# 5.1 서비스 소개

| 생성일 | @2021년 4월 2일 오후 12:53 |
| --- | --- |
| 태그 | |

MSA환경에서 앱 대부분은 :
- 클러스터 외부 클라이언트의 HTTP 요청
- 클러스터 내부의 다른 파드의 HTTP 요청
의 응답하며 작업 수행

> 파드가 다른 파드에게 제공하는 서비스를 사용할 시
> → 다른 파드를 찾는 방법이 필요

- 비 k8s환경

  : 시스템 관리자가 클라이언트 구성 파일에 서비스를 제공하는 서버에 각 클라이언트 앱 구성을 정확한 `IP` / `호스트네임`을 지정해서 진행 (옛 직장에서 이 방법으로 했다. 중앙 관리/모니터링 서버 - 앱서버)
- k8s 환경에서 위와 같이 할시... 동작하지 않는 이유
  1. 파드=일시적, 파드는 노드간 이동가능하고, 수가 변동적이다
  2. 파드의 IP 주소를 미리 알 수 없다 (노드에 파드가 스케줄 되는 바로 직전에 IP가 할당되므로)
  3. 수평 스케일링으로 여러 파드가 동일한 서비스 제공 → 각 파드가 다른 IP 주소를 갖고 있음 (모든 파드를 단일 IP 주소로 접근해야함)

## 예제로 `서비스` 설명

프론트엔드 웹서버와 백엔드 데이터베이스 서버가 있다고 가정

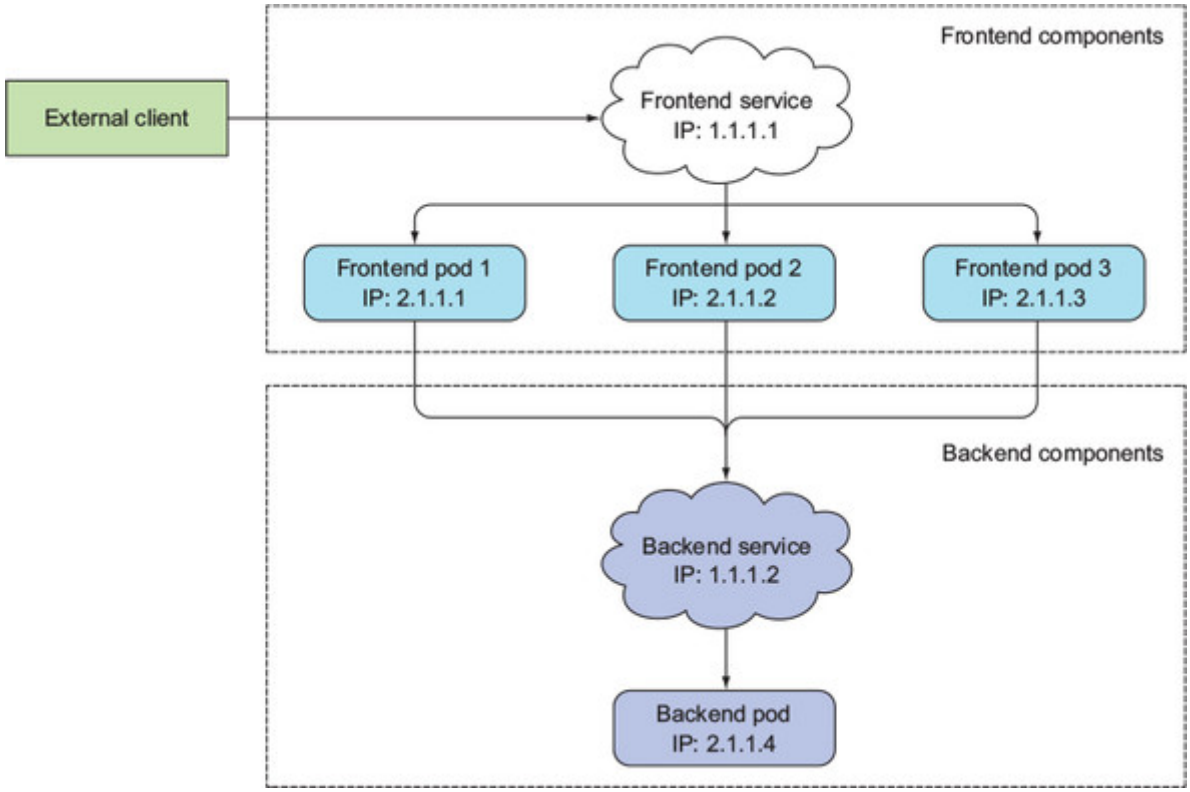
- 프론트 엔드 파드 여러개
- 백엔드 파드 한개

여기서 시스템을 기동할 시?

