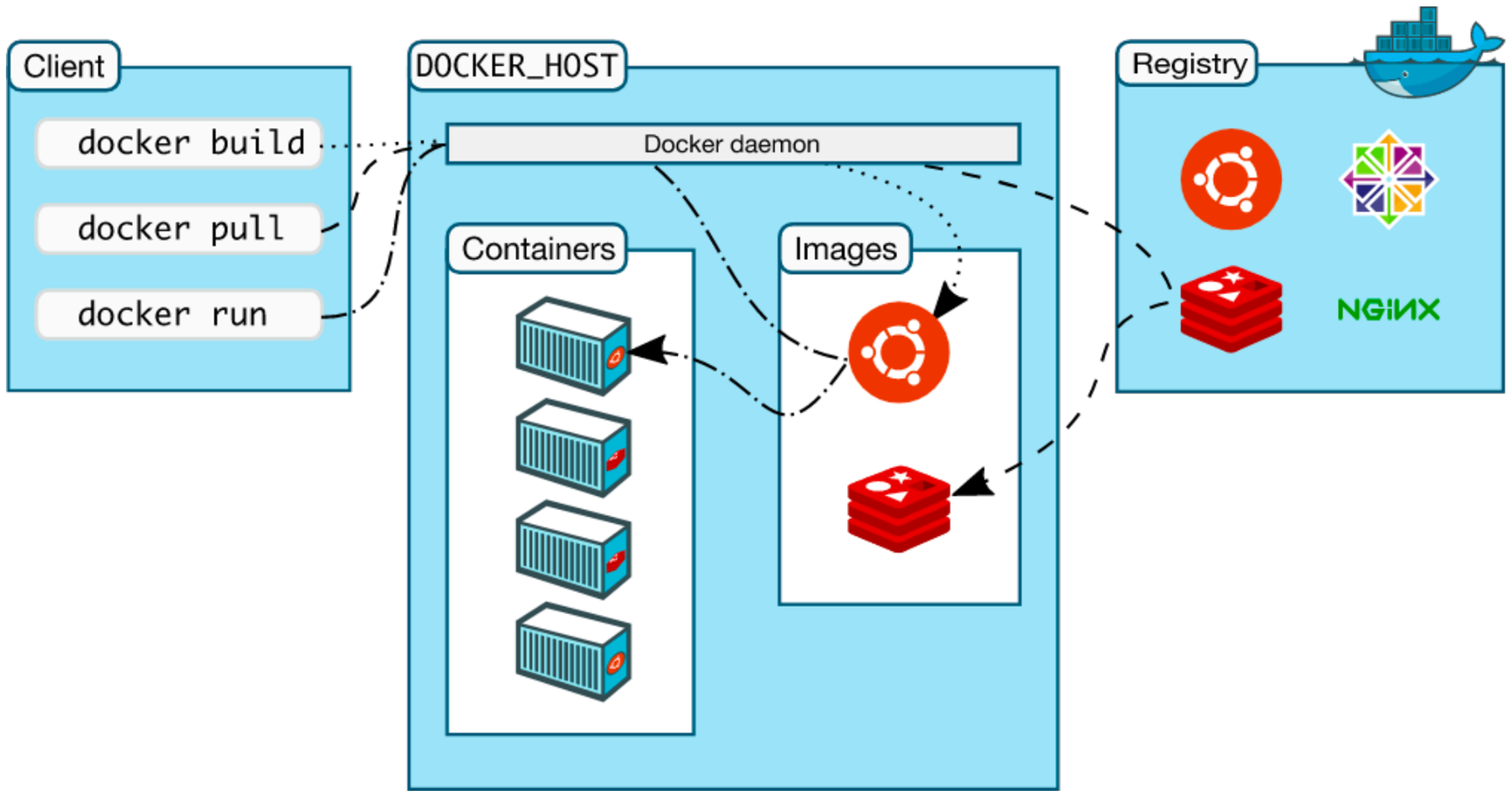
