

2022-2학기 클라우드컴퓨팅 실습 #6

2022년 2학기 클라우드컴퓨팅
실습 조교 김명현 (freckie@korea.ac.kr)

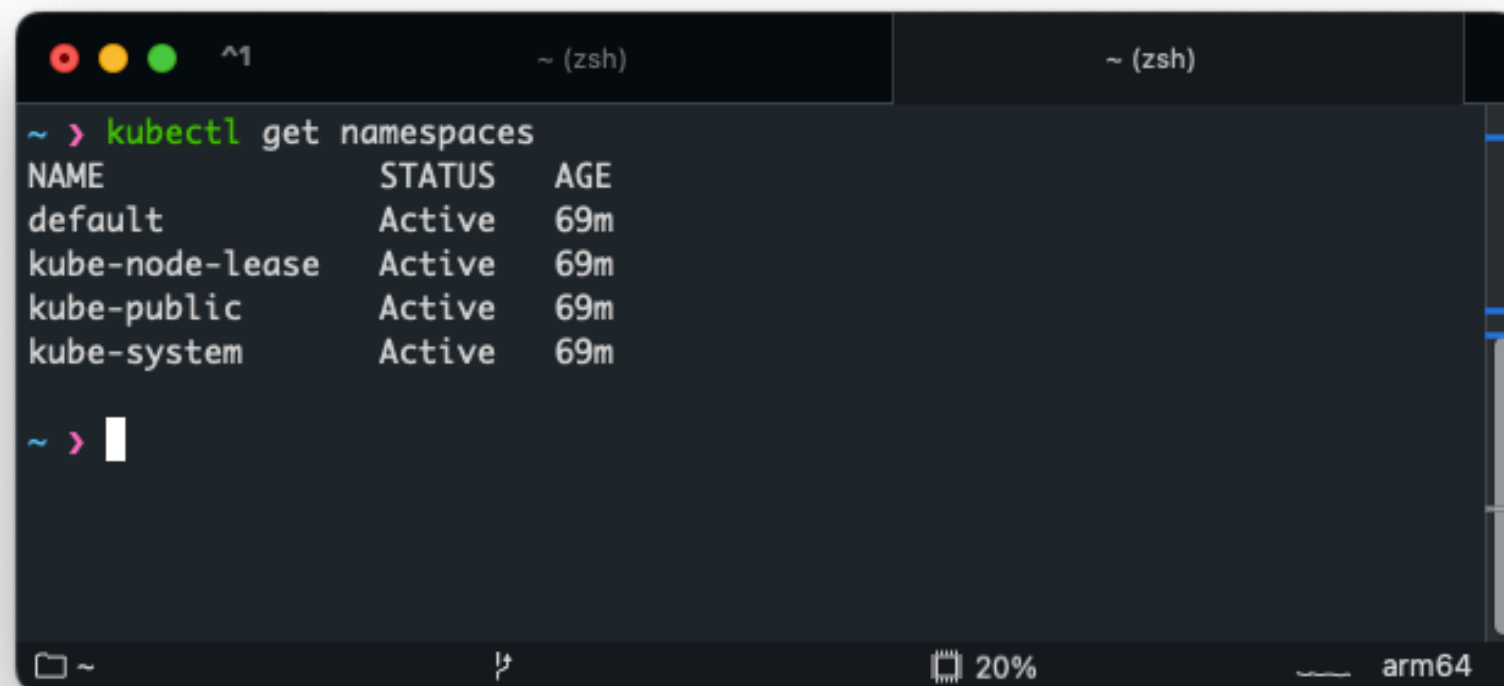
2022.11.30

Section 1

“Namespace”

Namespace

- 쿠버네티스의 리소스들을 논리적으로 분리
- 하나의 클러스터를 여러 개의 팀이나 프로젝트에 걸쳐서 많은 사용자가 있는 환경에서 서로를 분리하기 위한 오브젝트
- 리소스들의 이름이 적용되는 범위(scope)를 지정해주는 역할
 - 파드 이름 "testpod"은 namespace1에도 존재할 수 있고, namespace2에도 존재할 수 있음.



```
~ > kubectl get namespaces
NAME              STATUS    AGE
default           Active    69m
kube-node-lease   Active    69m
kube-public       Active    69m
kube-system       Active    69m
~ > 
```

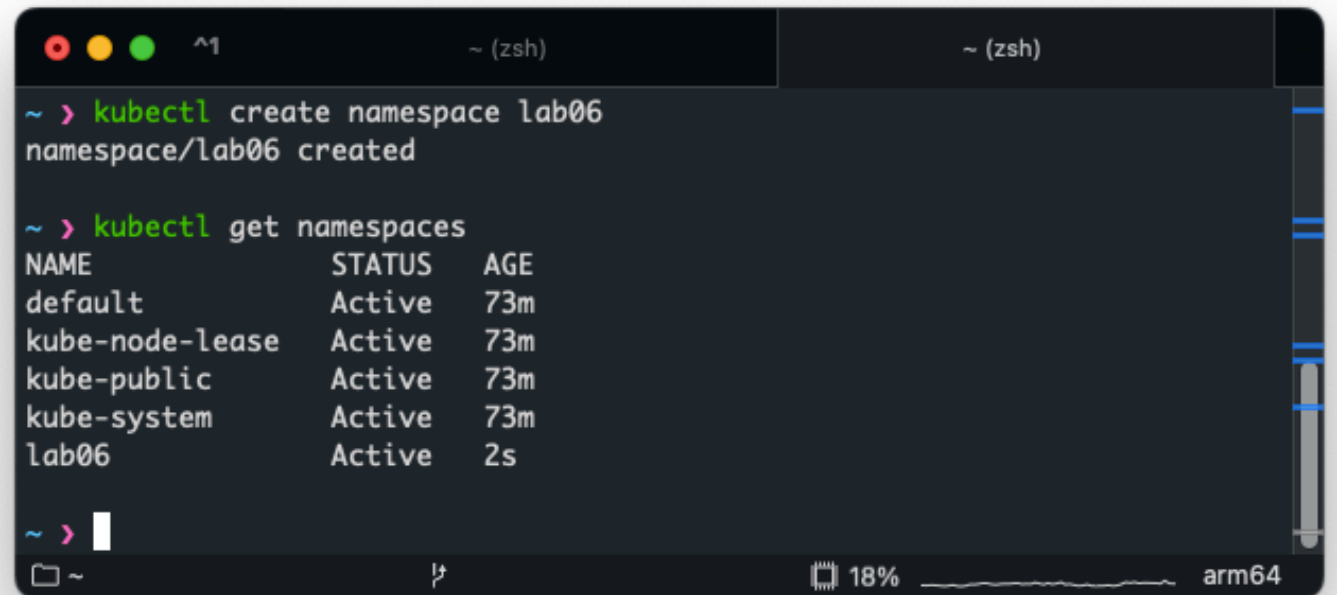
Reference : <https://kubernetes.io/ko/docs/concepts/overview/working-with-objects/namespaces/>

1. Namespace

Namespace

- 네임스페이스의 생성 (via kubectl)

\$ kubectl create namespace lab06



```
~ > kubectl create namespace lab06
namespace/lab06 created

~ > kubectl get namespaces
NAME                STATUS    AGE
default             Active    73m
kube-node-lease     Active    73m
kube-public         Active    73m
kube-system         Active    73m
lab06               Active    2s
```

- 네임스페이스의 생성 (via .yaml)

\$ kubectl apply -f namespace.yaml

```
# namespace.yaml

apiVersion: v1
kind: Namespace
metadata:
  name: lab6
```

Namespace

- 파드의 네임스페이스 지정

```
# pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  namespace: lab06
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
```

