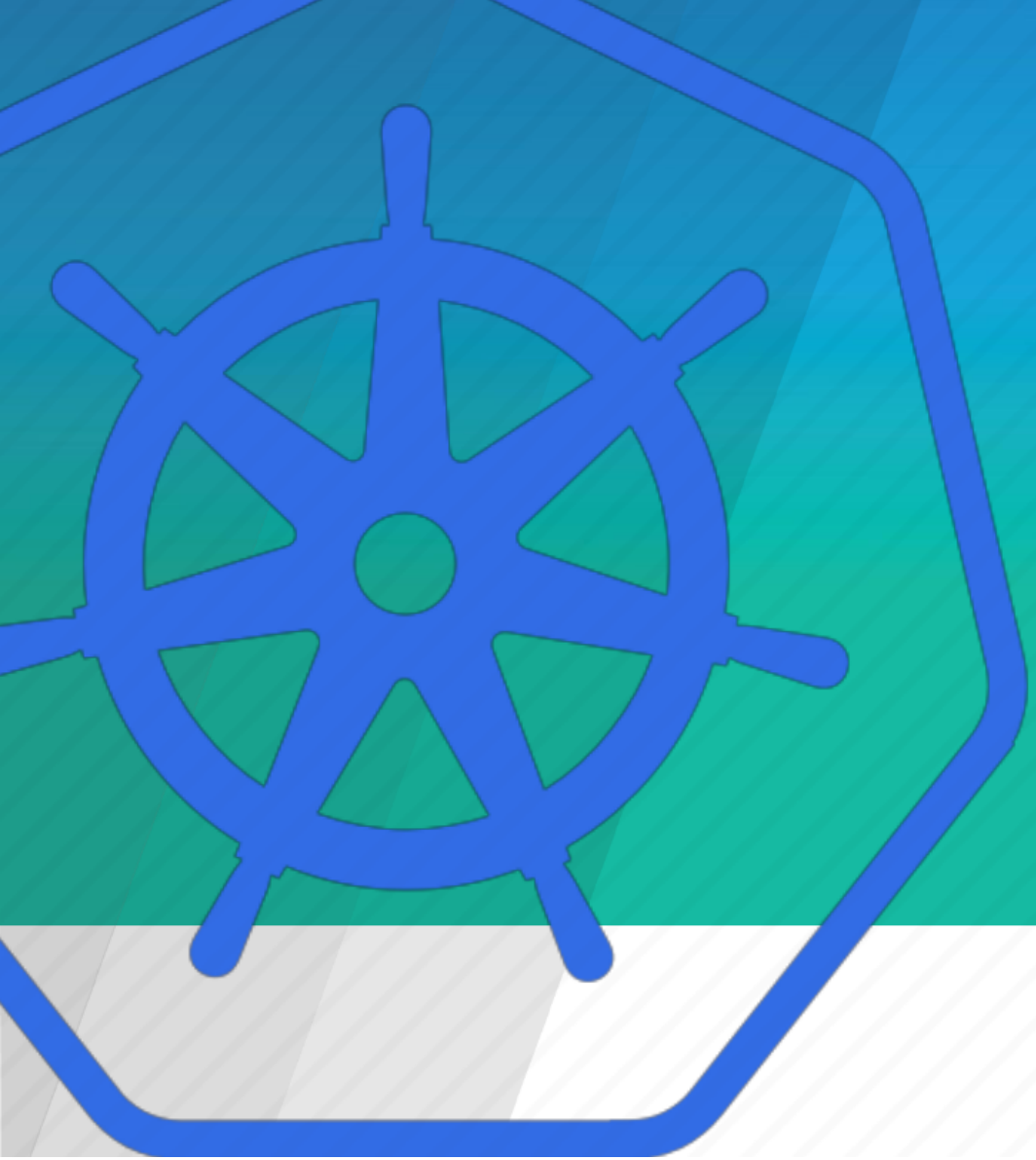


Demystifying the Nuts & Bolts of Kubernetes Architecture

Deployment101

docker

Deployment

Scenario: You deployed an app few months ago. Now you want to upgrade your app from v1 to v2.

Can you upgrade with Zero downtime?
Can you upgrade sequentially one after another?
Can you pause and resume upgrade process?
Rollback upgrade to previous stable release

Agenda

- Deployment Overview
- Features
- Types of Deployment
- Demo
  - Manifest File
  - Deploy Application with Replication Controller
  - Display & Validate
  - Test Use cases
  - Cleaning Up

Deployment

A Deployment controller provides declarative updates for Pods and ReplicaSets.

You describe a desired state in a Deployment, and the Deployment controller changes the actual state to the desired state at a controlled rate. You can define Deployments to create new ReplicaSets, or to remove existing Deployments and adopt all their resources with new Deployments.

