6. Controller

Auto Healing

Controller

ReplicationController
ReplicaSet
~~StatusfulSet~~
DaemonSet

Pod1
Node1

Pod1
Node2

pod나 node가 죽으면 다른 node에 새로 만들어줌
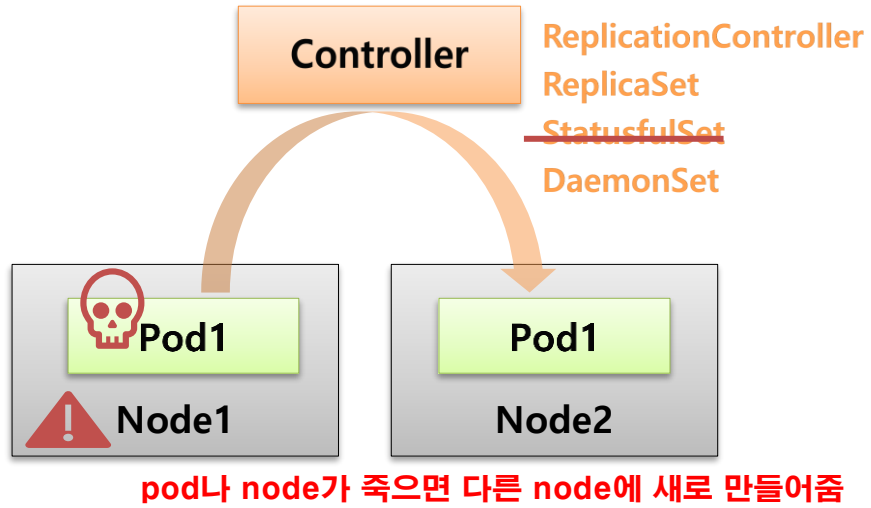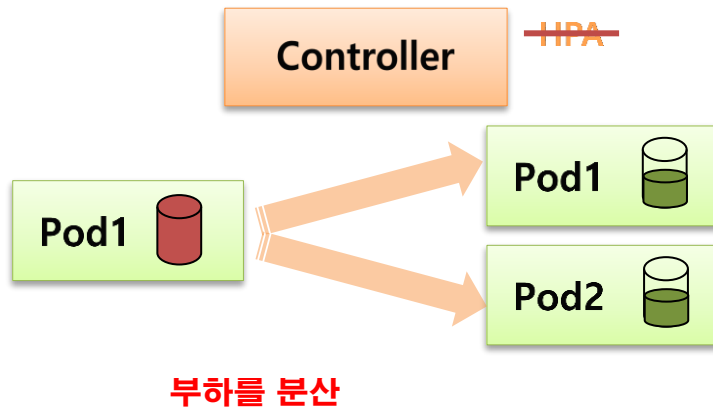
Software Update

버전업, 롤백 제공

Deployment

v1 Pod1   v1 Pod2
v1 Pod3   v1 Pod4
v1 Pod5   v1 Pod6

Controller

v2 Pod1   v2 Pod2
v2 Pod3   v2 Pod4
v2 Pod5   v2 Pod6

Auto Scaling

Controller

~~HPA~~

Pod1

Pod1

Pod2

부하를 분산

Job

Controller

CronJob
Job

Object
Pod   Pod   Pod

Resource
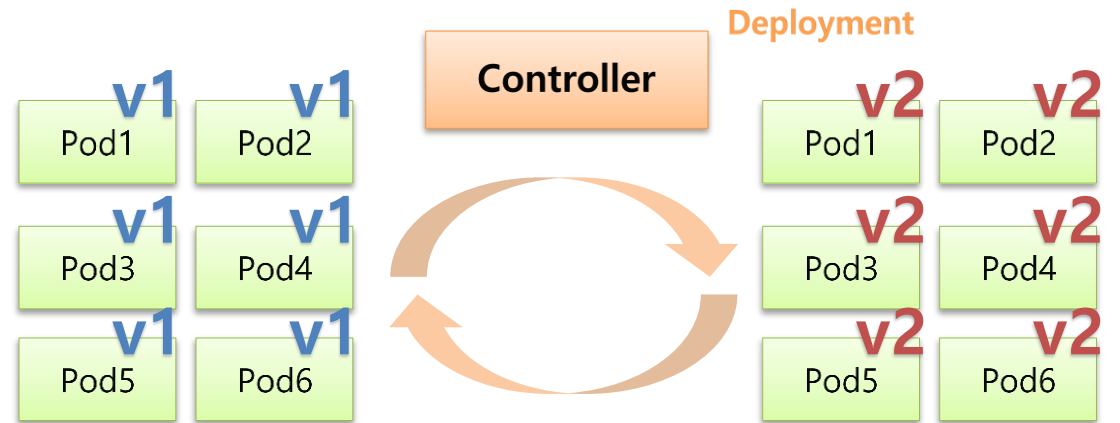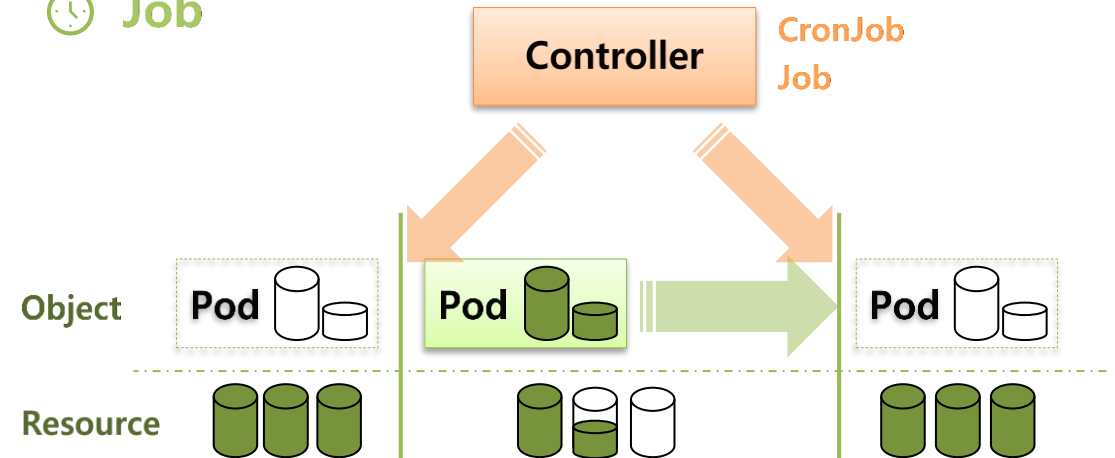
6. Controller - Replication Controller, ReplicaSet

