

2023 땅울림 씬머코딩

Docker와 컨테이너

진행자: 김선우

목차

01 Docker란?

02 Docker 기본

03 Docker 심화

04 Docker 이미지 만들기

05 미션

06 쿠버네티스 개론

1. Docker란?

Docker란?



redis



mongoDB®

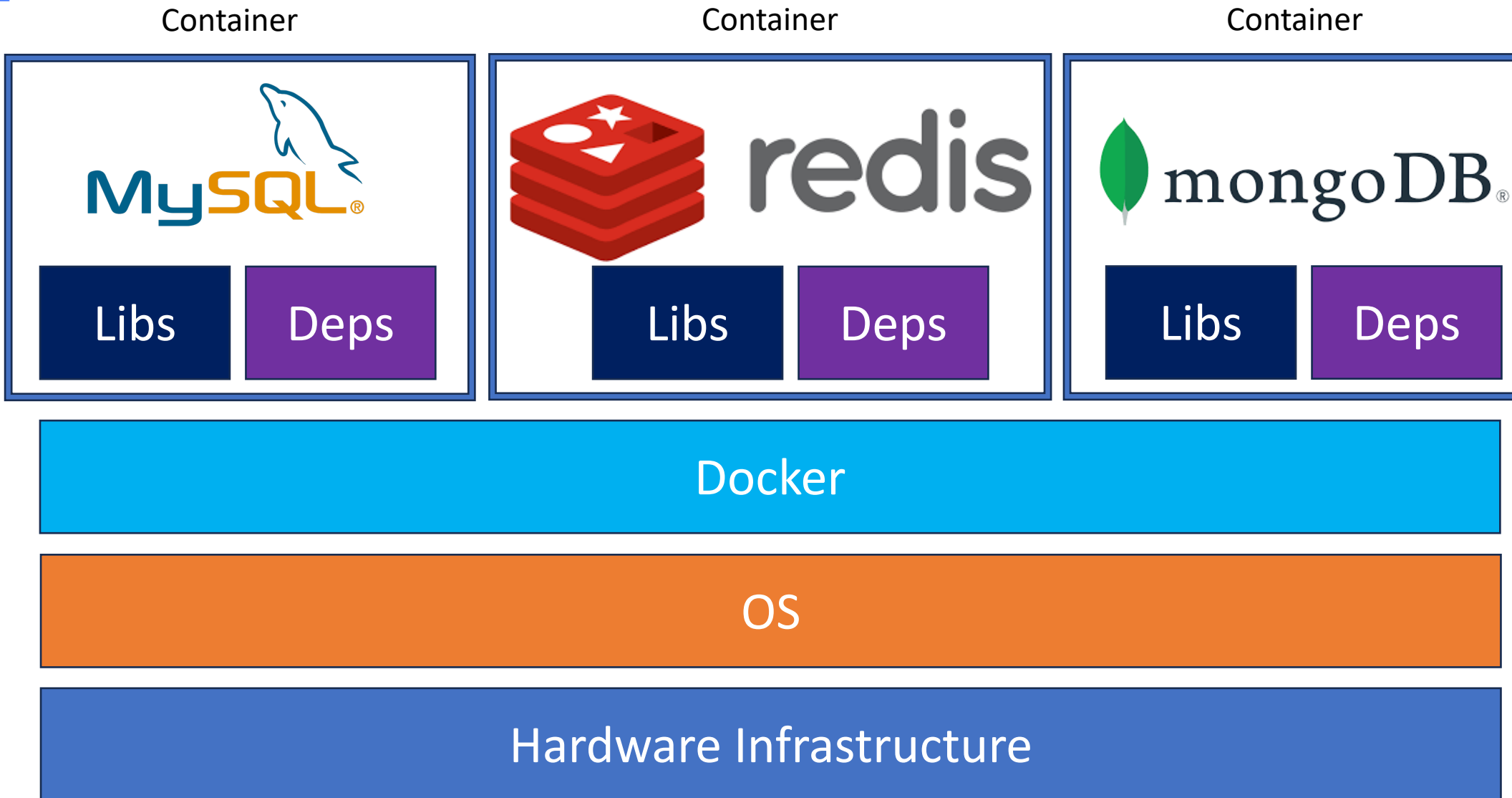
Libraries

Dependencies

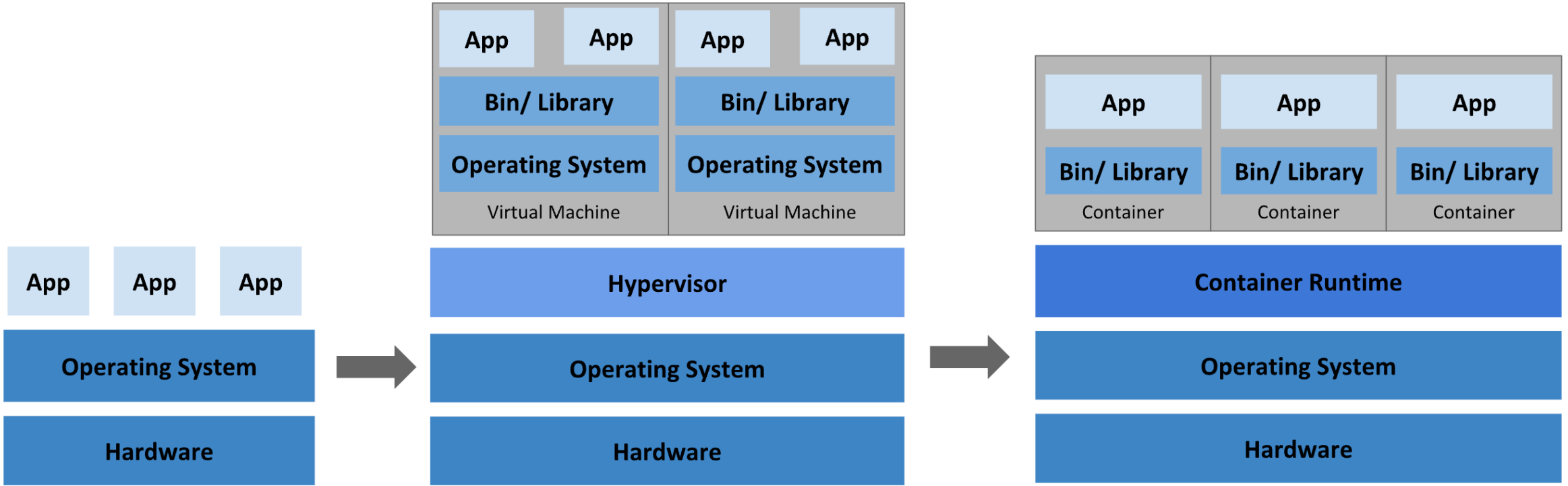
OS

Hardware Infrastructure

Docker란?



Docker란?



