6th Week

여섯번째 뵙겠습니다 ?!

▷ 출석 체크도 한 번 해보시면 어떠세요?!

- https://modulabs.co.kr/
- 모두연 홈페이지 → 로그인 → 마이페이지 → 참여한 랩·풀잎 → 자세히 보기 → 내 풀잎스쿨 출석 확인하기

▷ Ground Rule

- 가급적 지각/결석 하지 않기
- 가급적 Camera 켜 놓고 수업 참여하기
- 가급적 적극적으로 참여하기
- 3시간이 넘더라도 배고프다고 화내지 않기
- Slack 잊지 않기
- 꼭 끝까지 함께하기

잡담 &
지난 수업 관련 이야기

///

Agenda

- ▷ [10m] make-friendship
- ▷ [20m] Flip - Deployment
- ▷ [30m] Wrap-Up & Hands-On
- ▷ [10m] Break-Time

- ▷ [20m] Flip - StatefulSet
- ▷ [30m] Wrap-Up & Hands-On

- ▷ [20m] Quiz

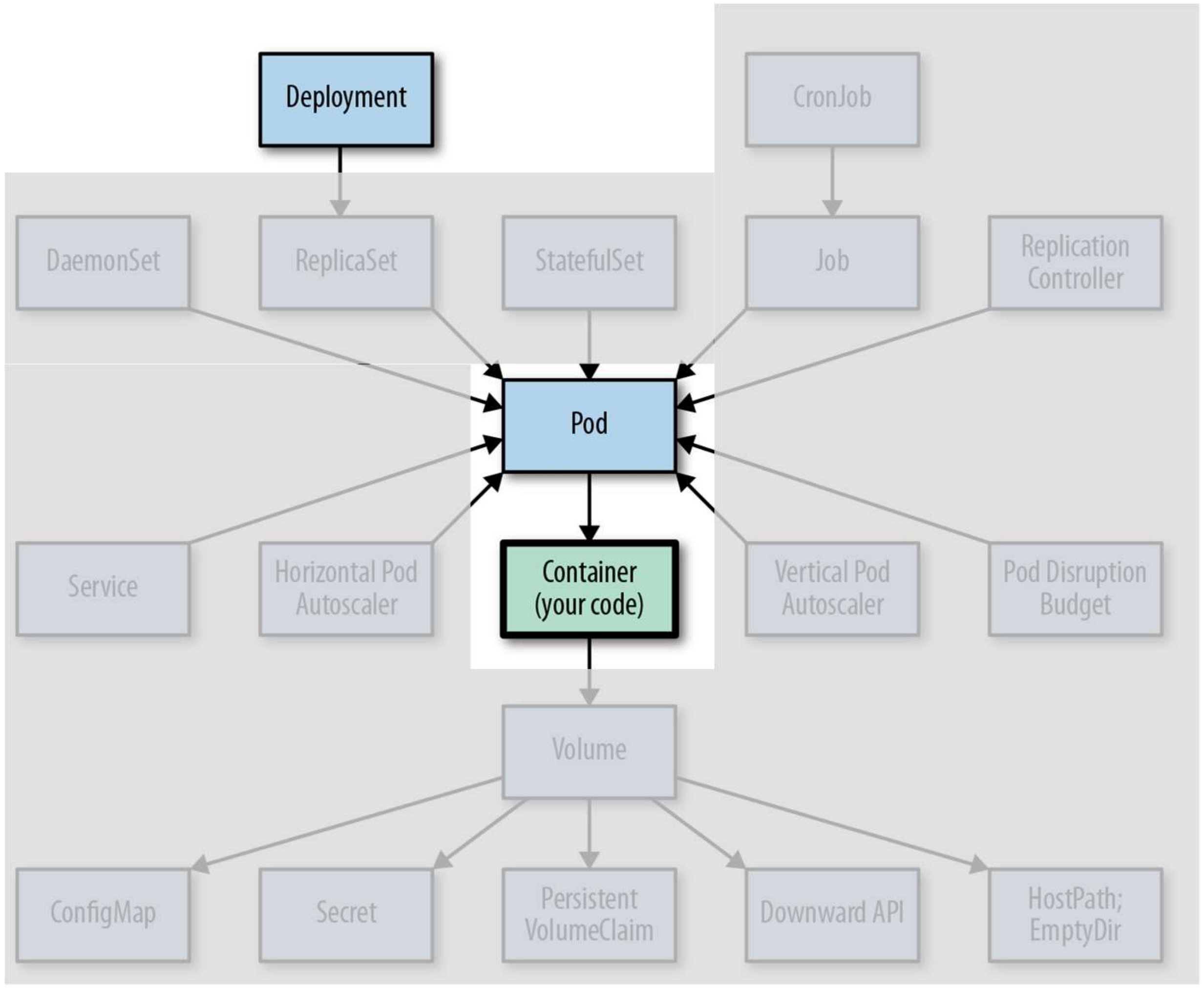
- ▷ [30m] MLOps 101

Kubernetes

Deployment

Today …

Deployment



///

Flip Learning

(Volume - Deployment)

김남형님

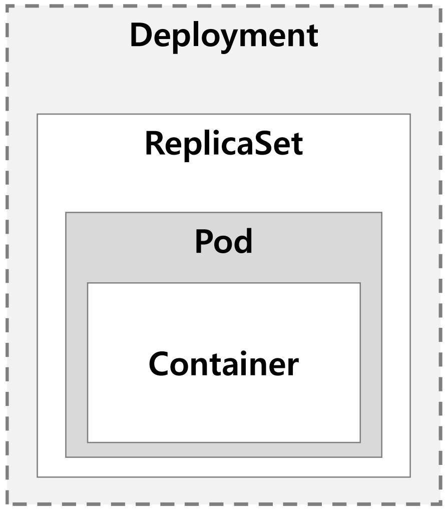
///

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/

Deployment

- 기본적인 YAML 구성은 ReplicaSet과 유사하다

dp-web-v1.yaml

```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
```

```
remote ❯ git clone https://github.com/whatwant-school/advanced-kubernetes.git
remote ❯ cd advanced-kubernetes

remote ❯ kubectl create -f ./06-week/Deployment/dp-web-v1.yaml

deployment.apps/dp-web created

remote ❯ kubectl get deployments -o wide

NAME     READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES                  SELECTOR
dp-web   3/3     3            3           19s   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-78f578d65c   3         3         3       48s   node-web     whatwant/node-web:1.0   app=node-web,pod-template-hash=78f578d65c


remote ❯ kubectl get pods -o wide

NAME                      READY   STATUS    RESTARTS   AGE   IP              NODE      NOMINATED NODE   READINESS
GATES
dp-web-78f578d65c-8xf69   1/1     Running   0          62s   10.233.103.57   worker2   <none>           <none>
dp-web-78f578d65c-hwchd   1/1     Running   0          62s   10.233.110.63   worker1   <none>           <none>
dp-web-78f578d65c-vjktk   1/1     Running   0          62s   10.233.103.56   worker2   <none>           <none>
```

Service

- LoadBalancer 만들어서 결과가 잘 나오는지 확인해보자.

svc-lb-web.yaml

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

spec:
  type: LoadBalancer

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

  selector:
    app: node-web
```

```
remote ❯ git clone https://github.com/whatwant-school/advanced-kubernetes.git
remote ❯ cd advanced-kubernetes

remote ❯ kubectl create -f ./06-week/Deployment/svc-lb-web.yaml

service/svc-lb 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        28d    <none>
svc-lb       LoadBalancer   10.233.51.99   192.168.100.240   80:31545/TCP   84m    app=node-web


