



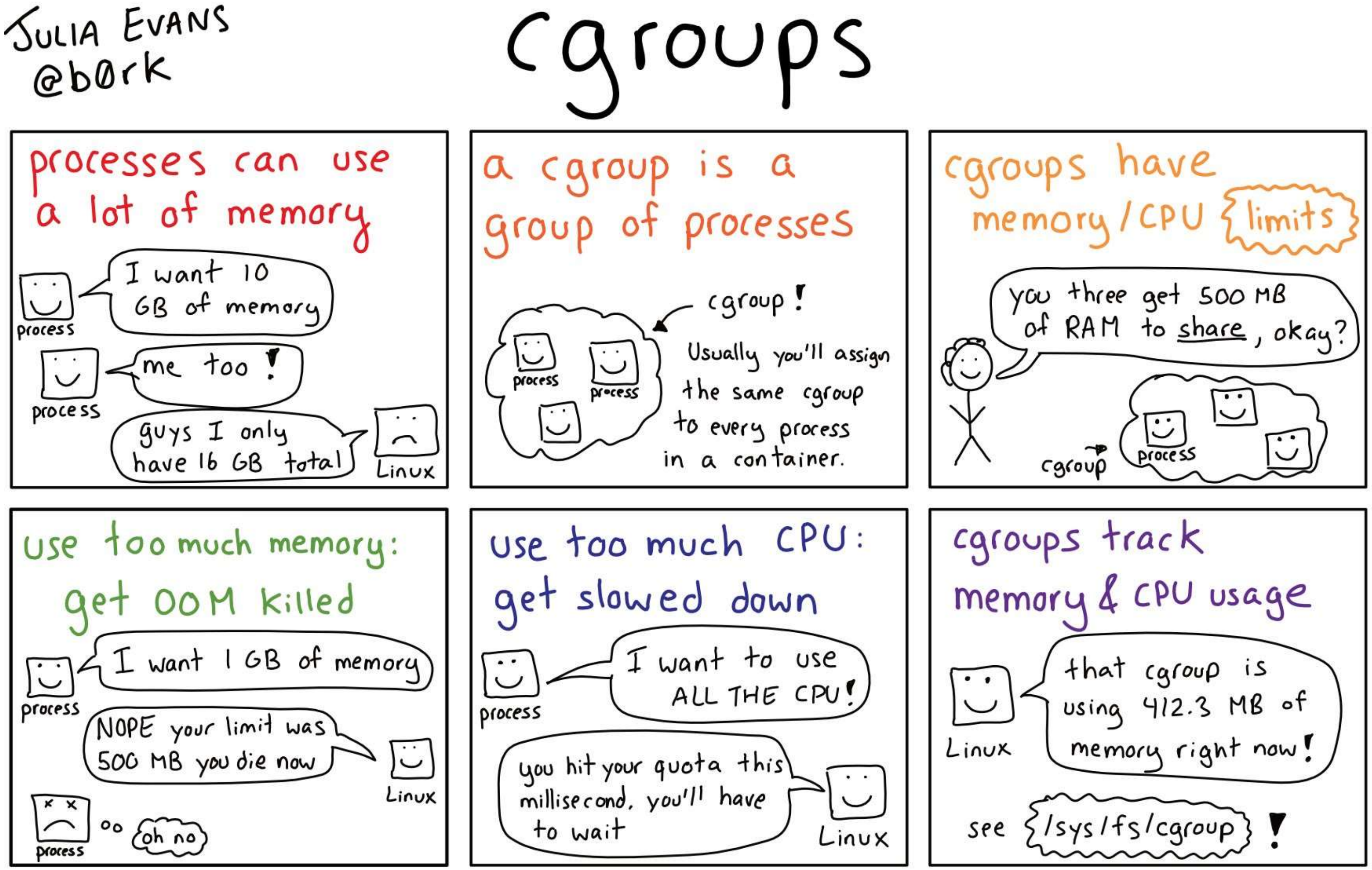
```
❯ sudo ls -al /proc/1/ns

합계 0
dr-x--x--x 2 root root 0 12월 24 23:32 .
dr-xr-xr-x 9 root root 0 12월 24 23:02 ..
lrwxrwxrwx 1 root root 0 12월 24 23:32 cgroup -> 'cgroup:[4026531835]'
lrwxrwxrwx 1 root root 0 12월 24 23:32 ipc -> 'ipc:[4026531839]'
lrwxrwxrwx 1 root root 0 12월 24 23:32 mnt -> 'mnt:[4026531840]'
lrwxrwxrwx 1 root root 0 12월 24 23:32 net -> 'net:[4026531992]'
lrwxrwxrwx 1 root root 0 12월 24 23:32 pid -> 'pid:[4026531836]'
lrwxrwxrwx 1 root root 0 12월 24 23:32 pid_for_children -> 'pid:[402
lrwxrwxrwx 1 root root 0 12월 24 23:32 user -> 'user:[4026531837]'
lrwxrwxrwx 1 root root 0 12월 24 23:32 uts -> 'uts:[4026531838]'
```

same namespaces

```
❯ sudo ls -al /proc/2/ns

합계 0
dr-x--x--x 2 root root 0 12월 24 23:36 .
dr-xr-xr-x 9 root root 0 12월 24 23:02 ..
lrwxrwxrwx 1 root root 0 12월 24 23:36 cgroup -> 'cgroup:[4026531835]'
lrwxrwxrwx 1 root root 0 12월 24 23:36 ipc -> 'ipc:[4026531839]'
lrwxrwxrwx 1 root root 0 12월 24 23:36 mnt -> 'mnt:[4026531840]'
lrwxrwxrwx 1 root root 0 12월 24 23:36 net -> 'net:[4026531992]'
lrwxrwxrwx 1 root root 0 12월 24 23:36 pid -> 'pid:[4026531836]'
lrwxrwxrwx 1 root root 0 12월 24 23:36 pid_for_children -> 'pid:[4026531836]'
lrwxrwxrwx 1 root root 0 12월 24 23:36 user -> 'user:[4026531837]'
lrwxrwxrwx 1 root root 0 12월 24 23:36 uts -> 'uts:[4026531838]'
```

※ 참고 : https://en.wikipedia.org/wiki/Linux_namespaces

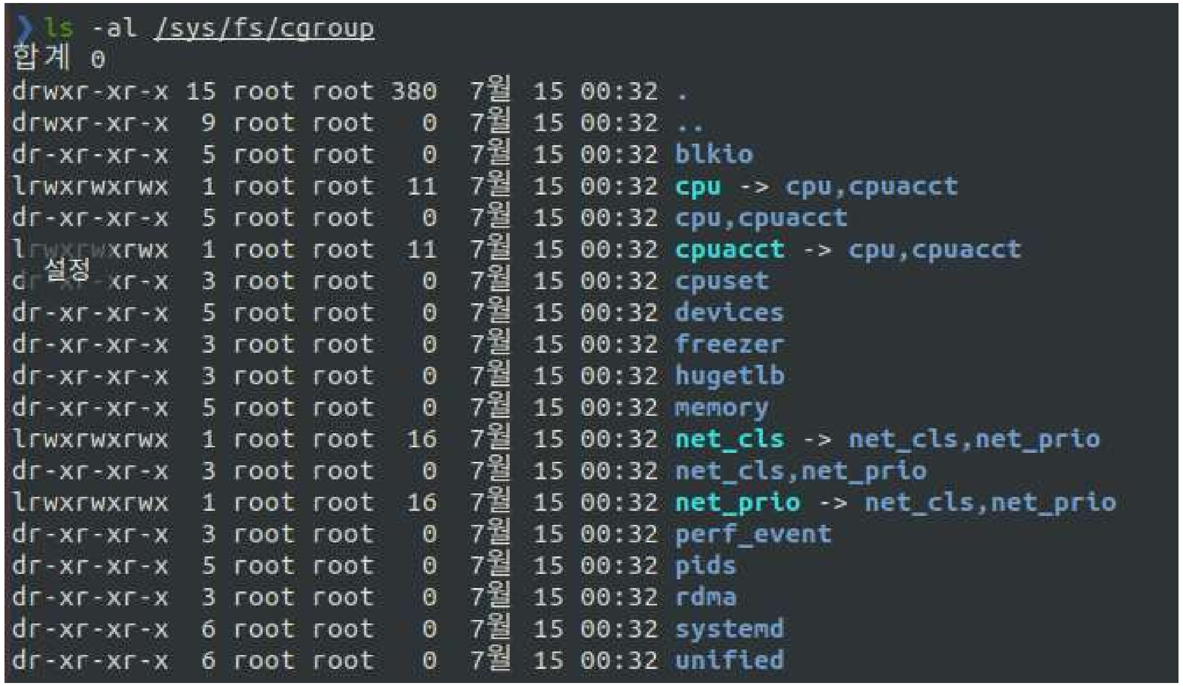
cgroups (control groups)


JULIA EVANS
@b0rk
cgroups
processes can use a lot of memory
I want 10 GB of memory
process
me too!
process
guys I only have 16 GB total
Linux
a cgroup is a group of processes
cgroup!
process
process
Usually you'll assign the same cgroup to every process in a container.
cgroups have memory/CPU limits
you three get 500 MB of RAM to share, okay?
cgroup
process
use too much memory: get OOM killed
I want 1 GB of memory
process
NOPE your limit was 500 MB you die now
Linux
process
oh no
use too much CPU: get slowed down
process
I want to use ALL THE CPU!
you hit your quota this millisecond, you'll have to wait
Linux
cgroups track memory & CPU usage
Linux
that cgroup is using 412.3 MB of memory right now!
see /sys/fs/cgroup!

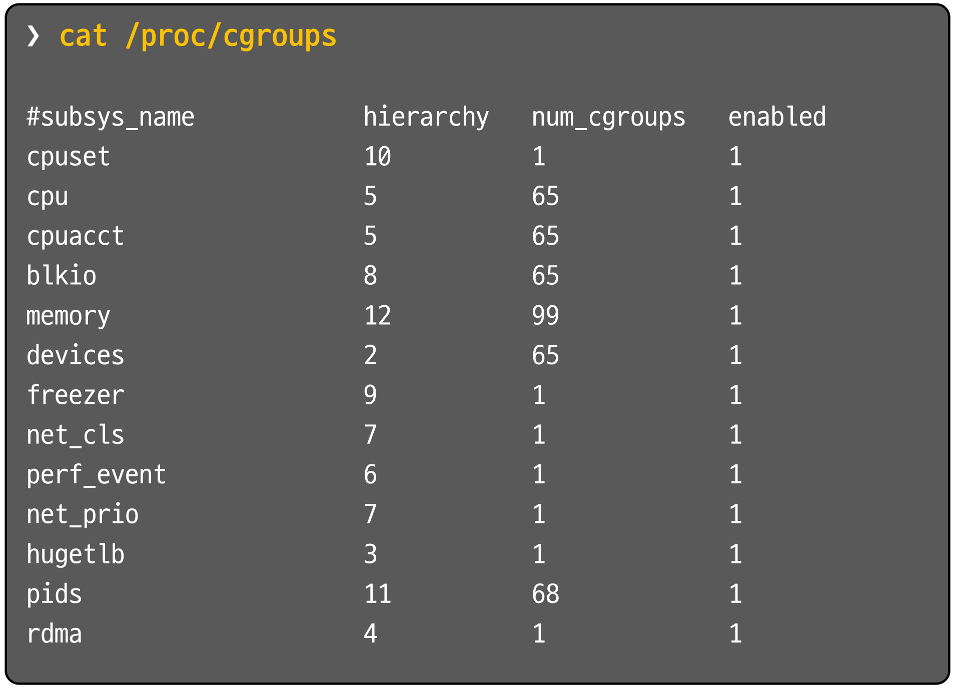
※ 참고 : https://twitter.com/b0rk/status/1214341831049252870

cgroups (control groups)

