

4.2 레플리케이션 컨트롤러 소개

🕒 생성일	@2021년 3월 25일 오후 11:21
🏷 태그	

레플리케이션컨트롤러: 파드가 항상 실행되도록 보장

- 사라진 파드를 감지해 교체 파드를 생성해줌
 - 클러스터에서 노드가 사라지거나
 - 노드에서 파드가 제거된 경우
- 레플리케이션컨트롤러는 파드의 여러 복제본(Replica)를 작성하고 관리

1. 레플리케이션컨트롤러의 동작

실행 중인 파드 목록을 지속적으로 모니터링. 특정 type의 실제 파드 수가 정상적인 상황의 수와 일치하는지 확인

(여기서 `type` 이란 파드 유형이 아니라 `label selector` 와 일치하는 파드 세트를 일컬음)



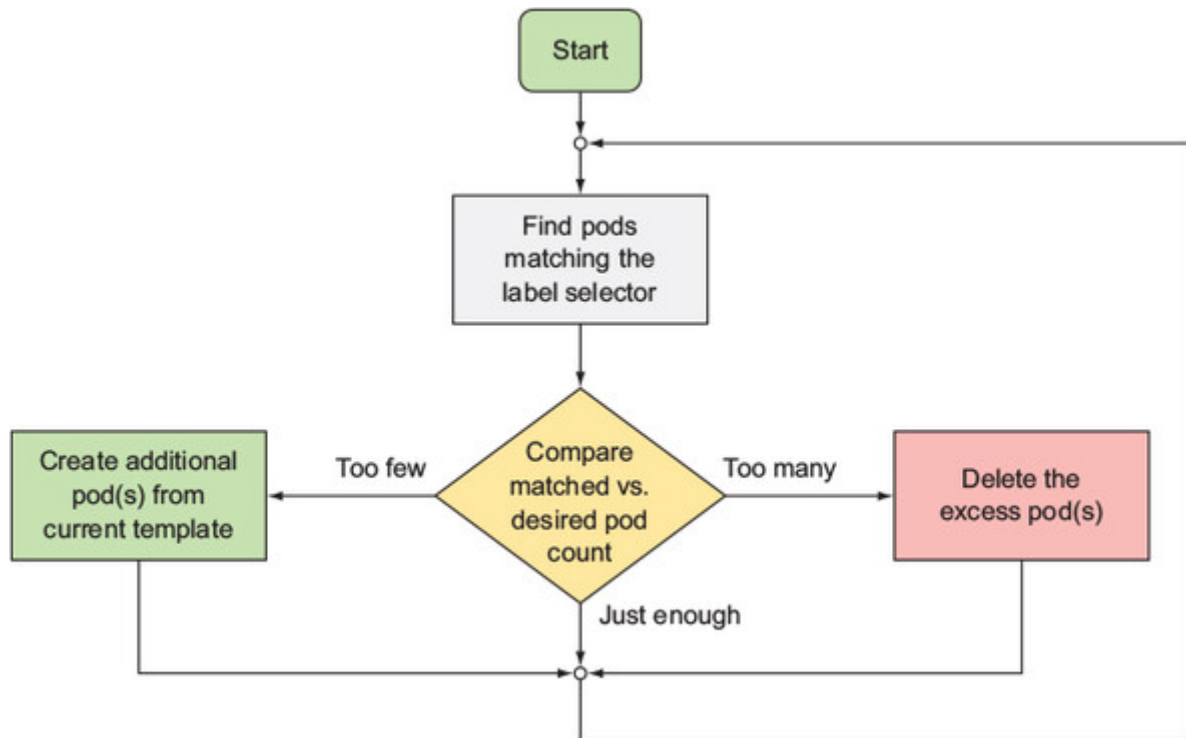
파드 수가 달라지는 case (거의 산업스파이수준) :