remote ❯ curl -s http://192.168.100.240
You've hit dp-web-78f578d65c-vjktk

remote ❯ curl -s http://192.168.100.240
You've hit dp-web-78f578d65c-hwchd

remote ❯ curl -s http://192.168.100.240
You've hit dp-web-78f578d65c-hwchd

remote ❯ curl -s http://192.168.100.240
You've hit dp-web-78f578d65c-8xf69
```

///

Change Pods

- 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

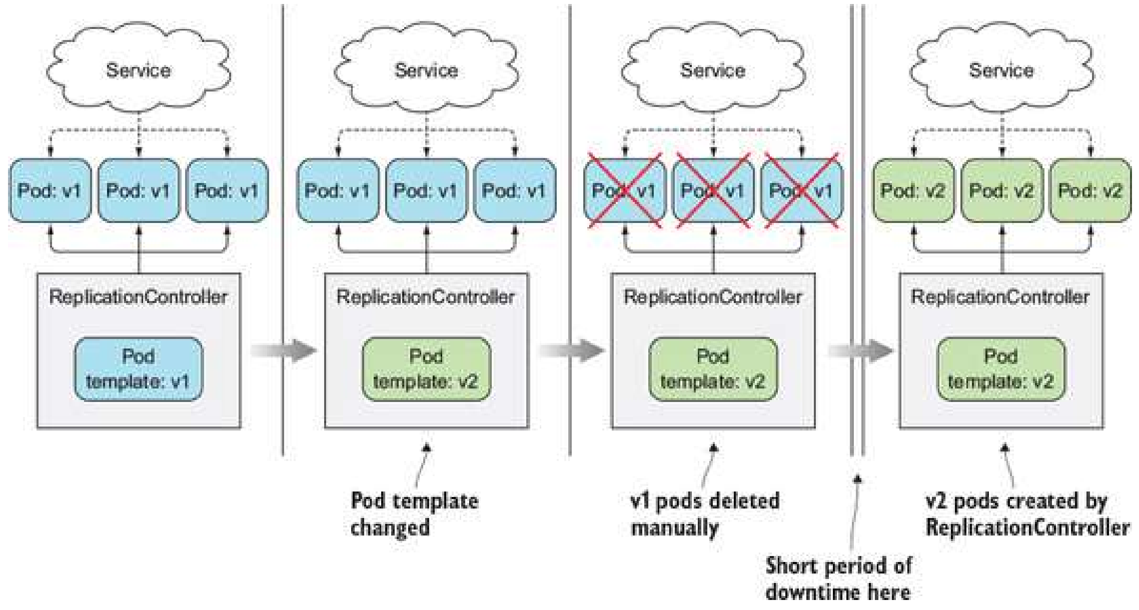
- Application의 변경(like version-up)이 필요한 경우 손쉽게 적용 가능

  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

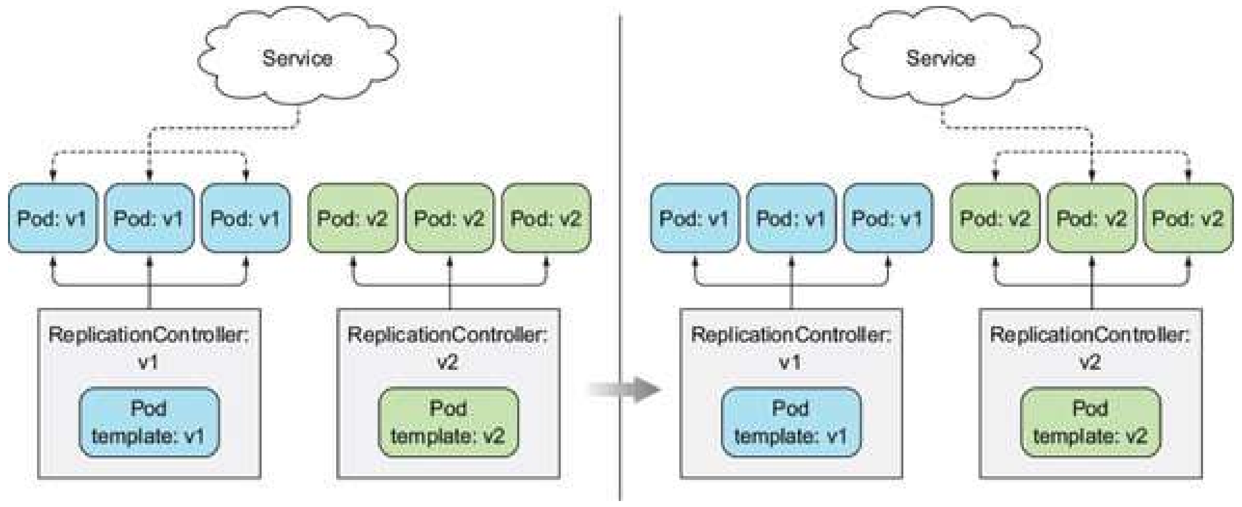
- 다운타임이 발생하지 않고 한 번에 여러 version의 application이 실행되는 것을 지원하는 경우

① 새로운 versio의 Template으로 신규 Pod 생성, 기존 versio은 지속 서비스 中

② 한 번에 Service를 신규 Pod를 바라보도록 전환

③ 전환 완료되면, 기존 Pod 삭제

= Blue-Green Deployment

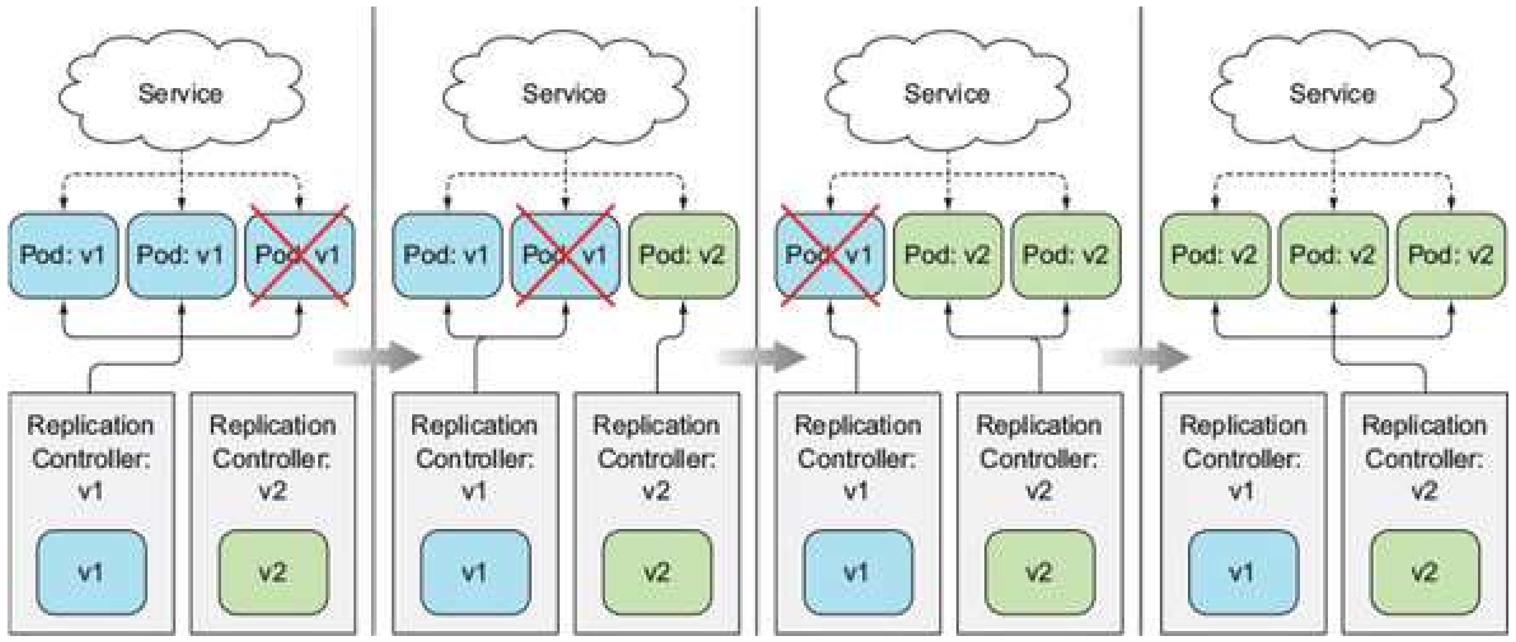


※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-9/29

#3. Rolling update - overview

- 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

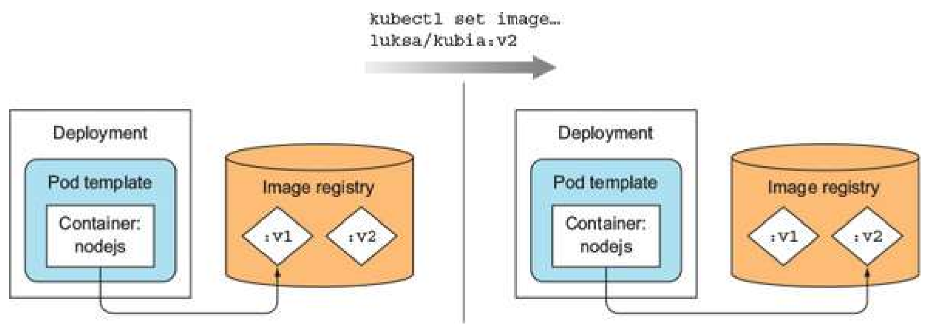
///

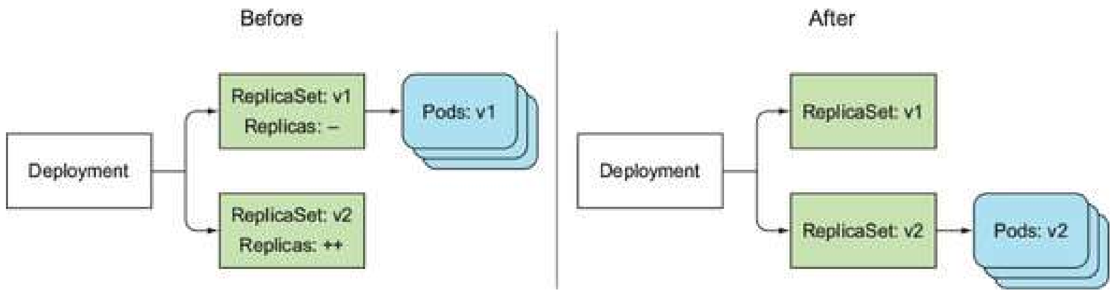
Kubernetes 리소스 수정 = Deployment 수정 방법

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

set image (Rolling update) - 1/2

- 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 실행해서 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 (Rolling update) - 2/2

- 앞에서 생성한 Deployment & Service를 활용해서 진행

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

You've hit dp-web-78f578d65c-vjktk
You've hit dp-web-78f578d65c-8xf69
You've hit dp-web-78f578d65c-8xf69
You've hit dp-web-78f578d65c-vjktk
You've hit dp-web-78f578d65c-vjktk
You've hit dp-web-78f578d65c-hwchd
You've hit dp-web-78f578d65c-8xf69
You've hit dp-web-78f578d65c-hwchd
You've hit dp-web-78f578d65c-vjktk
You've hit dp-web-78f578d65c-8xf69
You've hit dp-web-78f578d65c-8xf69
You've hit dp-web-78f578d65c-hwchd
You've hit dp-web-78f578d65c-vjktk
You've hit dp-web-78f578d65c-8xf69
You've hit dp-web-78f578d65c-8xf69
You've hit dp-web-64f47c76b8-snpdk (Ver2.0)
You've hit dp-web-64f47c76b8-snpdk (Ver2.0)
You've hit dp-web-78f578d65c-8xf69
You've hit dp-web-64f47c76b8-pkjvn (Ver2.0)
You've hit dp-web-64f47c76b8-wnw7m (Ver2.0)
You've hit dp-web-64f47c76b8-wnw7m (Ver2.0)
You've hit dp-web-64f47c76b8-pkjvn (Ver2.0)
You've hit dp-web-64f47c76b8-wnw7m (Ver2.0)
You've hit dp-web-64f47c76b8-snpdk (Ver2.0)
You've hit dp-web-64f47c76b8-pkjvn (Ver2.0)
```

```
remote ❯ kubectl get replicasets -o wide

NAME              DESIRED CURRENT READY AGE CONTAINERS IMAGES                 SELECTOR
dp-web-78f578d65c 3       3       3     48s node-web   whatwant/node-web:1.0 app=node-web,pod-template-hash=78f578d65c

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-64f47c76b8 3       3       3     45s  node-web   whatwant/node-web:2.0  app=node-web,pod-template-hash=64f47c76b8
dp-web-78f578d65c 0       0       0     19h  node-web   whatwant/node-web:1.0  app=node-web,pod-template-hash=78f578d65c

remote ❯ kubectl get pods -o wide

NAME                     READY   STATUS    RESTARTS   AGE   IP              NODE      NOMINATED NODE   READINESS GATES
dp-web-64f47c76b8-pkjvn  1/1     Running   0          56s   10.233.103.58   worker2   <none>           <none>
dp-web-64f47c76b8-snpdk  1/1     Running   0          59s   10.233.110.64   worker1   <none>           <none>
dp-web-64f47c76b8-wnw7m  1/1     Running   0          52s   10.233.110.65   worker1   <none>           <none>

remote ❯ kubectl rollout status deployment dp-web

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

rollout

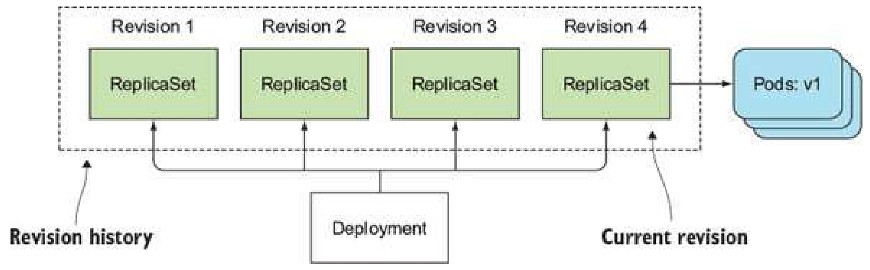
- `kubectl rollout` 명령어에 대해서 알아보자.

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

deployment.apps/dp-web
REVISION  CHANGE-CAUSE
1         <none>
2         <none>
```