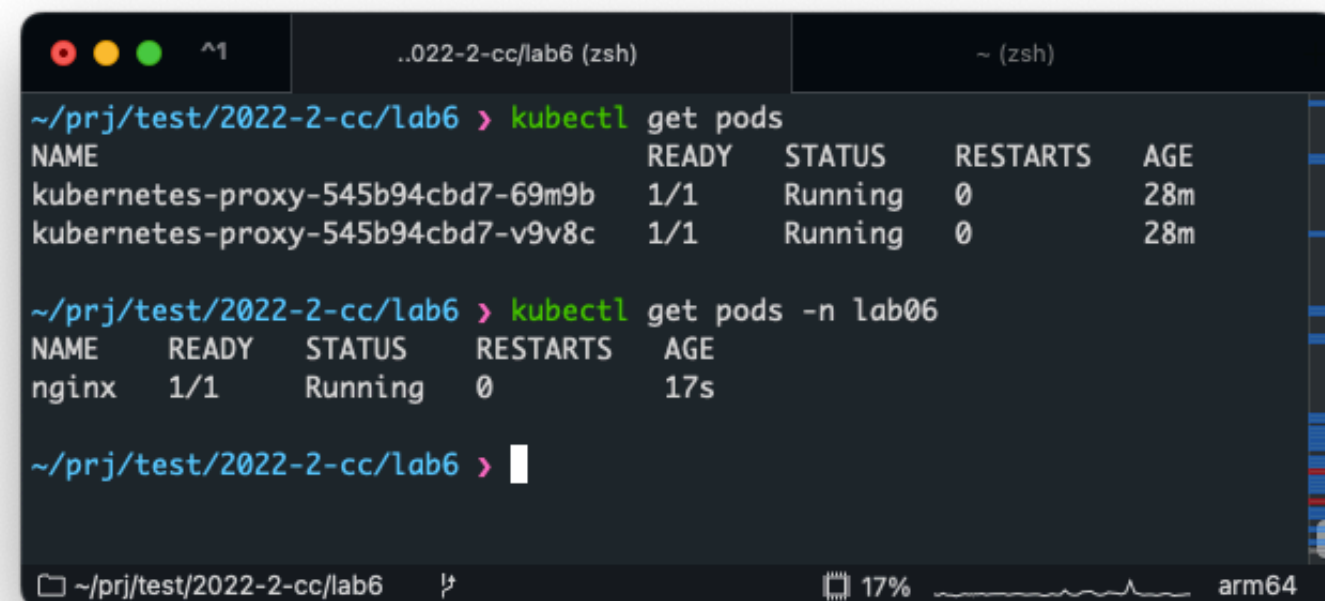
- 네임스페이스 내의 파드 검색

\$ kubectl get pods -n lab06

→ -n : namespace 지정 옵션

→ kubectl get 명령에 모두 지정 가능

- 네임스페이스에 상관 없이 모든 파드를
검색 → kubectl get pods -A



```
~/prj/test/2022-2-cc/lab6 > kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
kubernetes-proxy-545b94cbd7-69m9b   1/1     Running   0           28m
kubernetes-proxy-545b94cbd7-v9v8c   1/1     Running   0           28m

~/prj/test/2022-2-cc/lab6 > kubectl get pods -n lab06
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0           17s

~/prj/test/2022-2-cc/lab6 >
```

Section 2

“Basic Networking”

Introduction

- 쿠버네티스 네트워크에서 파드는 기본적으로 서로 격리되어 있음.
 - 다만, 대부분 애플리케이션은 외부의 요청에 대해 결과값을 반환하기를 원하고, 네트워크를 통해 파드끼리 통신해야하는 상황이 있다.

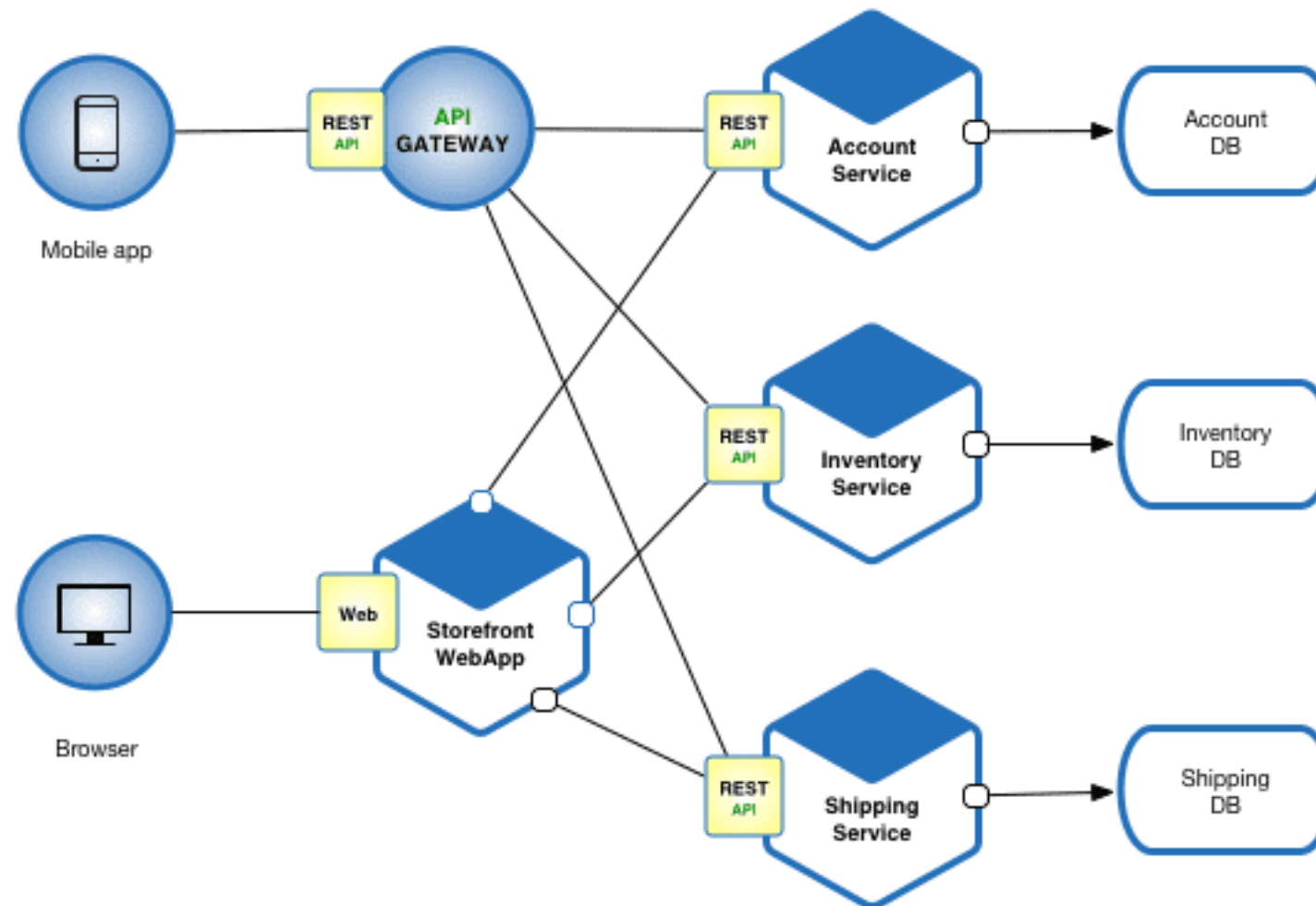


Figure Reference : <https://microservices.io>

Introduction

Q) 각 파드에는 IP가 할당되므로 파드가 다른 파드의 IP에 직접 요청하면 되지 않을까?

→ 다음 이유로 인해 쿠버네티스에서는 좋은 방법이 아님.

