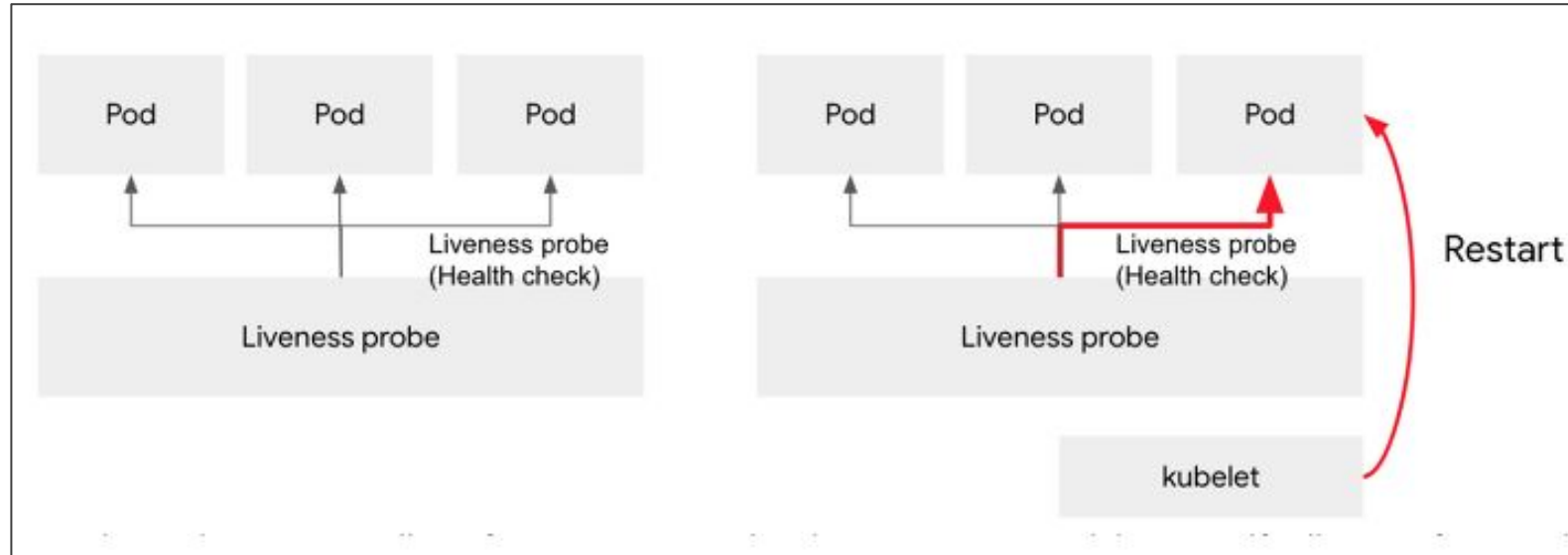


10. CTL | Liveness-probes

1. Liveness Probe - Concept

K8s는 liveness Probe 를 통해 컨테이너의 Health Check 를 수행 합니다. Health Check가 실패할 경우 POD를 다시 시작 합니다.



Probe 타입	설명
HTTP GET	지정된 IP/PORT에 HTTP GET 을 수행 합니다.
TCP Socket	지정된 IP/PORT에 TCP 연결을 수행 합니다.
EXEC	컨테이너 내부에서 명령어를 실행하고 종료 코드를 확인 합니다.

1. Liveness Probe 설정 값

- 공통 설정

Probe 설정값	설명	Default
initialDelaySeconds	pod 가 시작되고 최초의 liveness probe가 수행되기 까지 대기시간	0
periodSeconds	probe가 수행 되는 간격	10
timeoutSeconds	지정된 시간 안에 응답을 반환 해야 성공으로 간주 됨.	1
successThreshold	probe가 실패한 이후에 지정된 횟수 만큼 성공 할 경우 성공으로 간주	1
failureThreshold	probe가 지정한 개수 만큼 연속으로 실패하는 재가동 함	3