replicaset에만 있는 기능

Template

컨트롤러와 pod는 label과 selector로 연결

Replication

selector   templete

type:web   Pod:v2

Pod:v2

Template Update

Re-create

pod죽으면 재생성

⚠ **Pod**

type:web

❌ **Pod**

type:web

Pod:v2

type:web

**템플릿 update 후 기존 pod
죽이면 새로 생성되면서 버전 업글**

Replicas

**따로 controller와 pod를 만들지 않고
한번에 만들면 replicas의 수 만큼
pod를 만들어주기 때문에 하나만
만들어도 됨**

replicas의 수 만큼 pod유지

Replication

replicas

3

Scale Out

Pod   Pod   Pod

Replication

replicas   templete

2   Pod

⚠

Pod   Pod
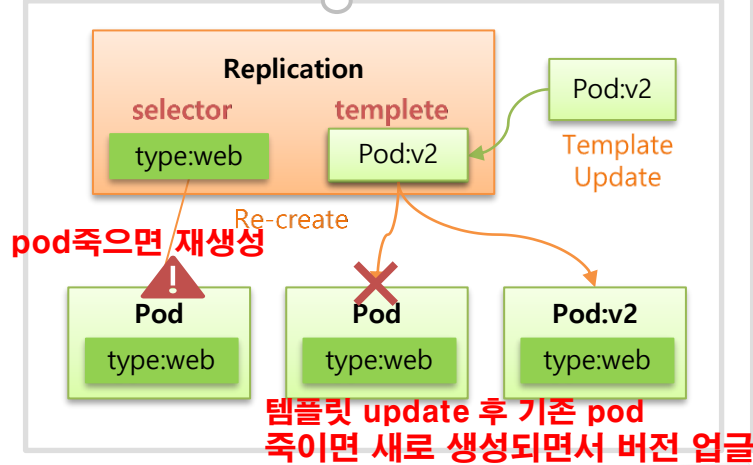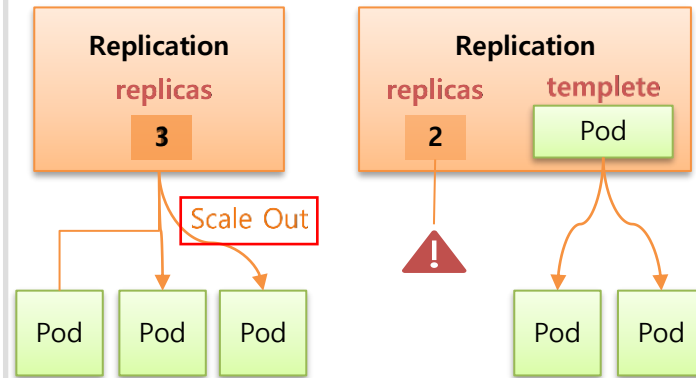
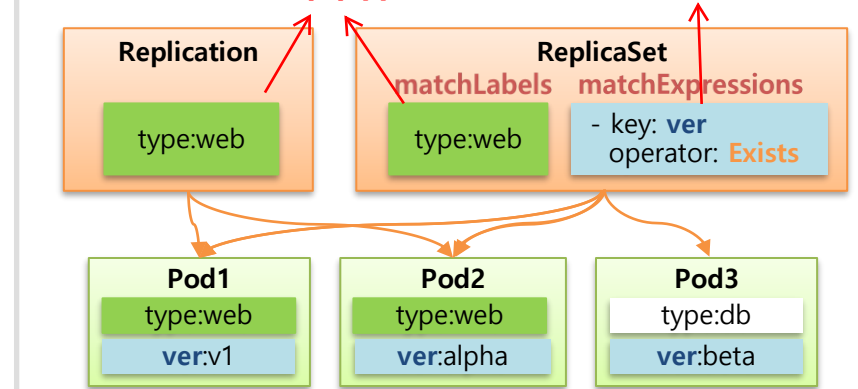## Selector

