5.3 외부 클라이언트에 서비스 노출

생성일	@2021년 4월 2일 오후 12:53
태그	

웹 서버와 같은 특정 서비스를 외부에 노출해 외부 클라이언트가 액세스 할 수 있게 노출 시키려면...

1. 노드 포트(NodePort)로 서비스 유형 설정
2. 로드밸런서로 서비스유형 설정

   : 노드포트 유형의 확장이 로드밸런서
3. 단일 IP 주소로 여러 서비스를 노출하는 인그레스 리소스 만들기

   : HTTP 레벨(네트워크 7계층)에서 작동하므로 4계층 서비스보다 더 많은 기능 제공

1. 노드포트 서비스 사용

서비스 생성 후 유형을 노드포트로 설정 → 노드포트 서비스가 만들어진다

1. k8s는 모든 노드에 특정 포트를 할당하고 서비스를 구성하는 파드로 들어오는 연결을 전달
2. 서비스의 내부 클러스터IP 뿐만 아니라 `모든 노드의 IP`와 할당된 `노드포트`로 서비스에 액세스 가능

노드포트 서비스 생성

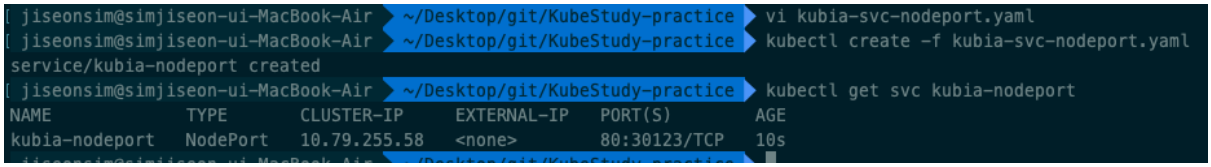
kubia-svc-nodeport.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: kubia-nodeport
spec:
  type: NodePort #서비스 유형이 노드포트
  ports:
  - port: 80 # 서비스 내부 클러스터IP의 포트
    targetPort: 8080 # 서비스 대상 파드의 포트
    nodePort: 30123 # 각 클러스터 노드의 포트 30123으로 서비스에 액세스 가능
  selector:
    app: kubia
```

- 노드포트 지정 안할시 k8s가 임의의 포트를 선택

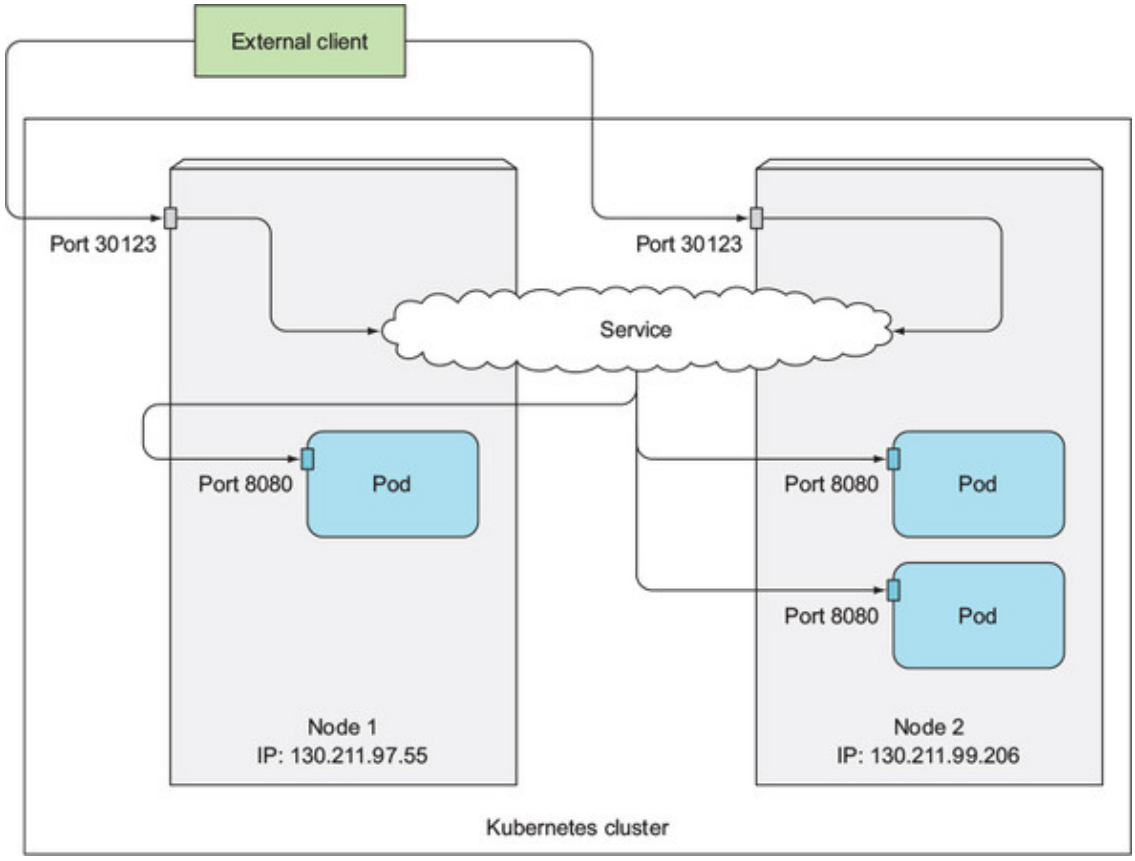
노드포트 서비스 확인

