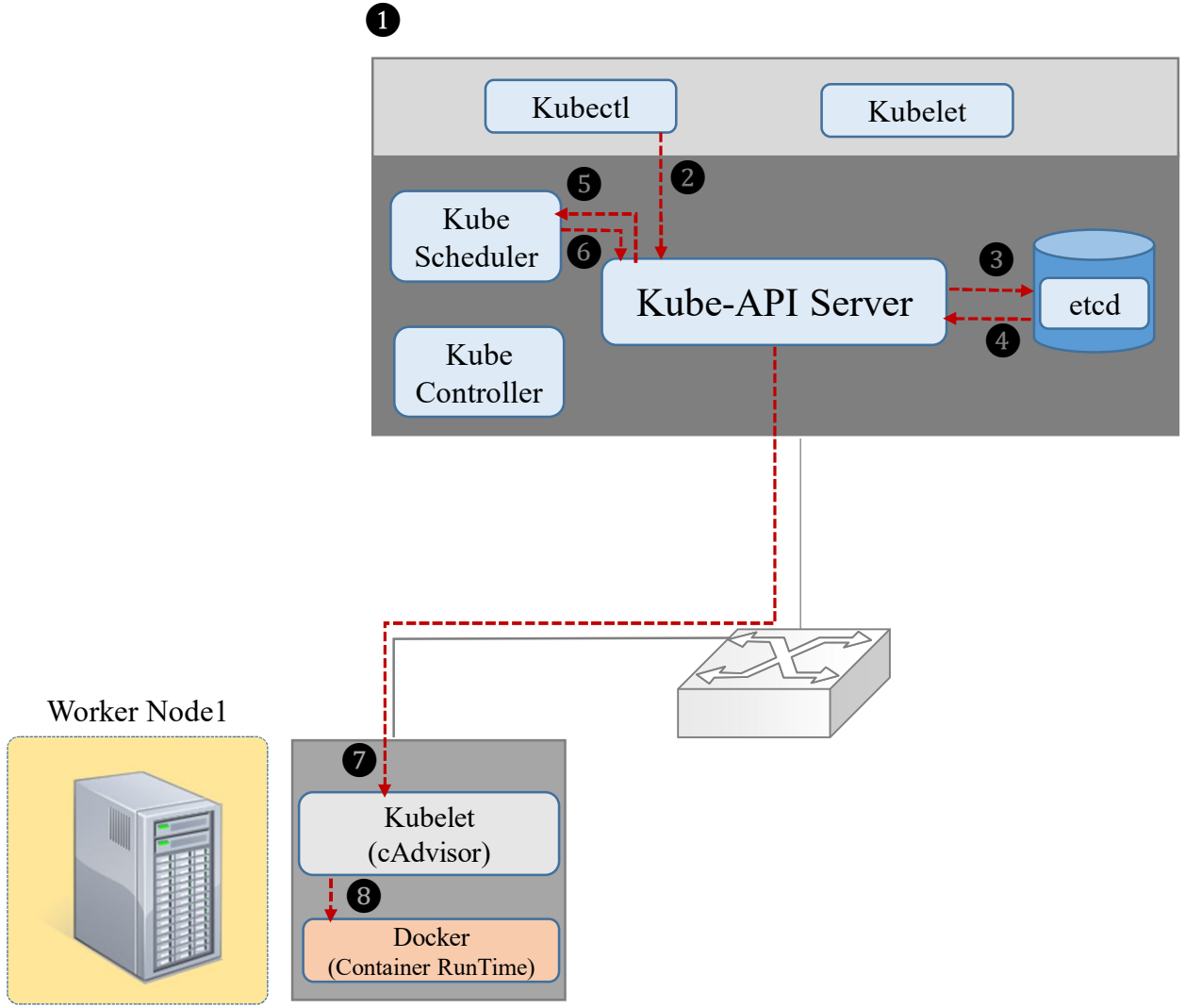
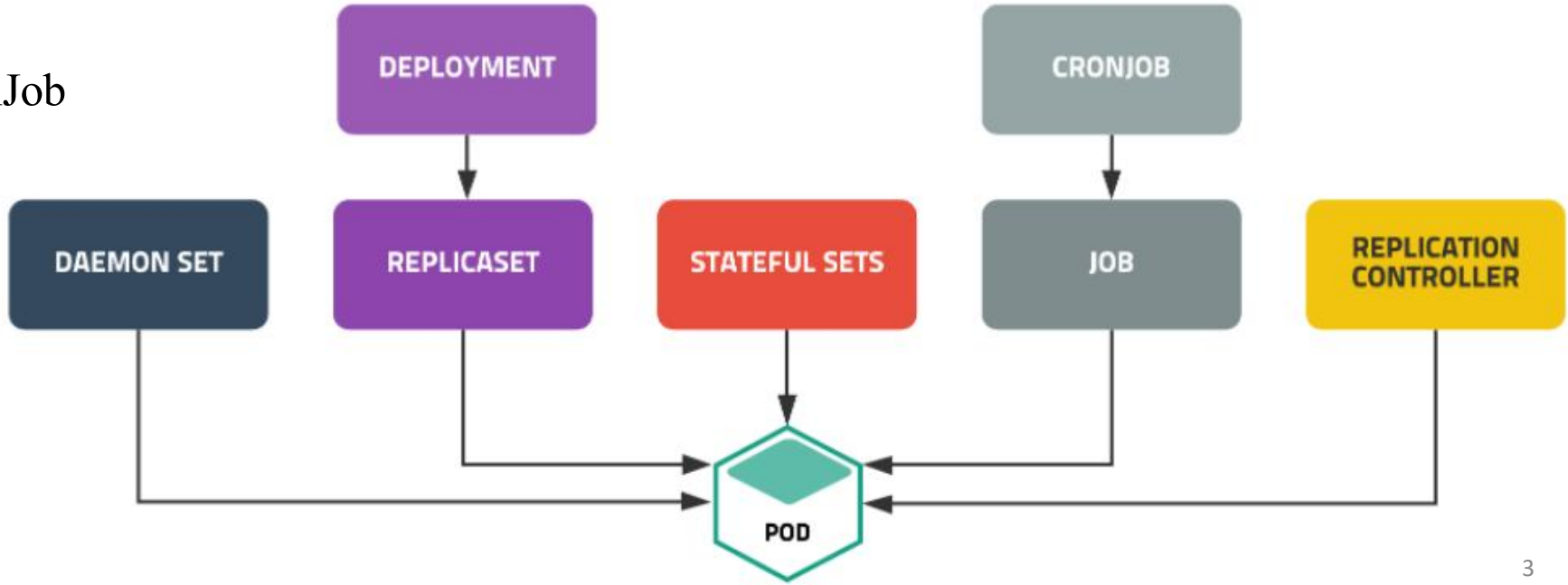
K8S Controller

Controller

- pod 개수를 보장
- 파드들을 관리하는 역할
  - 장시간 실행되는 파드 관리
  - 파드들의 주기적인 배치



- Replication Controller
- ReplicaSet
- Deployment
- DaemonSet
- Statefuleset
- Job
- CronJob



- Replication Controller
- ReplicaSet
- Deployment
- DaemonSet
- Statefuleset
- Job
- CronJob

```
apiVersion: v1
kind: ReplicationController
metadata:
  name: rc-main
spec:
  replicas: 2
  selector:
    app: main
    name: apache
    rel: stable
  template:
    metadata:
      labels:
        app: main
        name: apache
        rel: stable
    spec:
      containers:
      - name: webui
        image: httpd:2.2
        ports:
        - containerPort: 80
```

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nginx-replicaset
spec:
  template:
    metadata:
      name: nginx-replicaset
      labels:
        app: nginx-replicaset
    spec:
      containers:
      - name: nginx-replicaset
        image: nginx
        ports:
        - containerPort: 80
  replicas: 3
  selector:
    matchLabels:
      app: nginx-replicaset
```

1. Replication controller

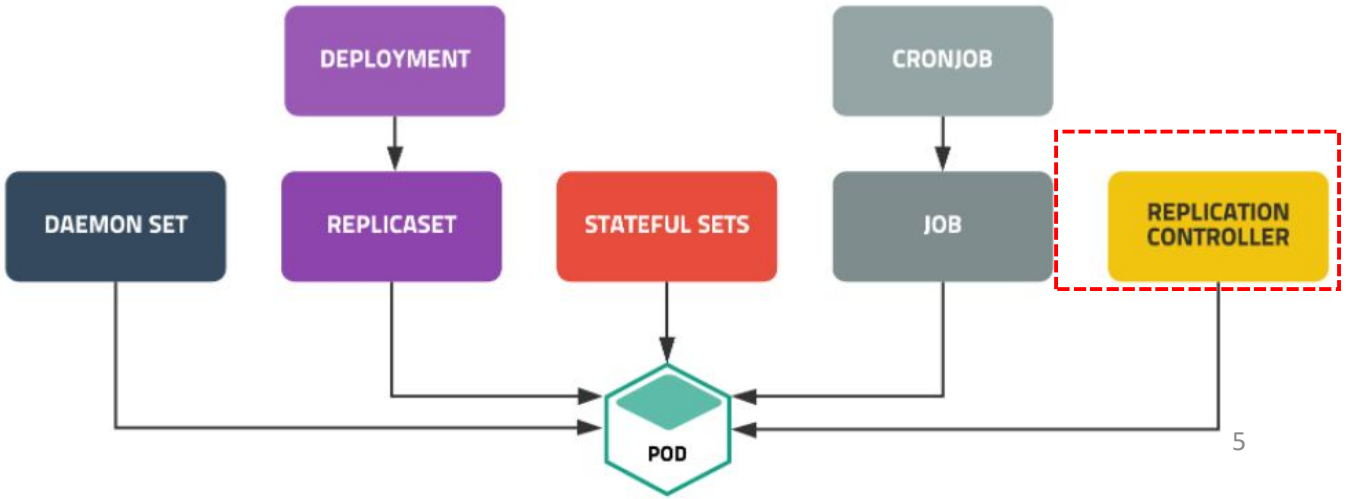
- 지정한 숫자만큼의 pod가 항상 클러스터 안에서 실행되도록 관리

  - 요구하는 Pod 개수 보장

  - 파드 집합의 실행을 항상 안정적으로 유지하는 것을 목표

  - 요구하는 pod 개수가 부족하면 template을 이용해 pod 추가

  - 요구하는 pod 수보다 많으면 최근에 생성된 pod 삭제



- 기본 구성
  - selector
  - replicas
  - template

```
apiVersion: apps/v1
kind: ReplicationController
metadata:
  name: <RC_name>
spec:
  replicas: <배포개수>
  selector:
     key: value
  template:
    <컨테이너 템플릿>
```

```
apiVersion: apps/v1
kind: ReplicationController
metadata:
  name: rc_test
spec:
  replicas: 3
  selector:
     app: test
  template:
    <컨테이너 템플릿>
```

실습 1. ReplicationController를 이용해서 Pod를 항상 3개를 유지하도록 한다.