- 누군가 같은 type 파드를 수동으로 생성
- 누군가 기존 파드의 타입 변경
- 누군가 의도하는 파드 수를 줄인다

- 파드가 적게 실행 중인 경우 : 파드 템플릿에서 새 복제본 만들
- 파드가 너무 많이 실행 중인 경우: 초과 복제본이 제거됨

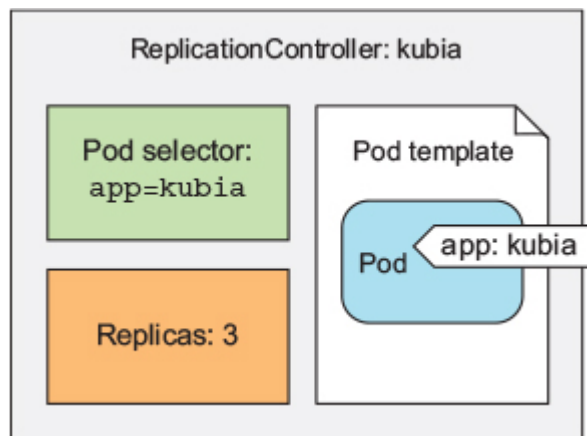
컨트롤러 조정 루프 소개

역할 : 정확한 수의 파드가 레이블 셀렉터와 일치하는지 확인, 불일치시 조치



레플리케이션 컨트롤러의 3 요소

3개 모두 언제든지 수정가능



1. 레이블 셀렉터 (label selector, 파드셀렉터) : 레플리케이션컨트롤러의 범위에 있는 파드를 결정
2. 레플리카 수 (replica count) : 실행할 파드의 desired count를 지정. 변경 시, 기존 파드에 영향을 미친다
3. 파드 템플릿 (pod template) : 새 파드 레플리카를 만들 때 사용

컨트롤러의 레이블 셀렉터/파드 템플릿 변경의 영향 이해

이 둘이 기존 파드에 영향을 미치지 않는 이유

- 레이블 셀렉터 변경 시 : 기존 파드가 레플리케이션컨트롤러의 범위 벗어남 → RC가 해당 파드 관리 중지
- 파드 템플릿 변경 시 : RC는 파드 생성 후 파드의 실제 콘텐츠(컨테이너 이미지, 환경변수 및 기타 사항)에 신경 안씀
 - 새 파드 생성시에만 영향 미침

레플리케이션컨트롤러 사용 시 이점

1. 기존 파드 사라지면 새 파드 시작 → 파드(또는 복제본)가 항상 실행되게 함
2. 클러스터 노드에 장애 발생 시 → 장애 발생한 노드에서 실행 중인 모든 RC 제어하의 파드에 교체 복제본 생성
3. 수동/자동으로 파드 수평확장

2. 레플리케이션 컨트롤러 생성

사실 이제 replication controller를 포함하는 replica set를 사용하고, rs보단 deployment를 권장한다고한다

`kubia-rc.yaml` 을 vi로 만든다

```
apiVersion: v1
kind: ReplicationController # RC정의
metadata:
  name: kubia # RC 이름
spec:
  replicas: 3 # 의도하는 파드 인스턴스 수
  selector: # 파드 셀렉터로 RC가 관리하는 파드 선택
    app: kubia
  template: # 새 파드에 사용할 파드 템플릿
    metadata:
      labels:
        app: kubia
    spec:
      containers:
        - name: kubia
          image: luksa/kubia
          ports:
            - containerPort: 8080
```

쿠버네티스가 레이블 셀렉터 `app=kubia` 와 일치하는 파드 인스턴스를 3개 유지하도록 하는 RC를 생성

- 셀렉터를 지정하지 않으면 템플릿의 레이블로 자동 설정됨
- 파드 레이블이 레이블 셀렉터와 완전 일치하지 않을 시, 컨트롤러에서 새 파드를 무한정 생성할 수 없다

```
kubectl create -f kubia-rc.yaml
```

```
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl create -f kubia-rc.yaml
replicationcontroller/kubia created
```

3. 레플리케이션컨트롤러 작동확인

app=kubia 인 레이블을 가진 파드가 없으므로 RC는 파드 템플릿에서 세개의 새 파드를 가동시켜야함

```
kubectl get pods
```

kubia-h9ct9	1/1	Running	0	5m1s
kubia-qgkgz	1/1	Running	0	5m1s
kubia-z6ffd	1/1	Running	0	5m1s