```
kubectl get svc kubia-nodeport
```



- 클러스터IP 10.79.255.58로 서비스에 액세스 가능
- PORTS : 클러스터 IP 내부 포트(80), 노드포트(30213) 표시

이 서비스에 접속할 수 있는 주소 목록
- 10.79.255.58:80
- 첫번째노드의 IP: 30123
- 두번째노드의 IP: 30123



외부 클라이언트가 노드포트 서비스에 액세스할 수 있도록 방화벽 규칙 변경

해당 노드포트로 서비스에 액세스하기 위해, 외부연결을 허용하도록 GCP 방화벽 구성

```
gcloud compute firewall-rules create kubia-svc-rule --allow=tcp:30123
```

```
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice gcloud compute firewall-rules create kubia-svc-rule --allow=tcp:30123
Creating firewall...⠹Created [https://www.googleapis.com/compute/v1/projects/kubestudy-kubia-307318/global/firewalls/kubia-svc-rule].
Creating firewall...done.
NAME           NETWORK  DIRECTION  PRIORITY  ALLOW     DENY  DISABLED
kubia-svc-rule  default  INGRESS    1000      tcp:30123       False
```

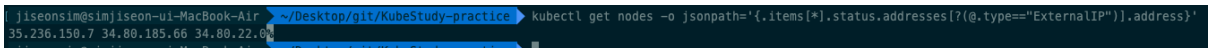
노드IP와 포트 30123으로 서비스에 액세스 가능

노드IP 확인하기 : JSONPath를 지정해 원하는 정보만 출력하도록 지시