```
apiVersion: v1
kind: ReplicationController
metadata:
  name: rc-nginx
spec:
  replicas: 3
  selector:
    app: webui
  template:
    metadata:
      name: nginx-pod
      labels:
        app: webui
    spec:
      containers:
      - name: nginx-container
        image: nginx:1.14
```

<<rc-nginx.yaml>>

watch kubectl get pods -o wide

kubectl create -f rc-nginx.yaml

kubectl get replicationcontrollers

kubectl get rc

kubectl describe rc rc-nginx

kubectl get pods

kubectl get pods --show-labels

kubectl delete pod rc-nginx-xxxx

kubectl delete pod --all

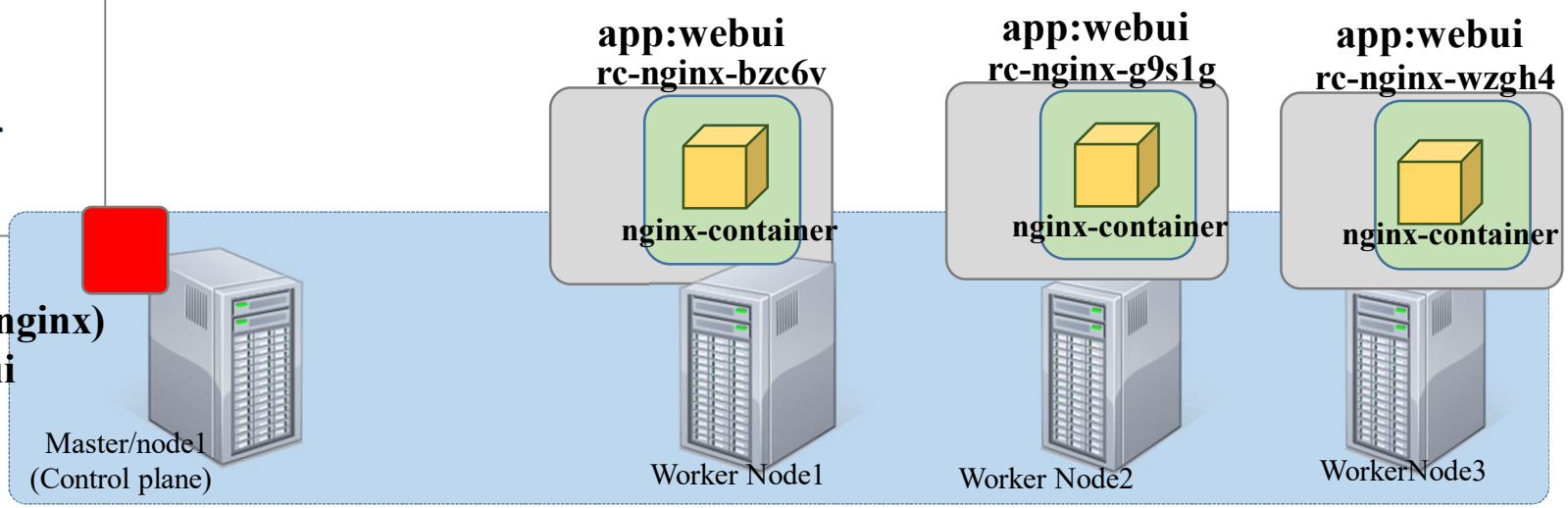
kubectl delete rc rc-nginx

```
apiVersion: v1
kind: ReplicationController
metadata:
  name: rc-nginx
spec:
  replicas: 3
  selector:
    app: webui
  template:
    metadata:
      name: nginx-pod
      labels:
        app: webui
    spec:
      containers:
      - name: nginx-container
        image: nginx:1.14
```

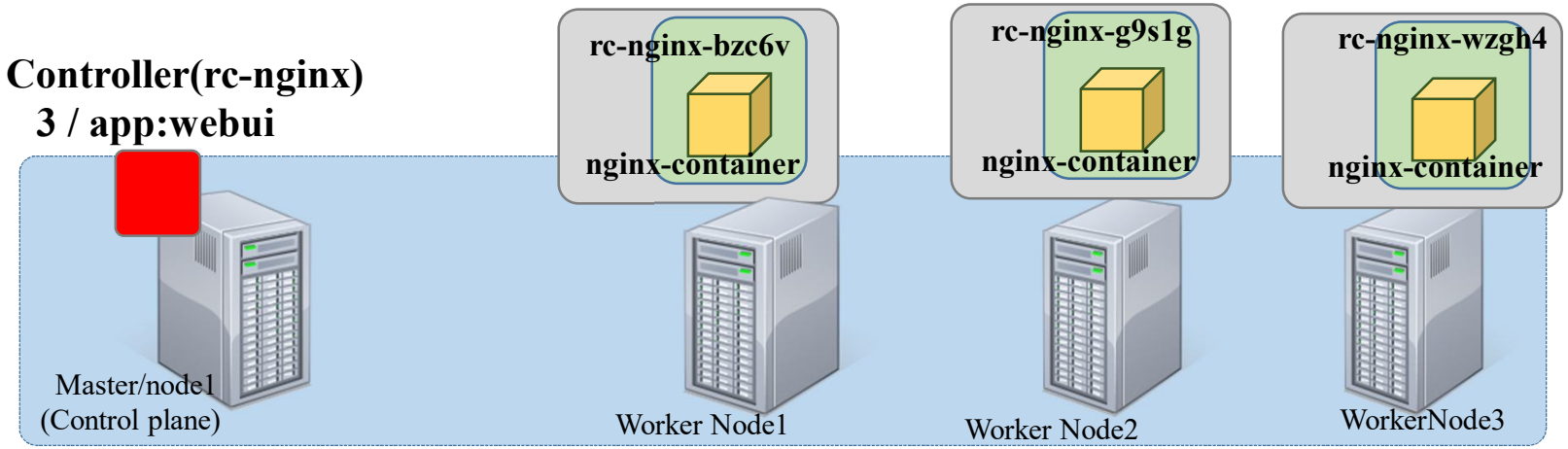
<<rc-nginx.yaml>>

kubectl create –f rc.nginx.yaml

```
[node1 ~]$ kubectl create -f rc-nginx.yaml
replicationcontroller/rc-nginx created
[node1 ~]$ kubectl get pods -o wide
NAME             READY   STATUS    RESTARTS   AGE     IP          NODE
rc-nginx-bzc6v   1/1     Running   0          2m24s   10.5.1.2    node2
rc-nginx-g9s1g   1/1     Running   0          2m24s   10.5.3.2    node3
rc-nginx-wzgh4   1/1     Running   0          2m24s   10.5.4.2    node4
[node1 ~]$
```



```
[node1 ~]$ kubectl get pods
NAME              READY   STATUS    RESTARTS   AGE
rc-nginx-bzc6v    1/1     Running   0          10m
rc-nginx-g9slg    1/1     Running   0          10m
rc-nginx-wzgh4    1/1     Running   0          10m
[node1 ~]$
[node1 ~]$ kubectl get pods --show-labels
NAME              READY   STATUS    RESTARTS   AGE   LABELS
rc-nginx-bzc6v    1/1     Running   0          11m   app=webui
rc-nginx-g9slg    1/1     Running   0          11m   app=webui
rc-nginx-wzgh4    1/1     Running   0          11m   app=webui
[node1 ~]$
[node1 ~]$ kubectl get rc
NAME       DESIRED   CURRENT   READY   AGE
rc-nginx   3         3         3       11m
```



kubectl describe rc rc-nginx