<none>로 나온다.
제대로 기록되기 위해서는
`--record=true`를 붙여줘야 한다.

```
remote ❯ kubectl set image deployment dp-web \
node-web=whatwant/node-web:1.0 --record=true

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


remote ❯ kubectl rollout history deployment/dp-web

deployment.apps/dp-web
REVISION CHANGE-CAUSE
2        <none>
3        kubectl set image deployment dp-web node-web=whatwant/node-web:1.0 --record=true


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
```

※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-9/199



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

deployment.apps/dp-web
REVISION CHANGE-CAUSE
3        kubectl set image deployment dp-web node-web=whatwant/node-web:1.0 --record=true
4        kubectl set image deployment dp-web node-web=whatwant/node-web:2.0 --record=true


remote ❯ kubectl get replicasets

NAME                DESIRED   CURRENT   READY   AGE
dp-web-64f47c76b8   3         3         3       9m48s
dp-web-78f578d65c   0         0         0       10m


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

deployment.apps/dp-web rolled back
```

`--to-revision`을 붙이지 않으면
직전 version으로 간다.

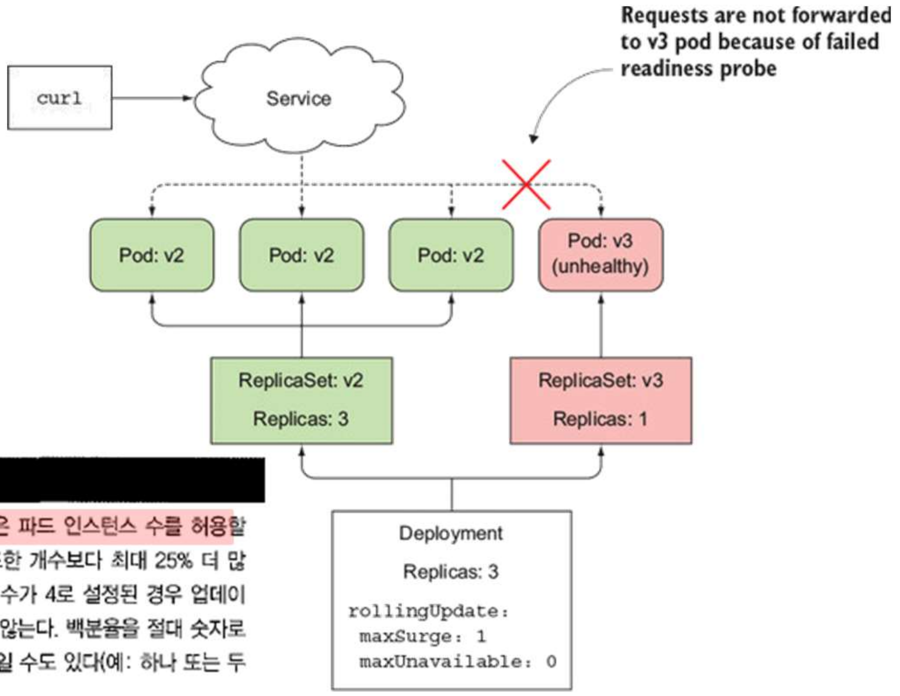
```
remote ❯ kubectl get replicasets

NAME                DESIRED   CURRENT   READY   AGE
dp-web-64f47c76b8   0         0         0       13m
dp-web-78f578d65c   3         3         3       14m
```

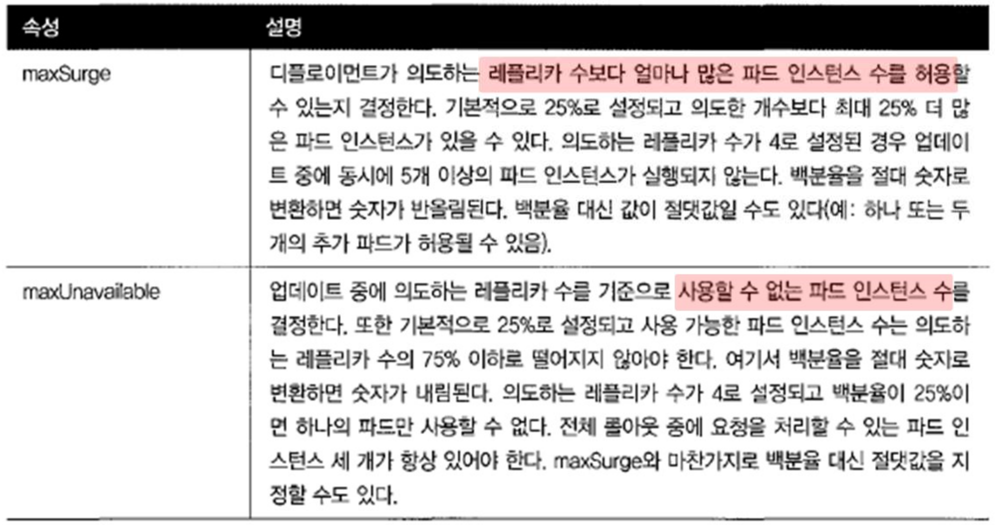
///

spec.strategy.RollingUpdate : 롤아웃 속도 제어 – 1/2

dp-web-strategy.yaml

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

spec:
  replicas: 3
  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
```