▷ Linux kernel feature that **limits**, accounts for, and isolates the resource usage (**CPU, memory, disk I/O, network**, etc.) of a **collection of processes**.



```
❯ ls -al /sys/fs/cgroup
합계 0
drwxr-xr-x 15 root root 380  7월  15 00:32 .
drwxr-xr-x  9 root root   0  7월  15 00:32 ..
dr-xr-xr-x  5 root root   0  7월  15 00:32 blkio
lrwxrwxrwx  1 root root  11  7월  15 00:32 cpu -> cpu,cpuacct
dr-xr-xr-x  5 root root   0  7월  15 00:32 cpu,cpuacct
lrwxrwxrwx  1 root root  11  7월  15 00:32 cpuacct -> cpu,cpuacct
dr-xr-xr-x  3 root root   0  7월  15 00:32 cpuset
dr-xr-xr-x  5 root root   0  7월  15 00:32 devices
dr-xr-xr-x  3 root root   0  7월  15 00:32 freezer
dr-xr-xr-x  3 root root   0  7월  15 00:32 hugetlb
dr-xr-xr-x  5 root root   0  7월  15 00:32 memory
lrwxrwxrwx  1 root root  16  7월  15 00:32 net_cls -> net_cls,net_prio
dr-xr-xr-x  3 root root   0  7월  15 00:32 net_cls,net_prio
lrwxrwxrwx  1 root root  16  7월  15 00:32 net_prio -> net_cls,net_prio
dr-xr-xr-x  3 root root   0  7월  15 00:32 perf_event
dr-xr-xr-x  5 root root   0  7월  15 00:32 pids
dr-xr-xr-x  3 root root   0  7월  15 00:32 rdma
dr-xr-xr-x  6 root root   0  7월  15 00:32 systemd
dr-xr-xr-x  6 root root   0  7월  15 00:32 unified
```



```
❯ cat /proc/cgroups

#subsys_name    hierarchy   num_cgroups   enabled
cpuset          10          1             1
cpu             5           65            1
cpuacct         5           65            1
blkio           8           65            1
memory          12          99            1
devices         2           65            1
freezer         9           1             1
net_cls         7           1             1
perf_event      6           1             1
net_prio        7           1             1
hugetlb         3           1             1
pids            11          68            1
rdma            4           1             1
```

※ 참고 : https://en.wikipedia.org/wiki/Cgroups

Container Network Model (CNM)

▷ Sandbox

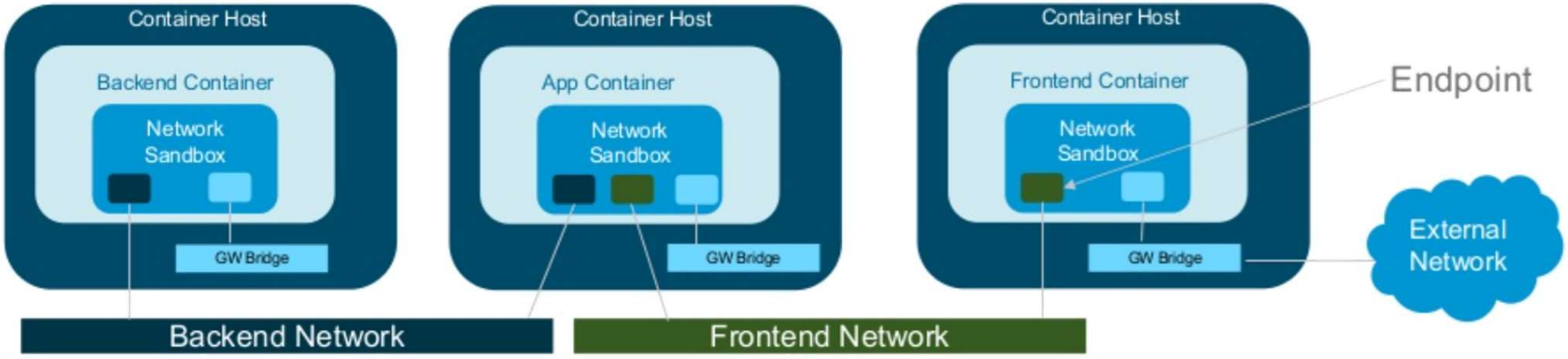
- A Sandbox contains the configuration of a container’s network stack.
- This includes management of the container's interfaces, routing table and DNS settings.
- An implementation of a Sandbox could be a Linux Network Namespace, a FreeBSD Jail or other similar concept.

▷ Endpoint

- An Endpoint joins a Sandbox to a Network.
- An implementation of an Endpoint could be a veth pair, an Open vSwitch internal port or similar

▷ Network

- A Network is a group of Endpoints that are able to communicate with each-other directly.
- An implementation of a Network could be a VXLAN Segment, a Linux bridge, a VLAN, etc.

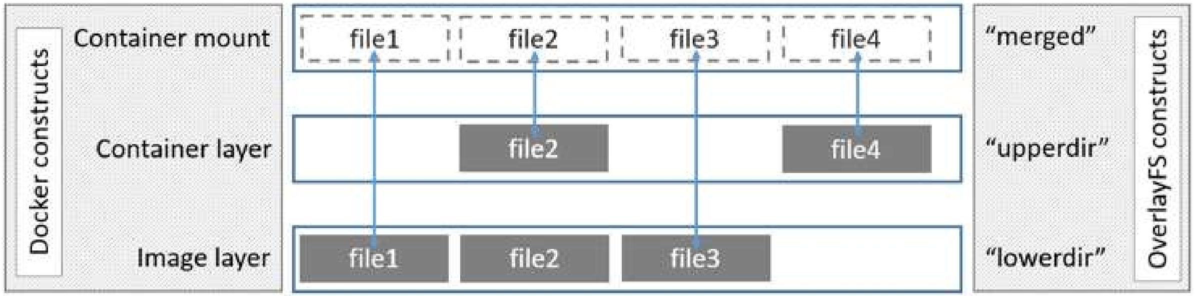


※ 참고 : https://www.slideshare.net/OpenNetworkingSummit/container-networking-deep-dive

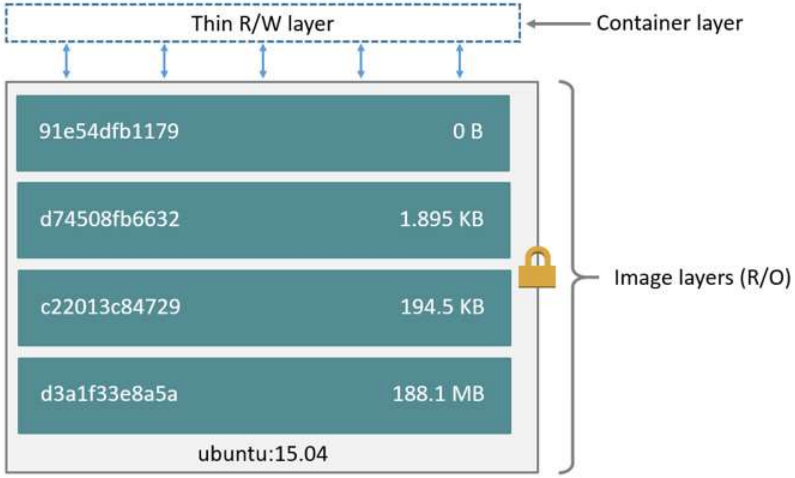
Container Storage Driver

▷ Container supports the following storage drivers

- **overlay2** : 기본 드라이버
- **aufs** : Docker 18.06 및 이전 버전에서 사용
- **fuse-overlayfs** : Rootless 지원 안되는 호스트에서 Rootless Docker를 사용할 때
- **devicemapper** : production 환경을 위해서는 direct-lvm 필요.
- **btrfs** and **zfs** : “snapshots” 같은 고급 기능을 지원하지만 설치와 유지보수가 까다로움.
- **vfs** : 테스트 목적으로만 사용하는 것을 권장



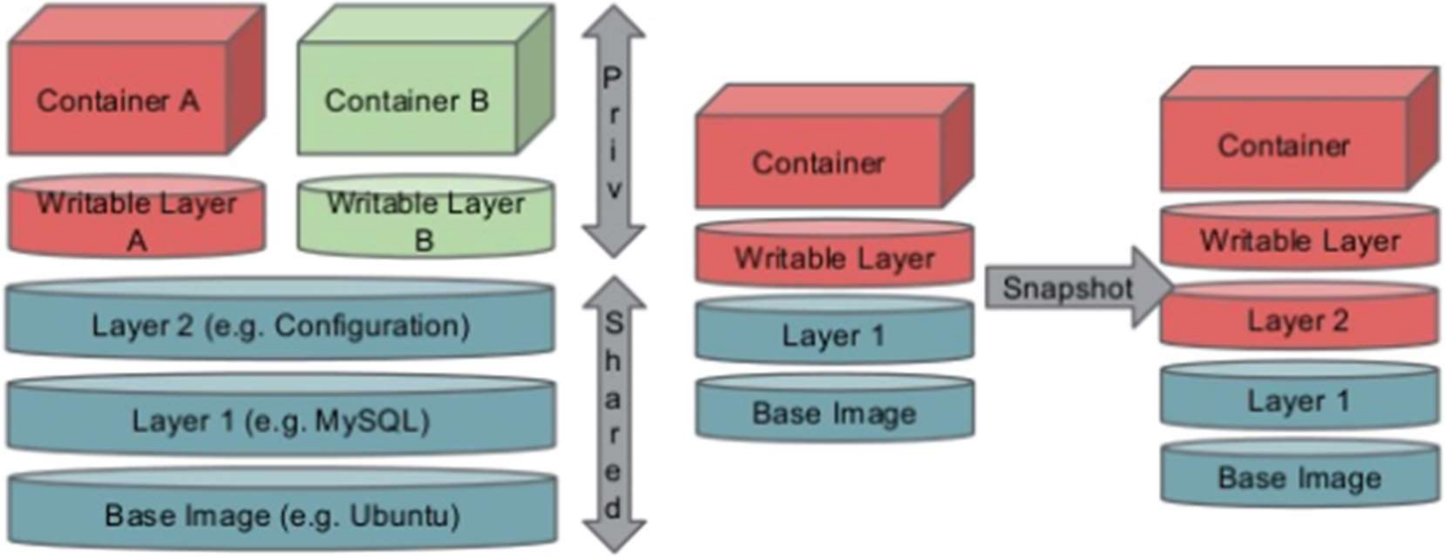
How the overlay driver works



Container
(based on ubuntu:15.04 image)