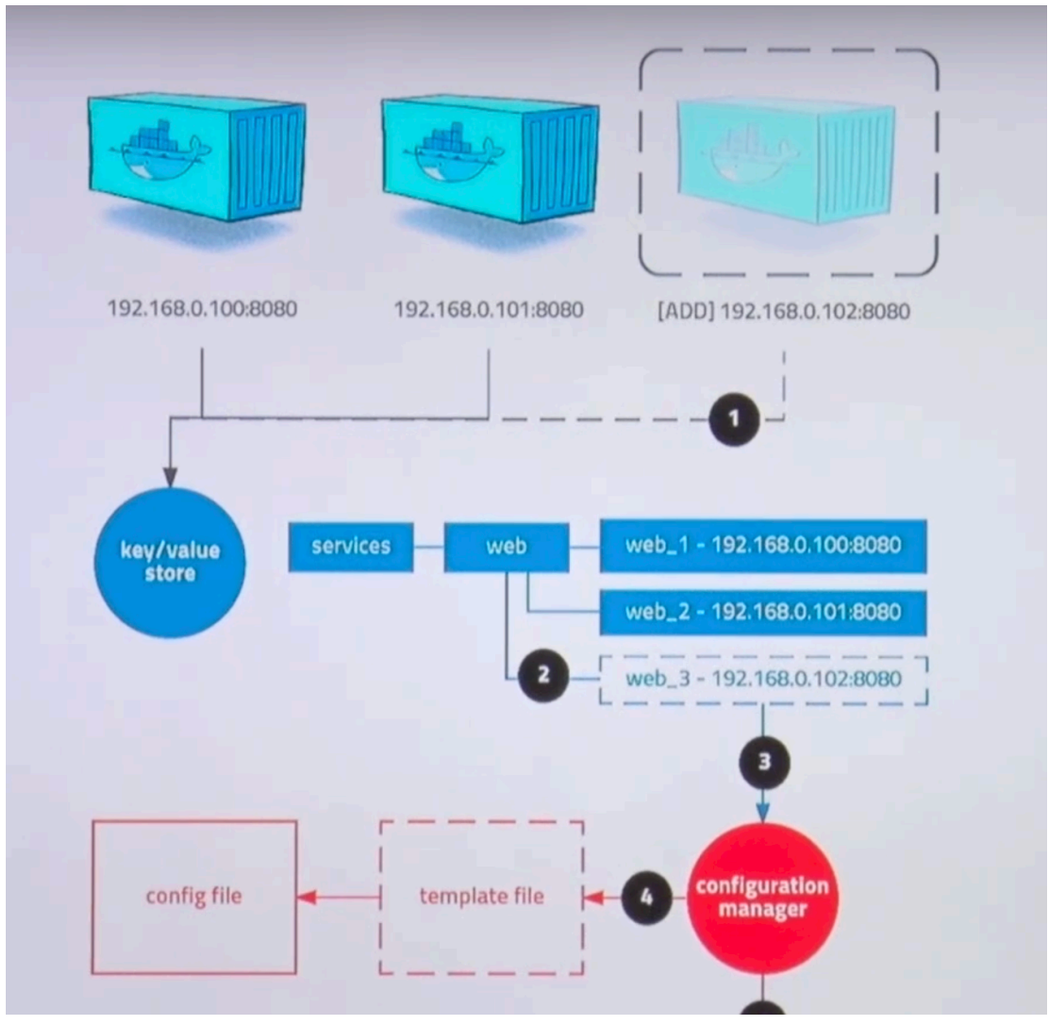
Introduction to Kubernetes