**key,value동일
해야 함**

상세 설정 가능

Replication

type:web

ReplicaSet

matchLabels   matchExpressions

type:web

- key: **ver**
operator: **Exists**

Pod1

type:web

ver:v1

Pod2

type:web

ver:alpha

Pod3

type:db

ver:beta
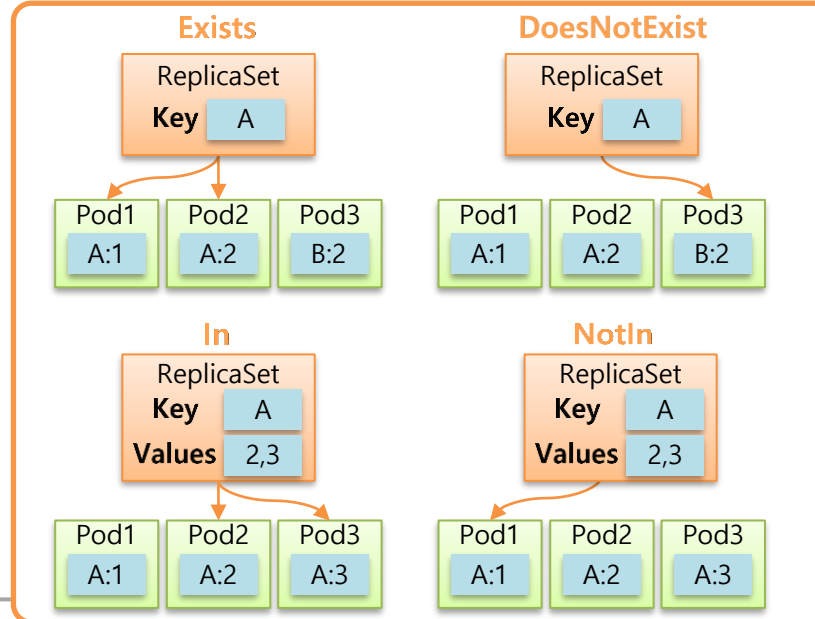
```
apiVersion: v1
kind: Pod
metadata:
 name: pod-1
 labels:
  type: web
spec:
 containers:
 - name: container
  image:
   tmkube/app:v1
```

```
apiVersion: v1
kind: ReplicationController
metadata:
 name: replication-1
spec:
 replicas: 1 -> 3
 selector:
  type: web
 template:
  metadata:
   name: pod-1
   labels:
    type: web
  spec:
   containers:
   - name: container
    image: tmkube/app:v2
```

