8th Week

여덟 번째 뵙겠습니다 ?!

▷ 잠시만 기다렸다가 30분 되면 시작하겠습니다~^^

▷ 계속 함께 해주셔서 고맙습니다~!!!!

- 복 받으실거에요~~~!!!

▷ Camera는 가급적 켜 주시면 대단히 감사하겠습니다 !!!

- 너무 부끄러우면 Snap Camera를 사용하시는 것 까지는~ ^^

▷ 오늘 수업 자료는 아래 링크에서 다운로드 받으실 수 있어요.

- https://github.com/whatwant-school/kubernetes

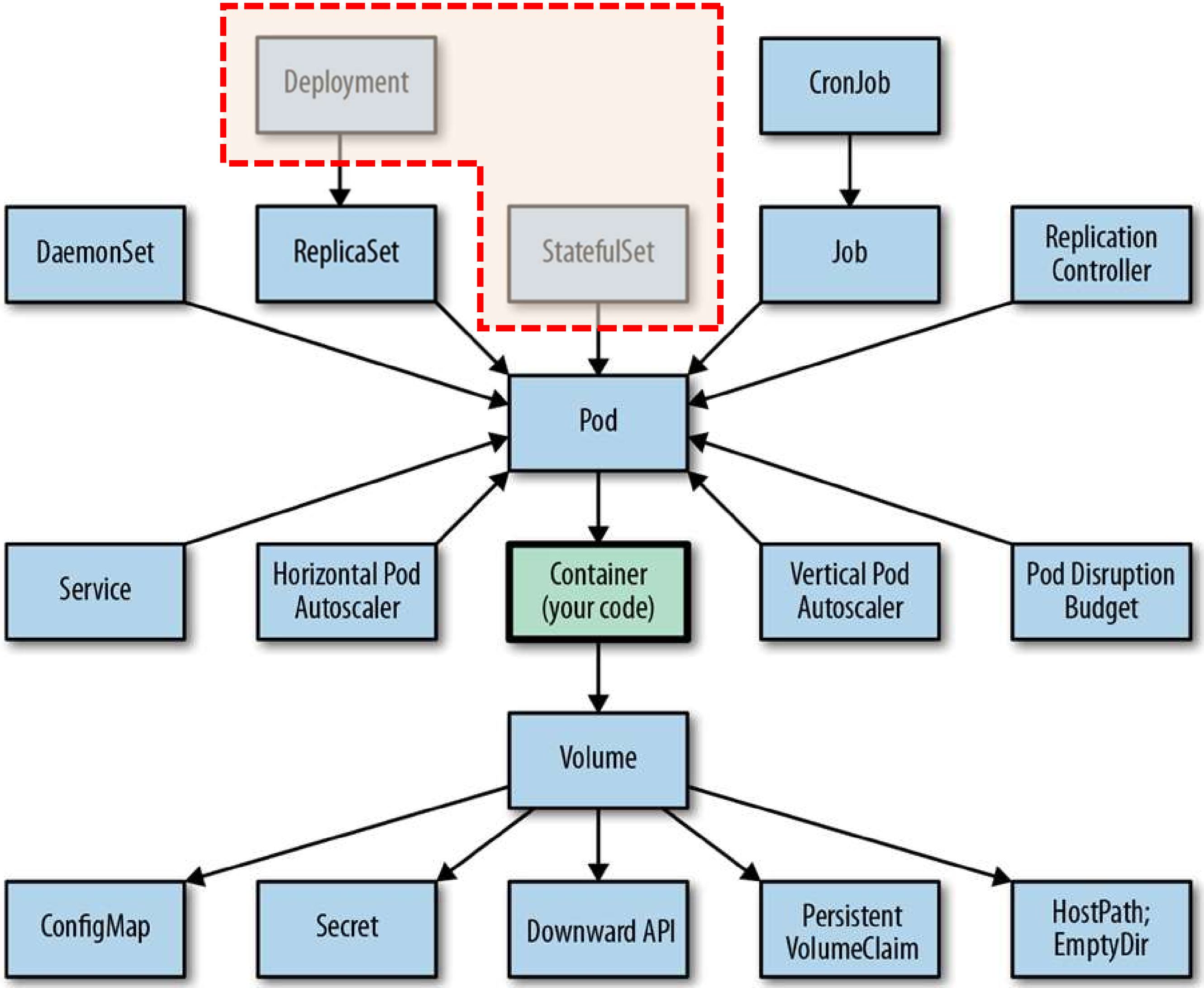
///

지난 수업 기억 나시나요?

https://kahoot.it/

///

Deployment
StatefulSet



※ 참고 : https://www.oreilly.com/library/view/kubernetes-patterns/9781492050278/ch01.html

///

Kubernetes

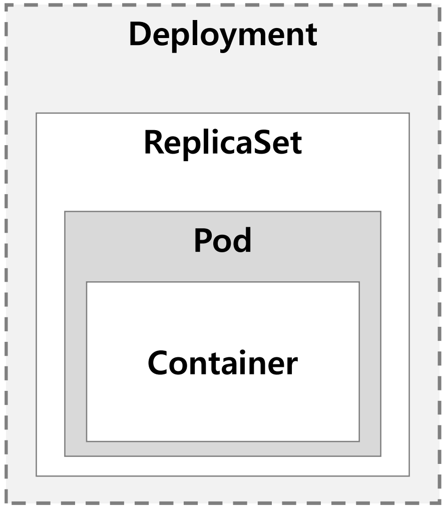
Deployment

Why Deployment

- 디플로이먼트(Deployment)는 Pod와 ReplicaSet에 대한 선언적 업데이트를 제공

  . 새로운 ReplicaSet을 생성하는 Deployment를 정의하거나 기존 Deployment를 제거하고, 모든 리소스를 새 Deployment에 적용할 수 있다.

- Deployment가 소유하는 ReplicaSet은 관리하지 않아야 한다.



※ 참고 : https://kubernetes.io/ko/docs/concepts/workloads/controllers/deployment/

YAML

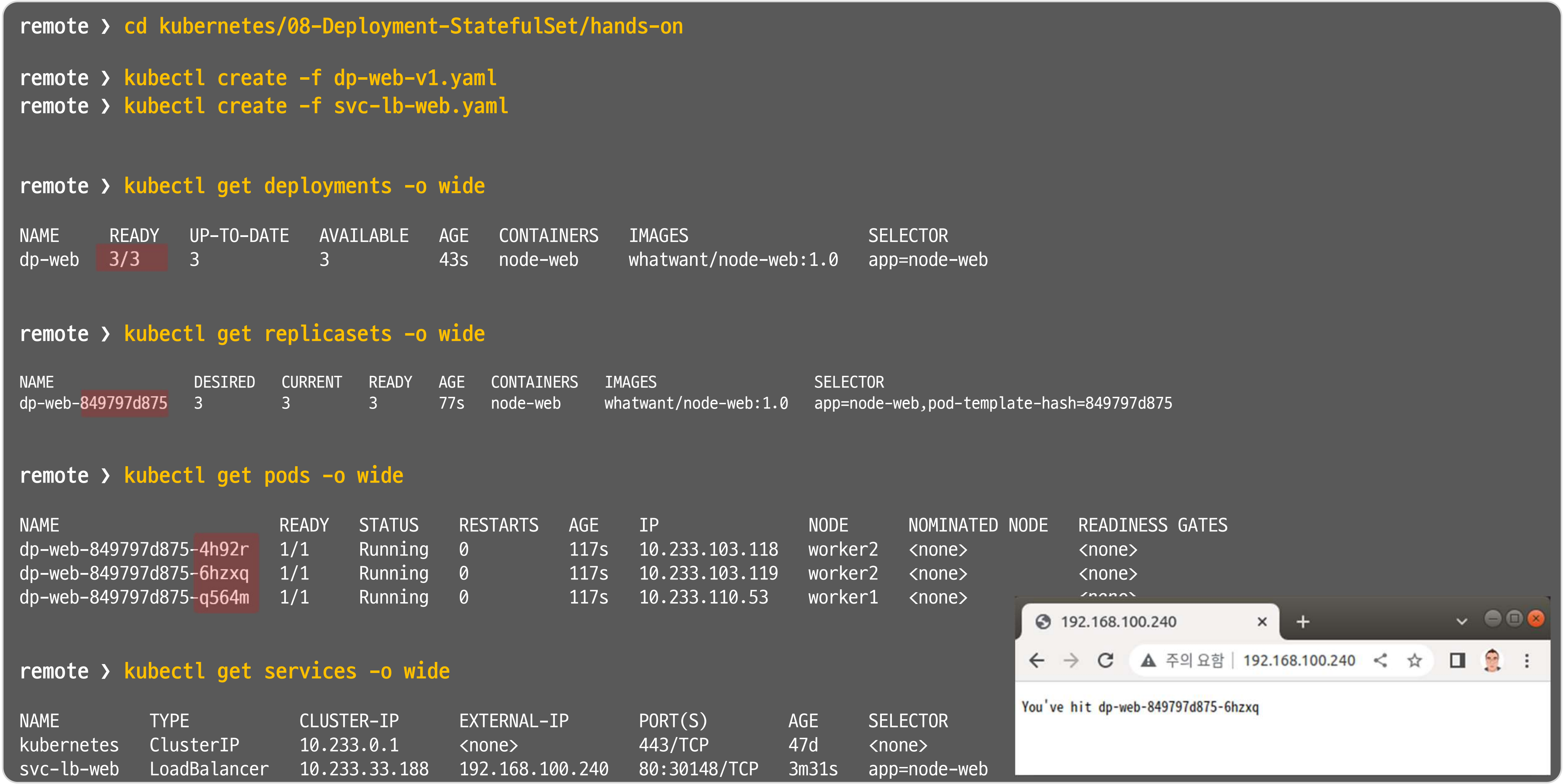
dp-web-v1.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: dp-web

spec:
  replicas: 3

  selector:
    matchLabels:
      app: node-web

  template:
    metadata:
      name: node-web
      labels:
        app: node-web

    spec:
      containers:
      - image: whatwant/node-web:1.0
        name: node-web
        ports:
        - containerPort: 8080
          protocol: TCP
        imagePullPolicy: Always
```

svc-lb-web.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: svc-lb-web

spec:
  type: LoadBalancer

  ports:
  - name: http
    port: 80
    protocol: TCP
    targetPort: 8080

  selector:
    app: node-web
```

Execute



```
remote ❯ cd kubernetes/08-Deployment-StatefulSet/hands-on

remote ❯ kubectl create -f dp-web-v1.yaml
remote ❯ kubectl create -f svc-lb-web.yaml


remote ❯ kubectl get deployments -o wide

NAME      READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES                  SELECTOR
dp-web    3/3     3            3           43s   node-web     whatwant/node-web:1.0   app=node-web


remote ❯ kubectl get replicasets -o wide

NAME                DESIRED   CURRENT   READY   AGE   CONTAINERS   IMAGES                  SELECTOR
dp-web-849797d875   3         3         3       77s   node-web     whatwant/node-web:1.0   app=node-web,pod-template-hash=849797d875


remote ❯ kubectl get pods -o wide

NAME                      READY   STATUS    RESTARTS   AGE    IP              NODE      NOMINATED NODE   READINESS GATES
dp-web-849797d875-4h92r   1/1     Running   0          117s   10.233.103.118  worker2   <none>           <none>
dp-web-849797d875-6hzxq   1/1     Running   0          117s   10.233.103.119  worker2   <none>           <none>
dp-web-849797d875-q564m   1/1     Running   0          117s   10.233.110.53   worker1   <none>           <none>


remote ❯ kubectl get services -o wide

NAME         TYPE           CLUSTER-IP     EXTERNAL-IP       PORT(S)        AGE     SELECTOR
kubernetes   ClusterIP      10.233.0.1     <none>            443/TCP        47d     <none>
svc-lb-web   LoadBalancer   10.233.33.188  192.168.100.240   80:30148/TCP   3m31s   app=node-web
```

```
You've hit dp-web-849797d875-6hzxq
```

///

Change Pods

How to

- 버전 업그레이드 또는 Application 변경 등의 작업을 할 때 선택할 수 있는 방법 3가지

#1. Deleting old pods and replacing them with new ones

(기존 Pods를 삭제하고 새로운 Pods로 교체)

#2. Switching from the old to the new version at once (Blue-Green Deployment)

(새로운 버전으로 한 번에 전환)

#3. Rolling update

(롤링 업데이트 / 무중단 배포)

#1. Deleting old pods and replacing them with new ones

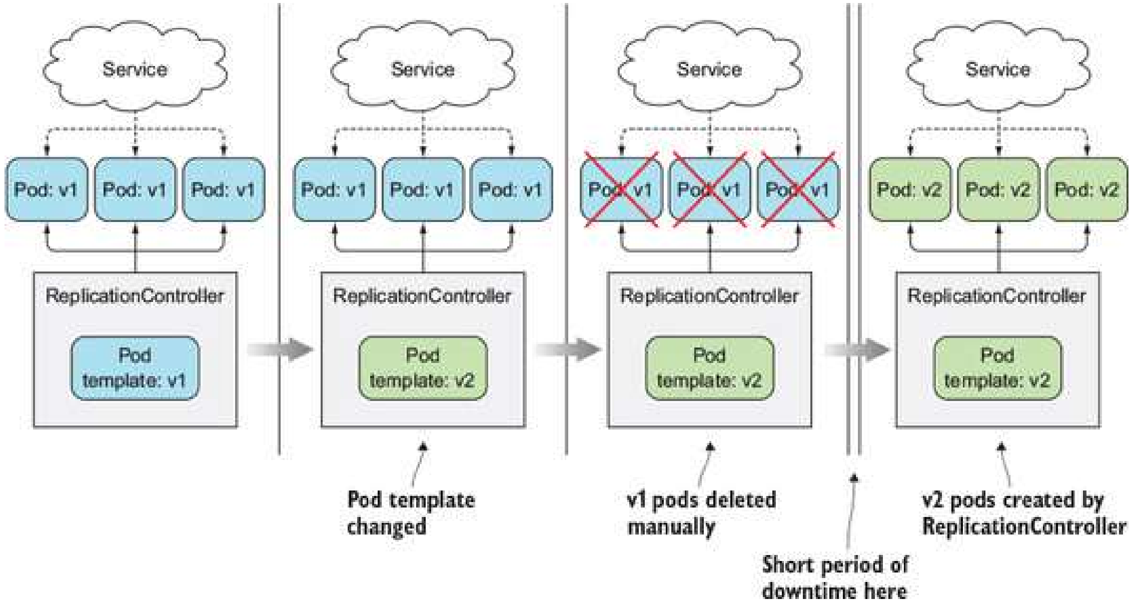
- Application의 변경(like version-up)이 필요한 경우 손쉽게 적용 가능

  ① Template에서 새로운 version으로 변경 작성

  ② Pod 삭제

  ③ 변경된 Template 기준으로 새로운 Pod 자동 생성

- 짧은 시간의 다운타임을 허용할 수 있다면, 가장 간단한 방법



※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-9/22

#2. Switching from the old to the new version at once

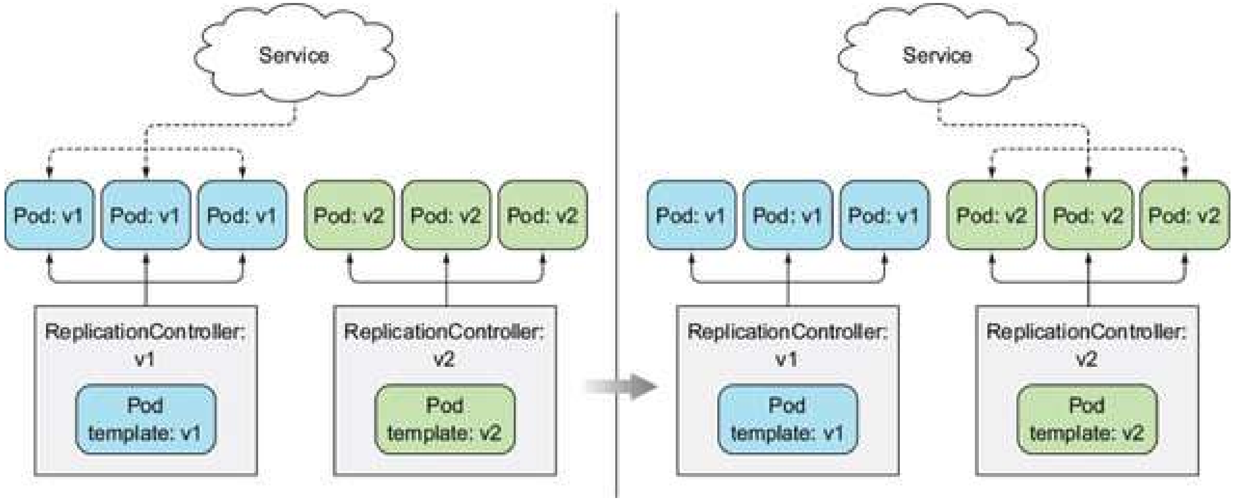
- 다운타임이 발생하지 않고 한 번에 여러 version의 application이 실행되는 것을 지원하는 경우

① 새로운 version의 Template으로 신규 Pod 생성, 기존 version은 지속 서비스 中

② 한 번에 Service를 신규 Pod를 바라보도록 전환

③ 전환 완료되면, 기존 Pod 삭제

= Blue-Green Deployment

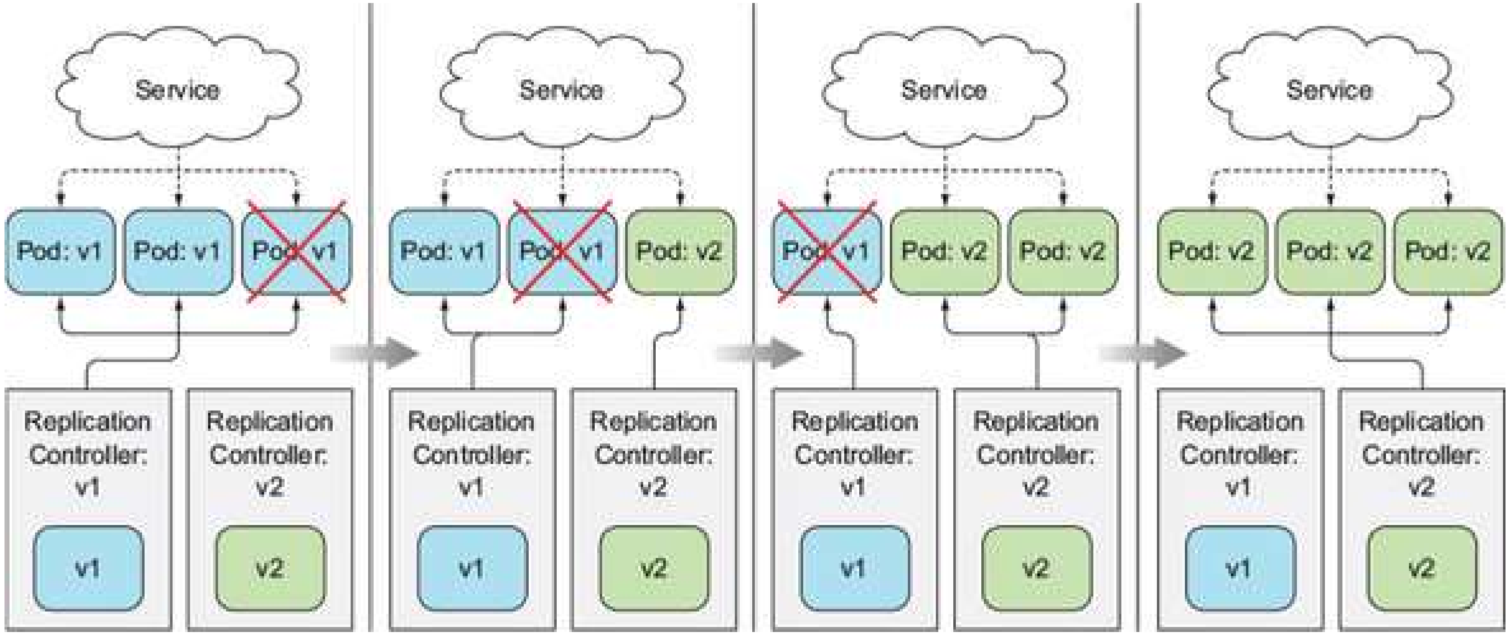


※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-9/29

#3. Rolling update

- Pod를 단계별로 교체

  . 수작업으로 진행하기에는 상당히 번거롭고, 실수할 여지가 많음 → kubernetes에서 제공해주는 여러 방법 존재


Service
Pod: v1
Pod: v1
Pod: v1
Replication Controller: v1
v1
Replication Controller: v2
v2
Service
Pod: v1
Pod: v1
Pod: v2
Replication Controller: v1
v1
Replication Controller: v2
v2
Service
Pod: v1
Pod: v2
Pod: v2
Replication Controller: v1
v1
Replication Controller: v2
v2
Service
Pod: v2
Pod: v2
Pod: v2
Replication Controller: v1
v1
Replication Controller: v2
v2

