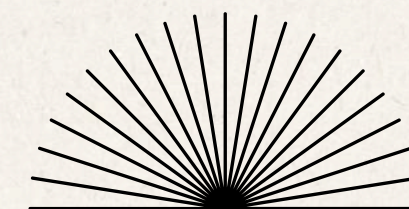


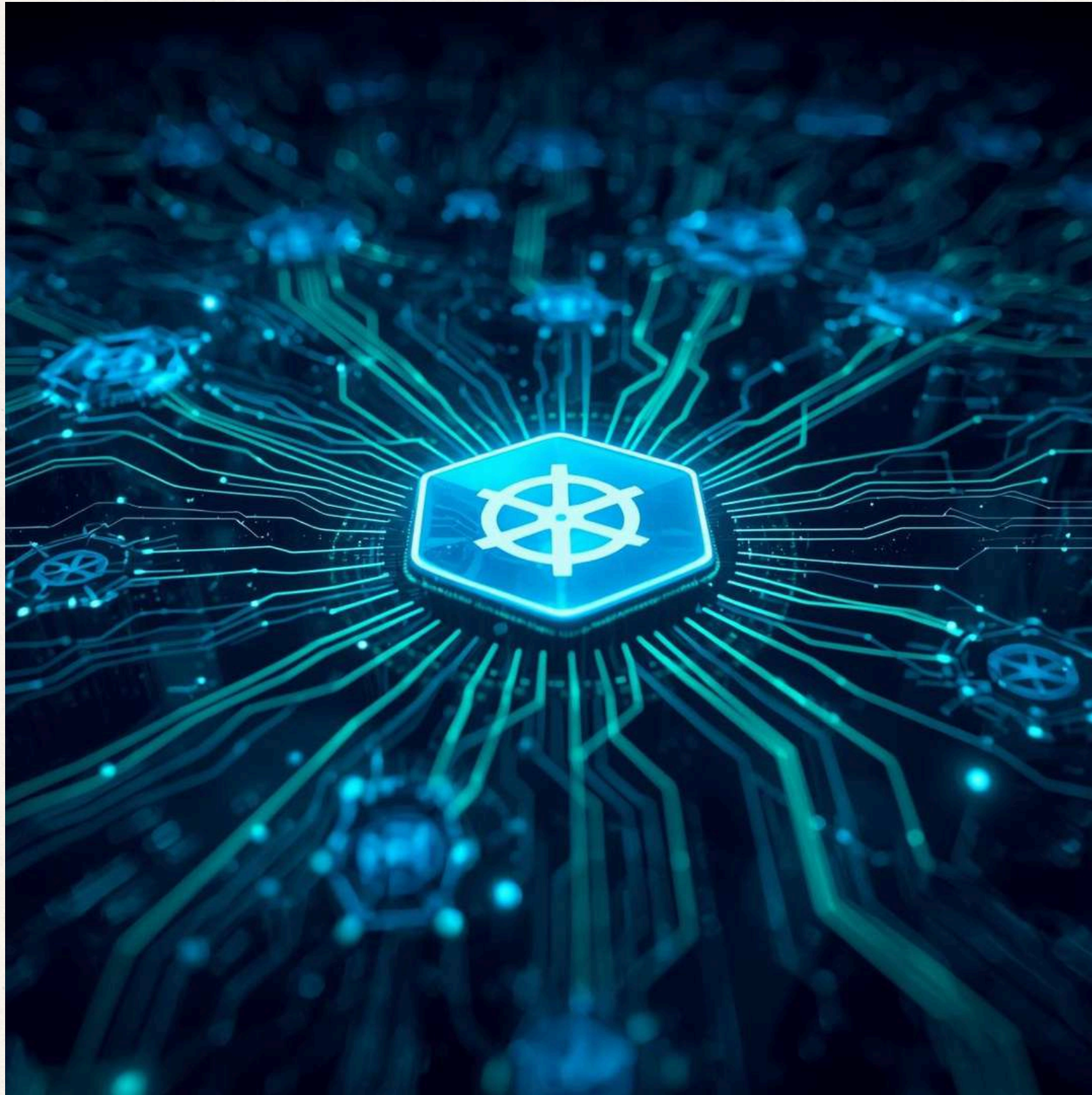


# 쿠버네티스 교과서

ch09





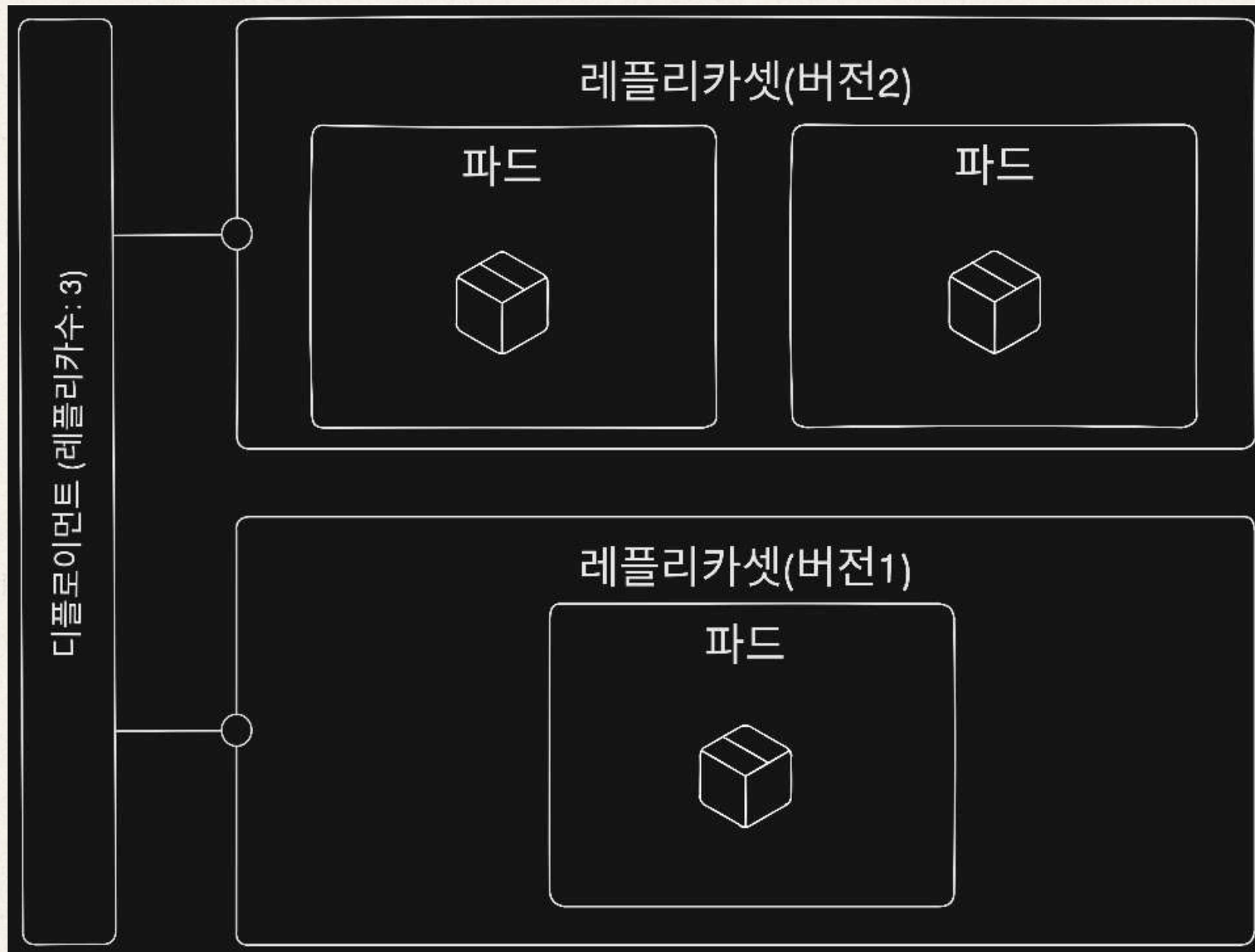


- 1 쿠버네티스 롤링 업데이트
- 2 롤아웃과 롤백을 이용한 디플로이먼트 업데이트
- 3 디플로이먼트의 롤링 업데이트 설정
- 4 데몬셋과 스테이트풀셋의 롤링 업데이트
- 5 릴리스 전략 이해하기
- 6 문제풀이