1) 파드는 언제든지 생성되고 소멸될 수 있다. (ephemeral)

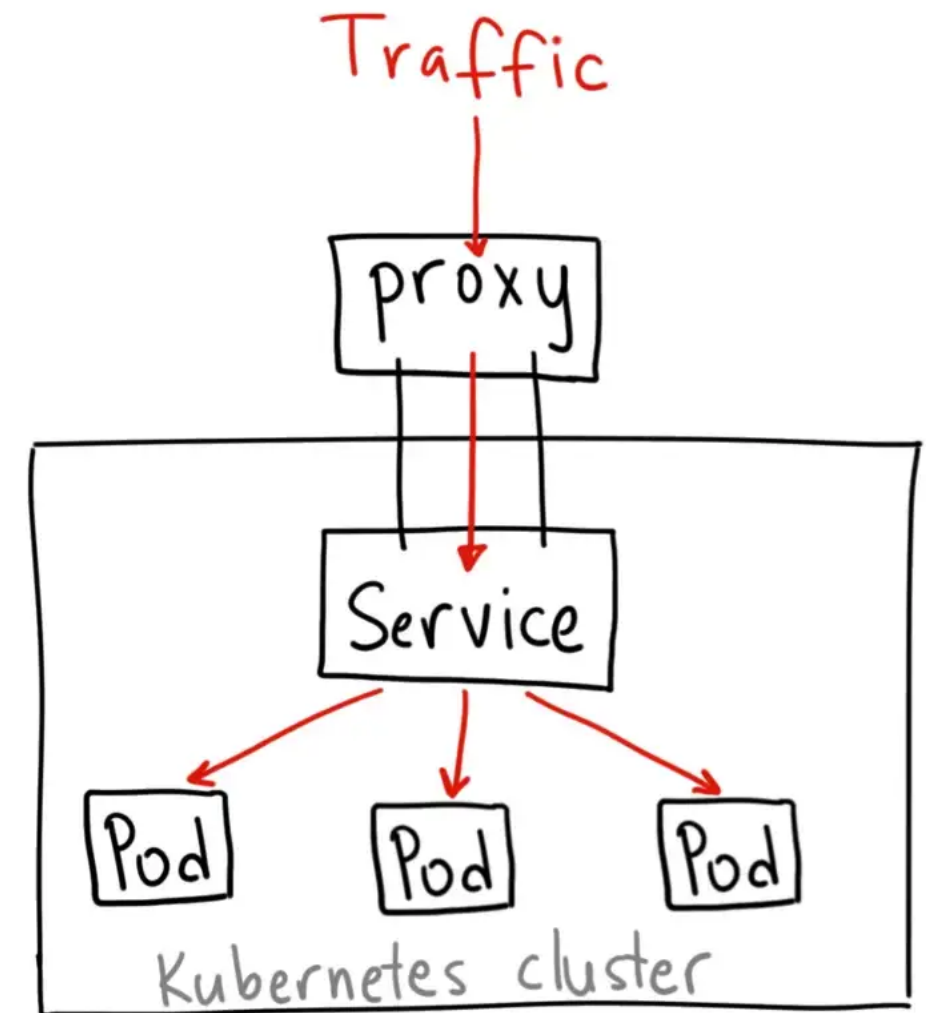
2) 쿠버네티스 시스템이 파드에 IP를 할당할 때, 파드가 배포될 노드가 선택된 후 실제 파드 시작 전에 IP가 할당된다.

→ 클라이언트가 타겟 파드의 IP 주소를 알기 어렵다.

3) scale-out 수행 시 파드의 수가 늘어나는데, 이때마다 새로 할당되는 IP를 매번 클라이언트가 알기는 어렵다.

Service

- 같은 애플리케이션을 구동하는 파드들에 대해서,
변하지 않는 하나의 접근 포인트를 제공함.
- 직접 변경하지 않는 한, 서비스가 노출하는 IP 주소와 포트는 변하지 않음.
- 서비스는 파드의 논리적 집합과, 그 파드들에 접근할 수 있는 정책을 정의하는 추상적인 개념



(ClusterIP, NodePort, LoadBalancer 타입의 서비스가 존재하며, 추후 설명)

- ☑ 파드 집합에서 실행 중인 애플리케이션을 네트워크 서비스로 노출하는 추상화 방법

Figure Reference : <https://medium.com/google-cloud/kubernetes-nodeport-vs-loadbalancer-vs-ingress-when-should-i-use-what-922f010849e0>

2. Basic Networking

Service

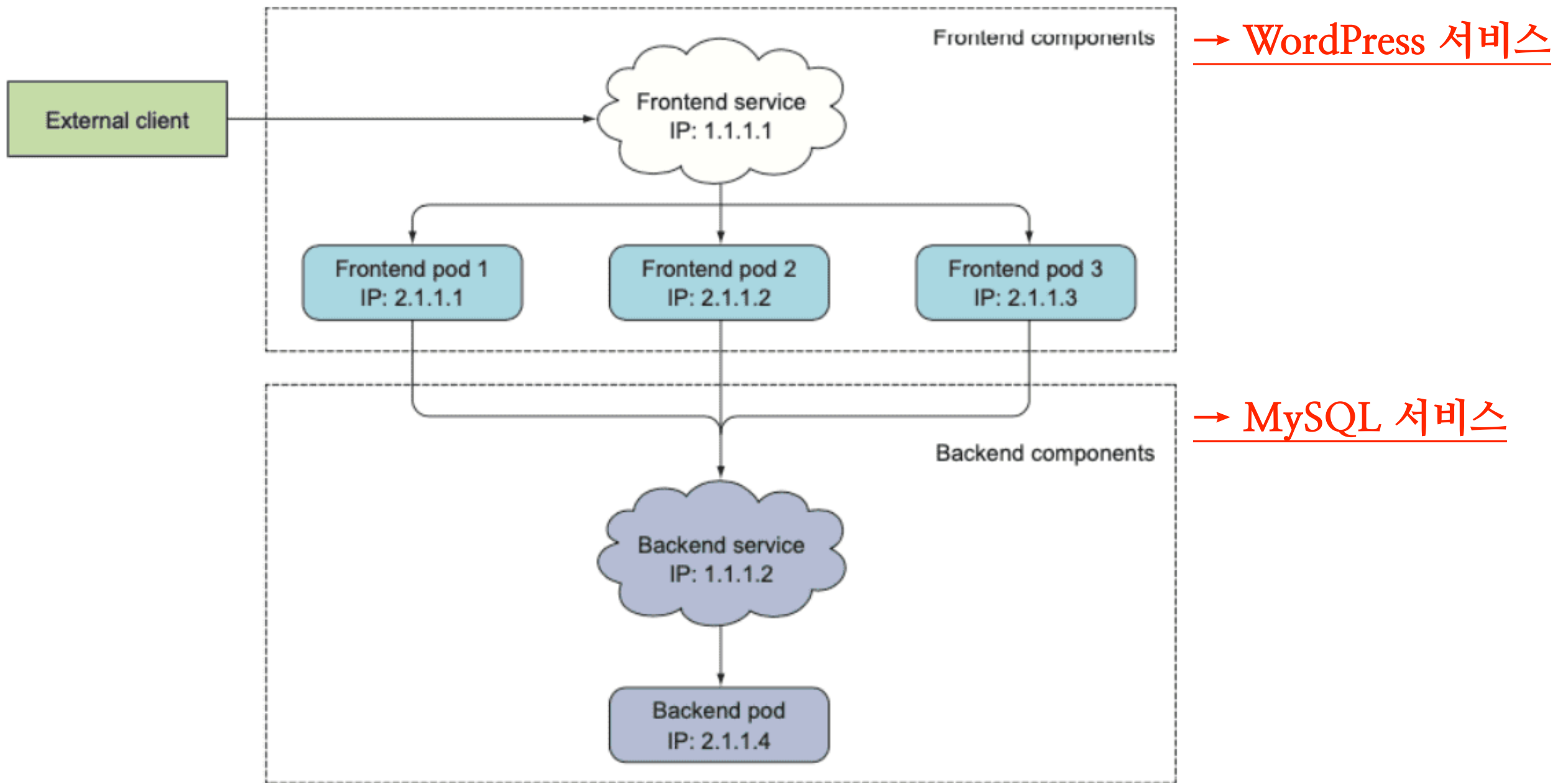


Figure Reference : Marco Luksa, "Kubernetes in Action", Manning Publications, 2017.

ClusterIP Service

- **ClusterIP 타입의 서비스** → 가장 기본적인 타입의 서비스
 - 파드를 클러스터 내부에서 접속 가능하도록 노출하는 역할
 - 클러스터 외부와는 격리되어 있으며 접속이 불가능함.
- ☑ DB 등의 서비스는 같은 클러스터에 배포된 프론트엔드 애플리케이션에서는 접속이 가능해야 하지만, 클러스터 외부의 유저의 접속은 막아야 한다.