※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-9/32

///

Rolling Update

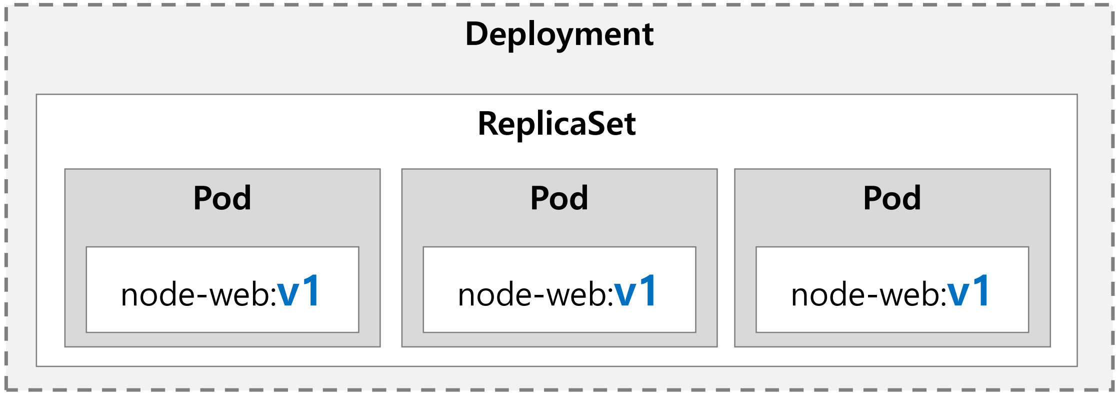
Kubernetes 리소스 수정 = Deployment 수정 방법

명령어	설명	예시
kubectl edit	기본 편집기로 오브젝트의 manifest를 오픈한다. 변경 후 파일을 저장하고 편집기를 종료하면 오브젝트가 업데이트 된다.	kubectl edit deployment node-web
kubectl patch	오브젝트의 개별 속성을 수정한다.	kubectl patch deployment web -p '{"spec": {"minReadySeconds": 10}}'
kubectl apply	전체 YAML/JSON 파일의 속성 값을 적용해 오브젝트를 수정한다. 파일에는 리소스의 전체 정의가 포함되어야 한다.	kubectl apply -f node-web-v2.yaml
kubectl replace	YAML/JSON 파일로 오브젝트를 새 것으로 교체한다. 오브젝트가 없을 때 실행하면 오류를 출력한다.	kubectl replace -f node-web-v2.yaml
kubectl set image	Pod, Deployment, ReplicaSet, DaemonSet, Job에 정의된 컨테이너 이미지를 변경한다.	kubectl set image deployment node-web nodejs=ww/node-web:v2.0

Ready (Status)

- 앞에서 생성한 Deployment & Service 적용된 상황

```
remote ❯ sh -c 'while true; do curl http://192.168.100.240; sleep 2; done'

You've hit dp-web-849797d875-4h92r
You've hit dp-web-849797d875-4h92r
You've hit dp-web-849797d875-4h92r
You've hit dp-web-849797d875-q564m
You've hit dp-web-849797d875-6hzxq
You've hit dp-web-849797d875-4h92r
You've hit dp-web-849797d875-q564m
You've hit dp-web-849797d875-6hzxq
You've hit dp-web-849797d875-4h92r
You've hit dp-web-849797d875-4h92r
^C
```

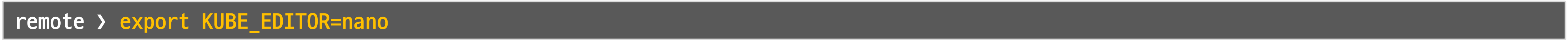


///

[Tip] Editor

- kubectl 기본 편집기를 변경 해보자 (default = vi)

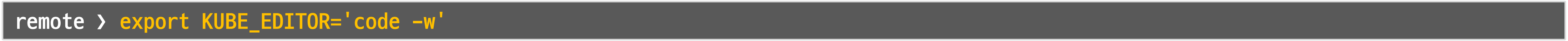
  . 개인적인 취향으로 일단 `nano` editor인 경우 아래와 같이 하면 된다.



```
remote ❯ export KUBE_EDITOR=nano
```

- Visual Studio Code를 기본 에디터로 하고 싶다면?!

  . 변경 사항을 kubectl에서 watch 할 수 있도록 '-w' 옵션도 추가해주면 좋다.

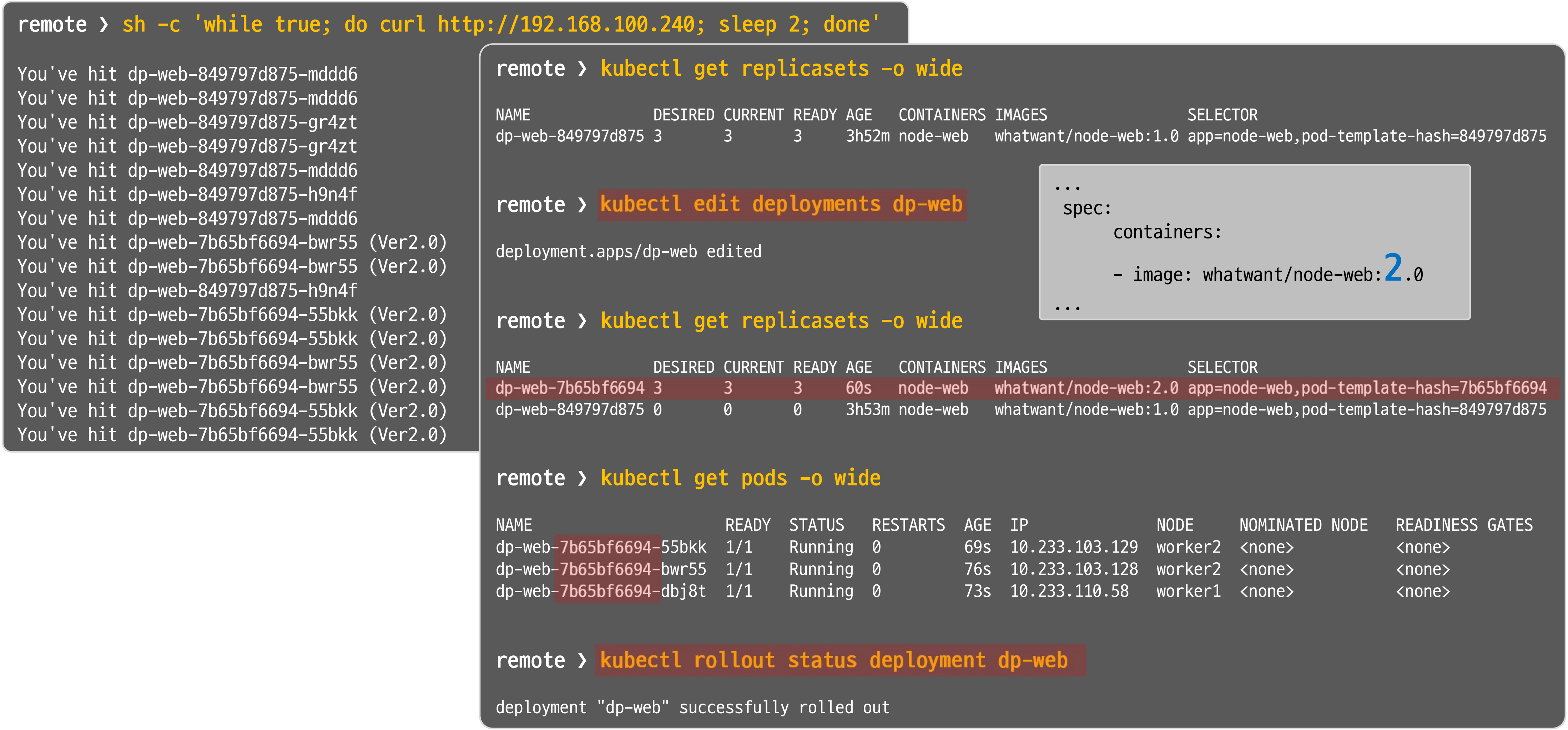


```
remote ❯ export KUBE_EDITOR='code -w'
```

///

edit - execute

- 모니터링을 걸어 놓고, `kubectl edit` 실행



```
remote ❯ sh -c 'while true; do curl http://192.168.100.240; sleep 2; done'

You've hit dp-web-849797d875-mddd6
You've hit dp-web-849797d875-mddd6
You've hit dp-web-849797d875-gr4zt
You've hit dp-web-849797d875-gr4zt
You've hit dp-web-849797d875-mddd6
You've hit dp-web-849797d875-h9n4f
You've hit dp-web-849797d875-mddd6
You've hit dp-web-7b65bf6694-bwr55 (Ver2.0)
You've hit dp-web-7b65bf6694-bwr55 (Ver2.0)
You've hit dp-web-849797d875-h9n4f
You've hit dp-web-7b65bf6694-55bkk (Ver2.0)
You've hit dp-web-7b65bf6694-55bkk (Ver2.0)
You've hit dp-web-7b65bf6694-bwr55 (Ver2.0)
You've hit dp-web-7b65bf6694-bwr55 (Ver2.0)
You've hit dp-web-7b65bf6694-55bkk (Ver2.0)
You've hit dp-web-7b65bf6694-55bkk (Ver2.0)
```

```
remote ❯ kubectl get replicasets -o wide

NAME              DESIRED CURRENT READY AGE   CONTAINERS IMAGES                SELECTOR
dp-web-849797d875 3       3       3     3h52m node-web   whatwant/node-web:1.0 app=node-web,pod-template-hash=849797d875

remote ❯ kubectl edit deployments dp-web

deployment.apps/dp-web edited

remote ❯ kubectl get replicasets -o wide

NAME              DESIRED CURRENT READY AGE   CONTAINERS IMAGES                SELECTOR
dp-web-7b65bf6694 3       3       3     60s   node-web   whatwant/node-web:2.0 app=node-web,pod-template-hash=7b65bf6694
dp-web-849797d875 0       0       0     3h53m node-web   whatwant/node-web:1.0 app=node-web,pod-template-hash=849797d875

remote ❯ kubectl get pods -o wide

NAME                     READY  STATUS   RESTARTS  AGE  IP              NODE     NOMINATED NODE   READINESS GATES
dp-web-7b65bf6694-55bkk  1/1    Running  0         69s  10.233.103.129  worker2  <none>           <none>
dp-web-7b65bf6694-bwr55  1/1    Running  0         76s  10.233.103.128  worker2  <none>           <none>
dp-web-7b65bf6694-dbj8t  1/1    Running  0         73s  10.233.110.58   worker1  <none>           <none>

remote ❯ kubectl rollout status deployment dp-web

deployment "dp-web" successfully rolled out
```

```
...
 spec:
      containers:
      - image: whatwant/node-web:2.0
...
```

///

Re - Ready

- 깔끔한 (?) 실습을 위해 Deployment를 삭제 후 다시 생성하자

```
remote ❯ kubectl delete deployments dp-web

remote ❯ kubectl create -f dp-web-v1.yaml
```

///

YAML

dp-web-v1.yaml

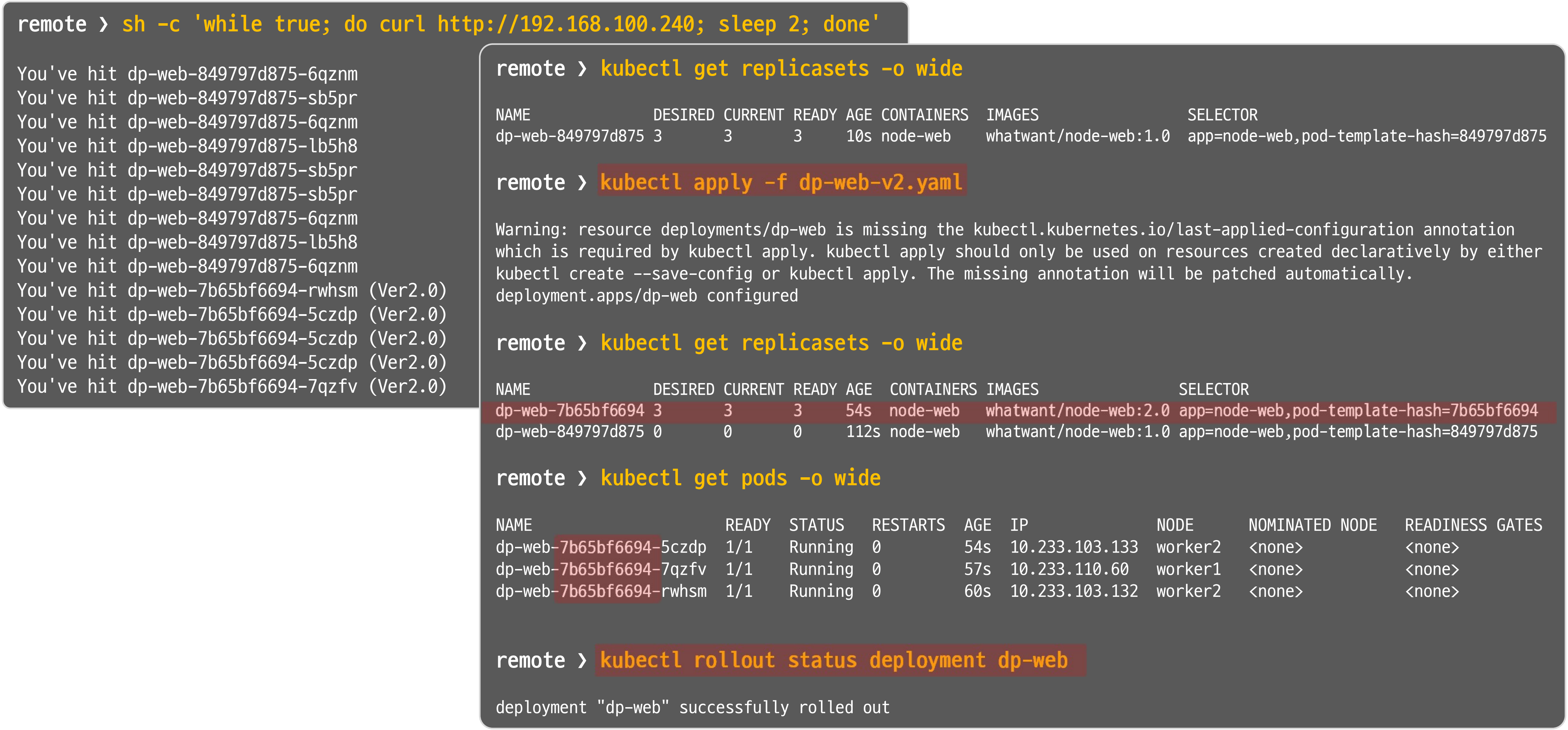
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: dp-web

spec:
  replicas: 3

  selector:
    matchLabels:
      app: node-web

  template:
    metadata:
      name: node-web
      labels:
        app: node-web

    spec:
      containers:
      - image: whatwant/node-web:1.0
        name: node-web
        ports:
        - containerPort: 8080
          protocol: TCP
        imagePullPolicy: Always
```

dp-web-v2.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: dp-web

spec:
  replicas: 3

  selector:
    matchLabels:
      app: node-web

  template:
    metadata:
      name: node-web
      labels:
        app: node-web

    spec:
      containers:
      - image: whatwant/node-web:2.0
        name: node-web
        ports:
        - containerPort: 8080
          protocol: TCP
        imagePullPolicy: Always
```

apply - execute

- 모니터링을 걸어 놓고, `kubectl edit` 실행



```
remote ❯ sh -c 'while true; do curl http://192.168.100.240; sleep 2; done'

You've hit dp-web-849797d875-6qznm
You've hit dp-web-849797d875-sb5pr
You've hit dp-web-849797d875-6qznm
You've hit dp-web-849797d875-lb5h8
You've hit dp-web-849797d875-sb5pr
You've hit dp-web-849797d875-sb5pr
You've hit dp-web-849797d875-6qznm
You've hit dp-web-849797d875-lb5h8
You've hit dp-web-849797d875-6qznm
You've hit dp-web-7b65bf6694-rwhsm (Ver2.0)
You've hit dp-web-7b65bf6694-5czdp (Ver2.0)
You've hit dp-web-7b65bf6694-5czdp (Ver2.0)
You've hit dp-web-7b65bf6694-5czdp (Ver2.0)
You've hit dp-web-7b65bf6694-7qzfv (Ver2.0)
```

```
remote ❯ kubectl get replicasets -o wide

NAME              DESIRED CURRENT READY AGE CONTAINERS  IMAGES                 SELECTOR
dp-web-849797d875 3       3       3     10s node-web    whatwant/node-web:1.0  app=node-web,pod-template-hash=849797d875

remote ❯ kubectl apply -f dp-web-v2.yaml

Warning: resource deployments/dp-web is missing the kubectl.kubernetes.io/last-applied-configuration annotation
which is required by kubectl apply. kubectl apply should only be used on resources created declaratively by either
kubectl create --save-config or kubectl apply. The missing annotation will be patched automatically.
deployment.apps/dp-web configured

remote ❯ kubectl get replicasets -o wide

NAME              DESIRED CURRENT READY AGE  CONTAINERS IMAGES                SELECTOR
dp-web-7b65bf6694 3       3       3     54s  node-web   whatwant/node-web:2.0 app=node-web,pod-template-hash=7b65bf6694
dp-web-849797d875 0       0       0     112s node-web   whatwant/node-web:1.0 app=node-web,pod-template-hash=849797d875

remote ❯ kubectl get pods -o wide

NAME                    READY  STATUS   RESTARTS   AGE  IP              NODE     NOMINATED NODE   READINESS GATES
dp-web-7b65bf6694-5czdp 1/1    Running  0          54s  10.233.103.133  worker2  <none>           <none>
dp-web-7b65bf6694-7qzfv 1/1    Running  0          57s  10.233.110.60   worker1  <none>           <none>
dp-web-7b65bf6694-rwhsm 1/1    Running  0          60s  10.233.103.132  worker2  <none>           <none>


remote ❯ kubectl rollout status deployment dp-web

deployment "dp-web" successfully rolled out
```

///

Re - Ready

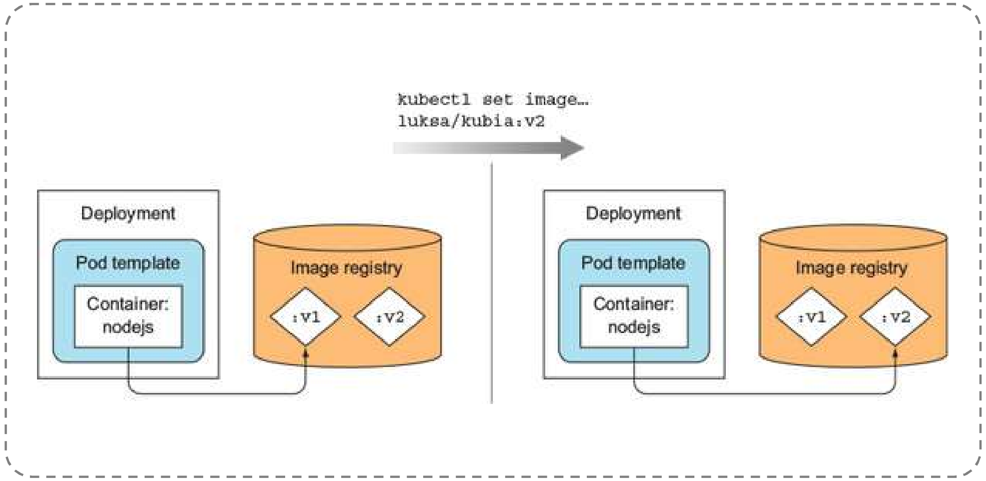
- 깔끔한 (?) 실습을 위해 Deployment를 삭제 후 다시 생성하자

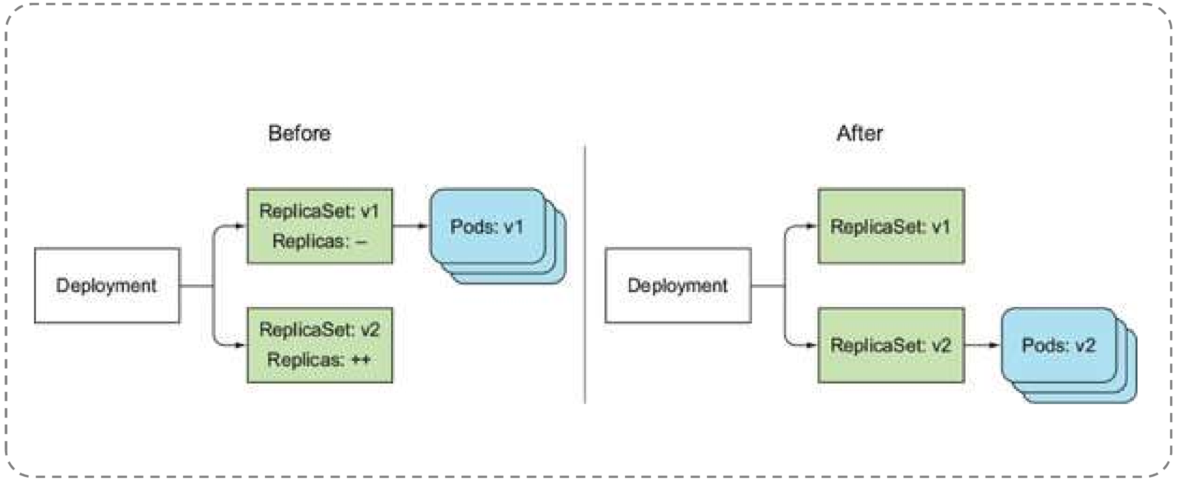
```
remote ❯ kubectl delete deployments dp-web

remote ❯ kubectl create -f dp-web-v1.yaml
```