※ 참고 : https://docs.docker.com/storage/storagedriver/overlayfs-driver/

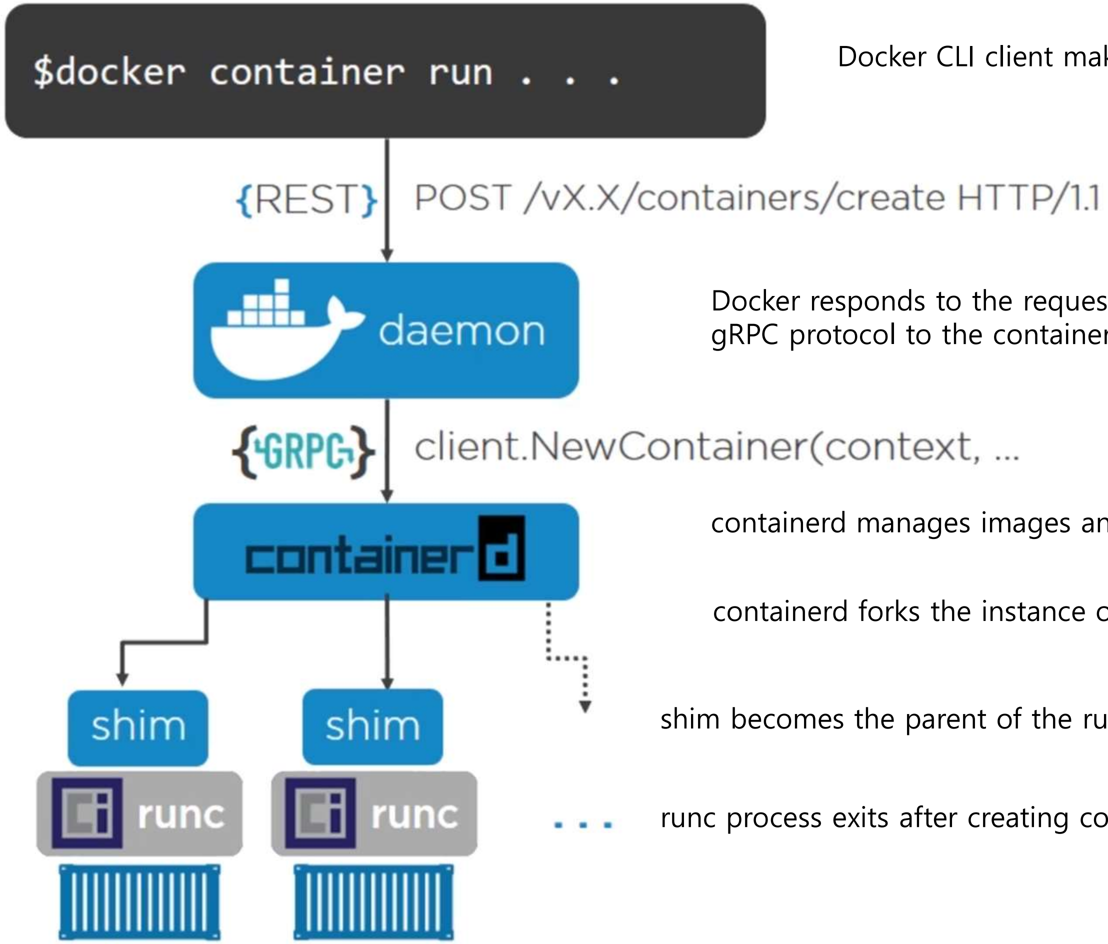
Container Storage Driver



※ 참고 : https://docs.docker.com/storage/storagedriver/overlayfs-driver/

Docker

Docker Engine Architecture



```
$docker container run . . .
```

{REST} POST /vX.X/containers/create HTTP/1.1

daemon

{GRPC} client.NewContainer(context, ...

containerd

shim

shim

runc

runc

. . .

Docker CLI client makes a REST API call to the Docker daemon server

Docker responds to the request and spins up a container by passing the message through gRPC protocol to the containerd process

containerd manages images and lifecycle of the containers: stop, pause, start, delete

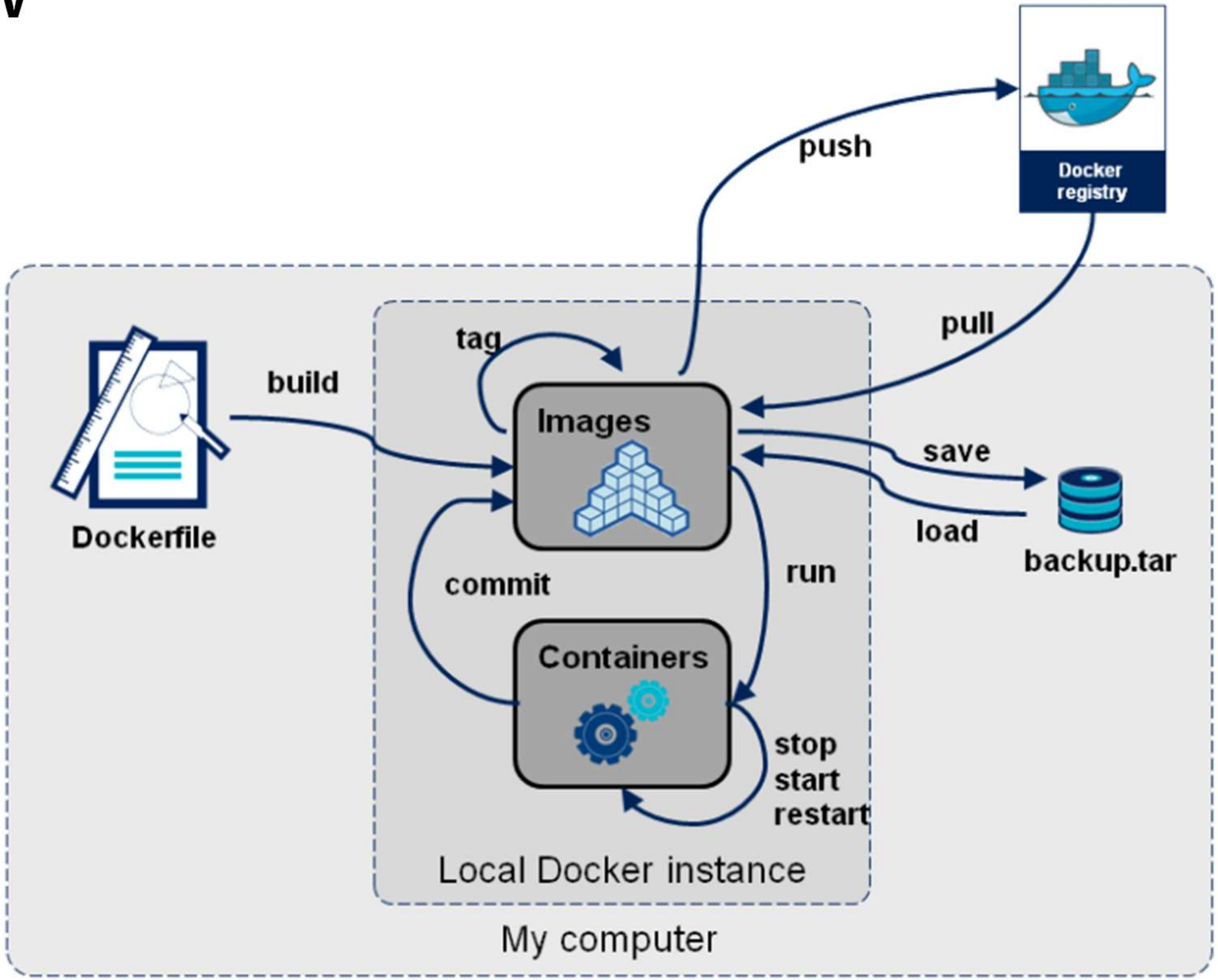
containerd forks the instance of runc for new containers

shim becomes the parent of the running container

runc process exits after creating container

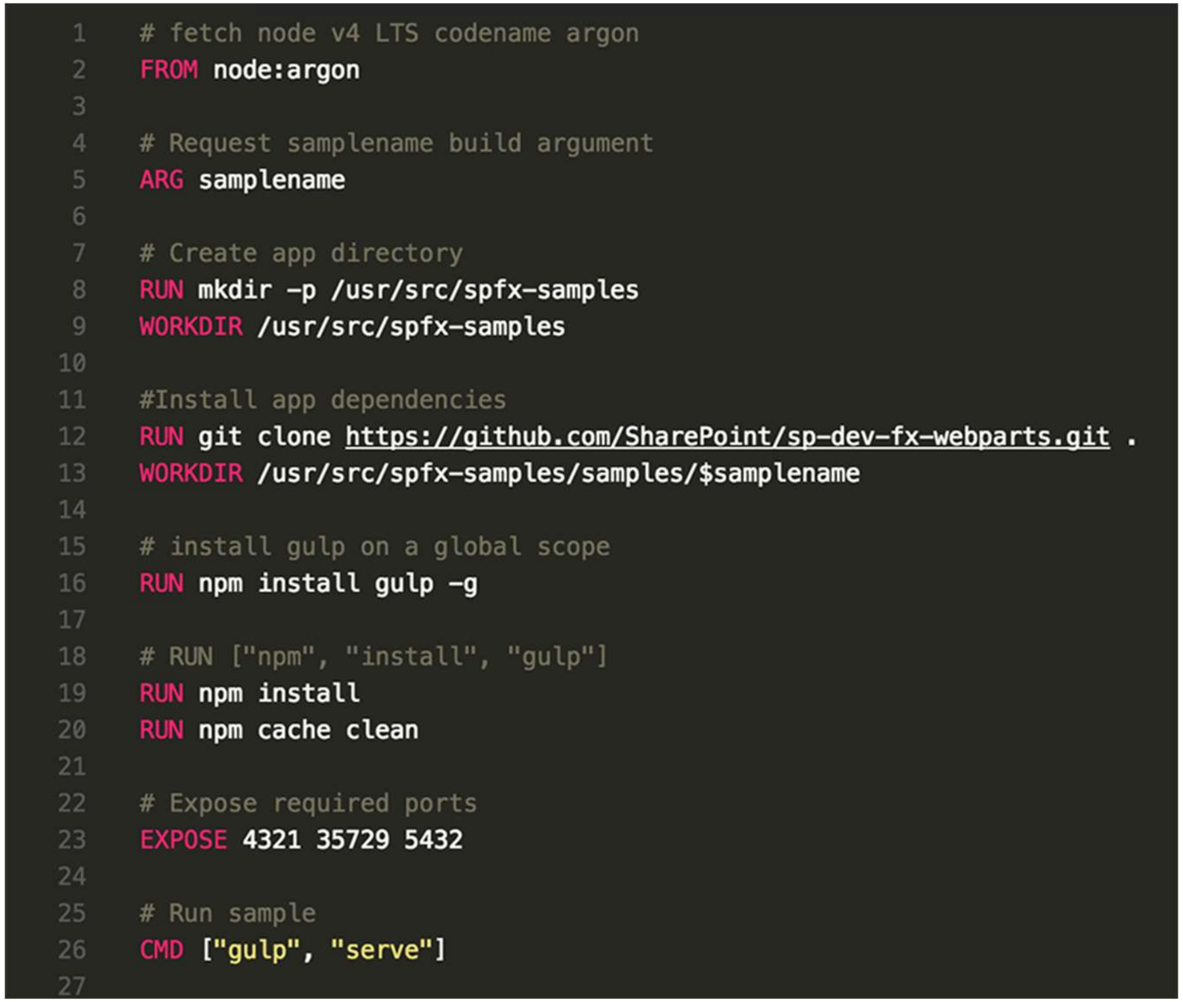
※ 참고 : https://betterprogramming.pub/docker-for-front-end-developers-c758a44e622f

