Managing Kubernetes

2021-04-23
written by whatwant

Agenda

Chapter1. Kubernetes Overview

1주차: **Docker and Kubernetes**

Chapter2. Kubernetes Core

2주차: **Environment & POD**

3주차: **Replication and other controllers**

4주차: **Services & Volumes**

5주차: **ConfigMaps and Secrets & Kubernetes REST API**

6주차: **Deployment & StatefulSet**

7주차: **Summary & HandsOn**

Chapter3. Kubernetes Managing

8주차: **Authentication and User Management & Authorization & Admission Control**

9주차: **Networking**

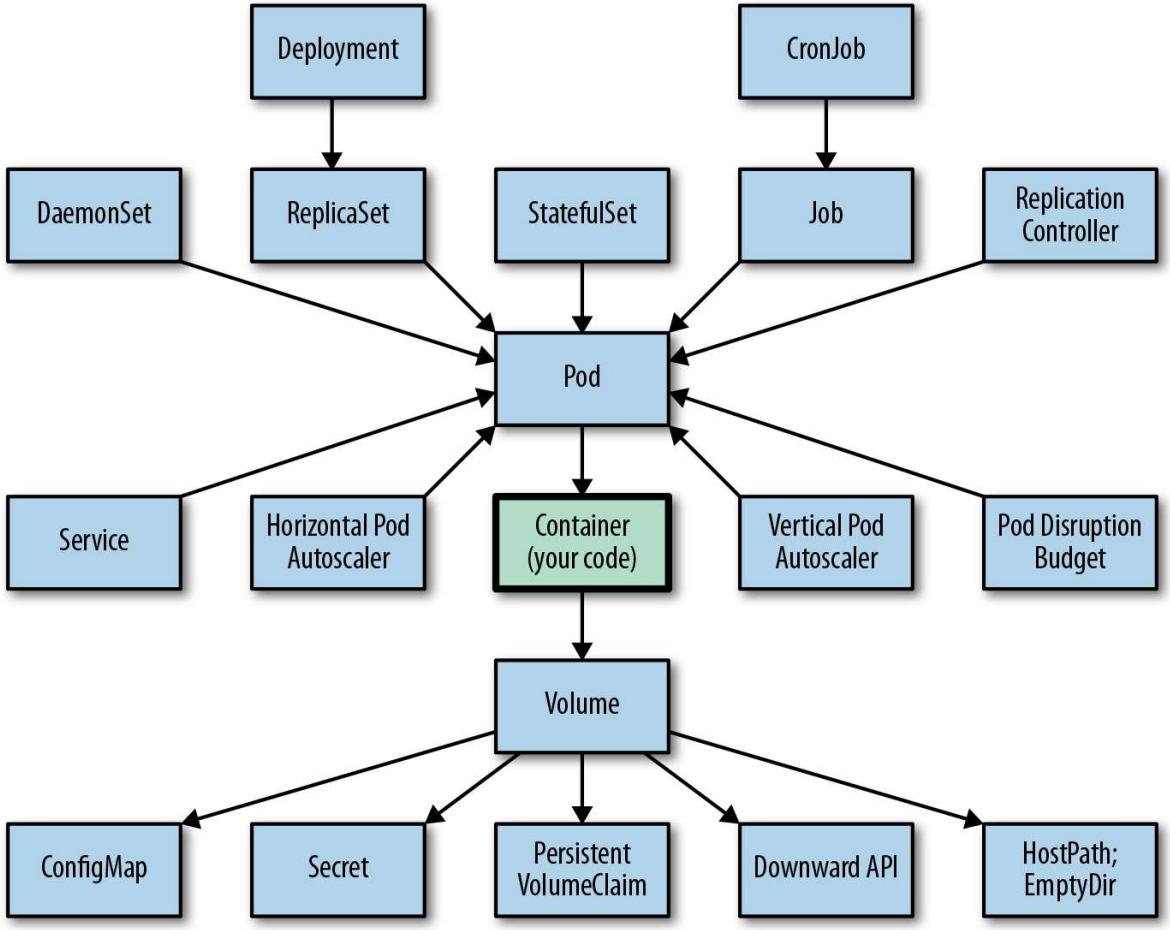
10주차: **Monitoring**

11주차: **Disaster Recovery**

※ 참고 : https://home.modulabs.co.kr/product/managing-kubernetes/

3 week
Replication and other controllers

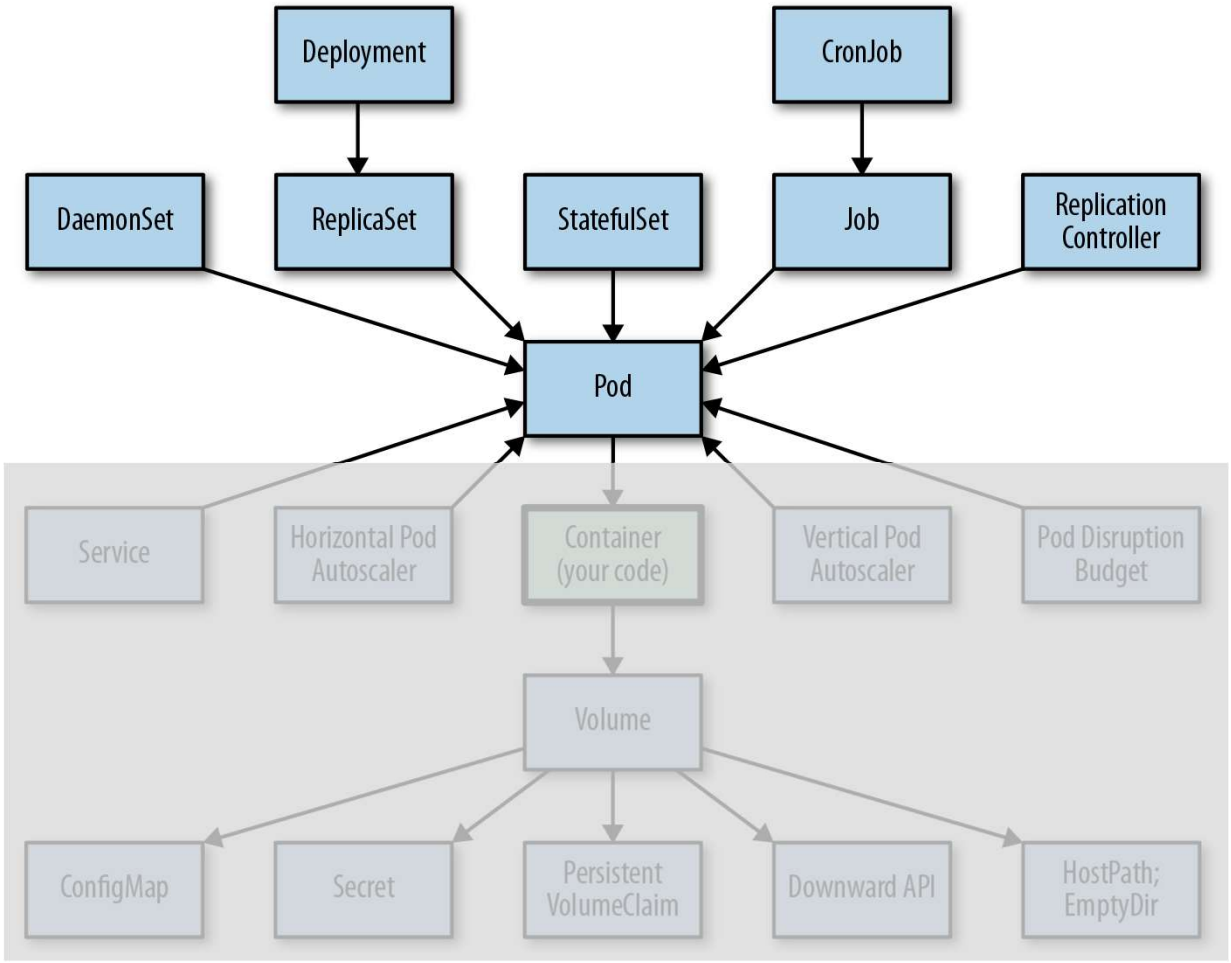
Kubernetes concepts for developers



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

Today ...

- livenessProbe
- ReplicationController
- ReplicaSet
- DaemonSet
- Job
- CronJob



livenessProbe – 1/3

- Pod의 spec에 각 container의 livenessProbe를 지정할 수 있음
  . **HTTP GET Probe**: HTTP GET 요청의 응답 코드를 확인 (2xx, 3xx이면 성공)
  . **TCP Socket Probe**: TCP 연결 성공 여부
  . **Exec Probe**: 명령을 실행하고 exit code가 0이면 성공