일치해야함

Operator 종류들

Exists

ReplicaSet

Key   A

Pod1   Pod2   Pod3

A:1   A:2   B:2

DoesNotExist

ReplicaSet

Key   A

Pod1   Pod2   Pod3

A:1   A:2   B:2

In

ReplicaSet

Key   A

Values   2,3

Pod1   Pod2   Pod3

A:1   A:2   A:3

NotIn

ReplicaSet

Key   A

Values   2,3

Pod1   Pod2   Pod3
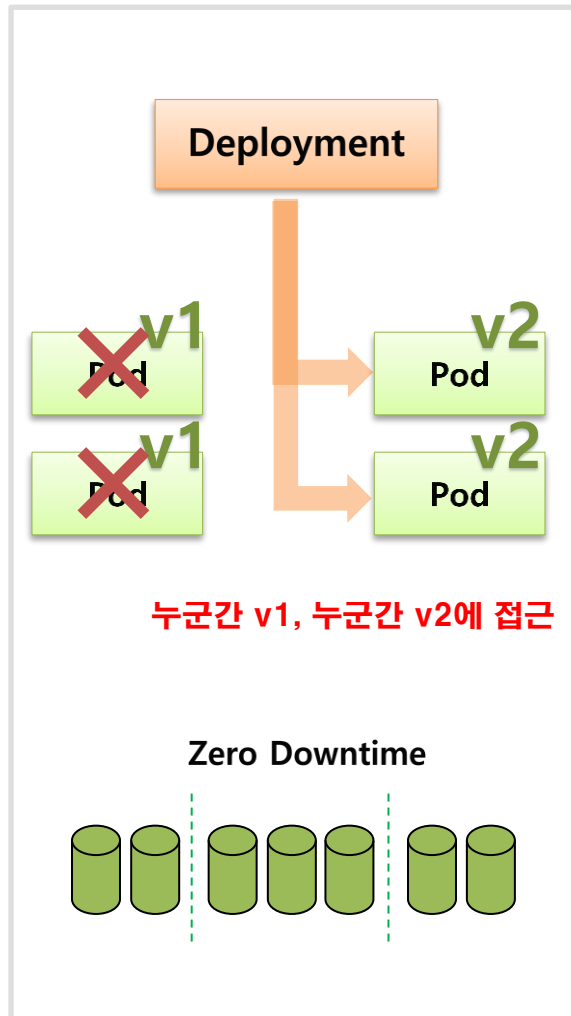
A:1   A:2   A:3

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
 name: replica-1
spec:
 replicas: 3
 selector:
  matchLabels:
   type: web
  matchExpressions:
  - {key: ver, operator: Exists}
 template:
  metadata:
   name: pod
...
```
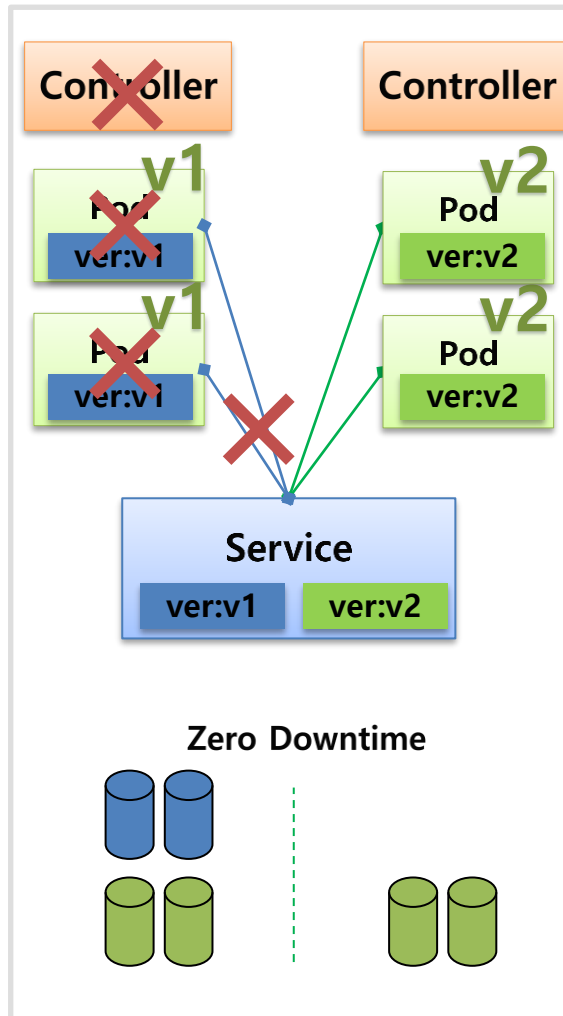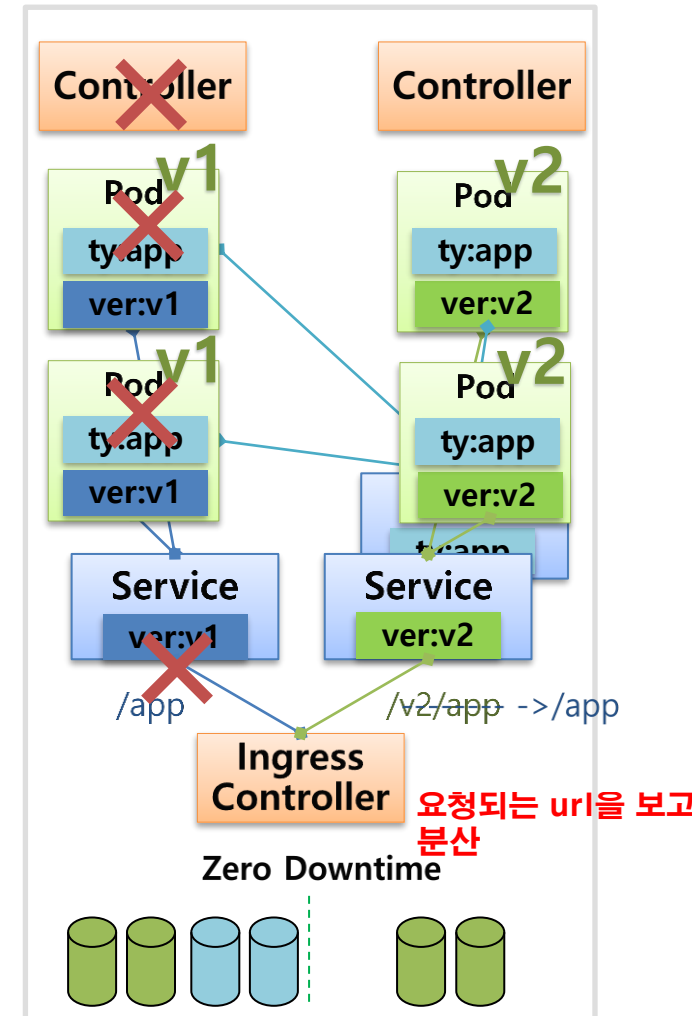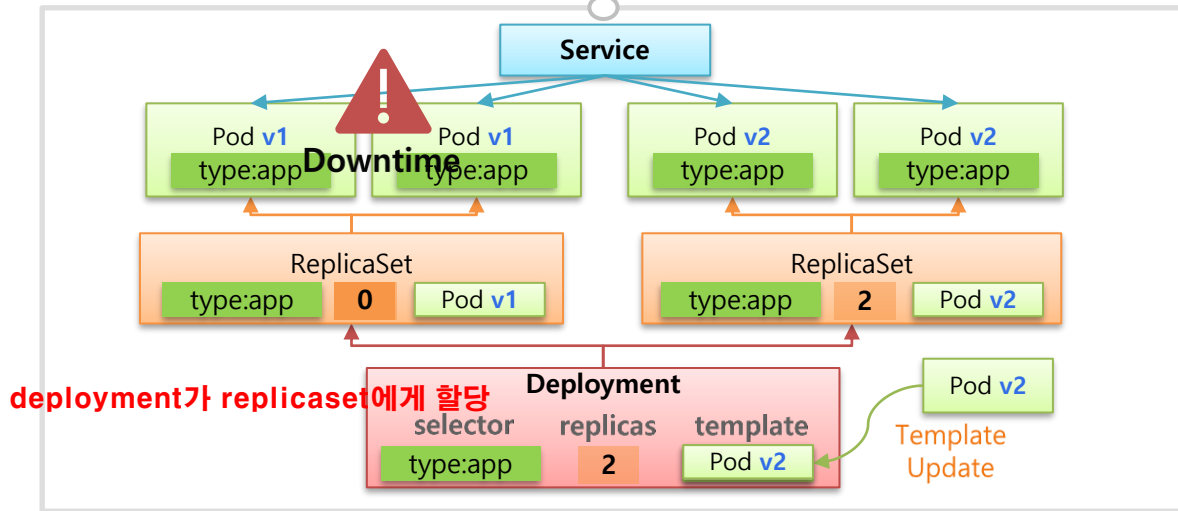
6. Controller - Deployment

ReCreate

Deployment

v1 Pod ~~(X)~~
v1 Pod ~~(X)~~

v2 Pod
v2 Pod

다운 타임 발생

Downtime

Rolling Update

Deployment

v1 Pod ~~(X)~~
v1 Pod ~~(X)~~

v2 Pod
v2 Pod

누군간 v1, 누군간 v2에 접근

Zero Downtime

Blue/Green

Controller ~~(X)~~
Controller

v1 Pod ~~(X)~~
ver:v1 ~~(X)~~

v1 Pod ~~(X)~~
ver:v1 ~~(X)~~

v2 Pod
ver:v2

v2 Pod
ver:v2

Service
ver:v1 | ver:v2

Zero Downtime

Canary

Controller ~~(X)~~
Controller

v1 Pod ~~(X)~~
ty:app
ver:v1

v1 Pod ~~(X)~~
ty:app
ver:v1

v2 Pod
ty:app
ver:v2

v2 Pod
ty:app
ver:v2

ty:app

Service
ver:v1 ~~(X)~~

Service
ver:v2

/app   /v2/app ->/app

Ingress Controller

요청되는 url을 보고 분산

Zero Downtime

6. Controller - Deployment

ReCreate

v2하나 먼저 생성하고 트래픽 분산

그리고 왼쪽 replicas를 1로 다운

Service

Pod **v1**
type:app

Downtime

Pod **v1**
type:app

Pod **v2**
type:app

Pod **v2**
type:app

ReplicaSet
type:app **0** Pod **v1**

ReplicaSet
type:app **2** Pod **v2**

deployment가 replicaset에게 할당

Deployment
selector replicas template
type:app **2** Pod **v2**

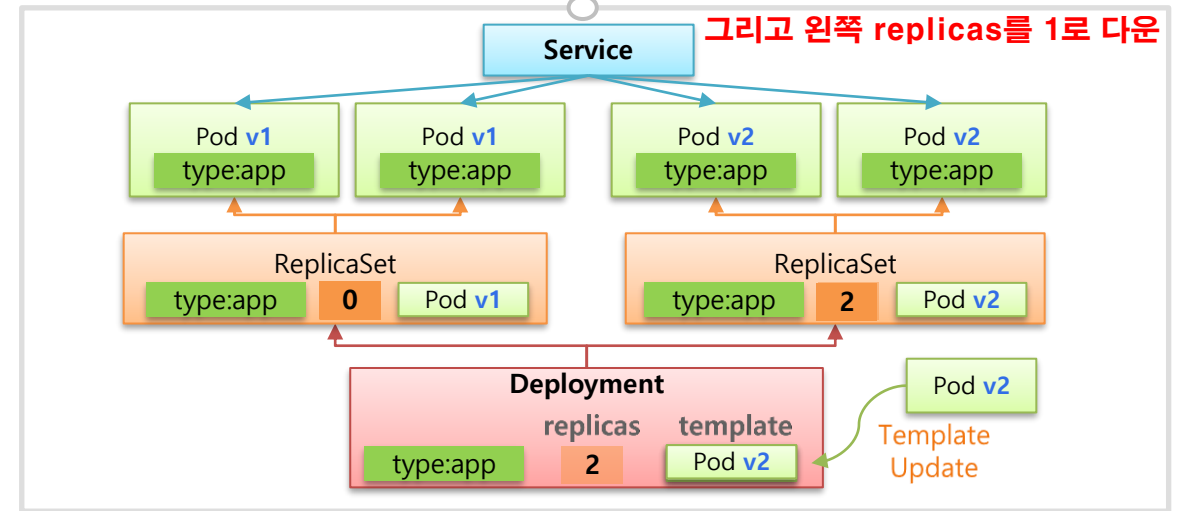Pod **v2**

Template Update