Traditional Deployment

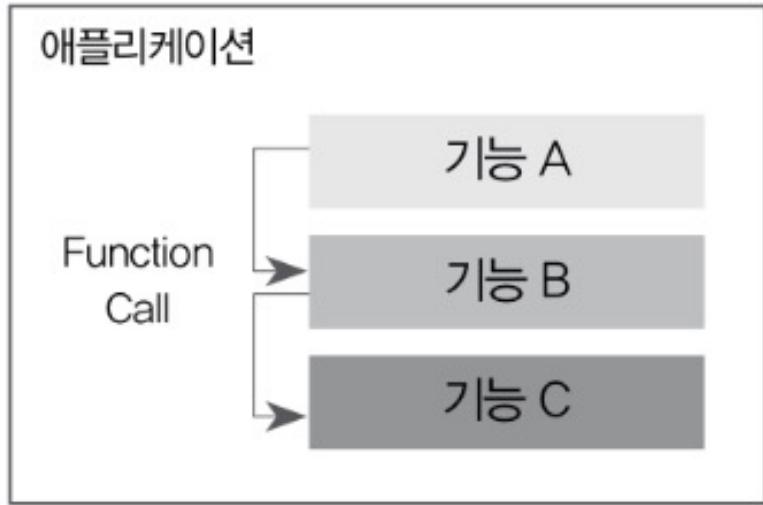
Virtualized Deployment

Container Deployment

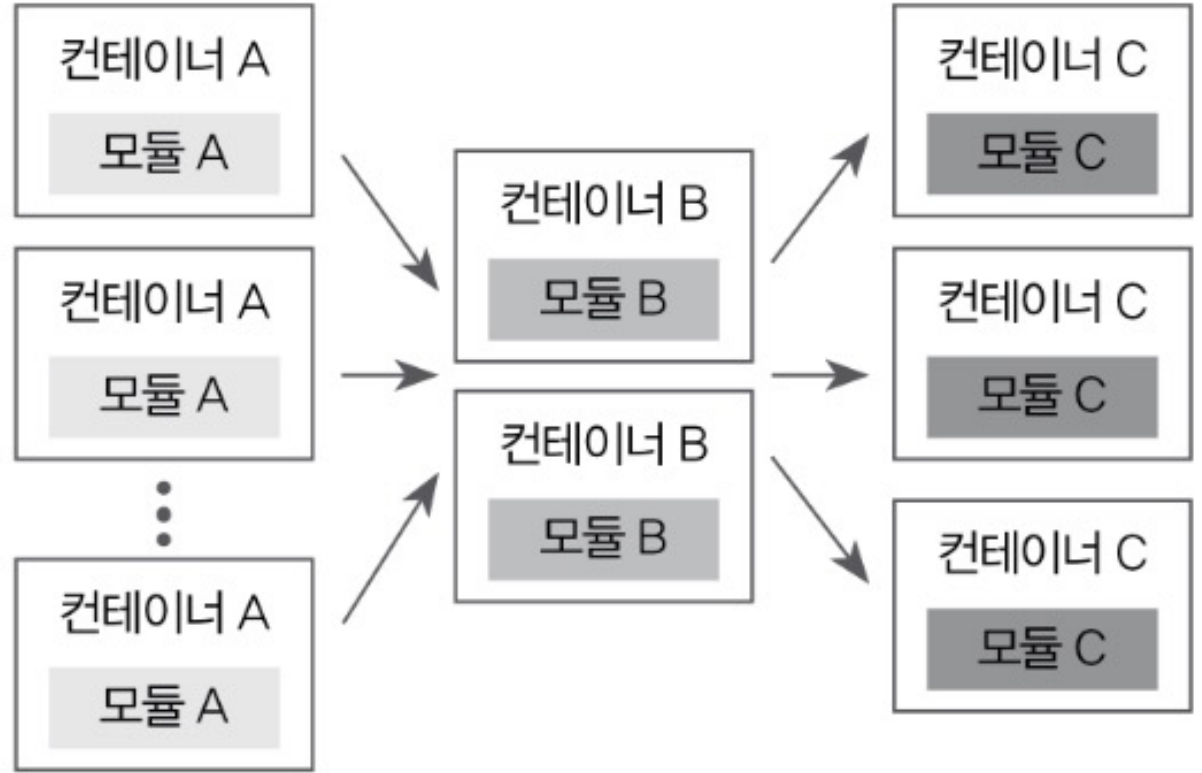
도커를 시작해야하는 이유

1. 컨테이너에 독립적인 개발환경을 보장받을 수 있음
2. 개발/운영 환경의 통합이 가능
3. 애플리케이션 배포 속도가 빠름

Docker란?



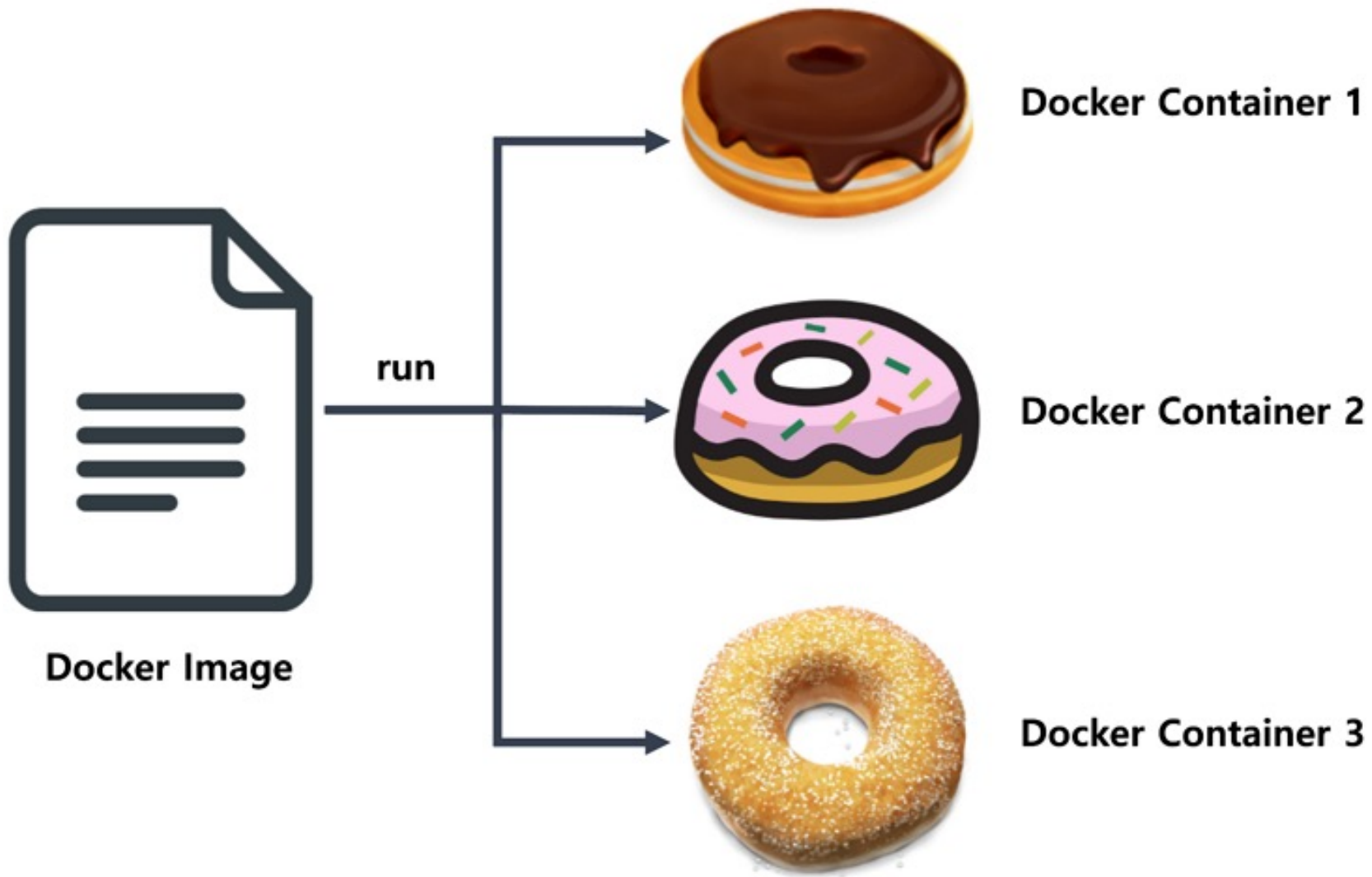
모놀리스(Monolith) 구조



마이크로서비스 구조

그림 1.2 모놀리스 구조와 마이크로서비스 구조의 예시

Docker란?



2. Docker 기본

Docker 기본

```
$ curl -fsSL https://get.docker.com -o get-docker.sh  
$ sh get-docker.sh
```

```
ubuntu@ip-172-31-47-100:/$ docker -v  
Docker version 24.0.5, build ced0996  
ubuntu@ip-172-31-47-100:/$ █
```

Docker 기본

```
$ sudo groupadd docker
$ sudo gpasswd -a $USER docker
$ newgrp docker
$ sudo docker login
```

```
Username: root
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

Docker 기본

\$ docker run nginx

```
ubuntu@ip-172-31-47-100:~$ docker run nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
648e0aadf75a: Pull complete
262696647b70: Pull complete
e66d0270d23f: Pull complete
55ac49bd649c: Pull complete
cbf42f5a00d2: Pull complete
8015f365966b: Pull complete
4cadff8bc2aa: Pull complete
Digest: sha256:67f9a4f10d147a6e04629340e6493c9703300ca23a2f7f3aa56fe615d75d31ca
Status: Downloaded newer image for nginx:latest
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/08/07 12:19:55 [notice] 1#1: using the "epoll" event method
2023/08/07 12:19:55 [notice] 1#1: nginx/1.25.1
2023/08/07 12:19:55 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2023/08/07 12:19:55 [notice] 1#1: OS: Linux 5.19.0-1025-aws
2023/08/07 12:19:55 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/08/07 12:19:55 [notice] 1#1: start worker processes
2023/08/07 12:19:55 [notice] 1#1: start worker process 29
```

Docker 기본

```
$ docker run -d nginx
```

```
$ docker ps
```

```
ubuntu@ip-172-31-47-100:~$ docker run -d nginx
2139984d1981e4892b476e334ba826aee913218d9bd61a22af1418fad7f962c6
ubuntu@ip-172-31-47-100:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS             NAMES
2139984d1981   nginx    "/docker-entrypoint..." 4 seconds ago  Up 2 seconds  80/tcp            elated_villani
ubuntu@ip-172-31-47-100:~$ █
```

Docker 기본

\$ docker ps -a

```
ubuntu@ip-172-31-47-100:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
2139984d1981   nginx    "/docker-entrypoint..." 4 seconds ago  Up 2 seconds  80/tcp       elated_villani
ubuntu@ip-172-31-47-100:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
2139984d1981   nginx    "/docker-entrypoint..." 42 seconds ago  Up 41 seconds  80/tcp       elated_villani
4d31fa0974c4   nginx    "/docker-entrypoint..." About a minute ago  Exited (0) About a minute ago  nostalgic_ganguly
```

\$ docker stop <종료할 컨테이너 id>