Docker container 구성



- 도커 엔진의 프로세스는 /usr/bin/dockerd 파일에서 실행 됩니다.
- 도커 엔진은 실제 컨테이너를 생성하고 이미지를 관리하며, 외부의 REST API로 통신한다.
- 도커 프로세스가 실행되어 서버에서 입력을 받을 준비가 된 상태를 도커 데몬이 관리한다.
- 도커의 이미지 생성은 dockerfile 스크립트를 활용하여 생성한다.



1. 도커 내부에서 컨테이너가 하나 생성이 되었을 때, key-value store에는 컨테이너 3대의 정보가 모두 담기게 된다.

2. Key-value store에서 nginx 프로파일을 변경 시키고, 도커 설정 매니저로 변경 사항을 프록시를 통해 알려준다.

3. Nginx 서버 재시작

4. 이렇게 서버를 재시작 할 수 있도록 도커 설정을 자동화 해주는 기능을 docker-gen 이라고 합니다.

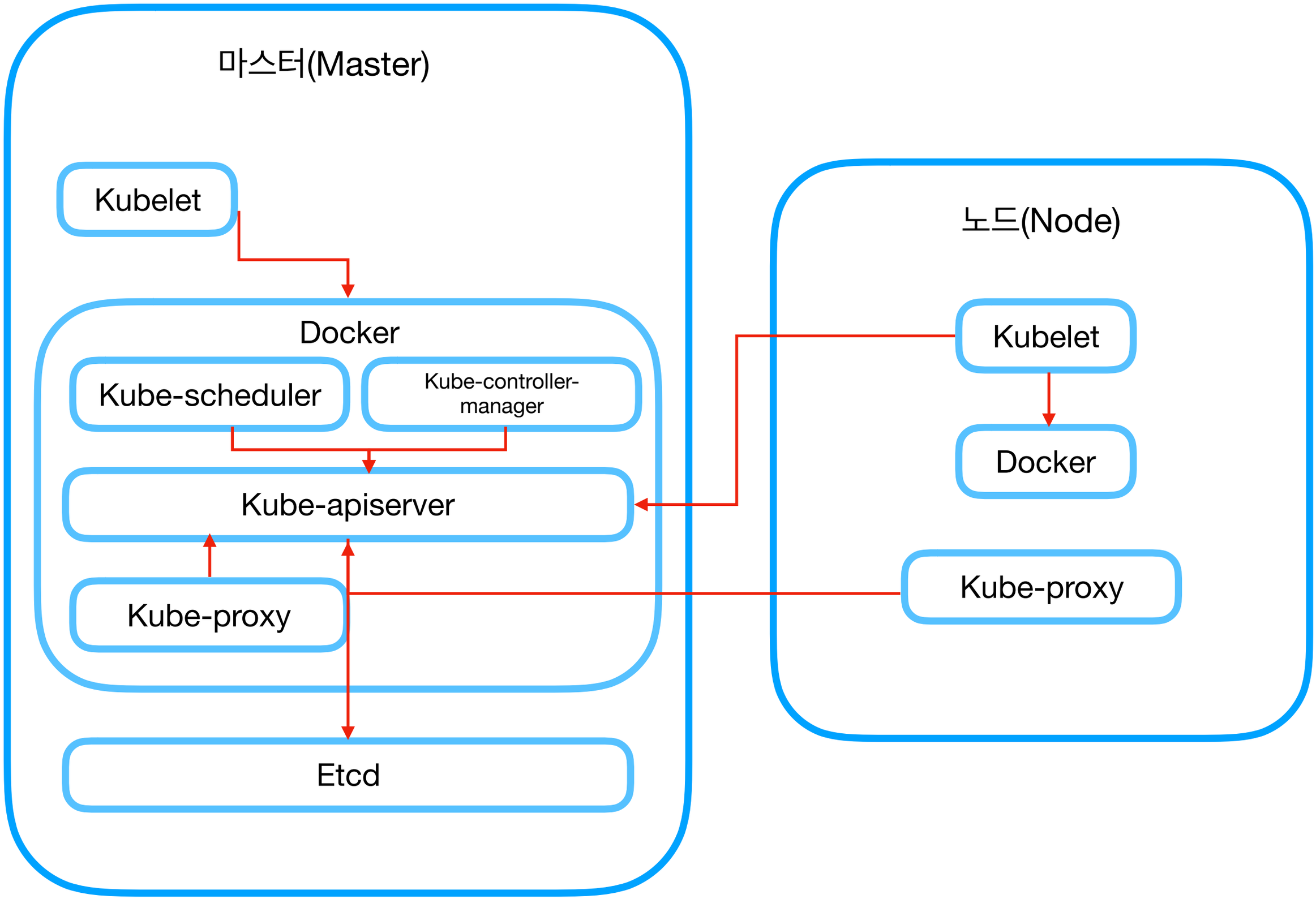
왜 Kubernetes가 살아 남았을까?