```
apiVersion: v1
kind: Service
metadata:
  name: svc-1
spec:
  selector:
    type: app
  ports:
  - port: 8080
    protocol: TCP
    targetPort: 8080
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-1
spec:
  selector:
    matchLabels:
      type: app
  replicas: 2
  strategy:
    type: Recreate
  revisionHistoryLimit: 1
  template:
    metadata:
      labels:
        type: app
    spec:
      containers:
      - name: container
        image: tmkube/app:v1
```

0인 replicaset을 하나만 남기게 함

이걸 통해 이전 버전으로 돌아갈 수 있음

Rolling Update (default)

Service

Pod **v1**
type:app

Pod **v1**
type:app

Pod **v2**
type:app

Pod **v2**
type:app

ReplicaSet
type:app **0** Pod **v1**

ReplicaSet
type:app **2** Pod **v2**

Deployment
replicas template
type:app **2** Pod **v2**

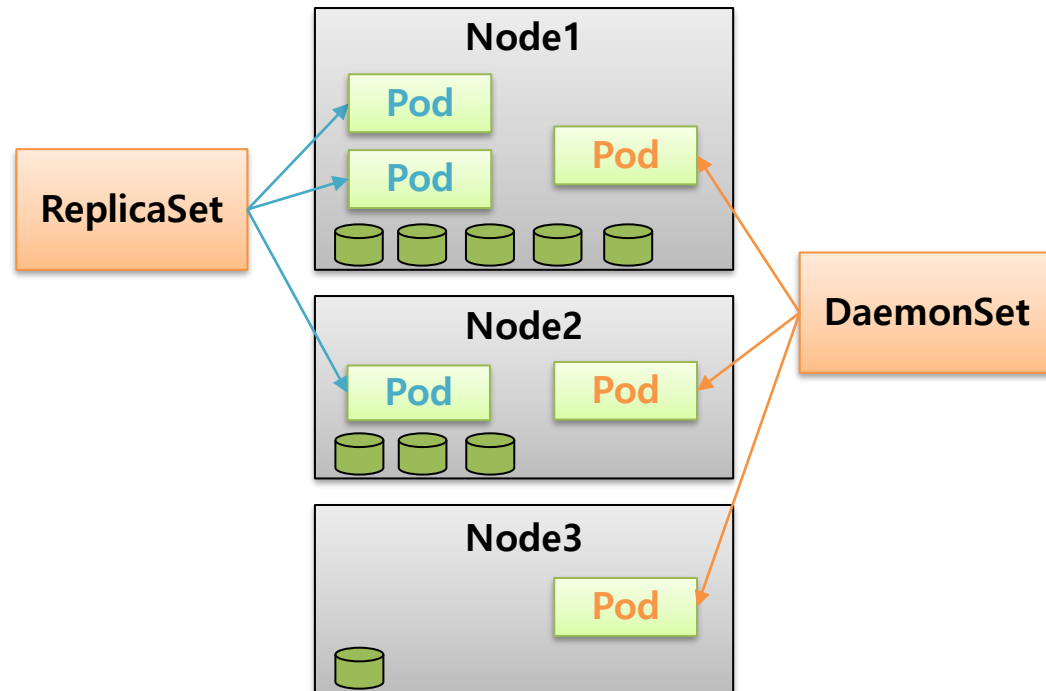Pod **v2**

Template Update

