

6.3 Replica Set

0929 임채림

6.3.1. 레플리카셋을 사용하는 이유

- Pod
 - 쿠버네티스의 기본 단위
 - 여러 개의 컨테이너를 추상화해 하나의 애플리케이션으로 동작하도록 하는 컨테이너 묶음
- Pod 삭제 방법 2가지
 - Kubectl delete로 파드를 삭제하면 파드 컨테이너도 삭제됨
 - 즉 yaml 파일에 파드만 정의해 생성하면 only 쿠버네티스 사용자에게 의해 관리됨

```
$ kubectl delete -f nginx-pod-with-ubuntu.yaml
```

```
$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
my-nginx-pod	0/2	ContainerCreating	0	3s

```
$ kubectl delete pods my-nginx-pod
```

6.3.1. 레플리카셋을 사용하는 이유

- MSA 구조의 Pod
 - 여러 개의 동일한 컨테이너를 생성하고 외부 요청이 각 컨테이너에 적절히 분배될 수 있어야 함
 - 기본 단위는 Pod - 즉 여러 개의 동일한 Pod를 생성해 외부 요청을 분배

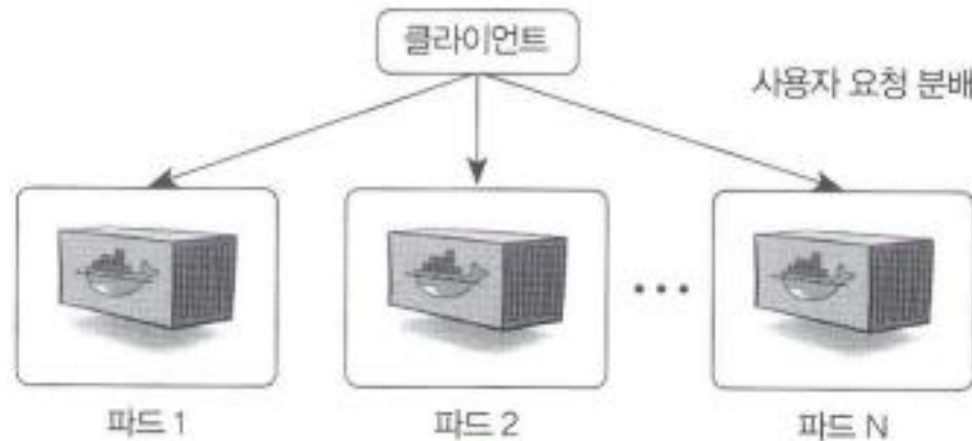


그림 6.6 여러 개의 파드로 사용자의 요청을 분배

6.3.1. 레플리카셋을 사용하는 이유

- 동일한 여러 개의 Pod 생성 방법: yaml로 수작업하기
 - 단점 1. 동일한 파드의 개수가 많아질수록 일일이 정의하면 비효율적
 - 단점 2. IF 파드 삭제됨/ 파드 위치 노드에 장애가 발생
 - 직접 파드를 삭제하고 다시 생성하지 않으면 해당 파드는 복구되지 않음
 - 다른 노드에서 다시 생성되지 않고 종료된 상태로 남아있음
- Replicaset
 - 쿠버네티스 오브젝트
 - 정해진 수의 동일한 Pod가 항상 실행되도록 관리
 - 노드 장애 등의 이유로 파드를 사용할 수 없다면 다른 노드에서 파드 생성
→ 파드를 직접 관리할 필요 없음!