- Probe가 실패하면 container를 재실행

liveness-probe.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: kubia-liveness
spec:
  containers:
  - image: luksa/kubia-unhealthy
    name: kubia
    livenessProbe:
      httpGet:
        path: /
        port: 8080
      initialDelaySeconds: 15
```

```
❯ ls -al
total 4.0K
-rw-rw-r-- 1 whatwant whatwant 226  4월 23 22:12 liveness-probe.yaml


❯ kubectl create -f liveness-probe.yaml
pod/kubia-liveness created


❯ kubectl get pods -w
NAME             READY   STATUS              RESTARTS   AGE
kubia-liveness   0/1     ContainerCreating   0          14s
kubia-liveness   1/1     Running                 0          19s
kubia-liveness   1/1     Running                 1          2m22s
kubia-liveness   1/1     Running                 2          4m21s
kubia-liveness   1/1     Running                 3          6m22s
kubia-liveness   1/1     Running                 4          8m22s
^C%
```

livenessProbe – 2/3

- **initialDelaySeconds**: App 실행에 소요되는 시간을 고려해서 값 결정 필요 (default: 0)

  . 1초 이상 응답이 없으면(timeout=1s) 실패로 간주하며 10초 주기로 확인

  . liveness가 3번 실패하면 container에 crash 발생한 것으로 간주하고 restart