2. Basic Networking

ClusterIP Service

- ClusterIP 생성 (via kubectl)

\$ kubectl apply -f deployment.yaml (실습#5에서 작성한 deployment.yaml 수정)

\$ kubectl expose deployment nginx-deployment -n lab06 --name=nginx-service

\$ kubectl get services -n lab06

```
# deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  namespace: lab06
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx-front
  template:
    metadata:
      name: nginx
    labels:
      app: nginx-front
    spec:
      containers:
        - name: nginx
          image: nginx:1.14.2
          ports:
            - containerPort: 80
```

실습5에서 작성했던
deployment.yaml

```
..022-2-cc/lab6 (zsh) ~ (zsh)
~/prj/test/2022-2-cc/lab6 > kubectl get pods -n lab06
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-5b4dcff5b7-5bp9f   1/1     Running   0           90s
nginx-deployment-5b4dcff5b7-79lhn   1/1     Running   0           90s
nginx-deployment-5b4dcff5b7-nfqzg   1/1     Running   0           90s

~/prj/test/2022-2-cc/lab6 >
```

```
..022-2-cc/lab6 (zsh) ~ (zsh)
~/prj/test/2022-2-cc/lab6 > kubectl expose deployment nginx-deployment --name=nginx-service -n lab06
service/nginx-service exposed

~/prj/test/2022-2-cc/lab6 > kubectl get services -n lab06
NAME             TYPE       CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
nginx-service    ClusterIP  172.16.255.35 <none>        80/TCP     4s

~/prj/test/2022-2-cc/lab6 >
```

2. Basic Networking

ClusterIP Service

- ClusterIP 생성 (via .yaml)

\$ kubectl apply -f deployment.yaml (실습#5에서 작성한 deployment.yaml 수정)

\$ kubectl apply -f service.yaml

```
# deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  namespace: lab06
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx-front
  template:
    metadata:
      name: nginx
      labels:
        app: nginx-front
    spec:
      containers:
        - name: nginx
          image: nginx:1.14.2
          ports:
            - containerPort: 80
```

실습5에서 작성했던
deployment.yaml

```
# service.yaml
kind: Service
apiVersion: v1
metadata:
  name: nginx-service
  namespace: lab06
spec:
  selector:
    app: nginx-front
  type: ClusterIP
  ports:
    - port: 8000
      targetPort: 80
```

```
..022-2-cc/lab6 (zsh) ~ (zsh)
~/prj/test/2022-2-cc/lab6 > kubectl apply -f service.yaml
service/nginx-service unchanged

~/prj/test/2022-2-cc/lab6 > kubectl get services -n lab06
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
nginx-service	ClusterIP	172.16.255.87	<none>	8000/TCP	98s

```
~/prj/test/2022-2-cc/lab6 >
```

- port: 서비스가 노출할 포트

- targetPort: 파드가 노출한 포트

{서비스IP}:port → {파드IP}:targetPort

172.16.255.87:8000 → (nginx파드):80

ClusterIP Service

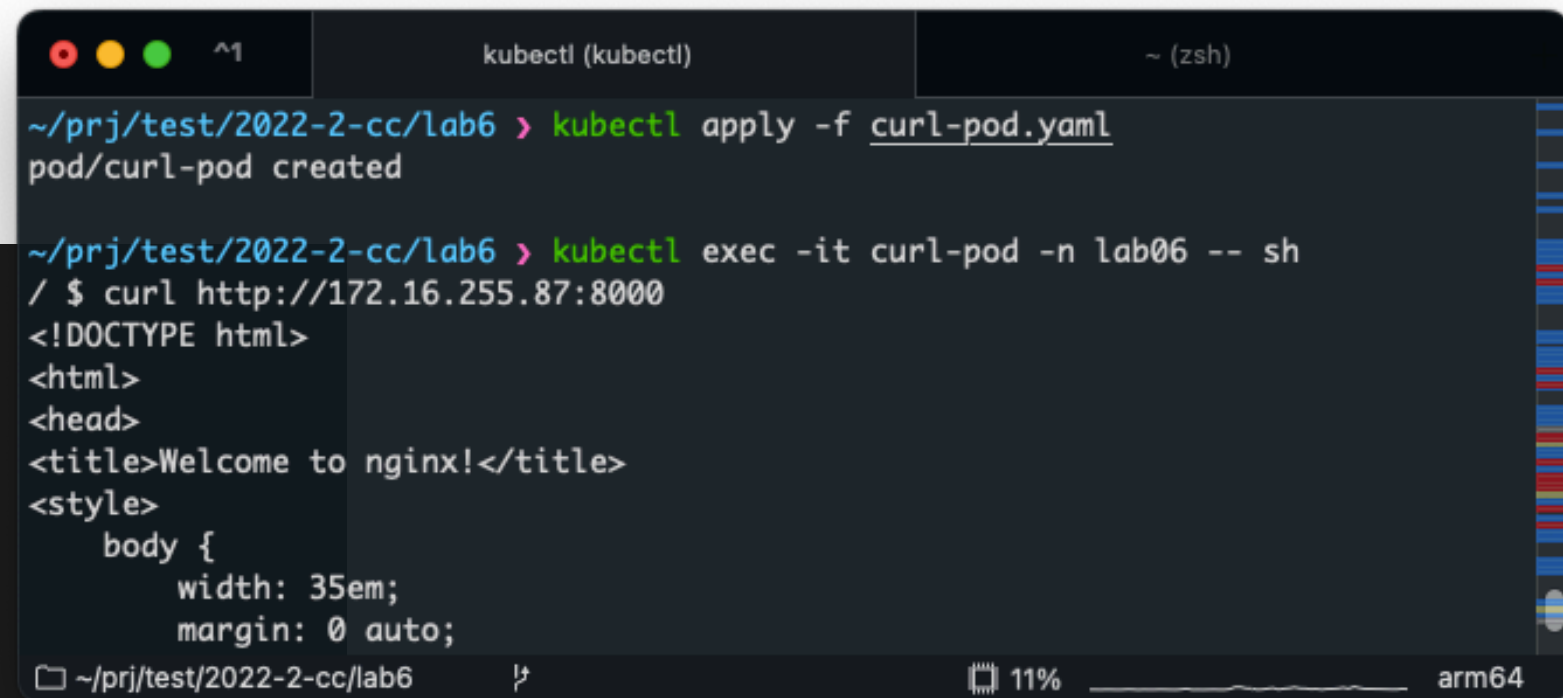
- ClusterIP Service 접근해보기

\$ kubectl apply -f curl-pod.yaml (클러스터 내부에서 curl 요청을 날리기 위한 파드)

\$ kubectl exec -it curl-pod -n lab06 -- sh (curl-pod에 접속해서 셸을 연다.)

(curl-pod) \$ curl http://172.16.255.87:8000