```
apiVersion: v1
kind: Pod
metadata:
  name: my-nginx-pod-a
spec:
  containers:
  - name: my-nginx-container
    image: nginx:latest
    ports:
    - containerPort: 80
      protocol: TCP
---
```

```
apiVersion: v1
kind: Pod
metadata:
  name: my-nginx-pod-b
spec:
  containers:
  - name: my-nginx-container
    image: nginx:latest
    ports:
    - containerPort: 80
      protocol: TCP
```

6.3.2 레플리카셋 사용하기

- Spec.replicas: 몇 개의 동일한 파드 유지시킬 것인지
- Spec.template 아래: 파드 생성시 사용할 템플릿 정의 (파드 템플릿 / 파드 스펙)

예제 6.3 chapter6/replicaset-nginx.yaml

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: replicaset-nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: my-nginx-pods-label
  template:
    metadata:
      name: my-nginx-pod
      labels:
        app: my-nginx-pods-label
    spec:
      containers:
        - name: nginx
          image: nginx:latest
          ports:
            - containerPort: 80
```

↑ 레플리카셋 정의

↓ 파드 정의

```
$ kubectl apply -f replicaset-nginx.yaml
replicaset.apps/replicaset-nginx created
```

```
$ kubectl get po
```

NAME	READY	STATUS	RESTARTS	AGE
replicaset-nginx-gvgdv	1/1	Running	0	60s
replicaset-nginx-pl7s9	1/1	Running	0	60s
replicaset-nginx-sl4r9	1/1	Running	0	60s

```
$ kubectl get rs
```

NAME	DESIRED	CURRENT	READY	AGE
replicaset-nginx	3	3	3	65s

- Apply로 yaml 읽어와 생성 & get으로 레플리카셋과 파드 목록 확인
- Pod 대신 po, replicaset 대신 rs로 줄여 쓰기 가능

6.3.2 레플리카셋 사용하기

- 레플리카셋의 파드 개수 변경을 위해 이미 생성된 레플리카셋 삭제 & 재생성 할 필요 X
- 쿠버네티스에서 이미 생성된 리소스의 속성을 변경하는 기능을 제공
 - Yaml 직접 바꾸기, kubectl edit, kubectl patch 등

```
$ kubectl apply -f replicaset-nginx-4pods.yaml  
replicaset.apps/replicaset-nginx configured
```

- Yaml을 수정하여 apply로 반영

```
$ kubectl delete rs replicaset-nginx  
replicaset.apps "replicaset-nginx" deleted
```

- Delete rs / delete -f 도 가능
- 레플리카셋이 삭제되면 pod도 삭제

6.3.3 레플리카셋의 동작 원리

- 레플리카셋과 파드 loosely coupled - Label Selector로 연결
- Metadata에 리소스 부가정보를 설정 가능
 - 리소스 이름, annotation, label 등
- Label
 - 쿠버네티스 리소스 분류, 서로 다른 오브젝트가 서로를 찾아야 할 때 사용
 - Ex) 레플리카셋은 spec.selector.matchLabel을 통해 생성시킬 파드를 찾음

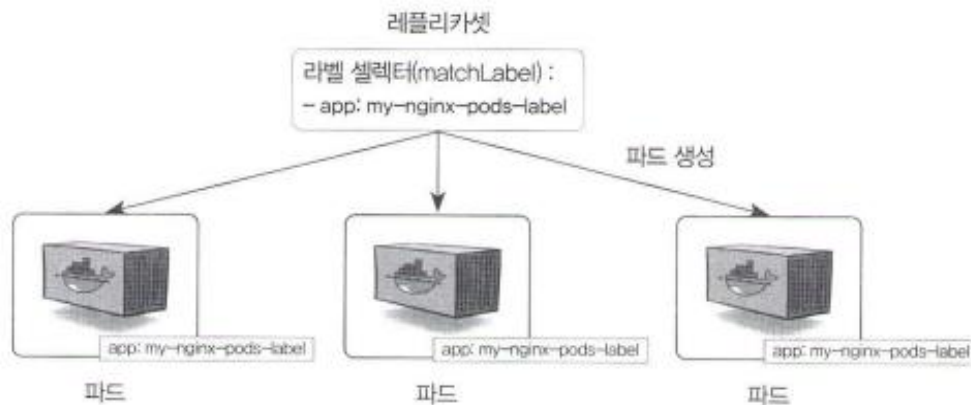


그림 6.7 레플리카셋은 라벨 선택자의 라벨을 가지는 파드를 replicas 개수만큼 생성

```
...
spec:
  replicas: 3
  selector:
    matchLabels:
      app: my-nginx-pods-label
  template:
    metadata:
      name: my-nginx-pod
      labels:
        app: my-nginx-pods-label
...
```