- 웹서버의 수에 상관없이 외부 클라이언트→ 프론트엔드 파드에 연결 가능 해야
- 프론트엔드 파드 → 백엔드 DB와 연결해야함 : 백엔드 DB파드가 클러스터 내에서 이동할때, 프론트엔드 파드 재설정하는것 피하자

💡 `서비스`로 해결하자

기본 목적 : 파드 그룹을 클러스터의 다른 파드에 노출 + 클러스터 외부로 노출




- **프론트엔드 서비스** → 프론트엔드 파드 1,2,3

  서비스로, 외부 클라가 파드에 연결할 수 있는 한 고정 IP 주소가 노출됨
- 프론트엔드 파드 1,2,3 → **백엔드 서비스** → 백엔드 파드

***장점***

- 파드가 변경되어도 서비스 IP는 고정이기 때문에 안정적
- 프론트엔드 파드에서 백엔드 서비스를 `DNS` / `환경변수 이름` 으로 쉽게 찾는것이 가능
- 로드밸런싱

# 1. 서비스 생성

서비스를 지원하는 파드가 다수라면, 어떤 파드가 서비스의 일부분인지 아닌지 구분 하는 방법?!

→ 레이블 셀렉터로 서비스-파드 그룹화 가능!

## kubectl expose로 서비스 생성

서비스를 생성하는 가장 쉬운 방법으로, 2장에서 RC를 노출했었다

```
kubectl expose rc kubia --type = LoadBalancer --name kubia-http
```

요렇게 했었었다 ㅎㅎ

## YAML로 서비스 수동 생성

`kubia-svc.yaml` 만들자

```
apiVersion: v1
kind: Service
metadata:
  name: kubia
spec:
  ports:
  - port: 80 # 서비스가 사용할 포트
    targetPort: 8080 #서비스가 포워드할 컨테이너 포트
  selector: # app=kubia 레이블을 가진 모든 파드가 이 서비스에 포함
    app: kubia
```

`app=kubia` 레이블 셀렉터와 일치하는 파드의 `포트 8080`로 라우팅

## 새 서비스 검사하기

```
kubectl get svc
```

```
( jiseonsim@simjiseon-ui-MacBook-Air ) ~/Desktop/git/KubeStudy-practice ) kubectl create -f kubia-svc.yaml
service/kubia created
( jiseonsim@simjiseon-ui-MacBook-Air ) ~/Desktop/git/KubeStudy-practice ) kubectl get svc
NAME                TYPE           CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes          ClusterIP      10.79.240.1     <none>           443/TCP          21d
kubia               ClusterIP      10.79.244.58    <none>           80/TCP           9s
kubia-deploy-http   LoadBalancer   10.79.247.39    35.194.216.183   8080:32689/TCP   20d
kubia-http          LoadBalancer   10.79.246.212   35.229.167.82    8080:31622/TCP   21d
```

네임스페이스의 모든 서비스 리소스 조회 → cluster IP 할당 확인

- 서비스에 할당된 IP 주소 : 10.79.244.58
- 클러스터 내부에서만 액세스 가능

## 클러스터 내에서 서비스 테스트

1. 서비스 클러스터 IP로 요청 보내고, 응답을 로그로 남겨서 파드 로그 검사
2. 노드에 ssh 접속 후 curl 실행
3. `kubectl exec`으로 기존 파드에서 curl 실행

여기서 3번을 해보자

## 실행 중인 컨테이너에 원격으로 명령어 실행

kubectl get pods에서 파드 하나 선택해서 exec 실행해보자.