```
# curl-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: curl-pod
  namespace: lab06
spec:
  containers:
  - name: curl
    image: curlimages/curl
    command: ['sh', '-c', 'echo hello, world! && sleep 3600']
```

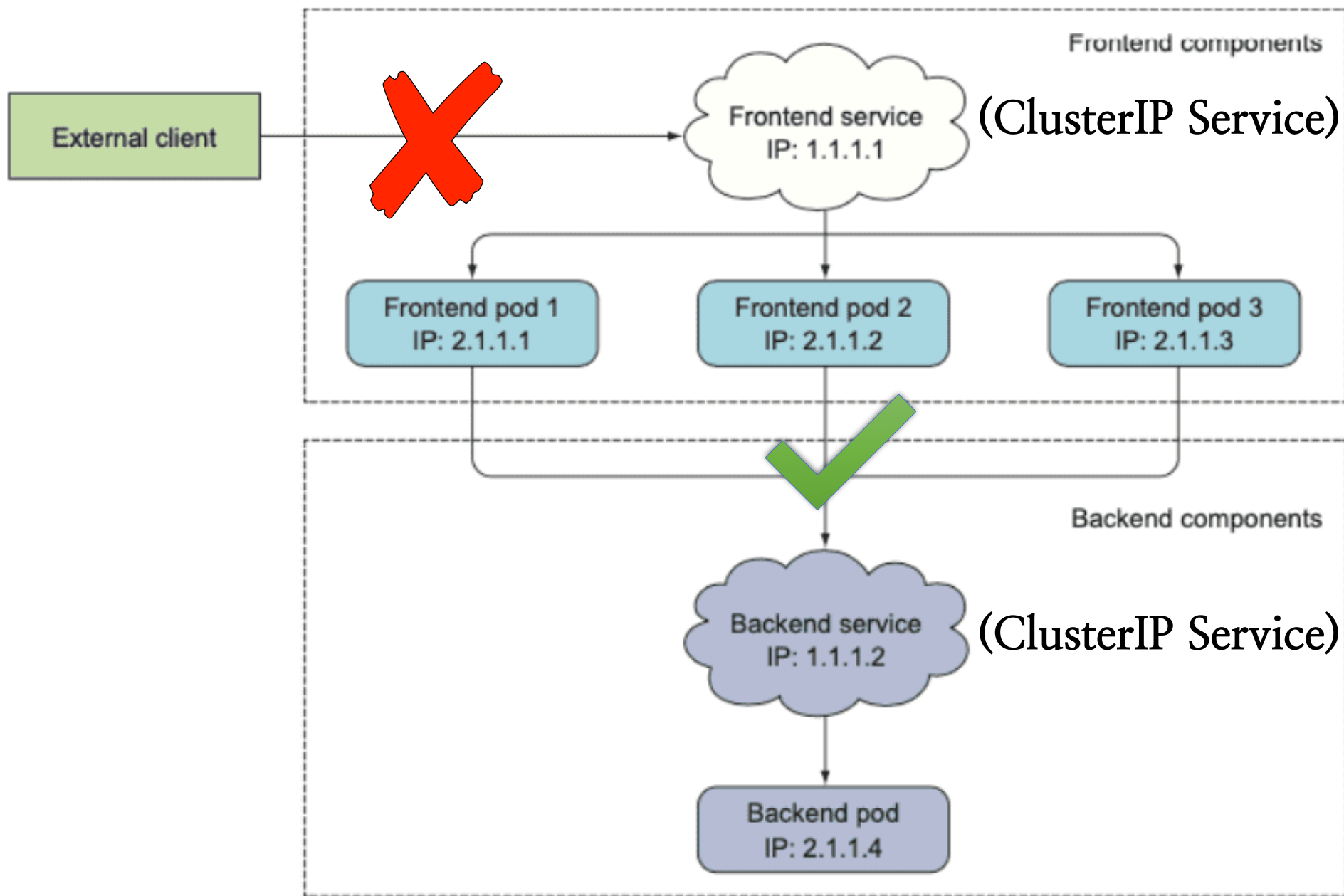


```
kubectl (kubectl) ~ (zsh)
~/prj/test/2022-2-cc/lab6 > kubectl apply -f curl-pod.yaml
pod/curl-pod created

~/prj/test/2022-2-cc/lab6 > kubectl exec -it curl-pod -n lab06 -- sh
/ $ curl http://172.16.255.87:8000
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
  }
</style>
</html>
```


2. Basic Networking

ClusterIP Service



→ WordPress 서비스

→ MySQL 서비스

Figure Reference : Marco Luksa, "Kubernetes in Action", Manning Publications, 2017.

NodePort Service

- **NodePort 타입의 서비스** → 클러스터 내 모든 노드들의 같은 포트에 서비스를 노출
 - ClusterIP 타입 서비스를 기반으로 구현됨.
 - 클러스터 외부에서도 접속 가능
 - 클러스터를 구성하는 임의의 노드에 접속해도 서비스에 연결됨.
 - ex) nodePort를 30123로 설정하면, {노드1_IP}:30123 이나 {노드2_IP}:30123 모두 접속이 가능함.
 - 외부에 서비스를 노출하기 위한 가장 간단한 방법

2. Basic Networking

NodePort Service

```
# service-nodeport.yaml
kind: Service
apiVersion: v1
metadata:
  name: nginx-service
  namespace: lab06
spec:
  selector:
    app: nginx-front
  type: NodePort
  ports:
    - port: 8000
      targetPort: 80
      nodePort: 30123
```

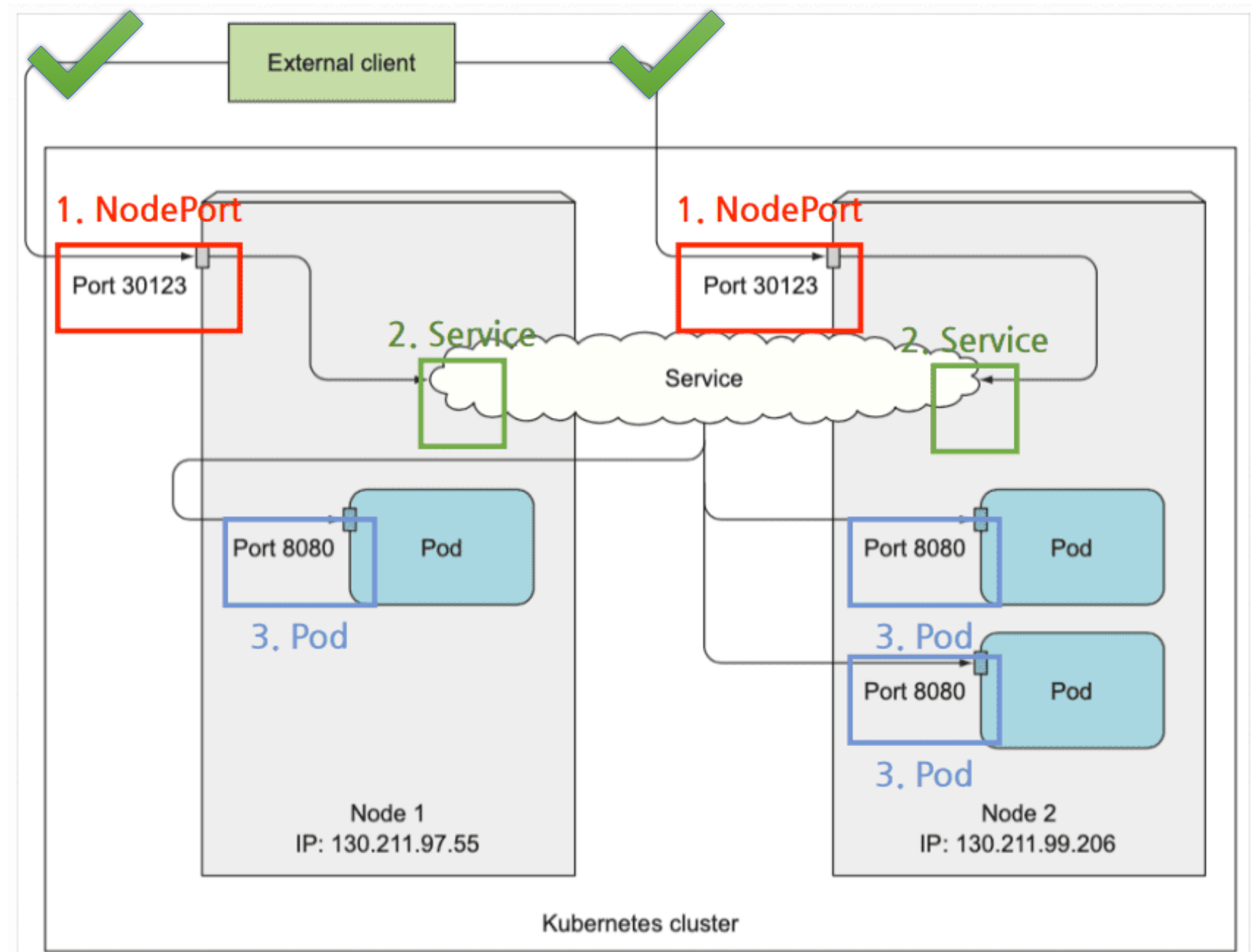


Figure Reference : Marco Luksa, "Kubernetes in Action", Manning Publications, 2017.

NodePort Service

```
# service-nodeport.yaml
kind: Service
apiVersion: v1
metadata:
  name: nginx-service
  namespace: lab06
spec:
  selector:
    app: nginx-front
  type: NodePort
  ports:
  - port: 8000
    targetPort: 80
    nodePort: 30123
```

- port: 서비스가 노출할 포트
- targetPort: 파드가 노출한 포트
- nodePort: 클러스터 외부로 노출할 포트
($30000 \leq nodePort \leq 32767$)

{노드IP}:nodePort → {서비스IP}:port → {파드IP}:targetPort

Figure Reference : Marco Luksa, "Kubernetes in Action", Manning Publications, 2017.

2. Basic Networking

NodePort Service

- NodePort Service 접근해보기

(로컬) \$ curl http://43.133.226.68:30123 (외부로 노출된 클러스터 노드의 IP)