```
❯ kubectl describe pod kubia-liveness
...
Events:
  Type     Reason     Age                   From               Message
  ----     ------     ----                  ----               -------
  Normal   Scheduled  30m                   default-scheduler  Successfully assigned default/kubia-liveness to worker1
  Normal   Pulled     30m                   kubelet            Successfully pulled image "luksa/kubia-unhealthy" in 17.390380214s
  Normal   Pulled     28m                   kubelet            Successfully pulled image "luksa/kubia-unhealthy" in 2.486105876s
  Normal   Created    26m (x3 over 30m)     kubelet            Created container kubia
  Normal   Started    26m (x3 over 30m)     kubelet            Started container kubia
  Normal   Pulled     26m                   kubelet            Successfully pulled image "luksa/kubia-unhealthy" in 2.435700445s
  Normal   Killing    25m (x3 over 29m)     kubelet            Container kubia failed liveness probe, will be restarted
  Normal   Pulled     24m                   kubelet            Successfully pulled image "luksa/kubia-unhealthy" in 2.437774548s
  Normal   Pulling    20m (x6 over 30m)     kubelet            Pulling image "luksa/kubia-unhealthy"
  Warning  Unhealthy  5m40s (x25 over 29m)  kubelet            Liveness probe failed: HTTP probe failed with statuscode: 500
  Warning  BackOff    50s (x49 over 16m)    kubelet            Back-off restarting failed container
```

livenessProbe – 3/3

[효과적인 livenessProbe 구성]

- **특정 URL 경로**(e.g, /health)에 요청하도록 Probe를 구성

  . /health에서 모든 주요 구성 요소가 살아 있는지 확인하도록 로직 구성

- Probe는 application의 **내부만 체크**하고 외부 요인의 영향을 받지 않도록 해야 함

  . database 장애로 인해 front-end의 Probe가 실패해서는 안된다

- Probe를 가볍게 유지

  . **1초 내에 완료**되어야 한다.

- Probe에 재시도 루프를 구현하지 마라 (=중복)

- container 레벨의 livenessProbe는 노드의 Kubelet이 수행

  . 마스터는 관여하지 않음

- Node crash로 인해 중단된 모든 Pod를 복구하는 것은 master의 몫

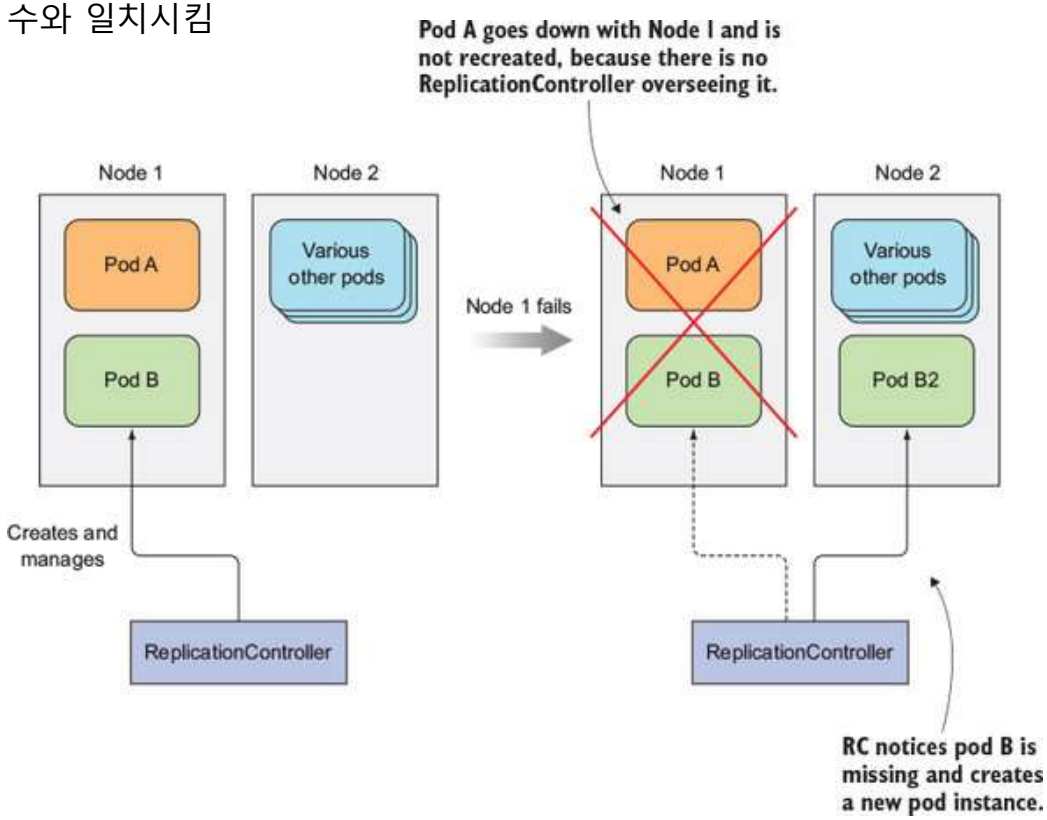
ReplicationController - 1/6

- Pod가 항상 실행되도록 보장하는 리소스

  . 어떤 이유로든 Pod가 사라지면 사라진 Pod를 감지해 교체 Pod를 생성

- 지속적으로 실행 중인 Pod 목록을 모니터링하고 선언된 수와 일치시킴

  . 너무 많은 Pod가 실행 중이면 줄이고, 적으면 늘림



※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-4/79

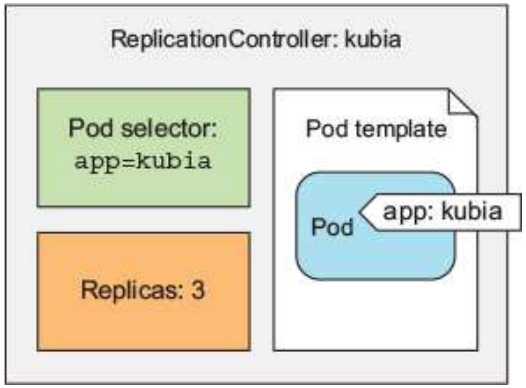
ReplicationController - 2/6

- ReplicationController 세 가지 필수 요소

  . **Pod Selector**: ReplicationController 범위에 있는 Pod 결정

  . **Replicas**: Pod가 실행되어야 하는 수

  . **Pod template**: 새로운 Pod replica를 만들 때 사용



replication-controller.yaml