///

set image - overview

- Container image의 버전을 업데이트하거나 변경할 때 사용





`kubectl set image`를 통해 image 변경 실행

`kubectl set image`를 실행했을 때 내부를 살펴보면

기존 Pod의 image를 변경하는 것이 아니라

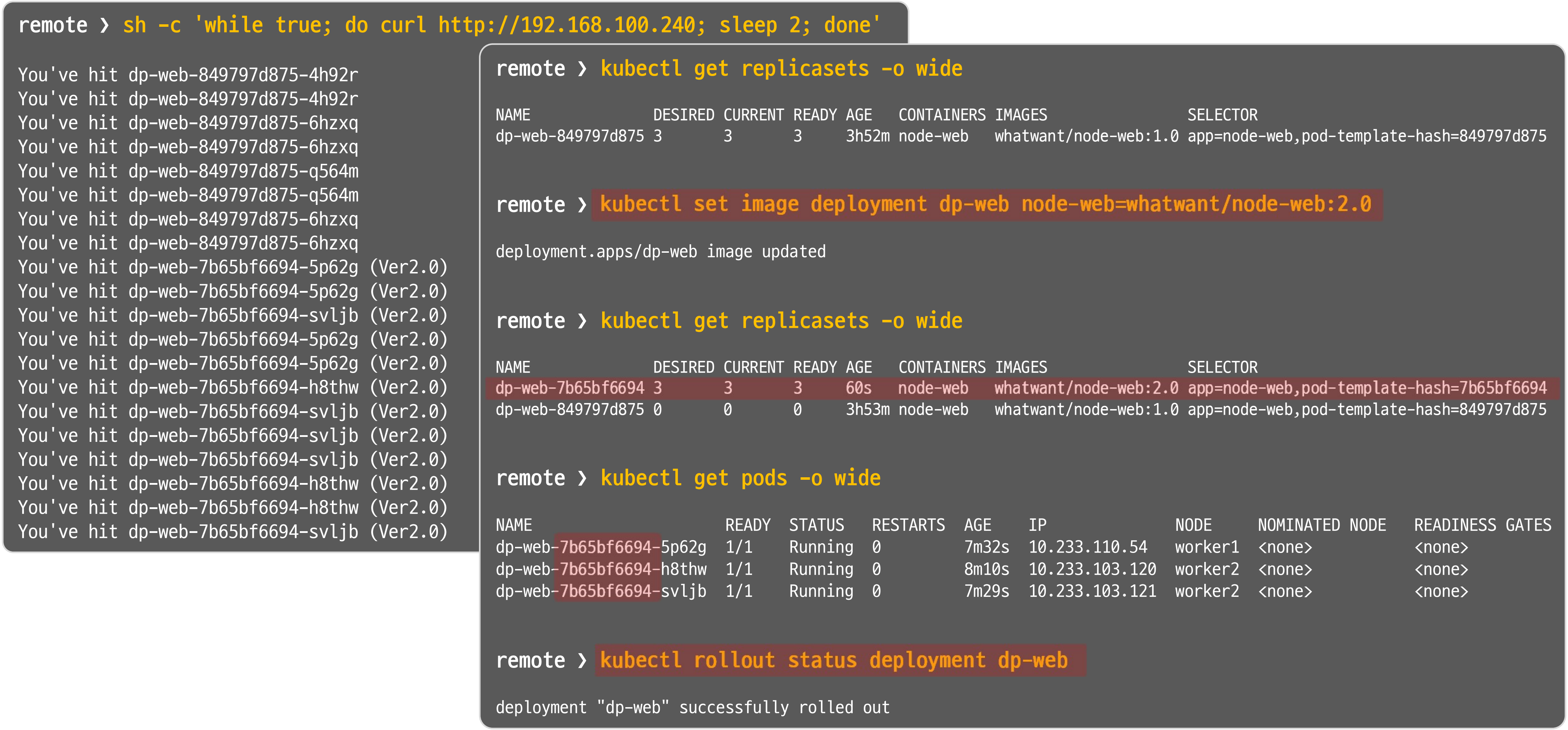
새로운 ReplicaSet을 생성해서 새로운 Pod를 생성하는 것을 볼 수 있다.

※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-9/154

※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-9/166

set image - execute

- 모니터링을 걸어 놓고, `kubectl set image` 실행



```
remote ❯ sh -c 'while true; do curl http://192.168.100.240; sleep 2; done'

You've hit dp-web-849797d875-4h92r
You've hit dp-web-849797d875-4h92r
You've hit dp-web-849797d875-6hzxq
You've hit dp-web-849797d875-6hzxq
You've hit dp-web-849797d875-q564m
You've hit dp-web-849797d875-q564m
You've hit dp-web-849797d875-6hzxq
You've hit dp-web-849797d875-6hzxq
You've hit dp-web-7b65bf6694-5p62g (Ver2.0)
You've hit dp-web-7b65bf6694-5p62g (Ver2.0)
You've hit dp-web-7b65bf6694-svljb (Ver2.0)
You've hit dp-web-7b65bf6694-5p62g (Ver2.0)
You've hit dp-web-7b65bf6694-5p62g (Ver2.0)
You've hit dp-web-7b65bf6694-h8thw (Ver2.0)
You've hit dp-web-7b65bf6694-svljb (Ver2.0)
You've hit dp-web-7b65bf6694-svljb (Ver2.0)
You've hit dp-web-7b65bf6694-svljb (Ver2.0)
You've hit dp-web-7b65bf6694-h8thw (Ver2.0)
You've hit dp-web-7b65bf6694-h8thw (Ver2.0)
You've hit dp-web-7b65bf6694-svljb (Ver2.0)
```

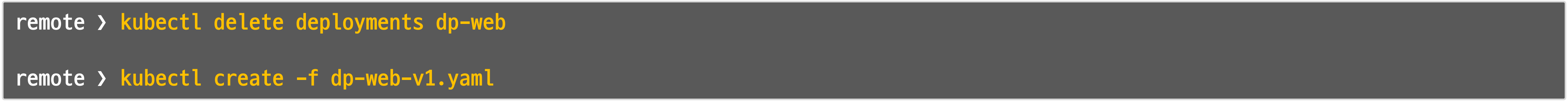
```
remote ❯ kubectl get replicasets -o wide

NAME              DESIRED CURRENT READY AGE   CONTAINERS IMAGES                SELECTOR
dp-web-849797d875 3       3       3     3h52m node-web   whatwant/node-web:1.0 app=node-web,pod-template-hash=849797d875

remote ❯ kubectl set image deployment dp-web node-web=whatwant/node-web:2.0

deployment.apps/dp-web image updated

remote ❯ kubectl get replicasets -o wide

NAME              DESIRED CURRENT READY AGE   CONTAINERS IMAGES                SELECTOR
dp-web-7b65bf6694 3       3       3     60s   node-web   whatwant/node-web:2.0 app=node-web,pod-template-hash=7b65bf6694
dp-web-849797d875 0       0       0     3h53m node-web   whatwant/node-web:1.0 app=node-web,pod-template-hash=849797d875

remote ❯ kubectl get pods -o wide

NAME                     READY  STATUS   RESTARTS  AGE   IP             NODE     NOMINATED NODE   READINESS GATES
dp-web-7b65bf6694-5p62g  1/1    Running  0         7m32s 10.233.110.54  worker1  <none>           <none>
dp-web-7b65bf6694-h8thw  1/1    Running  0         8m10s 10.233.103.120 worker2  <none>           <none>
dp-web-7b65bf6694-svljb  1/1    Running  0         7m29s 10.233.103.121 worker2  <none>           <none>

remote ❯ kubectl rollout status deployment dp-web

deployment "dp-web" successfully rolled out
```

///

rollout

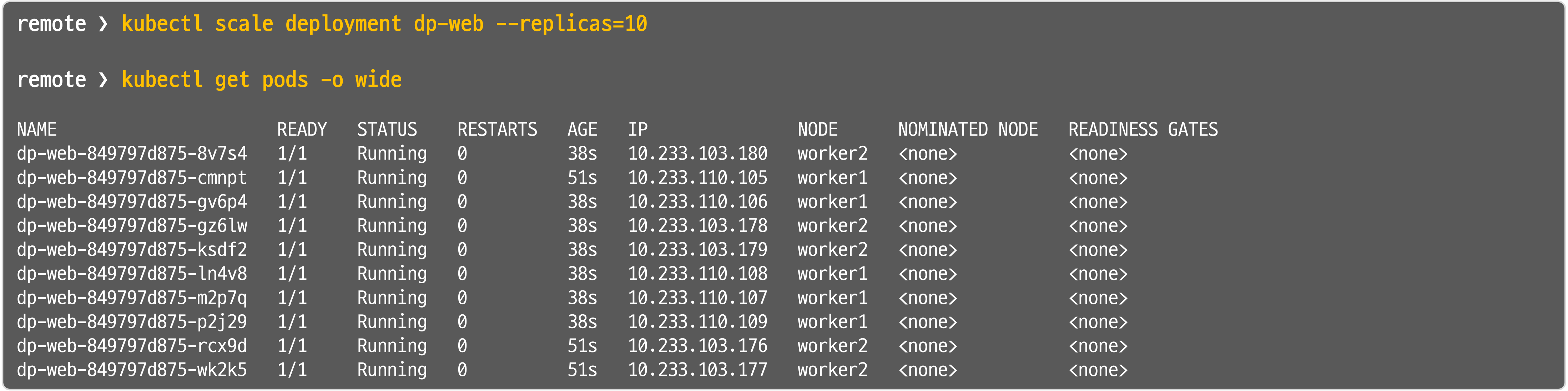
Ready

- 깔끔한 (?) 실습을 위해 Deployment를 삭제 후 다시 생성하자



```
remote ❯ kubectl delete deployments dp-web

remote ❯ kubectl create -f dp-web-v1.yaml
```

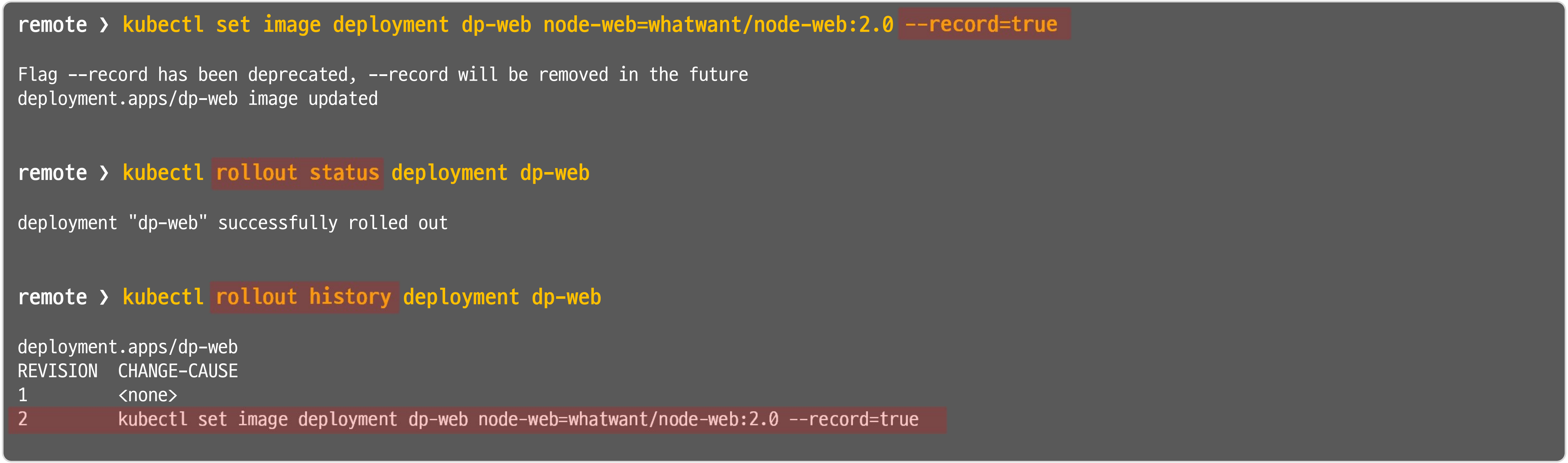
- Replicas 개수를 좀 늘려주자 (그래야 rollout 관련된 실습 내용이 잘 보일 것이기에...)



```
remote ❯ kubectl scale deployment dp-web --replicas=10

remote ❯ kubectl get pods -o wide

NAME                      READY   STATUS    RESTARTS   AGE   IP               NODE      NOMINATED NODE   READINESS GATES
dp-web-849797d875-8v7s4   1/1     Running   0          38s   10.233.103.180   worker2   <none>           <none>
dp-web-849797d875-cmnpt   1/1     Running   0          51s   10.233.110.105   worker1   <none>           <none>
dp-web-849797d875-gv6p4   1/1     Running   0          38s   10.233.110.106   worker1   <none>           <none>
dp-web-849797d875-gz6lw   1/1     Running   0          38s   10.233.103.178   worker2   <none>           <none>
dp-web-849797d875-ksdf2   1/1     Running   0          38s   10.233.103.179   worker2   <none>           <none>
dp-web-849797d875-ln4v8   1/1     Running   0          38s   10.233.110.108   worker1   <none>           <none>
dp-web-849797d875-m2p7q   1/1     Running   0          38s   10.233.110.107   worker1   <none>           <none>
dp-web-849797d875-p2j29   1/1     Running   0          38s   10.233.110.109   worker1   <none>           <none>
dp-web-849797d875-rcx9d   1/1     Running   0          51s   10.233.103.176   worker2   <none>           <none>
dp-web-849797d875-wk2k5   1/1     Running   0          51s   10.233.103.177   worker2   <none>           <none>
```

rollout history

- 옵션과 함께 `set image`를 진행한 뒤에 history를 확인해 보자.

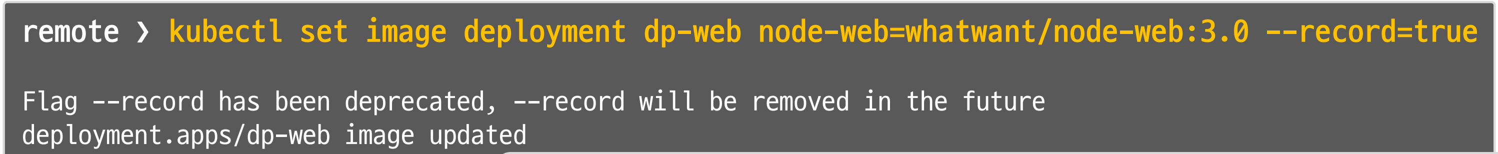


```
remote ❯ kubectl set image deployment dp-web node-web=whatwant/node-web:2.0 --record=true

Flag --record has been deprecated, --record will be removed in the future
deployment.apps/dp-web image updated


remote ❯ kubectl rollout status deployment dp-web

deployment "dp-web" successfully rolled out


remote ❯ kubectl rollout history deployment dp-web

deployment.apps/dp-web
REVISION  CHANGE-CAUSE
1         <none>
2         kubectl set image deployment dp-web node-web=whatwant/node-web:2.0 --record=true
```

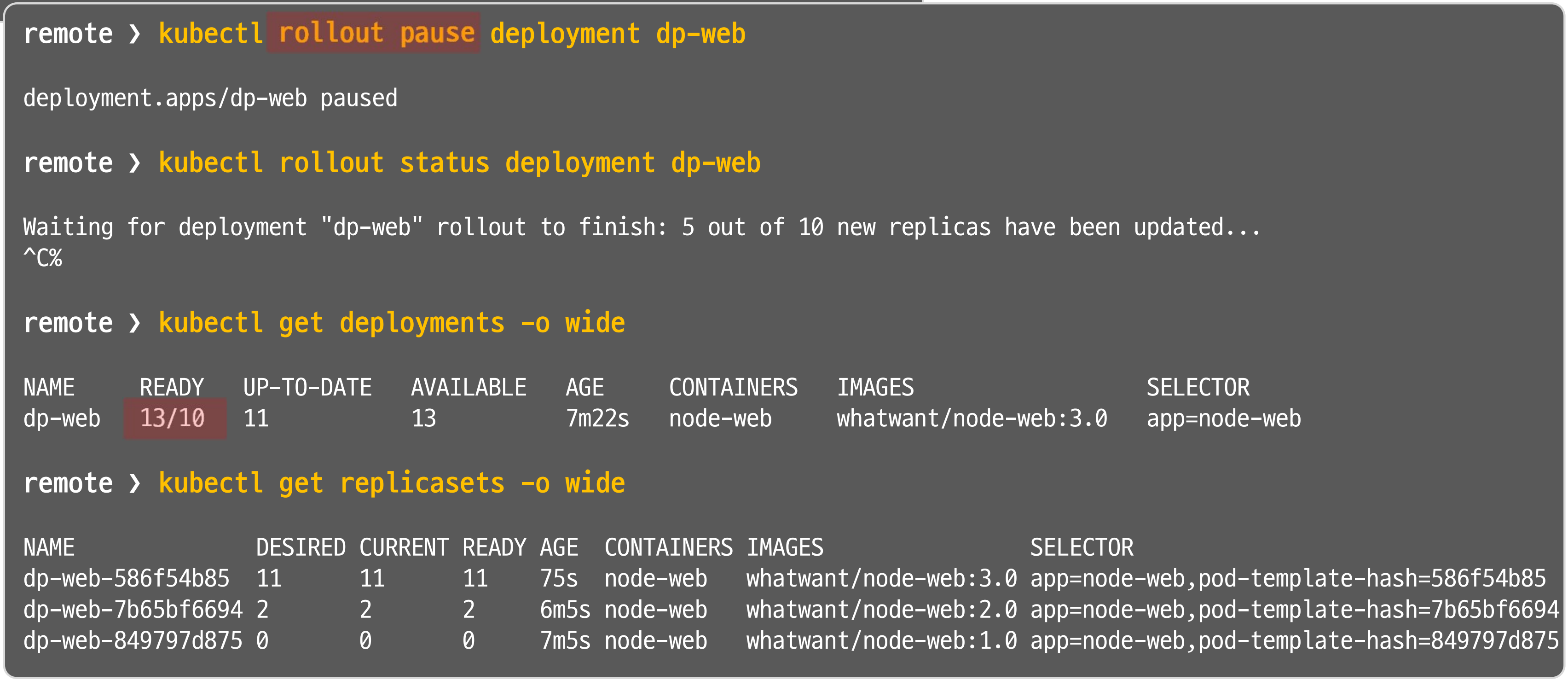
rollout pause

- 배포 업데이트 중에 문제가 발생하면, 잠시 멈춰야 한다.

. 업데이트 중에 재빨리 멈춰야 하기 때문에 터미널을 2개 띄워 놓고 빠르게 진행하는 것을 추천한다.