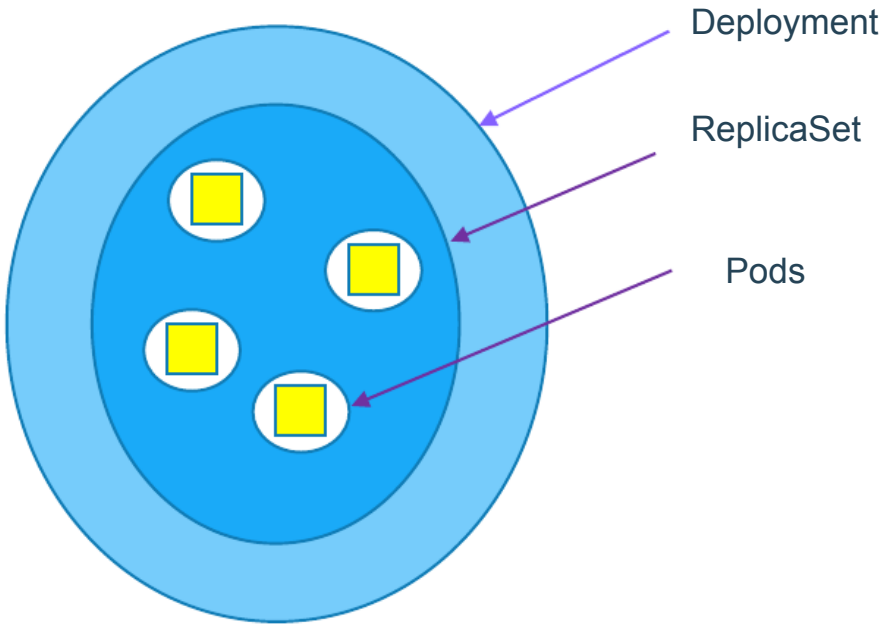
How is it different from Replicaset?

ReplicaSet doesn’t provide features like updates & roll backs.

A Single Deployment Manifest File

Do we need to create 3 different manifest files for each on these?

Answer is “No”. We can manage all 3 different objects(Pods, ReplicaSet & Deployment) using a single Deployment manifest file



docker

Features of Deployment

- Multiple Replicas
- Upgrade
- Rollback
- Scale Up or Down
- Pause & Resume

Deployment Types - Recreate

- Recreate

How it works?

Shutting down version A and then making sure, version A is turned off...
then bringing up version B.

Demerits:

During this, there will be a downtime of the service.

Easy to setup.

Deployment Type – Rolling Updates

- RollingUpdate(Ramped or Incremental)

- Default updating strategy in Kubernetes.
- It can take sometime for a complete update process

How it works?

Slowly rollout a version of app by replacing instances one after the other until all the instances are successfully rolled out.
Assume that there are 10 instances of version A which is running behind the LB. Then update strategy starts with one instance of version B is deployed When version B is ready to accept traffic, one instance of version A is removed from the pool

Deployment Type - Canary

- Canary

- Ideal deployment method for someone who want to test newer version before it is deployed 100%.

How it works?

This method is all about gradually shifting production traffic from version A to version B.

Lets imagine that there are about 10 instances of app version A running inside a cluster. You use Canary deployment when you don’t want to upgrade all of your instances. Let's say you upgraded your 2 instances of version A to version B then do some testing. If test results are good, then you upgrade remaining 8 instances to version B. Say, your version B is ready, then you completely shut down version A.