```
apiVersion: v1
kind: ReplicationController

metadata:
  name: kubia

spec:
  replicas: 3

  selector:
    app: kubia

  template:
    metadata:
      labels:
        app: kubia

    spec:
      containers:
      - name: kubia
        image: luksa/kubia
        ports:
        - containerPort: 8080
```

※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-4/92

ReplicationController - 3/6

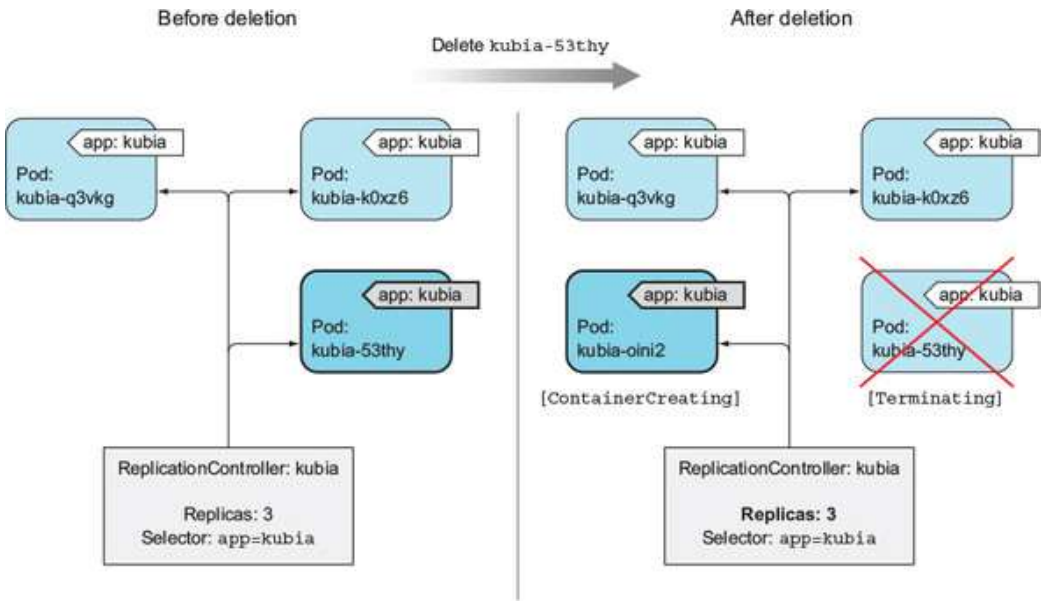
```
❯ kubectl create -f replication-controller.yaml
replicationcontroller/kubia created

❯ kubectl get pods -w
NAME          READY   STATUS    RESTARTS   AGE
kubia-kl546   0/1     Pending   0          0s
kubia-r2wq7   0/1     Pending   0          0s
kubia-v9rhr   0/1     Pending   0          0s
kubia-kl546   0/1     ContainerCreating   0          0s
kubia-r2wq7   0/1     ContainerCreating   0          0s
kubia-v9rhr   0/1     ContainerCreating   0          0s
kubia-r2wq7   1/1     Running             0          4s
kubia-kl546   1/1     Running             0          6s
kubia-v9rhr   1/1     Running             0          9s
^C%