```
ubuntu@ip-172-31-47-100:~$ docker stop 2139984d1981
2139984d1981
ubuntu@ip-172-31-47-100:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
2139984d1981   nginx    "/docker-entrypoint..." 2 minutes ago  Exited (0) 7 seconds ago  elated_villani
4d31fa0974c4   nginx    "/docker-entrypoint..." 3 minutes ago  Exited (0) 2 minutes ago  nostalgic_ganguly
ubuntu@ip-172-31-47-100:~$
```

Docker 기본

\$ docker rm <삭제할 컨테이너 id>

```
ubuntu@ip-172-31-47-100:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS              PORTS          NAMES
2139984d1981   nginx    "/docker-entrypoint..." 3 minutes ago   Exited (0) About a minute ago
4d31fa0974c4   nginx    "/docker-entrypoint..." 4 minutes ago   Exited (0) 4 minutes ago
ubuntu@ip-172-31-47-100:~$ docker rm 21
21
ubuntu@ip-172-31-47-100:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS              PORTS          NAMES
4d31fa0974c4   nginx    "/docker-entrypoint..." 4 minutes ago   Exited (0) 4 minutes ago
nostalgic_ganguly
```


Docker 기본

\$ docker pull <이미지 이름>

```
ubuntu@ip-172-31-47-100:~$ docker pull centos
Using default tag: latest
latest: Pulling from library/centos
a1d0c7532777: Pull complete
Digest: sha256:a27fd8080b517143cbbbab9dfb7c8571c40d67d534bbdee55bd6c473f432b177
Status: Downloaded newer image for centos:latest
docker.io/library/centos:latest
ubuntu@ip-172-31-47-100:~$ docker images
REPOSITORY      TAG           IMAGE ID       CREATED        SIZE
mysql           latest       54150e9955c4   4 days ago    577MB
nginx           latest       89da1fb6dcb9   10 days ago   187MB
ubuntu          latest       5a81c4b8502e   5 weeks ago   77.8MB
centos          latest       5d0da3dc9764   23 months ago 231MB
ubuntu@ip-172-31-47-100:~$
```

Docker 기본

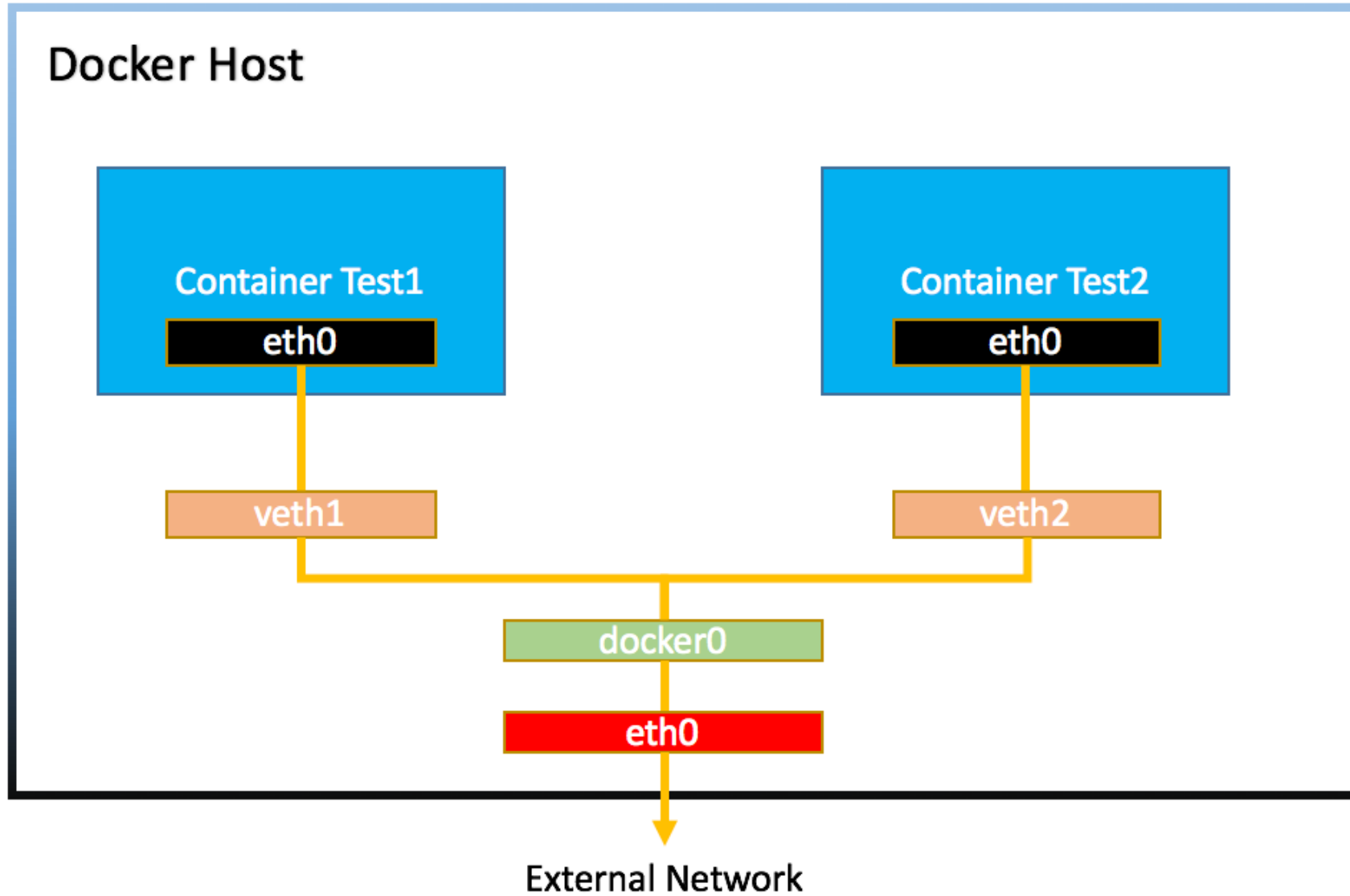
\$ docker images

\$ docker rmi <삭제할 이미지 id>