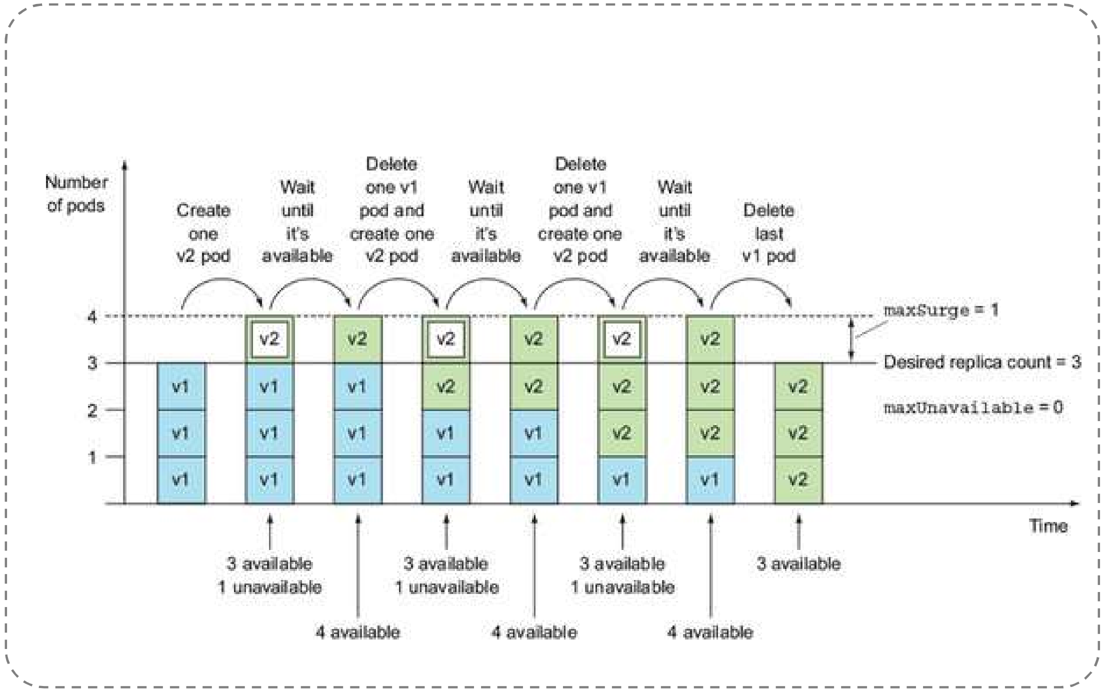


속성	설명
maxSurge	디플로이먼트가 의도하는 레플리카 수보다 얼마나 많은 파드 인스턴스 수를 허용할 수 있는지 결정한다. 기본적으로 25%로 설정되고 의도한 개수보다 최대 25% 더 많은 파드 인스턴스가 있을 수 있다. 의도하는 레플리카 수가 4로 설정된 경우 업데이트 중에 동시에 5개 이상의 파드 인스턴스가 실행되지 않는다. 백분율을 절대 숫자로 변환하면 숫자가 반올림된다. 백분율 대신 값이 절댓값일 수도 있다(예: 하나 또는 두 개의 추가 파드가 허용될 수 있음).
maxUnavailable	업데이트 중에 의도하는 레플리카 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.

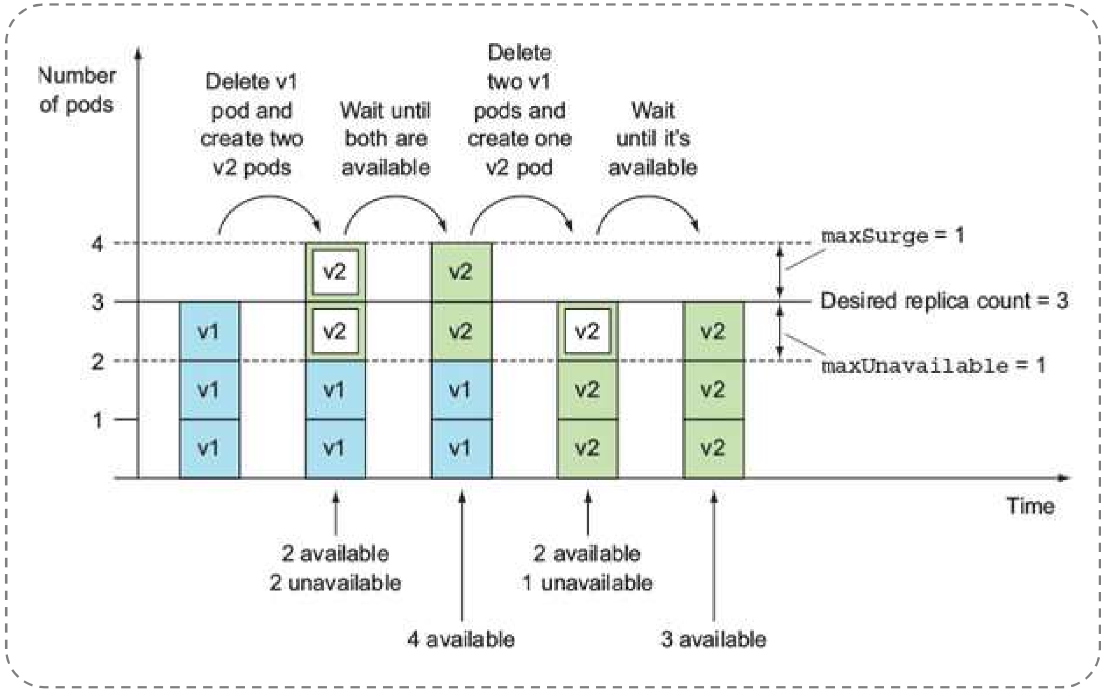
※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-9/262

※ 참고 : 마르코 룩샤, 『**Kubernetes IN ACTION**』, 강인호/황주필/이원기/임찬식 옮김-에이콘출판사/MANNING(2020), 419p

spec.strategy.RollingUpdate : 롤아웃 속도 제어 – 2/2



Number of pods
Create one v2 pod
Wait until it's available
Delete one v1 pod and create one v2 pod
Wait until it's available
Delete one v1 pod and create one v2 pod
Wait until it's available
Delete last v1 pod
maxSurge = 1
Desired replica count = 3
maxUnavailable = 0
Time
3 available 1 unavailable
4 available
3 available 1 unavailable
4 available
3 available 1 unavailable
4 available
3 available



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
maxSurge = 1
Desired replica count = 3
maxUnavailable = 1
Time
2 available 2 unavailable
4 available
2 available 1 unavailable
3 available

정상적으로 서비스 하고 있는 Pod가 최소한 몇 개가 되어야 하는지 …

동시에 실행되고 있는 Pod를 몇 개까지 감당할 수 있는 H/W 리소스를 갖고 있는지…

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

///

실수 방지 장치 = minReadySeconds & readinessProbe

dp-web-minreadysec.yaml

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

metadata:
  name: dp-web

spec:
  replicas: 3

  selector:
    matchLabels:
      app: node-web

  minReadySeconds: 10
  revisionHistoryLimit: 5
  progressDeadlineSeconds: 60

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

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

```yaml
    spec:
      containers:
      - image: whatwant/node-web:1.0
        name: node-web

        ports:
        - containerPort: 8080
          protocol: TCP

        imagePullPolicy: Always

        readinessProbe:
          httpGet:
            path: /
            port: 8080

          initialDelaySeconds: 5
          periodSeconds: 5
          successThreshold: 1
```

[minReadySeconds]

- 새로 배포된 컨테이너가 준비되기까지 대기할 시간 (기본값: 0초)
- Pod의 status 가 Ready가 될 때까지의 최소 대기 시간
- minReadySeconds로 설정된 시간 동안은 트래픽을 받지 않음
- minReadySeconds 이후 부터 pod의 READY를 확인하고 다음 단계로 진행

[revisionHistoryLimit]

- 되돌릴 수 있는 revision 개수 (기본값: 10)

[progressDeadlineSeconds]

- 지정된 시간이 초과되면 롤아웃이 자동으로 중단 (기본값: 10분)
- progressDeadlineSeconds를 경과해도 pod가 READY 상태가 되면
  rollout을 계속 수행한다.
- progressDeadlineSeconds는 반드시 minReadySeconds보다 커야 한다