```
kubectl get nodes -o jsonpath='{.items[*].status.addresses[?(@.type=="ExternalIP")].address}'
35.236.150.7 34.80.185.66 34.80.22.0
```

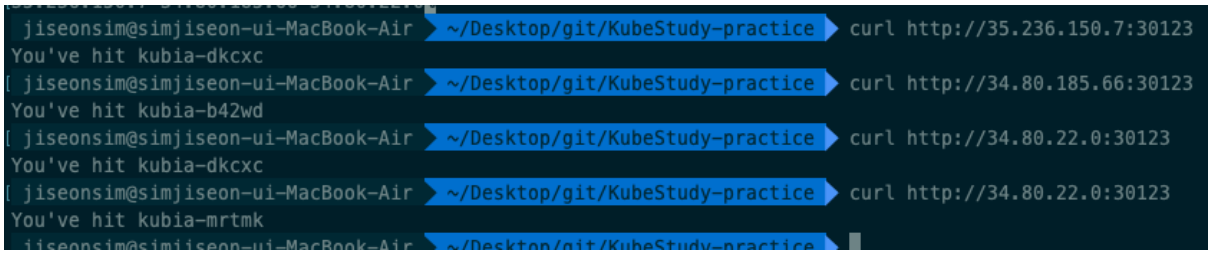
- `.items[*]` : items 소속의 모든 항목 조회

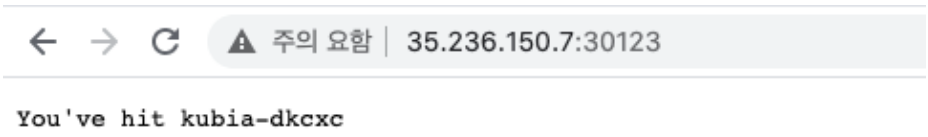
- `.status` : 각 항목의 status 속성 조회
- `.addresses[ 내용 ]` : type 속성이 ExternalIP로 설정된 항목으로 필터링



이제 노드의IP를 가지고 서비스에 액세스 해보기