```
apiVersion: v1
kind: Service
metadata:
  name: svc-2
spec:
  selector:
    type: app
  ports:
  - port: 8080
    protocol: TCP
    targetPort: 8080
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-2
spec:
  selector:
    matchLabels:
      type: app
  replicas: 2
  strategy:
    type: RollingUpdate
  minReadySeconds: 10
  template:
    metadata:
      labels:
        type: app
    spec:
      containers:
      - name: container
        image: tmkube/app:v1
```

v2를 생성하고 v1을 지우는 시간
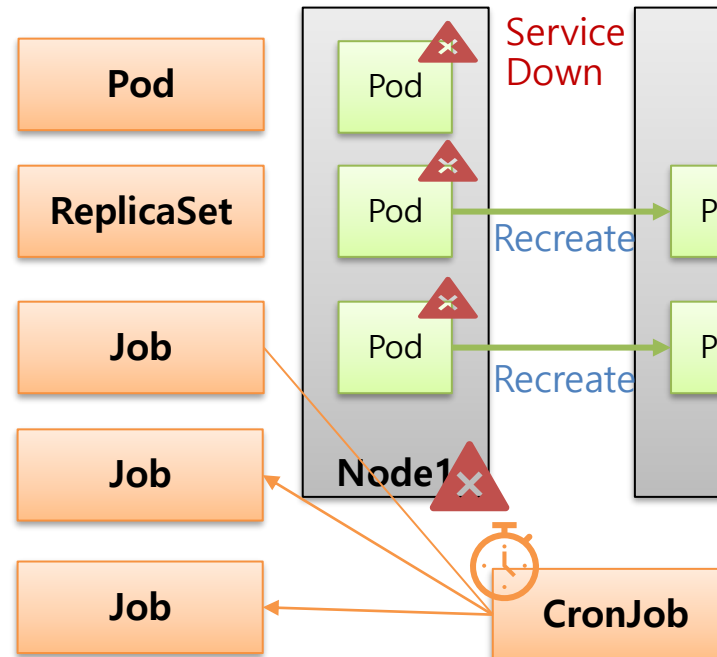
6. Controller - DaemonSet, Job, CronJob

6. Controller - DaemonSet, Job, CronJob

DaemonSet

DaemonSet

selector **template**
type:app   Pod

nodeSelector
os:centos
hostPort

Node1   **Node2**   **Node3**
Pod   Pod   Pod
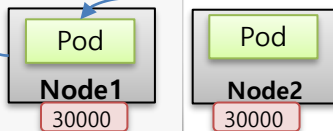os:centos   os:centos   os:ubuntu
18080   18080
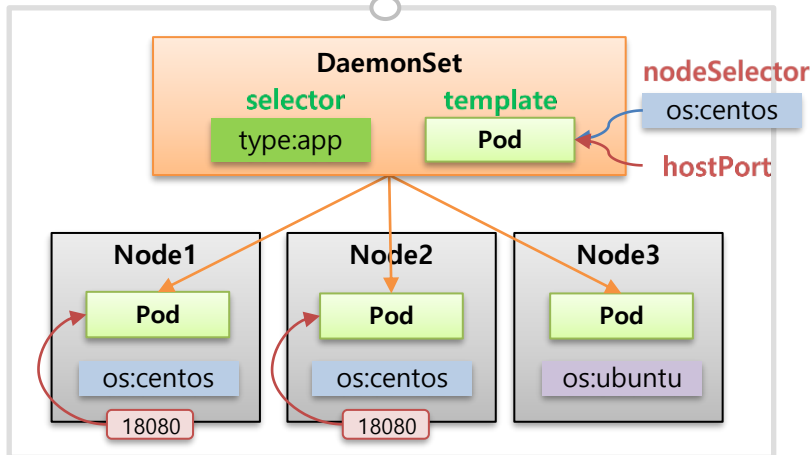
```
apiVersion: apps/v1
kind: DaemonSet
metadata:
 name: daemonset-1
spec:
 selector:
   matchLabels:
     type: app
 template:
   metadata:        spec:
     labels:         nodeSelector:
       type: app       os: centos
                     containers:
                     - name: container
                       image: tmkube/app
                       ports:
                       - containerPort: 8080
                       hostPort: 18080
```
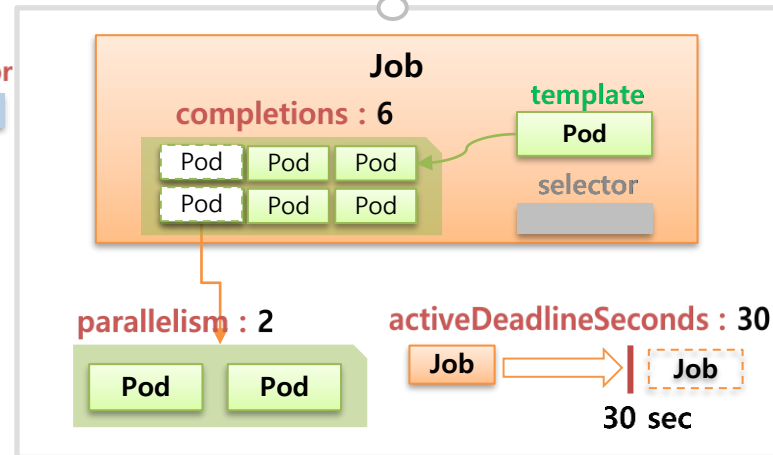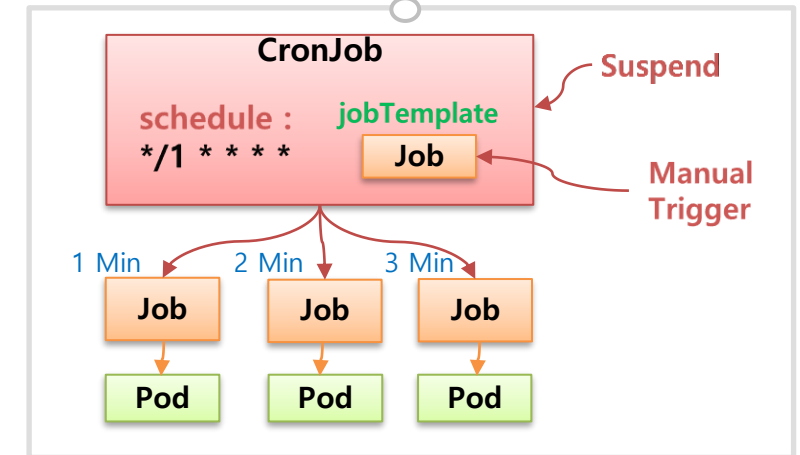
externalTrafficPolicy: Local

Service

Pod   Pod
Node1   **Node2**
30000   30000
External

Job

Job

completions : 6   **template**
Pod   Pod   Pod   Pod
Pod   Pod   Pod
selector

parallelism : 2   **activeDeadlineSeconds : 30**
Pod   Pod   Job → Job
                              30 sec