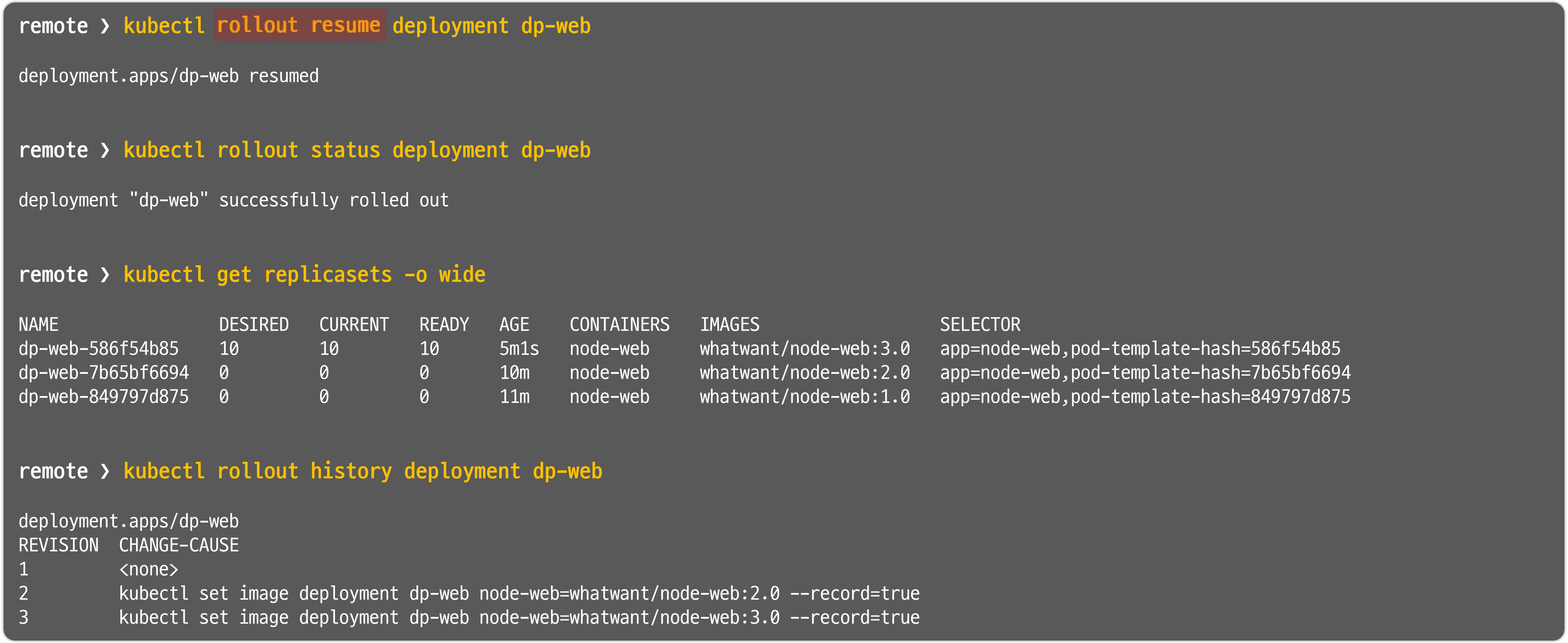
```
remote ❯ kubectl set image deployment dp-web node-web=whatwant/node-web:3.0 --record=true

Flag --record has been deprecated, --record will be removed in the future
deployment.apps/dp-web image updated
```



```
remote ❯ kubectl rollout pause deployment dp-web

deployment.apps/dp-web paused

remote ❯ kubectl rollout status deployment dp-web

Waiting for deployment "dp-web" rollout to finish: 5 out of 10 new replicas have been updated...
^C%

remote ❯ kubectl get deployments -o wide

NAME     READY   UP-TO-DATE   AVAILABLE   AGE     CONTAINERS   IMAGES                   SELECTOR
dp-web   13/10   11           13          7m22s   node-web     whatwant/node-web:3.0    app=node-web

remote ❯ kubectl get replicasets -o wide

NAME              DESIRED CURRENT READY AGE  CONTAINERS IMAGES                SELECTOR
dp-web-586f54b85  11      11      11    75s  node-web   whatwant/node-web:3.0 app=node-web,pod-template-hash=586f54b85
dp-web-7b65bf6694 2       2       2     6m5s node-web   whatwant/node-web:2.0 app=node-web,pod-template-hash=7b65bf6694
dp-web-849797d875 0       0       0     7m5s node-web   whatwant/node-web:1.0 app=node-web,pod-template-hash=849797d875
```

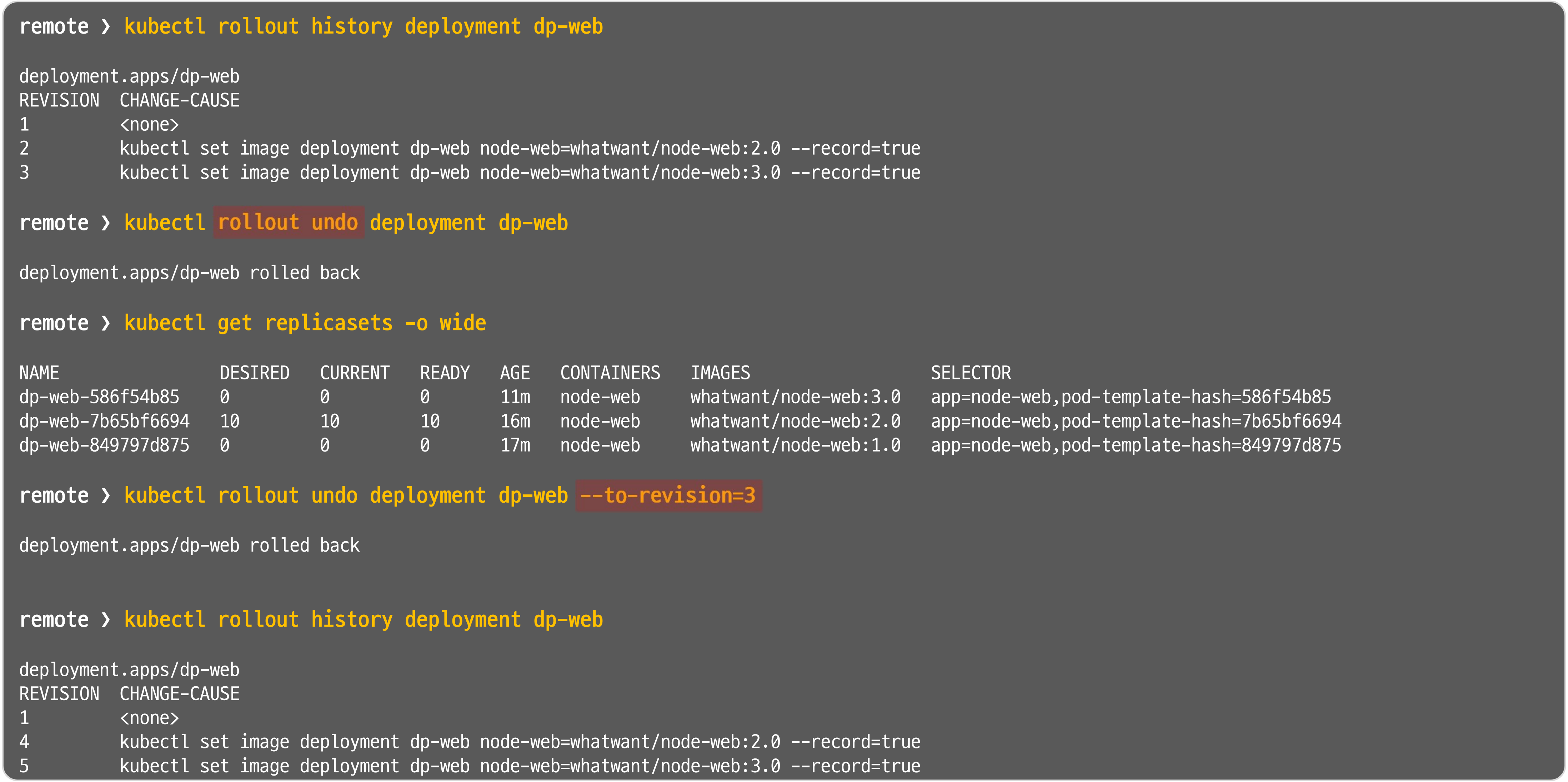
rollout resume

- 멈췄으면 다시 시작도 할 수 있어야 한다.



```
remote ❯ kubectl rollout resume deployment dp-web

deployment.apps/dp-web resumed


remote ❯ kubectl rollout status deployment dp-web

deployment "dp-web" successfully rolled out


remote ❯ kubectl get replicasets -o wide

NAME                DESIRED   CURRENT   READY   AGE    CONTAINERS   IMAGES                  SELECTOR
dp-web-586f54b85    10        10        10      5m1s   node-web     whatwant/node-web:3.0   app=node-web,pod-template-hash=586f54b85
dp-web-7b65bf6694   0         0         0       10m    node-web     whatwant/node-web:2.0   app=node-web,pod-template-hash=7b65bf6694
dp-web-849797d875   0         0         0       11m    node-web     whatwant/node-web:1.0   app=node-web,pod-template-hash=849797d875


remote ❯ kubectl rollout history deployment dp-web

deployment.apps/dp-web
REVISION  CHANGE-CAUSE
1         <none>
2         kubectl set image deployment dp-web node-web=whatwant/node-web:2.0 --record=true
3         kubectl set image deployment dp-web node-web=whatwant/node-web:3.0 --record=true
```

rollout undo (roll-back)

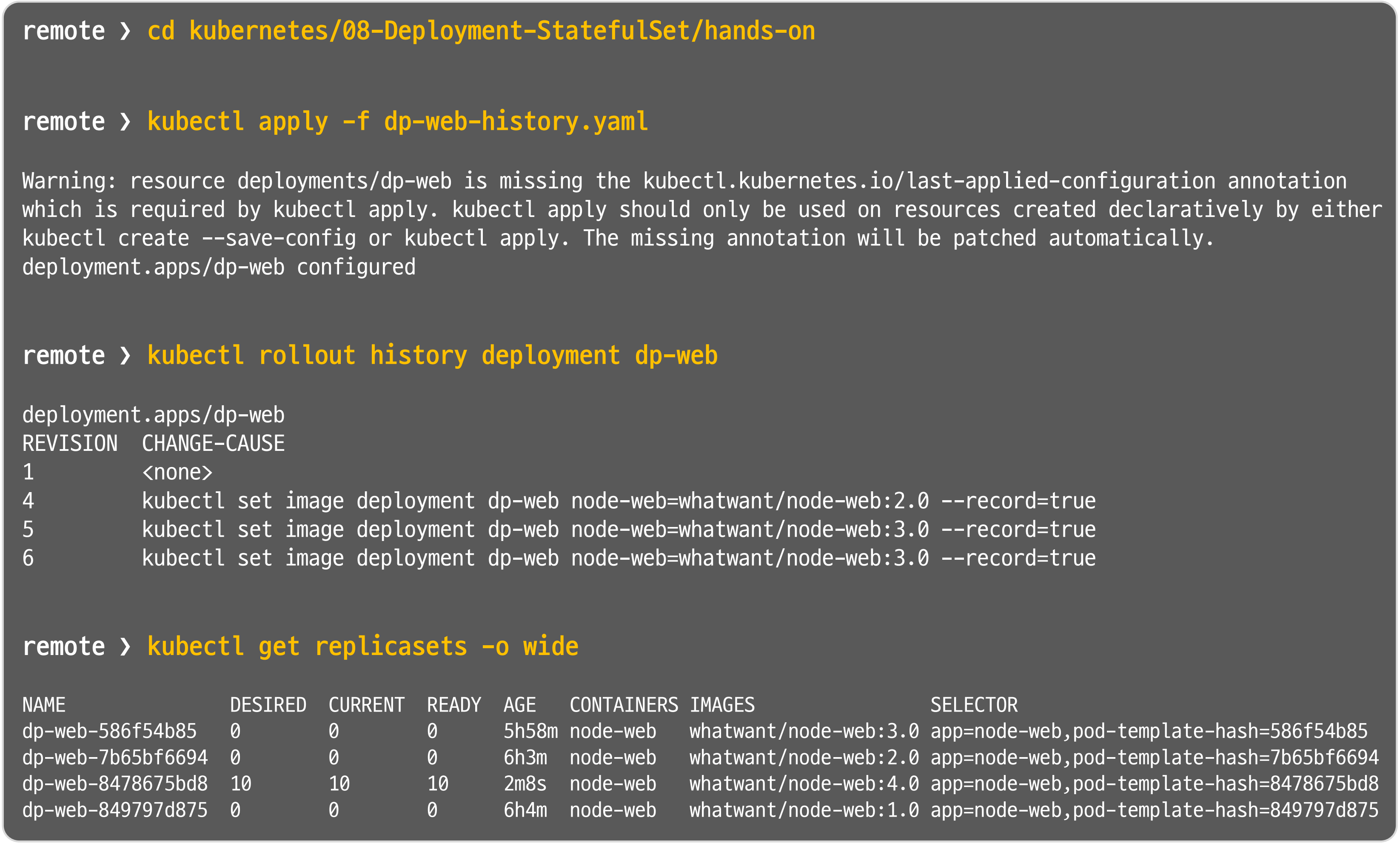


```
remote ❯ kubectl rollout history deployment dp-web

deployment.apps/dp-web
REVISION  CHANGE-CAUSE
1         <none>
2         kubectl set image deployment dp-web node-web=whatwant/node-web:2.0 --record=true
3         kubectl set image deployment dp-web node-web=whatwant/node-web:3.0 --record=true

remote ❯ kubectl rollout undo deployment dp-web

deployment.apps/dp-web rolled back

remote ❯ kubectl get replicasets -o wide

NAME                 DESIRED   CURRENT   READY   AGE   CONTAINERS   IMAGES                   SELECTOR
dp-web-586f54b85     0         0         0       11m   node-web     whatwant/node-web:3.0   app=node-web,pod-template-hash=586f54b85
dp-web-7b65bf6694    10        10        10      16m   node-web     whatwant/node-web:2.0   app=node-web,pod-template-hash=7b65bf6694
dp-web-849797d875    0         0         0       17m   node-web     whatwant/node-web:1.0   app=node-web,pod-template-hash=849797d875

remote ❯ kubectl rollout undo deployment dp-web --to-revision=3

deployment.apps/dp-web rolled back


remote ❯ kubectl rollout history deployment dp-web

deployment.apps/dp-web
REVISION  CHANGE-CAUSE
1         <none>
4         kubectl set image deployment dp-web node-web=whatwant/node-web:2.0 --record=true
5         kubectl set image deployment dp-web node-web=whatwant/node-web:3.0 --record=true
```

revisionHistoryLimit

dp-web-history.yaml

```yaml
apiVersion: apps/v1
kind: Deployment

metadata:
  name: dp-web

spec:
  replicas: 10

  selector:
    matchLabels:
      app: node-web

  template:
    metadata:
      name: node-web
      labels:
        app: node-web

    spec:
      containers:
      - image: whatwant/node-web:4.0
        name: node-web
        ports:
        - containerPort: 8080
          protocol: TCP
        imagePullPolicy: Always

  revisionHistoryLimit: 10
```



```
remote ❯ cd kubernetes/08-Deployment-StatefulSet/hands-on


remote ❯ kubectl apply -f dp-web-history.yaml

Warning: resource deployments/dp-web is missing the kubectl.kubernetes.io/last-applied-configuration annotation
which is required by kubectl apply. kubectl apply should only be used on resources created declaratively by either
kubectl create --save-config or kubectl apply. The missing annotation will be patched automatically.
deployment.apps/dp-web configured


remote ❯ kubectl rollout history deployment dp-web

deployment.apps/dp-web
REVISION  CHANGE-CAUSE
1         <none>
4         kubectl set image deployment dp-web node-web=whatwant/node-web:2.0 --record=true
5         kubectl set image deployment dp-web node-web=whatwant/node-web:3.0 --record=true
6         kubectl set image deployment dp-web node-web=whatwant/node-web:3.0 --record=true


remote ❯ kubectl get replicasets -o wide

NAME               DESIRED   CURRENT   READY   AGE    CONTAINERS  IMAGES                 SELECTOR
dp-web-586f54b85   0         0         0       5h58m  node-web    whatwant/node-web:3.0  app=node-web,pod-template-hash=586f54b85
dp-web-7b65bf6694  0         0         0       6h3m   node-web    whatwant/node-web:2.0  app=node-web,pod-template-hash=7b65bf6694
dp-web-8478675bd8  10        10        10      2m8s   node-web    whatwant/node-web:4.0  app=node-web,pod-template-hash=8478675bd8
dp-web-849797d875  0         0         0       6h4m   node-web    whatwant/node-web:1.0  app=node-web,pod-template-hash=849797d875
```

※ 참고 : https://kubernetes.io/ko/docs/concepts/workloads/controllers/deployment/#정책-초기화

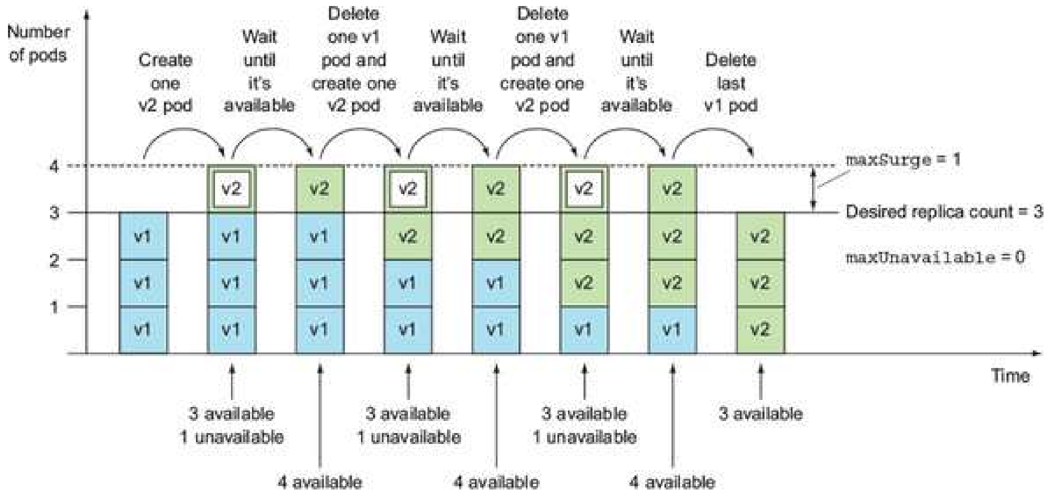
///

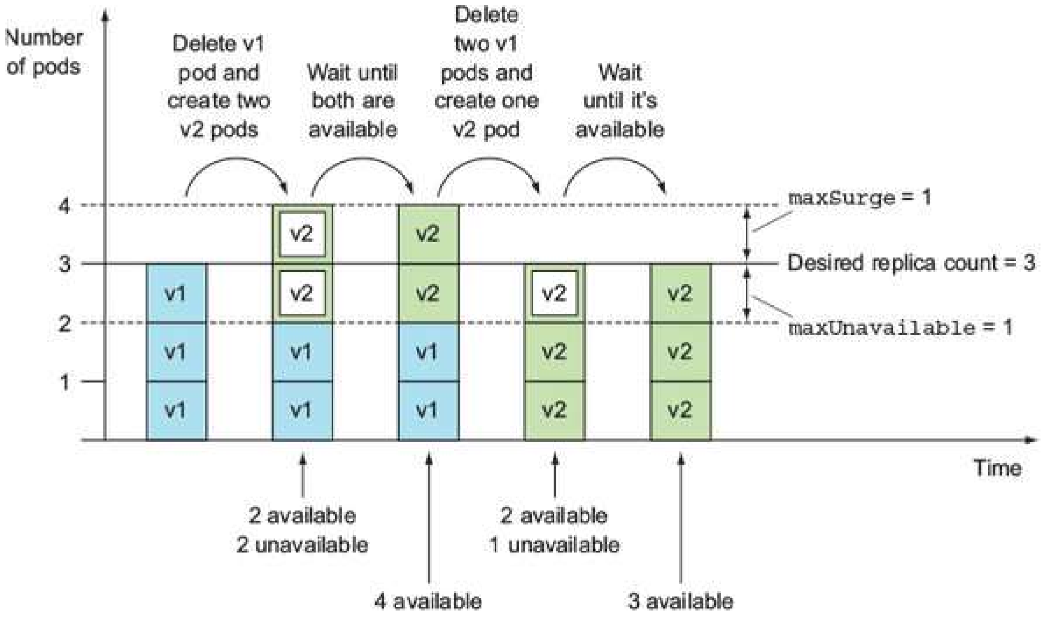
RollingUpdate

롤아웃 속도 제어

- 정상적으로 서비스 하고 있는 Pod가 최소한 몇 개가 되어야 하는지,

  동시에 실행되고 있는 Pod를 몇 개까지 감당할 수 있는 H/W 리소스를 갖고 있는지 조절함으로써 롤아웃 속도를 조절할 수 있다.

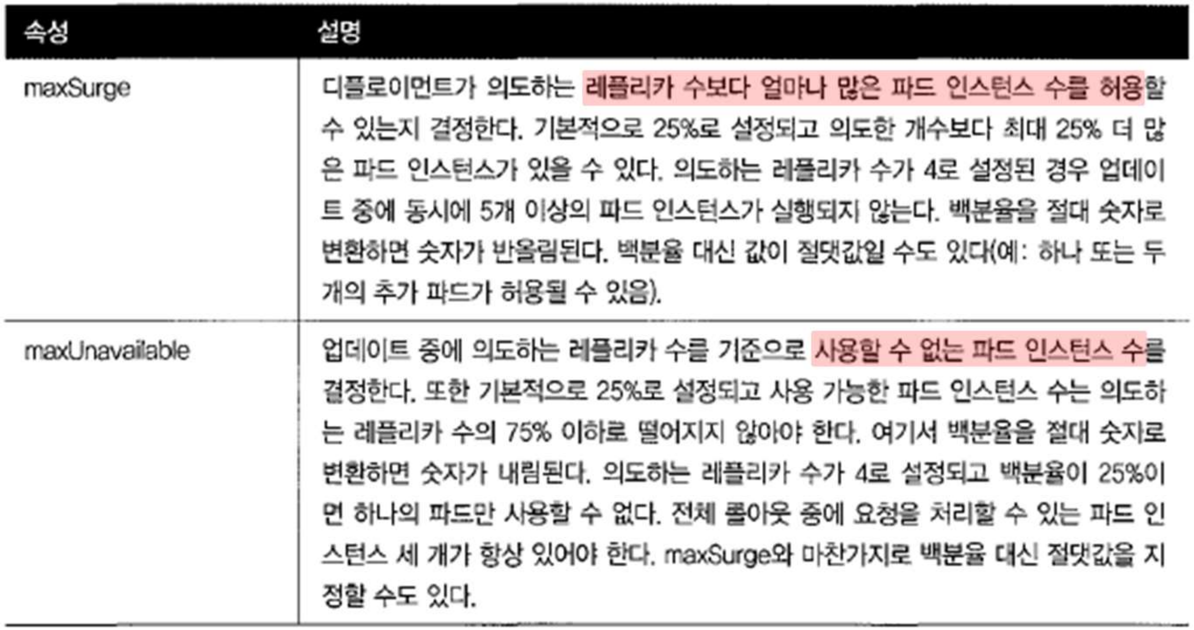




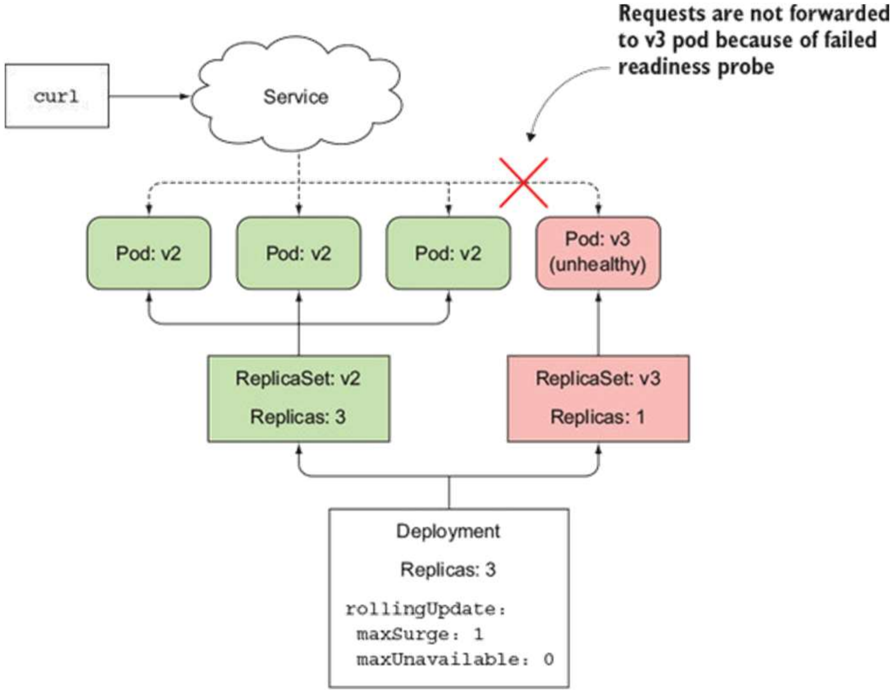
Number of pods
Delete v1 pod and create two v2 pods
Wait until both are available
Delete two v1 pods and create one v2 pod
Wait until it's available
4
3
2
1
v1
v2
maxSurge = 1
Desired replica count = 3
maxUnavailable = 1
Time
2 available
2 unavailable
4 available
2 available
1 unavailable
3 available

※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-9/215

maxSurge / maxUnavailable



속성	설명
maxSurge	디플로이먼트가 의도하는 레플리카 수보다 얼마나 많은 파드 인스턴스 수를 허용할 수 있는지 결정한다. 기본적으로 25%로 설정되고 의도한 개수보다 최대 25% 더 많은 파드 인스턴스가 있을 수 있다. 의도하는 레플리카 수가 4로 설정된 경우 업데이트 중에 동시에 5개 이상의 파드 인스턴스가 실행되지 않는다. 백분율을 절대 숫자로 변환하면 숫자가 반올림된다. 백분율 대신 값이 절댓값일 수도 있다(예: 하나 또는 두 개의 추가 파드가 허용될 수 있음).
maxUnavailable	업데이트 중에 의도하는 레플리카 수를 기준으로 사용할 수 없는 파드 인스턴스 수를 결정한다. 또한 기본적으로 25%로 설정되고 사용 가능한 파드 인스턴스 수는 의도하는 레플리카 수의 75% 이하로 떨어지지 않아야 한다. 여기서 백분율을 절대 숫자로 변환하면 숫자가 내림된다. 의도하는 레플리카 수가 4로 설정되고 백분율이 25%이면 하나의 파드만 사용할 수 없다. 전체 롤아웃 중에 요청을 처리할 수 있는 파드 인스턴스 세 개가 항상 있어야 한다. maxSurge와 마찬가지로 백분율 대신 절댓값을 지정할 수도 있다.

