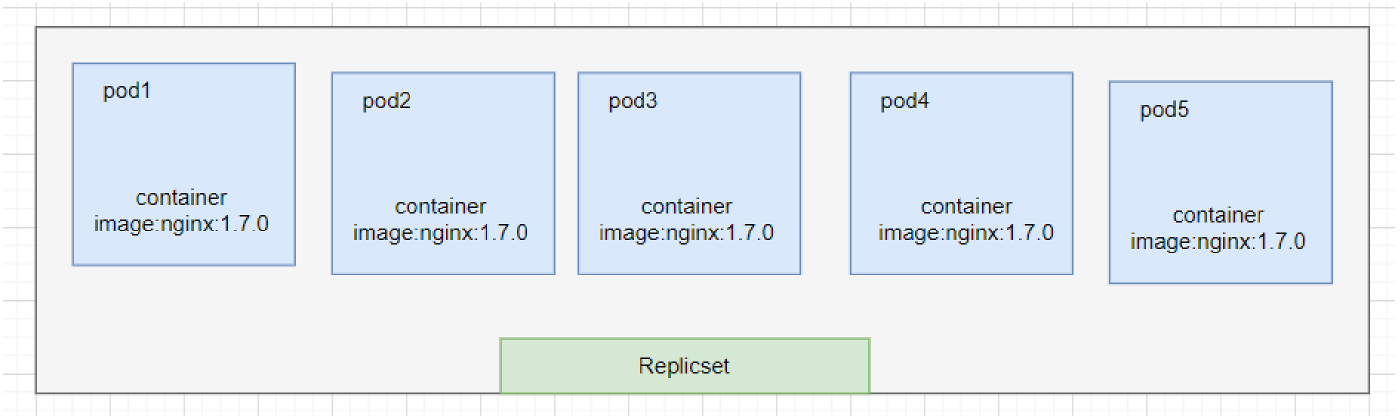

Deployment --

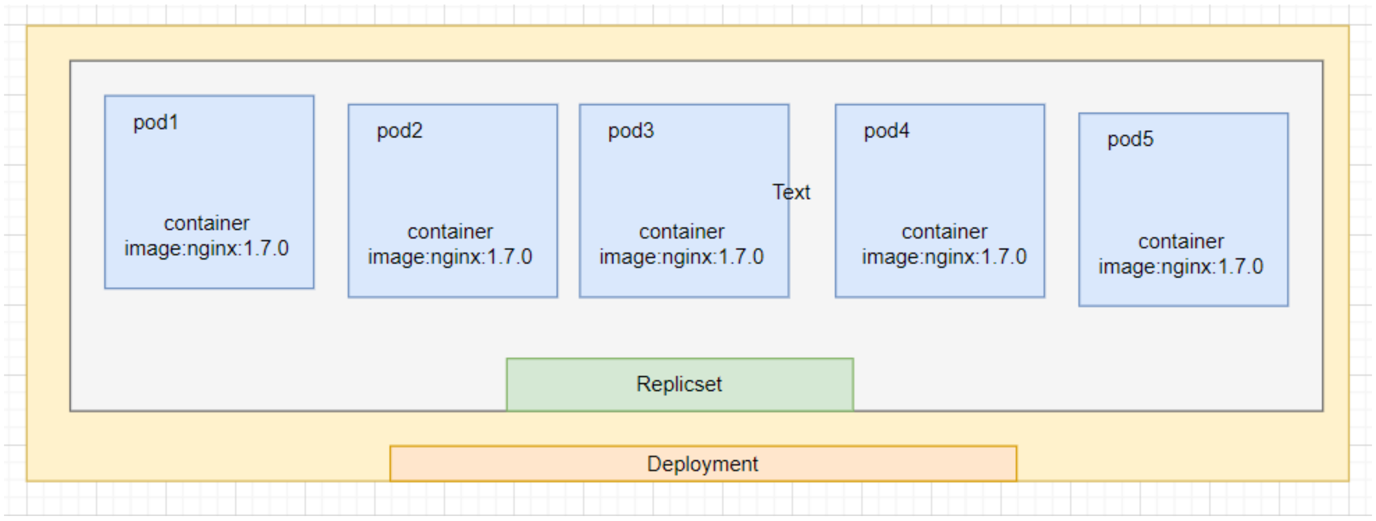
Suppose we have Replicaset that has 5 pods that running container with image that's tag is 1.7.0