```
apiVersion: batch/v1
kind: Job
metadata:
 name: job-1
spec:
 completions: 6
 parallelism: 2
 activeDeadlineSeconds: 30
 template:
  spec:
   restartPolicy: Never
   containers:
   - name: container
     image: tmkube/init
```
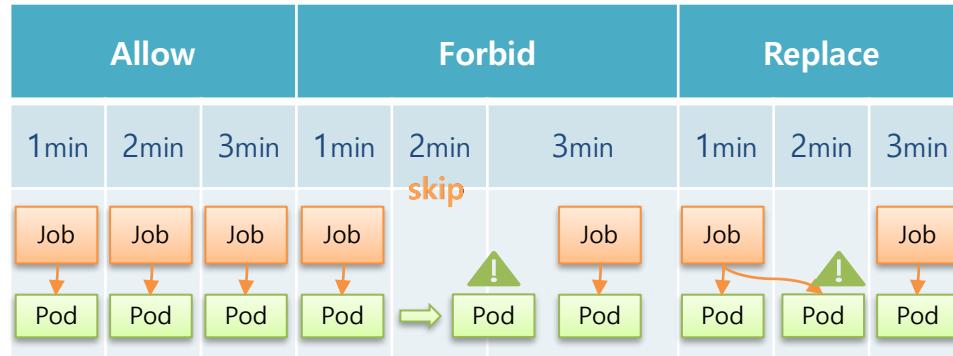
CronJob

CronJob

schedule :   **jobTemplate**
*/1 * * * *   Job

Suspend

Manual Trigger

1 Min   2 Min   3 Min
Job   Job   Job
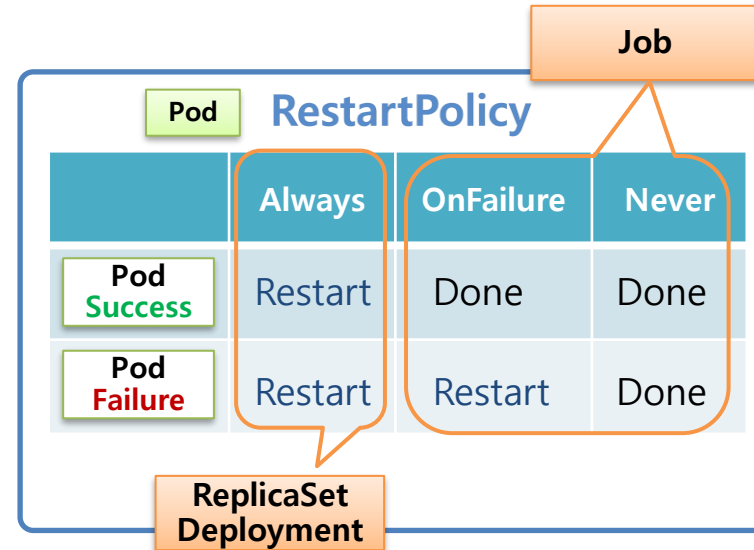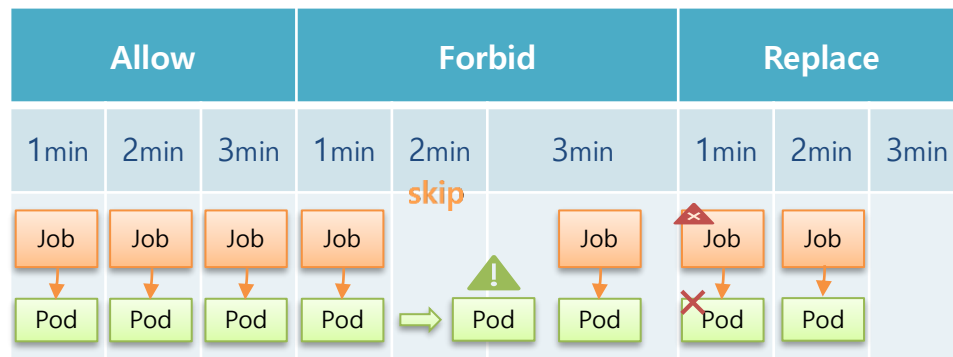Pod   Pod   Pod