CLOUD NATIVE
COMPUTING FOUNDATION

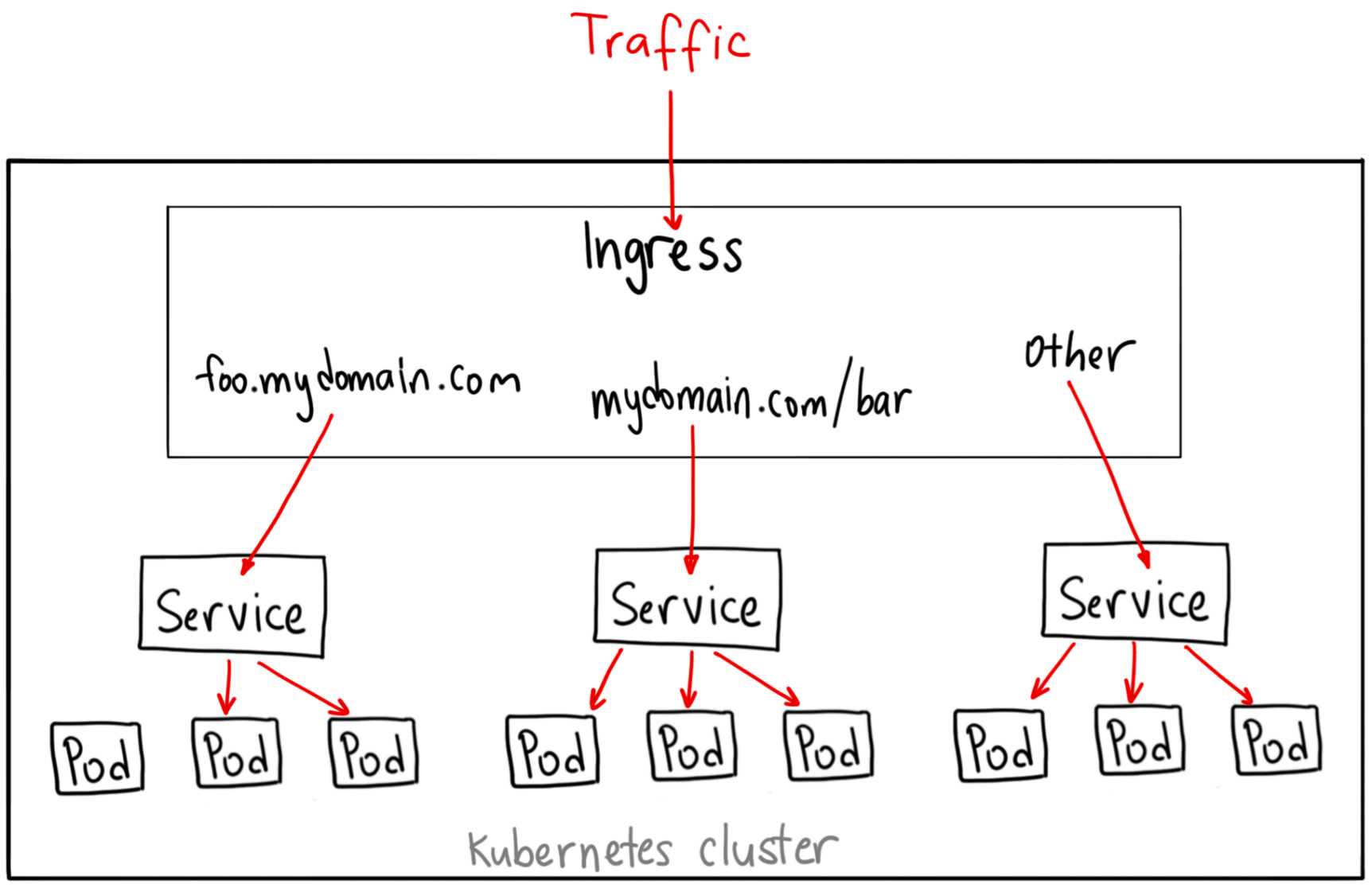
And

Google

Master - node 구조


마스터(Master)
Kubelet
Docker
Kube-scheduler
Kube-controller-manager
Kube-apiserver
Kube-proxy
Etcd
노드(Node)
Kubelet
Docker
Kube-proxy

Kubernetes - Ingress



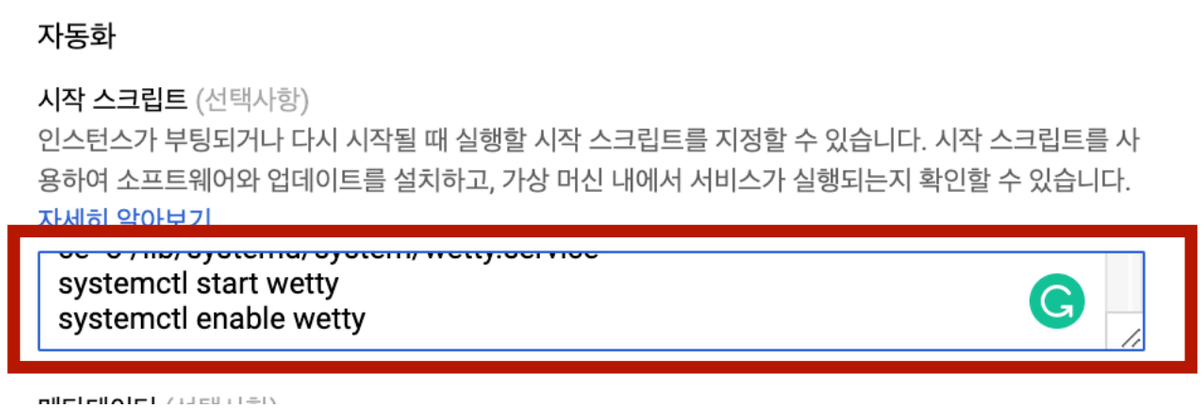
쿠버네티스에서 HTTPS 기반의 L7 로드밸런싱을 제공하는 기능을 Ingress 라고 한다.

Ingress는 서비스 앞에서 로드밸런서의 역할을 하며, url에 따라 라우팅하는 역할을 수행한다.