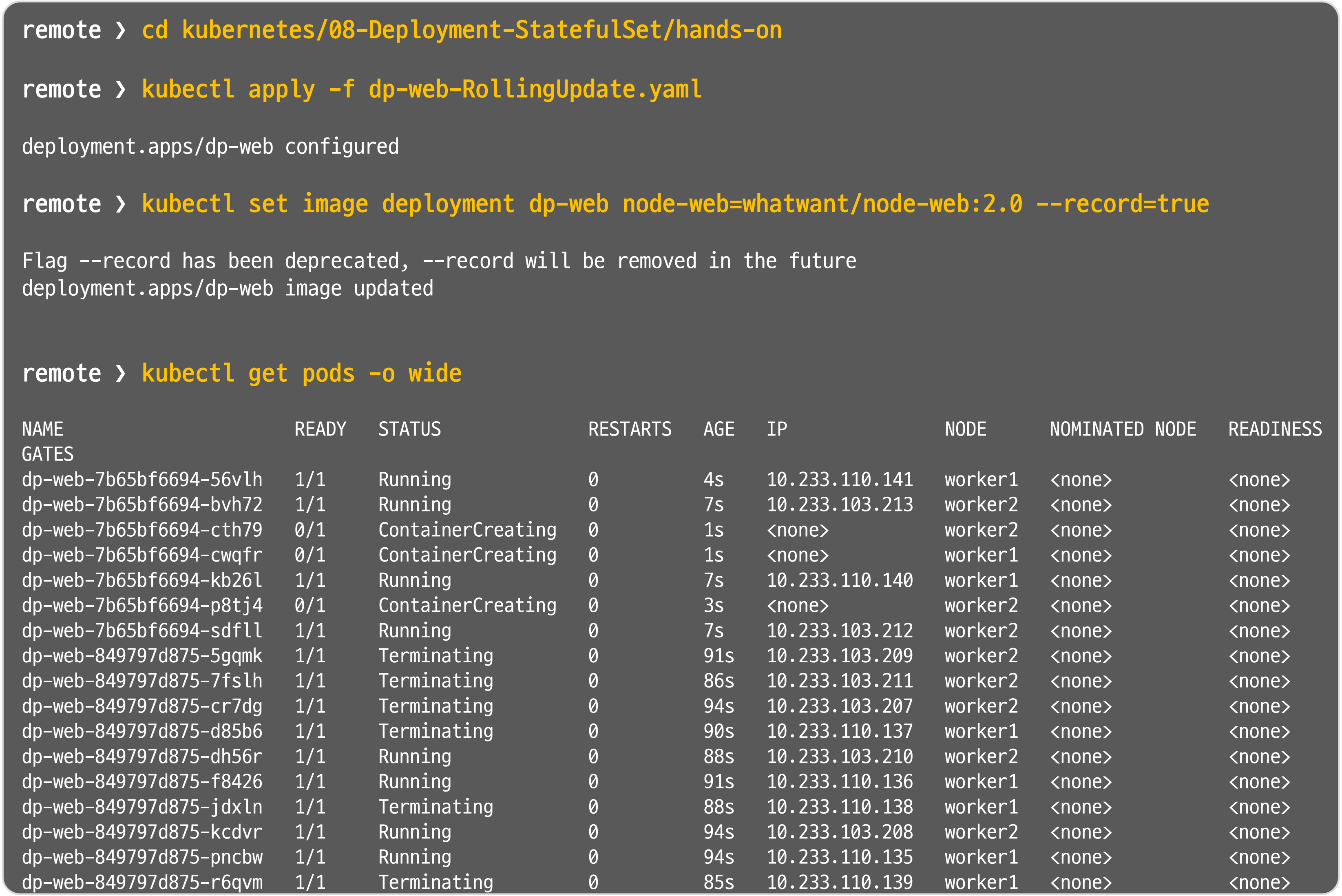


※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-9/262

YAML

dp-web-RollingUpdate.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: dp-web

spec:
  replicas: 10

  selector:
    matchLabels:
      app: node-web

  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 25%

  template:
    metadata:
      name: node-web
      labels:
        app: node-web
    spec:
      containers:
      - image: whatwant/node-web:1.0
        name: node-web
        ports:
        - containerPort: 8080
          protocol: TCP
        imagePullPolicy: Always
```



```
remote ❯ cd kubernetes/08-Deployment-StatefulSet/hands-on

remote ❯ kubectl apply -f dp-web-RollingUpdate.yaml

deployment.apps/dp-web configured

remote ❯ kubectl set image deployment dp-web node-web=whatwant/node-web:2.0 --record=true

Flag --record has been deprecated, --record will be removed in the future
deployment.apps/dp-web image updated


remote ❯ kubectl get pods -o wide

NAME                        READY   STATUS              RESTARTS   AGE   IP               NODE      NOMINATED NODE   READINESS
GATES
dp-web-7b65bf6694-56vlh     1/1     Running             0          4s    10.233.110.141   worker1   <none>           <none>
dp-web-7b65bf6694-bvh72     1/1     Running             0          7s    10.233.103.213   worker2   <none>           <none>
dp-web-7b65bf6694-cth79     0/1     ContainerCreating   0          1s    <none>           worker2   <none>           <none>
dp-web-7b65bf6694-cwqfr     0/1     ContainerCreating   0          1s    <none>           worker1   <none>           <none>
dp-web-7b65bf6694-kb26l     1/1     Running             0          7s    10.233.110.140   worker1   <none>           <none>
dp-web-7b65bf6694-p8tj4     0/1     ContainerCreating   0          3s    <none>           worker2   <none>           <none>
dp-web-7b65bf6694-sdfll     1/1     Running             0          7s    10.233.103.212   worker2   <none>           <none>
dp-web-849797d875-5gqmk     1/1     Terminating         0          91s   10.233.103.209   worker2   <none>           <none>
dp-web-849797d875-7fslh     1/1     Terminating         0          86s   10.233.103.211   worker2   <none>           <none>
dp-web-849797d875-cr7dg     1/1     Terminating         0          94s   10.233.103.207   worker2   <none>           <none>
dp-web-849797d875-d85b6     1/1     Terminating         0          90s   10.233.110.137   worker1   <none>           <none>
dp-web-849797d875-dh56r     1/1     Running             0          88s   10.233.103.210   worker2   <none>           <none>
dp-web-849797d875-f8426     1/1     Running             0          91s   10.233.110.136   worker1   <none>           <none>
dp-web-849797d875-jdxln     1/1     Terminating         0          88s   10.233.110.138   worker1   <none>           <none>
dp-web-849797d875-kcdvr     1/1     Running             0          94s   10.233.103.208   worker2   <none>           <none>
dp-web-849797d875-pncbw     1/1     Running             0          94s   10.233.110.135   worker1   <none>           <none>
dp-web-849797d875-r6qvm     1/1     Terminating         0          85s   10.233.110.139   worker1   <none>           <none>
```

///

Break

돌아오셨으면 채팅창에
재미있는 이야기
써주세요!

///

StatefulSet

Why
StatefulSet ?

개별 Pod에 각자 독립적인 저장공간 부여

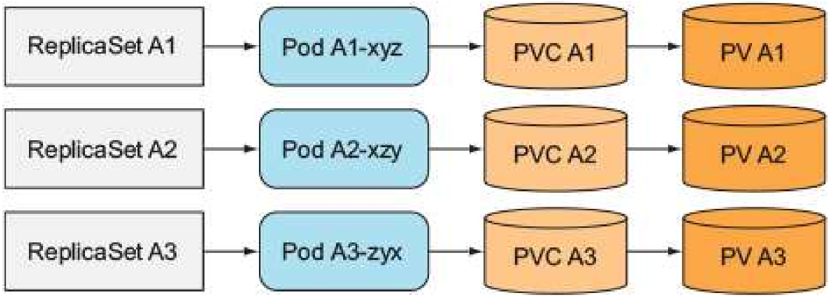
- Pod 인스턴스 별로 독립적인 저장공간을 갖도록 하려면,

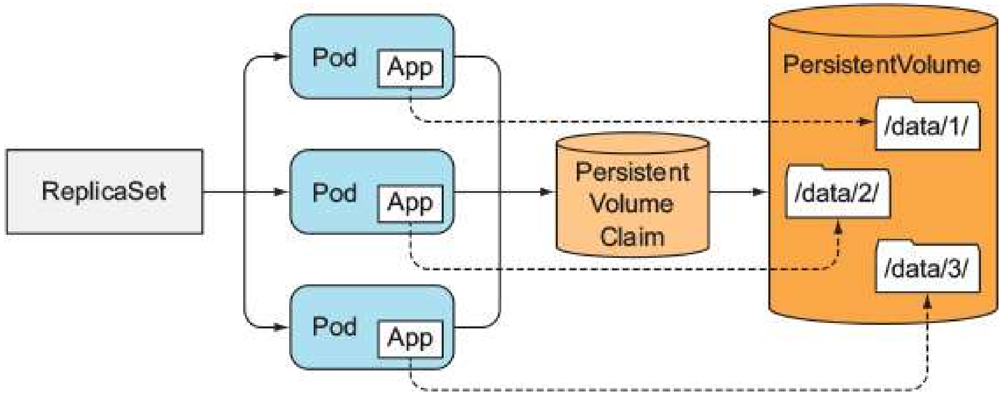
  . 수동으로 1개씩 Pod 생성

  . 1개의 Pod를 갖는 ReplicaSet을 다수 생성

  . 동일 Volume을 directory로 구분해서 사용

어렵고 귀찮음





※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-10/15

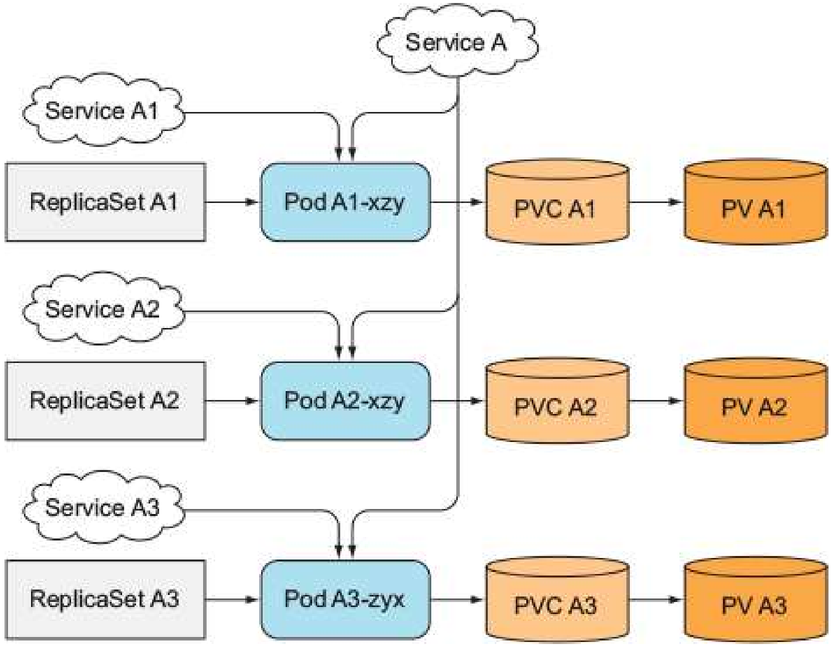
Hostname/IP가 고정이 필요한 경우

- stable identity를 요구하는 Application 존재

  . Pod가 재시작해도 기존 identity 유지 필요

  . identity : hostname, IP

어렵고 귀찮음



※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-10/30

StatefulSet vs ReplicaSet

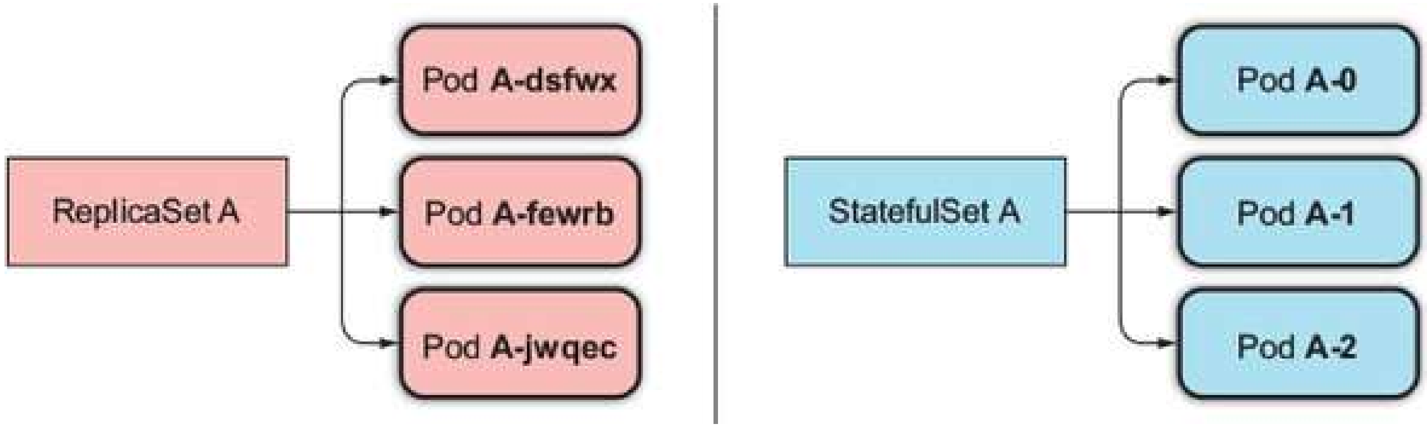
- 애완동물(Pet) vs 가축(Cattle)

  . 새로운(교체되는/재시작 하는) Pod 인스턴스는 교체되는 Pod와 hostname/IP 동일하게 실행됨

  . 각 Pod는 다른 Pod와 다른 자체 Volume 소유

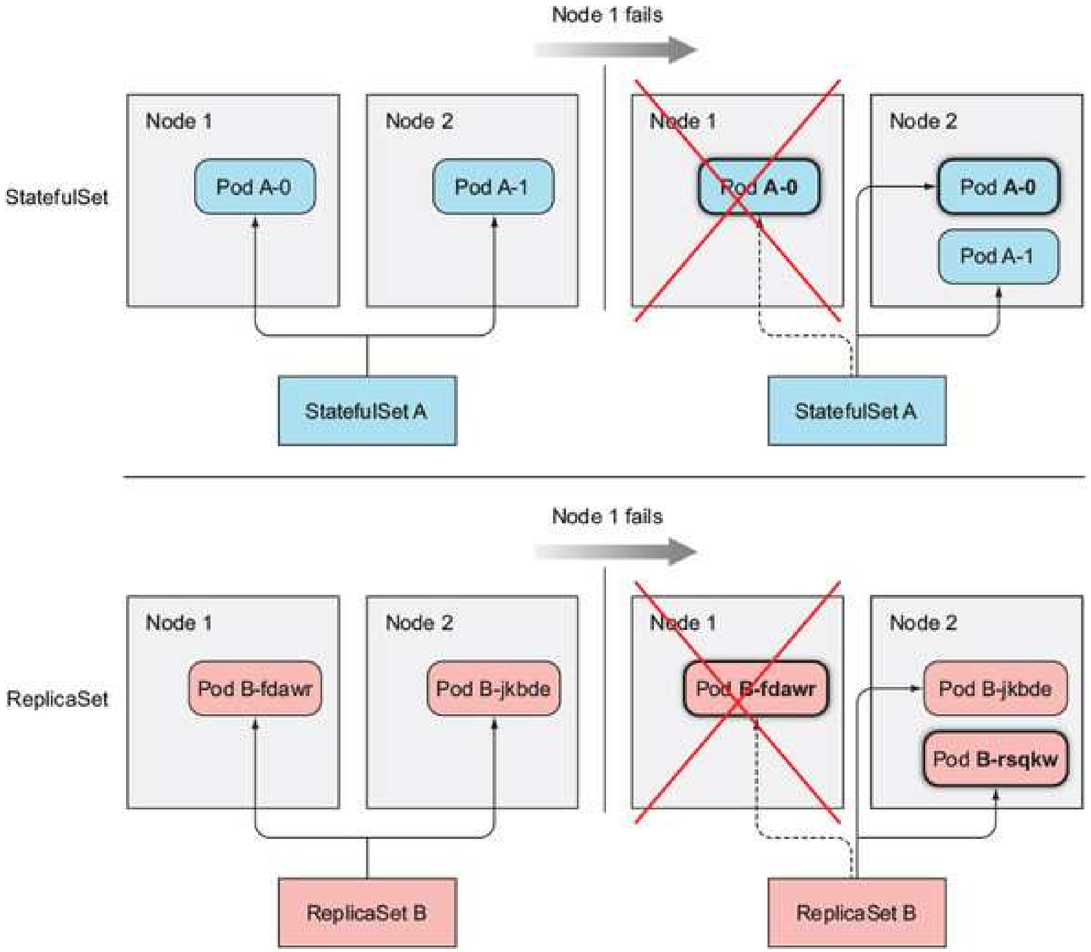
  . 새로운 Pod 인스턴스의 identity는 예측 가능

  . governing headless service : a-0.foo.default.svc.cluster.local



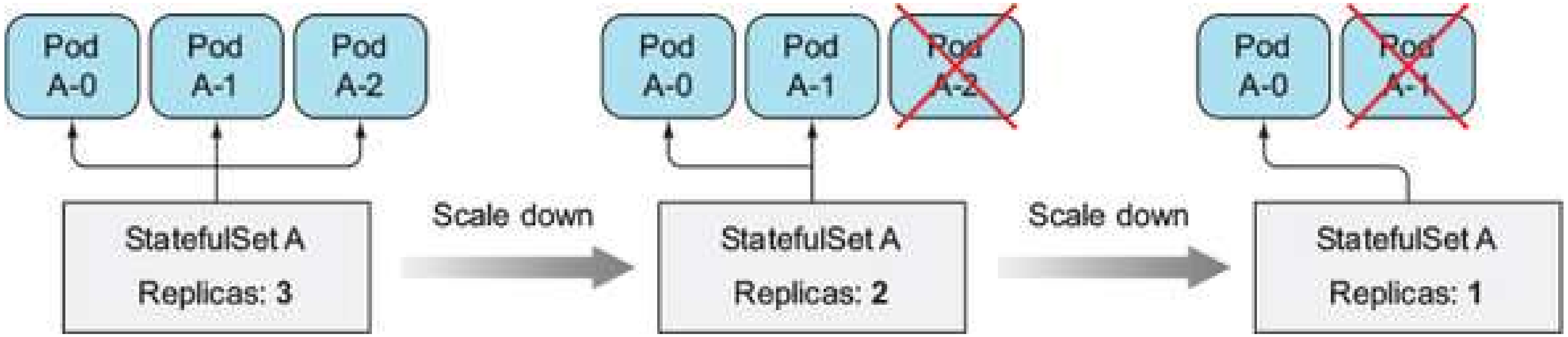
※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-10/49

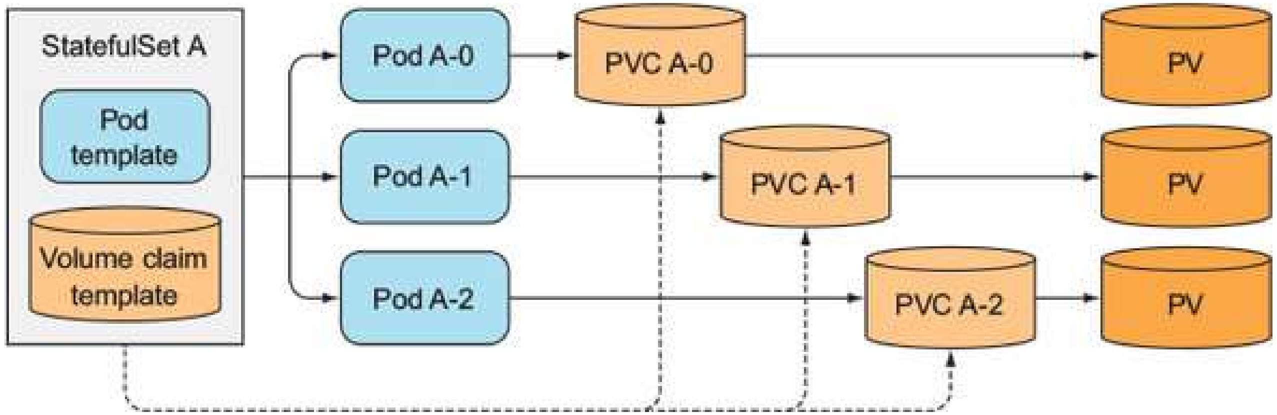
Restart(Replace) situation



※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-10/61

Scaling



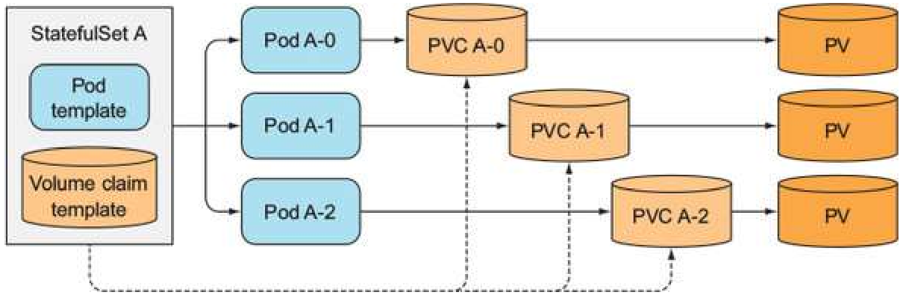


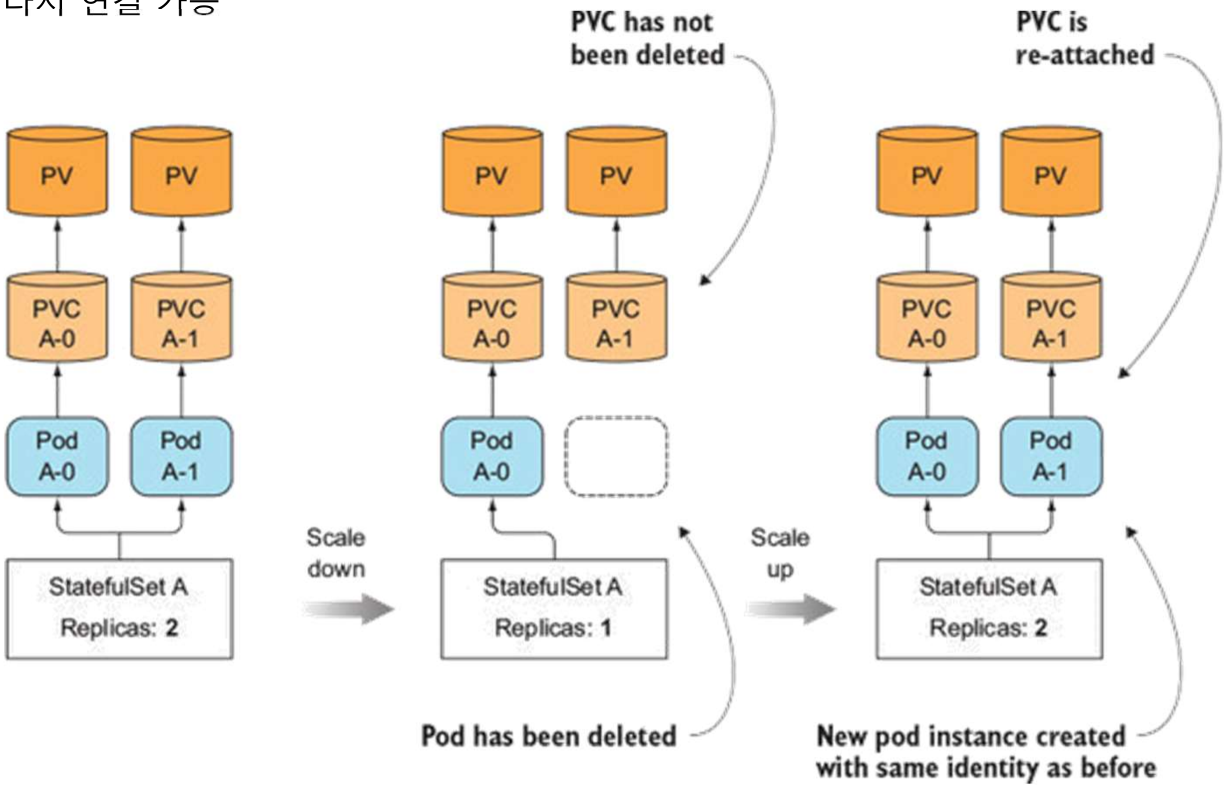
※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-10/61

Volume claim

- Volume claim template

- scale-down을 하더라도 volume은 삭제되지 않는다 → 나중에 다시 연결 가능





※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-10/61

///

Ready - Application

app-1.js