Docker flow



※ 참고 : https://blog.wonizz.tk/2019/07/31/docker-dockerfile/

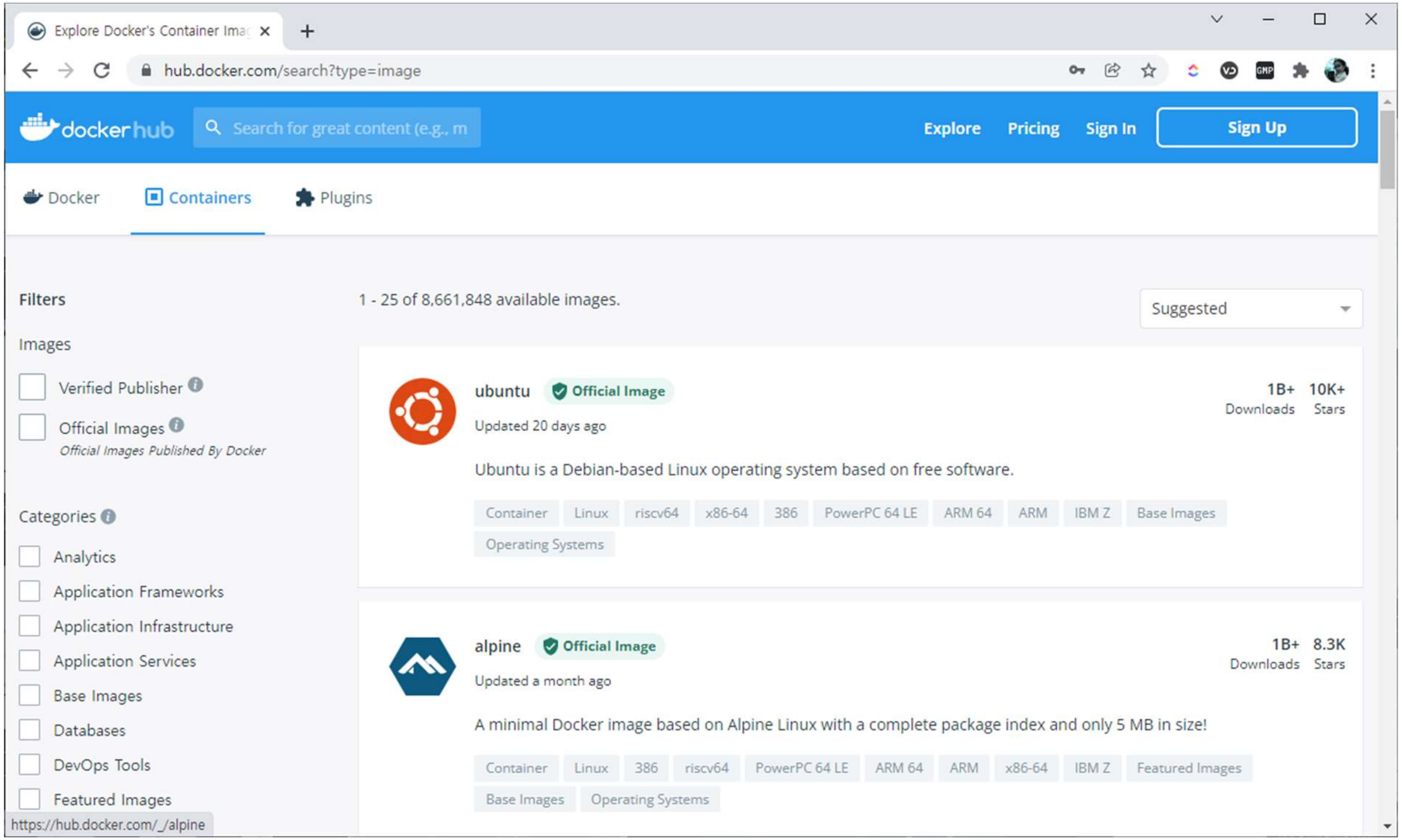
Dockerfile



```
# fetch node v4 LTS codename argon
FROM node:argon

# Request samplename build argument
ARG samplename

# Create app directory
RUN mkdir -p /usr/src/spfx-samples
WORKDIR /usr/src/spfx-samples

#Install app dependencies
RUN git clone https://github.com/SharePoint/sp-dev-fx-webparts.git .
WORKDIR /usr/src/spfx-samples/samples/$samplename

# install gulp on a global scope
RUN npm install gulp -g

# RUN ["npm", "install", "gulp"]
RUN npm install
RUN npm cache clean

# Expose required ports
EXPOSE 4321 35729 5432

# Run sample
CMD ["gulp", "serve"]

```

※ 참고 : https://n8d.at/how-to-run-sharepoint-pattern-and-practices-samples-through-docker/

Docker Hub (Registry)



Filters

Images

□ Verified Publisher

□ Official Images
Official Images Published By Docker

Categories

□ Analytics
□ Application Frameworks
□ Application Infrastructure
□ Application Services
□ Base Images
□ Databases
□ DevOps Tools
□ Featured Images

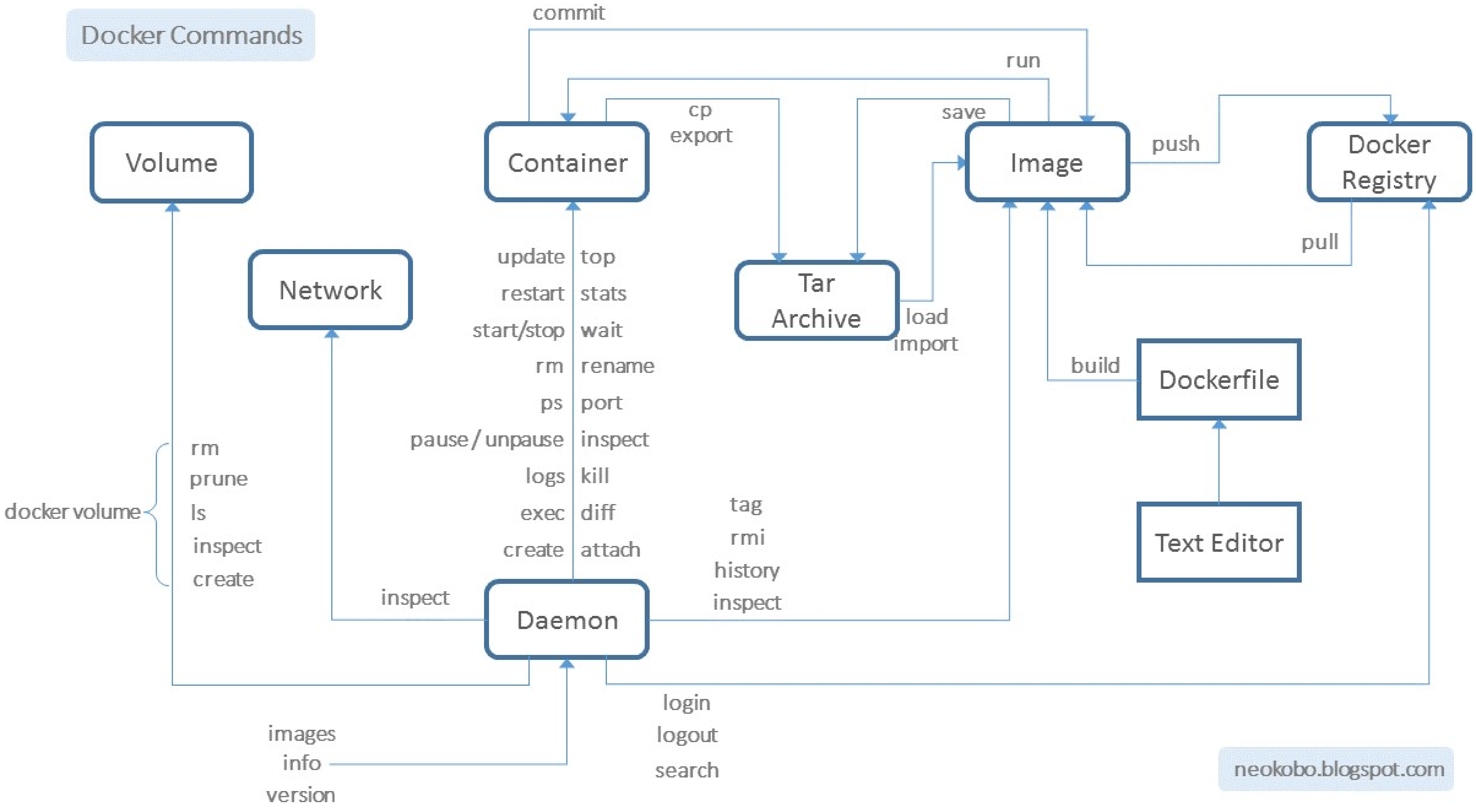
1 - 25 of 8,661,848 available images.

ubuntu Official Image — 1B+ Downloads, 10K+ Stars
Updated 20 days ago
Ubuntu is a Debian-based Linux operating system based on free software.

alpine Official Image — 1B+ Downloads, 8.3K Stars
Updated a month ago
A minimal Docker image based on Alpine Linux with a complete package index and only 5 MB in size!

※ 참고 : https://hub.docker.com/

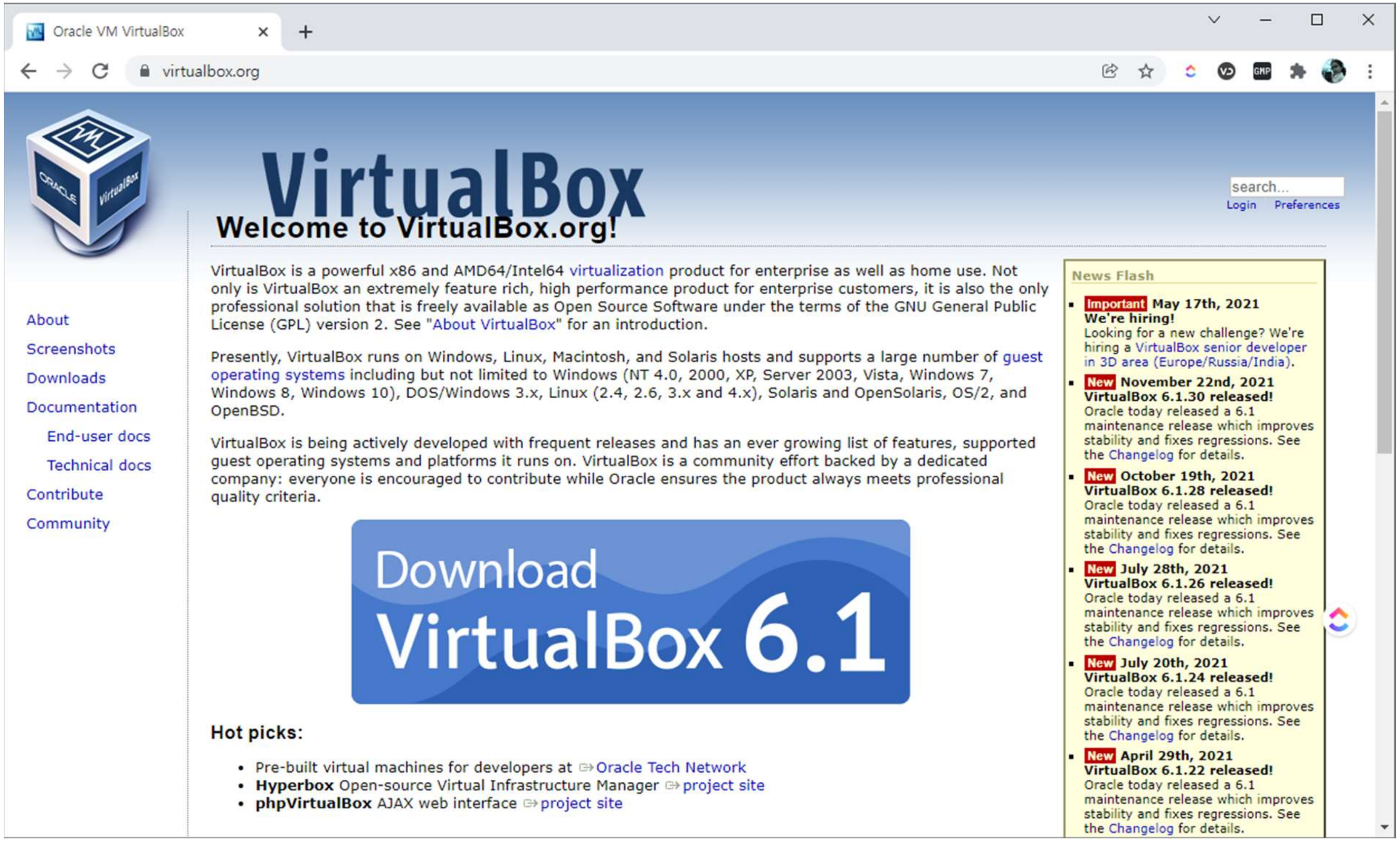
Docker command



※ 참고 : http://neokobo.blogspot.com/2017/12/docker-command-flowchart.html

Docker
Hands-On

VirtualBox Install



VirtualBox

Welcome to VirtualBox.org!

VirtualBox is a powerful x86 and AMD64/Intel64 virtualization product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2. See "About VirtualBox" for an introduction.

Presently, VirtualBox runs on Windows, Linux, Macintosh, and Solaris hosts and supports a large number of guest operating systems including but not limited to Windows (NT 4.0, 2000, XP, Server 2003, Vista, Windows 7, Windows 8, Windows 10), DOS/Windows 3.x, Linux (2.4, 2.6, 3.x and 4.x), Solaris and OpenSolaris, OS/2, and OpenBSD.

VirtualBox is being actively developed with frequent releases and has an ever growing list of features, supported guest operating systems and platforms it runs on. VirtualBox is a community effort backed by a dedicated company: everyone is encouraged to contribute while Oracle ensures the product always meets professional quality criteria.