# 1. 쿠버네티스 롤링 업데이트

03/23



롤아웃은 파드의 정의가 변경될 때만 발생

```
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl apply -f vweb/
service/vweb created
deployment.apps/vweb created
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl get rs -l app=vweb
NAME                DESIRED    CURRENT    READY    AGE
vweb-695cc8864d      2          2          2        9s
vweb-v1-695cc8864d   4          4          4        26h
vweb-v2-5c5cf94d68   4          4          4        26h
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl apply -f vweb/update/vweb-v1-scale.yaml
deployment.apps/vweb configured
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl get rs -l app=vweb
NAME                DESIRED    CURRENT    READY    AGE
vweb-695cc8864d      3          3          3        39s
vweb-v1-695cc8864d   4          4          4        26h
vweb-v2-5c5cf94d68   4          4          4        26h
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl rollout history deploy/vweb
deployment.apps/vweb
REVISION  CHANGE-CAUSE
1          <none>
```

파드의 정의가 변경된 것이 아닐때는 새로운 rs가 생성되지 않는다.

```
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl set image deployment/vweb web=kiamol/ch09-vweb:v2
deployment.apps/vweb image updated
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl get rs -l app=vweb
NAME                DESIRED    CURRENT    READY    AGE
vweb-695cc8864d      0          0          0        3m1s
vweb-6f45cfc7c9      3          3          3        11s
vweb-v1-695cc8864d   4          4          4        26h
vweb-v2-5c5cf94d68   4          4          4        26h
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl rollout history deploy/vweb
deployment.apps/vweb
REVISION  CHANGE-CAUSE
1          <none>
2          <none>
```

파드의 정의가 변경되었을때는 새로운 rs가 생성된다.



## 2.롤아웃과 롤백을 이용한 디플로이먼트 업데이트

04/23

```
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl apply -f vweb/update/vweb-v11.yaml --record
Flag --record has been deprecated, --record will be removed in the future
deployment.apps/vweb configured
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl get rs -l app=vweb --show-labels
NAME          DESIRED   CURRENT   READY   AGE    LABELS
vweb-695cc8864d 0         0         0       117s   app=vweb,pod-template-hash=695cc8864d,version=v1
vweb-6f45cfc7c9 0         0         0       64s    app=vweb,pod-template-hash=6f45cfc7c9,version=v1
vweb-c7c4b5bd7  3         3         3       24s    app=vweb,pod-template-hash=c7c4b5bd7,version=v1.1
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl rollout status deploy/vweb
deployment "vweb" successfully rolled out
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl rollout history deploy/vweb
deployment.apps/vweb
REVISION  CHANGE-CAUSE
1         <none>
2         <none>
3         kubectl apply --filename=vweb/update/vweb-v11.yaml --record=true

ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl get rs -l app=vweb -o=custom-columns=NAME:.metadata.name,REPLICAS:.status.replicas,REVISION:.metadata.annotations.deployment\.kubernetes\.io/revision
NAME          REPLICAS  REVISION
vweb-695cc8864d 0          <none>
vweb-6f45cfc7c9 0          <none>
vweb-c7c4b5bd7  3          <none>
```

파드 정의의 version레이블이  
디플로이먼트에 의해 rs의  
레이블로도 추가

“ --record”  
: 디플로이먼트 변경시 사용한 명령어를  
기록



## 2.롤아웃과 롤백을 이용한 디플로이먼트 업데이트

05/23

```
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl get svc vweb -o jsonpath='{http://{.status.loadBalancer.ingress[0].*}:8090/v.txt}'
ubuntu@local:~/files/study/k8s/kiamol/ch09$ cat url.txt
http://192.168.0.157:8090/v.txt
ubuntu@local:~/files/study/k8s/kiamol/ch09$ curl $(cat url.txt)
v1ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl apply -f vweb/update/vweb-v2.yaml --record
Flag --record has been deprecated, --record will be removed in the future
deployment.apps/vweb configured
ubuntu@local:~/files/study/k8s/kiamol/ch09$ curl $(cat url.txt)
v2ubuntu@local:~/files/study/k8s/kiamol/ch09$

ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl get rs -l app=vweb --show-labels
NAME          DESIRED  CURRENT  READY  AGE   LABELS
vweb-5c5cf94d68 3         3        3      68m   app=vweb,pod-template-hash=5c5cf94d68,version=v2
vweb-695cc8864d 0         0        0      88m   app=vweb,pod-template-hash=695cc8864d,version=v1
vweb-6f45cfc7c9 0         0        0      87m   app=vweb,pod-template-hash=6f45cfc7c9,version=v1
vweb-c7c4b5bd7 0         0        0      87m   app=vweb,pod-template-hash=c7c4b5bd7,version=v1.1
```

```
spec:
  replicas: 3
  selector:
    matchLabels:
      app: vweb
  template:
    metadata:
      labels:
        app: vweb
        version: v1.1
```