6.3.3 레플리카셋의 동작 원리

- 이미 해당 label의 pod가 생성되었을 경우 / 삭제되었을 경우 이를 확인하고 replicaset이 개수 유지
 - 이미 4개가 존재하면 하나를 삭제시킴

```
$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
my-nginx-pod	1/1	Running	0	44s
replicaset-nginx-c8vf5	0/1	ContainerCreating	0	1s
replicaset-nginx-frnxl	0/1	ContainerCreating	0	1s

3개 만들어야 하는데 이미 하나 있으면 2개만 만들

```
$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
replicaset-nginx-2fhd7	1/1	Running	0	23s
replicaset-nginx-c8vf5	1/1	Running	0	15m
replicaset-nginx-frnxl	1/1	Running	0	15m

하나 지워지면 하나 추가해서 만들

- --show-labels, -l

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

NAME	READY	STATUS	RESTARTS	AGE	LABELS
my-nginx-pod	1/1	Running	0	36s	app=my-nginx-pods-label

```
$ kubectl get pods -l app
```

NAME	READY	STATUS	RESTARTS
my-nginx-pod	1/1	Running	0

```
$ kubectl get pods -l app=my-nginx-pods-label
```

NAME	READY	STATUS	RESTARTS	AGE
my-nginx-pod	1/1	Running	0	44s

6.3.3 레플리카셋의 동작 원리

- 관리되던 pod의 라벨을 삭제하면 더 이상 레플리카셋의 관리 대상이 아님
 - Selector.matchLabel과 불일치
 - 직접 수동으로 생성한 파드와 동일, 레플리카셋을 삭제해도 삭제되지 않음

```
$ kubectl edit pods replicaset-nginx-2fhd7  
pod/replicaset-nginx-2fhd7 edited
```

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

NAME	READY	STATUS	RESTARTS	AGE	LABELS
replicaset-nginx-2fhd7	1/1	Running	0	14m	<none>
replicaset-nginx-c8vf5	1/1	Running	0	29m	app=my-nginx-pods-label
replicaset-nginx-frnxl	1/1	Running	0	29m	app=my-nginx-pods-label
replicaset-nginx-qksdx	1/1	Running	0	13s	app=my-nginx-pods-label

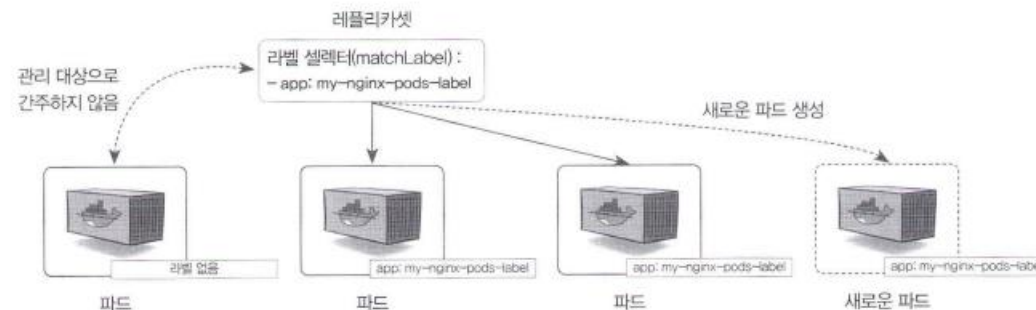


그림 6.9 레플리카셋은 라벨 선택기와 일치하는 파드만 관리 대상으로 간주