///

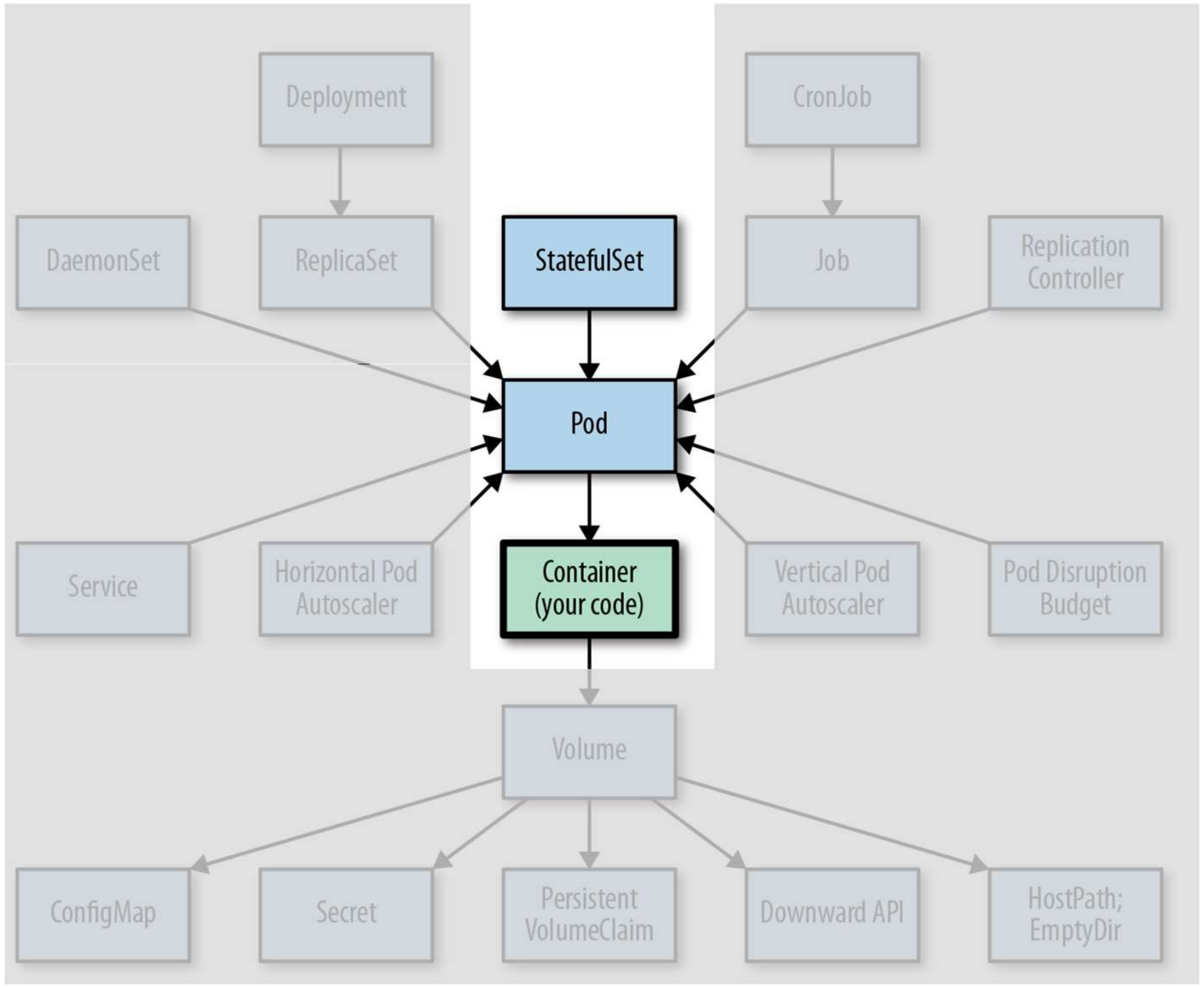
Break

///

StatefulSet

Today …

StatefulSet



///

Flip Learning

(Volume - StatefulSet)

이민준님

///

Why StatefulSet - 1/2

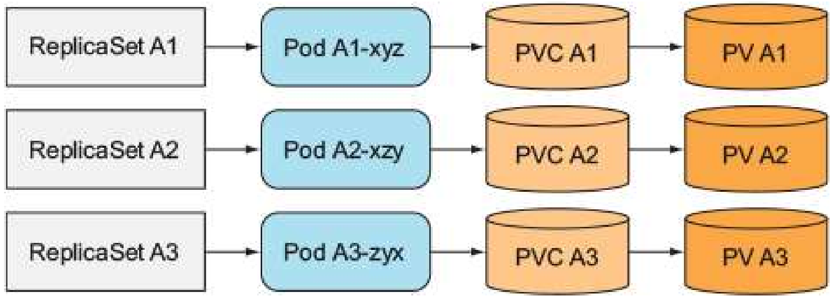
- Pod 인스턴스 별로 독립적인 저장공간을 갖도록 하려면,

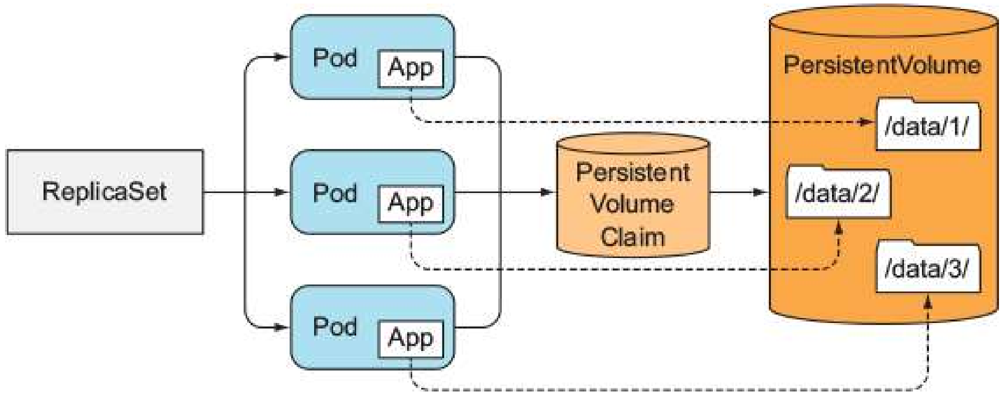
  . 수동으로 1개씩 Pod 생성

  . 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

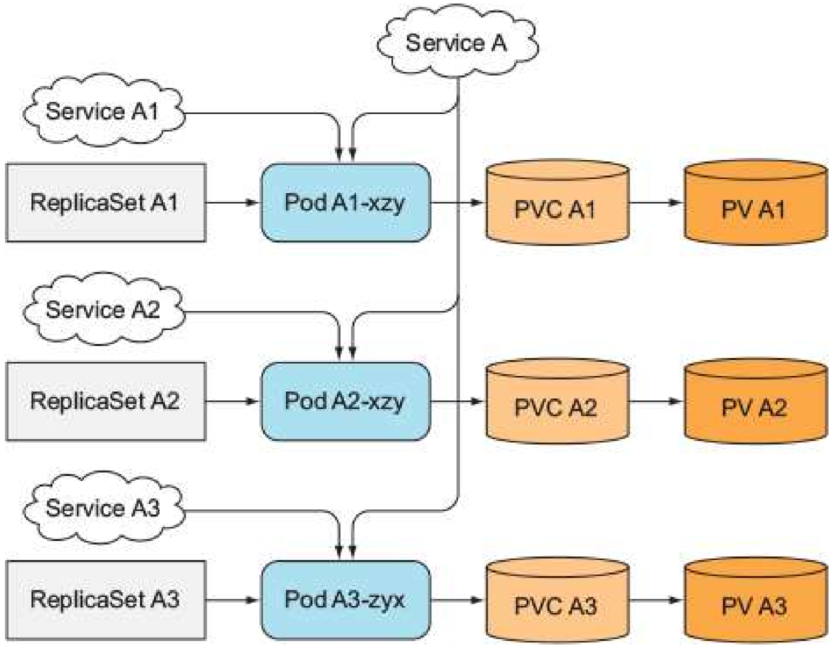
Why StatefulSet - 2/2

- 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 – 1/4

- 애완동물(Pet) vs 가축(Cattle)