(나의 경우)

```
kubectl exec kubia-blq2z -- curl -s http://10.79.244.58
```

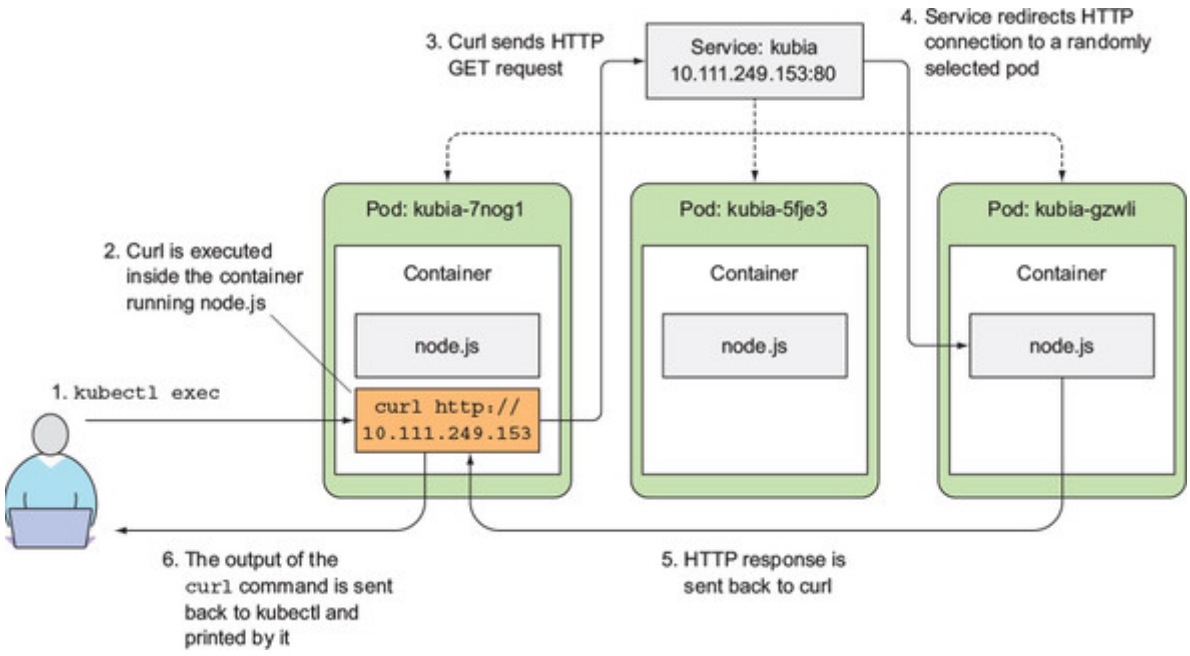
```
( ✘ jiseonsim@simjiseon-ui-MacBook-Air ) ~/Desktop/git/KubeStudy-practice ) kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
kubia                             1/1     Running   0          14m
kubia-blq2z                       1/1     Running   0          2m28s
kubia-deploy-7f5b6bd7d4-frqrg     1/1     Running   0          12m
kubia-deploy-7f5b6bd7d4-rm4pd     1/1     Running   0          12m
kubia-deploy-7f5b6bd7d4-zx2wh     1/1     Running   0          12m
kubia-lphfn                       1/1     Running   0          2m28s
kubia-vgzws                       1/1     Running   0          2m28s
( jiseonsim@simjiseon-ui-MacBook-Air ) ~/Desktop/git/KubeStudy-practice ) kubectl exec kubia-blq2z -- curl -s http://10.79.244.58
You've hit kubia-lphfn
```

쨘....! 잘 되는 것을 확인할 수 있다.

이미 app=kubia 파드를 지웠다면... 4.2장의 `kubia-rc.yaml`을 다시 create해보는 것을 추천

- 더블 대시 `--` 는 kubectl 명령줄 옵션의 끝을 의미 (이 후의 명령은 파드 내에서 실행되어야하는 명령)
- 더블 대시 사용안하면 `-s`가 `kubectl exec` 의 옵션인 것으로 간주하고 오류남

## kubectl exec으로 curl한 명령의 원리




1. exec 명령
2. curl은 실행
3. curl이 HTTP요청을 서비스 IP http://10.79.244.58 로 전송
4. 위 IP에는 3개 파드가 연결되어있음 → ***쿠버네티스 프록시***가 연결을 가로채서 그 중 임의의 파드로 요청 전달
5. 해당 파드내에서 실행중인 node.js가 요청 처리해서 응답
6. curl이 표준출력으로 응답 출력

## 서비스의 세션 어피니티 구성

***서비스 프록시***가 임의의 파드를 선택해서 `포워딩` 함

→ 요청시마다 다른 파드가 선택됨

> 특정 클라이언트의 모든 요청을 매번 같은 파드로 리디렉션하려면?

💡 서비스의 **세션 어피니티**(SessionAffinity) 속성을 기본값 None 대신 `ClientIP`로 설정

아까 `kubia-svc.yaml` 같은 설정에서 spec에 속성 추가한다

```
apiVersion: v1
kind: Service
spec:
  sessionAffinity: ClientIP
```

이렇게 하면 클라이언트 IP의 모든 요청을 동일파드로 전달함..

## 동일한 서비스에서 여러 개의 포트 노출

서비스는 여러 포트를 지원한다.