LABELS에 있는 version으로 레플리카셋 속 파드의 버전을 파악할수 있다.  
목록은 파드 템플릿의 해시값 순서대로 정렬되어 있음

롤아웃은 rs의 세부사항을 추상화 해준다



## 2. 롤아웃과 롤백을 이용한 디플로이먼트 업데이트

```
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl rollout history deploy/vweb
deployment.apps/vweb
REVISION  CHANGE-CAUSE
1          <none>
2          <none>
3          kubectl apply --filename=vweb/update/vweb-v11.yaml --record=true
4          kubectl apply --filename=vweb/update/vweb-v2.yaml --record=true

ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl get rs -l app=vweb -o=custom-columns=NAME:.metadata.name,REPLICAS:.status.replicas,VERSION:.metadata.labels.version,REVISION:.metadata.annotations.deployment\.kubernetes\.io/revision
NAME          REPLICAS  VERSION  REVISION
vweb-5c5cf94d68  3         v2       <none>
vweb-695cc8864d  0         v1       <none>
vweb-6f45cfc7c9  0         v1       <none>
vweb-c7c4b5bd7   0         v1.1     <none>

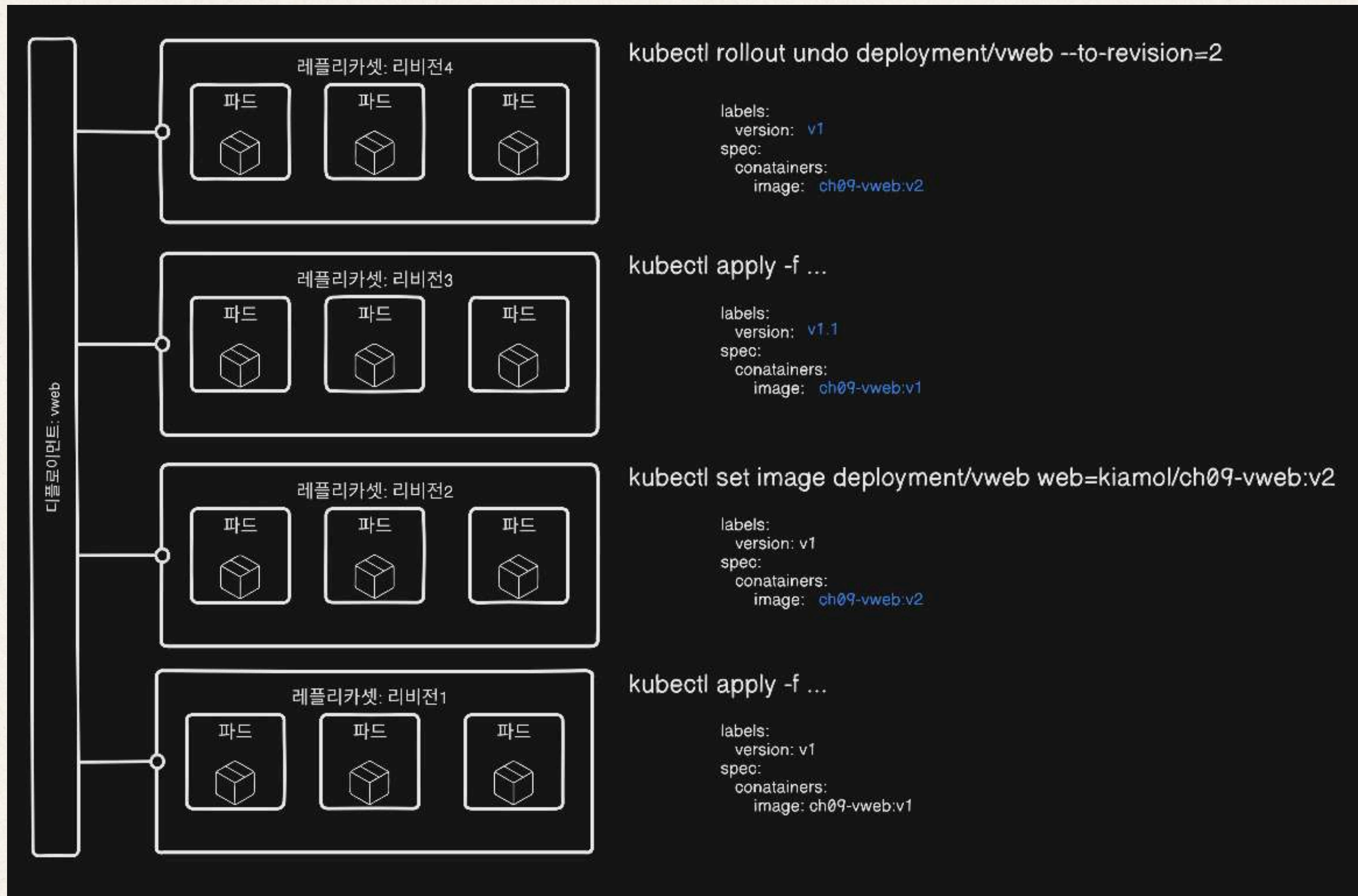
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl rollout undo deploy/vweb --dry-run
W0205 22:11:09.244353 128104 helpers.go:731] --dry-run is deprecated and can be replaced with --dry-run=client.
deployment.apps/vweb Pod Template:
  Labels:      app=vweb
              pod-template-hash=c7c4b5bd7
              version=v1.1
a
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl rollout undo deploy/vweb --to-revision=2
deployment.apps/vweb rolled back
ubuntu@local:~/files/study/k8s/kiamol/ch09$ curl $(cat url.txt)
v2ubuntu@local:~/files/study/k8s/kiamol/ch09$
```

롤아웃을 했으나 version이 v1이 아닌 여전히 v2인 상태



## 2.롤아웃과 롤백을 이용한 디플로이먼트 업데이트

07/23

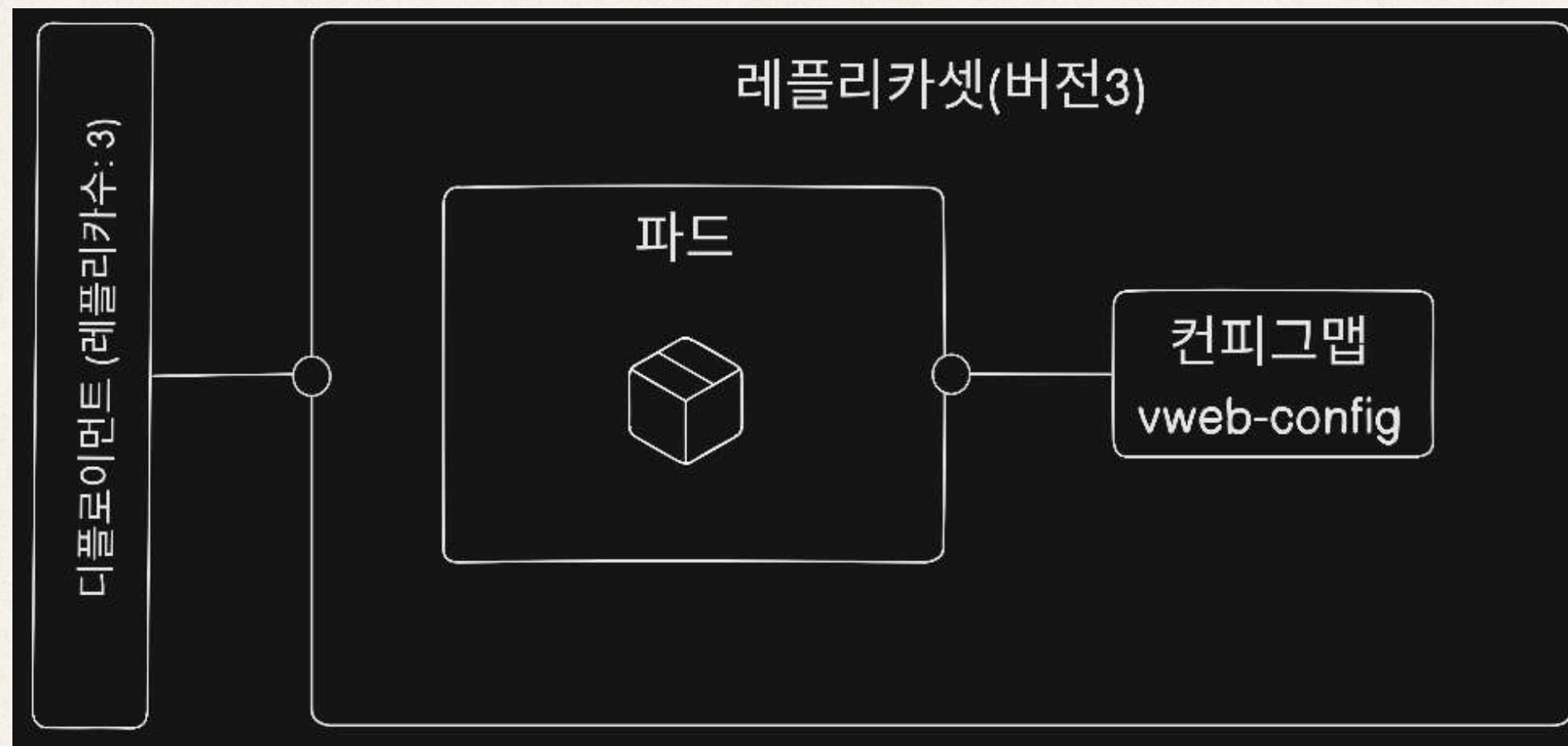




## 2. 롤아웃과 롤백을 이용한 디플로이먼트 업데이트

컨피그맵을 사용한 디플로이먼트 업데이트 p.286

08/23



```
spec:
  containers:
    - name: web
      image: kiamol/ch09-vweb:v2
      ports:
        - name: http
          containerPort: 80
      volumeMounts:
        - name: static
          mountPath: "/usr/share/nginx/html/"
          readOnly: true
  volumes:
    - name: static
      configMap:
        name: vweb-config
```

