Deployment

A Deployment is a higher-level Kubernetes resource that manages ReplicaSets and provides declarative updates for Pods.

- It automates scaling, rolling updates, and rollbacks of your applications.
- It ensures your app runs in the desired state, with zero downtime during updates.
- Under the hood, a Deployment creates and manages ReplicaSets to maintain the specified number of Pods.

When we create a deployment

- Deployment created (with your desired app specs).
- The Deployment creates a ReplicaSet behind the scenes.
- The ReplicaSet creates and manages the Pods based on the Pod template in the Deployment.

Why Use a Deployment?

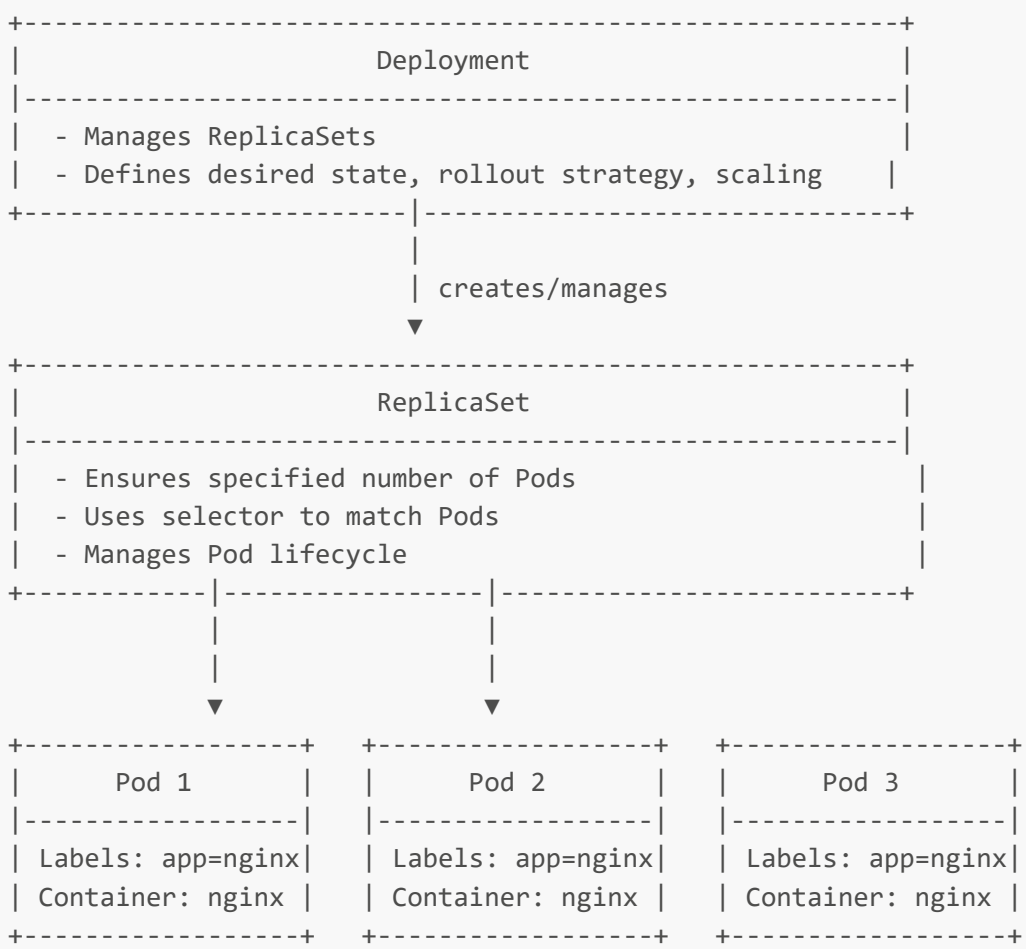
- Define desired state, Deployment handles changes
- Gradually updates Pods without downtime
- Easily revert to previous app versions
- Scale Pods up/down easily
- Restarts or replaces failed Pods
- Automatically creates & manages ReplicaSets

Deployment YAML example

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3                          # Desired number of Pods
  selector:
    matchLabels:
      app: nginx                       # Select Pods with this label
  template:
    metadata:
      labels:
        app: nginx                     # Labels assigned to Pods
    spec:
      containers:
        - name: nginx
          image: nginx:latest
          ports:
            - containerPort: 80
  strategy:
    type: RollingUpdate                 # Rolling update strategy
    rollingUpdate:
```

```
      maxUnavailable: 1                  # Max Pods unavailable during update
      maxSurge: 1                        # Max extra Pods created during update
```

Block diagram



kubectl deployment commands with --namespace

Command	Purpose
`kubectl get deployments -n <namespace>`	List Deployments in a specific namespace
`kubectl describe deployment <name> -n <namespace>`	Describe a Deployment in a specific namespace
`kubectl apply -f deployment.yaml -n <namespace>`	Create or update a Deployment in a namespace
`kubectl delete deployment <name> -n <namespace>`	Delete a Deployment from a namespace
`kubectl get pods -n <namespace>`	See Pods managed by the Deployment

Command	Purpose
`kubectl get rs -n <namespace>`	See ReplicaSets created by Deployment
`kubectl rollout status deployment/<name> -n <namespace>`	Check rollout progress in namespace
`kubectl rollout undo deployment/<name> -n <namespace>`	Rollback Deployment in a namespace
`kubectl scale deployment <name> --replicas=5 -n <namespace>`	Scale Deployment in a namespace
`kubectl set image deployment/<name> <container>=<image> -n <namespace>`	Update container image in namespace