Download VirtualBox 6.1

Hot picks:

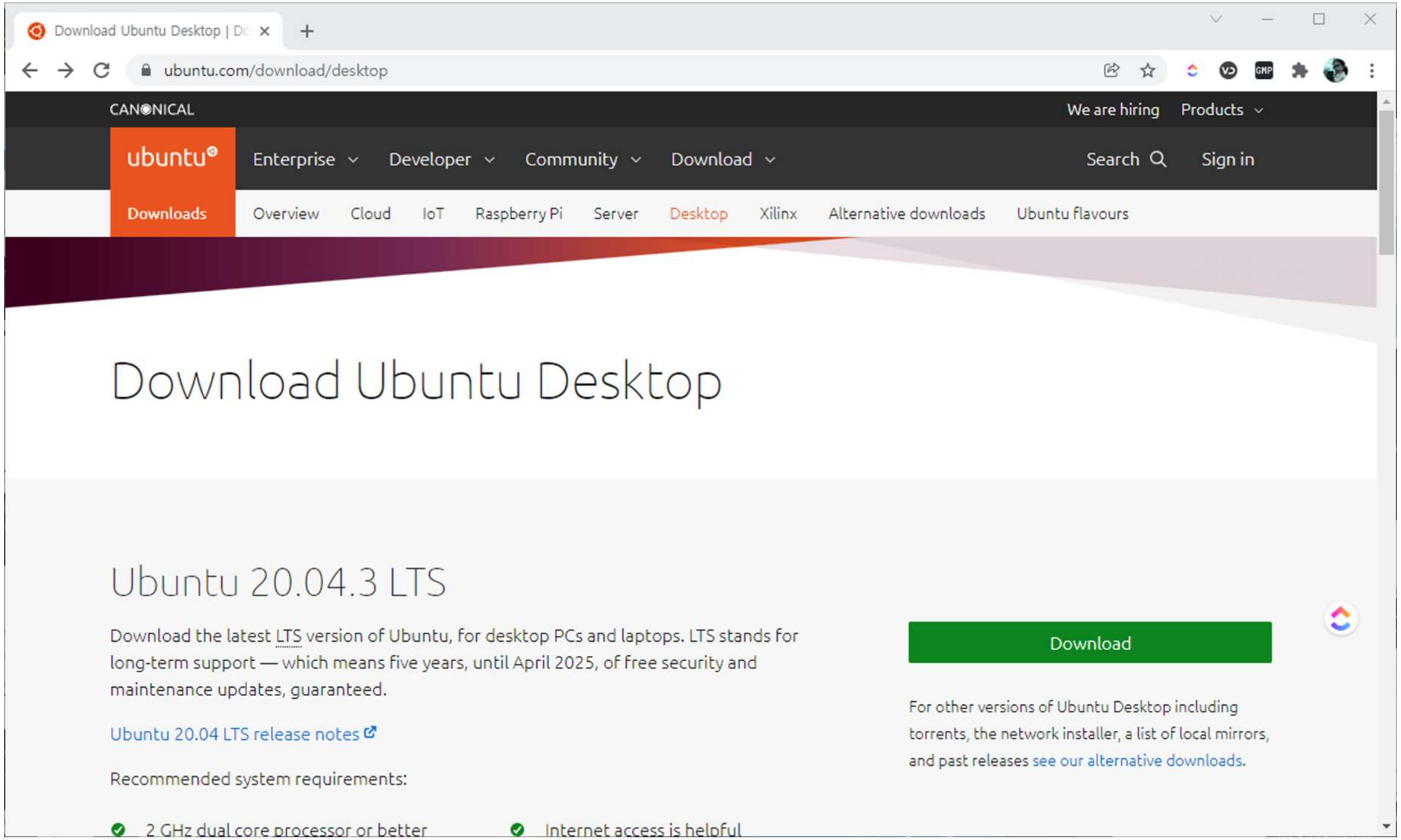
- Pre-built virtual machines for developers at ⇨Oracle Tech Network
- **Hyperbox** Open-source Virtual Infrastructure Manager ⇨project site
- **phpVirtualBox** AJAX web interface ⇨project site

News Flash

- Important **May 17th, 2021**
  We're hiring!
  Looking for a new challenge? We're hiring a VirtualBox senior developer in 3D area (Europe/Russia/India).
- New **November 22nd, 2021**
  VirtualBox 6.1.30 released!
  Oracle today released a 6.1 maintenance release which improves stability and fixes regressions. See the Changelog for details.
- New **October 19th, 2021**
  VirtualBox 6.1.28 released!
  Oracle today released a 6.1 maintenance release which improves stability and fixes regressions. See the Changelog for details.
- New **July 28th, 2021**
  VirtualBox 6.1.26 released!
  Oracle today released a 6.1 maintenance release which improves stability and fixes regressions. See the Changelog for details.
- New **July 20th, 2021**
  VirtualBox 6.1.24 released!
  Oracle today released a 6.1 maintenance release which improves stability and fixes regressions. See the Changelog for details.
- New **April 29th, 2021**
  VirtualBox 6.1.22 released!
  Oracle today released a 6.1 maintenance release which improves stability and fixes regressions. See the Changelog for details.

※ 참고 : https://www.virtualbox.org/

Ubuntu Install



Download Ubuntu Desktop

Ubuntu 20.04.3 LTS

Download the latest LTS version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support — which means five years, until April 2025, of free security and maintenance updates, guaranteed.

Ubuntu 20.04 LTS release notes

Recommended system requirements:

- 2 GHz dual core processor or better
- Internet access is helpful

For other versions of Ubuntu Desktop including torrents, the network installer, a list of local mirrors, and past releases see our alternative downloads.

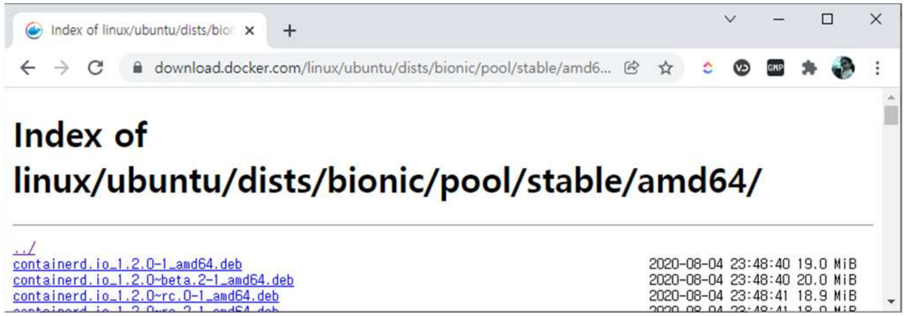
※ 참고 : https://ubuntu.com/download/desktop

Docker Install

Ubuntu 배포판 확인

```
❯ lsb_release -a

No LSB modules are available.
Distributor ID:         Ubuntu
Description:            Ubuntu 20.04.6 LTS
Release:    20.04
Codename:    focal
```

패키지 및 버전 확인


Index of linux/ubuntu/dists/bior
download.docker.com/linux/ubuntu/dists/bionic/pool/stable/amd6...

Index of linux/ubuntu/dists/bionic/pool/stable/amd64/

```
../
containerd.io_1.2.0-1_amd64.deb                    2020-08-04 23:48:40 19.0 MiB
containerd.io_1.2.0~beta.2-1_amd64.deb             2020-08-04 23:48:40 20.0 MiB
containerd.io_1.2.0~rc.0-1_amd64.deb               2020-08-04 23:48:41 18.9 MiB
```

https://download.docker.com/linux/ubuntu/dists/

```
❯ wget https://download.docker.com/linux/ubuntu/dists/focal/pool/stable/amd64/containerd.io_1.6.21-1_amd64.deb
❯ wget https://download.docker.com/linux/ubuntu/dists/focal/pool/stable/amd64/docker-ce-cli_23.0.6-1~ubuntu.20.04~focal_amd64.deb
❯ wget https://download.docker.com/linux/ubuntu/dists/focal/pool/stable/amd64/docker-ce_23.0.6-1~ubuntu.20.04~focal_amd64.deb

❯ sudo dpkg --install ./containerd.io_1.6.21-1_amd64.deb
❯ sudo dpkg --install ./docker-ce-cli_23.0.6-1~ubuntu.20.04~focal_amd64.deb
❯ sudo dpkg --install ./docker-ce_23.0.6-1~ubuntu.20.04~focal_amd64.deb

❯ sudo usermod -aG docker $USER

❯ docker --version
Docker version 23.0.6, build ef23cbc

❯ sudo reboot
```

docker build - OLD Style

index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Kubernetes</title>
</head>
<body>
  <h2>Hello, This is K8s World</h2>
</body>
</html>
```

Dockerfile

```
FROM nginx:latest

COPY ./index.html /usr/share/nginx/html/index.html
```

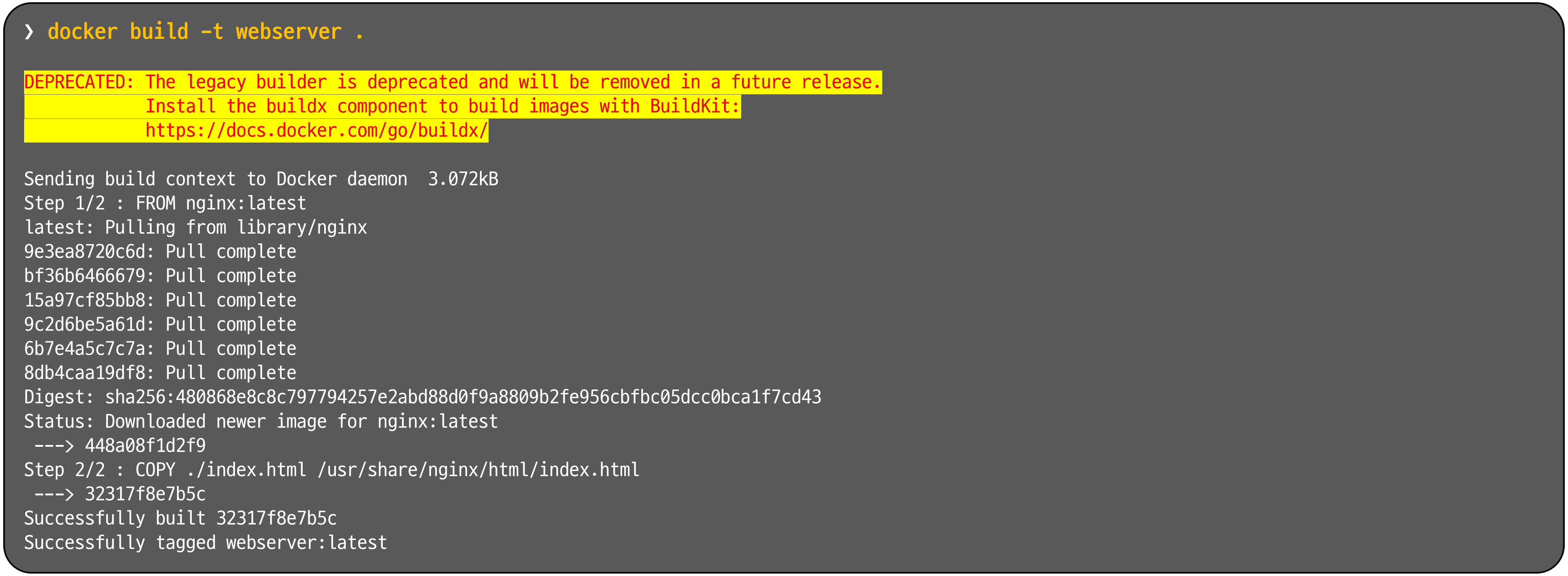
예제 파일 내려 받은 후 docker image 빌드

```
❯ git clone https://github.com/whatwant-school/kubernetes.git
❯ cd kubernetes/01_Container_and_Docker/hands-on

❯ docker build -t webserver .
...