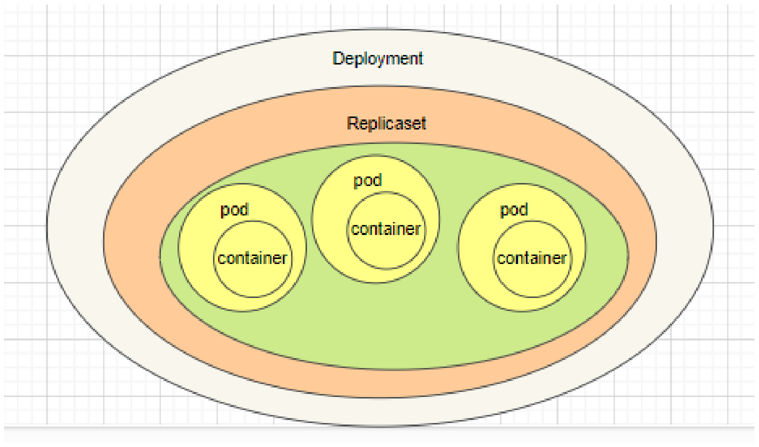
what about is there if newer version we need to use ? we need to update image version in all pods on replicaset.



We need object ------------------------> called Deployment.

Deployment -----------> k8s object allow rolling updates && rollling Back.


pod1
container image:nginx:1.7.0
pod2
container image:nginx:1.7.0
pod3
Text
container image:nginx:1.7.0
pod4
container image:nginx:1.7.0
pod5
container image:nginx:1.7.0
Replicset
Deployment



let's understand ->

by default when deployment is created there is replicaset created inside it holds number of desired pods.

Deployment strategy ----------->

1) recreate --->

when recreate strategy deployment applied -------->

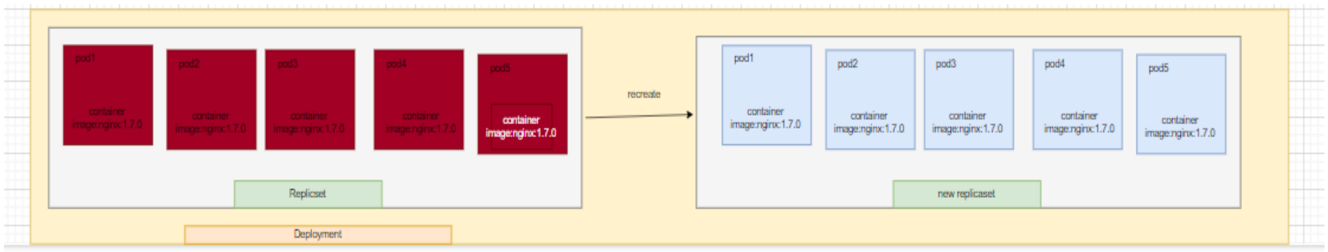
new replicaset is create and pods on old replicaset get destoryed once and new

pods are created on new replicaset .

After recreate have :