```
ubuntu@ip-172-31-47-100:~$ docker images
REPOSITORY      TAG          IMAGE ID       CREATED        SIZE
nginx           latest      89dalfb6dcb9  10 days ago   187MB
ubuntu@ip-172-31-47-100:~$ docker rm 89
Error response from daemon: No such container: 89
ubuntu@ip-172-31-47-100:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS          PORTS          NAMES
4d31fa0974c4   nginx    "/docker-entrypoint..." 7 minutes ago  Exited (0) 7 minutes ago
ubuntu@ip-172-31-47-100:~$ docker rm 4d
4d
ubuntu@ip-172-31-47-100:~$ docker rm 89
Error response from daemon: No such container: 89
ubuntu@ip-172-31-47-100:~$ docker rmi 89
Untagged: nginx:latest
Untagged: nginx@sha256:67f9a4f10d147a6e04629340e6493c9703300ca23a2f7f3aa56fe615d75d31ca
Deleted: sha256:89dalfb6dcb964dd35c3f41b7b93ffc35eaf20bc61f2e1335fea710a18424287
Deleted: sha256:e5afcbbf8f223b546aldb3d4f3c83064f346a2a8e17d4bfbaec1d12c90e2a6e3
Deleted: sha256:fda03119193d4611de17fa3d1eb9f02fb94333ac5d27ca507139a09ba0eabald
Deleted: sha256:04d32bbd70d3d7e3368290157afdfb502799784b7c60d87487e77c7aafd67d2d
Deleted: sha256:00d0e91fd006a5c96ec790434df1bb4ee545d84b34554ac2fbe5667568f916a1
Deleted: sha256:4f15baf3c136dbeff8c6f90737f0e54bd641095fd6441e359a1789ccbe554714
Deleted: sha256:748e3217b5fa76ff3ebd97186a6fcb595b92611ca87f480ea3d622e460c9a212
Deleted: sha256:c6e34807c2d51444c41c15f4fda65847faa2f43c9b4b976a2f6f476eca7429ce
ubuntu@ip-172-31-47-100:~$ docker images
REPOSITORY      TAG          IMAGE ID       CREATED        SIZE
ubuntu@ip-172-31-47-100:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS          PORTS          NAMES
ubuntu@ip-172-31-47-100:~$
```

3. Docker 심화

Docker 심화



Docker 심화

\$ docker run -d -p 80:80 nginx

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

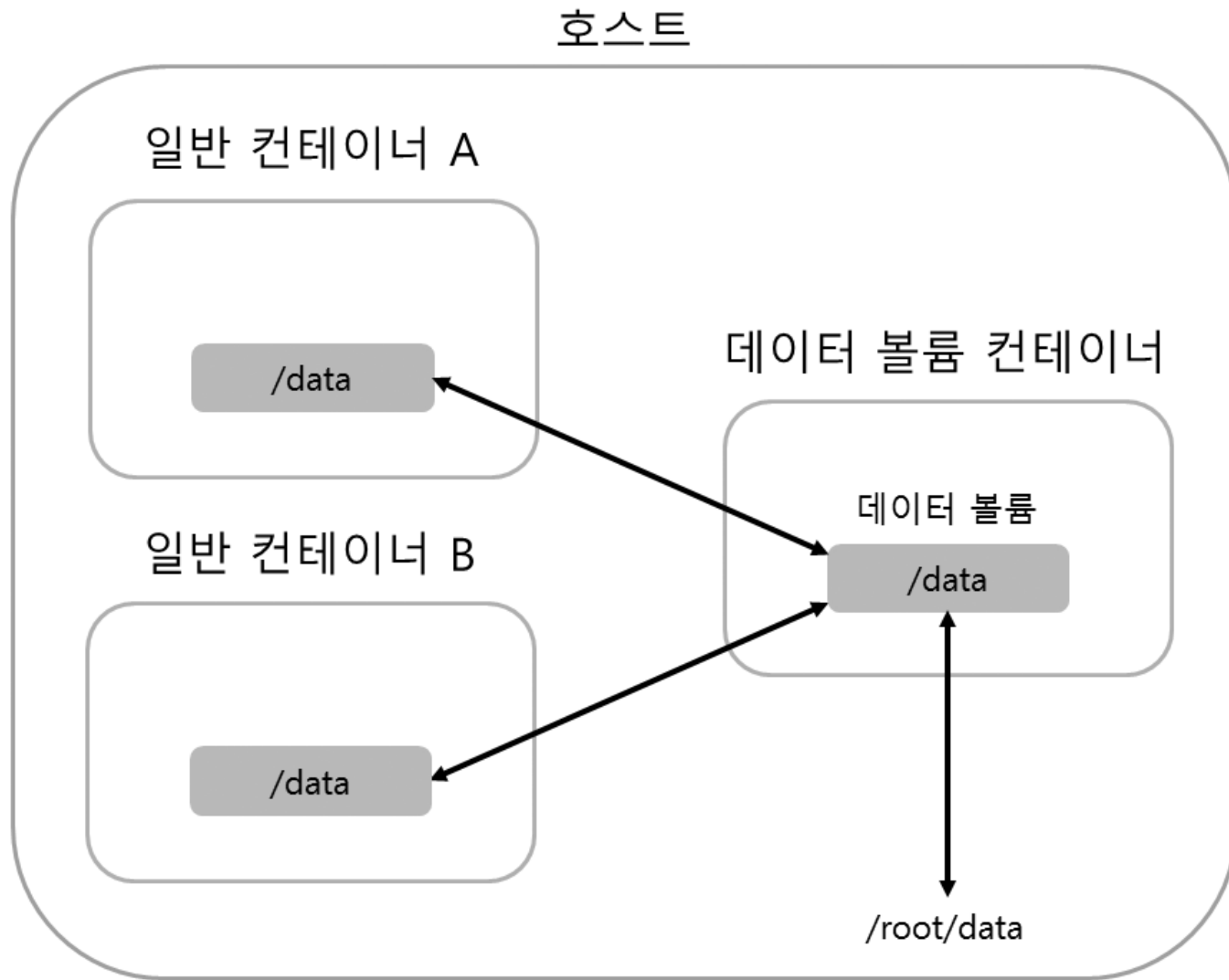
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

```
ubuntu@ip-172-31-47-100:~$ docker run -d -p 80:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
648e0aadf75a: Pull complete
262696647b70: Pull complete
e66d0270d23f: Pull complete
55ac49bd649c: Pull complete
cbf42f5a00d2: Pull complete
8015f365966b: Pull complete
4cadff8bc2aa: Pull complete
Digest: sha256:67f9a4f10d147a6e04629340e6493c9703300ca23a2f7f3aa56fe615d75d31ca
Status: Downloaded newer image for nginx:latest
f6eee686334bcb319c88c09bf0698907037a49fbc2ff564c500c90b2ecf5f5ae
ubuntu@ip-172-31-47-100:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
f6eee686334b	nginx	"/docker-entrypoint..."	5 seconds ago	Up 4 seconds	0.0.0.0:80->80/tcp, :::80->80/tcp	gallant_wright

Docker 심화



Docker 심화

```
$ docker run -i -t --name hello-volume -v /root/data:/data ubuntu /bin/bash
```

```
$ cd /data
```

```
$ touch hello2
```

```
$ docker run -i -t --volumes-from hello-volume --name hello ubuntu /bin/bash
```

```
$ ls /data
```

```
ubuntu@ip-172-31-47-100:~$ docker run -i -t --name hello-volume -v /root/data:/data ubuntu /bin/bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
3153aa388d02: Pull complete
Digest: sha256:0bced47fffa3361afa981854fcabcd4577cd43cebbb808cea2b1f33a3dd7f508
Status: Downloaded newer image for ubuntu:latest
root@14581be5f111:/# ls
bin boot data dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
root@14581be5f111:/# cd /data
root@14581be5f111:/data# pwd
/data
root@14581be5f111:/data# touch hello2
root@14581be5f111:/data# ls
hello2
root@14581be5f111:/data# exit
exit
ubuntu@ip-172-31-47-100:~$ ls
ubuntu@ip-172-31-47-100:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
ubuntu@ip-172-31-47-100:~$ docker run -i -t --volumes-from hello-volume --name hello ubuntu /bin/bash
root@0b92f296c6a8:/# ls
bin boot data dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
root@0b92f296c6a8:/# cd data
root@0b92f296c6a8:/data# ls
hello2
root@0b92f296c6a8:/data#
```

Docker 심화

```
$ docker run -d -p 80:80 nginx
$ docker exec -it <컨테이너 id> /bin/bash
$ cat /etc/*release*
$ exit
```

```
ubuntu@ip-172-31-47-100:~$ docker run -d -p 80:80 nginx
f48f7b72202c86fa89adcaf811bcc5adca8a809953de2a972bc0834efe8014a8
ubuntu@ip-172-31-47-100:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                                                                 NAMES
f48f7b72202c   nginx    "/docker-entrypoint..." 46 seconds ago Up 45 seconds 0.0.0.0:80->80/tcp, :::80->80/tcp great_robinson
ubuntu@ip-172-31-47-100:~$ docker exec -it f4 /bin/bash
root@f48f7b72202c:/# cat /etc/*release*
PRETTY_NAME="Debian GNU/Linux 12 (bookworm)"
NAME="Debian GNU/Linux"
VERSION_ID="12"
VERSION="12 (bookworm)"
VERSION_CODENAME=bookworm
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
root@f48f7b72202c:/# exit
exit
ubuntu@ip-172-31-47-100:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                                                                 NAMES
f48f7b72202c   nginx    "/docker-entrypoint..." About a minute ago Up About a minute 0.0.0.0:80->80/tcp, :::80->80/tcp great_robinson
```


Docker 심화

```
$ docker run -i -t -p 80:80 -d nginx /bin/bash
```

```
$ docker attach <컨테이너 id>
```

```
$ ctrl P Q
```

```
ubuntu@ip-172-31-47-100:~$ docker run -i -t -p 80:80 -d nginx /bin/bash
651b41d49a33dc47b2917146ac302f514c7e7c5d1fd13a6f91225eeb519cfcf1
ubuntu@ip-172-31-47-100:~$ docker attach 65
root@651b41d49a33:/# cat /etc/*release*
PRETTY_NAME="Debian GNU/Linux 12 (bookworm)"
NAME="Debian GNU/Linux"
VERSION_ID="12"
VERSION="12 (bookworm)"
VERSION_CODENAME=bookworm
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
root@651b41d49a33:/# cat /etc/*release*read escape sequence
ubuntu@ip-172-31-47-100:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                                                                 NAMES
651b41d49a33   nginx    "/docker-entrypoint...." 36 seconds ago Up 34 seconds  0.0.0.0:80->80/tcp, :::80->80/tcp  gifted_wu
```

Docker 심화



그림 2.1 이미지 이름의 구성

Docker 심화

Quick reference

- Maintained by: Canonical
- Where to get help: the Docker Community Slack, Server Fault, Unix & Linux, or Stack Overflow

Supported tags and respective Dockerfile links

- 20.04, focal-20230801, focal
- 22.04, jammy-20230624, jammy, latest
- 23.04, lunar-20230731, lunar, rolling
- 23.10, mantic-20230801, mantic, devel

Quick reference (cont.)

- Where to file issues: the cloud-images bug tracker (include the docker tag)
- Supported architectures: (more info) amd64, arm32v7, arm64v8, ppc64le, s390x
- Published image artifact details: repo-info repo's repos/ubuntu/ directory (history) (image metadata, transfer size, etc)
- Image updates: official-images repo's library/ubuntu label official-images repo's library/ubuntu file (history)
- Source of this description: docs repo's ubuntu/ directory (history)

The screenshot shows the Docker Hub page for the 'ubuntu' image. At the top, there's the Ubuntu logo and the text 'ubuntu DOCKER OFFICIAL IMAGE · 1B+ · 10K+'. Below that, there are tabs for 'Overview' and 'Tags', with 'Tags' selected. A search bar and a 'Sort by' dropdown (set to 'Newest') are visible. The main content area displays a list of tags with their respective digests, OS/architectures, vulnerability counts, and compressed sizes. Each tag entry includes a 'docker pull' button.