```
[node1 ~]$ kubectl describe rc rc-nginx
Name:         rc-nginx
Namespace:    default
Selector:     app=webui
Labels:       app=webui
Annotations:  <none>
Replicas:     3 current / 3 desired
Pods Status:  3 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=webui
  Containers:
   nginx-container:
    Image:        nginx:1.14
    Port:         <none>
    Host Port:    <none>
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Events:
  Type    Reason            Age    From                    Message
  ----    ------            ----   ----                    -------
  Normal  SuccessfulCreate  4m58s  replication-controller  Created pod: rc-nginx-wzgh4
  Normal  SuccessfulCreate  4m58s  replication-controller  Created pod: rc-nginx-g9slg
  Normal  SuccessfulCreate  4m58s  replication-controller  Created pod: rc-nginx-bzc6v
```

실습 2. Replication controller을 이용한 scale out/in

<<Controller Pod 생성>>

kubectl create –f rc-nginx.yaml ➔ kubectl get pods -o wide

<<scale-out>>

kubectl edit rc rc-nginx ➔ kubectl get pods -o wide

<<scale-in>>

kubectl scale rc rc-nginx --replicas=2

kubectl get pods -o wide

<<scale-out>> pod 개수를 3개에서 7개로 추가

kubectl edit rs rc-nginx

kubectl get pods -o wide

s
7
ESC
:wq!

<<scale-in>> 7개의 pod 개수를 2개로 줄임

kubectl scale rc rc-nginx --replicas=2

kubectl get pods -o wide

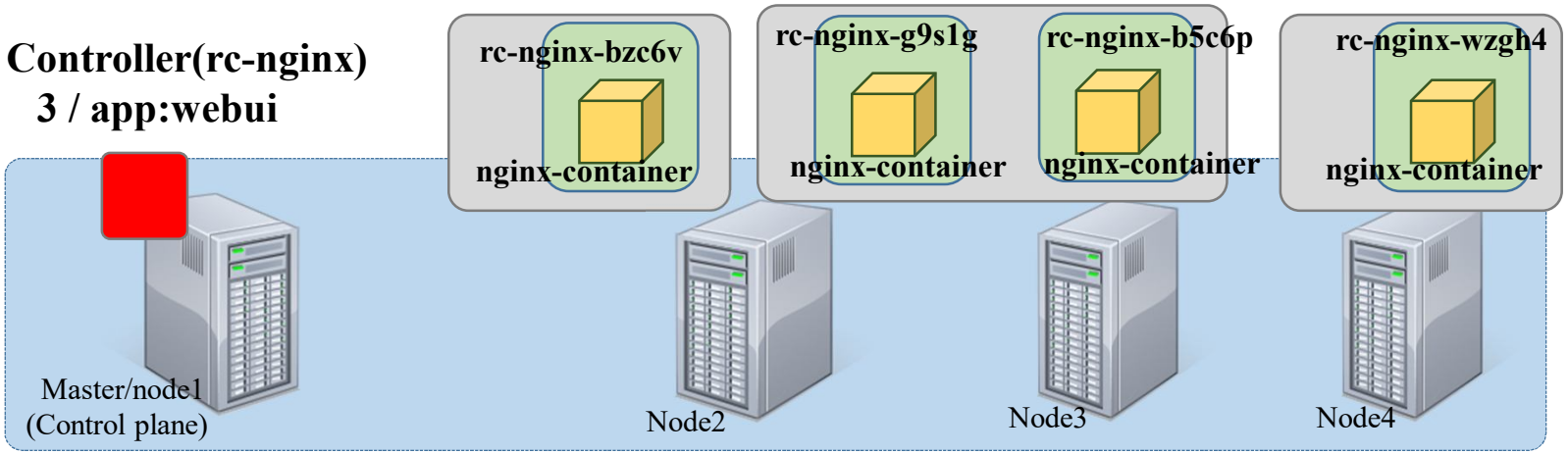
Scale-out

kubectl edit rc rc-nginx

```
# Please edit the object below. Lines beginning
# and an empty file will abort the edit. If an
# reopened with the relevant failures.
#
apiVersion: v1
kind: ReplicationController
metadata:
  creationTimestamp: "2022-05-05T01:37:34Z"
  generation: 1
  labels:
    app: webui
  name: rc-nginx
  namespace: default
  resourceVersion: "1606"
  uid: 6574e3ad-bf27-454d-87fc-8c8ad4359873
spec:
  replicas: 3
  selector:
    app: webui
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: webui
      name: nginx-pod
```

```
# Please edit the object below. Lines beginn
# and an empty file will abort the edit. If
# reopened with the relevant failures.
#
apiVersion: v1
kind: ReplicationController
metadata:
  creationTimestamp: "2022-05-05T01:37:34Z"
  generation: 1
  labels:
    app: webui
  name: rc-nginx
  namespace: default
  resourceVersion: "1606"
  uid: 6574e3ad-bf27-454d-87fc-8c8ad4359873
spec:
  replicas: 4
  selector:
    app: webui
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: webui
      name: nginx-pod
```

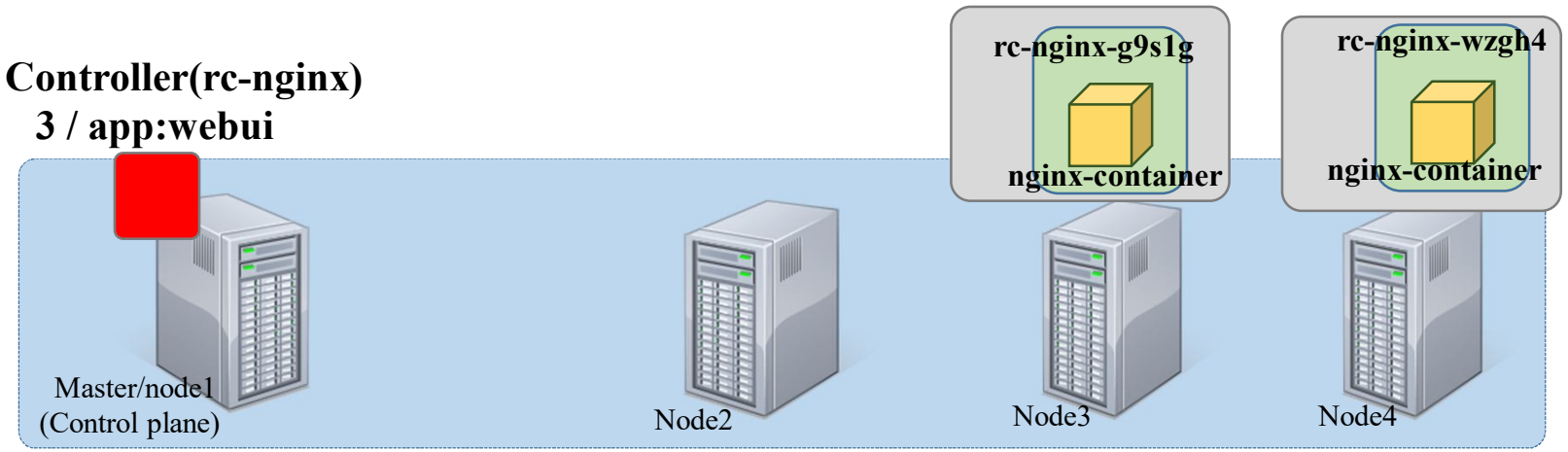
```
[node1 ~]$ kubectl edit rc rc-nginx
replicationcontroller/rc-nginx edited
[node1 ~]$ kubectl get pods
NAME              READY   STATUS    RESTARTS   AGE
rc-nginx-b5c6p    1/1     Running   0          11s
rc-nginx-bzc6v    1/1     Running   0          18m
rc-nginx-g9slg    1/1     Running   0          18m
rc-nginx-wzgh4    1/1     Running   0          18m
[node1 ~]$ kubectl get pods -o wide
NAME              READY   STATUS    RESTARTS   AGE   IP          NODE
rc-nginx-b5c6p    1/1     Running   0          25s   10.5.3.3    node3
rc-nginx-bzc6v    1/1     Running   0          18m   10.5.1.2    node2
rc-nginx-g9slg    1/1     Running   0          18m   10.5.3.2    node3
rc-nginx-wzgh4    1/1     Running   0          18m   10.5.4.2    node4
[node1 ~]$
```



Scale-in

kubectl scale rc rc-nginx --replicas=2
kubectl get pods -o wide

```
[node1 ~]$ kubectl get pods -o wide
NAME             READY   STATUS    RESTARTS   AGE   IP          NODE    NOMINATED NODE
rc-nginx-b5c6p   1/1     Running   0          25s   10.5.3.3    node3   <none>
rc-nginx-bzc6v   1/1     Running   0          18m   10.5.1.2    node2   <none>
rc-nginx-g9slg   1/1     Running   0          18m   10.5.3.2    node3   <none>
rc-nginx-wzgh4   1/1     Running   0          18m   10.5.4.2    node4   <none>
[node1 ~]$
[node1 ~]$ kubectl scale rc rc-nginx --replicas=2
replicationcontroller/rc-nginx scaled
[node1 ~]$ kubectl get pods -o wide
NAME             READY   STATUS    RESTARTS   AGE   IP          NODE    NOMINATED NODE
rc-nginx-g9slg   1/1     Running   0          20m   10.5.3.2    node3   <none>
rc-nginx-wzgh4   1/1     Running   0          20m   10.5.4.2    node4   <none>
[node1 ~]$
```



실습 3. Replication controller을 이용한 rolling update

<<Controller Pod 생성>>

kubectl create –f rc-nginx.yaml ➔ kubectl get pods -o wide

<<rolling-update>>

- kubectl edit rc rc-nginx
- kubectl describe pods rc-nginx-XXXX
- kubectl delete pod rc-nginx-XXXX
- kubectl get pods -o wide
- kubectl describe pods rc-nginx-XXXX

Rolling-update

- Pod들을 서비스 중단 없이(무중단) 애플리케이션 버전을 점진적으로 교체하는 것
- 점진적으로 서비스 중단 없이 새로운 버전으로 업데이트

RC(ReplicationController) Rolling-update

- **kubectl 명령 기반의 수동 롤링 업데이트**
- 자체적으로 롤링 업데이트 기능 없음
- 배포 이력을 저장하지 않기 때문에 롤백(Rollback)이 불가능
  - 업데이트·변경으로 문제가 발생했을 때, 이전의 정상 상태(버전)로 되돌리는 작업
  - 시스템 변경 후 장애가 발생하면, 이전에 검증된 안정 버전으로 되돌리는 복구

Rolling-update(1)

kubectl edit rc rc-nginx

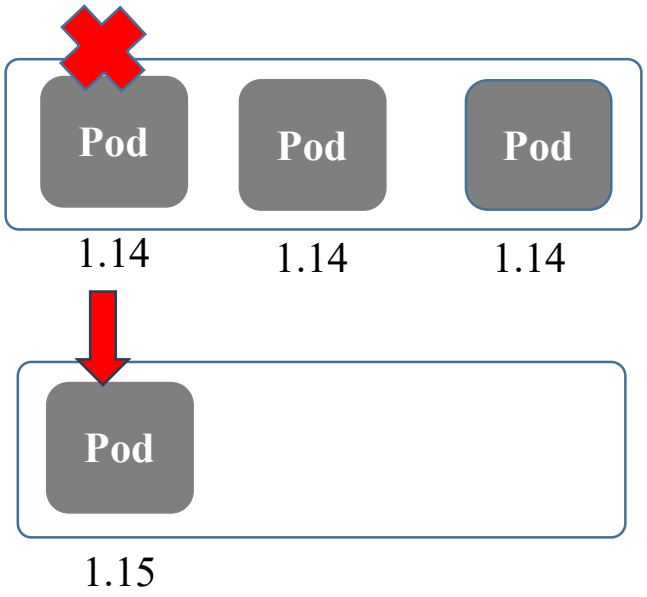
```
s
5
 ESC
:wq!
```