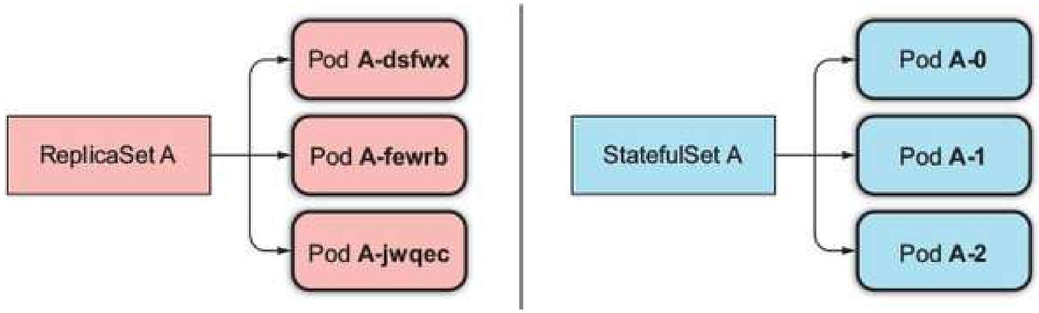
- StatefulSet

  . 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

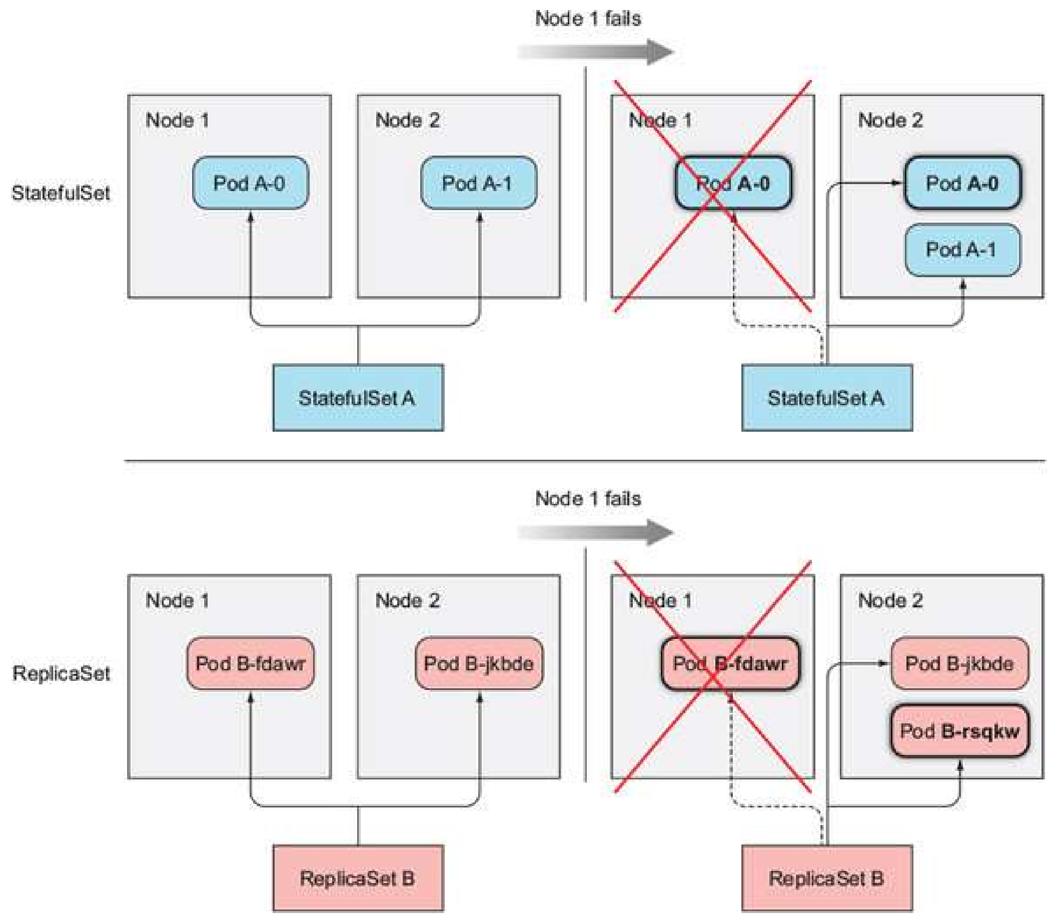
  . governing headless service : a-0.foo.default.svc.cluster.local



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

StatefulSet vs ReplicaSet – 2/4

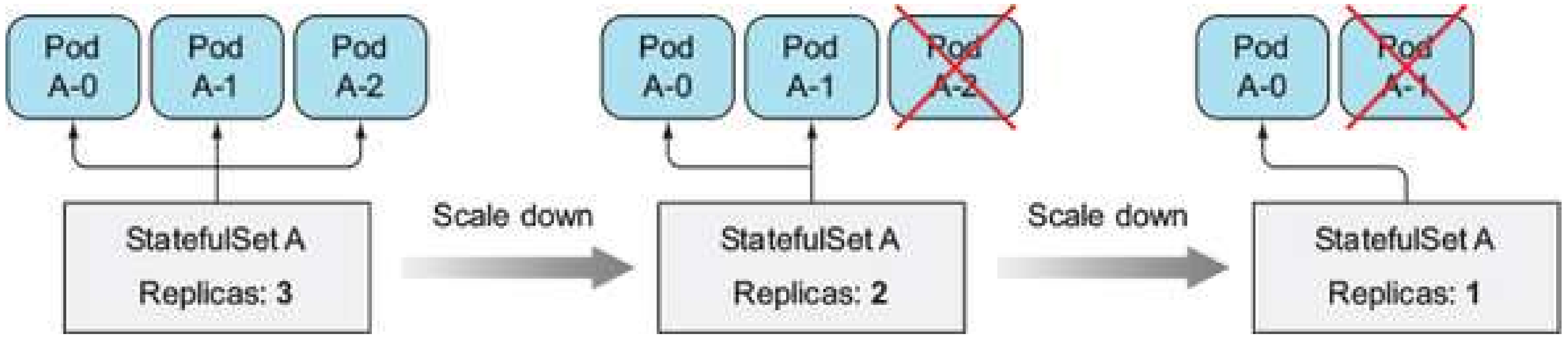
- Restart(Replace)