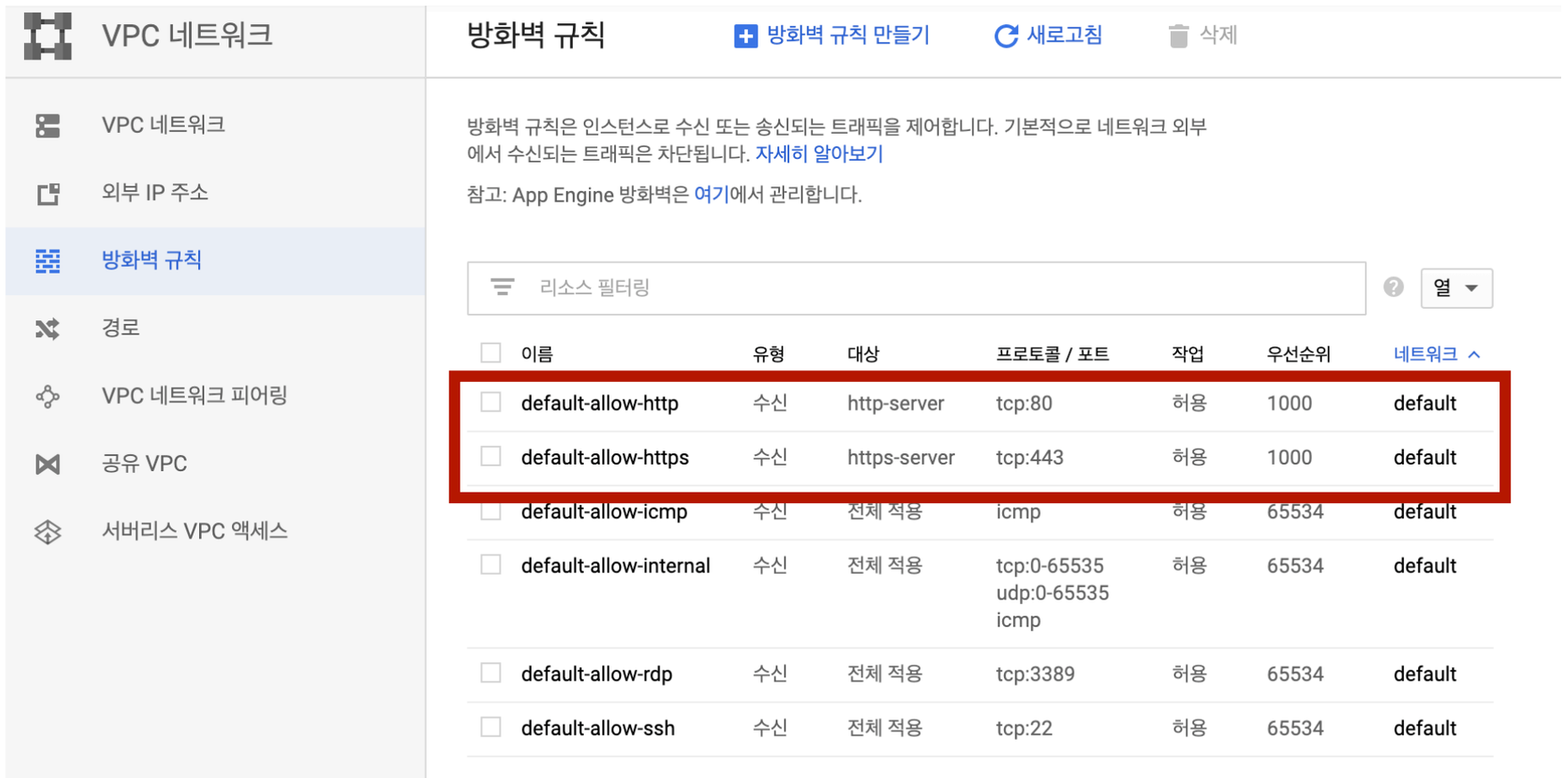
실습 환경 구성하기

1. 클라우드 플랫폼 기반으로 Docker와 Docker compose 설치하기

- 영상에서는 AWS Lightsail을 이용하였으나, GCP 클라우드 사용을 권장



1. VM 생성시, 시작 스크립트에 wetty install 코드 추가
2. 방화벽 규칙 > all tcp 규칙을 모두 허용으로 설정
> 모든 인스턴스에 적용



이름	외부 주소	리전	유형	버전	다음에서 사용 중
—	35.238.91.73	us-central1	임시	IPv4	VM 인스턴스 **ubuntu-1** (us-central1-a 영역)

주의 요함 | 35.238.91.73:4200

```
ubuntu-1 login: ubuntu
Password:
Last login: Wed Oct  2 05:03:40 UTC 2019 from localhost on pts/1
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1044-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Oct  2 06:27:03 UTC 2019

  System load:  0.16              Users logged in:                0
  Usage of /:   64.5% of 9.52GB   IP address for ens4:            10.128.0.2
  Memory usage: 18%               IP address for docker0:         172.17.0.1
  Swap usage:   0%                IP address for cni0:            10.42.0.1
  Processes:    130               IP address for br-f55f19afb233: 172.18.0.1


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

3 packages can be updated.
0 updates are security updates.


ubuntu@ubuntu-1:~$ docker-compose --version
docker-compose version 1.24.0, build 0aa59064
ubuntu@ubuntu-1:~$
```

