

## 2.3 쿠버네티스에 첫번째 애플리케이션 실행하기

🕒 생성일	@2021년 3월 12일 오전 12:52
🏷 태그	

보통 배포하고자 하는 모든 구성 요소를 기술한 JSON이나 YAML 매니페스트를 준비해야 하지만..

일단 간단히 명령어 한 줄로 애플리케이션을 실행해보자

### 1. Node.js 애플리케이션 구동하기

애플리케이션을 배포하기 위한 가장 간단한 방법

```
kubectl run
```

- JSON이나 YAML을 사용하지 않고 필요한 모든 구성 요소를 생성하는 방법

### 도커 허브에 푸시한 이미지를 쿠버네티스에서 실행하기

```
kubectl run kubia --image=superjisonic/kubia --port=8080 --generator=run/v1
```

- `--image` 로 실행하고자 하는 컨테이너 이미지 명시
- `--port` 로 쿠버네티스에 애플리케이션이 8080포트를 수신 대기해야한다는 사실을 알려 준다
- `--generator` (보통은 사용하지 않지만) 여기에서 디플로이먼트 대신 레플리케이션 컨트롤러를 생성하기 때문에 설정

근데 `generator` 옵션은 `deprecated` 되었다고 합니다....! deployment를 사용하라고,,,복제 권장 않기 위해!

### Pod 소개

쿠버네티스는 개별 컨테이너들을 직접 다루지 않음

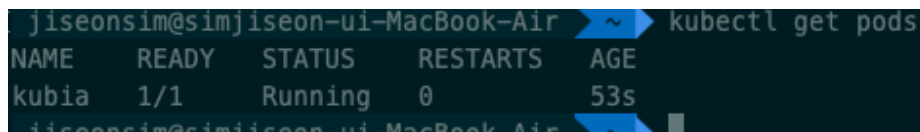
→ 대신 **함께 배치된 다수의 컨테이너 그룹** 이라는 개념인 'Pod' 사용

**파드**란..

- 하나 이상의 밀접하게 연관된 컨테이너의 그룹
- 같은 워커 노드에서 같은 리눅스 네임스페이스로 함께 실행됨
- 각 파드는 자체 IP, hostname, process 등이 있음 → 논리적으로 분리된 머신
  - 애플리케이션은 단일 컨테이너로 실행되는 단일 프로세스일 수도 있고
  - 개별 컨테이너에서 실행되는 주 애플리케이션 프로세스와 부가적으로 도와주는 프로세스로 이뤄질 수도 있다
- 파드에서 실행 중인 모든 컨테이너는 동일한 논리적인 머신에서 실행되는 것처럼 보이지만, 다른 파드에 실행 중인 컨테이너는 같은 워커 노드에서 실행 중이라 할지라도 다른 머신에서 실행 중인 것으로 나타난다