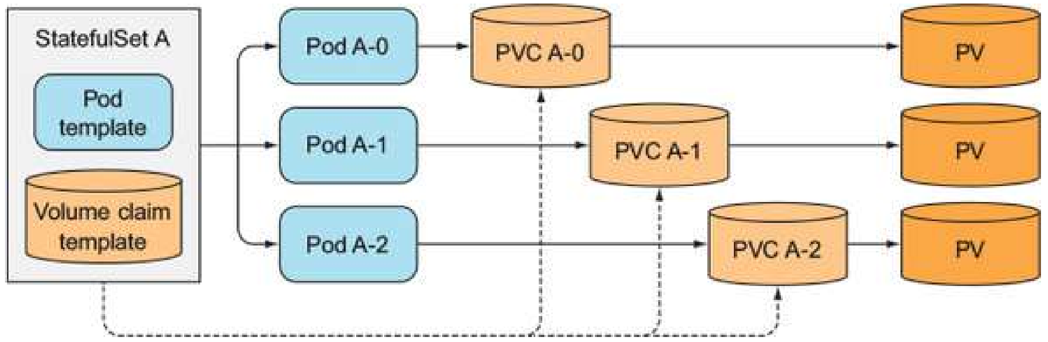


※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-10/61

StatefulSet vs ReplicaSet – 3/4

- Scaling



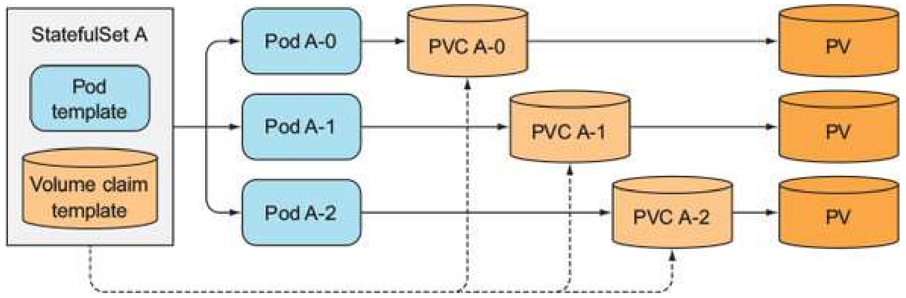


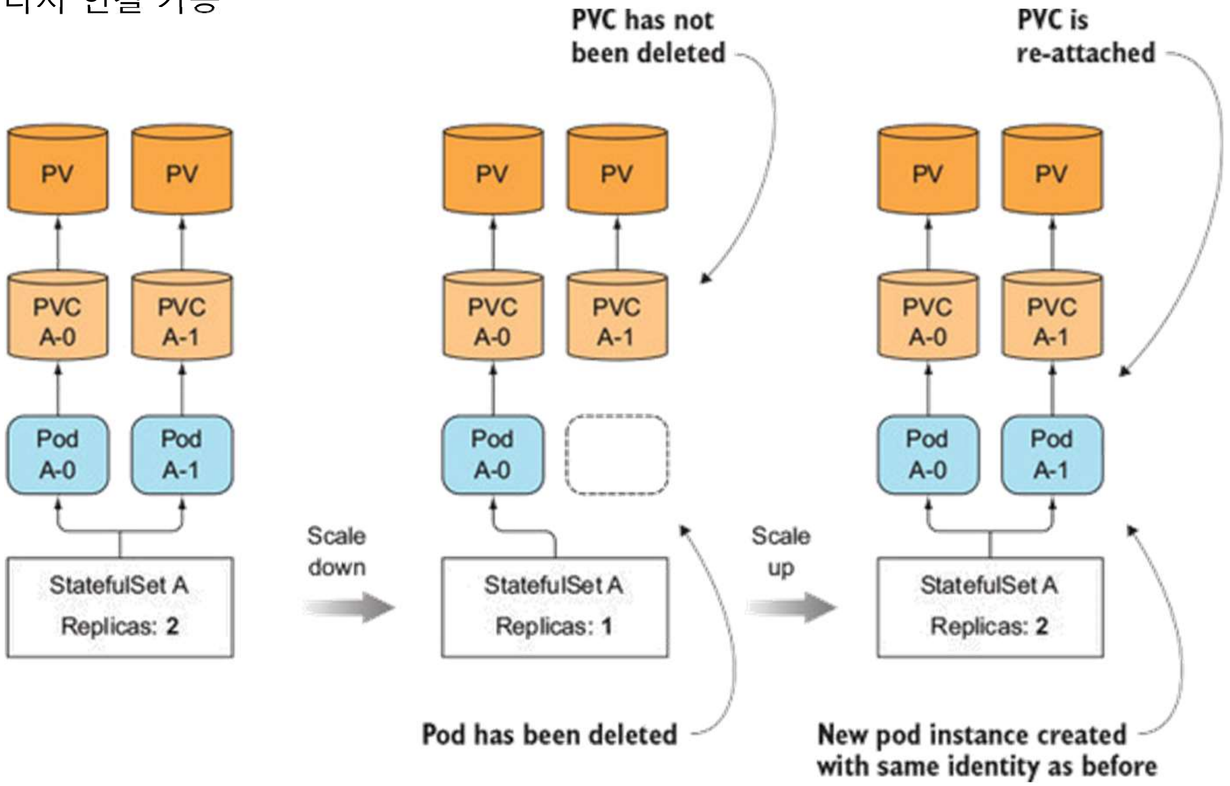
※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-10/61

StatefulSet vs ReplicaSet – 4/4

- Volume claim template

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





PVC has not been deleted
PVC is re-attached
PV
PVC A-0
PVC A-1
Pod A-0
Pod A-1
StatefulSet A Replicas: 2
Scale down
StatefulSet A Replicas: 1
Scale up
StatefulSet A Replicas: 2
Pod has been deleted
New pod instance created with same identity as before

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

///

StatefulSet 실습

Application

- StatefulSet 실습을 위한 application을 준비해보자

app.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");
  }
};
```

Dockerfile

```
FROM node:latest

ADD app.js /app.js

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

Application in Docker

- Docker 환경에서 application을 테스트 해보고, Docker Hub에 업로드 해보자

```
remote ❯ git clone https://github.com/whatwant-school/advanced-kubernetes.git
remote ❯ cd advanced-kubernetes/06-week/StatefulSet

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

remote ❯ mkdir /tmp/data

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

5d7f7b4858c6ed407d9c71718015256ae182692ce74d020787a61b15d89c47ed

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

You've hit 5d7f7b4858c6 (Ver3.0)
Data stored on this pod: No data posted yet


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

POST 형식으로 전달한 내용을 저장한다.

```
Data stored on pod 5d7f7b4858c6


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

You've hit 5d7f7b4858c6 (Ver3.0)
Data stored on this pod: Wow


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

PersistentVolume

persistentvolume.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

- 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
```

여러 개의 리소스를 정의할 때

`List` 형식을 사용할 수 있다.

```
remote ❯ git clone https://github.com/whatwant-school/advanced-kubernetes.git
remote ❯ cd advanced-kubernetes

remote ❯ kubectl create -f ./06-week/StatefulSet/persistentvolume.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                                   13s
pv-b   1Mi        RWO            Retain           Available                                   13s
pv-c   1Mi        RWO            Retain           Available                                   13s
```

Headless Service

- StatefulSet은 Headless Service가 필수 !!!

headless-service.yaml

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

spec:
  clusterIP: None

  selector:
    app: node-web

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

```
remote ❯ git clone https://github.com/whatwant-school/advanced-kubernetes.git
remote ❯ cd advanced-kubernetes

remote ❯ kubectl create -f ./06-week/StatefulSet/headless-service.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   32d   <none>
svc-web      ClusterIP   None         <none>        80/TCP    5s    app=node-web
```

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

StatefulSet

statefulset.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 ❯ git clone https://github.com/whatwant-school/advanced-kubernetes.git
remote ❯ cd advanced-kubernetes

remote ❯ kubectl create -f ./06-week/StatefulSet/statefulset.yaml
statefulset.apps/sf-web created

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 pods -o wide
NAME       READY   STATUS    RESTARTS   AGE    IP              NODE      NOMINATED NODE   READINESS GATES
sf-web-0   1/1     Running   0          113s   10.233.103.68   worker2   <none>           <none>
sf-web-1   1/1     Running   0          105s   10.233.110.75   worker1   <none>           <none>

remote ❯ kubectl get persistentvolumes
NAME   CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM                   STORAGECLASS   REASON   AGE
pv-a   1Mi        RWO            Retain           Bound       default/data-sf-web-0                           19m
pv-b   1Mi        RWO            Retain           Bound       default/data-sf-web-1                           19m
pv-c   1Mi        RWO            Retain           Available                                                   19m
```

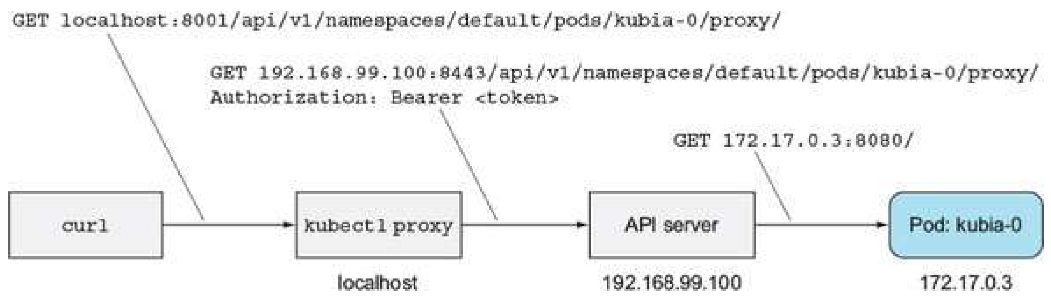
API Server & Proxy

- 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
localhost
API server
192.168.99.100
Pod: kubia-0
172.17.0.3

