

## 5.2 클러스터 외부에 있는 서비스 연결

🕒 생성일	@2021년 4월 2일 오후 12:53
🏷️ 태그	

쿠버네티스 서비스 기능으로 외부 서비스를 노출하려는 경우?!

- 서비스 로드밸런싱
- 서비스 검색

모두 활용해서 내부 서비스에 연결하는 것처럼 외부서비스에 연결 가능

### 1. 서비스 엔드포인트 소개

서비스는 파드에 직접 링크(연결) 되지 않는다

서비스 - 엔드포인트 리소스 - 파드

이렇게 엔드포인트가 존재한다

```
kubectl describe svc kuba
```

```
Commons terminated with exit code 130
[✖ jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice] kubectl describe svc kuba
Name:          kuba
Namespace:     default
Labels:        <none>
Annotations:   <none>
Selector:      app=kuba
Type:          ClusterIP
IP:            10.79.244.58
Port:          <unset> 80/TCP
TargetPort:    8080/TCP
Endpoints:     10.76.0.10:8080,10.76.1.19:8080,10.76.2.31:8080
Session Affinity: None
Events:        <none>
```

여기에서 `Selector: app=kuba` 가 엔드포인트 목록을 만드는 데 사용된다

`Endpoints: 10.76. ....` 가 서비스의 엔드포인트를 나타내는 파드IP와 Port 목록

## 💡 엔드포인트 리소스

: 서비스로 노출되는 파드의 IP 주소, Port 목록

엔드포인트 리소스는 다른 쿠버네티스 리소스와 유사해서 → `kubectl get` 사용함

```
kubectl get endpoints kuba
```

```
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl get endpoints kuba
NAME      ENDPOINTS                                     AGE
kuba      10.76.0.10:8080,10.76.1.19:8080,10.76.2.31:8080 115m
```

- 파드 셀렉터는 서비스 스펙에 정의돼 있지만 들어오는 연결을 전달할때 직접 사용하지 않음
  - IP와 포트 목록을 작성하는 데 사용됨
  - 엔드포인트 리소스에 저장됨
- 서비스 프록시는 이 목록중 하나의 IP를 선택함 → 들어온 연결을 대상 파드의 수신 대기 서버로 전달

## 2. 서비스 엔드포인트 수동 구성

서비스 엔드포인트 ↔ 서비스 : 분리시키면 수동으로 엔드포인트 구성하고 업데이트

- 쿠버네티스는 파드 셀렉터가 없으면 서비스에 포함된 파드가 무엇인지 알 수 없다.
- 파드 셀렉터 없이 만든다면, 엔드포인트 목록을 엔드포인트 리소스로 설정하고 **서비스와 엔드포인트 리소스를 모두 만들어야함**

### 셀렉터 없이 서비스 생성

파드 셀렉터가 없는 서비스를 만드는 yaml.. `external-service.yaml`

```
apiVersion: v1
kind: Service
metadata:
  name: external-service #서비스 이름은 엔드포인트 오브젝트 이름과 일치되어야함
spec: #서비스에 셀렉터 미정의
  ports:
    - port: 80
```

- 포트 80으로 들어오는 연결을 허용하는 서비스 external-service 임

- 서비스에 대한 파드 셀렉터 정의하지 않음

## 셀렉터가 없는 서비스에 관한 엔드포인트 리소스 생성

endpoints 리소스 : 서비스 속성이 아닌 별도의 리소스

셀렉터가 없기 때문에 수동으로 생성해줘야함 `external-service-endpoints.yaml`

```
apiVersion: v1
kind: Endpoints
metadata:
  name: external-service #아까 서비스 생성 yaml에서 서비스 이름과 동일
subsets:
  - addresses: #서비스가 연결을 전달할 endpoint의 IP
    - ip: 11.11.11.11
    - ip: 22.22.22.22
    ports:
      - port: 80 #엔드포인트 대상포트
```

- 엔드포인트 오브젝트 이름 == 서비스 이름
- endpoint IP 주소 모두 기록 + 대상 포트
- 목록에 대한 모든 IP에 대해 서비스 엔드포인트간 로드밸런싱함.
- 서비스 만들어진 후 컨테이너에 서비스의 환경변수가 포함된다
- 나중에 외부서비스를 쿠버네티스 내 파드로 `마이그레이션` 한 경우 `셀렉터 추가` 가능
  - 그러면 엔드포인트 `자동` 으로 관리하는것 가능
  - 벗... 셀렉터 제거시 바로 k8s는 엔드포인트 업데이트 멈춤.. 자닌한놈

## 3. 외부 서비스를 위한 별칭 생성

위의 방법보다 좀 더 간단하게.. FQDN(Fully Qualified Domain Name)으로 외부 서비스를 참조할 수 있음

### ExternalName 서비스 생성

외부 서비스의 별칭으로 사용되는 서비스를 만들려면..

spec의 `type` 필드를 `ExternalName` 으로 설정해서 서비스 리소스 생성하면됨

`api.somecompany.com`에 공개 API가 있다고 가정하는 `external-service-externalname.yaml` 생성

```
apiVersion: v1
kind: Service
metadata:
  name: external-service
spec:
  type: ExternalName #서비스 유형이 이렇게 설정됨
  externalName: someapi.somecompany.com #실제 서비스의 정규화된 도메인 이름
  ports:
    - port: 80
```

- 서비스가 생성되면 서비스의 FQDN 말고 `external-service.default.svc.cluster.local` 도메인 이름 (또는 `external-service`)로 외부 서비스에 연결가능
- 서비스를 사용하는 파드에서 실제 서비스 이름/위치가 숨겨짐
  - 나중에 `ExternalName` 속성 변경하거나 `ClusterIP`로 다시 유형 변경하고 서비스 스펙 생성 → 서비스 스펙 수정시 나중에 다른 서비스 가리킬 수 있음
  - (사실 이부분 이해 잘 안감.. 몬말이지>?!?!?)
- `ExternalName` 서비스는 DNS 레벨에서만 구현된다 → 서비스 프록시를 완전히 무시하고 바로 외부서비스에 직접 연결되어서.. `ClusterIP`를 얻지 못함