```
curl http://35.236.150.7:30123
curl http://34.80.185.66:30123
curl http://34.80.22.0:30123
```





인터넷에서 어떤노드든 30123으로 파드에 액세스 가능

- 클라이언트가 요청을 보내는 노드는 중요하지 않음
- 하지만, 한 노드에만 요청하는데 장애가 난 경우 ← 클라이언트는 서비스에 액세스 불가능

> 오프라인인 노드로 요청을 보내지 않고, 모든 노드에 요청을 분산 시킬 수 있는 로드밸런서를 배치하는 것이 좋다

2. 외부 로드밸런서로 서비스 노출

1. 클라우드 공급자에서 실행되는 쿠버네티스 클러스터의 경우

→ 클라우드 인프라에서 로드밸런서를 ***자동으로 프로비저닝***하는 기능 제공

- 노드포트 대신 `로드밸런서 서비스 유형`으로 설정하면 된다
- 로드밸런서 : 공개적으로 액세스 가능한 고유IP 가짐

2. 쿠버네티스가 로드밸런서 서비스 지원하지 않는 환경에서 실행 중인 경우

→ 로드밸런서는 프로비저닝 되지 않음

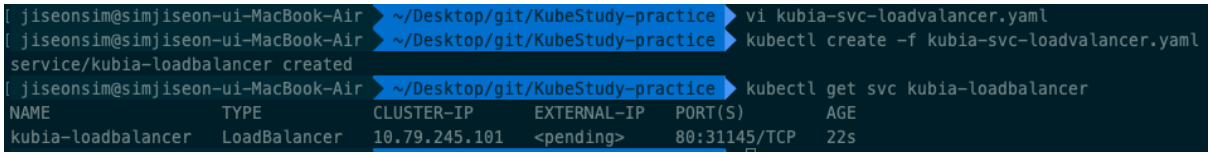
- 서비스는 노드포트 서비스처럼 작동
- 로드밸런서 서비스는 노드포트 서비스의 확장이기 때문

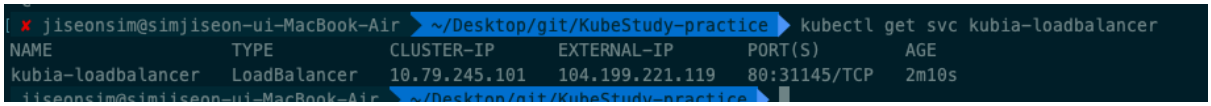
로드밸런서 서비스 생성

kubia-svc-loadvalancer.yaml

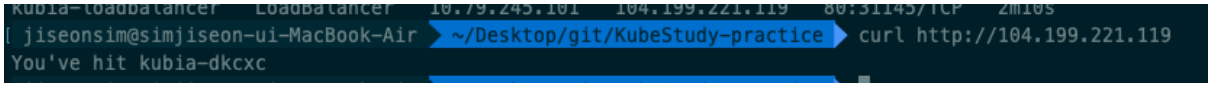
```
apiVersion: v1
kind: Service
metadata:
  name: kubia-loadbalancer
spec:
  type: LoadBalancer
  ports:
  - port: 80
    targetPort: 8080
  selector:
    app: kubia
```

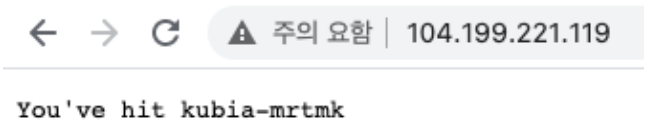
```
kubectl get svc kubia-loadbalancer