- 포트 80, 443 를 파드의 포트 8080과 8443으로 전달할 수 있다
- 이 경우 굳이 두 개의 서비스를 만들 필요가 없다
- 하나의 서비스를 사용해 멀티 포트 서비스를 사용하면 ***단일 클러스터 IP로 모든 서비스 포트가 노출됨***

`kubia-svc-named-ports.yaml`

```
apiVersion: v1
 kind: Service
metadata:
  name: kubia
spec:
  ports:
  - name: http
    port: 80 #서비스포트80 - 파드포트8080
    targetPort: 8080
  - name: https
    port: 443 #서비스포트443 - 파드포트8443
    targetPort: 8443
  selector:
    app: kubia #레이블 셀렉터는 항상 모든 서비스에 적용된다
```

## 이름이 지정된 포트 사용

각 파드의 포트에 이름을 지정하고, 서비스 스펙에서 이름으로 참조할 수 있다

- 8080은 http, 8443은 https로 이름 명명

예제 yaml은 위와 같다.... `- name :` 으로 관리

이렇게 포트 이름을 지정한다면
나중에 서비스 스펙을 변경하지 않고도 포트 번호 변경 가능

## 2. 서비스 검색

서비스의 IP로 클라이언트 파드 액세스 가능

> 클라이언트 파드가 서비스 IP 주소로 액세스 하려면?

쿠버네티스는 클라이언트 파드가 서비스IP 검색할 수 있는 방법 제공

### 환경변수를 통한 서비스 검색

클라이언트 파드가 생성되기 전에, 서비스를 먼저 생성해서 환경변수 설정 가능

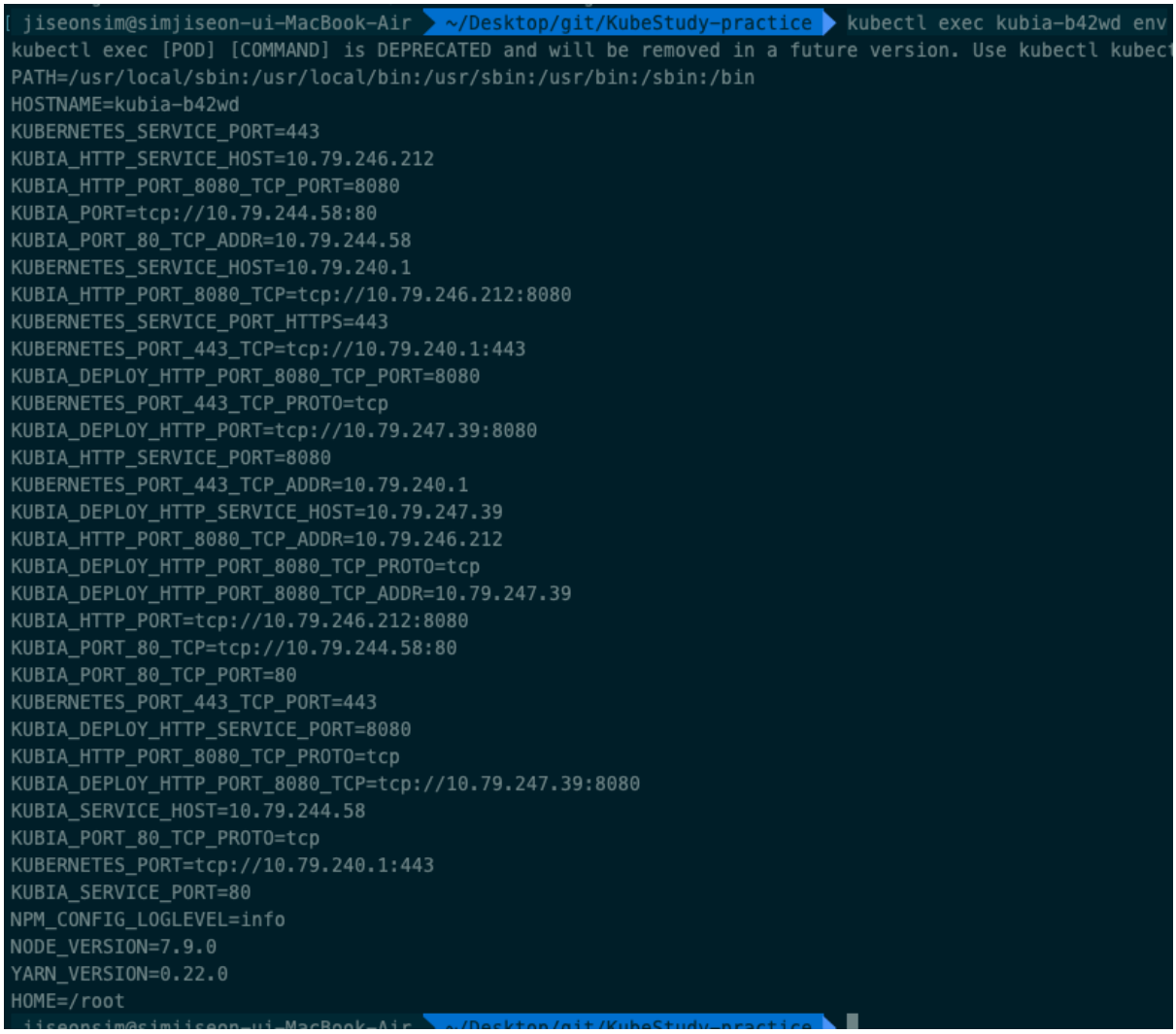
→ 해당 파드의 프로세스는 환경변수를 검사해 서비스의 IP 주소와 포트를 얻을 수 있다