❯ docker images

REPOSITORY    TAG       IMAGE ID       CREATED          SIZE
webserver     latest    6223db426adf   42 seconds ago   141MB
nginx         latest    f6987c8d6ed5   3 days ago       141MB
hello-world   latest    feb5d9fea6a5   3 months ago     13.3kB
```

※ 참고 : https://www.docker.com/blog/how-to-use-the-official-nginx-docker-image/

Deprecated ???

- docker build 했을 때 메시지를 자세히 보면... DEPRECATED 되었다고 나온다.
  . BuildKit 사용을 하라고 안내가 나온다.

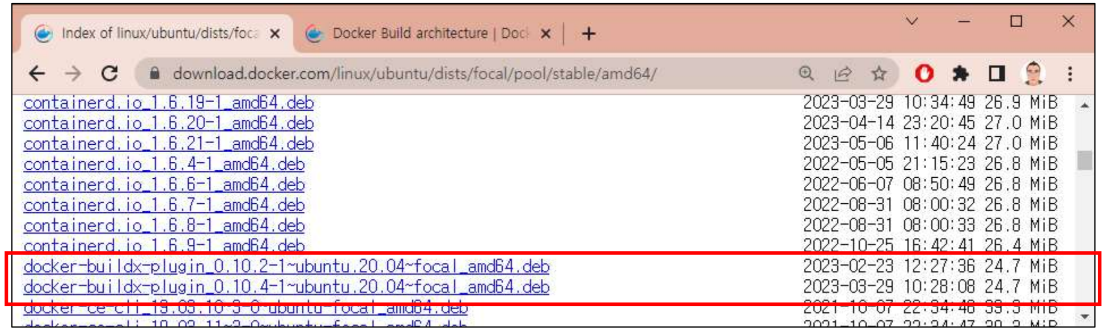


```
❯ docker build -t webserver .

DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  3.072kB
Step 1/2 : FROM nginx:latest
latest: Pulling from library/nginx
9e3ea8720c6d: Pull complete
bf36b6466679: Pull complete
15a97cf85bb8: Pull complete
9c2d6be5a61d: Pull complete
6b7e4a5c7c7a: Pull complete
8db4caa19df8: Pull complete
Digest: sha256:480868e8c8c797794257e2abd88d0f9a8809b2fe956cbfbc05dcc0bca1f7cd43
Status: Downloaded newer image for nginx:latest
 ---> 448a08f1d2f9
Step 2/2 : COPY ./index.html /usr/share/nginx/html/index.html
 ---> 32317f8e7b5c
Successfully built 32317f8e7b5c
Successfully tagged webserver:latest
```

※ 참고 : https://www.docker.com/blog/how-to-use-the-official-nginx-docker-image/

BuildKit Install

패키지 및 버전 확인


Index of linux/ubuntu/dists/foca
Docker Build architecture | Doc

download.docker.com/linux/ubuntu/dists/focal/pool/stable/amd64/

```
containerd.io_1.6.19-1_amd64.deb                                  2023-03-29 10:34:49 26.9 MiB
containerd.io_1.6.20-1_amd64.deb                                  2023-04-14 23:20:45 27.0 MiB
containerd.io_1.6.21-1_amd64.deb                                  2023-05-06 11:40:24 27.0 MiB
containerd.io_1.6.4-1_amd64.deb                                   2022-05-05 21:15:23 26.8 MiB
containerd.io_1.6.6-1_amd64.deb                                   2022-06-07 08:50:49 26.8 MiB
containerd.io_1.6.7-1_amd64.deb                                   2022-08-31 08:00:32 26.8 MiB
containerd.io_1.6.8-1_amd64.deb                                   2022-08-31 08:00:33 26.8 MiB
containerd.io_1.6.9-1_amd64.deb                                   2022-10-25 16:42:41 26.4 MiB
docker-buildx-plugin_0.10.2-1~ubuntu.20.04~focal_amd64.deb        2023-02-23 12:27:36 24.7 MiB
docker-buildx-plugin_0.10.4-1~ubuntu.20.04~focal_amd64.deb        2023-03-29 10:28:08 24.7 MiB
docker-ce-cli_19.03.10~3-0~ubuntu-focal_amd64.deb                 2021-10-07 22:34:48 39.3 MiB
```

https://download.docker.com/linux/ubuntu/dists/

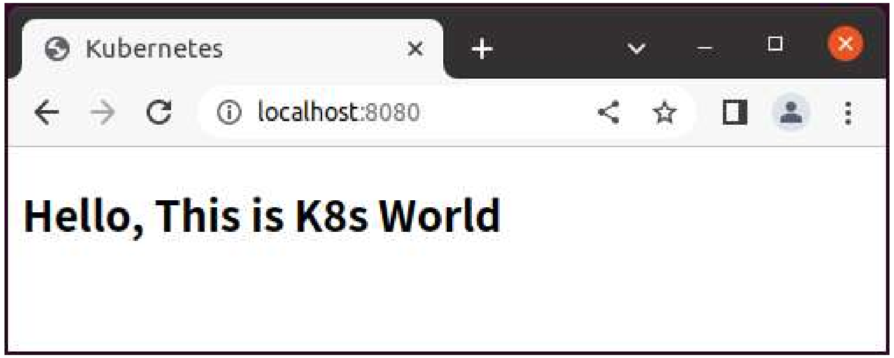
```
❯ wget https://download.docker.com/linux/ubuntu/dists/focal/pool/stable/amd64/docker-buildx-plugin_0.10.4-
1~ubuntu.20.04~focal_amd64.deb

❯ sudo dpkg --install ./docker-buildx-plugin_0.10.4-1~ubuntu.20.04~focal_amd64.deb

❯ docker buildx build -t webserver2 .

[+] Building 0.4s (7/7) FINISHED
 => [internal] load build definition from Dockerfile                                                   0.1s
 => => transferring dockerfile: 107B                                                                   0.0s
 => [internal] load .dockerignore                                                                      0.1s
 => => transferring context: 2B                                                                        0.0s
 => [internal] load metadata for docker.io/library/nginx:latest                                        0.0s
 => [internal] load build context                                                                      0.1s
...
```

Just do - docker run / ps

빌드한 이미지를 (container로) 실행하자

```
❯ docker run -it --rm -d -p 8080:80 --name web webserver

4db91157989678eb41d2bf9580a366617bdc3e0a814448848111a53bdc449871


❯ docker ps -al

CONTAINER ID   IMAGE        COMMAND                   CREATED          STATUS          PORTS                                    NAMES
4db911579896   webserver    "/docker-entrypoint.…"   23 seconds ago   Up 22 seconds   0.0.0.0:8080->80/tcp, :::8080->80/tcp    web
```

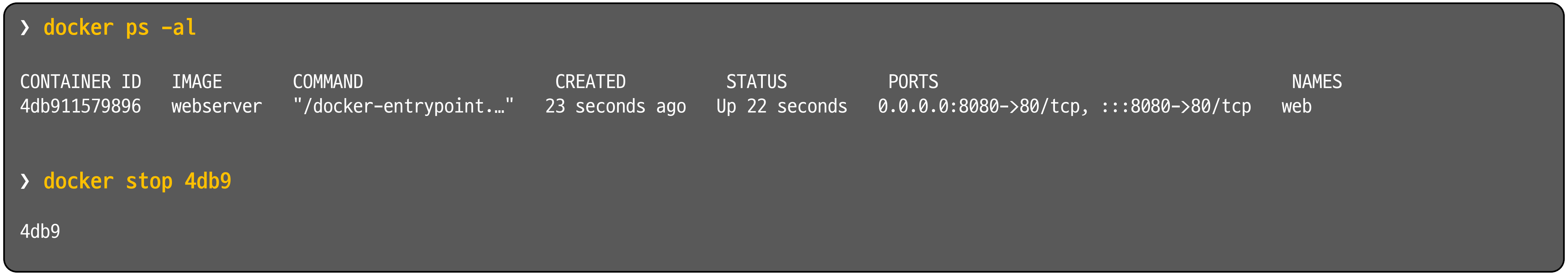
Chrome을 통해서 웹페이지 확인



Hello, This is K8s World

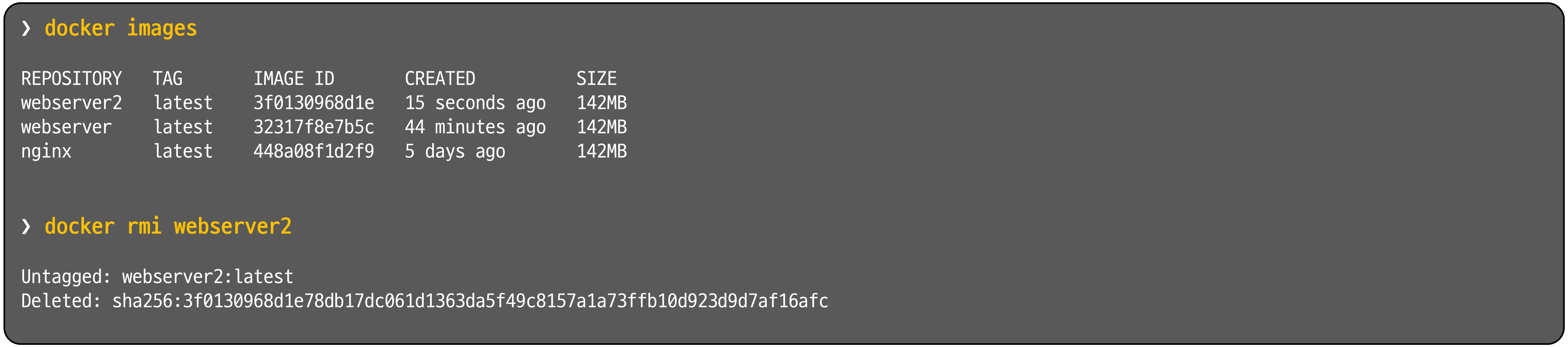
Just do - docker stop / images / rmi

동작하고 있는 container를 중단해보자



```
❯ docker ps -al

CONTAINER ID   IMAGE       COMMAND                  CREATED          STATUS          PORTS                                   NAMES
4db911579896   webserver   "/docker-entrypoint.…"   23 seconds ago   Up 22 seconds   0.0.0.0:8080->80/tcp, :::8080->80/tcp   web


❯ docker stop 4db9

4db9
```

등록되어 있는 image를 확인하고, 삭제해보자



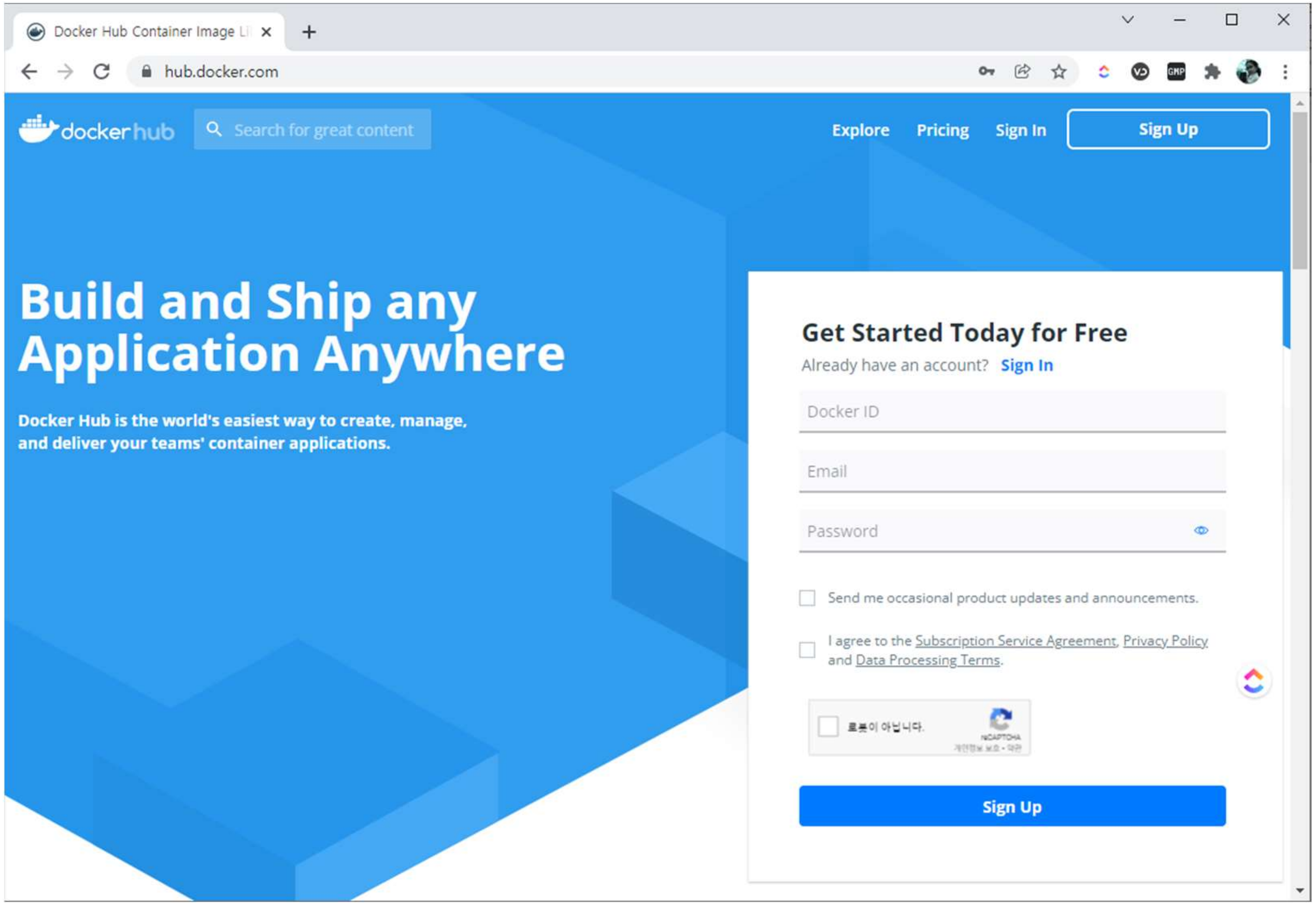
```
❯ docker images

REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
webserver2   latest    3f0130968d1e   15 seconds ago   142MB
webserver    latest    32317f8e7b5c   44 minutes ago   142MB
nginx        latest    448a08f1d2f9   5 days ago       142MB


❯ docker rmi webserver2

Untagged: webserver2:latest
Deleted: sha256:3f0130968d1e78db17dc061d1363da5f49c8157a1a73ffb10d923d9d7af16afc
```

///

Tip #1 - DockerHub

DockerHub - Sign Up / In

https://hub.docker.com/



Build and Ship any Application Anywhere

Docker Hub is the world's easiest way to create, manage, and deliver your teams' container applications.

Get Started Today for Free

Already have an account? Sign In

Docker ID

Email

Password

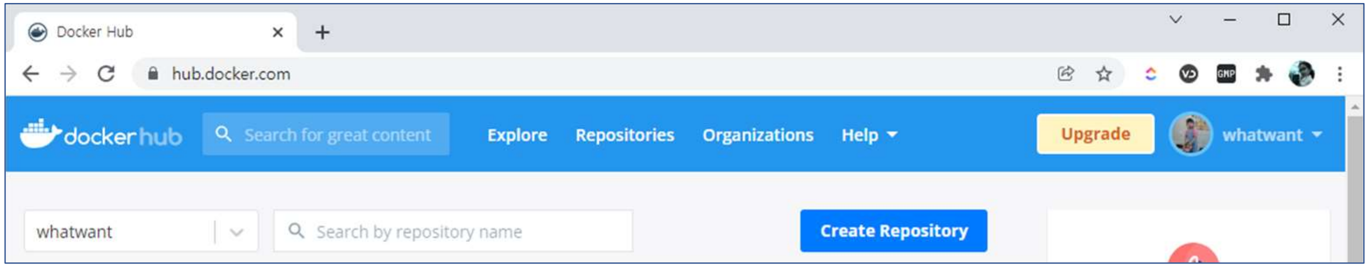
□ Send me occasional product updates and announcements.

□ I agree to the Subscription Service Agreement, Privacy Policy and Data Processing Terms.

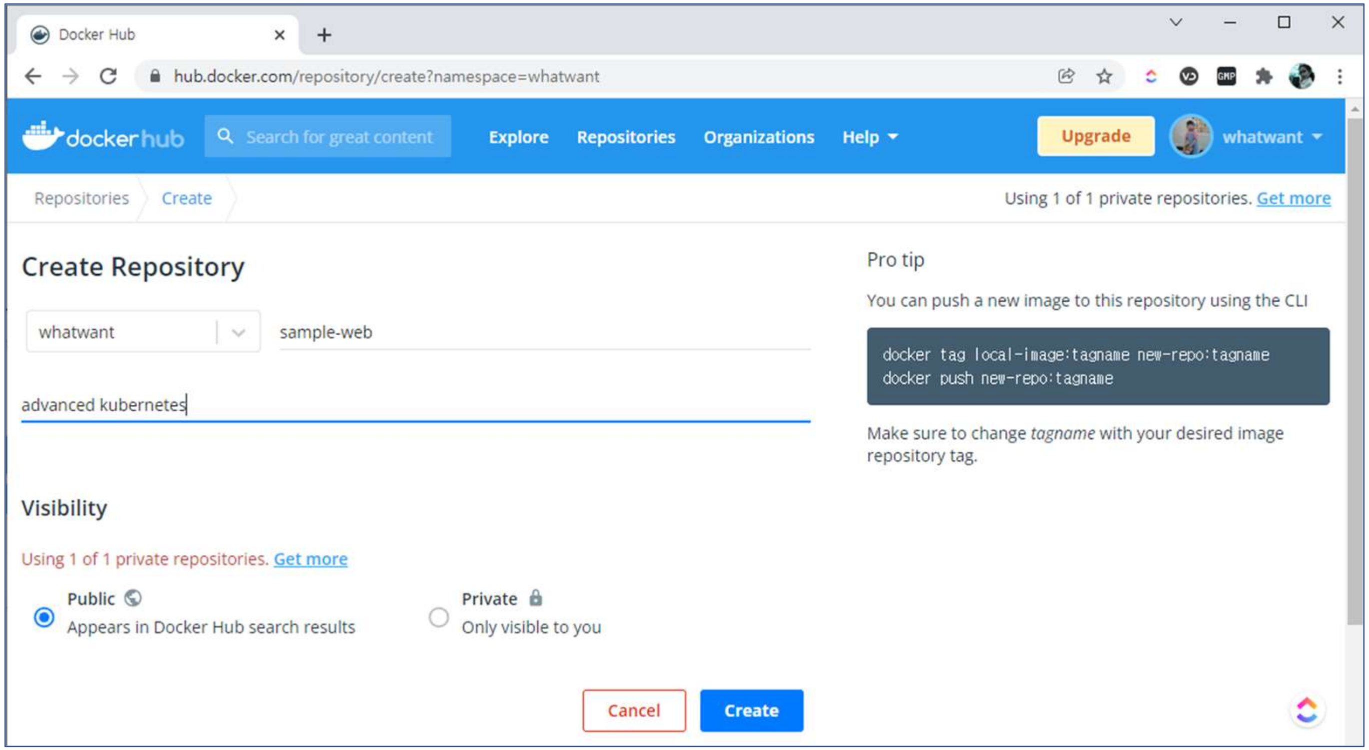
□ 로봇이 아닙니다.

Sign Up

DockerHub - Create Repository



whatwant



Using 1 of 1 private repositories. Get more

Create Repository

whatwant

sample-web

advanced kubernetes

Visibility

Using 1 of 1 private repositories. Get more

● Public
Appears in Docker Hub search results

○ Private
Only visible to you

Pro tip

You can push a new image to this repository using the CLI

```
docker tag local-image:tagname new-repo:tagname
docker push new-repo:tagname
```

Make sure to change *tagname* with your desired image repository tag.

DockerHub - docker login / tag / push

DockerHub 권한을 위해 로그인이 필요하다

```
❯ docker login

...
Login Succeeded
```

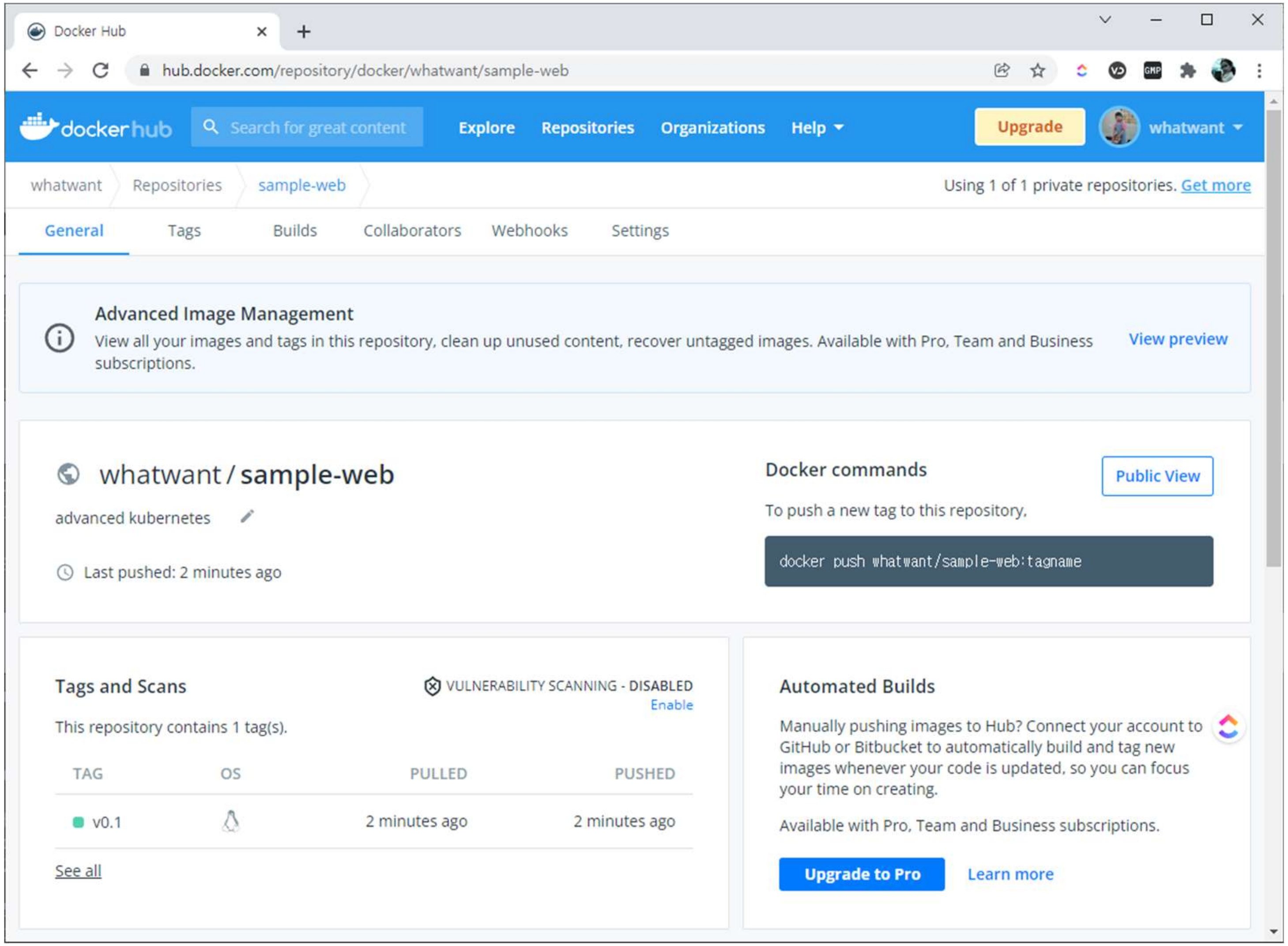
앞에서 진행해봤던 이미지를 재활용 해보자

```
❯ git clone https://github.com/whatwant-school/advanced-kubernetes.git
❯ cd advanced-kubernetes/01-week/

❯ docker build -t webserver .
...
```

업로드 하기 전에 tagging을 하고 push

```
❯ docker tag webserver:latest whatwant/sample-web:v0.1


❯ docker push whatwant/sample-web:v0.1

The push refers to repository [docker.io/whatwant/sample-web]
ba032a7dca37: Pushed
51a4ac025eb4: Mounted from library/nginx
...
2edcec3590a4: Mounted from library/nginx
v0.1: digest: sha256:f47f5ecb4f828d28f930a9c262f33066c5ca59e6b3f72c2ac882c71e3e981e31 size: 1777docker build -t webserver .
```

DockerHub - Repository



Using 1 of 1 private repositories. Get more

Advanced Image Management

View all your images and tags in this repository, clean up unused content, recover untagged images. Available with Pro, Team and Business subscriptions.

View preview

whatwant / sample-web

advanced kubernetes

Last pushed: 2 minutes ago

Docker commands

Public View

To push a new tag to this repository,