3. 외부 ip 주소를 사용하여 4200번으로 통신

2. k3s 설치하기

Install

```
curl -sfL https://get.k3s.io | sh -
sudo chown ubuntu:ubuntu /etc/rancher/k3s/k3s.yaml
```

- kubectl 명령어를 사용하여 마스터 노드 생성 확인

```
ubuntu@ubuntu-1:~$ kubectl get nodes
NAME       STATUS   ROLES    AGE   VERSION
ubuntu-1   Ready    master   63s   v1.15.4-k3s.1
```

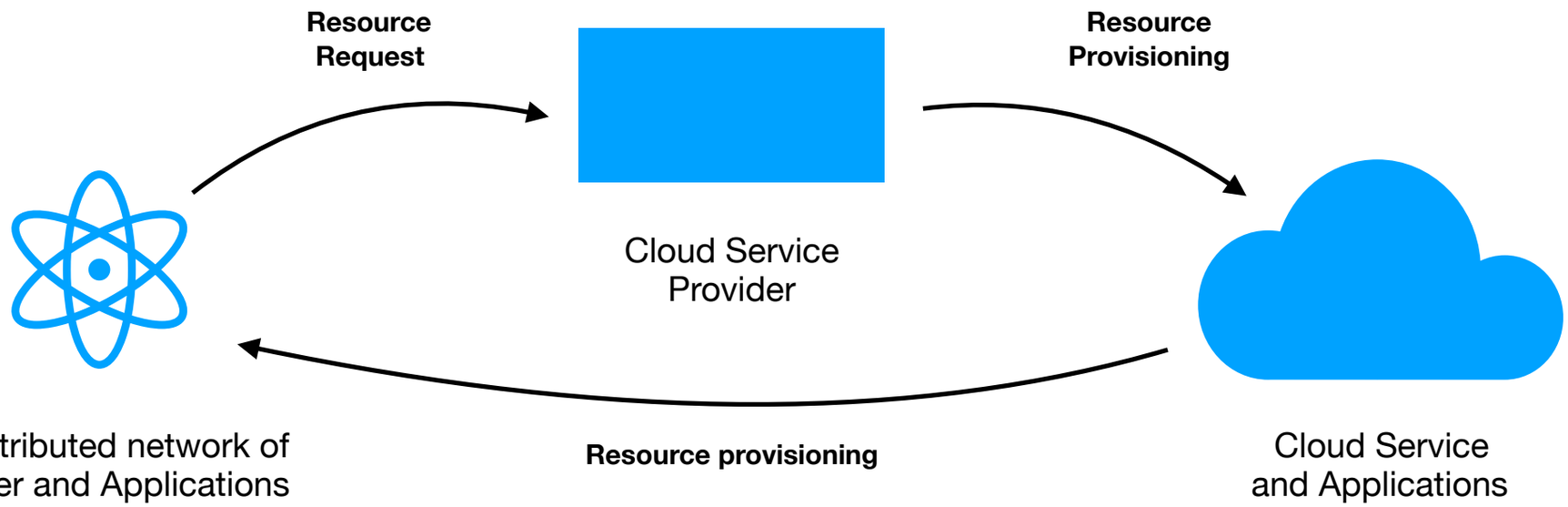
- k3s 설정 폴더에 config.yaml 복사

```
ubuntu@ubuntu-1:~$ cp /etc/rancher/k3s/k3s.yaml ~/.kube/config
ubuntu@ubuntu-1:~$
```

- storage class 대신, local path provisioner 설치 (kubectl apply, patch 명령어 사용)

```
ubuntu@ubuntu-1:~$ kubectl get storageclass
NAME                   PROVISIONER             AGE
local-path (default)   rancher.io/local-path   63s
```

What is a provisioning?