그래서 먼저,, 일단 파드를 다 삭제하고 RC에서 새로 파드를 만들어야한다

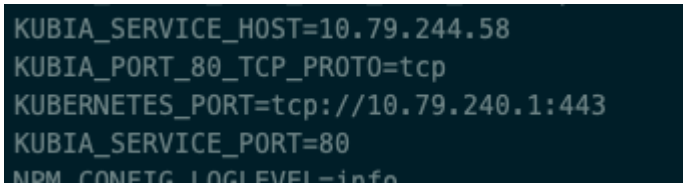
```
kubectl delete po --all
```

하고.. 새로 생성된 파드에서 검색해보자

```
kubectl exec kubia-b42wd env
```




여기서 좀 아래쪽에 보면




이렇게 있는데,

`KUBIA_SERVICE_HOST`, `KUBIA_SERVICE_PORT`가 각각 서비스의 **클러스터 IP**와 서비스가 제공되는 **포트**다.

이렇게 확인할 수 있다...

만약 프론트-백엔드 예제에서 처럼 백엔드 DB서버 파드를 `backend-database`라는 서비스로 노출했다고 하면,

프론트엔드 파드에서 환경변수
- BACKEND_DATABASE_SERVICE_HOST와
- BACKEND_DATABASE_SERVICE_PORT로 확인가능

하지만 환경변수보다 **DNS의 도메인**으로 검색하는것이 더 일반적...!

## DNS를 통한 서비스 검색

3장에서 진행한 `kube-system` 네임스페이스에서 `kube-dns` 조회했었음

→ 이 파드는 DNS서버를 실행하며, 클러스터에서 실행 중인 다른 모든 파드는 자동으로 이를 사용함 (서비스의 내부 DNS)

- 각 파드 스펙의 dnsPolicy 속성으로 컨트롤 가능
- 각 서비스는 내부 DNS 서버에서 DNS항목을 가져옴
- 클라이언트 파드는 서비스 이름을 알고 있으므로 환경변수 대신 FQDN(정규화된 도메인 이름)으로 액세스

## FQDN을 통한 서비스 연결

프론트-백엔드 예제에서 프론트엔드 파드는

**backend-database.default.svc.cluster.local**

FQDN 으로 백엔드 데이터베이스 서비스에 연결 가능

- backend-database = 서비스 이름
- default = 서비스가 정의된 네임스페이스 *(생략가능)*
- svc.cluster.local = 모든 클러스터의 로컬 서비스 이름에 사용되는 클러스터 도메인 접미사 *(생략가능)*

이번에 bash shell을 실행해서 컨테이너에 접근 후, FDQN으로 kubia 서비스에 연결해보자

## 파드의 컨테이너 내에서 bash shell 실행

```
kubectl exec -it kubia-b42wd bash
```

이라고 되어있지만,, 경고문이 뜬다.. 곧 deprecated된다고..

```
kubectl exec -it kubia-b42wd -- bash
```

아까 언급된 더블대시를 쓰자


여기서 kubia 서비스에 액세스 하기

```
curl http://kubia.default.svc.cluster.local
curl http://kubia.default
curl http://kubia
```


이게 다 파드 컨테이너 내부의 DNS resolver가 구성되어있어서 이다.... 개쩐당

멋있어요 k8s!

얘를 확인해보면...

```
cat /etc/resolv.conf
```


줄줄이 컨피규레이션을 볼 수 있당

## 서비스 IP에 핑을 할 수 없는 이유

서비스의 클러스터 IP = 가상 IP 이기 때문에..

ping은 서비스 포트와 결합된 경우에만 의미가 있다

(This problem will be continued in chapter11...)