```
# service-nodeport.yaml
kind: Service
apiVersion: v1
metadata:
  name: nginx-service
  namespace: lab06
spec:
  selector:
    app: nginx-front
  type: NodePort
  ports:
    - port: 8000
      targetPort: 80
      nodePort: 30123
```

```
..022-2-cc/lab6 (zsh) ~ (zsh)
~/prj/test/2022-2-cc/lab6 > kubectl apply -f service-nodeport.yaml
service/nginx-service created

~/prj/test/2022-2-cc/lab6 > kubectl get services -n lab06
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
nginx-service NodePort     172.16.255.159 <none>        8000:30123/TCP   12s

~/prj/test/2022-2-cc/lab6 >
```

```
..022-2-cc/lab6 (zsh) ~ (zsh)
~/prj/test/2022-2-cc/lab6 14s > curl http://43.133.226.68:30123
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
```

2. Basic Networking

The screenshot displays the Tencent Cloud console interface for managing a Kubernetes Engine (TKE) cluster. The left sidebar shows the navigation menu with options like Overview, Cluster, Service Mesh, Application, Helm, Images, Ops, Cluster Ops, TMP, Log Collection(Legacy), and Configuration management. The main content area shows the details for a specific cluster named 'tke-2021010760(c...)' in the Seoul region. The 'Node management' section is active, showing a list of nodes. A red box highlights the IP address 43.133.226.68 and the memory 10.82.0.9 Gi in the node table.

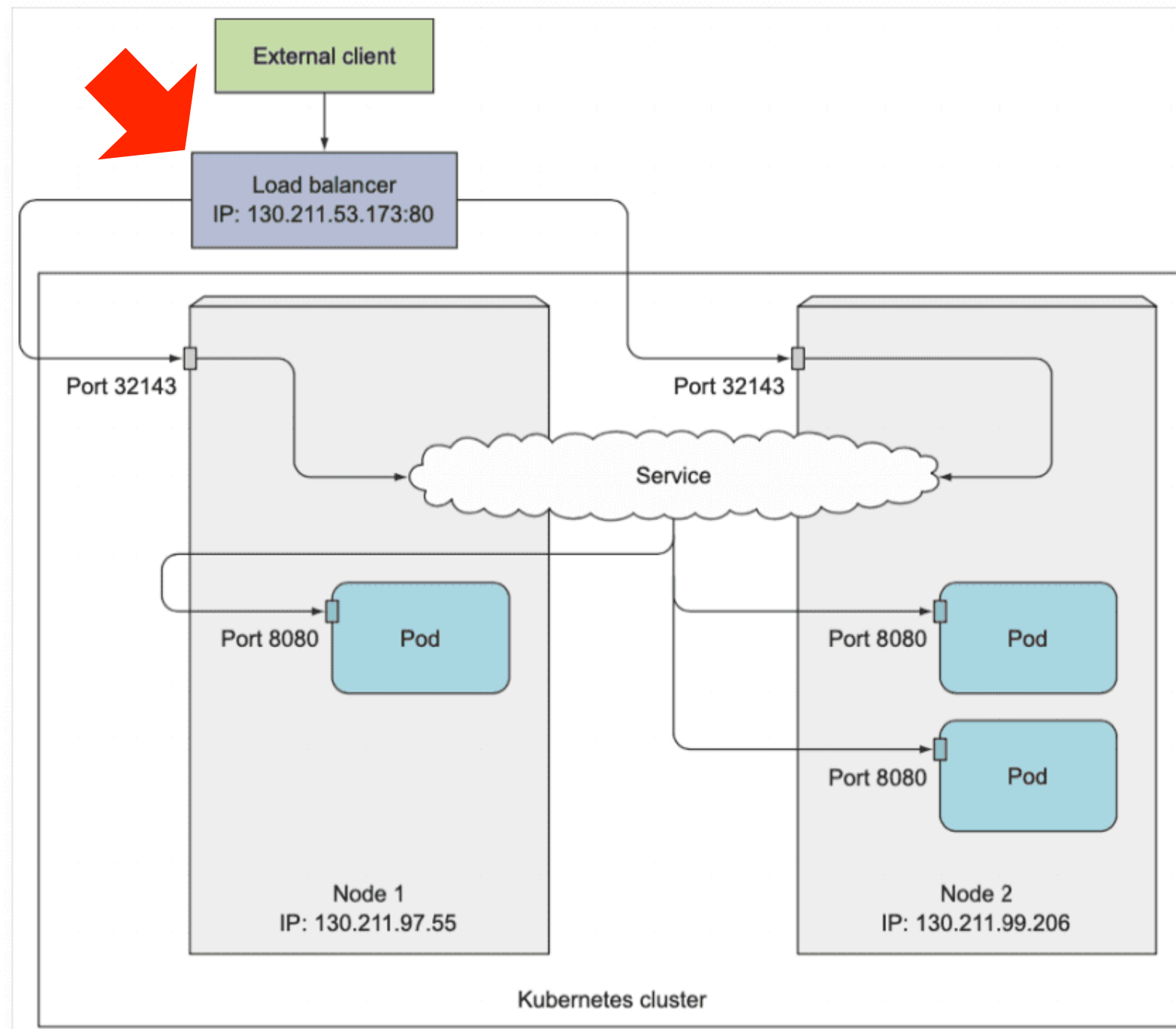
Node...	Ku...	Ru...	Config...	IP ad...	Res...	Billin...	Operation
<input type="checkbox"/> ins-ka2wm	H			43.133.226.68	CPU: 0.91 / 1.90 - core	Pay-a...	Remove Cordon
<input type="checkbox"/> tke_cls-gxjjbcrx_w orker	ea	...	v1.... co...	10.82.0.9	Memo ry: 0.60 / 1.34 Gi	Created I	More

LoadBalancer Service

- LoadBalancer 타입의 서비스
 - NodePort 타입 서비스를 기반으로 구현됨.
 - 클러스터 외부에서도 접속 가능
 - 상용 클라우드 플랫폼에서 제공하는 load balancer와 연동
 - 상용 클라우드 서비스를 사용하지 않는 경우 연동이 어려움
 - 로드밸런싱 기능 제공

2. Basic Networking

LoadBalancer Service



Section 3

“ 과제 #2 ”

개요

주제: WordPress, MySQL을 이용해 설치형 블로그를 쿠버네티스에 배포해보기

기간 : 11.30(수) ~ 12.12(월) 10:29

- 실습 수업이 더 없는 이유로 해설 및 예시 답안은 12.14(수) 블랙보드에 업로드됩니다.

결과물

- 다음을 학번_이름.zip 으로 압축하여 블랙보드 과제2 제출란에 제출 (총 파일 3개)
 - 보고서 (학번_이름.pdf)
 - db.yaml
 - blog.yaml

상세 - 조건

- WordPress, MySQL은 컨테이너 이미지를 사용 (실습#4 docker-compose 참고)
 - 사용하는 이미지 예시 : mysql:5.7, wordpress:latest
- 과제에서 배포하는 모든 리소스들은 assignment2 라는 이름의 네임스페이스를 만들어서 그 안에서 수행
- MySQL은 클러스터 내에서만 접근 가능하도록 서비스를 생성하고,
- WordPress는 클러스터 외부에서도 접근 가능하도록 서비스를 생성

상세 - 결과물 안내

결과물 1) db.yaml 파일

- MySQL과 관련된 내용을 모두 포함하여 작성
 - (deployment, service 등)

결과물 2) blog.yaml 파일