```
const http = require('http');
const os = require('os');
const fs = require('fs');
const dataFile = "/var/data/data.txt";

function fileExists(file) {
  try {
    fs.statSync(file);
    return true;
  } catch (e) {
    return false;
  }
}

var handler = function(request, response) {
  if (request.method == 'POST') {
    var file = fs.createWriteStream(dataFile);
    file.on('open', function (fd) {
      request.pipe(file);
      console.log("New data has been received and stored.");
      response.writeHead(200);
      response.end("Data stored on pod " + os.hostname() + "₩n");
    });

  } else {
    var data = fileExists(dataFile) ? fs.readFileSync(dataFile, 'utf8') : "No data posted yet";
    response.writeHead(200);
    response.write("You've hit " + os.hostname() + " (Ver3.0)₩n");
    response.end("Data stored on this pod: " + data + "₩n");
  }
};
```

```
var www = http.createServer(handler);
www.listen(8080);
```

Dockerfile-1

```
FROM node:latest

COPY ./app-1.js /app.js

ENTRYPOINT ["node", "app.js"]
```

Ready - make Container

- Dockerfile 파일명을 지정해서 build 해보자

```
remote ❯ cd kubernetes/08-Deployment-StatefulSet/hands-on

remote ❯ docker build -t whatwant/node-web:3.0 -f Dockerfile-1 .

remote ❯ docker push whatwant/node-web:3.0
```

- container로 동작을 확인해보고, 깔끔히 정리도 해보자.

```
remote ❯ mkdir /tmp/data

remote ❯ docker run -it -d -p 8080:8080 -v /tmp/data:/var/data --name web whatwant/node-web:3.0

remote ❯ curl -s http://localhost:8080

You've hit b268e8b5cf12 (Ver3.0)
Data stored on this pod: No data posted yet


remote ❯ curl -X POST -d "Wow" http://localhost:8080

Data stored on pod b268e8b5cf12


remote ❯ curl -s http://localhost:8080

You've hit b268e8b5cf12 (Ver3.0)
Data stored on this pod: Wow
```

```
remote ❯ docker stop web

remote ❯ docker rm web
```

Ready - PersistentVolume

- List 리소스를 이용하면 여러 개의 리소스를 하나의 파일로 정의할 수 있다.

pv-statefulset.yaml

```yaml
kind: List
apiVersion: v1

items:
- apiVersion: v1
  kind: PersistentVolume
  metadata:
    name: pv-a
  spec:
    capacity:
      storage: 1Mi
    accessModes:
      - ReadWriteOnce
    persistentVolumeReclaimPolicy: Retain
    hostPath:
      path: /tmp/pv-a
      type: DirectoryOrCreate
```

```yaml
- apiVersion: v1
  kind: PersistentVolume
  metadata:
    name: pv-b
  spec:
    capacity:
      storage: 1Mi
    accessModes:
      - ReadWriteOnce
    persistentVolumeReclaimPolicy: Retain
    hostPath:
      path: /tmp/pv-b
      type: DirectoryOrCreate
```

```yaml
- apiVersion: v1
  kind: PersistentVolume
  metadata:
    name: pv-c
  spec:
    capacity:
      storage: 1Mi
    accessModes:
      - ReadWriteOnce
    persistentVolumeReclaimPolicy: Retain
    hostPath:
      path: /tmp/pv-c
      type: DirectoryOrCreate
```

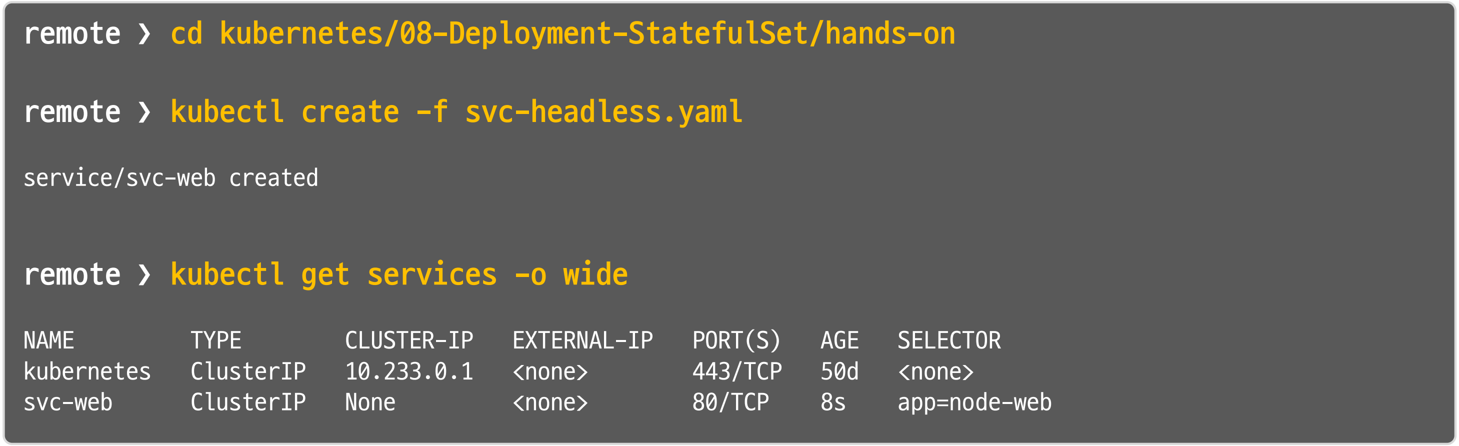
```
remote ❯ cd kubernetes/08-Deployment-StatefulSet/hands-on

remote ❯ kubectl create -f pv-statefulset.yaml

persistentvolume/pv-a created
persistentvolume/pv-b created
persistentvolume/pv-c created


remote ❯ kubectl get persistentvolumes

NAME   CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM   STORAGECLASS   REASON   AGE
pv-a   1Mi        RWO            Retain           Available                                   10s
pv-b   1Mi        RWO            Retain           Available                                   10s
pv-c   1Mi        RWO            Retain           Available                                   10s
```

Ready - Headless Service

- StatefulSet은 Headless Service가 필요하다.

svc-headless.yaml

```
apiVersion: v1
kind: Service

metadata:
  name: svc-web

spec:
  clusterIP: None

  selector:
    app: node-web

  ports:
  - name: http
    port: 80
```



```
remote ❯ cd kubernetes/08-Deployment-StatefulSet/hands-on

remote ❯ kubectl create -f svc-headless.yaml

service/svc-web created


remote ❯ kubectl get services -o wide

NAME         TYPE         CLUSTER-IP    EXTERNAL-IP   PORT(S)    AGE    SELECTOR
kubernetes   ClusterIP    10.233.0.1    <none>        443/TCP    50d    <none>
svc-web      ClusterIP    None          <none>        80/TCP     8s     app=node-web
```

※ 참고 : https://kubernetes.io/ko/docs/concepts/workloads/controllers/statefulset/

///

StatefulSet

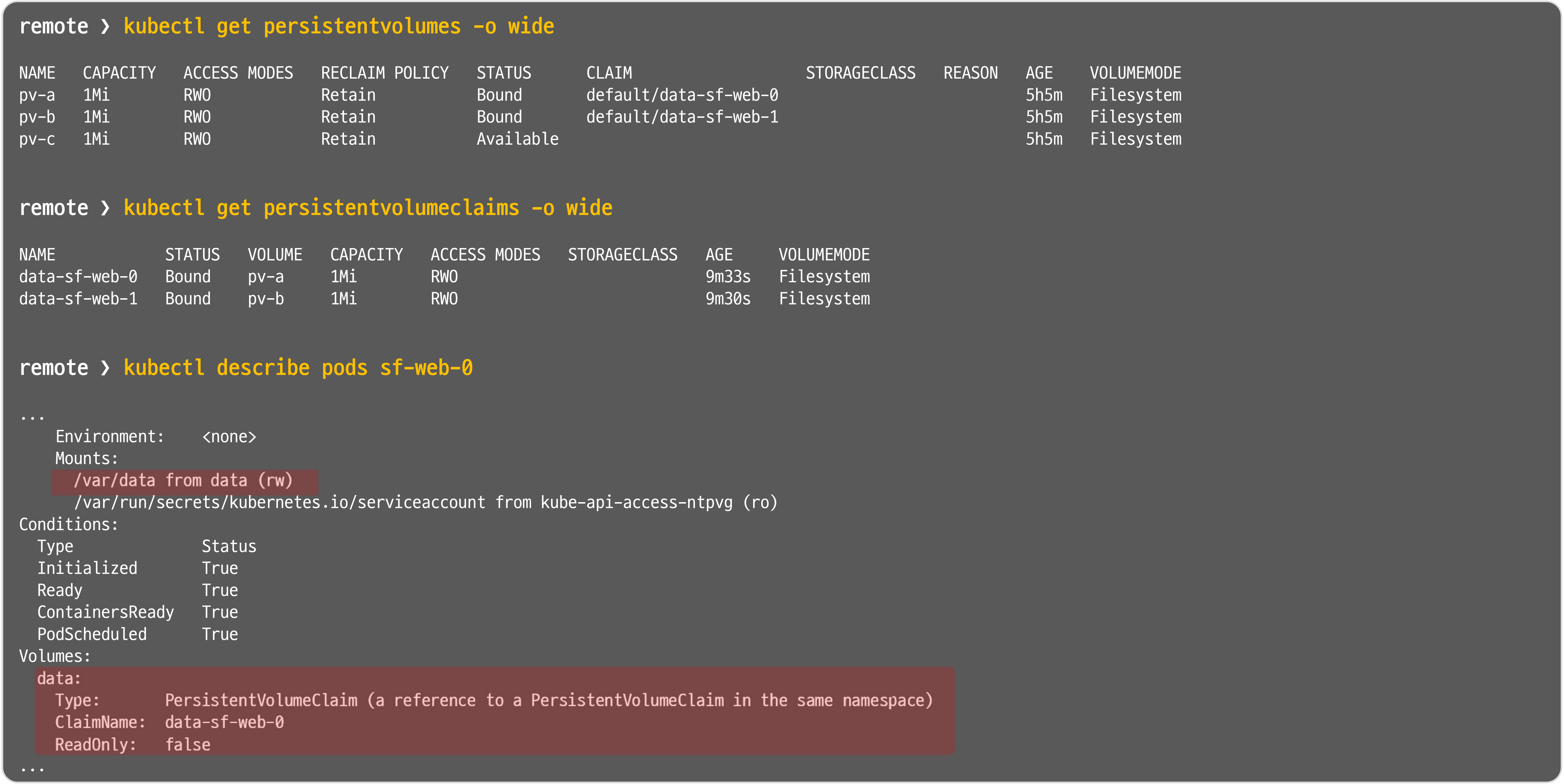
sf-web.yaml

```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: sf-web

spec:
  serviceName: svc-web

  replicas: 2

  selector:
    matchLabels:
      app: node-web

  template:
    metadata:
      labels:
        app: node-web

    spec:
      containers:
      - name: node-web
        image: whatwant/node-web:3.0
        ports:
        - name: http
          containerPort: 8080

        volumeMounts:
        - name: data
          mountPath: /var/data
```

```yaml
volumeClaimTemplates:
- metadata:
    name: data
  spec:
    resources:
      requests:
        storage: 1Mi
    accessModes:
    - ReadWriteOnce
```

```
remote ❯ cd kubernetes/08-Deployment-StatefulSet/hands-on

remote ❯ kubectl create -f sf-web.yaml


remote ❯ kubectl get statefulsets -o wide

NAME     READY   AGE   CONTAINERS   IMAGES
sf-web   2/2     99s   node-web     whatwant/node-web:3.0


remote ❯ kubectl get replicasets -o wide

No resources found in default namespace.


remote ❯ kubectl get pods -o wide

NAME       READY   STATUS    RESTARTS   AGE    IP               NODE      NOMINATED NODE   READINESS GATES
sf-web-0   1/1     Running   0          2m59s  10.233.103.222   worker2   <none>           <none>
sf-web-1   1/1     Running   0          2m56s  10.233.110.150   worker1   <none>           <none>
```

Check



```
remote ❯ kubectl get persistentvolumes -o wide

NAME   CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM                     STORAGECLASS   REASON   AGE    VOLUMEMODE
pv-a   1Mi        RWO            Retain           Bound       default/data-sf-web-0                             5h5m   Filesystem
pv-b   1Mi        RWO            Retain           Bound       default/data-sf-web-1                             5h5m   Filesystem
pv-c   1Mi        RWO            Retain           Available                                                     5h5m   Filesystem


remote ❯ kubectl get persistentvolumeclaims -o wide

NAME            STATUS   VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS   AGE     VOLUMEMODE
data-sf-web-0   Bound    pv-a     1Mi        RWO                           9m33s   Filesystem
data-sf-web-1   Bound    pv-b     1Mi        RWO                           9m30s   Filesystem


remote ❯ kubectl describe pods sf-web-0

...
    Environment:    <none>
    Mounts:
      /var/data from data (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-ntpvg (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  data:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:  data-sf-web-0
    ReadOnly:   false
...
```

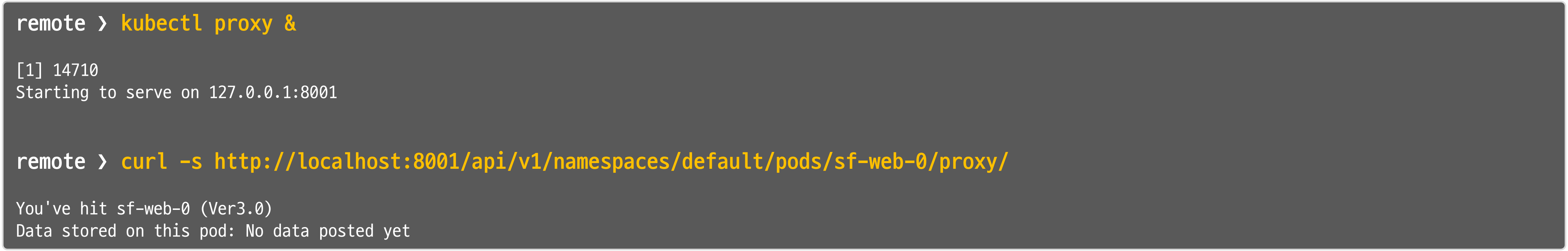
///

API Server & Proxy

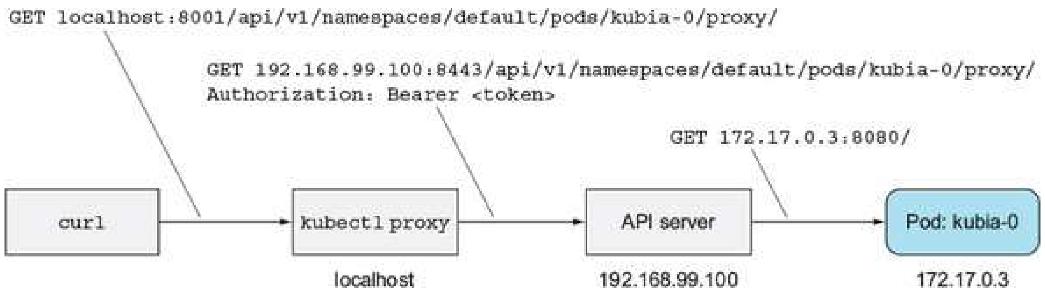
- API Server를 통해 개별 Pod에 직접 Proxy 연결 가능 (StatefulSet에서만 적용되는 것이 아니라 본래 가능)