파드에 컨피그맵을 볼륨으로 마운트 하고 컨피그맵의 내용을 수정하여  
파드의 애플리케이션을 업데이트 할수 있다. (hot reload: 핫 리로드 방식)

- 발생할수 있는 문제점
- 파드의 정의를 수정하는 것이 아니라  
rollout이 발생하지 않음
- : 이전 버전으로 rollback을 할수 없음



## 2.롤아웃과 롤백을 이용한 디플로이먼트 업데이트

실습

09/23

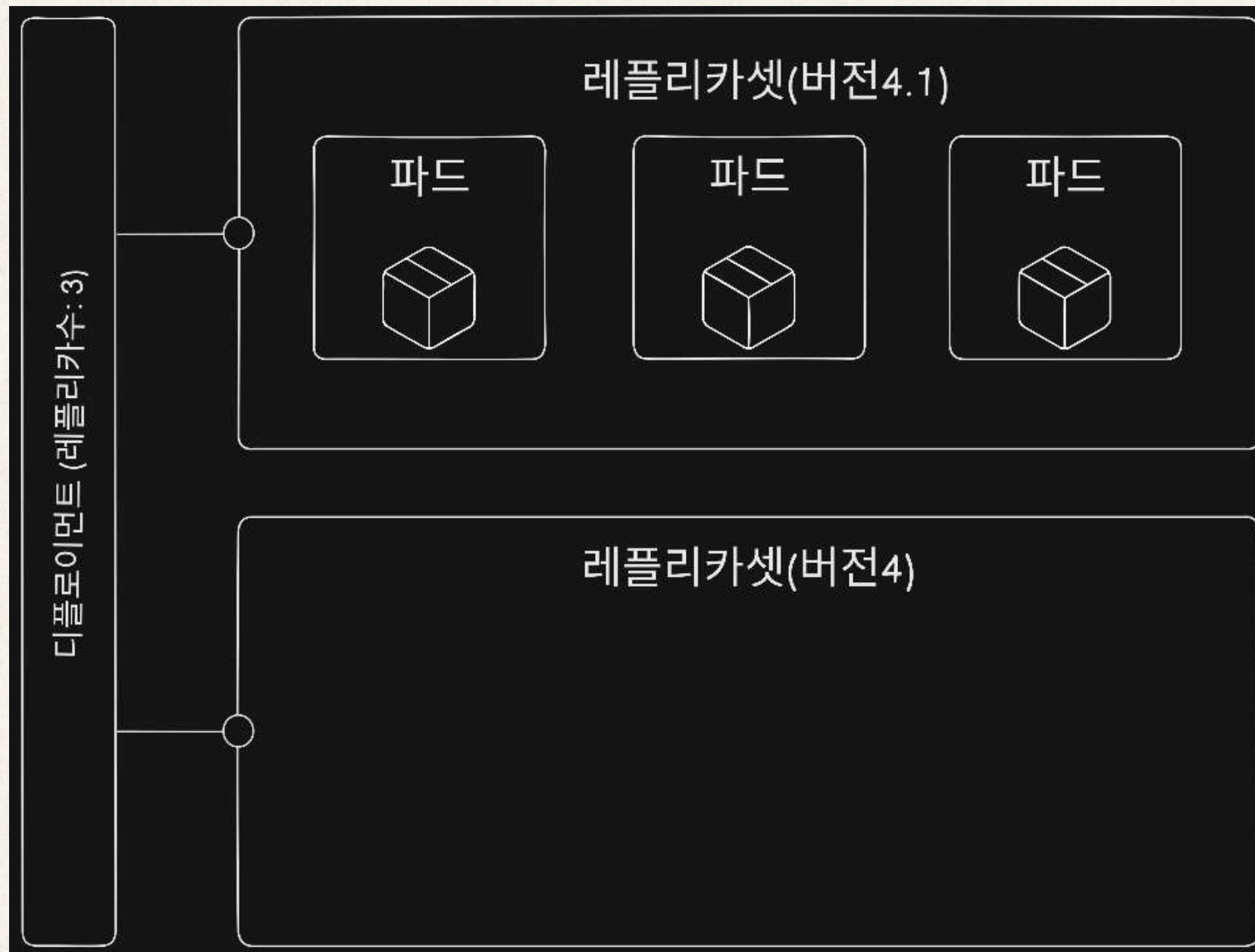
```
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl delete deploy vweb
deployment.apps "vweb" deleted from default namespace
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl apply -f vweb/update/vweb-v3-with-configMap.yaml --record
Flag --record has been deprecated, --record will be removed in the future
configmap/vweb-config configured
deployment.apps/vweb created
ubuntu@local:~/files/study/k8s/kiamol/ch09$ curl $(cat url.txt)
v3-from-configubuntu@local:~/files/study/k8s$ kubectl apply -f vweb/update/vweb-configMap-v31.yaml --record
Flag --record has been deprecated, --record will be removed in the future
configmap/vweb-config configured
ubuntu@local:~/files/study/k8s/kiamol/ch09$ sleep 120
ubuntu@local:~/files/study/k8s/kiamol/ch09$ curl $(cat url.txt)
v3.1ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl rollout history deploy/vweb
deployment.apps/vweb
REVISION  CHANGE-CAUSE
1          kubectl apply --filename=vweb/update/vweb-v3-with-configMap.yaml --record=true
```



## 2. 롤아웃과 롤백을 이용한 디플로이먼트 업데이트

파드의 정의 파일에서 새로운 컨피그맵을 사용하도록 수정 p288

10/23



```
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl delete deploy vweb
deployment.apps "vweb" deleted from default namespace
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl apply -f vweb/update/vweb-v4-with-configMap.yaml --record
Flag --record has been deprecated, --record will be removed in the future
configmap/vweb-config-v4 created
deployment.apps/vweb created
ubuntu@local:~/files/study/k8s/kiamol/ch09$ curl $(cat url.txt)
v4-from-config
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl apply -f vweb/update/vweb-v41-with-configMap.yaml --record
Flag --record has been deprecated, --record will be removed in the future
configmap/vweb-config-v41 created
deployment.apps/vweb configured
ubuntu@local:~/files/study/k8s/kiamol/ch09$ curl $(cat url.txt)
v4.1-from-config
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl rollout history deploy/vweb
deployment.apps/vweb
REVISION  CHANGE-CAUSE
1          kubectl apply --filename=vweb/update/vweb-v4-with-configMap.yaml --record=true
2          kubectl apply --filename=vweb/update/vweb-v41-with-configMap.yaml --record=true

ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl rollout undo deploy/vweb
deployment.apps/vweb rolled back
ubuntu@local:~/files/study/k8s/kiamol/ch09$ curl $(cat url.txt)
v4-from-config
ubuntu@local:~/files/study/k8s/kiamol/ch09$
```