❯ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
kubia-kl546   1/1     Running   0          32s
kubia-r2wq7   1/1     Running   0          32s
kubia-v9rhr   1/1     Running   0          32s

❯ kubectl get replicationcontrollers
NAME    DESIRED   CURRENT   READY   AGE
kubia   3         3         3       3m45s
```

```
❯ kubectl delete pod kubia-kl546
pod "kubia-kl546" deleted

❯ kubectl get pods -w
NAME          READY   STATUS    RESTARTS   AGE
kubia-kl546   1/1     Running   0          9m27s
kubia-r2wq7   1/1     Running   0          9m27s
kubia-v9rhr   1/1     Running   0          9m27s
kubia-kl546   1/1     Terminating   0          9m50s
kubia-ds578   0/1     Pending       0          0s
kubia-ds578   0/1     ContainerCreating   0          0s
kubia-ds578   1/1     Running             0          3s
kubia-kl546   0/1     Terminating         0          10m
^C%

❯ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
kubia-ds578   1/1     Running   0          62s
kubia-r2wq7   1/1     Running   0          10m
kubia-v9rhr   1/1     Running   0          10m

❯ kubectl get replicationcontrollers
NAME    DESIRED   CURRENT   READY   AGE
kubia   3         3         3       13m
```

ReplicationController - 4/6

- label을 변경해 ReplicationController의 관리에서 벗어난 Pod가 발생하면 복구를 시작

  . 특정 Pod에서 버그가 발생했을 경우 label을 변경해 서비스에서 제외하고 디버깅 용도로 활용




```
❯ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
kubia-ds578   1/1     Running   0          39m
kubia-r2wq7   1/1     Running   0          49m
kubia-v9rhr   1/1     Running   0          49m

❯ kubectl label pod kubia-ds578 app=foo --overwrite
pod/kubia-ds578 labeled

❯ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
kubia-ds578   1/1     Running   0          43m
kubia-lhsdb   1/1     Running   0          23s
kubia-r2wq7   1/1     Running   0          53m
kubia-v9rhr   1/1     Running   0          53m
```

※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-4/141

ReplicationController - 5/6

- kubectl edit: 리소스 업데이트

  . Template 변경: 기존 Pod에는 영향을 미치지 않으며 새로 생성되는 Pod에만 적용

  . Replicas 변경: 변경된 replica count로 수평 Pod 스케일링

  . Selector 변경: 모든 Pod가 범위를 벗어나게 되며 새로운 Pod를 생성

```
❯ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
kubia-lhsdb   1/1     Running   0          20m
kubia-r2wq7   1/1     Running   0          73m
kubia-v9rhr   1/1     Running   0          73m

❯ export KUBE_EDITOR=nano

❯ kubectl edit replicationcontrollers kubia
replicationcontroller/kubia edited
```

```
❯ kubectl get pods
NAME          READY   STATUS        RESTARTS   AGE
kubia-lhsdb   1/1     Terminating   0          20m
kubia-r2wq7   1/1     Running       0          73m
kubia-v9rhr   1/1     Running       0          73m

❯ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
kubia-r2wq7   1/1     Running   0          74m
kubia-v9rhr   1/1     Running   0          74m
```

ReplicationController - 6/6

- Pod도 함께 삭제할지 말지 결정할 수 있음 (cascade 옵션)

  . default: --cascade=true

```
❯ kubectl create -f replication-controller.yaml
replicationcontroller/kubia created

❯ kubectl get replicationcontrollers
NAME    DESIRED   CURRENT   READY   AGE
kubia   3         3         3       28s

❯ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
kubia-59p6f   1/1     Running   0          32s
kubia-jtdjk   1/1     Running   0          32s
kubia-xx9d7   1/1     Running   0          32s
```

```
❯ kubectl delete replicationcontrollers kubia --cascade=false
warning: --cascade=false is deprecated (boolean value) and can be replaced with --cascade=orphan.
replicationcontroller "kubia" deleted

❯ kubectl get replicationcontrollers
No resources found in default namespace.