```
spec:
  containers:
  - image: nginx:1.14
    imagePullPolicy: IfNotPresent
    name: nginx-container
    resources: {}
    terminationMessagePath: /dev/termination-log
    terminationMessagePolicy: File
  dnsPolicy: ClusterFirst
  restartPolicy: Always
  schedulerName: default-scheduler
  securityContext: {}
  terminationGracePeriodSeconds: 30
```

```
spec:
  containers:
  - image: nginx:1.15
    imagePullPolicy: IfNotPresent
    name: nginx-container
    resources: {}
    terminationMessagePath: /dev/termination-log
    terminationMessagePolicy: File
  dnsPolicy: ClusterFirst
  restartPolicy: Always
  schedulerName: default-scheduler
  securityContext: {}
  terminationGracePeriodSeconds: 30
```

Rolling-update(2)

kubectl describe pods rc-nginx-XXXX




```
[node1 ~]$ kubectl get pods
NAME              READY   STATUS    RESTARTS   AGE
rc-nginx-g9slg    1/1     Running   0          31m
rc-nginx-wzgh4    1/1     Running   0          31m
[node1 ~]$ kubectl describe pods rc-nginx-g9slg
Name:         rc-nginx-g9slg
Namespace:    default
Priority:     0
Node:         node3/192.168.0.6
Start Time:   Thu, 05 May 2022 01:37:34 +0000
Labels:       app=webui
Annotations:  <none>
Status:       Running
IP:           10.5.3.2
IPs:
  IP:           10.5.3.2
Controlled By:  ReplicationController/rc-nginx
Containers:
  nginx-container:
    Container ID:   docker://0bc6a40c0b228bd8816fdb60c41
    Image:          nginx:1.14
    Image ID:       docker-pullable://nginx@sha256:f798
```

Rolling-update(3)

kubectl delete pod rc-nginx-XXXX

kubectl get pods -o wide

kubectl describe pods rc-nginx-XXXX

```
[node1 ~]$ kubectl delete pod rc-nginx-g9slg
pod "rc-nginx-g9slg" deleted
[node1 ~]$ kubectl get pods -o wide
NAME             READY   STATUS    RESTARTS   AGE   IP
rc-nginx-jq6sc   1/1     Running   0          62s   10.5.1.3
rc-nginx-wzgh4   1/1     Running   0          38m   10.5.4.2
[node1 ~]$
[node1 ~]$ kubectl describe pods rc-nginx-jq6sc
Name:         rc-nginx-jq6sc
Namespace:    default
Priority:     0
Node:         node2/192.168.0.7
Start Time:   Thu, 05 May 2022 02:14:36 +0000
Labels:       app=webui
Annotations:  <none>
Status:       Running
IP:           10.5.1.3
IPs:
  IP:           10.5.1.3
Controlled By:  ReplicationController/rc-nginx
```

실습 4. Replication controller 삭제

- kubectl delete pod rc-nginx-xxxx
- kubectl delete pod --all

- kubectl delete rc rc-nginx //controller를 삭제하면 pod 삭제
- kubectl get pods -o -wide

* Pod는 남겨두고 controller만 삭제

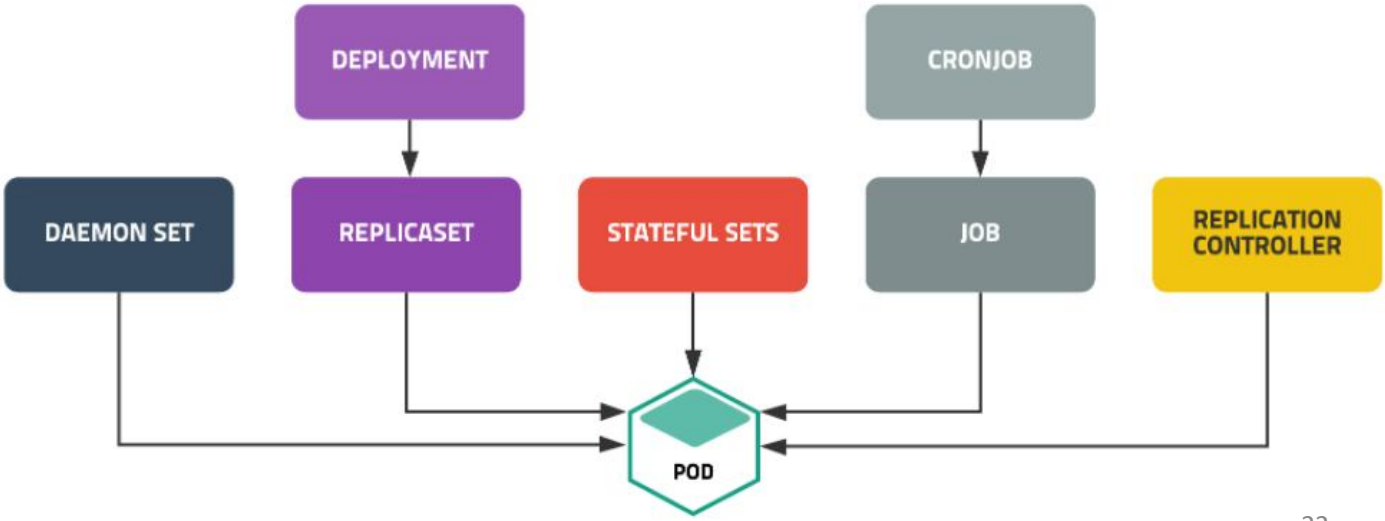
  kubectl delete rc rc-nginx --cascade=orphan

2. Replicaset

- Replication controller의 발전형으로 Replication controller와 같은 동작 수행
- Replication controller보다 풍부한 selector 지원

  - 집합기반(set-based)의 selector 지원

- rolling-update 시 deployment 사용

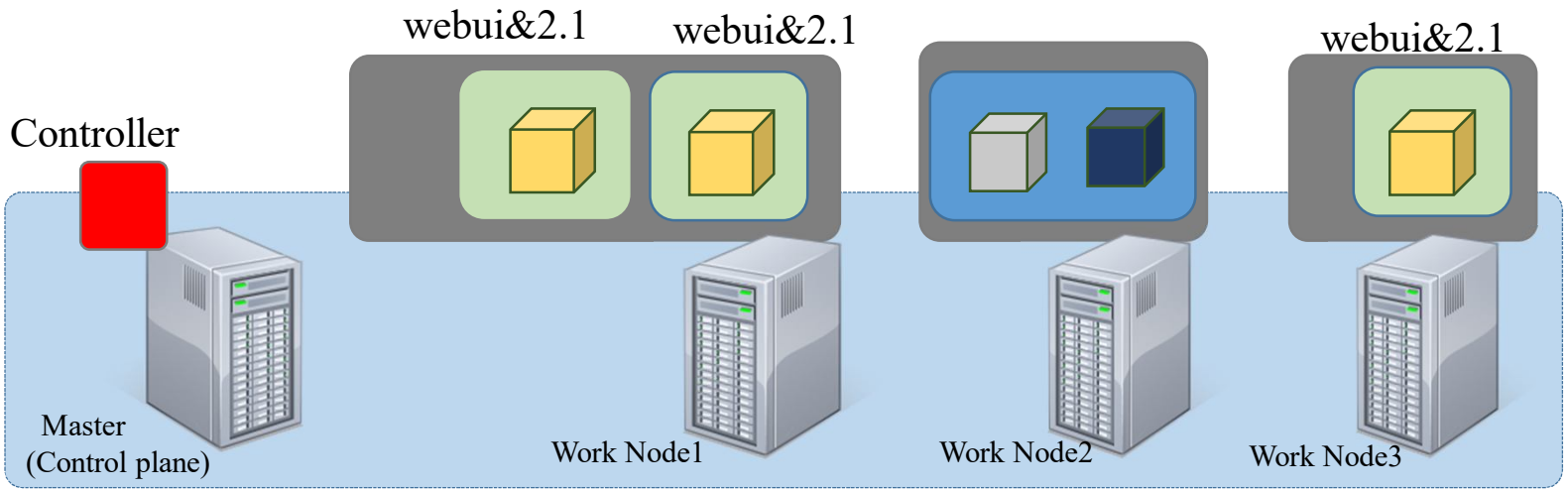


```yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: rc-nginx
spec:
  replicas: 3
  selector:
    app: webui
  template:
    metadata:
      name: nginx-pod
      labels:
        app: webui
    spec:
      containers:
      - name: nginx-container
        image: nginx:1.14
```

```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: rs-nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: webui
  template:
    metadata:
      name: nginx-pod
      labels:
        app: webui
    spec:
      containers:
      - name: nginx-container
        image: nginx:1.14
```

```
replicas: 3
selector:
  app: webui
  version: "2.1"
```

```
replicas: 3
selector:
  matchLabels:
    app:webui
  matchExpressions:
  - {key: version, operator: In, value: ["2.1" ]}
```



matchExpression
연산자(operator)

In	Key와 values를 지정하여 key, value가 일치하는 pod만연결
NotIn	Key는 일치하고 values는 일치하지 않는 pod에 연결
Exists	key에 맞는 label의 pod를 연결

```
replicas: 3
selector:
  matchLabels:
    app:webui
  matchExpressions:
  - {key: version, operator: In, value: ["2.1","2.2"]}
```

2.1버전 또는 2.2버전 존재