파드 정의에서 새로운 컨피그맵을 사용하도록 수정

파드 정의의 내용을  
수정하는 것이므로  
이전의 버전의 레플리카셋의  
이력이 있으므로 이전 버전으로  
rollout(rollback) 가능



### 3.디플로이먼트의 롤링 업데이트 설정

11/23

```
spec:
  replicas: 3
  selector:
    matchLabels:
      app: vweb
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: vweb
        version: v2
    spec:
      containers:
```

```
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl delete deploy vweb
deployment.apps "vweb" deleted from default namespace
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl apply -f vweb-strategies/vweb-recreate-v2.yaml

deployment.apps/vweb created
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl get rs -l app=vweb
NAME                                DESIRED   CURRENT   READY   AGE
vweb-5c5cf94d68                     3         3         3       11s
ubuntu@local:~/files/study/k8s/kiamol/ch09$ curl $(cat url.txt)
v2ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl describe deploy vweb
Name:                                vweb
Namespace:                          default
CreationTimestamp: Thu, 05 Feb 2026 23:27:44 +0900
Labels:                             kiamol=ch09
Annotations: deployment.kubernetes.io/revision: 1
Selector:                           app=vweb
Replicas: 3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType: Recreate
```

Recreate는 RollintUpdate와는 달리 기존 레플리카셋의 파드 수를 0까지 감소시킨후에 새 레플리카셋의 파드 수를 증가 시킨다.

-업데이트 전략  
→ 롤링 업데이트  
→ 리크리에이트



### 3.디플로이먼트의 롤링 업데이트 설정

12/23

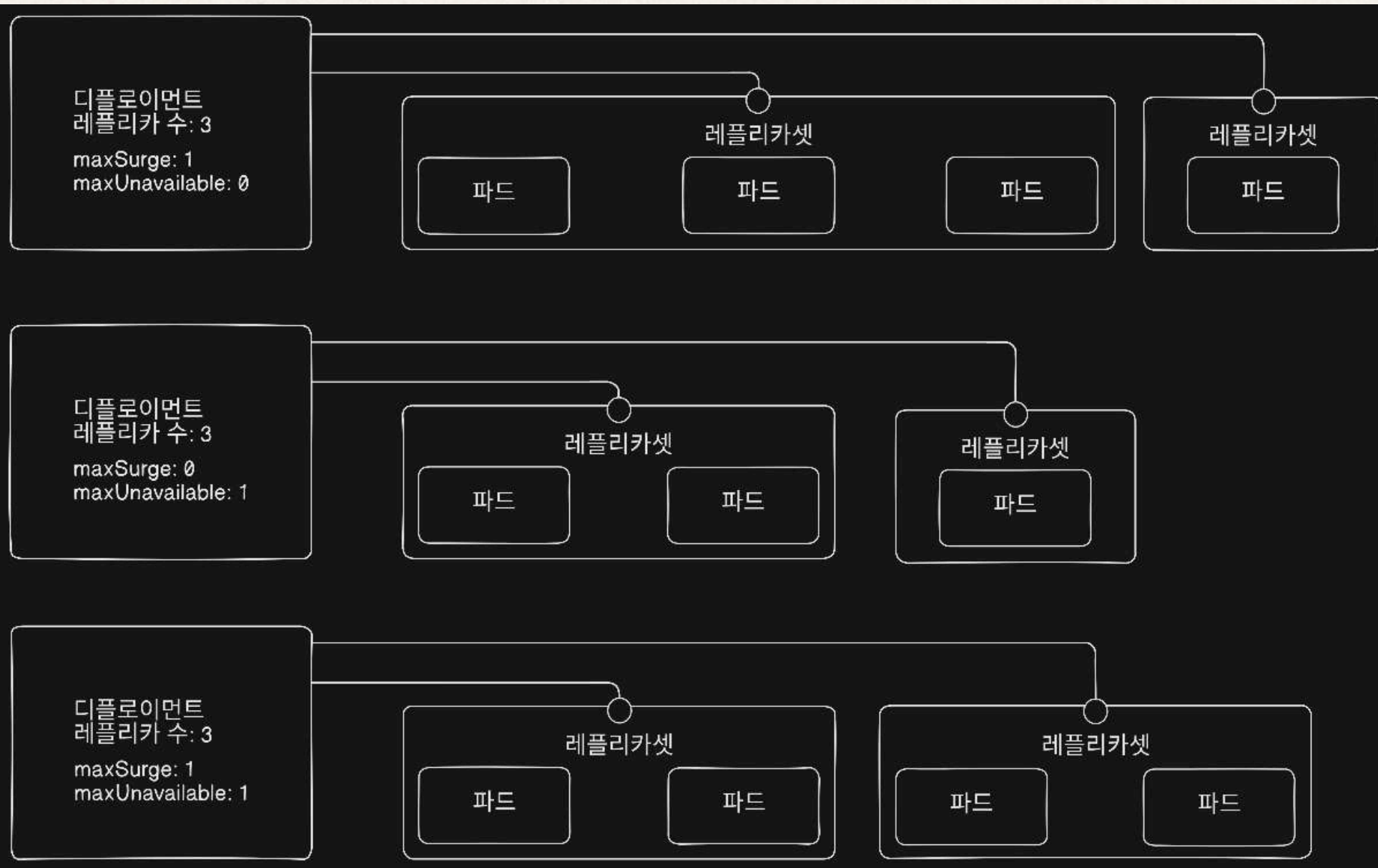
```
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl apply -f vweb-strategies/vweb-recreate-v3.yaml
deployment.apps/vweb configured
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl rollout status deploy/vweb --timeout=2s
Waiting for deployment "vweb" rollout to finish: 0 of 3 updated replicas are available...
error: timed out waiting for the condition
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl get rs -l app=vweb
NAME                DESIRED    CURRENT    READY    AGE
vweb-5c5cf94d68      0          0          0        8m17s
vweb-f6977cddd       3          3          0        30s
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl get pods -l app=vweb
NAME                READY    STATUS              RESTARTS    AGE
vweb-f6977cddd-csgw2 0/1      CrashLoopBackOff    2 (16s ago) 41s
vweb-f6977cddd-skppt 0/1      CrashLoopBackOff    2 (20s ago) 41s
vweb-f6977cddd-x5r8x 0/1      CrashLoopBackOff    2 (17s ago) 41s
ubuntu@local:~/files/study/k8s/kiamol/ch09$ curl $(cat url.txt)
curl: (7) Failed to connect to 192.168.0.157 port 8090 after 0 ms: Couldn't connect to server
```

Recreate 전략 사용시 새 파드에 오류가 있다면 이전의 파드가 모두 제거되었기 때문에 애플리케이션에 접근이 불가하다

-rolling update는 rollout 전략의 기본값이다.