```
<apiServerHost>:<port>/api/v1/namespaces/default/pods/<pod-name>/proxy/<path>
```

- `kubectl proxy`를 통해서 API Server 연결 가능

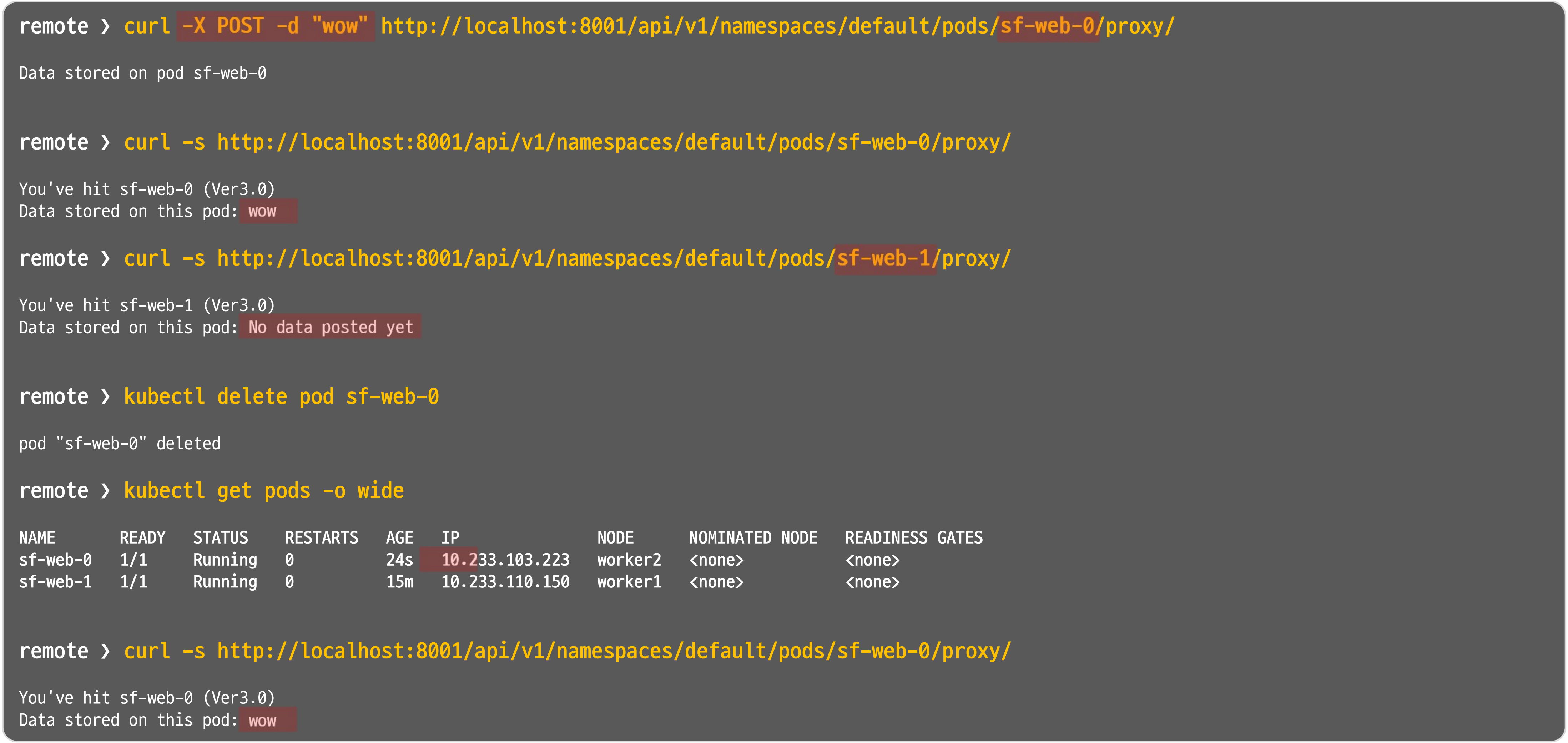


```
remote ❯ kubectl proxy &

[1] 14710
Starting to serve on 127.0.0.1:8001


remote ❯ curl -s http://localhost:8001/api/v1/namespaces/default/pods/sf-web-0/proxy/

You've hit sf-web-0 (Ver3.0)
Data stored on this pod: No data posted yet
```


GET localhost:8001/api/v1/namespaces/default/pods/kubia-0/proxy/
GET 192.168.99.100:8443/api/v1/namespaces/default/pods/kubia-0/proxy/
Authorization: Bearer <token>
GET 172.17.0.3:8080/
curl
kubectl proxy
API server
Pod: kubia-0
localhost
192.168.99.100
172.17.0.3

※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-10/153

save file & delete pod



```
remote ❯ curl -X POST -d "wow" http://localhost:8001/api/v1/namespaces/default/pods/sf-web-0/proxy/

Data stored on pod sf-web-0


remote ❯ curl -s http://localhost:8001/api/v1/namespaces/default/pods/sf-web-0/proxy/

You've hit sf-web-0 (Ver3.0)
Data stored on this pod: wow

remote ❯ curl -s http://localhost:8001/api/v1/namespaces/default/pods/sf-web-1/proxy/

You've hit sf-web-1 (Ver3.0)
Data stored on this pod: No data posted yet


remote ❯ kubectl delete pod sf-web-0

pod "sf-web-0" deleted

remote ❯ kubectl get pods -o wide

NAME       READY   STATUS    RESTARTS   AGE   IP               NODE      NOMINATED NODE   READINESS GATES
sf-web-0   1/1     Running   0          24s   10.233.103.223   worker2   <none>           <none>
sf-web-1   1/1     Running   0          15m   10.233.110.150   worker1   <none>           <none>


remote ❯ curl -s http://localhost:8001/api/v1/namespaces/default/pods/sf-web-0/proxy/

You've hit sf-web-0 (Ver3.0)
Data stored on this pod: wow
```

DNS

- dig 명령어로 확인 가능

Record	설명
A	도메인의 IP 주소를 갖고 있는 레코드
CNAME	하나의 도메인이나 하위 도메인을 다른 도메인으로 전달하며, IP 주소를 제공하지는 않습니다.
MX	이메일을 이메일 서버로 전송합니다.
TXT	관리자가 텍스트 메모를 레코드에 저장할 수 있습니다.
NS	DNS 항목의 이름 서버를 저장합니다.
SOA	도메인에 대한 관리자 정보를 저장합니다.
SRV	특정 서비스에 대한 포트를 지정합니다.
PTR	리버스 조회에서 도메인 이름을 제공합니다.

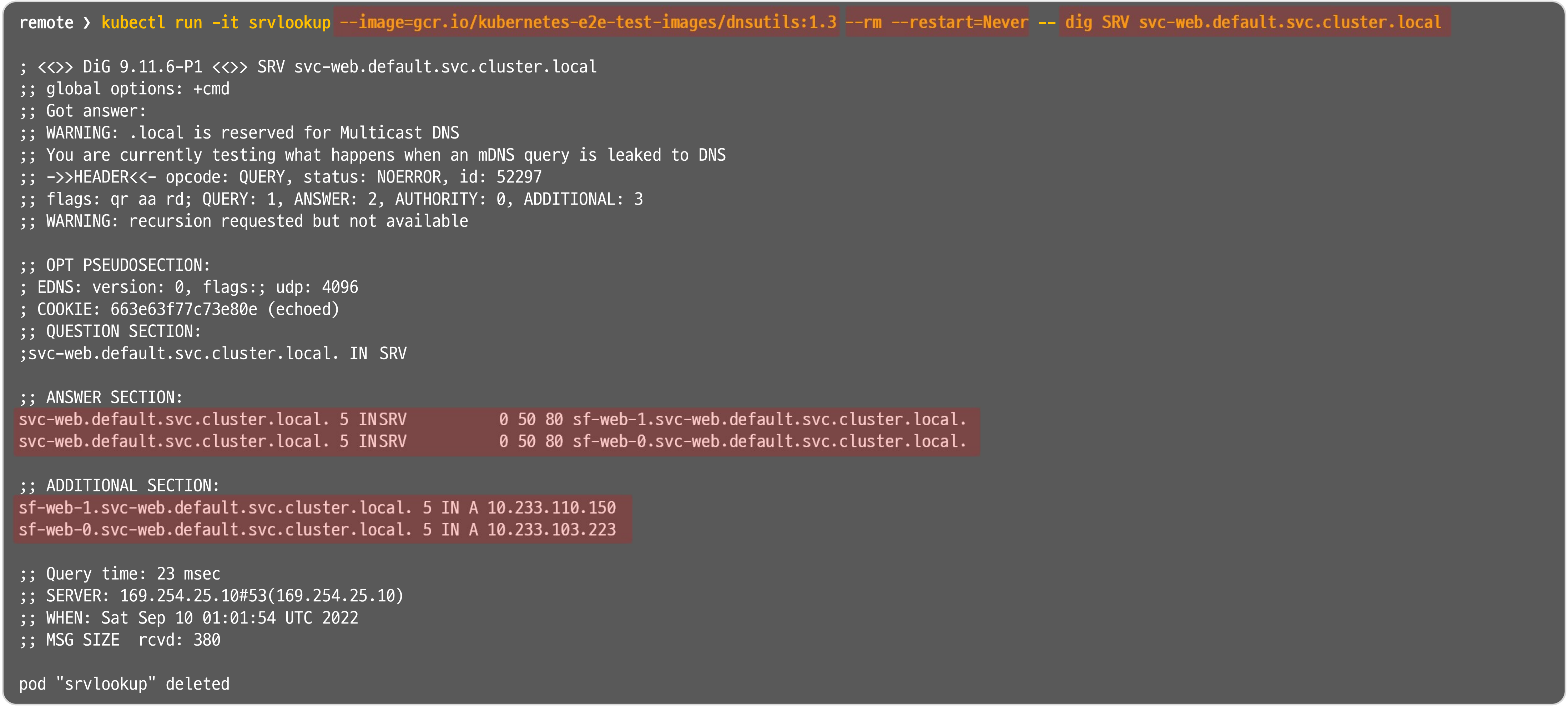
```
Dig (Domain Information Groper) is a powerful command-line tool for querying DNS name servers.
```

※ 참고 : https://www.cloudflare.com/ko-kr/learning/dns/dns-records/

※ 참고 : https://linuxize.com/post/how-to-use-dig-command-to-query-dns-in-linux/

Discovering peers (다른 Pod 찾기)

- 앞에서 생성한 Headless Service의 DNS 정보를 dig 명령어로 확인해보자.



```
remote ❯ kubectl run -it srvlookup --image=gcr.io/kubernetes-e2e-test-images/dnsutils:1.3 --rm --restart=Never -- dig SRV svc-web.default.svc.cluster.local

; <<>> DiG 9.11.6-P1 <<>> SRV svc-web.default.svc.cluster.local
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52297
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 3
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 663e63f77c73e80e (echoed)
;; QUESTION SECTION:
;svc-web.default.svc.cluster.local. IN  SRV

;; ANSWER SECTION:
svc-web.default.svc.cluster.local. 5 INSRV          0 50 80 sf-web-1.svc-web.default.svc.cluster.local.
svc-web.default.svc.cluster.local. 5 INSRV          0 50 80 sf-web-0.svc-web.default.svc.cluster.local.

;; ADDITIONAL SECTION:
sf-web-1.svc-web.default.svc.cluster.local. 5 IN A 10.233.110.150
sf-web-0.svc-web.default.svc.cluster.local. 5 IN A 10.233.103.223

;; Query time: 23 msec
;; SERVER: 169.254.25.10#53(169.254.25.10)
;; WHEN: Sat Sep 10 01:01:54 UTC 2022
;; MSG SIZE  rcvd: 380

pod "srvlookup" deleted
```

///

2nd Application

app-2.js

```
const http = require('http');
const os = require('os');
const fs = require('fs');
const dns = require('dns');

const dataFile = "/var/data/kubia.txt";
const serviceName = "svc-web.default.svc.cluster.local";
const port = 8080;


function fileExists(file) {
  ... 파일 유무 확인 ...
}

function httpGet(reqOptions, callback) {
  ... GET 방식으로 접근하여 본문 읽어오기 ...
}

var handler = function(request, response) {
  if (request.method == 'POST') {
    ... 파일 저장 ...
      response.end("Data stored on pod " + os.hostname() + "₩n");
    });

  } else {
    response.writeHead(200);
    if (request.url == '/data') {
      var data = fileExists(dataFile) ? fs.readFileSync(dataFile, 'utf8') : "No data posted yet";
      response.end(data);

    } else {
      response.write("You've hit " + os.hostname() + "₩n");
      response.write("Data stored in the cluster:₩n");
```

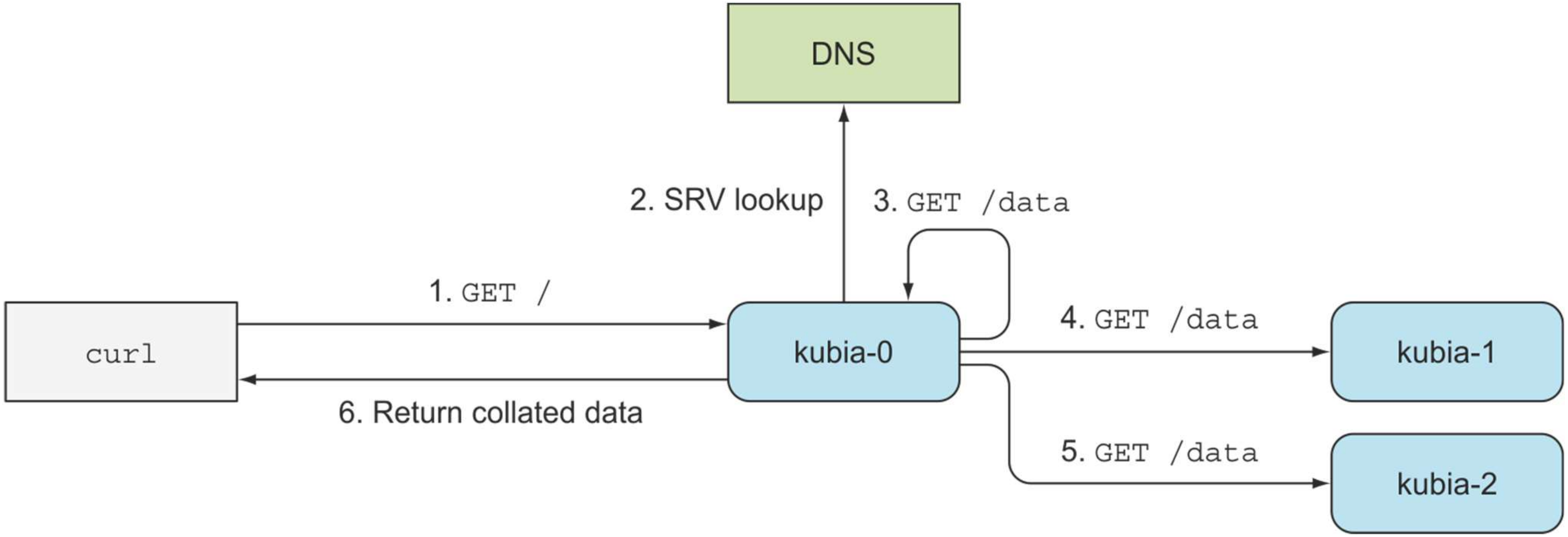
```
      dns.resolveSrv(serviceName, function (err, addresses) {
        if (err) {
          response.end("Could not look up DNS SRV records: " + err);
          return;
        }

        var numResponses = 0;
        if (addresses.length == 0) {
          response.end("No peers discovered.");

        } else {
          addresses.forEach(function (item) {
            var requestOptions = {
              host: item.name,
              port: port,
              path: '/data'
            };

            httpGet(requestOptions, function (returnedData) {
              numResponses++;
              response.write("- " + item.name + ": " + returnedData + "₩n");
              if (numResponses == addresses.length) {
                response.end();
              }
            });
          });
        }
      });
    }
  }
};

var www = http.createServer(handler);
www.listen(port);
```

make container

Dockerfile-2

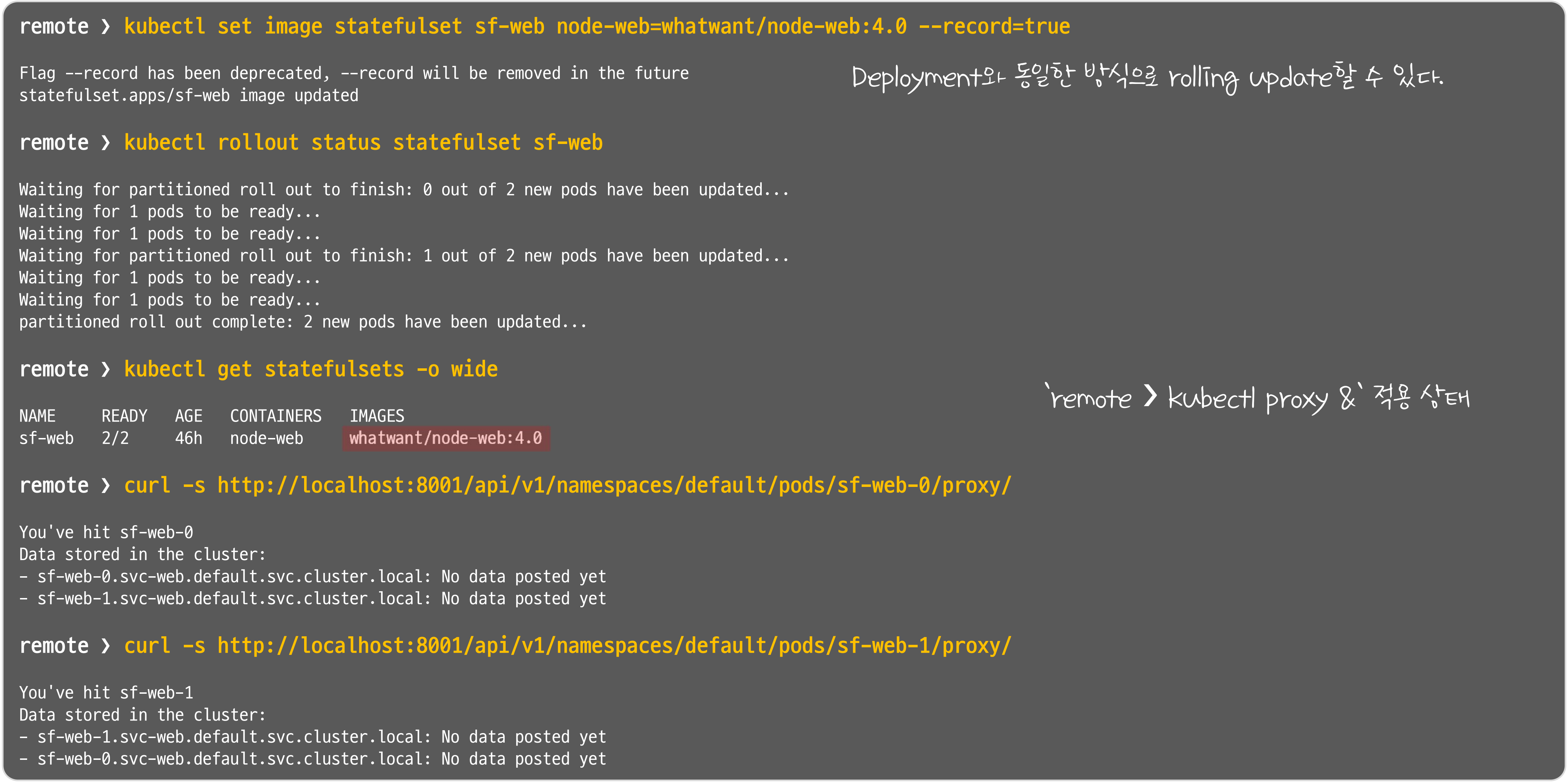
```
FROM node:latest

ADD app-2.js /app.js

ENTRYPOINT ["node", "app.js"]
```

```
remote ❯ cd kubernetes/08-Deployment-StatefulSet/hands-on

remote ❯ docker build -t whatwant/node-web:4.0 -f Dockerfile-2 .