TAG	DIGEST	OS/ARCH	VULNERABILITIES	COMPRESSED SIZE
latest	b060ffe8e15	linux/amd64	0 H 1 M 9 L	28.17 MB
	f01654f07625	linux/arm/v7	0 H 1 M 9 L	24.93 MB
	fb4a67ec973b	linux/arm64/v8	0 H 1 M 9 L	26.08 MB
	+2 more...			
rolling	3787e67b05cc	linux/amd64	None found	25.6 MB
	02d4ef011cf1	linux/arm/v7	None found	23.5 MB
	33a3158bf45e	linux/arm64/v8	None found	24.88 MB
	+2 more...			
mantic-20230801	56ccaa299853	linux/amd64	None found	25.82 MB
	2c581d2179fa	linux/arm/v7	None found	23.45 MB
	ea1b8a574050	linux/arm64/v8	None found	25.04 MB
	+2 more...			
mantic	56ccaa299853	linux/amd64	None found	25.82 MB
	2c581d2179fa	linux/arm/v7	None found	23.45 MB
	ea1b8a574050	linux/arm64/v8	None found	25.04 MB
	+2 more...			

4. 도커 이미지 만들기

JDK17 설치

```
$ sudo apt update
$ sudo apt install openjdk-17-jdk
```

sudo란?

- Super User Do의 약자(최근에는 Substitute User Do도 쓰임)
- 유닉스 및 리눅스 계열에서 다른 사용자의 보안 권한과 관련된 프로그램을 구동할 수 있게 도와주는 것
- etc/sudoers파일에 지정된 사용자만 sudo명령어를 사용 가능
- 현재 계정에서 다른 계정의 권한만 빌림
- root가 아닌 사용자가 root에 준하는 능력으로 sudo 다음에 나오는 명령을 실행하게 하는 명령어.
- 슈퍼유저, 관리자 권한을 가지지만, 근본적으로 해당 사용자가 내리는 명령

apt란?

Advanced Package Tool의 약자로 Debian시스템에 포함된 핵심 도구들의 집합체이다. 즉, 우분투에서 쓰이는 데비안 계열의 패키지를 관리하는데 쓰이는 도구이다. Apt를 활용해 다음과 같은 작업들을 처리할 수 있다.

- 응용 프로그램 설치
- 응용 프로그램 삭제
- 응용 프로그램을 항상 최신 버전으로 유지하기
- 그 이상 작업들...

```
ubuntu@ip-172-31-46-168:~$ java --version
openjdk 17.0.7 2023-04-18
OpenJDK Runtime Environment (build 17.0.7+7-Ubuntu-0ubuntu122.04.2)
OpenJDK 64-Bit Server VM (build 17.0.7+7-Ubuntu-0ubuntu122.04.2, mixed mode, sharing)
```

Git 설치

```
$ sudo apt install git
```

```
ubuntu@ip-172-31-46-168:~$ git --version  
git version 2.34.1
```

Docker 이미지 만들기

```
$ git clone https://github.com/LandvibeDev/spring-landlog.git
```

```
ubuntu@ip-172-31-46-168:~$ git clone https://github.com/LandvibeDev/spring-landlog.git
Cloning into 'spring-landlog'...
remote: Enumerating objects: 1825, done.
remote: Counting objects: 100% (482/482), done.
remote: Compressing objects: 100% (152/152), done.
remote: Total 1825 (delta 363), reused 337 (delta 323), pack-reused 1343
Receiving objects: 100% (1825/1825), 223.64 KiB | 9.72 MiB/s, done.
Resolving deltas: 100% (613/613), done.
ubuntu@ip-172-31-46-168:~$ ls
spring-landlog
```

Docker 이미지 만들기

```
ubuntu@ip-172-31-46-168:~$ cd spring-landlog
ubuntu@ip-172-31-46-168:~/spring-landlog$ git branch -a
* main
remotes/origin/HEAD -> origin/main
remotes/origin/dongha
remotes/origin/hyungseok
remotes/origin/jaeseung
remotes/origin/jumin
remotes/origin/junyoung
remotes/origin/keonhee
remotes/origin/main
remotes/origin/seungcheol
remotes/origin/sewon
remotes/origin/suhwan
remotes/origin/yonghyun
ubuntu@ip-172-31-46-168:~/spring-landlog$ git checkout remotes/origin/keonhee
Note: switching to 'remotes/origin/keonhee'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 68b0e5c Step 1  1. 로그인  2. 블로그  3. Restful API  4. Lombok  5. Swagger  6. 에러 핸들링
ubuntu@ip-172-31-46-168:~/spring-landlog$ █
```

```
$ cd spring-landlog
$ git branch -a
$ git checkout remotes/origin/<본인이름>
```


Docker 이미지 만들기

```
$ ./gradlew build
$ ls build/libs
$ java -jar build/libs/<jar파일명>
```

```
ubuntu@ip-172-31-46-168:~/spring-landlog$ ./gradlew build
```

```
BUILD SUCCESSFUL in 3s
```

```
7 actionable tasks: 7 up-to-date
```

```
ubuntu@ip-172-31-46-168:~/spring-landlog$ ls build/libs
```

```
landlog-0.0.1-SNAPSHOT-plain.jar landlog-0.0.1-SNAPSHOT.jar
```

```
ubuntu@ip-172-31-46-168:~/spring-landlog$ java -jar build/libs/landlog-0.0.1-SNAPSHOT.jar
```

```

  ____ _
 / ___ \| | | |
 \___ \| |_| |
  ___) | | | |
 / ___ \| |_| |
 \___) | |_| |
    ___/ |___|_|
    |___)

:: Spring Boot ::                (v3.1.0)

```

```

2023-07-24T14:20:52.756Z INFO 5784 --- [
ng-landlog)
2023-07-24T14:20:52.762Z INFO 5784 --- [
2023-07-24T14:20:55.216Z INFO 5784 --- [
2023-07-24T14:20:55.244Z INFO 5784 --- [
2023-07-24T14:20:55.244Z INFO 5784 --- [
2023-07-24T14:20:55.546Z INFO 5784 --- [
2023-07-24T14:20:55.554Z INFO 5784 --- [
2023-07-24T14:20:57.045Z INFO 5784 --- [
2023-07-24T14:20:57.094Z INFO 5784 --- [
main] c.landvibe.landlog.LandlogApplication : Starting LandlogApplication v0.0
main] c.landvibe.landlog.LandlogApplication : No active profile set, falling b
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s):
main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache
main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded Web
main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: init
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080
main] c.landvibe.landlog.LandlogApplication : Started LandlogApplication in 5.

```

Docker 이미지 만들기

The screenshot displays the AWS Management Console interface for EC2 instances. On the left, a navigation menu includes sections for 'EC2 대시보드', '인스턴스', 'AMI', 'Elastic Block Store', '네트워크 및 보안', '로드 밸런싱', and 'Auto Scaling'. The main area shows a table of EC2 instances with columns for Name, ID, Status, Type, Checks, Health, Region, DNS, Private IP, Public IP, and Monitoring. Two instances are listed: 'Nginx' and '스프링 부트', both in a 'Running' state. Below the table, the details for instance 'i-070cc704f9868c16f' (Spring Boot) are shown, including its ID, IP addresses, and VPC ID. A browser window is overlaid on the right, showing the URL 'http://15.164.49.113:8080' and the page content 'Hello Spring' with a '회원 가입' (Sign Up) button. A red box highlights the URL in the browser, and another red box highlights the public IPv4 address '15.164.49.113' in the instance details.

Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	경보 상태	가용 영역	퍼블릭 IPv4 DNS	퍼블릭 IPv4 주소	탄력적 IP	IPv6 IP	모니터링
Nginx	i-0dda5a5c27c165b8f	실행 중	t2.micro	2/2개 검사 통과	경보 없음	ap-northeast-2c	ec2-3-35-10-187.ap-no...	3.35.10.187	-	-	dis
스프링 부트	i-070cc704f9868c16f	실행 중	t2.micro	2/2개 검사 통과	경보 없음	ap-northeast-2c	ec2-15-164-49-113.ap-...	15.164.49.113	-	-	dis

인스턴스: i-070cc704f9868c16f(스프링 부트)

퍼블릭 IPv4 주소: 15.164.49.113 | [개방 주소법](#)

인스턴스 상태: **실행 중**

프라이빗 IP DNS 이름(IPv4만 해당): ip-172-31-46-168.ap-northeast-2.compute.internal

인스턴스 유형: t2.micro

VPC ID: vpc-0873dc58cdec3033c

퍼블릭 IPv4 주소: 172.31.46.168

퍼블릭 IPv4 DNS: ec2-15-164-49-113.ap-northeast-2.comput

탄력적 IP 주소: -

AWS Compute Optimizer 찾기: [권장 사항을 위해 AWS Compute Optimizer에 업](#)

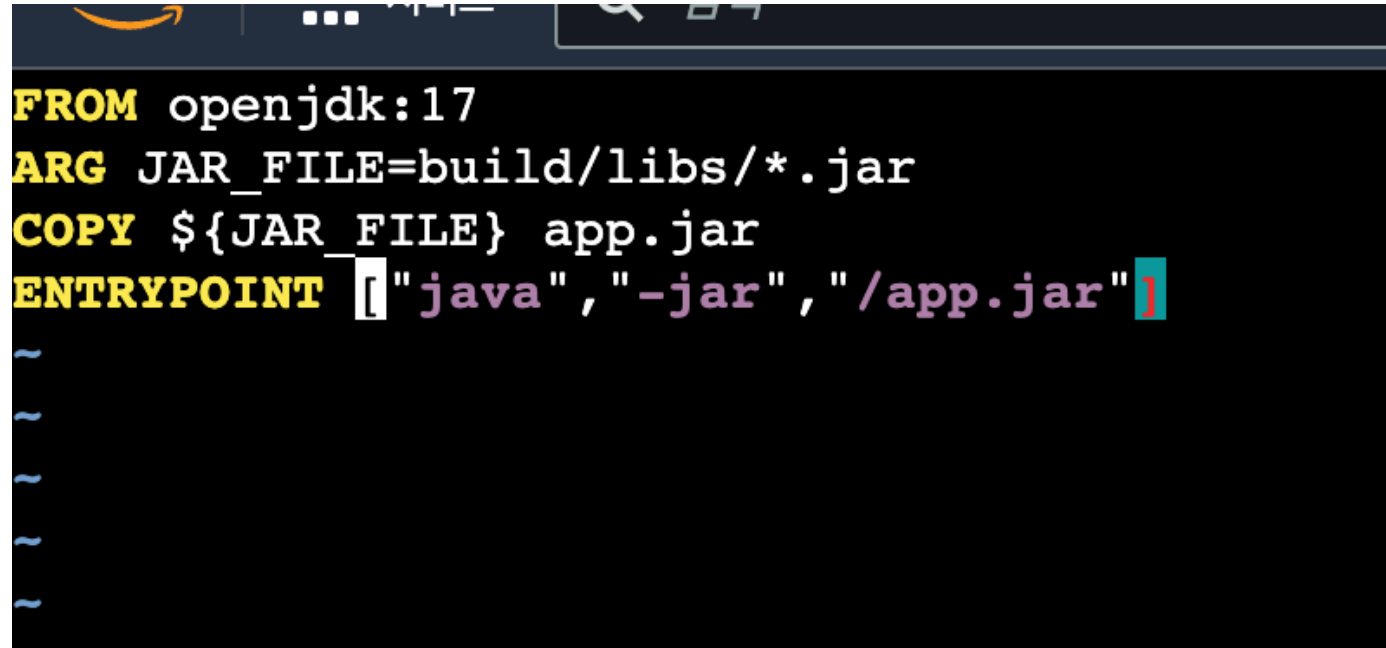
Docker 이미지 만들기

지시어	설명
FROM	베이스 이미지 지정
RUN	이미지를 지정하면서 실행할 명령 지정
ENTRYPOINT	컨테이너의 어플 지정 (컨테이너 시작할 때 실행할 명령어)
EXPOSE	컨테이너의 포트 지정
ADD	이미지 생성 시 파일 추가
COPY	이미지 생성 시 파일 복사
WORKDIR	컨테이너 작업 디렉토리 지정
MAINTAINER	이미지 작성자 명시
CMD	컨테이너의 어플 지정 (컨테이너 시작할 때 실행할 명령어)
LABEL	이미지의 라벨 지정
ENV	컨테이너의 환경 변수 지정
VOLUME	컨테이너의 볼륨 지정
USER	컨테이너의 사용자 지정

Docker 이미지 만들기

\$ vi Dockerfile

```
FROM openjdk:17
ARG JAR_FILE=build/libs/*.jar
COPY ${JAR_FILE} app.jar
ENTRYPOINT ["java","-jar","/app.jar"]
```

A screenshot of a terminal window with a dark background. The text is displayed in a monospaced font with syntax highlighting: 'FROM' is yellow, 'ARG' is yellow, 'COPY' is yellow, and 'ENTRYPOINT' is yellow. The string values are in purple. The cursor is at the end of the last line. There are four tilde characters on the lines below.

```
FROM openjdk:17
ARG JAR_FILE=build/libs/*.jar
COPY ${JAR_FILE} app.jar
ENTRYPOINT ["java","-jar","/app.jar"]
~
~
~
~
```

Docker 이미지 만들기

```
$ docker build -tag <계정명>/<이미지명>:<버전> .  
$ docker push <계정명>/<이미지명>:<버전>
```

```
root@ip-172-31-47-100:/home/ubuntu/spring-landlog# docker build --tag xeonu/landlog:1.0.0 .  
[+] Building 1.7s (8/8) FINISHED  
=> [internal] load build definition from Dockerfile  
=> => transferring dockerfile: 146B  
=> [internal] load .dockerignore  
=> => transferring context: 2B  
=> [internal] load metadata for docker.io/library/openjdk:17  
=> [auth] library/openjdk:pull token for registry-1.docker.io  
=> [internal] load build context  
=> => transferring context: 179B  
=> [1/2] FROM docker.io/library/openjdk:17@sha256:528707081fdb9562eb819128a9f85ae7fe000e2fbaeaf9f87662e7b3f38cb7d8  
=> CACHED [2/2] COPY build/libs/*.jar app.jar  
=> exporting to image  
=> => exporting layers  
=> => writing image sha256:6bad6dc04e275fcad4c4135567c620aa03fa9acd5c926d3cal1f9a91a9bec2251  
=> => naming to docker.io/xeonu/landlog:1.0.0  
root@ip-172-31-47-100:/home/ubuntu/spring-landlog# docker images  
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE  
xeonu/landlog       1.0.0       6bad6dc04e27     14 minutes ago  492MB
```

5. 미션

미션

인스턴스 A: nginx

인스턴스 B: spring 프로젝트

Docker를 이용하여 각 인스턴스에 필요한 프로그램 컨테이너를 생성하고 외부의 사용자가 인스턴스 A로 접속하면 인스턴스 B의 WAS로 접속하도록 인프라를 구성하시오.

6. 쿠버네티스 개론

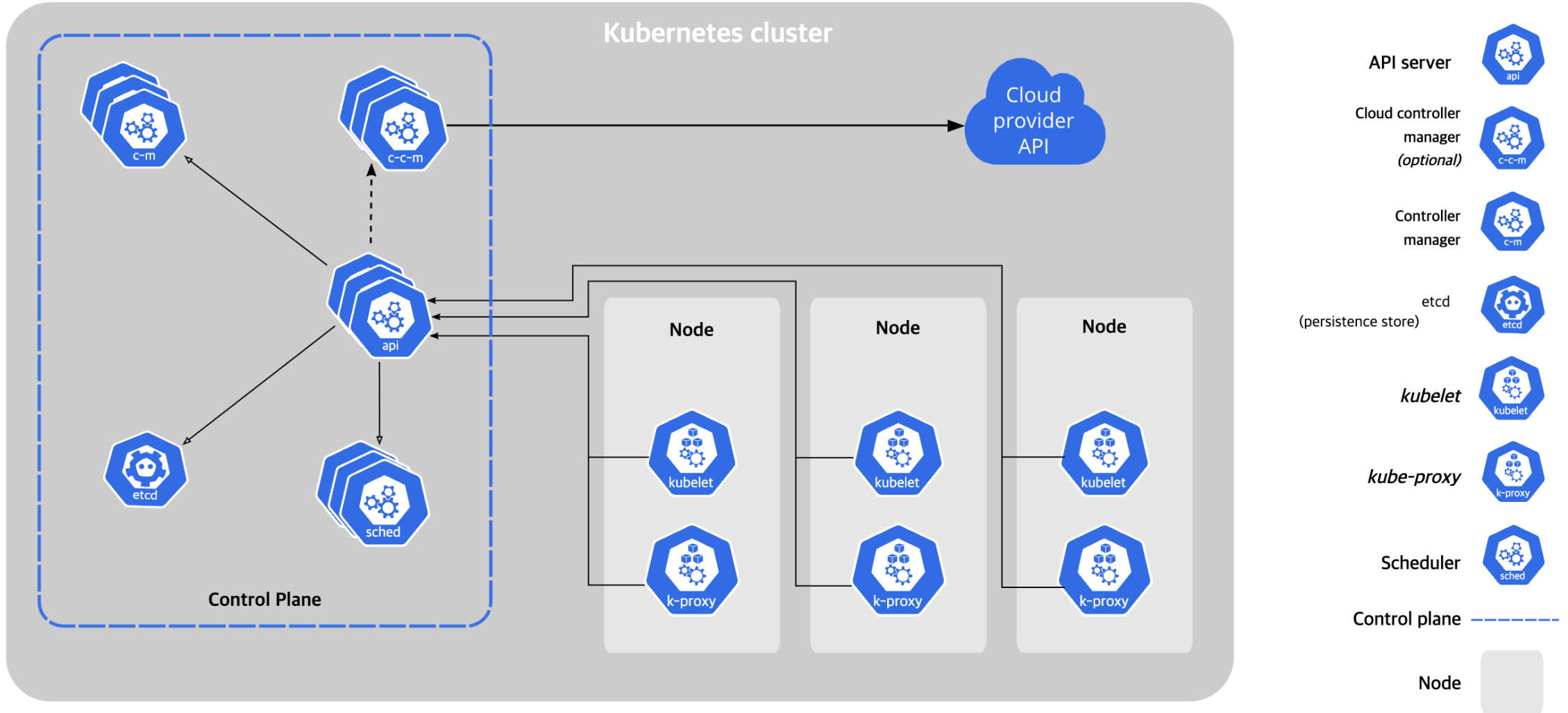
쿠버네티스 개론

100개의 컨테이너가 필요하다면?