### 3.디플로이먼트의 롤링 업데이트 설정



```
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl apply -f vweb-strategies/vweb-rollingUpdate-v2
.yaml
deployment.apps/vweb configured
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl get po -l app=vweb
NAME                                READY   STATUS    RESTARTS   AGE
vweb-5c5cf94d68-kz6zg               1/1     Running   0           7s
vweb-5c5cf94d68-vn8s7               1/1     Running   0           9s
vweb-5c5cf94d68-vxlzv               1/1     Running   0           8s
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl rollout status deploy/vweb
deployment "vweb" successfully rolled out
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl get rs -l app=vweb
NAME                                DESIRED   CURRENT   READY   AGE
vweb-5c5cf94d68                     3         3         3       30m
vweb-f6977cddd                      0         0         0       22m
ubuntu@local:~/files/study/k8s/kiamol/ch09$ curl $(cat url.txt)
v2ubuntu@local:~/files/study/k8s/kiamol/ch09$
```

```
strategy:
  type: RollingUpdate
  rollingUpdate:
    maxUnavailable: 0
    maxSurge: 1
```

예제의 전략 정의 내용

-프로세스 (경우에 따라서 pod을 먼저 제거하고 그다음에 새로운 pod을 생성 할수도 있음)

→ Deployment에 변경(새 PodTemplate)이 생기면 새 ReplicaSet을 생성한다.

→ 새 ReplicaSet에 파드를 1개 생성.

->새 Pod가 Ready 되면, 기존(old) ReplicaSet을 1개 스케일 다운한다.

->위 과정을 반복해서 새 ReplicaSet은 목표 개수까지 올리고, old ReplicaSet은 0까지 내린다.

-옵션

→ maxUnavailable

:업데이트동안 사용할수 없는  
파드의 최대 수

: 기존 레플리카셋에서 동시에  
종료되는 파드수

: 기본값 25%

→ maxSurge

: 새 레플리카셋에 동시에 함께  
시작되는 파드 수

: 기본값 25%



총 3개의 레플리카가 존재할때

```
StrategyType:      RollingUpdate
MinReadySeconds:    0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
```

1. maxSurge: 1, maxUnavailable: 0

→ 새로운 레플리카셋에 새로운 파드를 하나 생성한후, 파드가 Ready 상태이면  
기존의 파드를 1개 제거

2. maxSurge: 0, maxUnavailable: 1

→ 기존의 레플리카셋의 파드를 하나 제거한 후에 새로운 레플리카셋에 파드를  
생성

3. maxSurge: 1, maxUnavailable: 1

→ 기존 레플리카셋에 파드가 먼저 제거될수도 있고, 새로운 레플리카셋의  
파드가 새로 생성될수도 있음

→ 최대 존재 가능 pod: 4

→ 최소 존재 가능 pod: 2

-옵션

→ maxUnavailable

: 정의된값

→ maxSurge

: 전체 replicas수 + maxSurge 값



### 3.디플로이먼트의 롤링 업데이트 설정

15/23

```
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl apply -f vweb-strategies/vweb-rollingUpdate-v3
.yaml
deployment.apps/vweb configured
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl get po -l app=vweb
NAME                                READY   STATUS              RESTARTS   AGE
vweb-5c5cf94d68-dbdw6               1/1     Running             0           31s
vweb-5c5cf94d68-mvmrr               1/1     Running             0           31s
vweb-f6977cddd-d2t25                0/1     CrashLoopBackOff    1 (10s ago) 12s
vweb-f6977cddd-xrkft               0/1     CrashLoopBackOff    1 (10s ago) 12s
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl rollout status deploy/vweb
Waiting for deployment "vweb" rollout to finish: 2 out of 3 new replicas have been updated...
error: deployment "vweb" exceeded its progress deadline
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl get rs -l app=vweb
NAME                                DESIRED   CURRENT   READY   AGE
vweb-5c5cf94d68                     2         2         2       9m47s
vweb-f6977cddd                      2         2         0       9m28s
ubuntu@local:~/files/study/k8s/kiamol/ch09$ curl $(cat url.txt)
v2ubuntu@local:~/files/study/k8s/kiamol/ch09$
```

```
spec:
  replicas: 3
  selector:
    matchLabels:
      app: vweb
  minReadySeconds: 30
  progressDeadlineSeconds: 120
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 1
```

업데이트된 pod에 오류가 있으므로  
rollout이 더이상 진행되지 못함  
→ 쿠버네티스는 pod의 재시작간의  
시간 간격을 최대 5분까지 증가시킨다.

-옵션

→ minReadySeconds

: Pod이 Ready가 상태인 시점부터 시간을 카운트

: 지정한 시간동안 pod이 Ready 상태인지 기다린후에  
pod이 정상 상태이면 다음 업데이트를 진행

: pod이 Ready상태가 된후에 문제가 생기면 타이머 리셋

→progressDeadlineSeconds

: 시간을 초과할때 신규 파드의  
상태를 실패로 간주하는 타임아웃 시간

: 한번 Pod가 Available이 되면, 다음 Pod의 rollout Available  
에 적용되는 타이머가 초기화 된다.