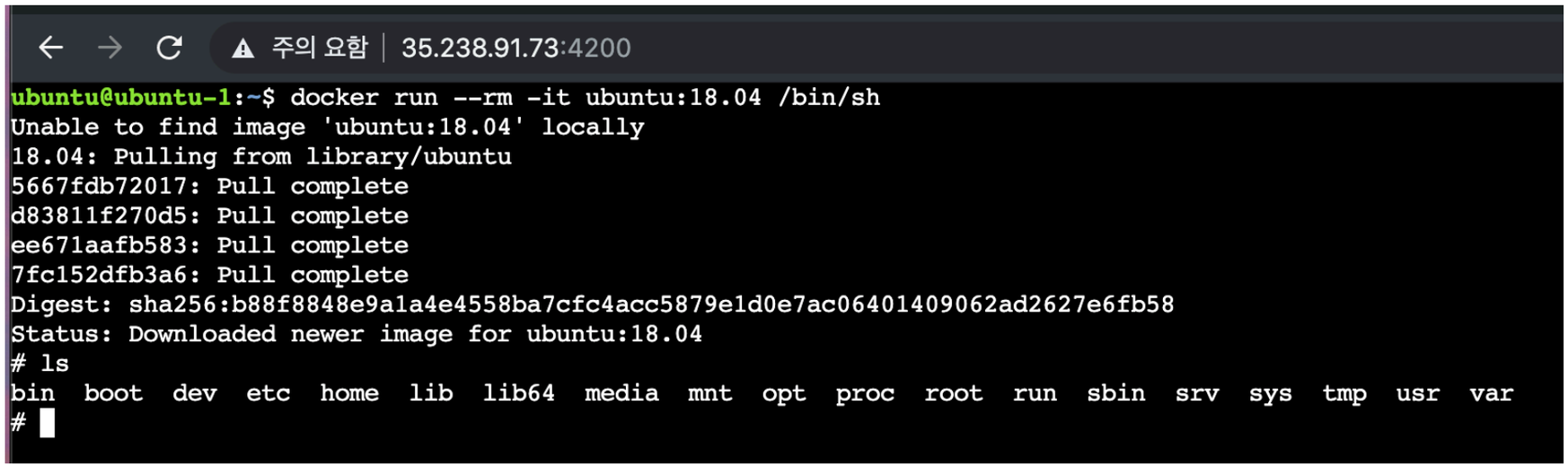


사용자의 요구에 맞게 system resource를 할당해 두었다가
필요 시 시스템을 즉시 사용할 수 있는 상태로 미리 준비 해두는 과정

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: standard
provisioner: kubernetes.io/aws-ebs
parameters:
  type: gp2
reclaimPolicy: Retain
allowVolumeExpansion: true
mountOptions:
  - debug
volumeBindingMode: Immediate
```

Docker 실습

- Ubuntu container 생성하기



- 이미지를 다운받아 웹 애플리케이션 생성하기

- MySQL 설치, DB 생성/삭제 작업하기

```
ubuntu@ubuntu-1:~$ docker run -d -p 3306:3306 \
>   -e MYSQL_ALLOW_EMPTY_PASSWORD=true \
>   --name mysql \
>   mysql:5.7
Unable to find image 'mysql:5.7' locally
5.7: Pulling from library/mysql
8f91359f1fff: Pull complete
6bbb1c853362: Pull complete
e6e554c0af6f: Pull complete
f391c1a77330: Pull complete
414a8a88eabc: Pull complete
fee78658f4dd: Pull complete
9568f6bff01b: Pull complete
76041efb6f83: Pull complete
ea54dbd83183: Pull complete
566857d8f022: Pull complete
01c09495c6e7: Pull complete
Digest: sha256:f7985e36c668bb862a0e506f4ef9acdd1254cdf690469816f99633898895f7fa
Status: Downloaded newer image for mysql:5.7
17bea4886f6b090bff55ea77d3d86b147389362ec27db71d1939c7aa3e7b50ce
ubuntu@ubuntu-1:~$ docker exec -it mysql mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.27 MySQL Community Server (GPL)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

이름을 mysql로 지정 하였으므로,
가상 네트워크 환경에서
컨테이너 끼리 통신할때는 port 대신
name 환경 변수로 통신이 가능함

```
ubuntu@ubuntu-1:~$ docker run -d -p 8000:80 \
>   --network=app-network \
>   -e WORDPRESS_DB_HOST=mysql \
>   -e WORDPRESS_DB_NAME=wp \
>   -e WORDPRESS_DB_USER=wp \
>   -e WORDPRESS_DB_PASSWORD=wp \
>   wordpress
79e29fbb30b68f2d3de61aaa2a1f97ac8320f901b2aa6bc2de8799774028e934
ubuntu@ubuntu-1:~$
```

- 컨테이너 목록 확인

$ docker ps -a

```
ubuntu@ubuntu-1:~$ docker ps -a
CONTAINER ID        IMAGE                             COMMAND                  CREATED             STATUS              PORTS                               NAM
ES
778cf5ef9e40        wordpress                         "docker-entrypoint.s…"   2 minutes ago       Up 2 minutes        0.0.0.0:8000->80/tcp                dis
tracted_cohen
17bea4886f6b        mysql:5.7                         "docker-entrypoint.s…"   5 minutes ago       Up 5 minutes        0.0.0.0:3306->3306/tcp, 33060/tcp   mys
ql
75e43aeed67e        subicura/docker-workshop-app:1    "/bin/sh -c 'bundle …"   10 minutes ago      Up 10 minutes       0.0.0.0:4567->4567/tcp              hap
py_lalande
```

컨테이너는 생성 후, 바로 삭제 단계로 갈 수 없다.
도커 내부에서 생성 -> 중지 -> 삭제 단계를 거친뒤에 컨테이너가 삭제된다.