컨테이너에 장애가 발생한다면?

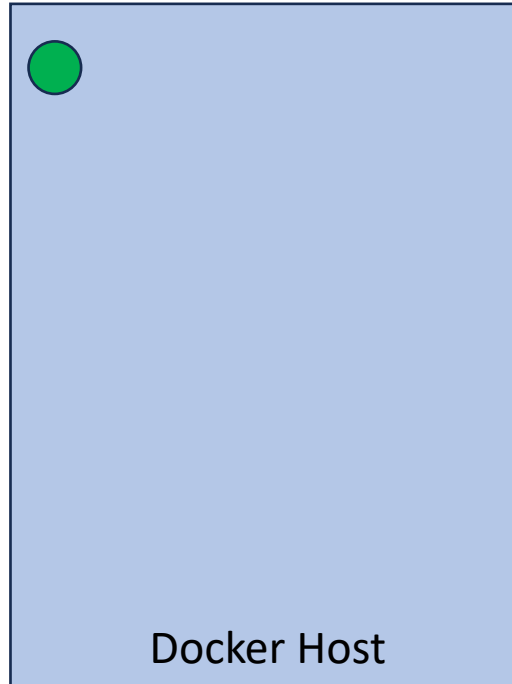
실시간으로 스케일 아웃 확장이 필요하다면?

쿠버네티스 개론



쿠버네티스 개론

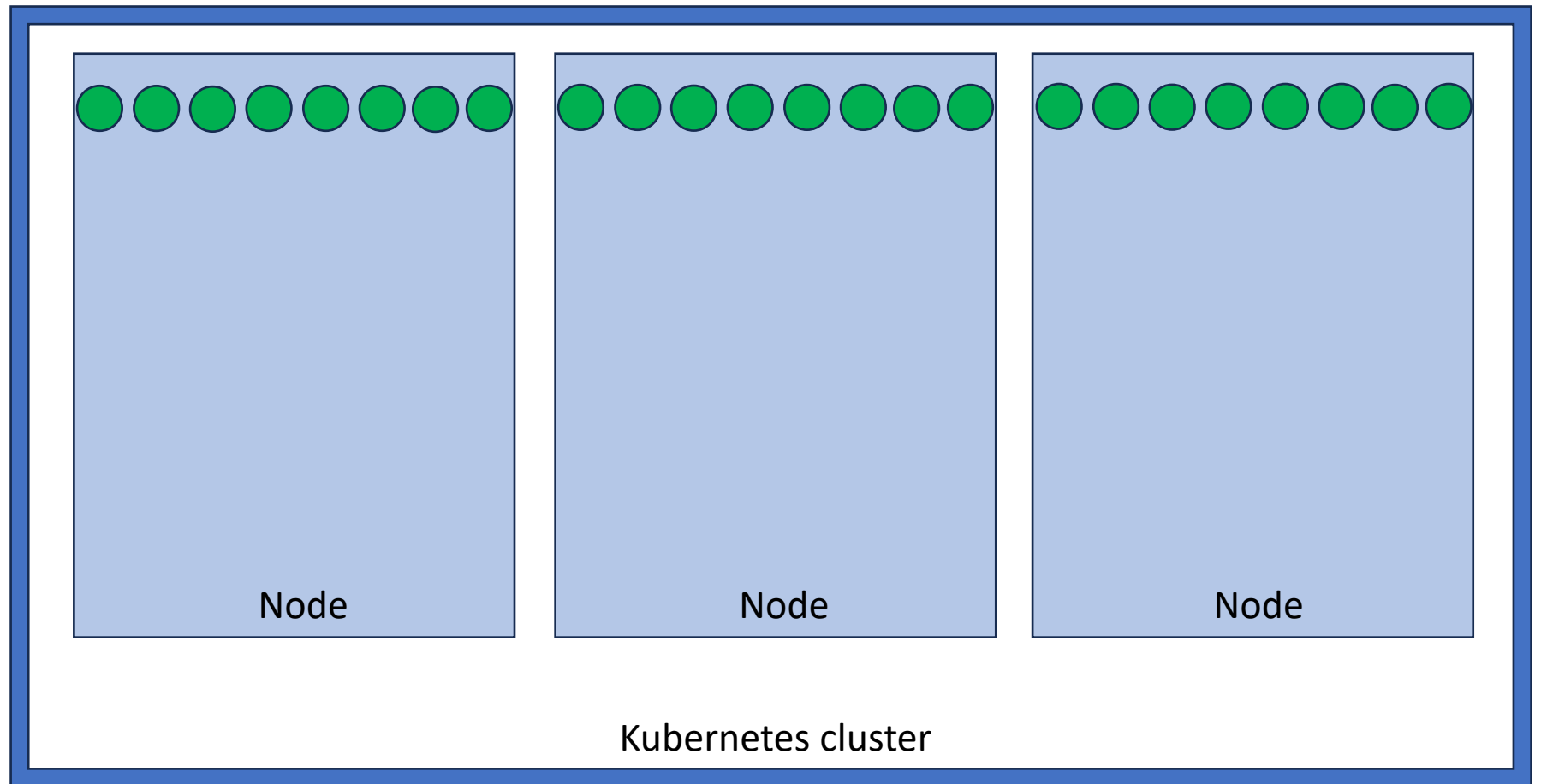
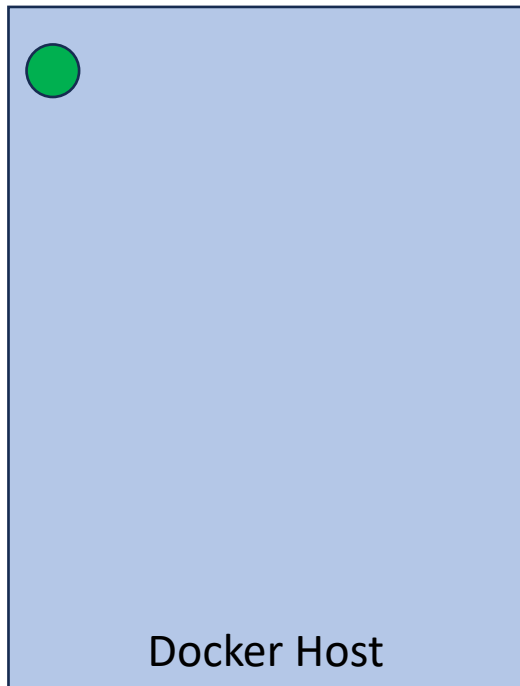
```
$ docker run landlog
```