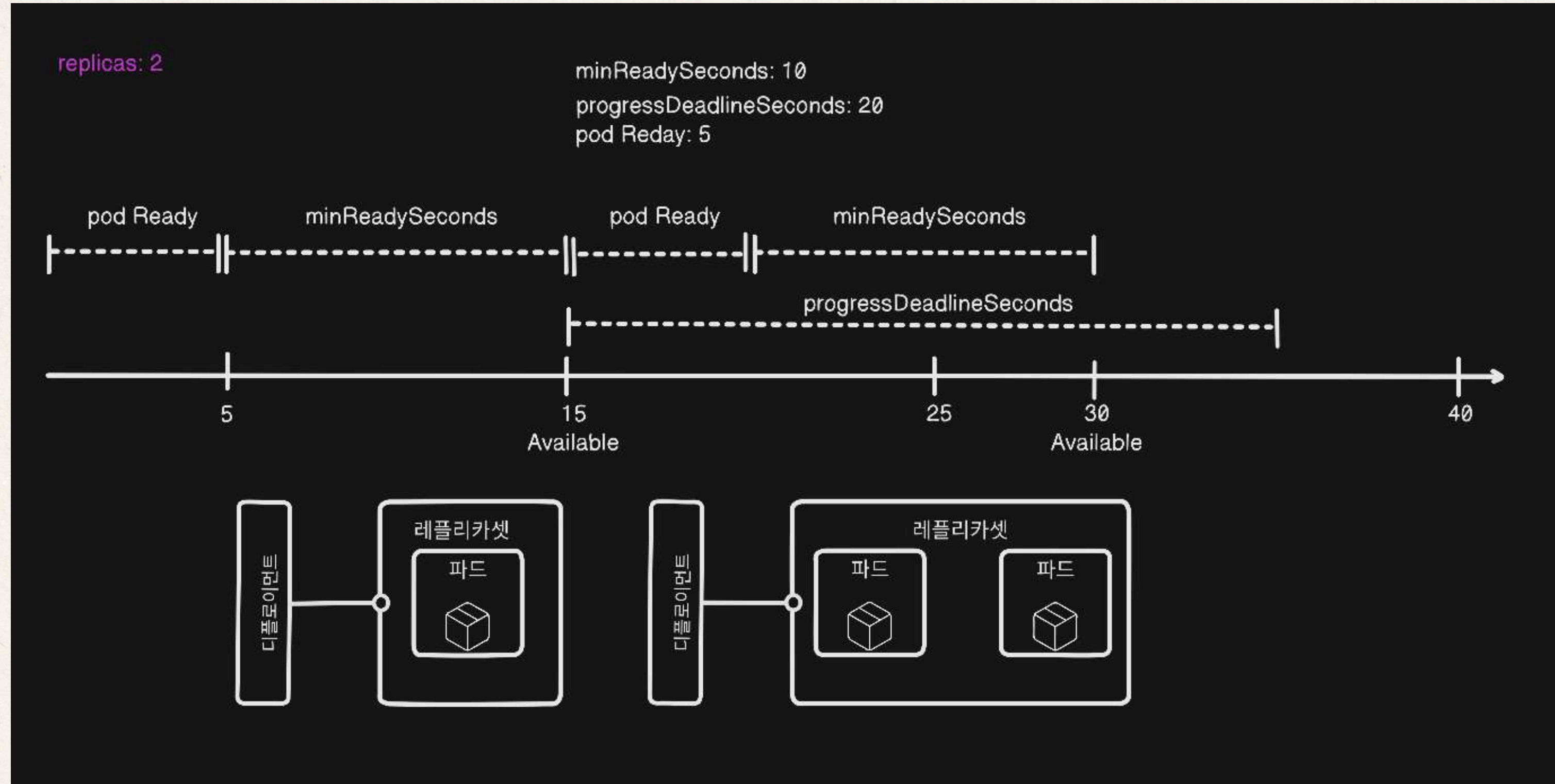
: Available 증가가 한번이라도 있었으면 실패가 아님



### 3.디플로이먼트의 롤링 업데이트 설정

-예시

16/23





## 4.데몬셋과 스테이트풀셋의 롤링 업데이트

17/23

```
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl delete all -l kiamol=ch09
deployment.apps "vweb" deleted from default namespace
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl apply -f todo-list/db/ -f to
do-list/web -f todo-list/proxy
configmap/todo-db-config created
configmap/todo-db-env created
configmap/todo-db-scripts created
secret/todo-db-secret created
service/todo-db created
statefulset.apps/todo-db created
configmap/todo-web-config created
secret/todo-web-secret created
service/todo-web created
deployment.apps/todo-web created
configmap/todo-proxy-configmap created
service/todo-proxy created
daemonset.apps/todo-proxy created
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl get svc todo-proxy -o jsonpa
th='http://{.status.loadBalancer.ingress[0].*}:8091'
http://192.168.0.157 VIP:8091ubuntu@local:~/files/study/k8s/kiamol/ch09$
```

KIAMOL To-Do List [This is TEST] [Source Code](#)

# TODO List

There is much still to do.

But there's always room to [add more](#).

Item	Date Added
test	Friday, 06 February 2026



```
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl apply -f todo-list/proxy/update/nginx-rollingUpdate.yaml
daemonset.apps/todo-proxy configured
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl get po -l app=todo-proxy --watch
NAME                READY   STATUS    RESTARTS   AGE
todo-proxy-k6nw2    1/1     Running   0           17s
^Cubuntu@local:~/files/study/k8s/kiamol/ch09$
```

단일 클러스터에서는 proxy 파트가 클러스터 전체에서 1개뿐 이므로 새로운 파드가 생성될 동안 서비스에 접근하지 못한다.

→ 클러스터에 노드가 여러개가 있을때는 maxUnavailable의 값을 조절해서 한번에 여러개의 파드를 업데이트 할수 있다.

- 데몬셋의 업데이트 전략
  - rollingupdate
  - OnDelete
  - : 파드의 정의를 수정해도 쿠버네티스가 자동으로 파드를 교체하지 않음
  - 1. 기존의 파드가 삭제된후 쿠버네티스가 새로운 파드를 생성



```
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl apply -f todo-list/db/update/todo-db-rollingUpdate-partition.yaml
statefulset.apps/todo-db configured
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl rollout status statefulset/todo-db
Waiting for partitioned roll out to finish: 0 out of 1 new pods have been updated...
Waiting for 1 pods to be ready...
Waiting for 1 pods to be ready...
Waiting for 1 pods to be ready...
partitioned roll out complete: 1 new pods have been updated...
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl get pods -l app=todo-db -o=custom-columns=NAME:.metadata.name,IMAGE:.spec.containers[0].image,START_TIME:.status.startTime
NAME          IMAGE      START_TIME
todo-db-0     <none>     2026-02-06T12:00:34Z
todo-db-1     <none>     2026-02-06T12:38:29Z
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl apply -f todo-list/web/update/todo-web-readonly.yaml
deployment.apps/todo-web configured
```

```
updateStrategy:
  type: RollingUpdate
  rollingUpdate:
    partition: 1 # only updates Pod 1
```

partition은 업데이트 되지 않고 기존의 버전을 유지하는 pod개수를 의미한다.

스테이트풀셋은 maxSurge나 maxUnavailable은 사용할수 없고 partition 값을 사용하여 전체 업데이트해야 하는 파드의 비율을 설정할수 있다.

-스테이트 풀셋의 업데이트 전략  
→ 지정한 순서대로 업데이트  
: 마지막 파드부터 시작해서  
첫번째 파드까지 진행



## 4.데몬셋과 스테이트풀셋의 롤링 업데이트

20/23

```
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl apply -f todo-list/db/update/todo-db-rollingUpdate.yaml
statefulset.apps/todo-db configured
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl rollout status sstatefulset/todo-db
error: the server doesn't have a resource type "sstatefulset"
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl rollout status statefulset/todo-db
waiting for statefulset rolling update to complete 1 pods at revision todo-db-669f6cf649...
Waiting for 1 pods to be ready...
Waiting for 1 pods to be ready...
Waiting for 1 pods to be ready...
statefulset rolling update complete 2 pods at revision todo-db-669f6cf649...
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl get pods -l app=todo-db -o=custom-columns=NAME:.metadata.name,IMAGE:.spec.containers[0].image,START_TIME:.status.startTime
NAME          IMAGE          START_TIME
todo-db-0      postgres:11.8-alpine  2026-02-06T12:45:59Z
todo-db-1      postgres:11.8-alpine  2026-02-06T12:38:29Z
ubuntu@local:~/files/study/k8s/kiamol/ch09$ kubectl apply -f todo-list/web/todo-web.yaml
deployment.apps/todo-web configured
```

KIAMOL To-Do List

[This is TEST]

Source Code

Home

New item

To-do list

Diagnostics

# TODO List

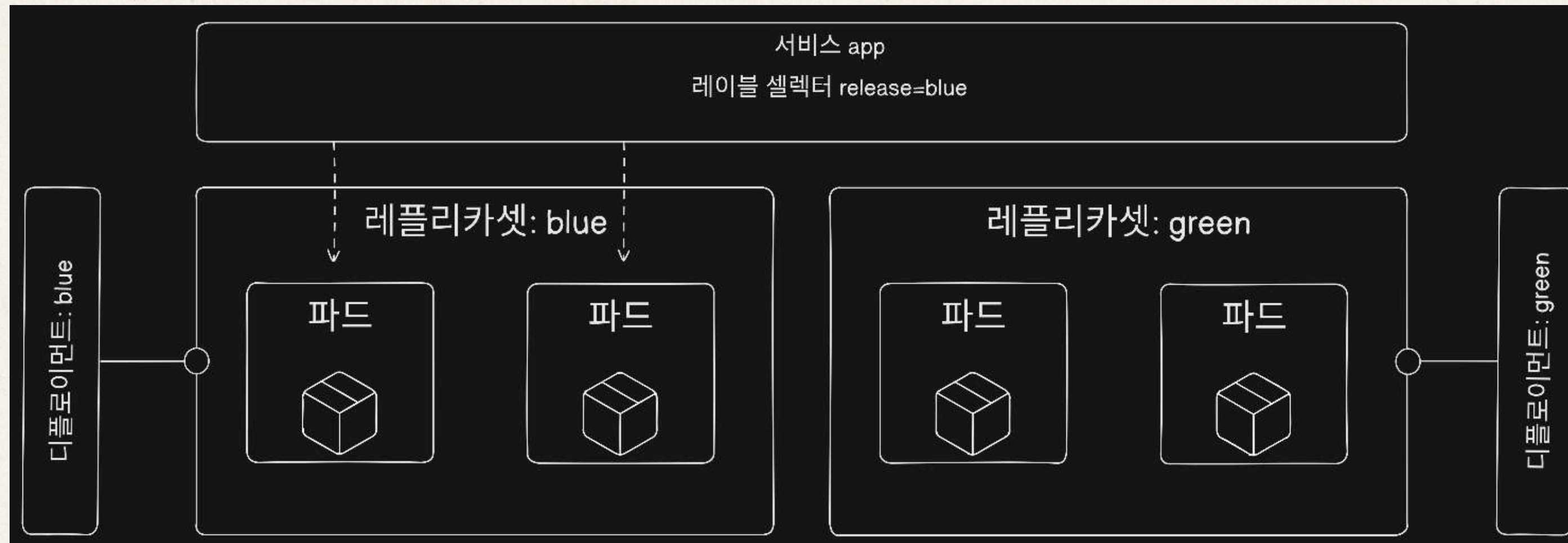
There is much still to do.