❯ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
kubia-59p6f   1/1     Running   0          72s
kubia-jtdjk   1/1     Running   0          72s
kubia-xx9d7   1/1     Running   0          72s
```

ReplicaSet - 1/2

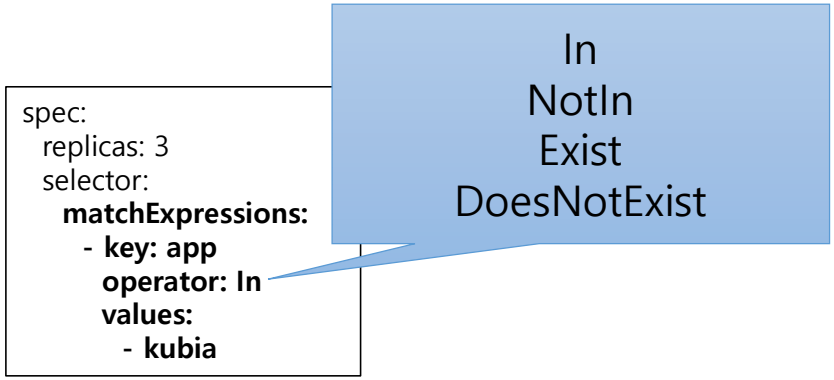
- ReplicationController와 똑같이 동작

  . 여기에 더해 selector에서 풍부한 표현식을 사용할 수 있음

  . 특정 키가 있는 label을 갖는 Pod를 매칭 (**matchLabels** / **matchExpressions**)

  . label 조건을 Or/And로 정의

- 일반적으로 직접 사용하지는 않고 Deployment 리소스를 생성할 때 자동으로 생성됨



ReplicaSet - 2/2

replicaset-matchlabels.yaml

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

metadata:
  name: rc-labels

spec:
  replicas: 3
  selector:
    matchLabels:
      app: kubia

  template:
    metadata:
      labels:
        app: kubia
    spec:
      containers:
      - name: kubia
        image: luksa/kubia
        ports:
        - containerPort: 8080
```

replicaset-matchexpressions.yaml

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

metadata:
  name: rc-expressions

spec:
  replicas: 3
  selector:
    matchExpressions:
      - key: app
        operator: In
        values:
          - kubia

  template:
    metadata:
      labels:
        app: kubia
    spec:
      containers:
      - name: kubia
        image: luksa/kubia
        ports:
        - containerPort: 8080
```

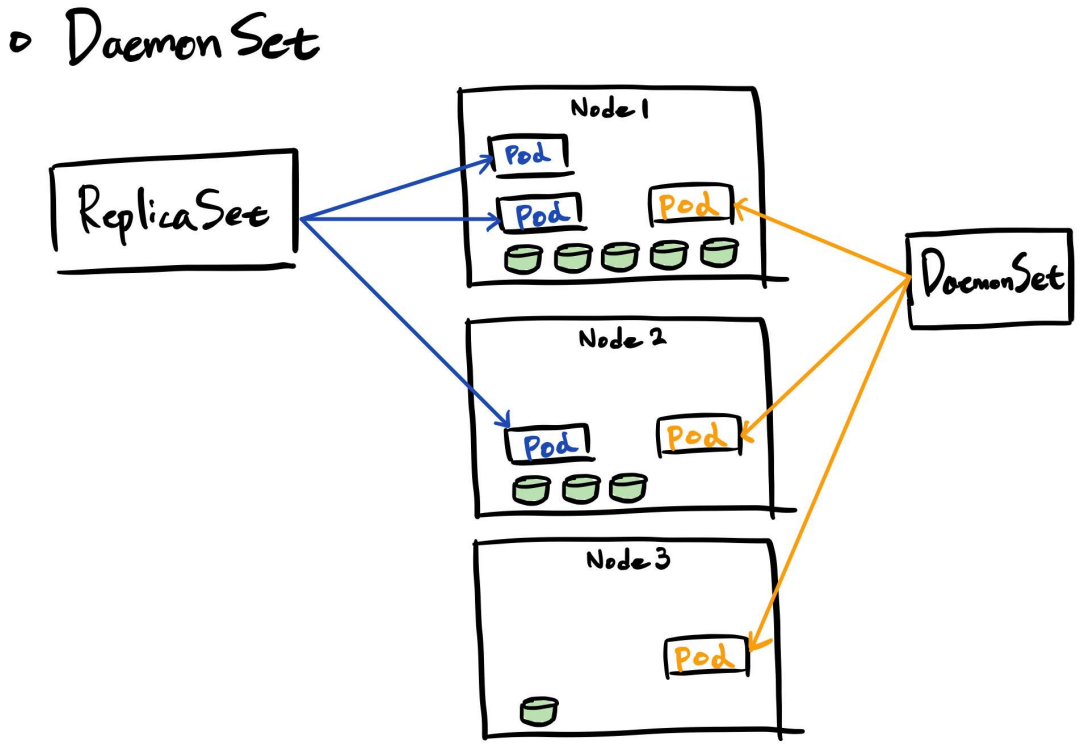
```
❯ kubectl create -f replicaset-matchlabels.yaml
replicaset.apps/rc-labels created

❯ kubectl get pods
NAME              READY   STATUS    RESTARTS   AGE
rc-labels-2htpz   1/1     Running   0          63s
rc-labels-vl67z   1/1     Running   0          63s
rc-labels-zq56g   1/1     Running   0          63s


❯ kubectl create -f replicaset-matchexpressions.yaml
replicaset.apps/rc-expressions created