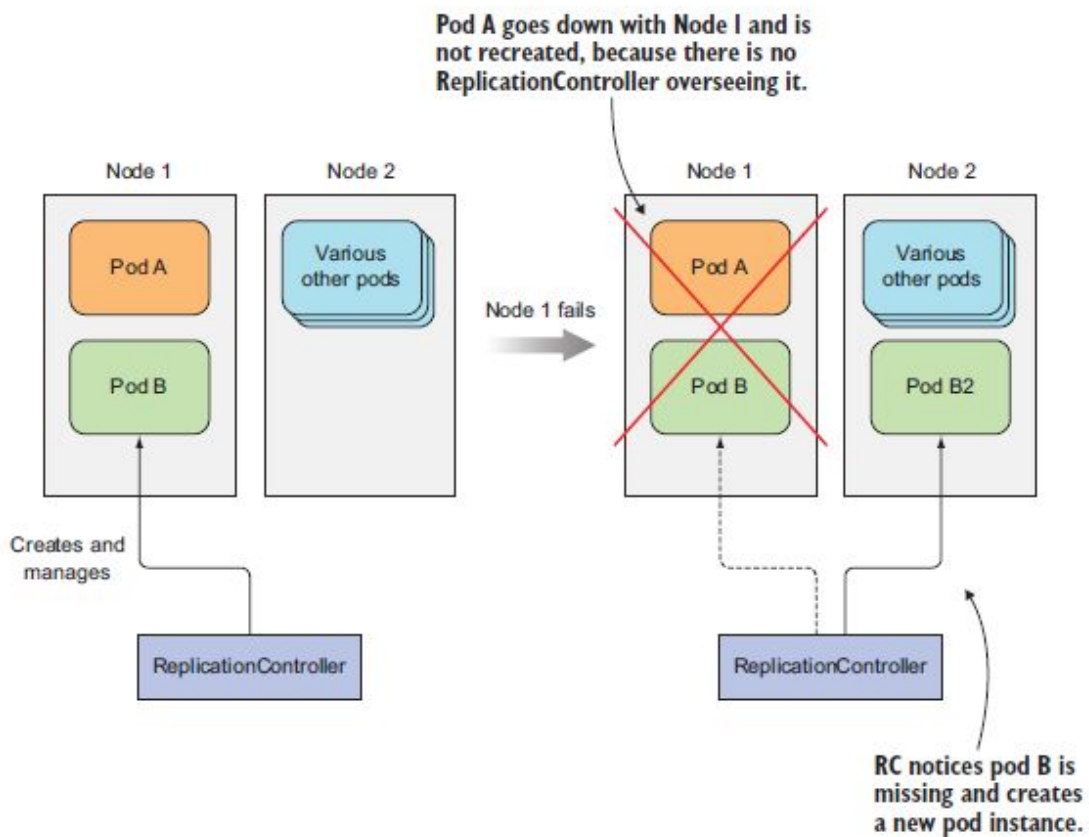
- HTTP 추가 설정

Probe 설정값	설명	Default
host	접속할 호스트명. httpHeaders 에 설정 해도 무방	pod ip
scheme	접속 방법 선택. HTTP 또는 HTTPS	HTTP
path	HTTP 서버 접속 경로	
httpHeaders	request 의 커스텀 헤더. 설정	
port	접속할 컨테이너의 포트 번호 또는 이름 (1~ 65536 사이값)	

11. CTL | Replication Controller

2. Replication Controller - Concept

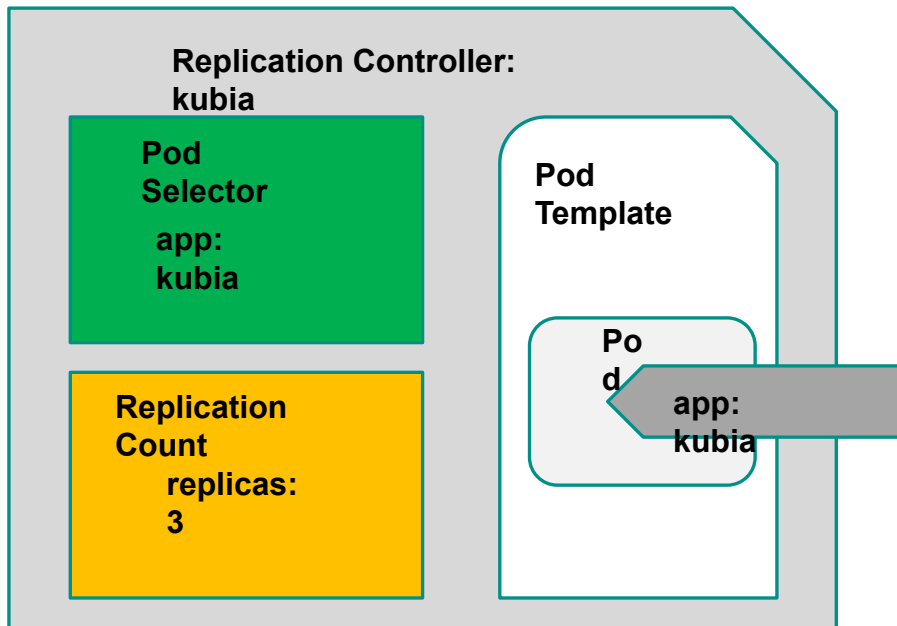
Replication Controller 는 어떠한 이유로는 POD가 사라지게 되면 대체 할 POD 생성합니다.