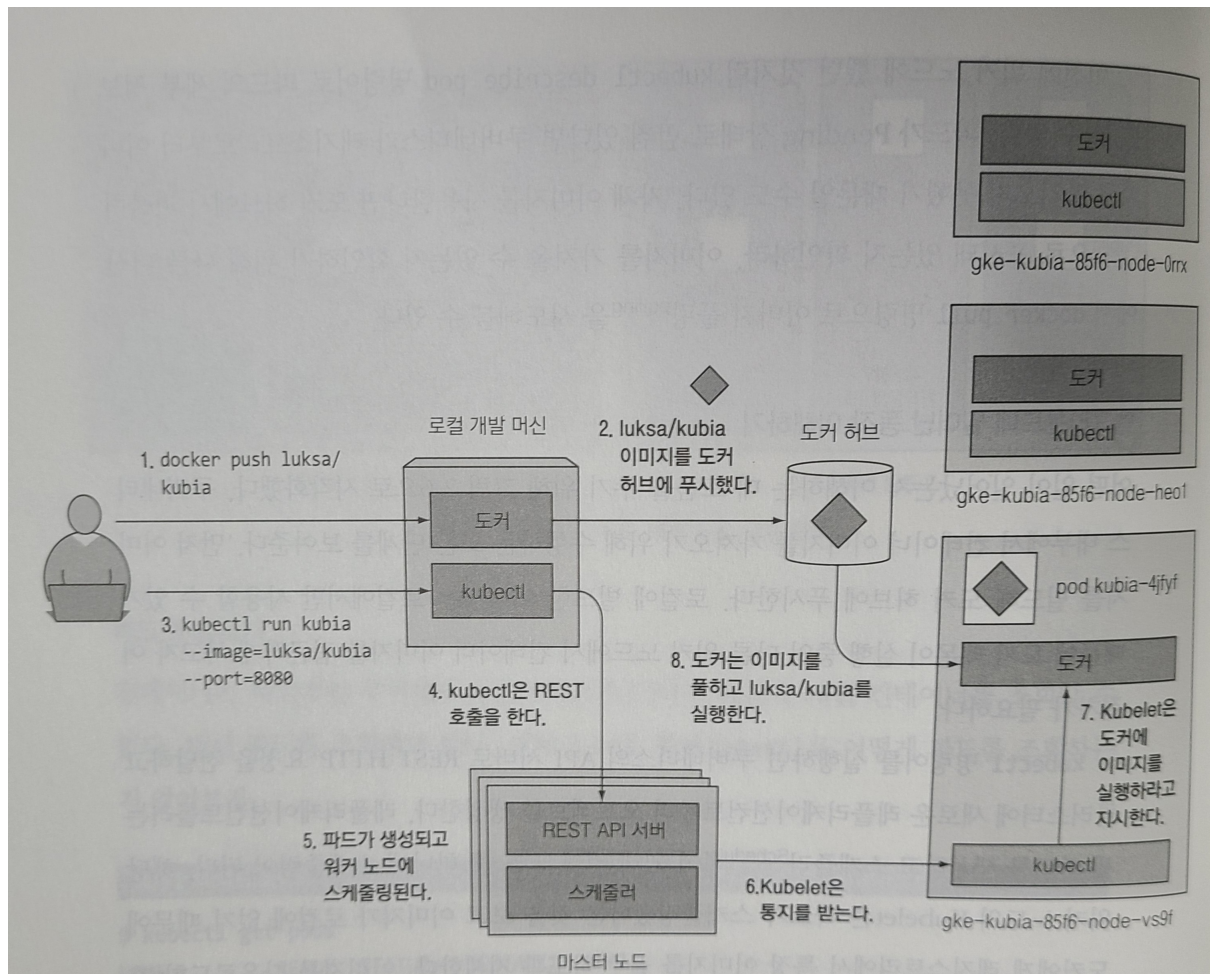
## 파드 조회하기

```
kubectl get pods
```



```
jiseonsim@simjiseon-ui-MacBook-Air ~ % kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
kubia     1/1     Running   0           53s
```

## 백 그라운드에 일어난 동작 이해하기



1. 이미지 빌드 후 도커 허브에 푸시
2. 로컬의 도커 데몬이 도커 허브에 푸시
3. kubectl 명령어로 푸시한 도커 이미지를 실행한다
4. kubectl이 REST로 쿠버네티스 api 서버로 요청 전달
5. 클러스터에 새로운 레플리케이션 컨트롤러 오브젝트가 생성된다
  - 새 Pod가 생성되어 스케줄러에 의해 워커 노드 중 하나에 스케줄링(=파드가 특정 노드에 할당됨, 즉시 실행) 된다
6. 해당 워커노드의 Kubelet은 파드가 스케줄링 됐다는 것을 보고 이미지를 확인한다
7. 이미지가 로컬에 없기 때문에 도커 (컨테이너 런타임) 에게 레지스트리에서 특정 이미지를 Pull 하게끔 지시한다
8. Pull 한 이미지를 도커가 실행한다

여기서 다른 두개 노드는 파드가 스케줄링되지 않았기 때문에 특별한 역할을 하지 않는다

## 2. 웹 애플리케이션에 접근하기

각 파드는 자체 IP 주소를 가지고 있지만 이 주소는 클러스터 내부에 있으며 외부에서 접근 불가..!