But there's always room to [add more](#).

Item	Date Added
test	Friday, 06 February 2026
† † †	Friday, 06 February 2026
test	Friday, 06 February 2026

디플로이먼트와 데몬셋, 스테이트풀셋은 모두 기본적으로 롤링 업데이트를 사용  
→ 여러가지 버전이 동시에 동작할수 없는 제약이 있음





롤아웃 도중에 사용자가 새로운 파드가 아닌 기존의 파드로 부터 응답을 받게 되면 UX 문제가 일어날수 있다.  
 → 웹 애플리케이션 쿠키, 고급 트래픽 라우팅 등을 사용하여 해결 가능,  
 혹은 고급 트래픽을 사용하여 사용자가 새 버전에 남도록 할수 있음

- 특징
- 장점
- :롤링 업데이트에 비하면 비교적 간단
- 단점
- :컴퓨팅 리소스를 많이 소모



```
spec:
  replicas: 4
  selector:
    matchLabels:
      app: vweb
      version: v1
```

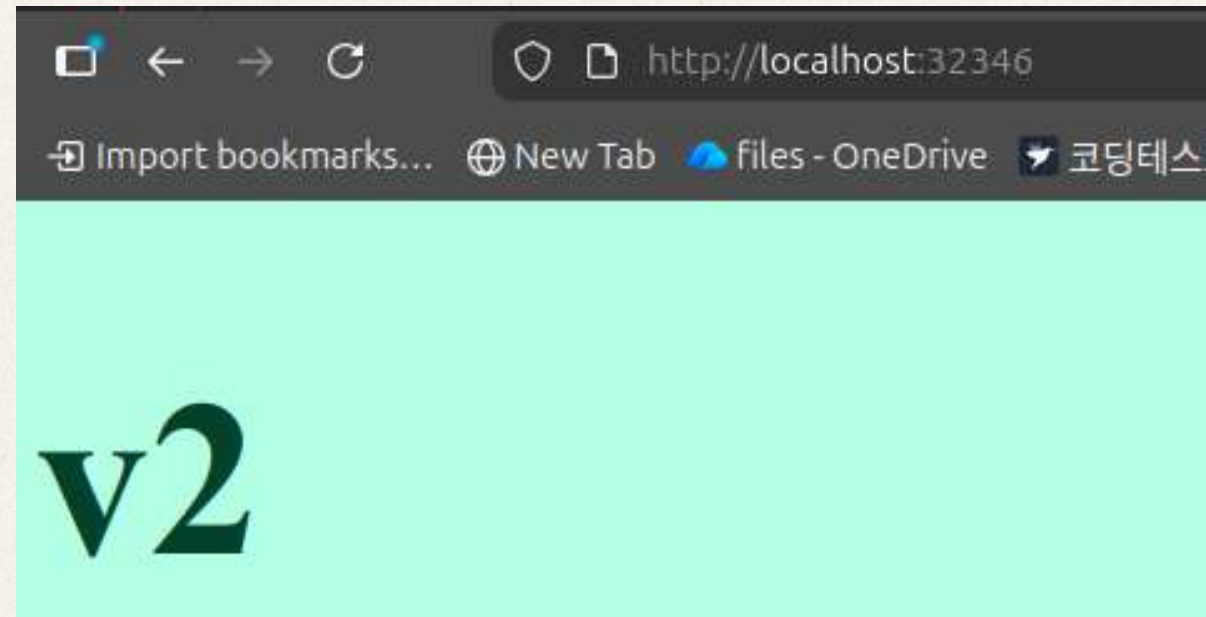
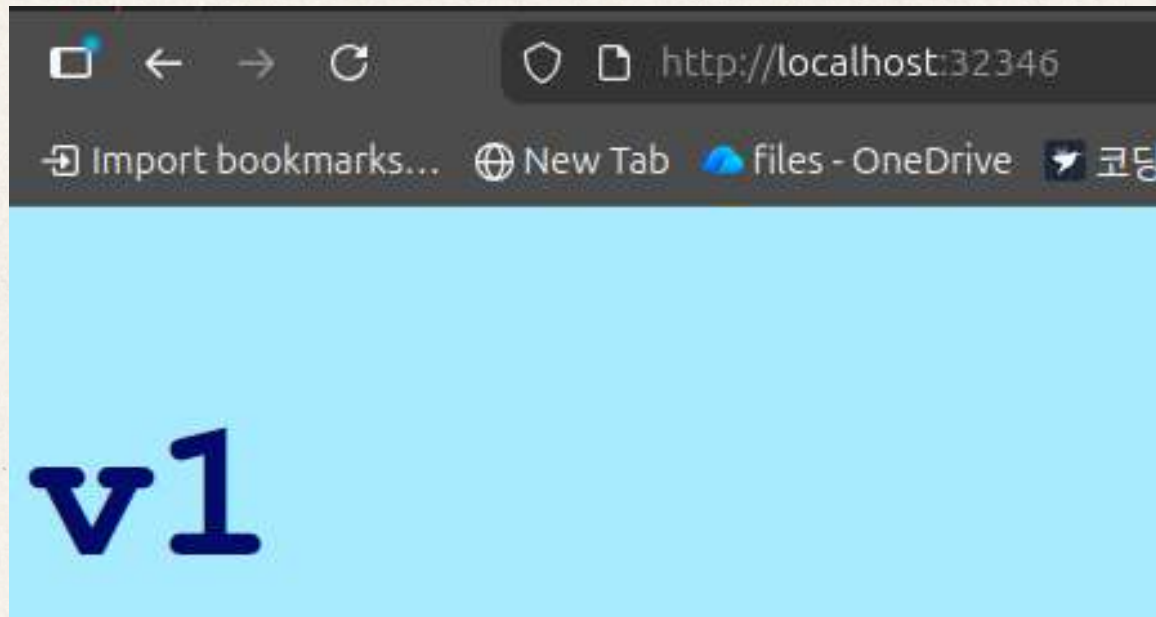
v1

```
spec:
  ports:
    - port: 8090
      targetPort: http
  selector:
    app: vweb
    version: v1
```

서비스

```
spec:
  replicas: 4
  selector:
    matchLabels:
      app: vweb
      version: v2
```

v2





# Thank you

---