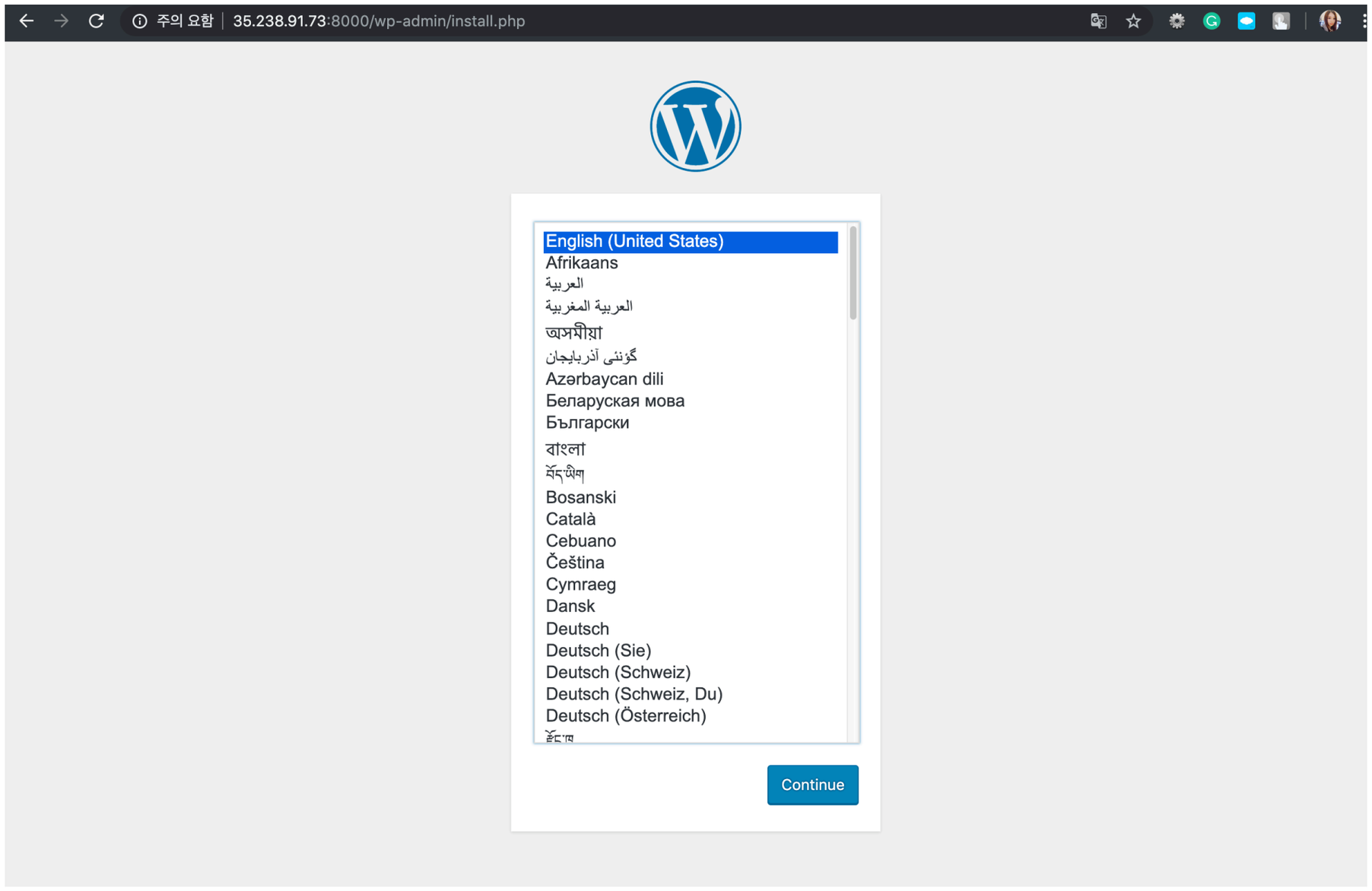
- 컨테이너 중지, 삭제

$ docker stop/rm

```
ubuntu@ubuntu-1:~$ docker stop 778cf5ef9e40 17bea4886f6b 75e43aeed67e
778cf5ef9e40
17bea4886f6b
75e43aeed67e
ubuntu@ubuntu-1:~$ docker rm 778cf5ef9e40 17bea4886f6b 75e43aeed67e
778cf5ef9e40
17bea4886f6b
75e43aeed67e
```

container image로 각각 웹의 frontend 와 backend를 구성할 수 있다.

immutable한 운영은 서버를 어렵지 않게 생성할 수 있고,
유지보수가 수월 하다는 장점이 있다.



Katacoda 실습