삭제된 파드에 관한 RC의 반응 확인

먼저 파드 중 하나를 수동으로 삭제해서 RC가 어떻게 새 파드를 생성해 3개를 맞추는지 확인

```
kubectl delete pod kubia-h9ct9
kubectl get pods
```

kubia-h9ct9	1/1	Terminating	0	7m10s
kubia-qgkgz	1/1	Running	0	7m10s
kubia-xqw8l	1/1	Running	0	8s
kubia-z6ffd	1/1	Running	0	7m10s

terminating하는 중에 이미 새 파드 생성함

레플리케이션컨트롤러 정보 얻기

```
kubectl get rc
```

NAME	DESIRED	CURRENT	READY	AGE
kubia	3	3	3	8m12s

막 생성된 따끈한 RC 정보가 보여진다

이제 kubia 이름의 rc를 describe로 알아보자

```
kubectl describe rc kubia
```

```
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl describe rc kubia
Name:          kubia
Namespace:     default
Selector:      app=kubia
Labels:        app=kubia
Annotations:   <none>
Replicas:      3 current / 3 desired
Pods Status:   3 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=kubia
  Containers:
    kubia:
      Image:   luksa/kubia
      Port:    8080/TCP
      Host Port: 0/TCP
      Environment: <none>
      Mounts:     <none>
      Volumes:    <none>
Events:
  Type      Reason            Age   From                  Message
  ----      -
  Normal    SuccessfulCreate   9m50s replication-controller Created pod: kubia-z6ffd
  Normal    SuccessfulCreate   9m50s replication-controller Created pod: kubia-qgkgz
  Normal    SuccessfulCreate   9m50s replication-controller Created pod: kubia-h9ct9
  Normal    SuccessfulCreate   2m48s replication-controller Created pod: kubia-xqw8l
```

아까 종료시킨 파드는 완전히 terminated되어서 running 갯수는 3개이다

컨트롤러가 새로운 파드를 생성한 원인 정확히 이해하기

- RC는 삭제되는 파드에 대해 즉시 **통지**를 받음 : 이 통지 자체가 대체 파드를 생성하게 하는것은 아님
트리거 역할

노드 장애 대응

이 예제를 GKE에서 실행중이라면, 쿠버네티스 클러스터에 노드가 3개 있을 것

GKE 명령을 사용해서 노드 중 하나의 네트워크를 끊는다

(귀찮아서 나중에 정리하겠음)

- 대강 노드에 장애가 발생해도 시스템이 스스로 치유해서 해당 노드에 스케줄된 파드를 새로 생성한다

-고 한다)

4. 레플리케이션컨트롤러의 범위 안팎으로 파드 이동하기

- 레플리케이션컨트롤러
: 파드를 생성했다면, → 어떤 식으로든 레플리케이션컨트롤러와 묶이지 않음
 - 레이블 셀렉터와 일치하는 파드만 관리
 - 파드의 레이블 변경 → RC 범위에서 제거되거나 추가될 수 있음

| RC의 레이블 셀렉터와 일치하지 않게 파드의 레이블을 변경했다면?

해당 파드 : 수동으로 만든 다른 파드처럼 됨 (장애 발생시 리스케줄X)

레플리케이션 컨트롤러가 관리하는 파드에 레이블 추가

```
kubectl label pod kubia-qgkgz type=special
```

```
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl label pod kubia-qgkgz type=special
pod/kubia-qgkgz labeled
```

관리되는 파드에 레이블을 추가해도 RC가 상관하지 않는지 확인하기 위해, `type=special` 레이블을 추가해보자.

```
kubectl get pods --show-labels
```

잘 추가된것을 확인할 수 있다.

```
kubia-qgkgz          1/1      Running    0           27h    app=kubia,type=special
kubia-xqw8l          1/1      Running    0           26h    app=kubia
kubia-z6ffd          1/1      Running    0           27h    app=kubia
```

관리되는 파드의 레이블 변경

이제 레이블 `app=kubia`를 다른것으로 변경 → RC와 레이블 셀렉터간 불일치 발생

3개 중 2개의 파드만 일치하게 됨 : RC는 파드 수를 3개로 되돌리기 위해 새 파드를 시작

```
kubectl label pod kubia-qgkgz app=foo --overwrite
```

* `--overwrite` 인수를 표시하지 않으면 레이블을 변경하지 않는다.