❯ kubectl get pods
NAME                 READY   STATUS    RESTARTS   AGE
rc-expressions-p8fbx   1/1     Running   0          28s
rc-expressions-xjpk6   1/1     Running   0          28s
rc-expressions-xmcqx   1/1     Running   0          28s
rc-labels-2htpz        1/1     Running   0          106s
rc-labels-vl67z        1/1     Running   0          106s
rc-labels-zq56g        1/1     Running   0          106s
```

DaemonSet - 1/2

- DaemonSet을 활용하면 cluster의 모든 node에, node당 하나의 Pod를 배포할 수 있음

  . 시스템 수준의 작업을 수행하는 인프라 관련 Pod (로깅, 모니터링), Kube-proxy도 DaemonSet의 일종

- 특정 node에만 Pod를 배포하려면 nodeSelector 속성 지정

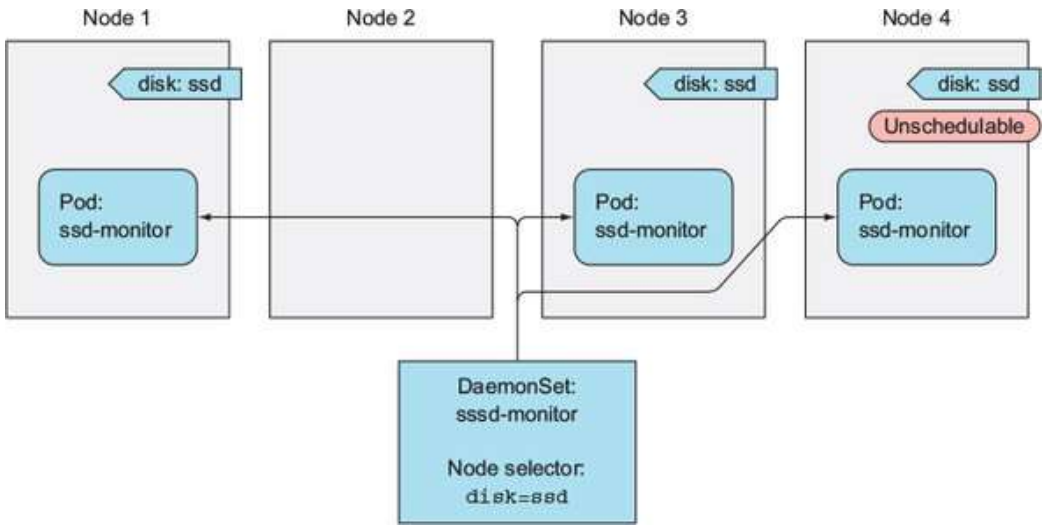


※ 참고 : https://zunoxi.github.io/devops/2020/11/07/devops-k8s-daemonset/

DaemonSet - 2/2

daemonset.yaml

```
apiVersion: apps/v1
kind: DaemonSet

metadata:
  name: ssh-monitor

spec:
  selector:
    matchLabels:
      app: ssd-monitor

  template:
    metadata:
      labels:
        app: ssd-monitor

    spec:
      nodeSelector:
        disk: ssd
      containers:
      - name: main
        image: luksa/ssd-monitor
```



※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-4/287

Job

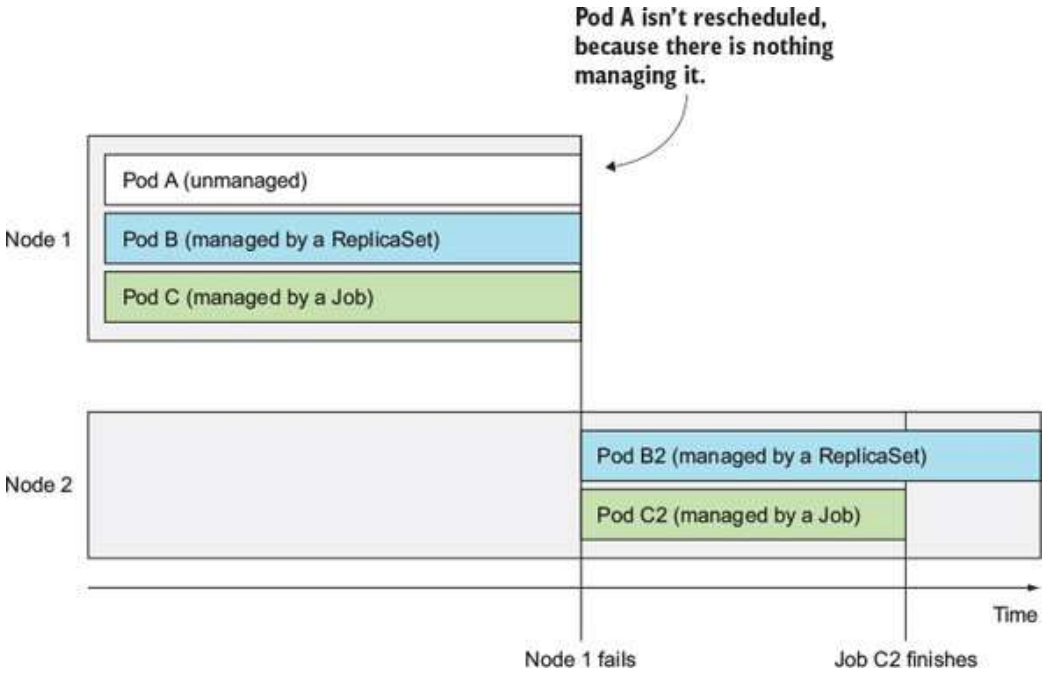
- 완료 가능한 단일 태스크를 수행하는 Pod를 실행

  . 프로세스 종료 이후에도 다시 실행되지 않음

  . node 장애 발생 時 다시 실행할지 말지 여부를 결정할 수 있음

job.yaml