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

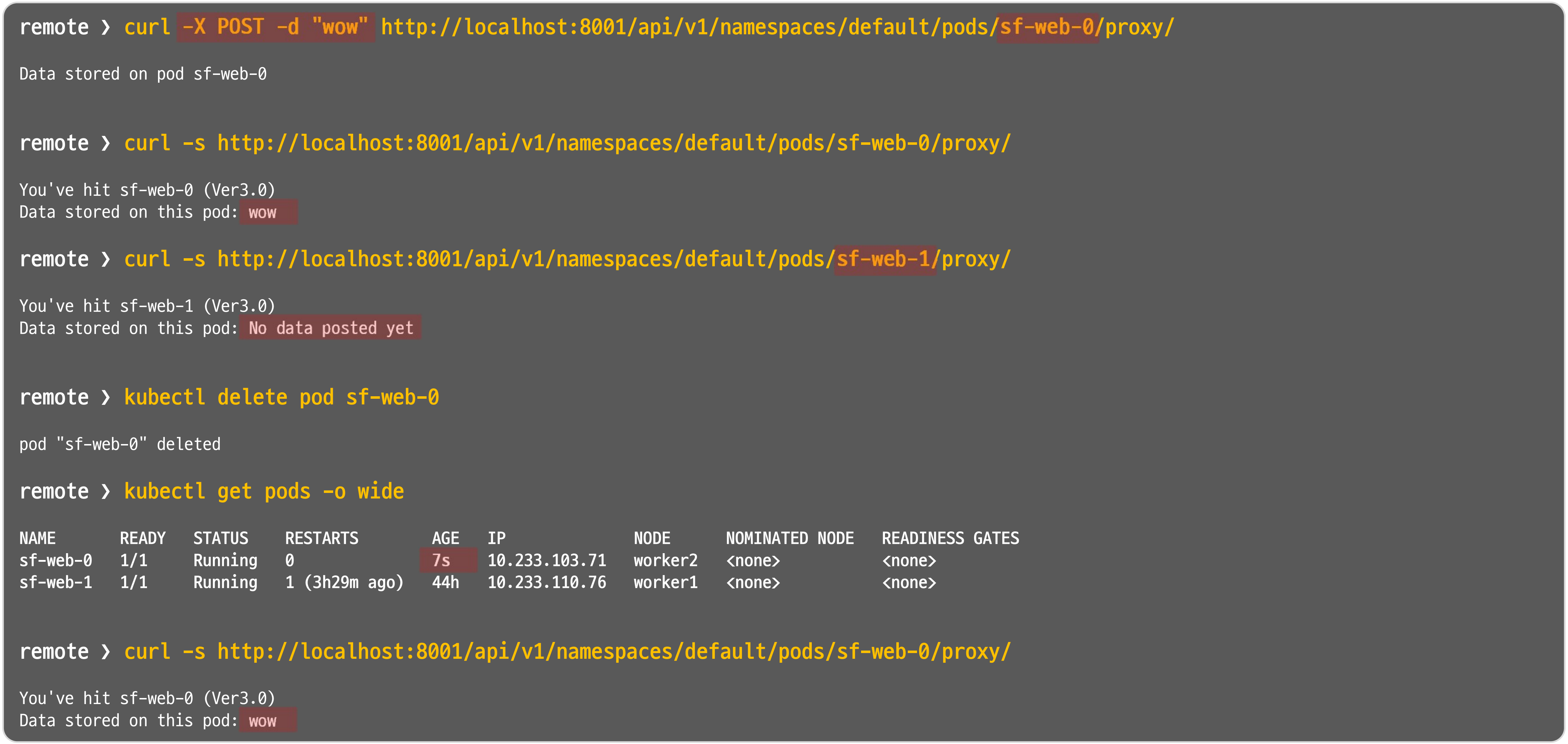
describe

- StatefulSet으로 생성된 Pod의 상세 정보를 살펴보자

```
remote ❯ kubectl describe pods sf-web-0
Name:         sf-web-0
Namespace:    default
...
Controlled By:  StatefulSet/sf-web
Containers:
  node-web:
    Container ID:   containerd://d97f1308f77fb44fdfe9feedd0b23db6b9730638481b97a64c47b8b2bba8febf
    Image:          whatwant/node-web:3.0
...
    Environment:    <none>
    Mounts:
      /var/data from data (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-ktp44 (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
  kube-api-access-ktp44:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
...
```

save file & delete pod

- StatefulSet으로 생성된 Pod 각각 구분되어 있는 volume 확인 및 Pod 재시작 했음에도 기존 Volume 그대로 연결됨을 확인



```
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               7s    10.233.103.71   worker2   <none>           <none>
sf-web-1   1/1     Running   1 (3h29m ago)   44h   10.233.110.76   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/

StatefulSet – 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: 50162
;; 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: c28f4d0ef55c8f0e (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-0.svc-web.default.svc.cluster.local.
svc-web.default.svc.cluster.local. 5 INSRV          0 50 80 sf-web-1.svc-web.default.svc.cluster.local.

;; ADDITIONAL SECTION:
sf-web-0.svc-web.default.svc.cluster.local. 5 IN A 10.233.103.71
sf-web-1.svc-web.default.svc.cluster.local. 5 IN A 10.233.110.76

;; Query time: 3 msec
;; SERVER: 169.254.25.10#53(169.254.25.10)
;; WHEN: Wed Feb 23 17:18:53 UTC 2022
;; MSG SIZE  rcvd: 380

pod "srvlookup" deleted
```

1회성으로 명령어를 실행하기 위한 사용법

///

new app - 1/2

app.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");
```

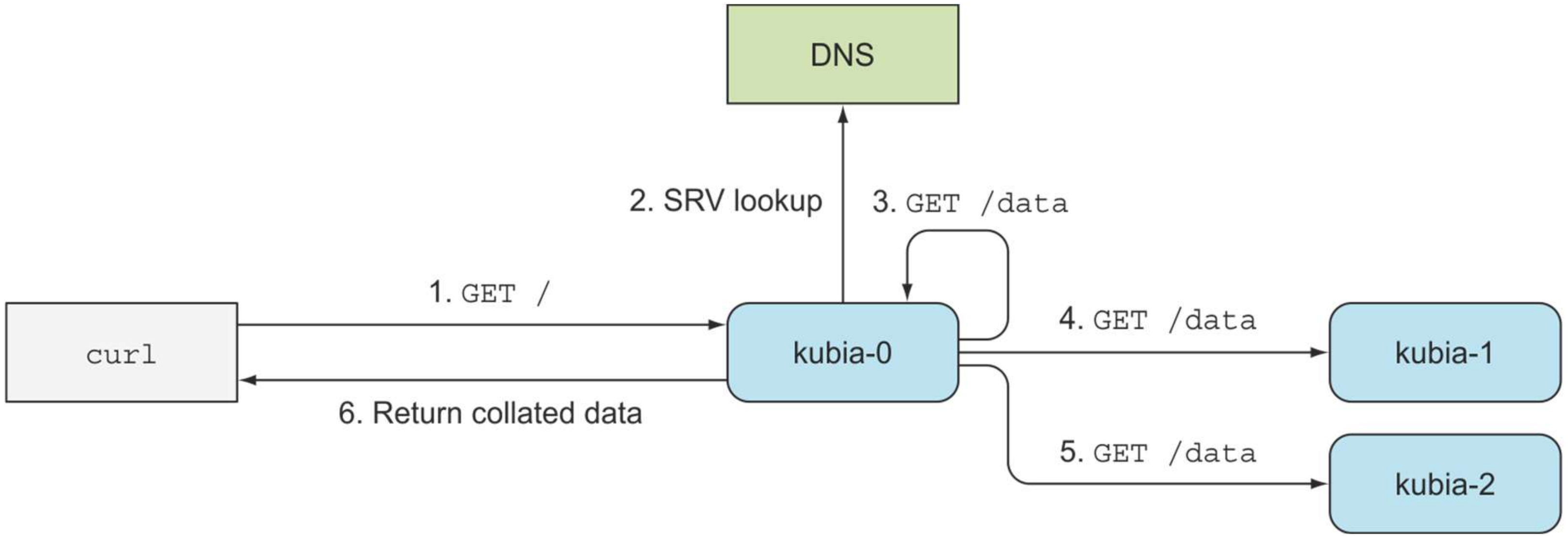
```
      dns.resolveSrv(serviceName, function (err, addresses) {
        if (err) {
          response.end("Could not look up DNS SRV records: " + err);
          return;
        }

        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);
```

new app - 2/2

Dockerfile

```
FROM node:latest

ADD app.js /app.js

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

```
remote ❯ git clone https://github.com/whatwant-school/advanced-kubernetes.git
remote ❯ cd advanced-kubernetes

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

///

Service

- Headless Service가 아닌 Load Balancing이 되는 Service를 구성해보자

lb-service.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
```

3개의 Pod가

골고루(?) 선택되는 것을

확인해볼 수 있다.

```
remote ❯ git clone https://github.com/whatwant-school/advanced-kubernetes.git
remote ❯ cd advanced-kubernetes

remote ❯ kubectl create -f ./06-week/StatefulSet/lb-service.yaml

service/svc-lb-web created

remote ❯ kubectl get services

NAME         TYPE          CLUSTER-IP     EXTERNAL-IP      PORT(S)        AGE
kubernetes   ClusterIP     10.233.0.1     <none>           443/TCP        33d
svc-lb-web   LoadBalancer  10.233.7.237   192.168.100.240  80:31580/TCP   3m22s
svc-web      ClusterIP     None           <none>           80/TCP         24h
```

```
remote ❯ curl -s http://192.168.100.240
You've hit sf-web-2
Data stored in the cluster:
- sf-web-1.svc-web.default.svc.cluster.local: wow
- sf-web-2.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 "hahaha" http://192.168.100.240
Data stored on pod sf-web-1


remote ❯ curl -s http://192.168.100.240
You've hit sf-web-2
Data stored in the cluster:
- sf-web-2.svc-web.default.svc.cluster.local: No data posted yet
- sf-web-1.svc-web.default.svc.cluster.local: hahaha
- sf-web-0.svc-web.default.svc.cluster.local: No data posted yet

remote ❯ curl -X POST -d "what" http://192.168.100.240
Data stored on pod sf-web-0


remote ❯ curl -s http://192.168.100.240
You've hit sf-web-1
Data stored in the cluster:
- sf-web-1.svc-web.default.svc.cluster.local: hahaha
- sf-web-0.svc-web.default.svc.cluster.local: what
- sf-web-2.svc-web.default.svc.cluster.local: No data posted yet
```

///

https://kahoot.it/

[Score]

이민준 (10)
김남형 (6)
이혜정 (4)
박남준 (3)
김상호 (2)
이원준 (2)
정현찬 (1)
김정은 (1)

///

1	김상호		
2	남상대		
3	최원준		
4	정현찬	3주차	ReplicaSet/DaemonSet/Job/CronJob
5	이혜정	3주차	ClusterIP/NodePort/ExternalName
6	박남준	4주차	Ingress
7	김언동	2주차	Pods & Namespace
8	김남형	5주차 6주차	ConfigMap/Secret/downwardAPI Deployment
9	이민준	6주차	StatefulSet
10	이원준	5주차	emptyDir/hostPath/PV
11	김정은		

///

뜬금없이 MLOps

- https://speakerdeck.com/mlopskr/mlops-cuncu-jeongug-sidae-jeongri-byeonseongyun