remote ❯ docker push whatwant/node-web:4.0
```



※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-10/215

StatefulSet – Rolling Update

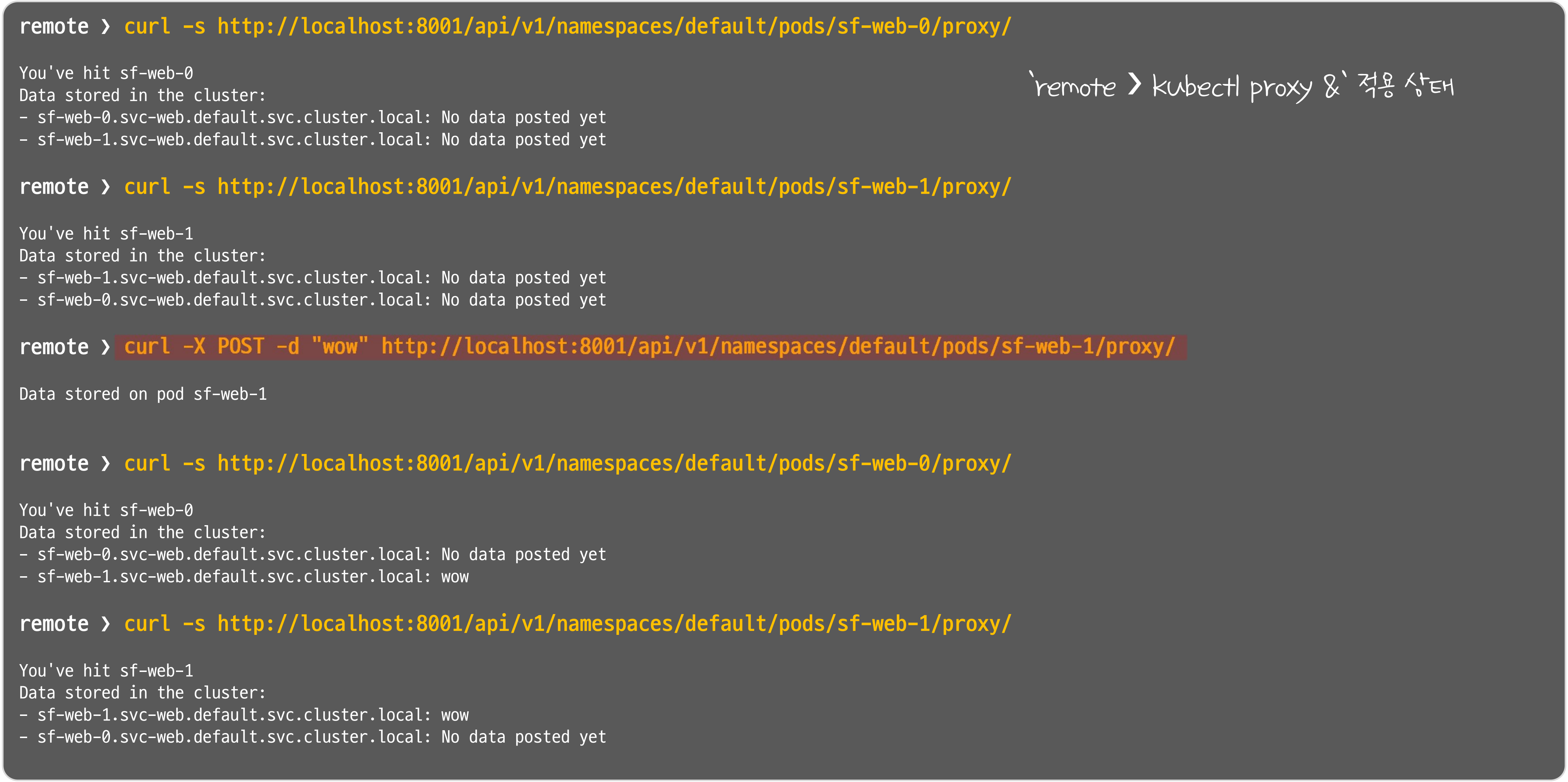


```
remote ❯ kubectl set image statefulset sf-web node-web=whatwant/node-web:4.0 --record=true

Flag --record has been deprecated, --record will be removed in the future
statefulset.apps/sf-web image updated

remote ❯ kubectl rollout status statefulset sf-web

Waiting for partitioned roll out to finish: 0 out of 2 new pods have been updated...
Waiting for 1 pods to be ready...
Waiting for 1 pods to be ready...
Waiting for partitioned roll out to finish: 1 out of 2 new pods have been updated...
Waiting for 1 pods to be ready...
Waiting for 1 pods to be ready...
partitioned roll out complete: 2 new pods have been updated...

remote ❯ kubectl get statefulsets -o wide

NAME     READY   AGE   CONTAINERS   IMAGES
sf-web   2/2     46h   node-web     whatwant/node-web:4.0

remote ❯ curl -s http://localhost:8001/api/v1/namespaces/default/pods/sf-web-0/proxy/

You've hit sf-web-0
Data stored in the cluster:
- sf-web-0.svc-web.default.svc.cluster.local: No data posted yet
- sf-web-1.svc-web.default.svc.cluster.local: No data posted yet

remote ❯ curl -s http://localhost:8001/api/v1/namespaces/default/pods/sf-web-1/proxy/

You've hit sf-web-1
Data stored in the cluster:
- sf-web-1.svc-web.default.svc.cluster.local: No data posted yet
- sf-web-0.svc-web.default.svc.cluster.local: No data posted yet
```

Deployment와 동일한 방식으로 rolling update할 수 있다.

`remote ❯ kubectl proxy &` 적용 상태

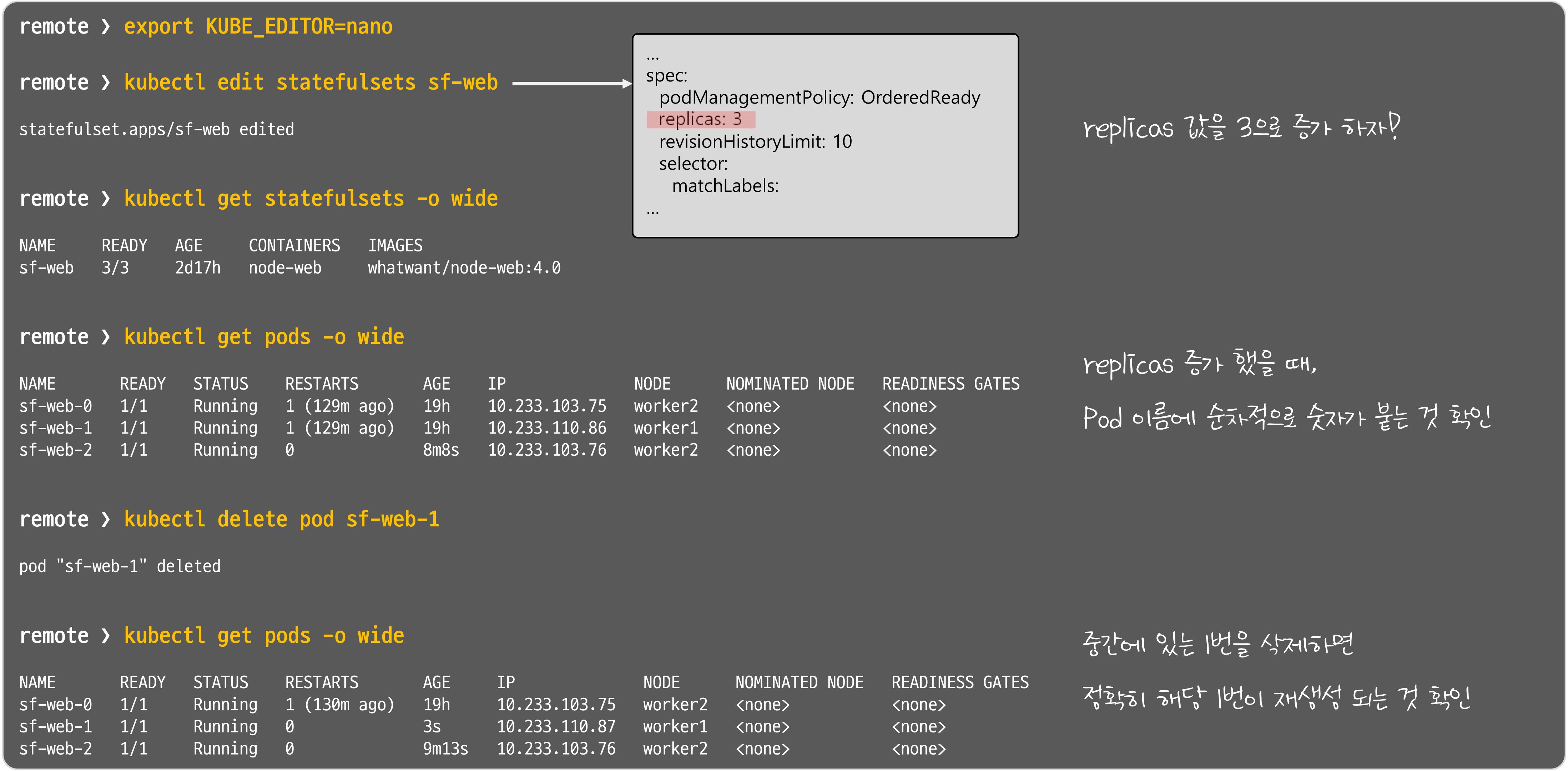
StatefulSet – check



```
remote ❯ curl -s http://localhost:8001/api/v1/namespaces/default/pods/sf-web-0/proxy/

You've hit sf-web-0
Data stored in the cluster:
- sf-web-0.svc-web.default.svc.cluster.local: No data posted yet
- sf-web-1.svc-web.default.svc.cluster.local: No data posted yet

remote ❯ curl -s http://localhost:8001/api/v1/namespaces/default/pods/sf-web-1/proxy/

You've hit sf-web-1
Data stored in the cluster:
- sf-web-1.svc-web.default.svc.cluster.local: No data posted yet
- sf-web-0.svc-web.default.svc.cluster.local: No data posted yet

remote ❯ curl -X POST -d "wow" http://localhost:8001/api/v1/namespaces/default/pods/sf-web-1/proxy/

Data stored on pod sf-web-1


remote ❯ curl -s http://localhost:8001/api/v1/namespaces/default/pods/sf-web-0/proxy/

You've hit sf-web-0
Data stored in the cluster:
- sf-web-0.svc-web.default.svc.cluster.local: No data posted yet
- sf-web-1.svc-web.default.svc.cluster.local: wow

remote ❯ curl -s http://localhost:8001/api/v1/namespaces/default/pods/sf-web-1/proxy/

You've hit sf-web-1
Data stored in the cluster:
- sf-web-1.svc-web.default.svc.cluster.local: wow
- sf-web-0.svc-web.default.svc.cluster.local: No data posted yet
```

`remote ❯ kubectl proxy &` 적용 상태

StatefulSet – Delete



```
remote ❯ export KUBE_EDITOR=nano

remote ❯ kubectl edit statefulsets sf-web

statefulset.apps/sf-web edited


remote ❯ kubectl get statefulsets -o wide

NAME     READY   AGE     CONTAINERS   IMAGES
sf-web   3/3     2d17h   node-web     whatwant/node-web:4.0


remote ❯ kubectl get pods -o wide

NAME       READY   STATUS    RESTARTS       AGE    IP              NODE      NOMINATED NODE   READINESS GATES
sf-web-0   1/1     Running   1 (129m ago)   19h    10.233.103.75   worker2   <none>           <none>
sf-web-1   1/1     Running   1 (129m ago)   19h    10.233.110.86   worker1   <none>           <none>
sf-web-2   1/1     Running   0              8m8s   10.233.103.76   worker2   <none>           <none>


remote ❯ kubectl delete pod sf-web-1

pod "sf-web-1" deleted


remote ❯ kubectl get pods -o wide

NAME       READY   STATUS    RESTARTS       AGE     IP              NODE      NOMINATED NODE   READINESS GATES
sf-web-0   1/1     Running   1 (130m ago)   19h     10.233.103.75   worker2   <none>           <none>
sf-web-1   1/1     Running   0              3s      10.233.110.87   worker1   <none>           <none>
sf-web-2   1/1     Running   0              9m13s   10.233.103.76   worker2   <none>           <none>
```

```
…
spec:
  podManagementPolicy: OrderedReady
  replicas: 3
  revisionHistoryLimit: 10
  selector:
    matchLabels:
…
```

replicas 값을 3으로 증가 하자!

replicas 증가 했을 때,

Pod 이름에 순차적으로 숫자가 붙는 것 확인

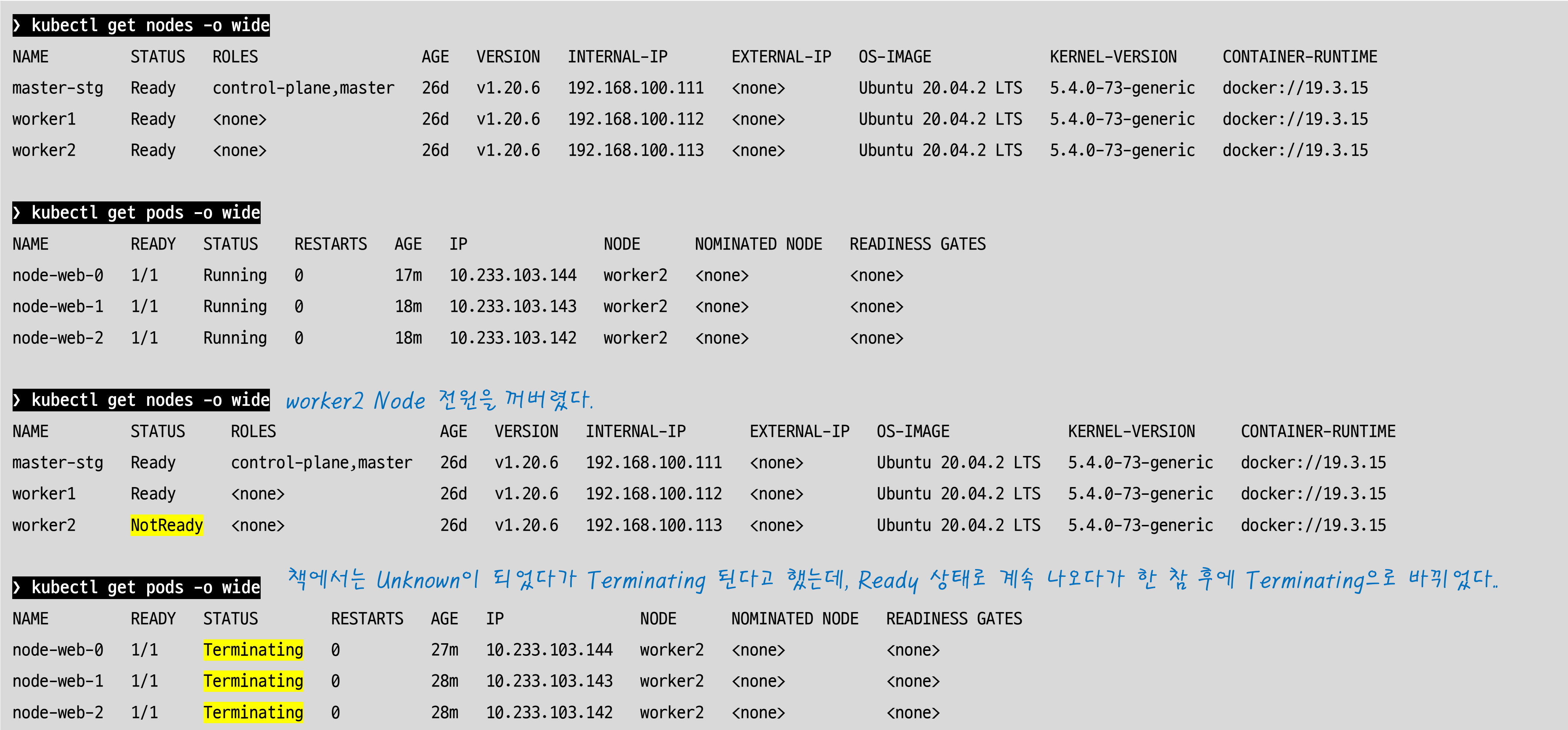
중간에 있는 1번을 삭제하면

정확히 해당 1번이 재생성 되는 것 확인

///

실습 : StatefulSet 장애 – Worker Node 오류 1/2

- Stateful Pod가 실행 중인 Worker Node에 장애가 발생하면 ?



```
❯ kubectl get nodes -o wide
NAME         STATUS   ROLES                  AGE   VERSION   INTERNAL-IP       EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION     CONTAINER-RUNTIME
master-stg   Ready    control-plane,master   26d   v1.20.6   192.168.100.111   <none>        Ubuntu 20.04.2 LTS   5.4.0-73-generic   docker://19.3.15
worker1      Ready    <none>                 26d   v1.20.6   192.168.100.112   <none>        Ubuntu 20.04.2 LTS   5.4.0-73-generic   docker://19.3.15
worker2      Ready    <none>                 26d   v1.20.6   192.168.100.113   <none>        Ubuntu 20.04.2 LTS   5.4.0-73-generic   docker://19.3.15

❯ kubectl get pods -o wide
NAME         READY   STATUS    RESTARTS   AGE   IP               NODE      NOMINATED NODE   READINESS GATES
node-web-0   1/1     Running   0          17m   10.233.103.144   worker2   <none>           <none>
node-web-1   1/1     Running   0          18m   10.233.103.143   worker2   <none>           <none>
node-web-2   1/1     Running   0          18m   10.233.103.142   worker2   <none>           <none>

❯ kubectl get nodes -o wide
```

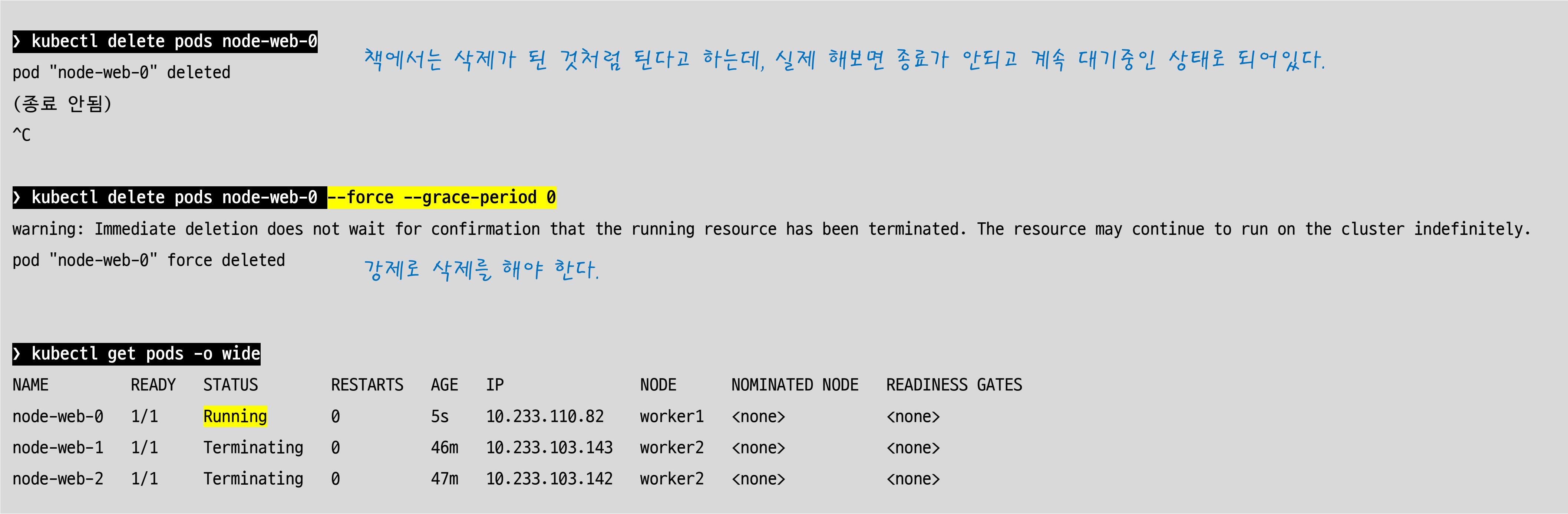
worker2 Node 전원을 꺼버렸다.

```
NAME         STATUS     ROLES                  AGE   VERSION   INTERNAL-IP       EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION     CONTAINER-RUNTIME
master-stg   Ready      control-plane,master   26d   v1.20.6   192.168.100.111   <none>        Ubuntu 20.04.2 LTS   5.4.0-73-generic   docker://19.3.15
worker1      Ready      <none>                 26d   v1.20.6   192.168.100.112   <none>        Ubuntu 20.04.2 LTS   5.4.0-73-generic   docker://19.3.15
worker2      NotReady   <none>                 26d   v1.20.6   192.168.100.113   <none>        Ubuntu 20.04.2 LTS   5.4.0-73-generic   docker://19.3.15

❯ kubectl get pods -o wide
```

책에서는 Unknown이 되었다가 Terminating 된다고 했는데, Ready 상태로 계속 나오다가 한 참 후에 Terminating으로 바뀌었다.

```
NAME         READY   STATUS        RESTARTS   AGE   IP               NODE      NOMINATED NODE   READINESS GATES
node-web-0   1/1     Terminating   0          27m   10.233.103.144   worker2   <none>           <none>
node-web-1   1/1     Terminating   0          28m   10.233.103.143   worker2   <none>           <none>
node-web-2   1/1     Terminating   0          28m   10.233.103.142   worker2   <none>           <none>
```

실습 : StatefulSet 장애 – Worker Node 오류 2/2



```
❯ kubectl delete pods node-web-0
pod "node-web-0" deleted
(종료 안됨)
^C
```

책에서는 삭제가 된 것처럼 된다고 하는데, 실제 해보면 종료가 안되고 계속 대기중인 상태로 되어있다.

```
❯ kubectl delete pods node-web-0 --force --grace-period 0
warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may continue to run on the cluster indefinitely.
pod "node-web-0" force deleted
```

강제로 삭제를 해야 한다.

```
❯ kubectl get pods -o wide
NAME         READY   STATUS        RESTARTS   AGE   IP              NODE      NOMINATED NODE   READINESS GATES
node-web-0   1/1     Running       0          5s    10.233.110.82   worker1   <none>           <none>
node-web-1   1/1     Terminating   0          46m   10.233.103.143  worker2   <none>           <none>
node-web-2   1/1     Terminating   0          47m   10.233.103.142  worker2   <none>           <none>
```

///