- RC는 사라진 PodB를 대체할 새로운 Pod B2를 다른 노드에 생성
- RC가 없는 Pod A 는 완전히 사라짐
- RC는 지속적으로 모니터링 하고 지정된 개수가 맞는지 지속적으로 체크함

2. Replication Controller – 3가지 구성 요소

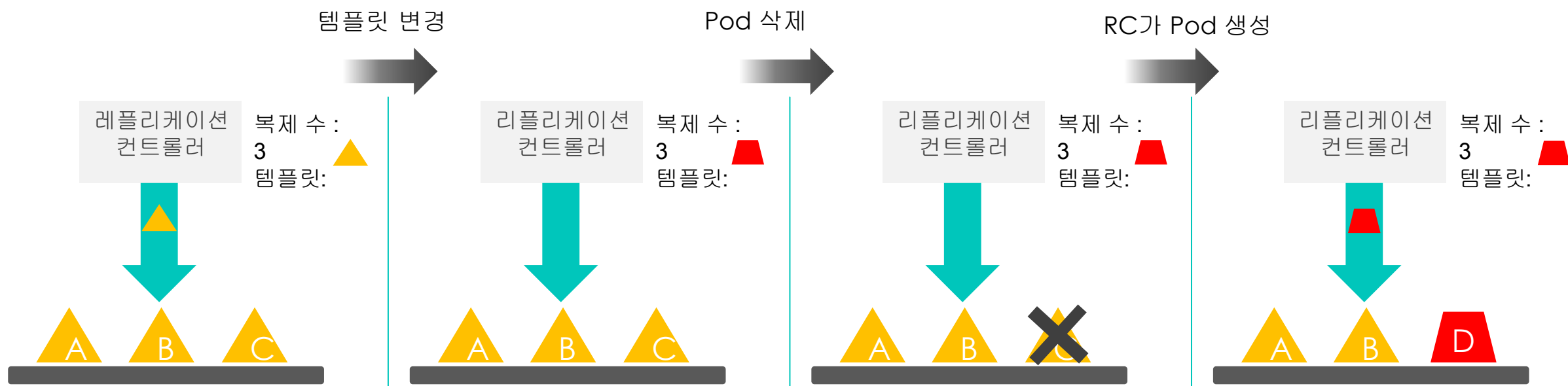
- RC 범위에 있는 포드를 결정하는 라벨 셀렉터를 사용함
- 실행해야 하는 POD의 원하는 수를 지정하는 복제본수
- 새로운 POD 복제본을 만들때 사용해야 하는 POD Template



라벨 셀렉터가 변경되면 POD가 RC의 범위를 벗어남.

2. Replication Controller – Pod Template 변경

- Pod Template 은 언제든지 수정 가능
- Pod 수정 시점이 아니라 새로운 Pod 가 생성 되는 시점에 적용 됨



```
kubectl edit rc <replication controller name>
```

2. Replication Controller – 장점

- Pod가 없을경우, 새 Pod를 시작해 항상 실행 되도록 합니다.
- 특정 노드에 장애가 발생 하더라도 해당 노드의 모든 Pod는 다른 노드에서 동일한 개수로 재 수행됨
- 손쉽게 스케일링이 가능함

2. Replication Controller – 생성방법

```
apiVersion: v1
kind: ReplicationController
metadata:
  name: nginx
spec:
```

```
  replicas: 3
```

```
  selector:
    app: nginx
```

```
  template:
    metadata:
      name: nginx
    labels:
      app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx
          ports:
            - containerPort: 80
```