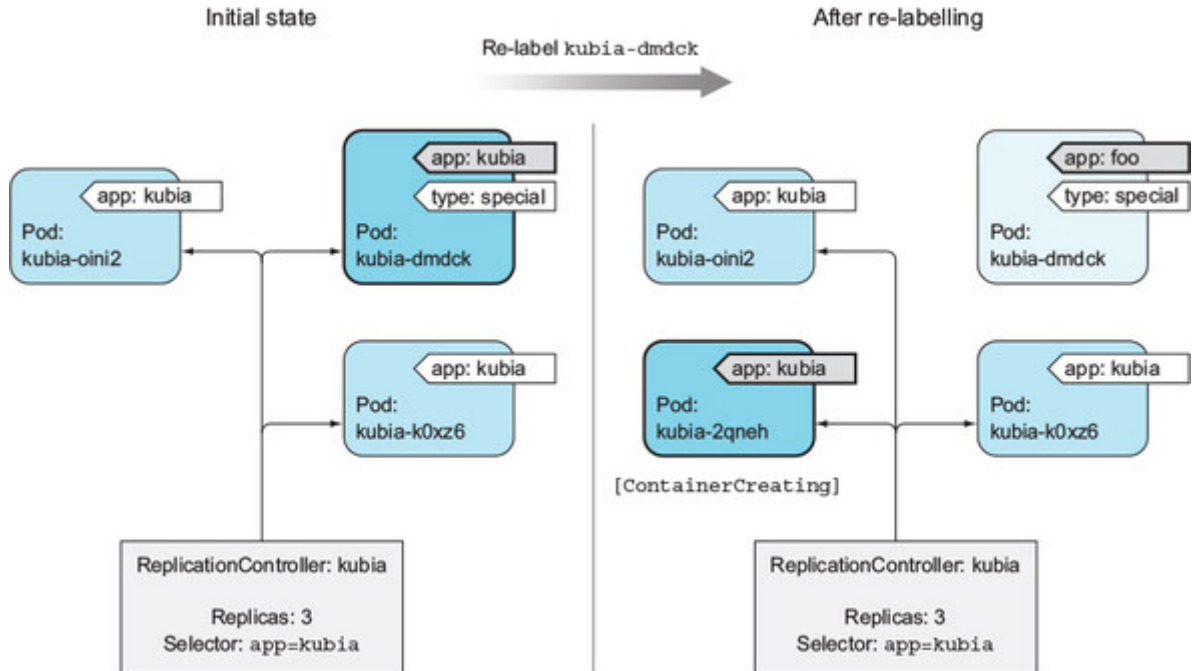
** 새 레이블 추가시 기존 레이블 값 변경방지

```
kubectl get pods -L app
# -L app 옵션은 앱의 레이블을 표시하기 위해 사용한다
```

```
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl label pod kubia-qgkgz app=foo --overwrite
pod/kubia-qgkgz labeled
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl get pods -L app
NAME                                READY   STATUS    RESTARTS   AGE   APP
kubia-6gkf5                         1/1     Running   0          25s   kubia
kubia-deploy-7f5b6bd7d4-9chqt       1/1     Running   0          27h   kubia-deploy
kubia-deploy-7f5b6bd7d4-g9wlw       1/1     Running   0          27h   kubia-deploy
kubia-deploy-7f5b6bd7d4-hcv9h       1/1     Running   0          27h   kubia-deploy
kubia-qgkgz                         1/1     Running   0          27h   foo
kubia-xqw8l                         1/1     Running   0          26h   kubia
kubia-z6ffd                         1/1     Running   0          27h   kubia
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$
```

- 전체 파드 조회시, 맨 위에 RC 범위에서 제거한 파드를 대체하는 새로 생성된 파드를 확인할 수 있다.