- WordPress와 관련된 내용을 모두 포함하여 작성
 - (deployment, service 등)

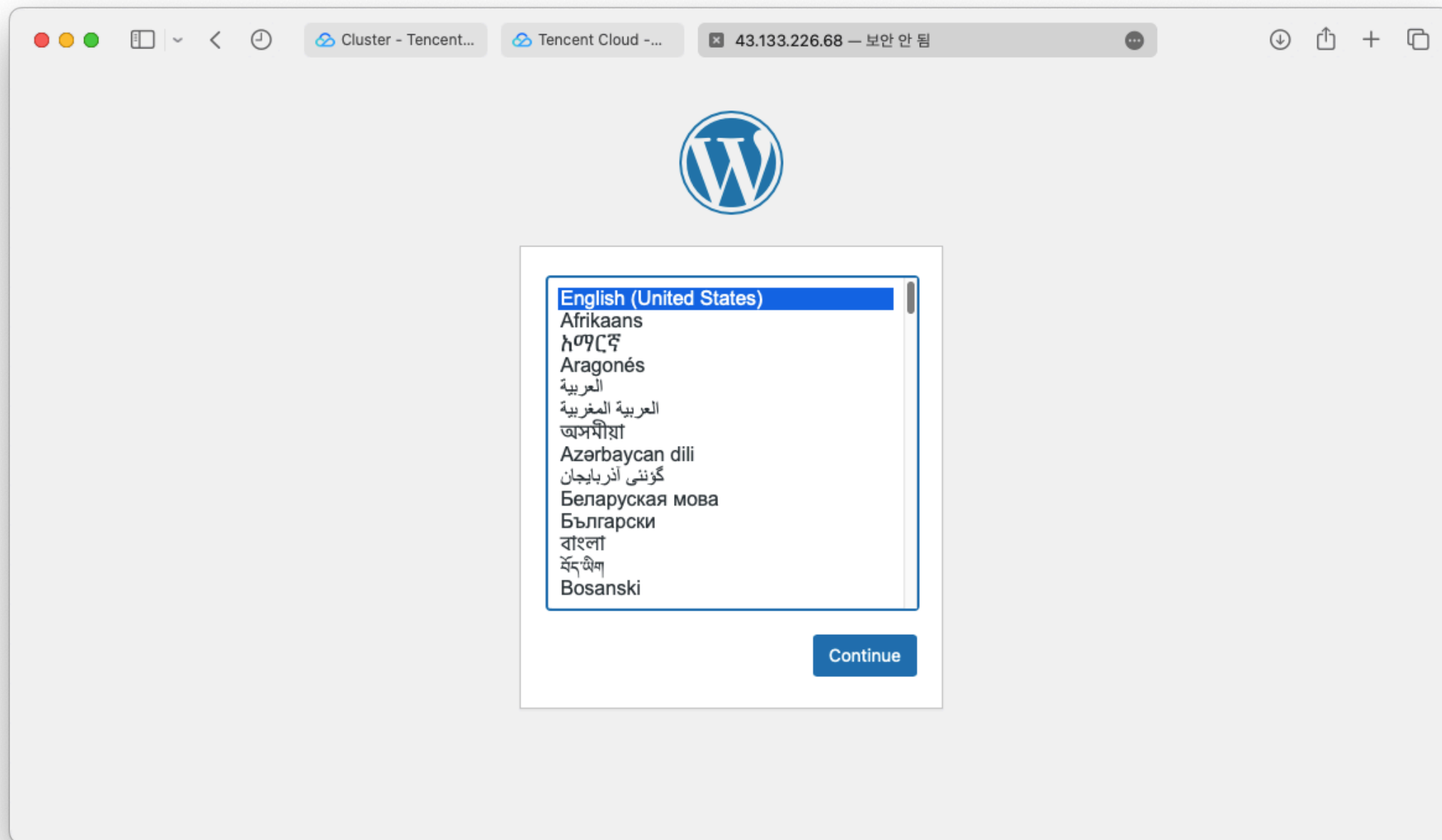
상세 - 결과물 안내

결과물 3) 보고서 pdf 파일

- 간단한 결과 보고서 작성 후 pdf 파일로 제출
- 보고서에는 다음 그림만 포함 (글 작성할 필요 없음)
 1. WordPress 접속 이미지
 2. Tencent Cloud의 자기 클러스터에서 다음 내용들을 캡처한 이미지
 - Workload / Deployment
 - Service and Route / Service
 3. 컴포넌트 간 네트워크 흐름을 그림으로 간단하게 그려보기 (다음은 필수적으로 포함)
 - db service, db deployment, db pod
 - blog service, blog deployment, blog pod
 - 외부 유저

상세 - 결과물 예시

- WordPress 접속 성공 이미지 (외부에서 node_ip:nodeport 로 접속했을 때)



3. 과제 #2

상세 - 결과물 예시

- Workload / Deployment 캡처 (namespace를 assignment2로 설정)

The screenshot shows the Tencent Cloud console interface. The left sidebar contains navigation menus for 'Basic information', 'Node management', 'Namespace', 'Workload' (with sub-items: Deployment, StatefulSet, DaemonSet, Job, CronJob), 'HPA', 'Service and route', and 'Configuration management'. The main panel is titled 'Deployment' and shows a table of deployments in the 'assignment2' namespace. An orange arrow points to the namespace dropdown menu.

Name	Labels	Selector	Number of run...	Request/Limits	Operation
<input type="checkbox"/> blog-deploy	-	app:blog	1/1	CPU: Unlimited / Unlimited MEM: Unlimited / Unlimited	Update Pod quantity Update Pod configuration More ▾
<input type="checkbox"/> db-deploy	-	app:db	1/1	CPU: Unlimited / Unlimited MEM: Unlimited / Unlimited	Update Pod quantity Update Pod configuration More ▾

Page 1 20 / page

상세 - 결과물 예시

- Service and Route / Service 캡처 (namespace를 assignment2로 설정)

The screenshot shows the Tencent Cloud console interface. The left sidebar contains navigation menus for 'Basic information', 'Node management', 'Namespace', 'Workload', 'HPA', 'Service and route', 'Configuration management', and 'Authorization management'. The 'Service and route' menu is expanded, showing 'Service', 'Ingress', and 'NginxIngress'. The main content area is titled 'Service' and shows a 'Create' button and a dropdown menu set to 'assignment2'. Below this is a table listing services:

Name	Labels	Type	Selector	IP address	Time created	Operation
blog-service	-	NodePort	app:blog	172.16.255.30 (Service IP)	2022-11-29 2...	Update configuration Edit YAML Delete
db-service	-	ClusterIP	app:db	172.16.255.89 (Service IP)	2022-11-29 2...	Update configuration Edit YAML Delete

Page 1 20 / page

상세 - 결과물 예시

- 네트워크 흐름도 작성)
 - db service, db deployment, db pod
 - blog service, blog deployment, blog pod
 - 외부 유저
- 외부 유저가 wordpress 웹에 접속 요청을 하면, 그 요청이 어떻게 전달되는지 그리기

힌트

힌트 1) 여러 리소스 내용을 한 yaml 파일에 작성하는 방법

→ --- 로 구분!

