

4.3 레플리케이션 컨트롤러 대신 레플리카셋 사용하기

🕒 생성일	@2021년 3월 25일 오후 11:21
🏷 태그	

레플리카셋(ReplicaSet): 차세대 레플리케이션 컨트롤러. 완전히 대체 가능

- 보통 직접 RS를 생성하지는 않음
- 상위 수준의 디플로이먼트(Deployment) 리소스를 생성할 때 자동으로 생성되게 한다\

1. 레플리카셋과 레플리케이션컨트롤러 비교

레플리카셋이 RC보다 더 나은점

- 파드 셀렉터를 좀 더 **풍부한 표현식** 으로 사용가능
 - 특정 레이블이 없는 파드나 레이블의 값과 상관없이 특정 레이블의 키를 갖는 파드를 매칭 가능
 - RC: 특정 레이블이 있는 파드만 매칭 가능
- 각각 `env=production`, `env=devel` 레이블을 가진 파드 세트를 하나의 레플리카셋으로 모두 매칭시켜 그룹으로 취급 가능
 - RC: 위 두 파드세트를 동시에 매칭시킬 수 없음
- 실제 값이 무엇이든, env 키가 있는 레이블을 갖는 모든 파드를 매칭시킬 수 있다 (`env=*`)

2. 레플리카셋 정의

이제,,, 레플리카셋이 예전 RC에서 생성했다가 버려져서 혼자가 된 파드를 어떻게 다루는지 보자..

vi로 `kubia-replicaset.yaml` 을 만들어주자.

```
apiVersion: apps/v1 # 레플리카셋은 해당 경로에 속한다
kind: ReplicaSet
metadata:
  name: kubia
```

```
spec:
  replicas: 3
  selector:
    matchLabels: #RC와 유사한 간단한 matchLabels 셀렉터를 사용
      app: kubia
  template: #템플릿은 RC와 동일
    metadata:
      labels:
        app: kubia
    spec:
      containers:
        - name: kubia
          image: luksa/kubia
```

- selector 속성 바로 아래에 나열하는 대신 `selector.matchLabels` 아래에 지정

3. 레플리카셋 생성 및 검사

이제 레플리카셋을 생성해보자.

```
kubectl create -f kubia-replicaset.yaml
kubectl get rs
kubectl describe rs
```

yaml 작성시 `apiVersion` 에 주의하자! (책에서 나온것 그대로 하면 에러 뜬다. `apps/v1` 으로 꼭 ,, 수정하기)

```
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl create -f kubia-replicaset.yaml
error: unable to recognize "kubia-replicaset.yaml": no matches for kind "ReplicaSet" in version "apps/v1beta2"
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ vi kubia-replicaset.yaml
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl create -f kubia-replicaset.yaml
replicaset.apps/kubia created
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl get rs
```

NAME	DESIRED	CURRENT	READY	AGE
kubia	3	3	3	44s
kubia-deploy-7f5b6bd7d4	3	3	3	14d

```
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl describe rs
Name:          kubia
Namespace:     default
Selector:      app=kubia
Labels:        <none>
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
```

4. 레플리카셋의 더욱 표현적인 레이블 셀렉터 사용하기

`matchLabels` 셀렉터는 사실상 RC와 같은 기능이다. 더 강력한 `matchExpressions` 를 사용해 보자.

matchExpressions사용한 YAML 만들기

`kubia-replicaset-matchexpressions.yaml` 을 vi로 생성

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: kubia
spec:
  replicas: 3
  selector:
    matchExpressions:
      - key: app #파드의 키가 app인 레이블을 포함해야함
        operator: In
        values:
          - kubia #레이블의 값은 kubia이어야함
  template:
    metadata:
      labels:
        app: kubia
    spec:
      containers:
        - name: kubia
          image: luksa/kubia
```

- `In` : 레이블의 값이 지정된 값 중 하나와 일치해야함
- `NotIn` : 레이블의 값이 지정된 값과 불일치 해야함
- `Exists` : 파드는 지정된 키를 가진 레이블이 포함되어야함. value필드 사용하면 안됨

- `DoesNotExist` : 지정된 키 가진 레이블이 포함되지 않아야함. value 필드 지정 X

여러 표현식 지정한다면, **모든 표현식이 true**여야 셀렉터와 파드가 매칭된다.

→ matchLabels와 matchExpressions 모두 지정시, 모든 label 일치 + 모든 표현식 true

5. 레플리카셋 정리

이제 레플리카셋잘알이 되었으니, 삭제까지 해보면서 클러스터 정리해보자.

```
kubectl delete rs kubia  
kubectl get rs
```

```
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl delete rs kubia  
replicaset.apps "kubia" deleted
```