```
apiVersion: batch/v1
kind: CronJob
metadata:
 name: cron-job
spec:
 schedule: "*/1 * * * *"
 concurrencyPolicy: Allow
 jobTemplate:
   spec:
     template:
       spec:
         restartPolicy: Never
         containers:
         - name: container
           image: tmkube/app
```

concurencyPolicy

Allow			Forbid			Replace		
1min	2min	3min	1min	2min skip	3min	1min	2min	3min
Job	Job	Job	Job		Job	Job		Job
Pod	Pod	Pod	Pod	Pod	Pod	Pod	Pod	Pod

concurencyPolicy

Allow			Forbid			Replace		
1min	2min	3min	1min	2min skip	3min	1min	2min	3min
Job	Job	Job	Job		Job	Job	Job	
Pod	Pod	Pod	Pod	Pod	Pod	Pod	Pod	

RestartPolicy

Pod

Job

	Always	OnFailure	Never
Pod Success	Restart	Done	Done
Pod Failure	Restart	Restart	Done

ReplicaSet Deployment

```
apiVersion: batch/v1
kind: Job
metadata:
 name: job-1
spec:
  completions: 6
  parallelism: 2
  activeDeadlineSeconds: 30
  template:
   spec:
    restartPolicy: Never
    containers:
    - name: container
      image: tmkube/init
```