쿠버네티스 개론

```
$ docker run landlog
```

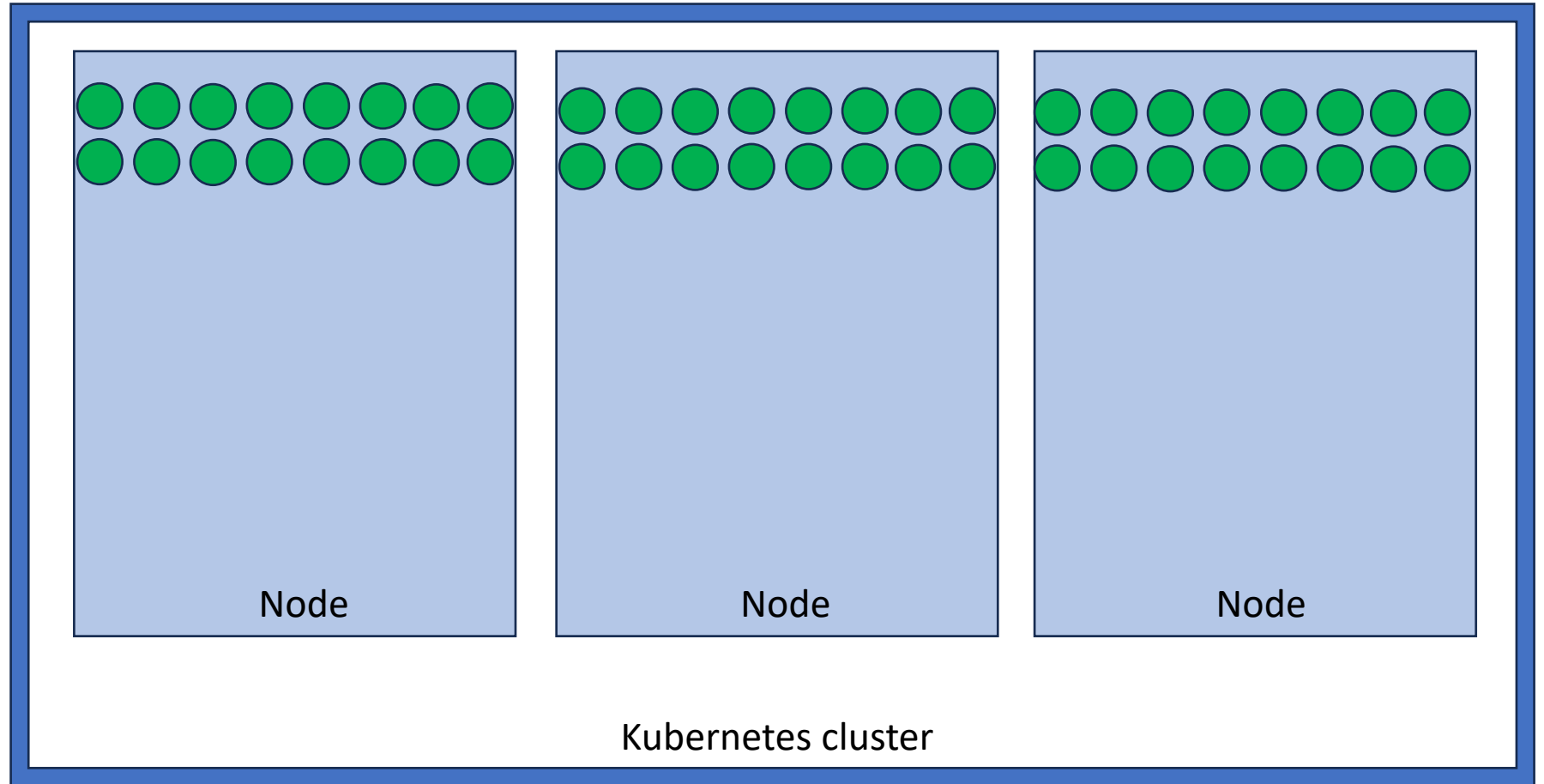
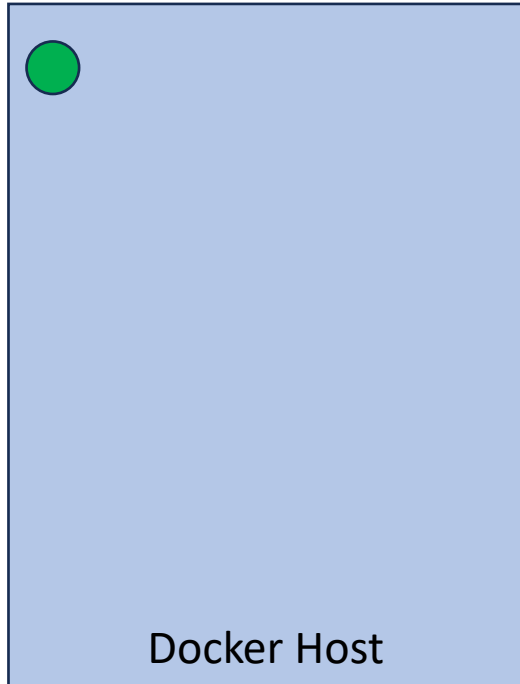
```
$ kubectl run --replicas=1000 landlog
```



쿠버네티스 개론

```
$ docker run landlog
```

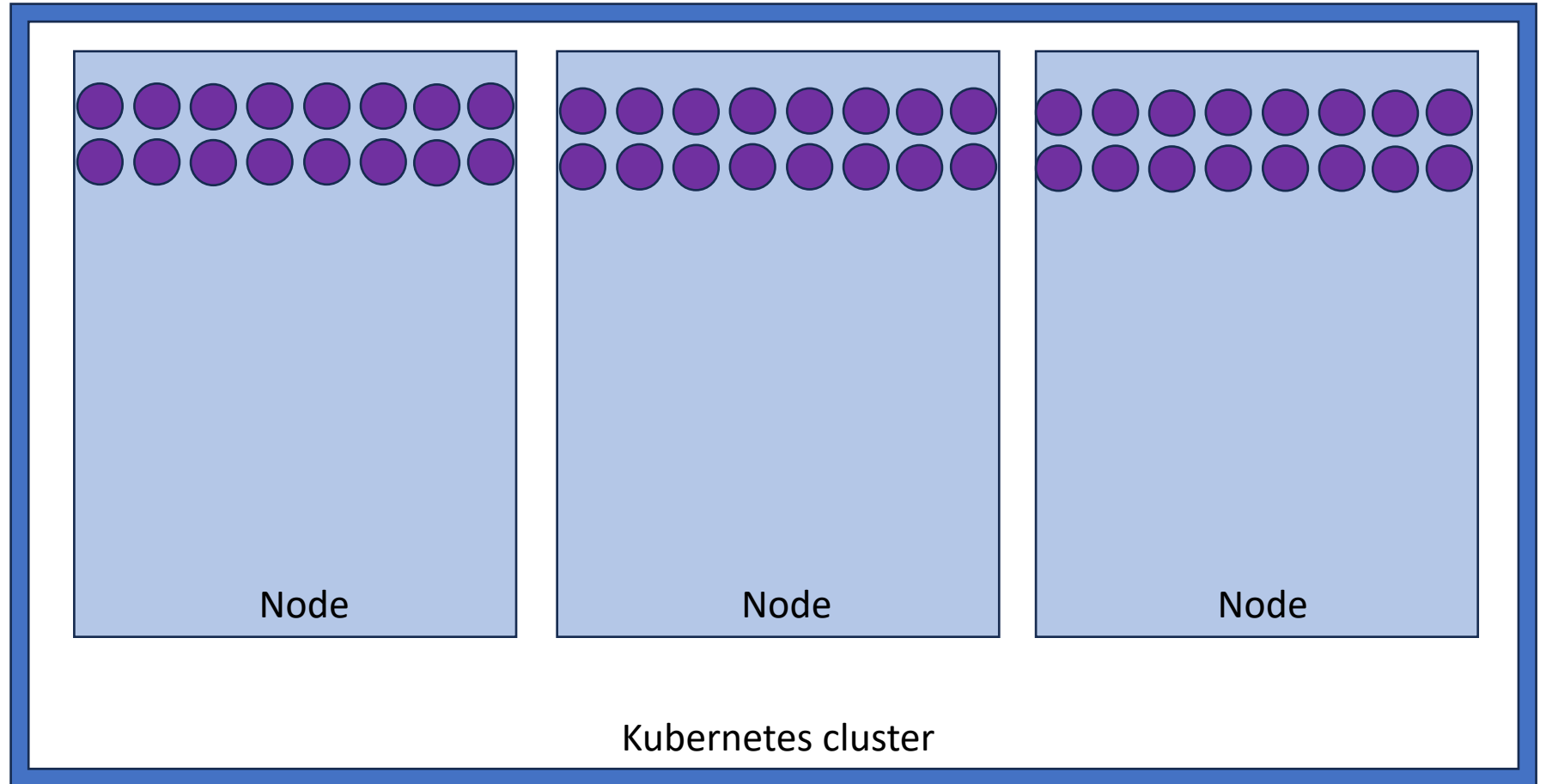
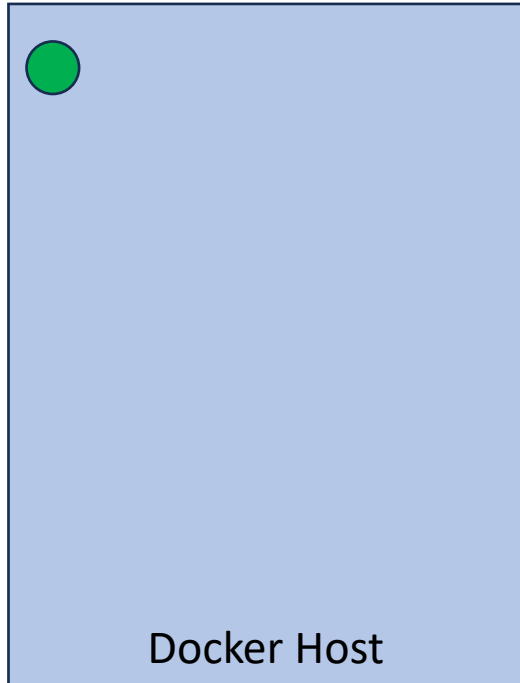
```
$ kubectl scale --replicas=2000 landlog
```



쿠버네티스 개론

```
$ docker run landlog
```

```
$ kubectl rolling-update landlog --image = landlog:2.0.0
```



쿠버네티스 개론

```
$ docker run landlog
```

```
$ kubectl rolling-update landlog --rollback
```