마찬가지로 일반적인 Cluster IP 서비스를 생성하면 클러스터 내부에서만 접근 가능

### 외부에서 파드에 접근을 가능하게 하려면..

서비스 오브젝트를 통해 노출해야한다..!

→ Load Balancer 유형의 특별한 서비스를 생성해야한다

| 즉, 생성된 외부 로드 밸런서의 Public IP를 통해 파드에 연결

### 서비스 오브젝트 생성하기

먼저 레플리케이션컨트롤러를 노출하도록 명령한다

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

책에선 이렇게 되어있지만 ^^ 도커 이미지를 run 할때 줬었던 옵션 `--generator` 가 deprecated 되었기때문에 에러 발생한다 ^^ → 리플레이션컨트롤러가 없다고..

<https://www.debugcn.com/ko/article/53165242.html> 참고

대신 다음 두가지 방법이 있다

1. kubia pod 자체를 expose 시키는것

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

```
jiseonsim@simjiseon-ui-MacBook-Air ~$ kubectl expose pod kubia --type=LoadBalancer --name kubia-http
service/kubia-http exposed
jiseonsim@simjiseon-ui-MacBook-Air ~$
```

2. Deployment를 만들어서 노출

deprecated 된 이유가 deployment를 권장하는 것이기 때문에 그냥 deployment를 맹글자..

```
kubectl create deploy kubia-deploy --image=superjisonic/kubia
```

```
jiseonsim@simjiseon-ui-MacBook-Air ~$ kubectl create deploy kubia-deploy --image=superjisonic/kubia
deployment.apps/kubia-deploy created
jiseonsim@simjiseon-ui-MacBook-Air ~$ kubectl get deploy
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
kubia-deploy  1/1     1            1           28s
```

```
kubectl expose deploy kubia-deploy --type=LoadBalancer --name kubia-deploy-http --port=8080
```

```
jiseonsim@simjiseon-ui-MacBook-Air ~$ kubectl expose deploy kubia-deploy --type=LoadBalancer --name kubia-deploy-http --port=8080
service/kubia-deploy-http exposed
```

잘 노출되었다고 한다.. ^^

## 새로 생성된 서비스 오브젝트 조회하기

```
kubectl get services
```

```
jiseonsim@simjiseon-ui-MacBook-Air ~$ kubectl get services
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
kubernetes    ClusterIP     10.79.240.1    <none>         443/TCP          30m
kubia-http    LoadBalancer 10.79.246.212  35.229.167.82  8080:31622/TCP   2m30s
```

- 이제 <http://35.229.167.82:8080> 으로 외부 어디서나 접근 가능

## External IP를 이용해 서비스 접근

```
curl 35.229.167.82:8080
```

```
jiseonsim@simjiseon-ui-MacBook-Air ~$ curl 35.229.167.82:8080
You've hit kubia
```

kubia라고 os의 호스트네임이 잘 뜬다

### 3. 시스템의 논리적인 부분

현재까지 한것

- 가상머신 3대로 워커 노드를 만들어 도커와 Kublet 실행
- 마스터 노드가 전체 시스템 제어

#### 레플리케이션 컨트롤러, 파드, 서비스가 서로 동작하는 방식 이해

kubectl run 명령 수행시,

1. 레플리케이션 컨트롤러 생성 : kubectl run kubectl Replicas: 1
2. 레플리케이션 컨트롤러가 실제 파드를 생성 : kubectl run kubectl
3. 쿠버네티스에 레플리케이션 컨트롤러에 의해 관리되는 모든 파드를 단일 서비스로 노출하도록 명령  
서비스 : kubectl run kubectl-http  
내부 IP : 10.79.246.212  
외부 IP : 35.229.167.82
4. 클러스터 외부에서 접근이 가능하다 (외부IP:8080)