```
apiVersion: v1
kind: Namespace
metadata:
  name: assignment2
---
kind: Service
apiVersion: v1
metadata:
  name: nginx-service
  namespace: assignment2
spec:
  selector:
    app: nginx-front
  type: ClusterIP
  ports:
    - port: 8000
      targetPort: 80
```


힌트

힌트 2) WordPress와 MySQL에 필요한 환경 변수들

```
environment:
- MYSQL_ROOT_PASSWORD=somewordpress
- MYSQL_DATABASE=wordpress
- MYSQL_USER=wordpress
- MYSQL_PASSWORD=wordpress
```

MySQL 구성에 필요한 환경 변수들 (실습#4에서 발췌)

```
environment:
- WORDPRESS_DB_HOST=db
- WORDPRESS_DB_USER=wordpress
- WORDPRESS_DB_PASSWORD=wordpress
- WORDPRESS_DB_NAME=wordpress
```

WordPress 구성에 필요한 환경 변수들 (실습#4에서 발췌)

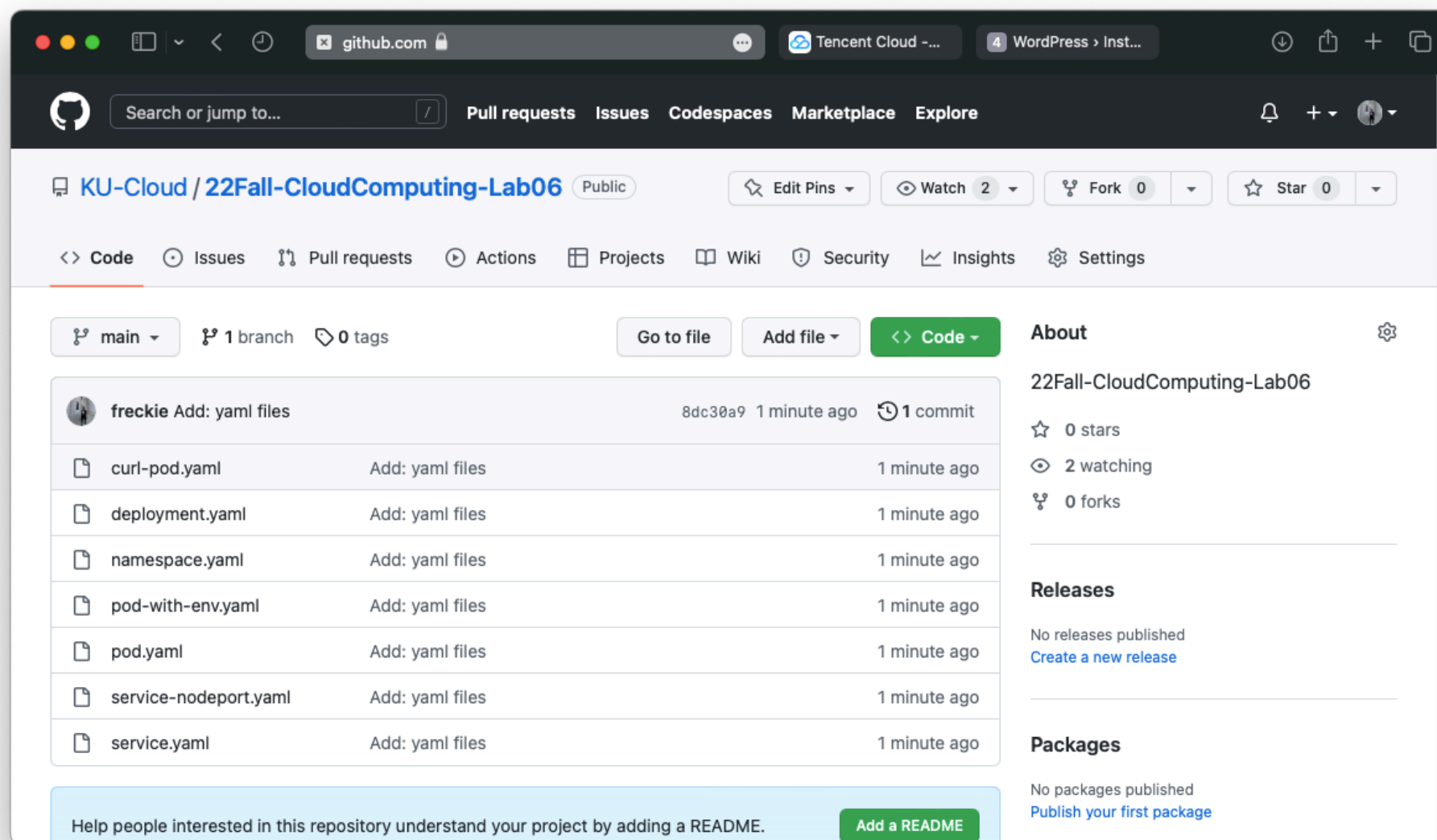
```
# pod-with-env.yaml
apiVersion: v1
kind: Pod
metadata:
  name: db
  namespace: lab06
spec:
  containers:
  - name: mysql
    image: mysql:5.7
    env:
    - name: MYSQL_USER
      value: wordpress
    - name: MYSQL_DATABASE
      value: wordpress
    - name: MYSQL_PASSWORD
      value: wordpress
    - name: MYSQL_ROOT_PASSWORD
      value: wordpress
    ports:
    - containerPort: 3306
```

Pod에 환경 변수 입력하는 방법

“ 추가자료 ”

실습 자료에 포함된 yaml 파일 공유

→ <https://github.com/KU-Cloud/22Fall-CloudComputing-Lab06>



Tencent Kubernetes Engine 클러스터의 SecurityGroup 설정

- 이번 실습에서는 다양한 포트를 사용하기 때문에, 외부에서도 해당 클러스터에 접근할 수 있도록 보안 정책의 완화가 필요함.
- 클러스터의 SecurityGroup을 미리 생성해둔 ALL_OPENED로 설정하면, 모든 포트에 대해 외부에서 접근이 가능함.

The screenshot displays the Tencent Cloud console interface for a Kubernetes Engine cluster. The left sidebar shows the navigation menu with 'Cluster' selected. The main content area is divided into two sections: 'Basic information' and 'Cluster API Server information'. The 'Basic information' section includes details like runtime components (containerd), cluster description (N/A), Tencent Cloud tags, deletion protection (Disabled), and time created (2022-11-29 15:40:23). The 'Cluster API Server information' section shows internet access (Enabled), security group (sg-94qul15f), access IP (43.133.89.46), kubeconfig, and private network access (Disabled). An orange arrow points to the 'Basic information' tab, and an orange callout points to the 'Edit' button for the security group.

Tencent Cloud Overview Products + Ticket 99+ Billing Center English

Tencent Kubernetes Engine

- Overview
- Cluster**
- Service Mesh
- Application
- Helm
- Images
- Ops
- Cluster Ops
- TMP
- Log Collection(Legacy)

Basic information

- Node management
- Namespace
- Workload
- HPA
- Service and route
- Configuration management
- Authorization management
- Storage
- Add-on management
- Log

Runtime components ⓘ containerd

Cluster description N/A

Tencent Cloud tags ...

Deletion Protection ⓘ Disabled

Time created 2022-11-29 15:40:23

VPC-CNI mode Disabled

Service CIDR block 172.16.252.0/22

Kube-proxy proxy mode iptables

Cluster API Server information

Internet access Enabled

Security group: sg-94qul15f Edit

Access IP 43.133.89.46 Copy

KubeConfig ap Copy Download

Private network access Disabled

Kubeconfig permission management

Connecting to Kubernetes cluster through Kubectl:

The screenshot shows the Tencent Cloud console interface. A modal dialog titled "Modify security group" is open. It contains a blue information box with the following text: "Note that the cluster apiserver will be exposed to the public network if Public Network Access is enabled. You need to configure access control policies via security group, which will be bound to the CLB used as the public network entry." Below this, the "Original security group" is listed as "sg-94qul15f". The "New security group" dropdown menu is open, showing a list of options: "sg-94qul15f | ALL_OPENED", "ALL_OPENED", and "sg-94qul15f | ALL...". An orange arrow points to the "ALL_OPENED" option in the dropdown. Another orange arrow points to the "Confirm" button at the bottom of the dialog. The background shows the Tencent Cloud console with the "Tencent Kubernetes Engine" section selected.

kubectl 기초 명령 정리

kubectl get {resource_type} {resource_name(optional)}

→ resource_type에 해당하는 리소스들을 조회하여 모두 출력

ex) `kubectl get pods` (Pod들 조회)

ex) `kubectl get pod nginx` (nginx라는 이름의 Pod 조회)

ex) `kubectl get services -n lab06` (lab06 네임스페이스에 속한 Service들 조회)

ex) `kubectl get services -A` (모든 네임스페이스에 있는 Service들 조회)

kubectl 기초 명령 정리

kubectl describe {resource_type} {resource_name}

→ resource_type에 해당하며 resource_name 이름을 가진 리소스에 대한 상세 정보 출력

ex) **kubectl describe pod nginx** (nginx라는 이름의 Pod 상세 정보 조회)

ex) **kubectl describe service nginx-svc -n lab06** (lab06 네임스페이스에 속한 nginx-svc 라는 이름의 Service 상세 정보 조회)

kubectl 기초 명령 정리

kubectl apply -f {filename}

→ filename(주로 yaml) 파일을 쿠버네티스 컨텍스트에 전달

ex) `kubectl apply -f deployments.yaml` (deployments.yaml 파일에 서술된 리소스를 생성)

kubectl delete -f {filename}

→ `kubectl apply -f` 로 생성했던 리소스를 삭제

ex) `kubectl delete -f deployments.yaml`

Q & A