```
replicas: 3
selector:
  matchLabels:
    app:webui
  matchExpressions:
  - {key: version, operator: NotIn, value: ["2.1","2.2"]}
```

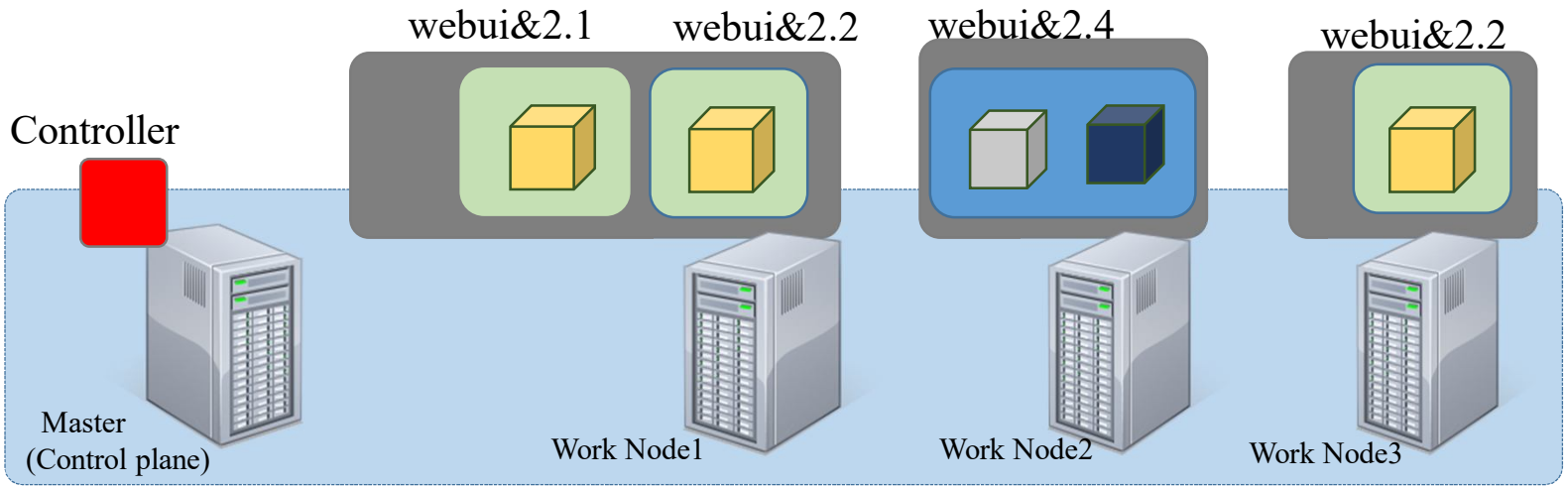
버전이 존재해서는 안됨

```
replicas: 3
selector:
  matchLabels:
    app:webui
  matchExpressions:
  - {key: version, operator: Exists}
```

버전이 존재만하면 됨

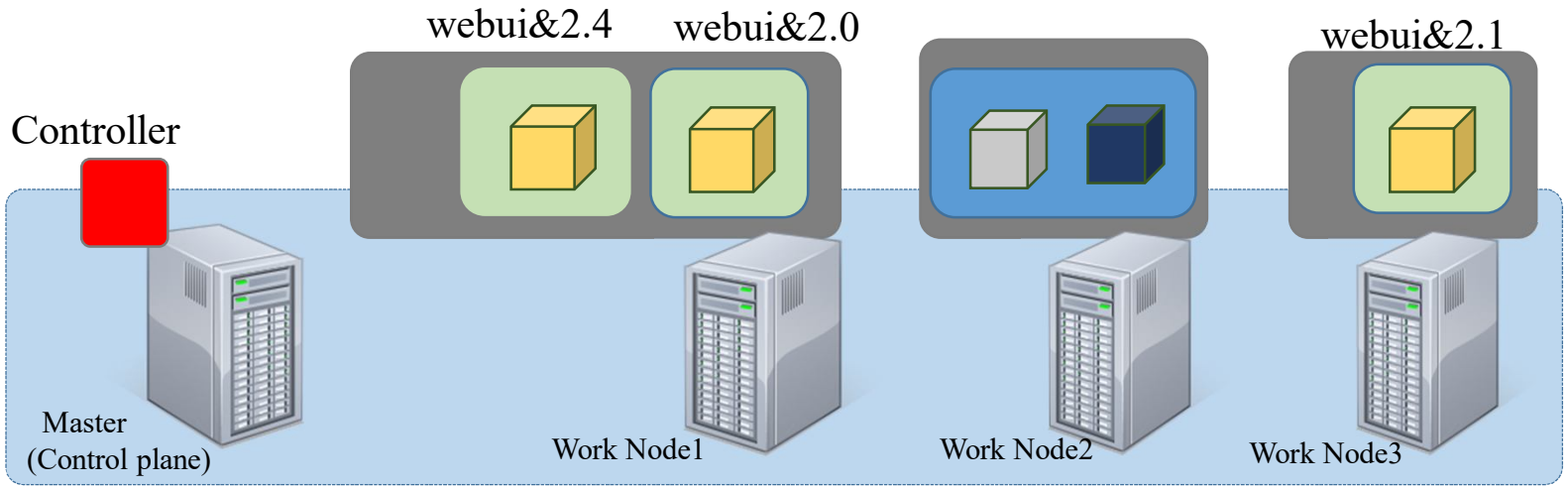
```
replicas: 3
selector:
  matchLabels:
    app:webui
  matchExpressions:
  - {key: version, operator: In, value: ["2.1","2.2"]}
```

label webui 그리고 2.1버전 또는 2.2버전인 pod가 3개 존재



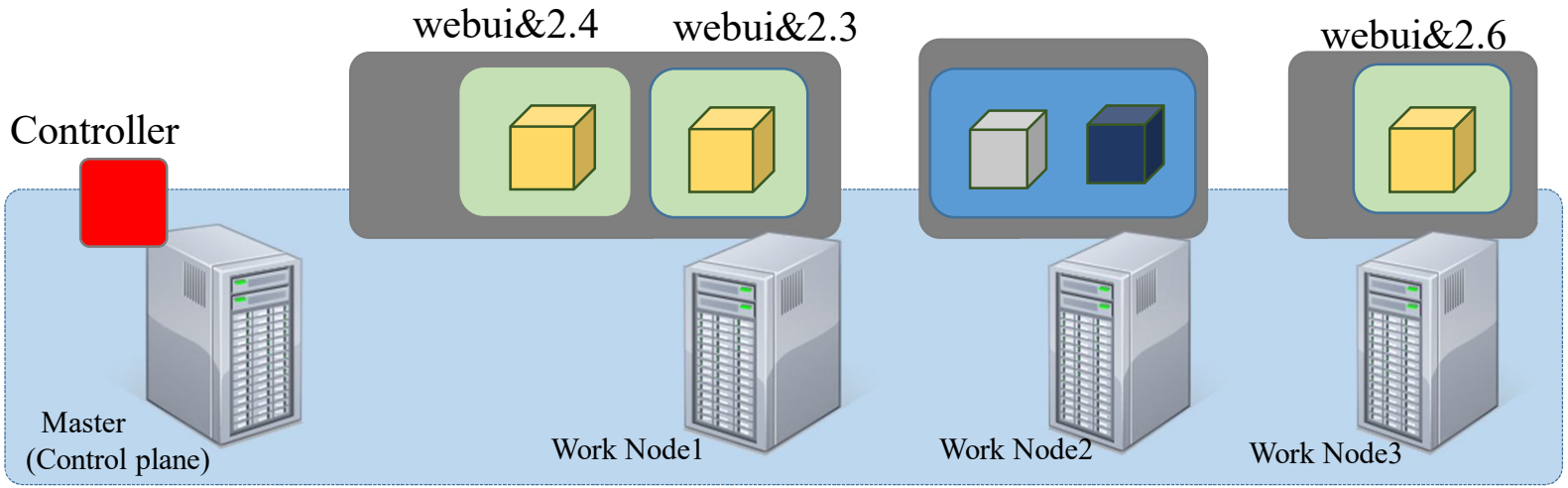
```
replicas: 3
selector:
  matchLabels:
    app:webui
  matchExpressions:
  - {key: version, operator: Exists}
```

label webui 그리고 version 값이 존재하는 pod가 3개 유지



```
replicas: 3
selector:
  matchLabels:
    app:webui
  matchExpressions:
  - {key: version, operator: NotIn, value: ["2.1","2.2"]}
```

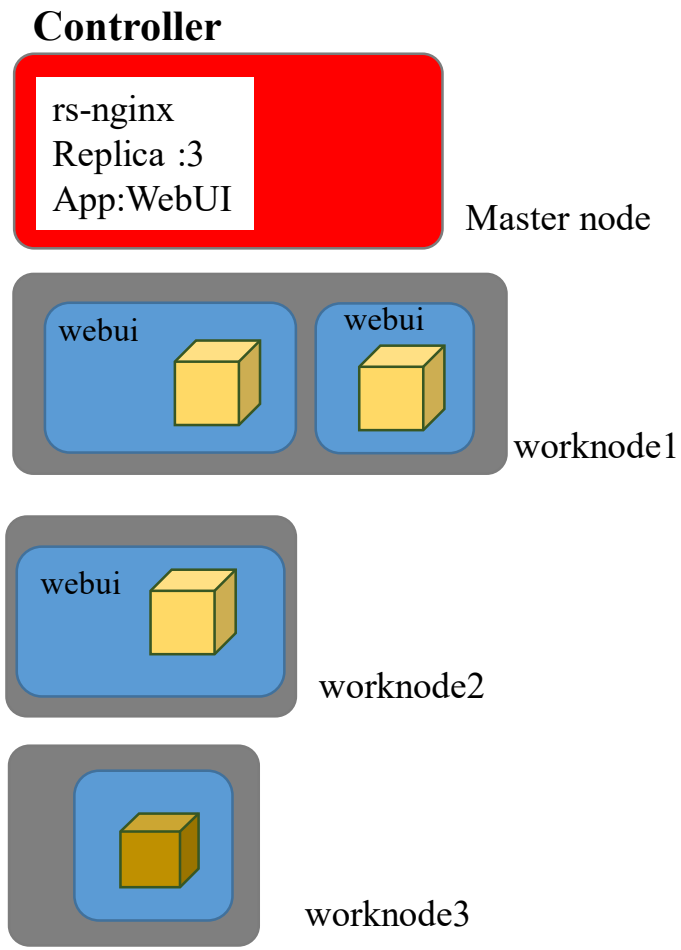
label webui 그리고 version 버전이 2.1 또는 2.2 가 아닌 pod들이 3개 존재



실습 5. ReplicaSet을 이용해서 Pod를 항상 3개를 유지하도록 한다.

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: rs-nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: webui
  template:
    metadata:
      name: nginx-pod
      labels:
        app: webui
    spec:
      containers:
      - name: nginx-container
        image: nginx:1.14
```