#### 파드와 컨테이너의 이해

파드는 원하는 만큼의 컨테이너를 포함시킬 수 있다

- 파드(kubectl)
- 현재는 파드가 하나의 컨테이너를 가지고 있음
- 자체의 고유한 사설 IP 주소와 호스트 이름을 가짐 (그래서 app.js 에서 'You've hit kubia' 가 뜨는것)
- 컨테이너 내부
  - Node.js 프로세스가 있음
  - 포트 8080에 바인딩 됨

#### 레플리케이션 컨트롤러의 역할 이해

kubectl run 레플리케이션컨트롤러

- 항상 정확히 **하나의** 파드 인스턴스를 실행하도록 지정  
→ 파드 레플리카를 지정하지 않았기때문에 파드를 하나만 생성하는 것.

- 파드가 사라지면 사라진 파드를 대체하기 위해 새 파드 생성

일반적인 레플리케이션 컨트롤러

- 여러개의 파드 복제본을 생성하고 항상 실행 상태로 만듦

## 서비스(kubia-http)가 필요한 이유

: 파드는 일시적이기 때문 (언제든 사라질 수 있다)

→ 파드가 실행중인 노드에 장애 발생, 파드 삭제, 비정상 노드에서 파드 제거 등

→ 이때 레플리케이션 컨트롤러에 의해 생성된 파드(다른 IP 주소 할당됨)로 대체됨



서비스는 다음을 해결해준다

- 항상 변경되는 파드의 IP주소 문제
- 여러개의 파드를 **단일 IP & Port** 로 노출 시키는 문제

[서비스가 하는 일]

서비스가 생성되면 정적 IP를 할당 받음 (서비스 존속하는 동안 변경되지 않음)

클라이언트 → 서비스의 IP 주소를 통해 연결 → 서비스는 파드들의 IP에 관계 없이, 그 중 하나로 연결해 요청 처리

- 서비스는 동일한 서비스를 제공하는 하나 이상의 파드 그룹의 정적 위치를 나타냄
- 서비스의 IP와 포트로 유입된 요청 → 그 순간 서비스 내의 파드 중 하나에 전달되는 것

## 4. 애플리케이션 수평 확장

쿠버네티스 장점 : 간단하게 배포 확장할 수 있음

→ 파드의 개수를 늘려보자

### 현재 레플리케이션 컨트롤러가 관리중인 파드 살펴보기

```
kubectl get replicationcontrollers
```

책에서는 이렇게 나와있지만 ㅎㅎ..

이제 kubectl run명령은 ReplicationController없이 독립형 포드를 생성하기 때문에... 나는

```
kubectl get pods
```

애를 실행해줌

```
jiseonsim@simjiseon-ui-MacBook-Air ~$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
kubia     1/1     Running   0           9h
```

## 의도하는 레플리카 수 늘리기

파드의 레플리카 수를 늘려서 scale out 해보자

replication controller는 deprecated 됨 → 아까 LoadBalancer로 expose 하는 방법 중 2번째였던 deployment로 진행하자..

```
kubectl scale deploy kubia-deploy --replicas=3
```

```
jiseonsim@simjiseon-ui-MacBook-Air ~$ kubectl scale deploy kubia-deploy --replicas=3
deployment.apps/kubia-deploy scaled
jiseonsim@simjiseon-ui-MacBook-Air ~$ kubectl get pods
NAME                                READY   STATUS             RESTARTS   AGE
kubia                              1/1     Running            0           9h
kubia-deploy-7f5b6bd7d4-85rwx      0/1     ContainerCreating  0           8s
kubia-deploy-7f5b6bd7d4-m4sp7      1/1     Running            0           8s
kubia-deploy-7f5b6bd7d4-rrpz7      1/1     Running            0           2m5s
```

```
jiseonsim@simjiseon-ui-MacBook-Air ~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
kubia                              1/1     Running   0           10h
kubia-deploy-7f5b6bd7d4-85rwx      1/1     Running   0           7m7s
kubia-deploy-7f5b6bd7d4-m4sp7      1/1     Running   0           7m7s
kubia-deploy-7f5b6bd7d4-rrpz7      1/1     Running   0           9m4s
```

잘 생성된것을 볼 수 있다

scale 명령어 하나로 인스턴스를 언제든지 추가할 수 있다.