```



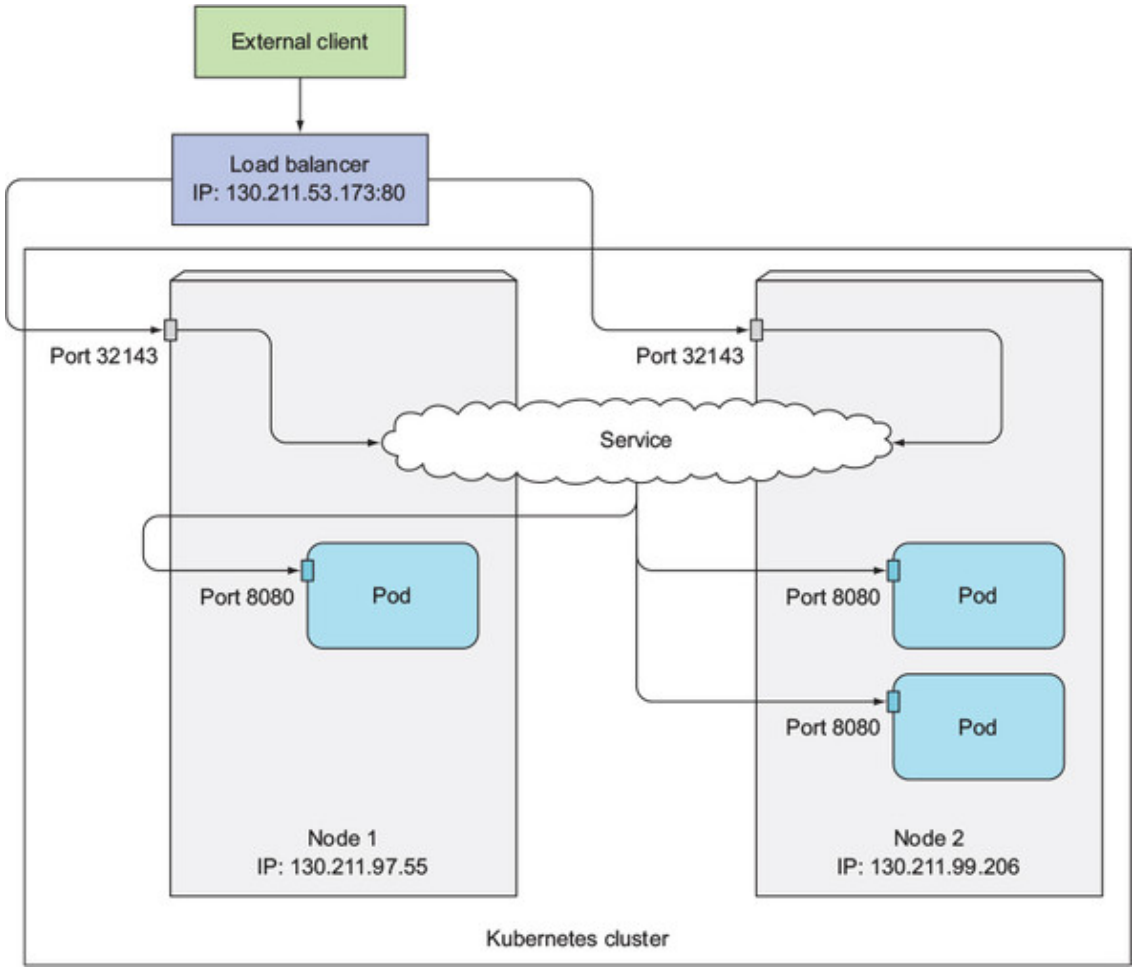


외부 아이피인 `104.199.221.119` 로 액세스 해보자





- 노드포트와 달리 방화벽이 필요없었다



- 외부 클라이언트는 로드밸런서의 80포트에 연결
- 노드에 암묵적으로 할당된 노드포트 32143로 라우팅한다 (나의 경우 31145였다)
- 파드 인스턴스로 전달

3. 외부 연결의 특성 이해

외부에서 서비스로 들어오는 연결에 주의해야할 것들

불필요한 네트워크 홉의 이해와 예방

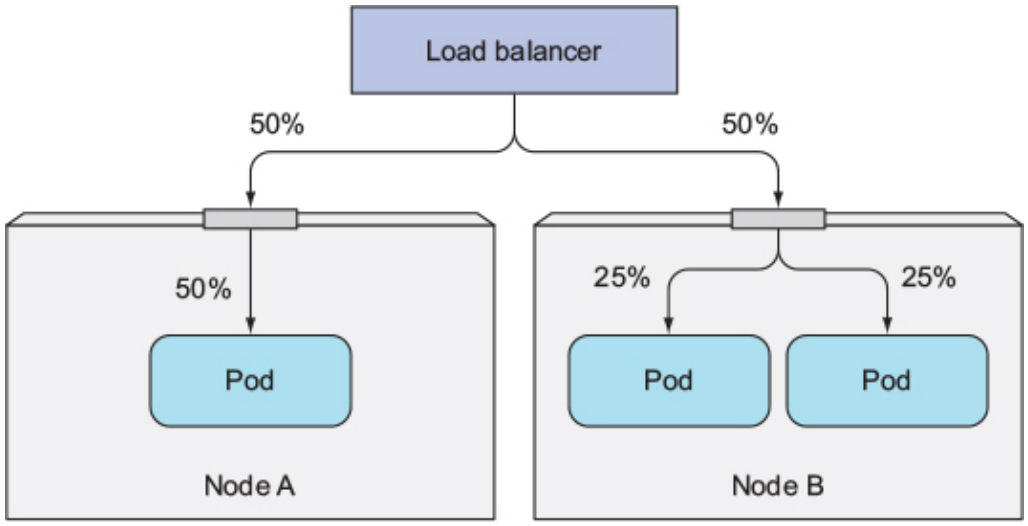
네트워크홉 : nw에서 출발지와 목적지 사이에 위치한 경로의 한 부분을 의미

외부 클라이언트가 노드포트/로드밸런서로 서비스에 접속한다
→ 파드에 도달하려면 추가적인 네트워크 홉이 필요할 수도 있음 (긍정적인 경우는 아님)

 `externalTrafficPolicy` 필드 설정

- 외부의 연결을 수신한 노드에서 ***실행 중인 파드***로만 외부 트래픽을 전달하도록 서비스를 구성
- 추가 홉을 방지할 수 있다.

```
spec:
  externalTrafficPolicy: Local
  ...
```



단점: 서비스의 노드포트로 외부 연결이 열린 경우 서비스 프록시는 로컬에 실행 중인 파드를 선택하는데, 이때 로컬 파드가 존재 하지 않으면 연결이 중단됨

→ 로드 밸런서는, 로컬파드가 하나 이상 있는 노에만 연결을 전달해야함.

→ 균등하게 모든 파드에 연결이 분산되지 않는다.

클라이언트 IP가 보존되지 않음 인식.

노드포트로 연결 수신시, 패킷에서 소스네트워크 주소변환(SNAT)이 수행됨 → 패킷의 소스 IP(클라이언트IP)가 변경됨

- 웹서버의 경우 액세스 로그에 브라우저IP 표시 불가
- 로컬 외부 트래픽 정책은 연결을 수신하는 노드 ↔ 파드 호스팅하는 노드 사이에 추가 홉이 없어서 클라이언트 IP 보존 가능(SNAT 수행X)