

9.2 레플리케이션 컨트롤러로 자동 롤링 업데이트 수행

🕒 생성일	@2021년 5월 2일 오전 12:04
🏷️ 태그	

kubectl를 사용해서 롤링업데이트 수행시 프로세스가 간단해짐

1. 애플리케이션의 초기버전 실행

v1 애플리케이션 생성

`luksa/kubia:v1` 이름으로 이미지를 만들어서 도커허브에 푸시

단일 YAML 파일을 사용한 애플리케이션 실행과 서비스 노출

애플리케이션을 호출하기 위해 rc와 로드밸런서 서비스를 생성 → 두개 리소스를 하나의 YAML로 만들고 `kubctl create`해서 쿠버네티스 API에 게시

(`kubia-rc-and-service-v1.yaml` 참조)

- 대시 3개(`---`) 로 구분해서 여러 리소스 정의 포함 가능
- 서비스의 외부 아이피를 찾아서 서비스 요청

```
kubectl get svc kubia
while true ; do curl http://130.211.109.222; done
```

2. kubectl을 이용한 롤링 업데이트

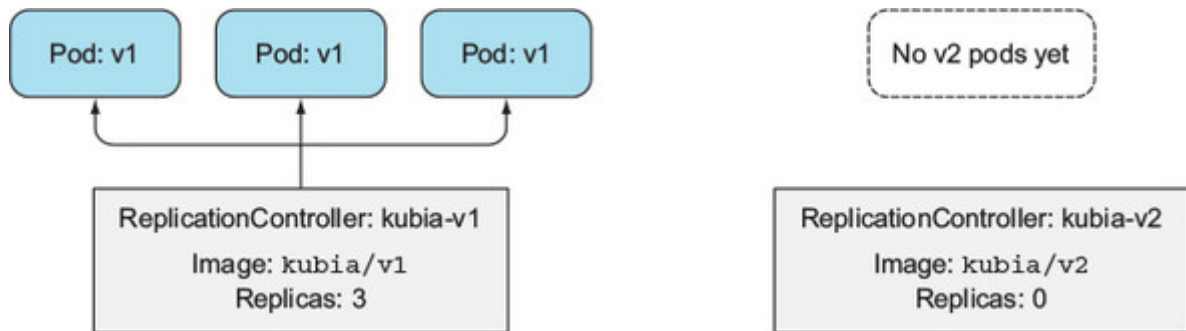
애플리케이션 버전2를 http response "this is v2"를 주는걸로 업데이트 한다

→ `luksa/kubia:v2` 이미지를 사용

[롤링 업데이트를 수행하기 위한 명령어]

```
kubectl rolling-update kubia-v1 kubia-v2 --image=luksa/kubia:v2
```

명령어 수행시 kubia-v2의 새 rc가 즉시 만들어짐



- 의도하는 초기 레플리카 수가 0으로 설정됨

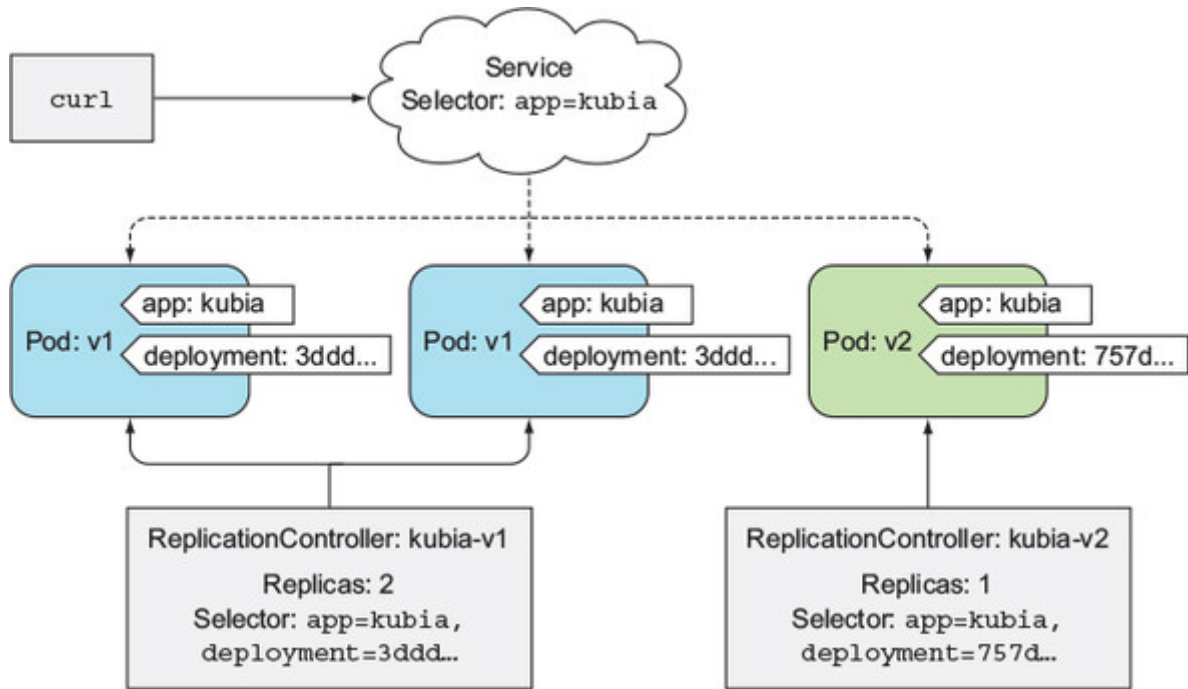
롤링 업데이트가 시작되기 전 kubectl이 수행한 단계 이해하기

1. kubectl이 kubia-v1의 rc를 복사
2. 해당 파드 템플릿에서 이미지를 변경해 새 rc를 만들
3. rc의 레이블 셀렉터를 자세히 보면 rc까지 수정된것을 알 수 있음 → deployment 레이블도 추가로 포함되어있음

[주의점]

- 새 rc와 이전 rc가 동일한 파드 세트에서 작동하면 안됨
- 롤링 업데이트 프로세스에는 첫번째 rc의 셀렉터도 수정이된다
 - kubectl이 rc의 셀렉터를 변경하기 전 실행 중인 파드의 레이블을 먼저 수정
 - 첫번째 컨트롤러의 셀렉터와 매칭되는 파드가 한개도 없는것이 아님

레플리케이션 컨트롤러 두 개를 스케일업해서 새 파드로 교체



kubectl은 이제 새 컨트롤러를 스케일업해서 파드를 하나씩 교체하기 시작

1. 컨트롤러가 첫번째 v2파드를 만듦
2. kubectl은 이전 rc를 하나씩 스케일 다운함
3. 업데이트 프로세스가 v2 파드에 대한 요청 비율이 점점 더 높아짐
 1. v1을 삭제하고 새 이미지를 실행하는 파드로 교체되기 때문
4. 마지막에 이전 rc 파드가 0으로 스케일 다운되면서 끝남

3. 더이상 kubectl rolling-update를 사용하지 않는 이유

1. 저자 스스로 만든 오브젝트를 쿠버네티스가 수정하는 것을 좋아하지 않음
2. kubectl 클라이언트가 업데이트를 수행하는 동안 네트워크가 끊겨지면 업데이트 프로세스가 중간상태에서 끝남
3. 실제 명령을 나타냄 : 의도하는 시스템의 상태를 선언하면 쿠버네티스가 알아서 그것을 달성하는 젤 좋은 방법을 찾아내야함

→ 그래서 디플로이먼트 리소스가 도입됨