- APP이 foo로 되어있는 파드는 더이상 RC가 관리하지 않는 파드



- `app=kubia` 에서 `app=foo` 로 레이블을 변경하면 RC의 범위에서 벗어남
- 컨트롤러의 레플리카 수가 3으로 설정되어있고 두개의 파드만 레이블 셀렉터와 일치
 - RC는 kubia-2qneh 를 기동해서 파드 수를 3개로 다시 만듦

컨트롤러에서 파드를 제거하는 실제 사례

ex) 일정 시간이 지난 후 파드가 제대로 동작하지 않는 버그 : RC가 새 파드로 교체하도록 하고, 문제를 재연하거나 디버그를 해보고, 이후 문제 완료시 파드 삭제

레플리케이션컨트롤러의 레이블 셀렉터 변경

｜ 파드의 레이블을 변경하는 대신 RC의 레이블 셀렉터를 수정한다면?

>> 모든 파드가 RC 범위를 벗어남 → 새로 3개의 파드가 생성도미

5. 파드 템플릿 변경

: RC의 파드 템플릿은 언제든지 수정 가능

BUT 기존 파드 수정시 해당 파드 삭제 → 이후 RC가 새 템플릿을 기반으로 새 파드로 교체하도록 해야함

RC를 편집하는 방법

```
kubectl edit rc kubia
```

```

Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: ReplicationController
metadata:
  creationTimestamp: "2021-03-25T18:37:27Z"
  generation: 1
  labels:
    app: kubia
managedFields:
- apiVersion: v1
  fieldsType: FieldsV1
  fieldsV1:
    f:metadata:
      f:labels:
        .: {}
      f:app: {}
    f:spec:
      f:replicas: {}
      f:selector:
        .: {}
      f:app: {}
      f:template:
        .: {}
        f:metadata:
          .: {}
          f:creationTimestamp: {}
          f:labels:
            .: {}
            f:app: {}
        f:spec:
          .: {}
          f:containers:
            .: {}
            k:{"name":"kubia"}:
              .: {}
              f:image: {}
              f:imagePullPolicy: {}
              f:name: {}
              f:ports:
                .: {}
                k:{"containerPort":8080,"protocol":"TCP"}:

```

"/var/folders/vq/jt2xq9jx091c7nd71qlm00x40000gn/T/kubectl-edit-ot6nd.yaml" 106L, 2776C

이런식으로 RC의 YAML 정의가 열린다. 여기서

1. 파드 템플릿 섹션을 찾는다
2. 메타데이터에 레이블 추가
3. 변경사항 저장, vi 종료
4. kubectl이 RC 업데이트 후 다음 메시지 출력



다른 텍스트 편집기 사용하도록 kubectl edit 설정

`~/.bashrc` 같은 설정파일에

```
export KUBE_EDITOR = "/usr/bin/nano"
```

를 저장해보자

6. 수평 파드 스케일링

파드 수를 늘리거나 줄이는 일 ⇒ RC 리소스의 `replicas` 필드 값을 변경하기만 하면된다.

레플리케이션컨트롤러 스케일 업(확장)

RC가 10개의 파드 인스턴스 실행되도록 설정하기

```
kubectl scale rc kubia --replicas=10
```

또는,

레플리케이션컨트롤러의 정의를 편집해 스케일링하기

```
kubectl edit rc kubia
```

해서 replicas: 3을 `replicas: 10`으로 설정한다

확인해보자.

```
kubectl get rc
```

```
replicationcontroller/kubia edited
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl get rc
NAME      DESIRED   CURRENT   READY   AGE
kubia     10        10        10      27h
```

kubectl scale 명령으로 스케일 다운(축소)하기

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

= edit을 통해 vi로 `spec.replicas` 필드 값을 수정하는 것과 동일

스케일링에 대한 선언적 접근 방법 이해

선언적 접근 방식: 내가 의도하는 상태를 지정하는 것.

- 추가로 실행해야할 인스턴스 수를 쿠버네티스에게 명시적으로 알려줘야함

7. 레플리케이션컨트롤러 삭제

`--cascade=false` 옵션을 추가하면 해당 파드를 계속 실행시킬 수 있다.

```
kubectl delete rc kubia --cascade=false
```