```
apiVersion: batch/v1
kind: Job

metadata:
  name: batch-job

spec:
  template:
    metadata:
      labels:
        app: batch-job
    spec:
      restartPolicy: OnFailure
      containers:
      - name: main
        image: luksa/batch-job
```



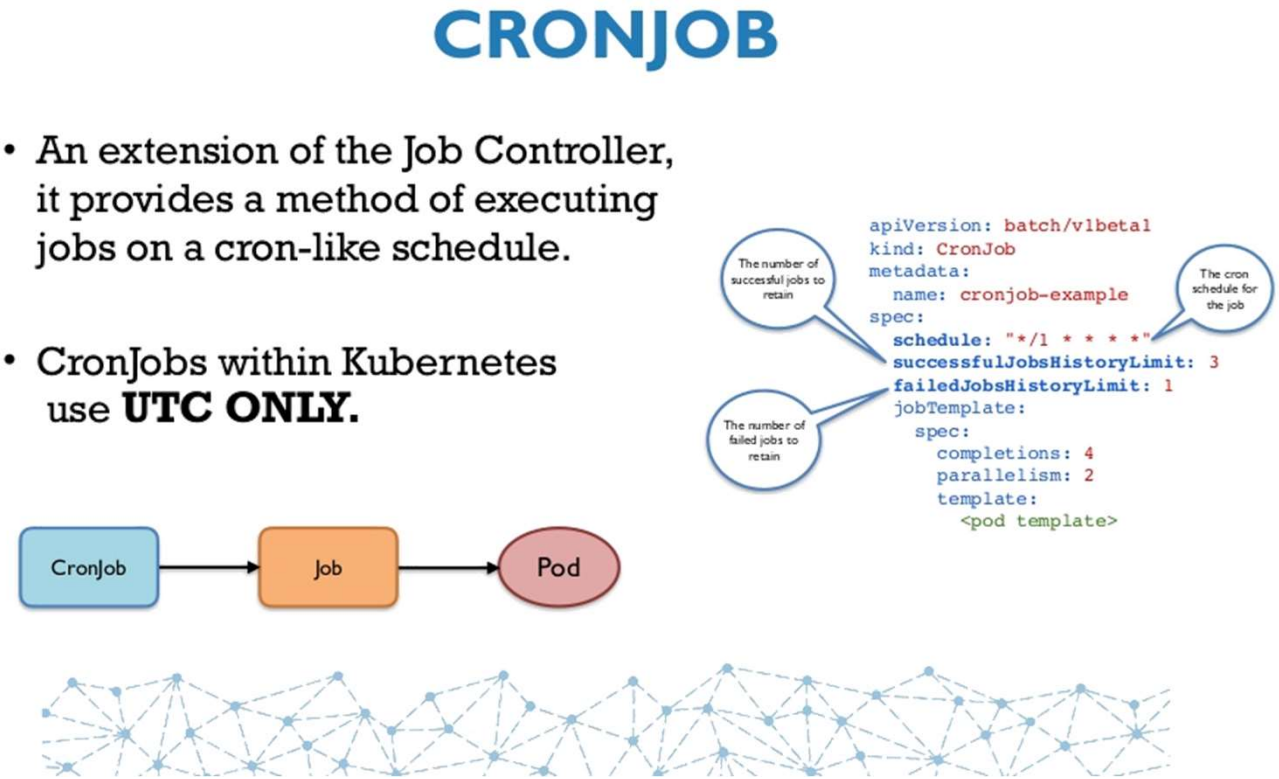
※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-4/321

CronJob

- 특정 시간 또는 지정된 간격으로 Job을 반복 실행

cronjob.yaml

```
apiVersion: batch/v1beta1
kind: CronJob

metadata:
  name: batch-job-every-fifteen-minutes

spec:
  schedule: "0,15,30,45 * * * *"
  jobTemplate:
    spec:
      template:
        metadata:
          labels:
            app: periodic-batch-job
        spec:
          restartPolicy: OnFailure
          containers:
          - name: main
            image: luksa/batch-job
```



※ 참고 : https://www.slideshare.net/RonnyTrommer/devjam-2019-introduction-to-kubernetes

https://kahoot.it/