2 replicasets {old has zero pods   new has desired number of replicas}

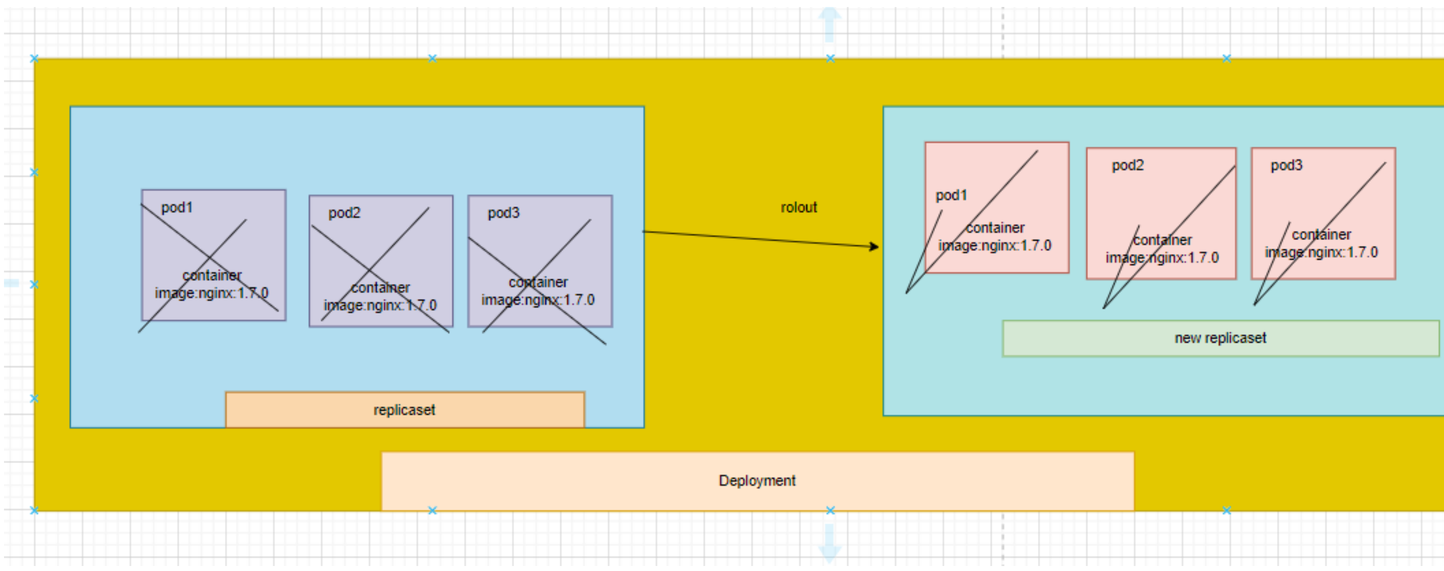
Issue -------------> Application down time


pod1
container image:nginx:1.7.0
pod2
container image:nginx:1.7.0
pod3
container image:nginx:1.7.0
pod4
container image:nginx:1.7.0
pod5
container image:nginx:1.7.0
Replicset
recreate
pod1
container image:nginx:1.7.0
pod2
container image:nginx:1.7.0
pod3
container image:nginx:1.7.0
pod4
container image:nginx:1.7.0
pod5
container image:nginx:1.7.0
new replicaset
Deployment

2) roll update --> deployment strategy in which new replicaset is created but we have down one pod on old replicaset and provision new one on new replicaset and this process repeated until all pods on old replicaset got down and new replicaset has all desired number of pods.

Application is accessiable all time.(No down time)

Default deployment strategy is roll update.



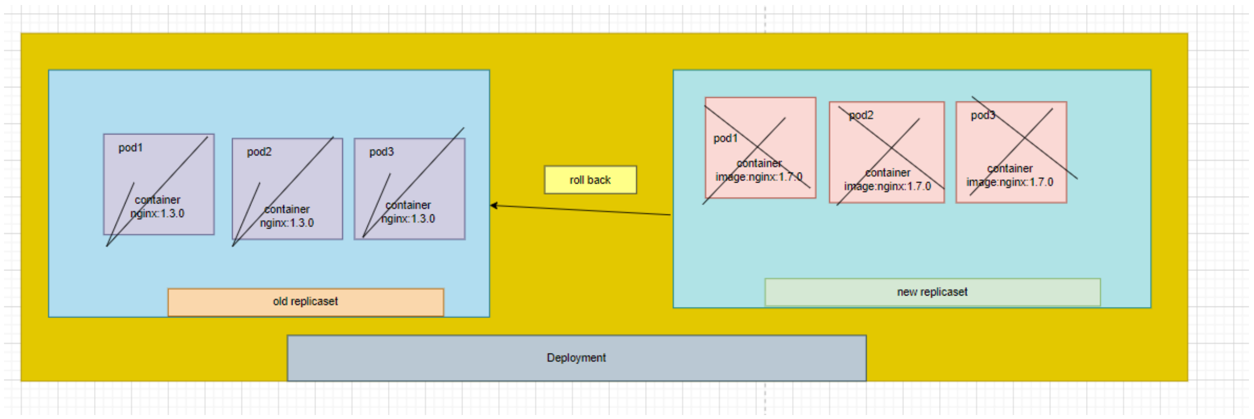
What about our newer version of image has issue and we want to rollback ?