## 서비스 호출 시 모든 파드가 요청을 받는지 확인

다시 서비스를 갖고와보자..

```
[ jiseonsim@simjiseon-ui-MacBook-Air ~ ]$ kubectl get services
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.79.240.1	<none>	443/TCP	10h
kubia-deploy-http	LoadBalancer	10.79.247.39	35.194.216.183	8080:32689/TCP	7m59s
kubia-http	LoadBalancer	10.79.246.212	35.229.167.82	8080:31622/TCP	10h

여기에서 인스턴스를 3개로 추가한 kubia-deploy-http 외부 IP로 curl을 날려보자

```
curl 35.194.216.183:8080
```

```
[ jiseonsim@simjiseon-ui-MacBook-Air ~ ]$ curl 35.194.216.183:8080
You've hit kubia-deploy-7f5b6bd7d4-85rwx
[ jiseonsim@simjiseon-ui-MacBook-Air ~ ]$ curl 35.194.216.183:8080
You've hit kubia-deploy-7f5b6bd7d4-rrpz7
[ jiseonsim@simjiseon-ui-MacBook-Air ~ ]$ curl 35.194.216.183:8080
You've hit kubia-deploy-7f5b6bd7d4-m4sp7
```

요청이 무작위로 다른 파드 호출 → 해당 파드가 클러스터 내에서 이동/생성/삭제로 인해 IP가 변경되지만 서비스는 항상 정적 IP로 동일한 주소를 가짐

[BTW] kubia-http는 스케일링 안했기 때문에 언제든지.. 하나만 갖고온다

## 시스템의 새로운 상태 시각화

- 서비스에서 요청을 파드 3개에 모두 전송한다.
- 세개의 파드 인스턴스가 단일 서비스 IP와 포트로 노출된다.

## 5. 애플리케이션이 실행중인 노드 검사하기

파드가 스케줄링된 노드와 관계 없이

→ 컨테이너 내부에 실행 중인 모든 애플리케이션은 동일한 유형의 운영체제 환경을 갖는다.

→ 각 파드는 자체 IP를 갖고 있기 때문에 같은노드/다른노드 상관없이 통신 가능

## 파드를 조회할 때 파드 IP와 실행중인 노드 표시하기

```
kubectl get pods -o wide
```

```
jiseonsim@simjiseon-ui-MacBook-Air ~ % kubectl get pods -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP            NODE                                NOMINATED NODE   READINESS GATES
kubia                                1/1     Running   0           10h   10.76.2.7     gke-kubia-default-pool-378e3d66-fgr7   <none>            <none>
kubia-deploy-7f5b6bd7d4-85rwx       1/1     Running   0           29m   10.76.1.4     gke-kubia-default-pool-378e3d66-f4rr   <none>            <none>
kubia-deploy-7f5b6bd7d4-m4sp7       1/1     Running   0           29m   10.76.2.9     gke-kubia-default-pool-378e3d66-fgr7   <none>            <none>
kubia-deploy-7f5b6bd7d4-rrpz7       1/1     Running   0           31m   10.76.2.8     gke-kubia-default-pool-378e3d66-fgr7   <none>            <none>
```

## kubectl describe로 파드 세부 정보 살펴보기

```
kubectl describe pod kubia-deploy-7f5b6bd7d4-85rwx
```

## 6. 쿠버네티스 대시보드 소개

GKE를 사용하는 경우.. 다음 명령으로 대시보드 URL 찾을 수 있다

```
kubectl cluster-info
```

근데 뭔가 이상하다...

deprecated 된듯,,?

그냥 GCP 플랫폼에서 보도록 하자 ^^