복제본 개수



라벨 셀렉터



Pod Template

Pod 생성시에 Template 기반으로 Pod 생성

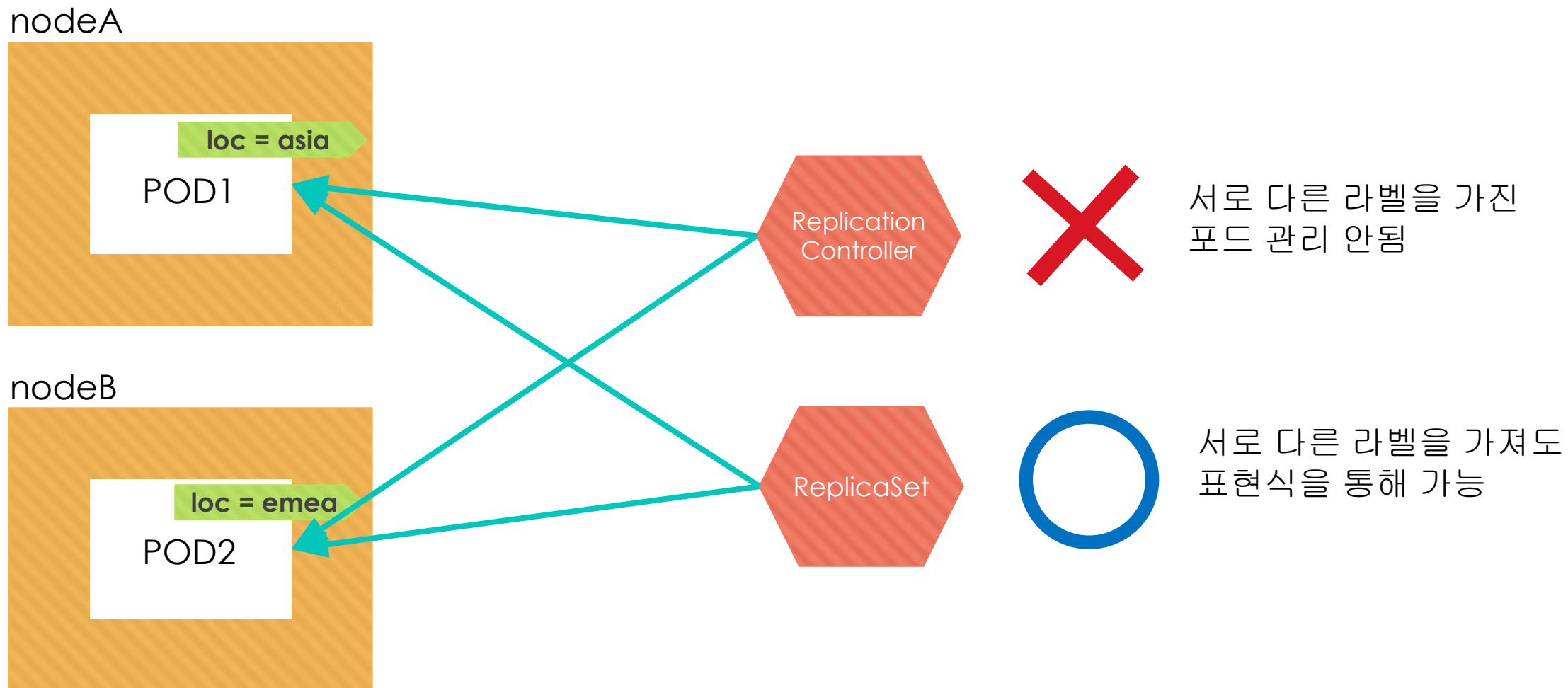
12. CTL | ReplicaSet

3. ReplicaSet - Concept

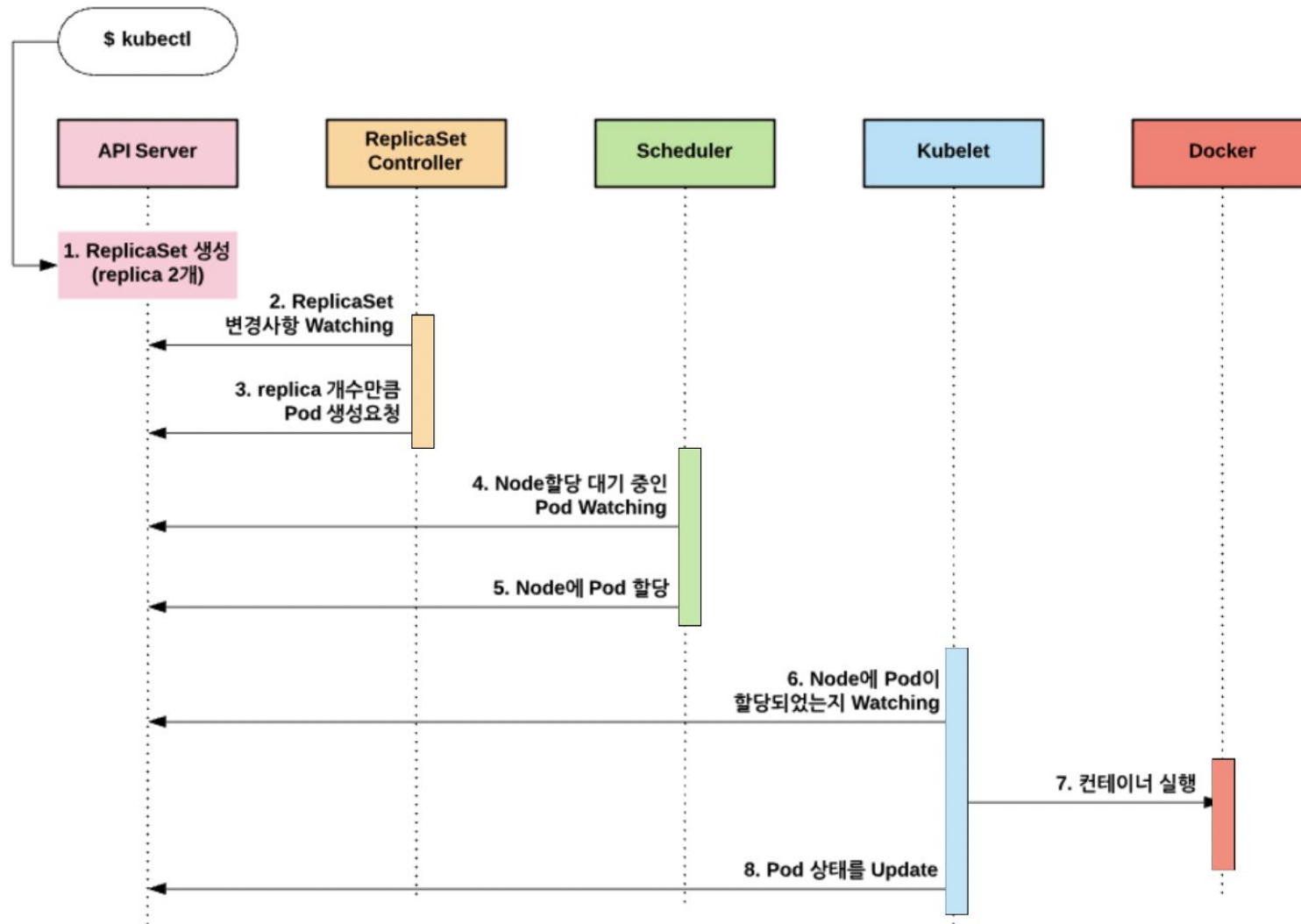
- Replication Controller 를 대체 하기 위해 나옴
- 일반적으로 ReplicaSet을 직접 생성 하기 보다 Deployment 를 이용해서 생성함
- Replication Controller 와 동일하게 동작 하지만 풍부한 표현 식을 통해 POD를 선택 할 수 있습니다.
- 표현식에는 matchLabels 와 matchExpressions 가 있다

연산자	설명
In	라벨값이 지정된 값 중 하나와 일치해야 함.
NotIn	라벨 값이. 지정된 값과 일치해서는 안됨
Exists	Pod에 지정된 키가 있는 라벨이 포함 되어야 함(값은 중요하지 않음)
DoesNotExist	Pod에 지정된 키다 있는 라벨이 포함 되면 안됨(Value 속성 지정 하면 안됨)

3. ReplicaSet – Replication Controller 와 의 차이점



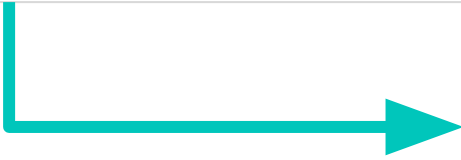
3. ReplicaSet – LifeCycle



3. ReplicaSet – 표현식의 예

- 예시 처럼 RC 보다 풍부한 표현 문구와 식을 사용 할 수 있습니다.
- 실제 운영 환경에서는 보단 많은 시스템 속성들이 실제로 사용 됩니다.

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nodejs-rs
selector:
# ReplicationController 는 matchLabels 만 사용가능
matchLabels:
  app: redis
matchExpressions:
  - {key: tier, operator: In, values: [cache]}
  - {key: environment, operator: NotIn, values: [dev]}
  - {key: service, operator: Exists, values: [user]}
  - {key: service, operator: DoesNotExist, values: [db]}
```



tier=cache
environment=prod
service=user
service=middleware

3. ReplicaSet 조회 및 삭제

- 예시 처럼 RC 보다 풍부한 표현 문구와 식을 사용 할 수 있습니다.
- 실제 운영 환경에서는 보단 많은 시스템 속성들이 실제로 사용 됩니다.

조회

```
] kubectl get rs
```

상세한 조회

```
] kubectl describe rs <ReplicaSet-Name>
```

삭제 (POD 까지 삭제)

```
] kubectl delete rs <ReplicaSet-Name>
```

삭제 (POD 는 남기고 삭제)

```
] kubectl delete rs <ReplicaSet-Name> --cascade=false
```