rollback ----------->

we have 2 replicasets old replica has zero pods && new replica has 3 pods .

when rollback is triggered pods on new replicaset one by one is destoryed and for each one destored on new replicaset new pod is created on old replicaset with old version of image.

At end of roll back process new replicaset has zero pods but old replicset has 3 pods.



How to create deployment (imperative way) --------------?

$ kubectl create deployment deployment_name --image=image_name --replicas=n

To check rollout status

```
$ kubectl rollout status
deployment/deployment_name
```

```
$ kubectl rollout history
deployment/deployment_name
```

------------yaml file for Deployment -----------

yaml file consist from 4 main component

1) apiVersion ---------> version of object we try to create

2) kind ---------------> object kind {Deployment || POd || Replicaset || ReplicationController}

3) metadata ------------> data about object we try to create {name || namespace || labels}

4) spec --> specification for object we try to create

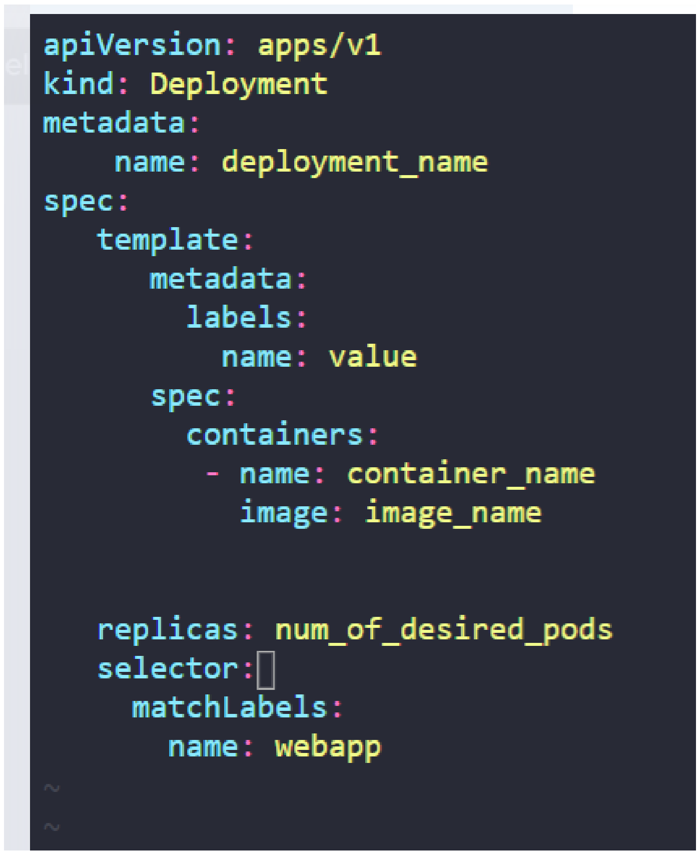
consist from

4.1) replicas: num_of_replicas

4.2) strategy : Deployment strategy {Roll update or recreate}

4.3) selector {to identify pods that replicaset can manage that not create as part of replicaset}

4.4) template {consist of metadata && spec that defined on pod definition}



```
apiVersion: apps/v1
kind: Deployment
metadata:
    name: deployment_name
spec:
   template:
      metadata:
        labels:
          name: value
      spec:
        containers:
         - name: container_name
           image: image_name


   replicas: num_of_desired_pods
   selector:
     matchLabels:
       name: webapp
~
~
```

know how to create deployment on declarative way ?

```
$ kubectl create -f  file_name
```