<<rs-nginx.yaml>>



<<ReplicaSet Controller와 Pod 생성>>

kubectl delete rc rc-nginx //controller를 삭제하면 pod 삭제

kubectl get pods -o -wide

kubectl create -f rs-nginx.yaml 또는 kebectl apply -f rs-nginx

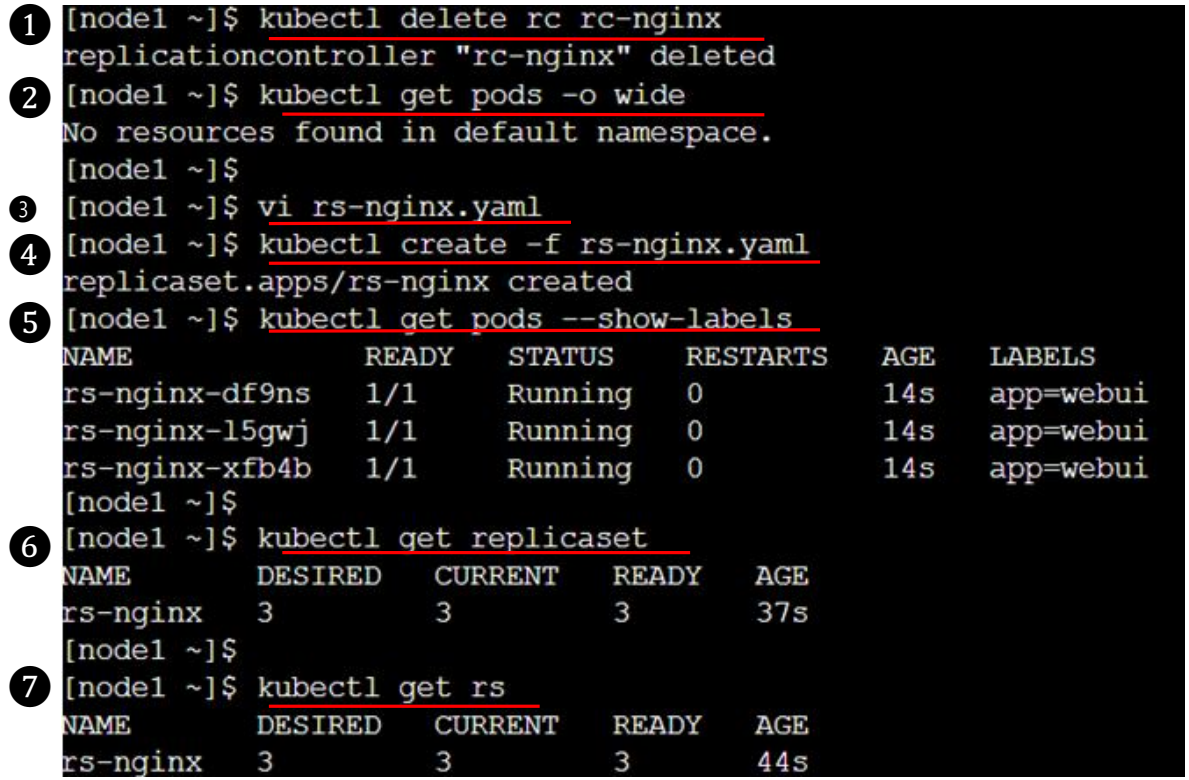
kubectl get pods --show-lable

kubectl get rs

kubectl create -f rs-nginx.yaml 또는 kebectl apply -f rs-nginx

kubectl get pods --show-label

kubectl get rs



kubectl delete pods rs-nginx-XXXXX

kubectl get pods --show-lable

```
❶ [node1 ~]$ kubectl delete pods rs-nginx-df9ns
pod "rs-nginx-df9ns" deleted
❷ [node1 ~]$ kubectl get pods --show-labels
NAME              READY   STATUS    RESTARTS   AGE   LABELS
rs-nginx-l5gwj    1/1     Running   0          91s   app=webui
rs-nginx-lvn54    1/1     Running   0          15s   app=webui
rs-nginx-xfb4b    1/1     Running   0          91s   app=webui
```

kubectl scale rs rs-nginx --replicas=2

kubectl get pods --show-lable

```
❶ [node1 ~]$ kubectl scale rs rs-nginx --replicas=2
replicaset.apps/rs-nginx scaled
❷ [node1 ~]$ kubectl get pods --show-labels
NAME              READY   STATUS        RESTARTS   AGE     LABELS
rs-nginx-l5gwj    1/1     Running       0          2m52s   app=webui
rs-nginx-lvn54    0/1     Terminating   0          96s     app=webui
rs-nginx-xfb4b    1/1     Running       0          2m52s   app=webui
❸ [node1 ~]$ kubectl get pods --show-labels
NAME              READY   STATUS    RESTARTS   AGE   LABELS
rs-nginx-l5gwj    1/1     Running   0          3m3s  app=webui
rs-nginx-xfb4b    1/1     Running   0          3m3s  app=webui
```

2) Replicaset와 Pod와의 연관 관계

* Pod는 남겨두고 controller만 삭제

kubectl delete rs rs-nginx --cascade=orphan

kubectl delete replicaset rs-nginx --cascade=orphan

ReplicaSet
Controller

```
[node1 ~]$ kubectl delete rs rs-nginx --cascade=false
warning: --cascade=false is deprecated (boolean value) and can be replaced with --cascade=orphan.
replicaset.apps "rs-nginx" deleted
[node1 ~]$ kubectl get rs
No resources found in default namespace.
[node1 ~]$ kubectl get pods --show-labels
NAME             READY   STATUS    RESTARTS   AGE   LABELS
rs-nginx-l5gwj   1/1     Running   0          13m   app=webui
rs-nginx-xfb4b   1/1     Running   0          13m   app=webui
[node1 ~]$
```

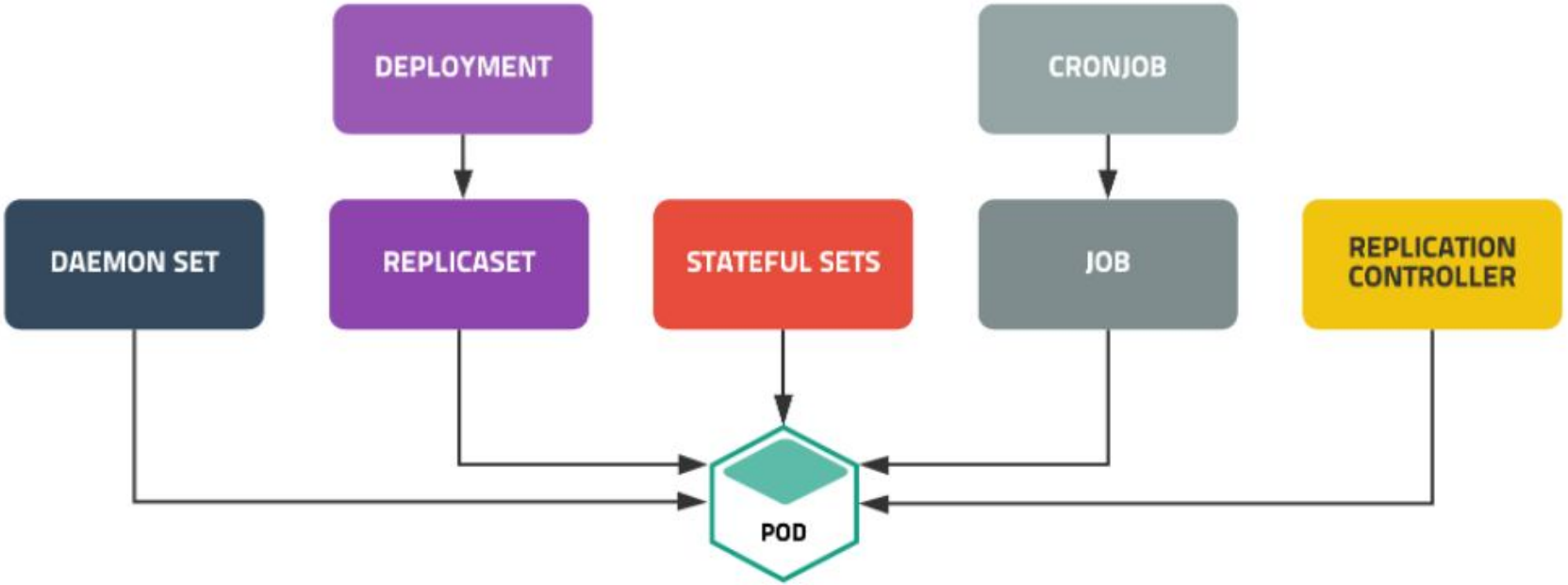
* kubectl delete rs rs-nginx

3. Deployment

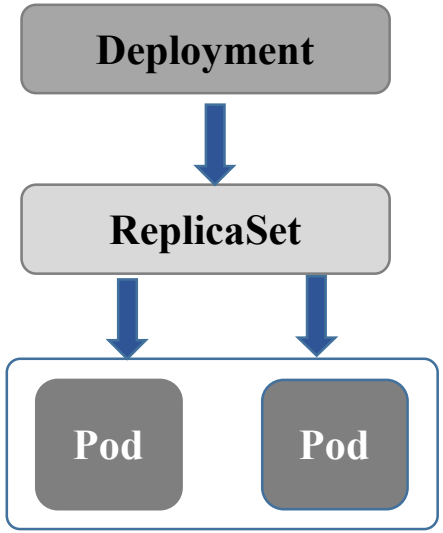
- ReplicaSet을 제어해주는 부모역할
- ReplicaSet을 컨트롤해서 Pod 수를 조절
- Rolling update & Rolling Back