```
docker push whatwant/sample-web:tagname
```

Tags and Scans

VULNERABILITY SCANNING - DISABLED
Enable

This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
v0.1		2 minutes ago	2 minutes ago

See all

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

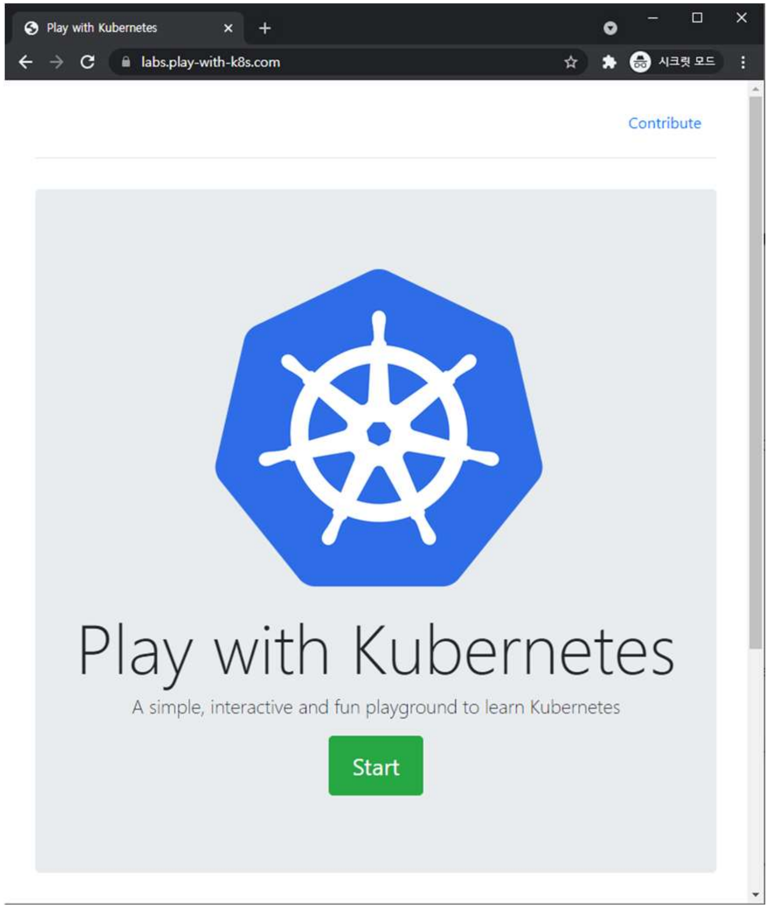
Available with Pro, Team and Business subscriptions.

Upgrade to Pro   Learn more

Tip #2 - Play with Kubernetes

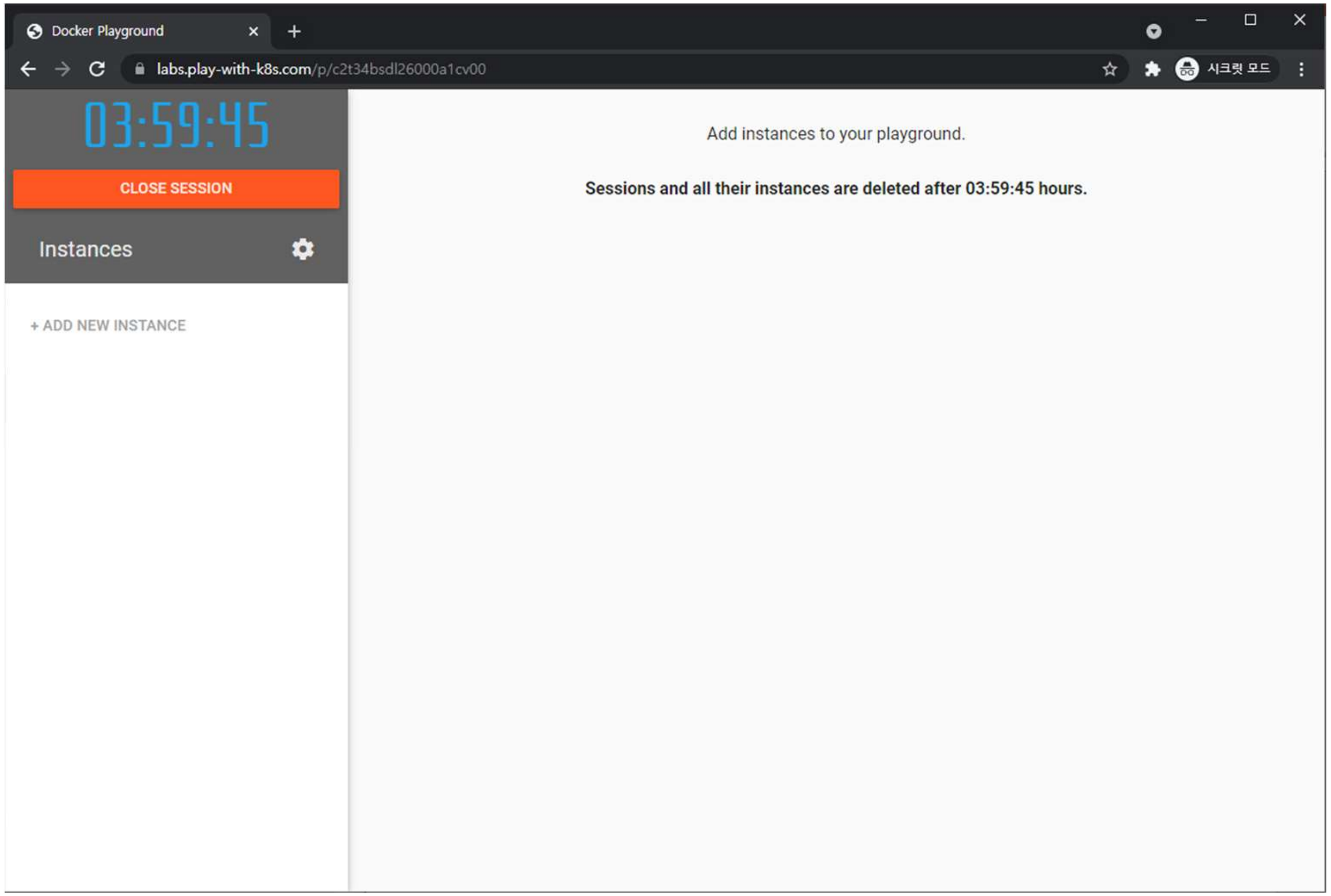
실습 환경 : https://labs.play-with-k8s.com/

- 여러 개의 instance 생성 가능, but 4시간 무료 사용
- disk 관련된 제약 等 불편함은 존재



Play with Kubernetes

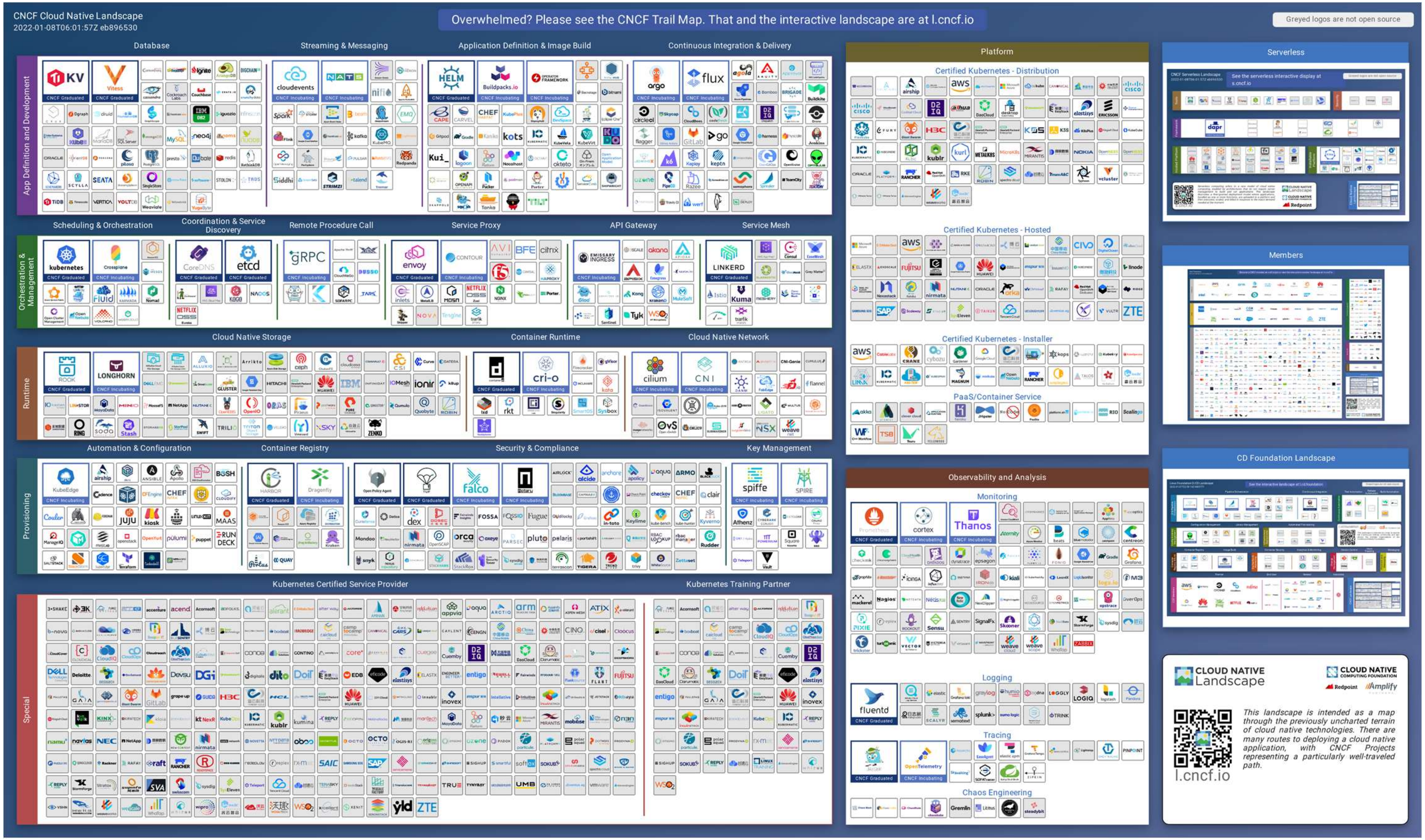
A simple, interactive and fun playground to learn Kubernetes



Add instances to your playground.

Sessions and all their instances are deleted after 03:59:45 hours.

Tip #3 - CNCF


CNCF Cloud Native Landscape
2022-01-08T06:01:57Z eb896530
Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io
Greyed logos are not open source
App Definition and Development
Database
Streaming & Messaging
Application Definition & Image Build
Continuous Integration & Delivery
Orchestration & Management
Scheduling & Orchestration
Coordination & Service Discovery
Remote Procedure Call
Service Proxy
API Gateway
Service Mesh
Runtime
Cloud Native Storage
Container Runtime
Cloud Native Network
Provisioning
Automation & Configuration
Container Registry
Security & Compliance
Key Management
Special
Kubernetes Certified Service Provider
Kubernetes Training Partner
Platform
Certified Kubernetes - Distribution
Certified Kubernetes - Hosted
Certified Kubernetes - Installer
PaaS/Container Service
Observability and Analysis
Monitoring
Logging
Tracing
Chaos Engineering
Serverless
Members
CD Foundation Landscape
CLOUD NATIVE Landscape
CLOUD NATIVE COMPUTING FOUNDATION
Redpoint
Amplify
l.cncf.io
This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.

※ 참고 : https://landscape.cncf.io/images/landscape.png

///