Rolling Update

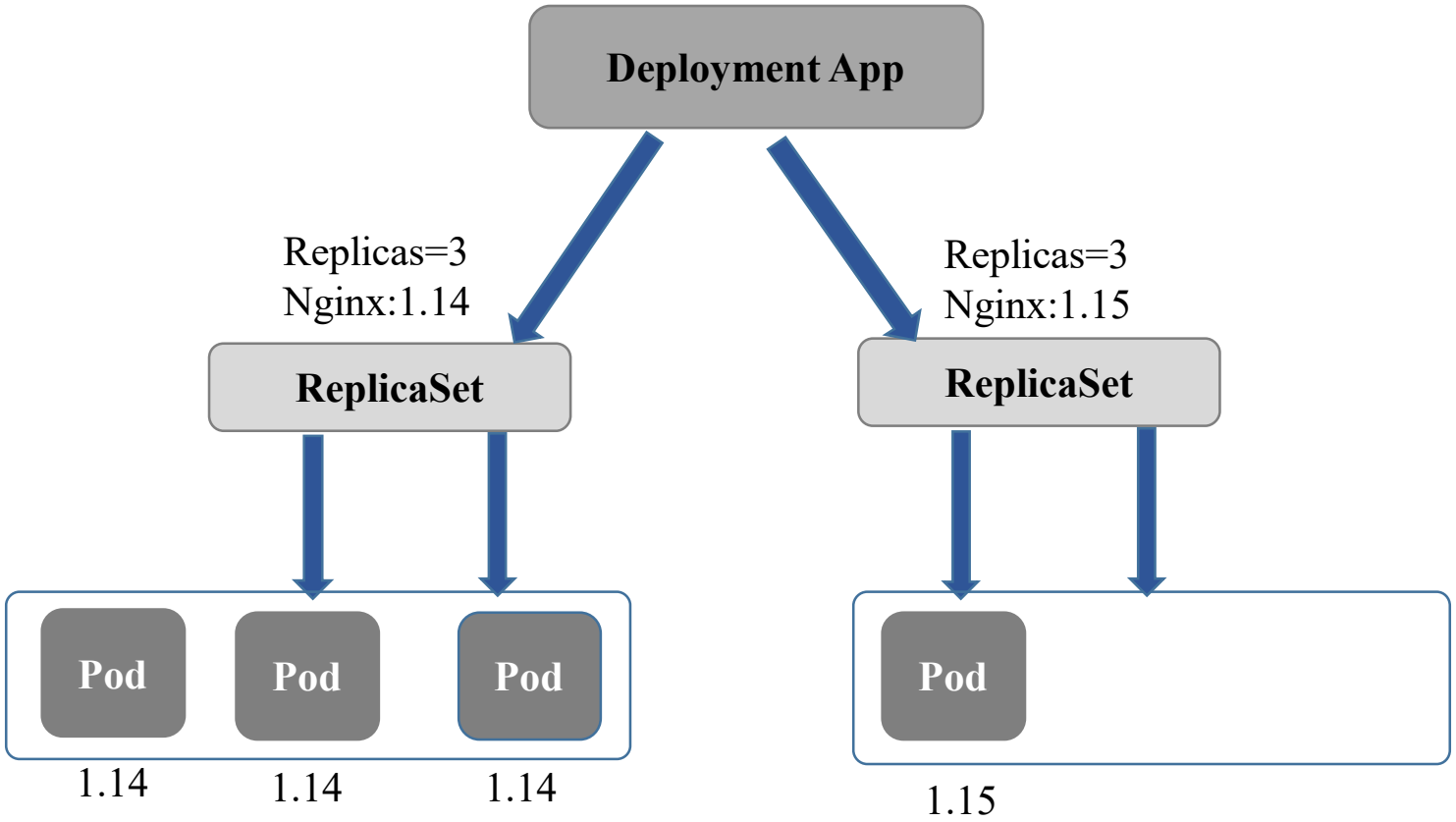
점진적으로 서비스 중단 없이 새로운 버전으로 업데이트



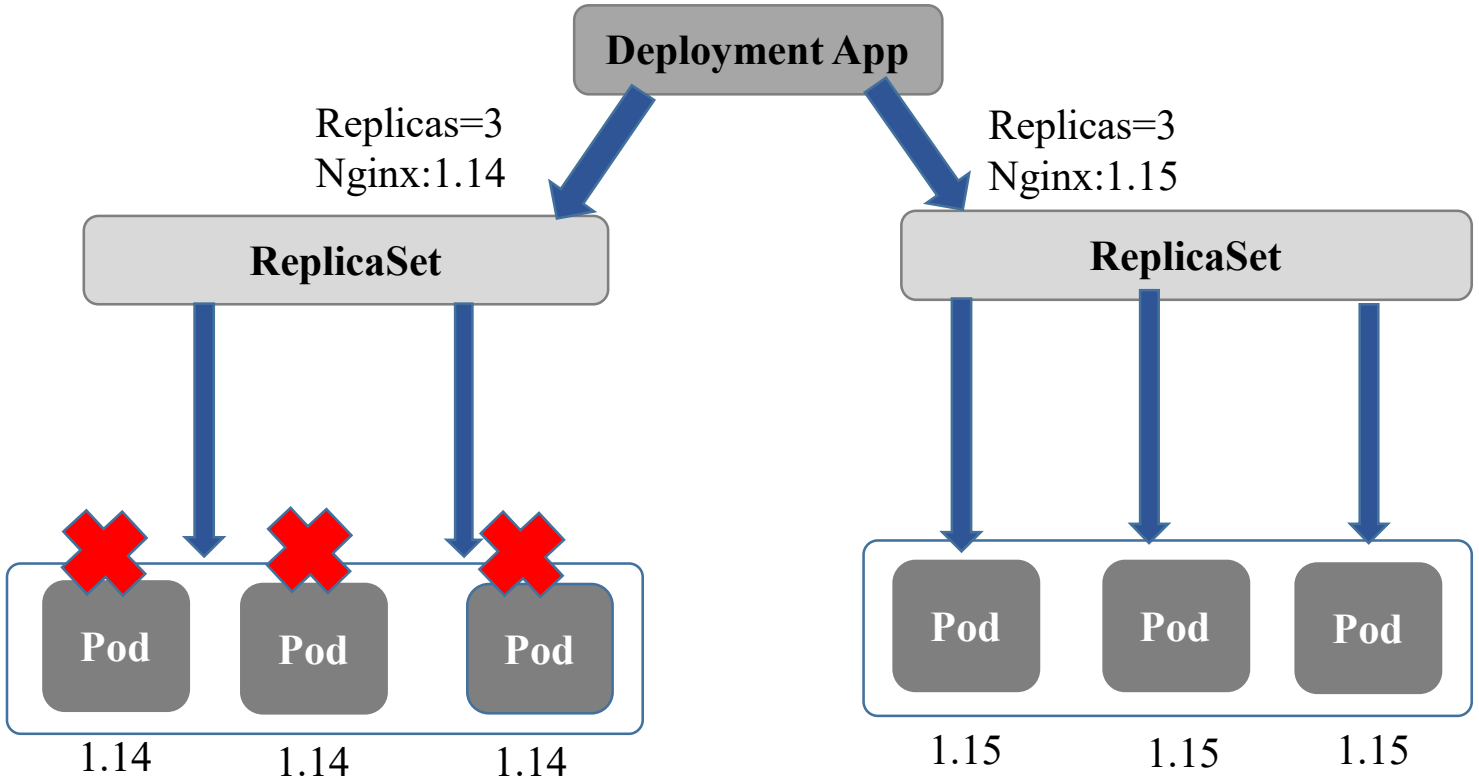
- ReplicaSet을 컨트롤해서 Pod수를 조절 실행시켜야 할 pod 개수를 유지
- Replicaset를 관리하면서 App 배포를 더 세밀하게 관리
  - App을 배포 시 rolling update
  - App 배포 도중 잠시 멈추거나 다시 배포 가능
  - App 배포 이후 이전 버전으로 rolling back 가능



kubectl set image deployment app-deploy app=nginx:1.15 --record

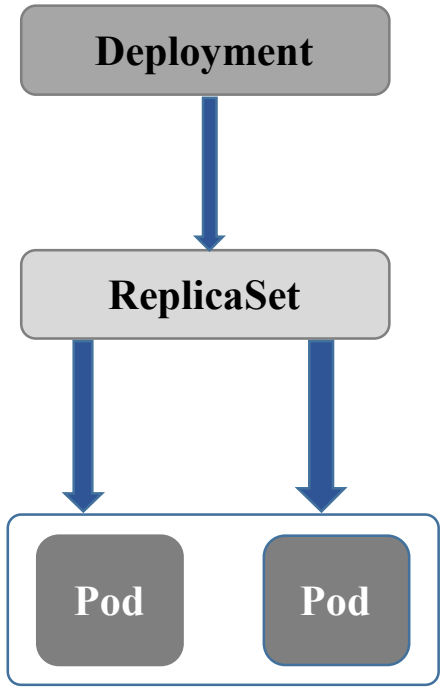


kubectl set image deployment app-deploy app=nginx:1.15 --record



<< rs-nginx.yaml >>

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: rs-nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: webui
  template:
    metadata:
      name: nginx-pod
      labels:
        app: webui
    spec:
      containers:
      - name: nginx-container
        image: nginx:1.14
```



<< deploy-nginx.yaml >>

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deploy-nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: webui
  template:
    metadata:
      name: nginx-pod
      labels:
        app: webui
    spec:
      containers:
      - name: nginx-container
        image: nginx:1.14
```

실습 6. Deployment를 이용한 Pod 생성/유지

<< deploy-nginx.yaml >>

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deploy-nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: webui
  template:
    metadata:
      name: nginx-pod
      labels:
        app: webui
    spec:
      containers:
      - name: nginx-container
        image: nginx:1.14
```

<<Deployment 생성>>
kubectl create -f deploy-nginx.yaml
kubectl get pods -o wide
kubectl get deploy,rs,pod

<<pod 삭제>>
kubectl delete pod deploy-nginx-XXXX-XXX

<<replicaset 삭제>>
kubectl get rs
kubectl get rs deploy-nginx-XXXX

<<Deployment 삭제>>
kubectl get deploy
kubectl delete deployment deploy-nginx

kubectl create –f deploy-nginx.yaml

kubectl get pods –o wide

kubectl get deploy,rs,pod

```
[node1 ~]$ vi deploy-nginx.yaml
[node1 ~]$ kubectl create -f deploy-nginx.yaml
deployment.apps/deploy-nginx created
[node1 ~]$ kubectl get pods -o wide
NAME                            READY   STATUS    RESTARTS   AGE   IP          NODE
deploy-nginx-6d4c4cc4b8-4mjrg   1/1     Running   0          31s   10.5.3.2    node4
deploy-nginx-6d4c4cc4b8-dnlcc   1/1     Running   0          31s   10.5.2.2    node3
deploy-nginx-6d4c4cc4b8-xz2s9   1/1     Running   0          31s   10.5.1.2    node2
[node1 ~]$
[node1 ~]$ kubectl get deploy,rs,pod
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/deploy-nginx   3/3     3            3           45s

NAME                                      DESIRED   CURRENT   READY   AGE
replicaset.apps/deploy-nginx-6d4c4cc4b8   3         3         3       45s

NAME                                READY   STATUS    RESTARTS   AGE
pod/deploy-nginx-6d4c4cc4b8-4mjrg   1/1     Running   0          45s
pod/deploy-nginx-6d4c4cc4b8-dnlcc   1/1     Running   0          45s
pod/deploy-nginx-6d4c4cc4b8-xz2s9   1/1     Running   0          45s
[node1 ~]$
```

kubectl delete pods deploy-nginx-XXXXX-XXXXXX

kubectl get pods

kubectl get rs deploy-nginx-XXXX

```
[node1 ~]$ kubectl delete pods deploy-nginx-6d4c4cc4b8-4mjrg
pod "deploy-nginx-6d4c4cc4b8-4mjrg" deleted
[node1 ~]$ kubectl get pods
NAME                            READY   STATUS    RESTARTS   AGE
deploy-nginx-6d4c4cc4b8-dnlcc   1/1     Running   0          5m44s
deploy-nginx-6d4c4cc4b8-q4dft   1/1     Running   0          41s
deploy-nginx-6d4c4cc4b8-xz2s9   1/1     Running   0          5m44s
[node1 ~]$ kubectl get rs deploy-nginx-6d4c4cc4b8
NAME                      DESIRED   CURRENT   READY   AGE
deploy-nginx-6d4c4cc4b8   3         3         3       7m9s
[node1 ~]$
```

<<replicaset 삭제>>

kubectl get rs

kubectl delete deploy-nginx-XXXX

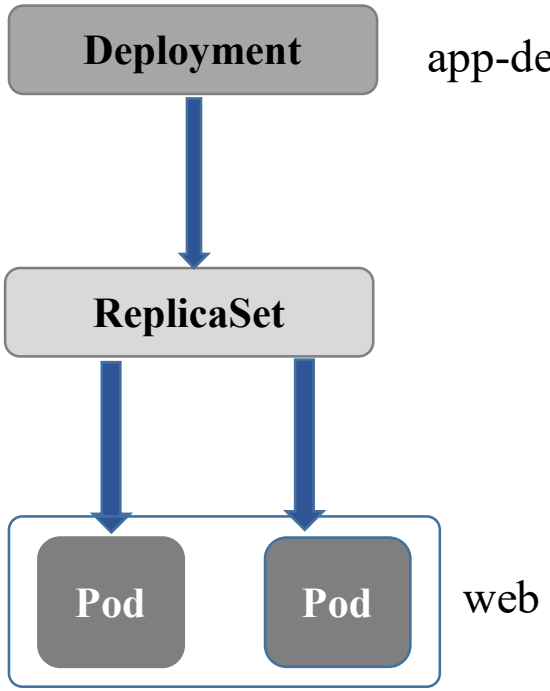
<<Deployment 삭제>>

kubectl get deploy

kubectl delete deployment deploy-nginx

실습 7. Deployment Rolling Update

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: app-deploy
spec:
  selector:
    matchLabels:
      app: webui
  replicas: 3
  template:
    metadata:
      labels:
        app: webui
    spec:
      containers:
      - image: nginx:1.14
        name: web
        ports:
        - containerPort: 80
```



<< deployment-exam1.yaml >>

kubectl create –f deployment-exam1.yam **--record**

kubectl get deployment,rs,pod

```
controlplane $ vi deployment-exam1.yaml
controlplane $ kubectl create -f deployment-exam1.yaml --record
deployment.apps/app-deploy created
controlplane $ kubectl get deployment,rs,pod
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/app-deploy     0/3     3            0           14s

NAME                                      DESIRED   CURRENT   READY   AGE
replicaset.apps/app-deploy-7f6fffdcdd     3         3         0       14s

NAME                                  READY   STATUS              RESTARTS   AGE
pod/app-deploy-7f6fffdcdd-55vmx       0/1     ContainerCreating   0          14s
pod/app-deploy-7f6fffdcdd-6hh55       0/1     ContainerCreating   0          14s
pod/app-deploy-7f6fffdcdd-lql4w       0/1     ContainerCreating   0          14s
controlplane $
```

```
NAME                                READY   STATUS              RESTARTS   AGE
pod/app-deploy-7f6fffdcdd-55vmx     0/1     ContainerCreating   0          14s
pod/app-deploy-7f6fffdcdd-6hh55     0/1     ContainerCreating   0          14s
pod/app-deploy-7f6fffdcdd-lql4w     0/1     ContainerCreating   0          14s
```

kubectl describe pods app-deploy-XXXXX-XXXX

```
controlplane $ kubectl describe pods app-deploy-7f6fffdcdd-55vmx
Name:         app-deploy-7f6fffdcdd-55vmx
Namespace:    default
Priority:     0
Node:         node01/10.0.0.10
Start Time:   Wed, 11 May 2022 11:32:13 +0000
Labels:       app=webui
              pod-template-hash=7f6fffdcdd
Annotations:  <none>
Status:       Running
IP:           10.244.1.3
IPs:
  IP:           10.244.1.3
Controlled By:  ReplicaSet/app-deploy-7f6fffdcdd
Containers:
  web:
    Container ID:   docker://4d318941b1917aaacbe3d3a5aa6a9ab7a45656db96c16792754
    Image:          nginx:1.14
    Image ID:       docker-pullable://nginx@sha256:f7988fb6c02e0ce69257d9bd9cf37
```

kubectl set image deployment app-deploy web=nginx:1.15 --record

```
controlplane $ kubectl set image deployment app-deploy web=nginx:1.15 --record
deployment.apps/app-deploy image updated
controlplane $ kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
app-deploy-67777997f5-689lv   1/1     Running   0          9s
app-deploy-7f6fffdcdd-55vmx   1/1     Running   0          3m2s
app-deploy-7f6fffdcdd-6hh55   1/1     Running   0          3m2s
app-deploy-7f6fffdcdd-lql4w   1/1     Running   0          3m2s
controlplane $ kubectl get pods
NAME                          READY   STATUS              RESTARTS   AGE
app-deploy-67777997f5-689lv   1/1     Running             0          11s
app-deploy-67777997f5-15wwm   0/1     ContainerCreating   0          2s
app-deploy-7f6fffdcdd-55vmx   1/1     Running             0          3m4s
app-deploy-7f6fffdcdd-6hh55   1/1     Running             0          3m4s
app-deploy-7f6fffdcdd-lql4w   1/1     Terminating         0          3m4s
```

kubectl get pods

```
controlplane $ kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
app-deploy-67777997f5-689lv   1/1     Running   0          46s
app-deploy-67777997f5-h9fmd   1/1     Running   0          34s
app-deploy-67777997f5-15wwm   1/1     Running   0          37s
controlplane $
```

```
controlplane $ kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
app-deploy-67777997f5-689lv   1/1     Running   0          46s
app-deploy-67777997f5-h9fmd   1/1     Running   0          34s
app-deploy-67777997f5-15wwm   1/1     Running   0          37s
controlplane $
```

```
controlplane $ kubectl describe pods app-deploy-67777997f5-689lv
Name:         app-deploy-67777997f5-689lv
Namespace:    default
Priority:     0
Node:         node01/10.0.0.10
Start Time:   Wed, 11 May 2022 11:35:06 +0000
Labels:       app=webui
              pod-template-hash=67777997f5
Annotations:  <none>
Status:       Running
IP:           10.244.1.6
IPs:
  IP:           10.244.1.6
Controlled By:  ReplicaSet/app-deploy-67777997f5
Containers:
  web:
    Container ID:   docker://1000a6255b6986d395a1de9f2a724ac3ea26d8bb6b290d31ee6
    Image:          nginx:1.15
    Image ID:       docker-pullable://nginx@sha256:23b4dcdf0d34d4a129755fc6f52e1
```

Deployment Rolling Update

kubectl set image deploy app-deploy web=nginx:1.16 --record
kubectl set image deploy app-deploy web=nginx:1.17 --record
kubectl set image deploy app-deploy web=nginx:1.15 --record

kubectl rollout status deployment app-deploy
kubectl rollout pause deployment app-deploy
kubectl rollout resume deployment app-deploy

kubectl rollout history deploy *app-deploy*

```
controlplane $ kubectl rollout history deploy app-deploy
deployment.apps/app-deploy
REVISION  CHANGE-CAUSE
7         kubectl create --filename=deployment-exam1.yaml --record=true
10        kubectl set image deployment app-deploy web=nginx:1.16 --record=true
11        kubectl set image deployment app-deploy web=nginx:1.17 --record=true
12        kubectl set image deployment app-deploy web=nginx:1.15 --record=true
```

Deployment Rolling Update

kubectl set image deploy app-deploy web=nginx:1.16 --record
kubectl set image deploy app-deploy web=nginx:1.17 --record
kubectl set image deploy app-deploy web=nginx:1.15 --record

kubectl rollout status deployment app-deploy
kubectl rollout pause deployment app-deploy
kubectl rollout resume deployment app-deploy

kubectl rollout history deploy *app-deploy*

```
controlplane $ kubectl rollout history deploy app-deploy
deployment.apps/app-deploy
REVISION  CHANGE-CAUSE
7         kubectl create --filename=deployment-exam1.yaml --record=true
10        kubectl set image deployment app-deploy web=nginx:1.16 --record=true
11        kubectl set image deployment app-deploy web=nginx:1.17 --record=true
12        kubectl set image deployment app-deploy web=nginx:1.15 --record=true
```

실습 8. Deployment RollingBack

kubectl rollout history deploy app-deploy

kubectl rollout undo deploy app-deploy

kubectl rollout undo deploy app-deploy --to-revision=2

kubectl describe pods app-deploy-XXXXX-XXXX

kubectl create –f deployment-exam2.yaml **--record**

Kubectl get deployment,rs,pod

nano deployment-exam2.yaml

kubectl apply –f deployment-exam2.yaml

kubectl rollout history deploy deploy-nginx