Deployment Type – Blue Green

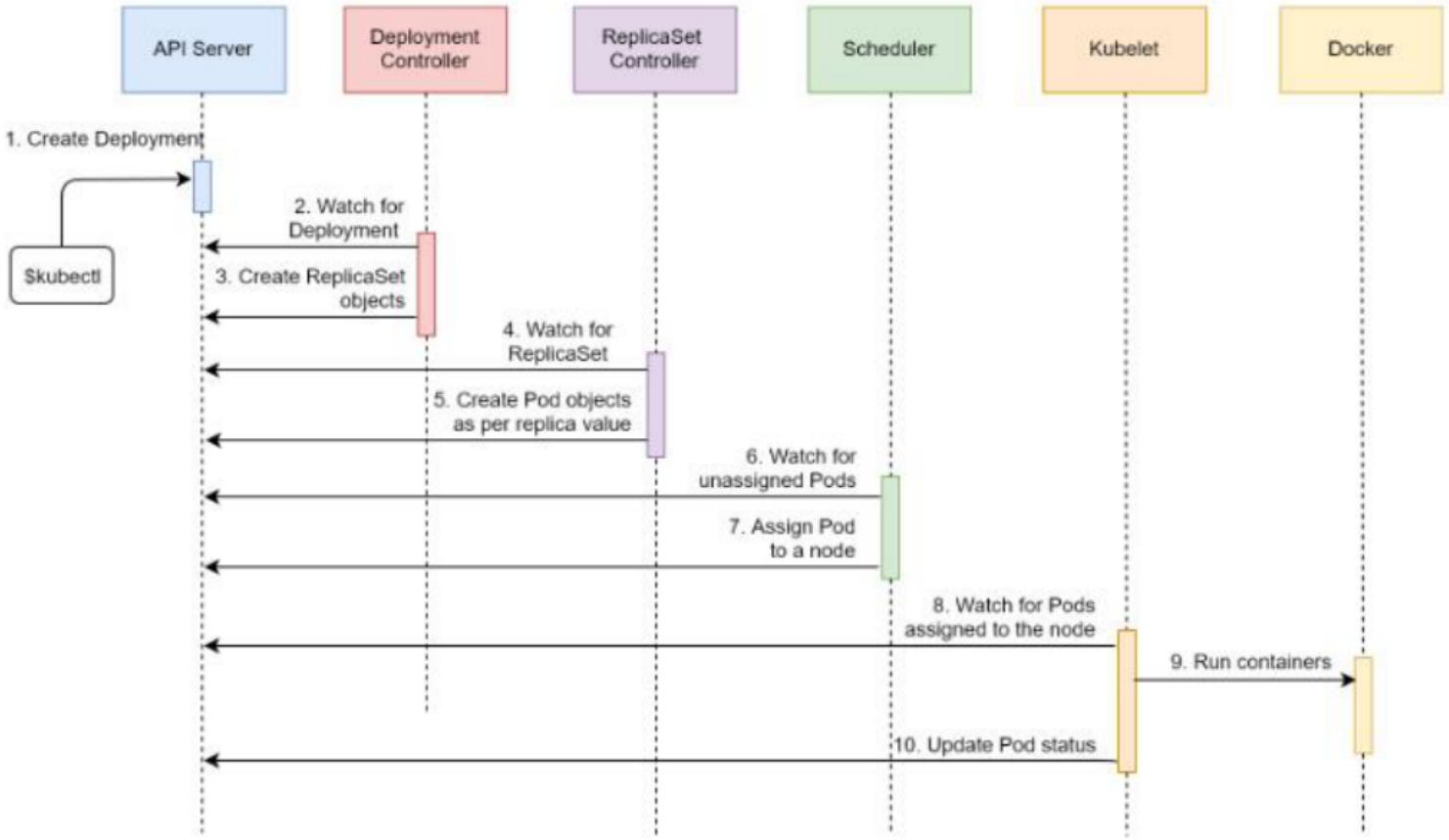
- Blue Green

- Instance roll out and roll back.

How it works?

Using this method, version B(which is GREEN) is deployed along side version A(which is BLUE) with exactly same amount of instances.
After testing new version with all the requirement, the traffic is switched from version A to version B at the LB level.

A Typical Deployment Workflow



Credits" Viktor Farcic

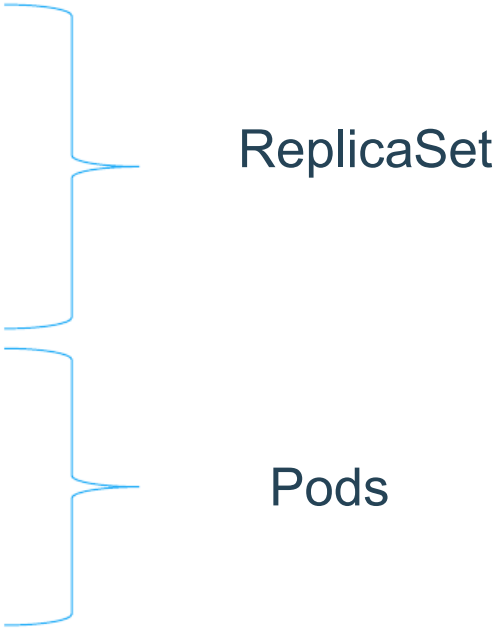
docker

Demo - Deployment

- Manifest file
- Deploy app using RS
- Display and validate RS
- Test – Node Fails
- Test – Scale Up
- Test – Scale Down

Deployment Manifest File

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



docker

Deployment

```
[node1 lab03-creating-deployment-3replicas-nginx]$ ls
README.md  nginx-deploy.yaml
[node1 lab03-creating-deployment-3replicas-nginx]$ kubectl create -f nginx-deploy.yaml
deployment.apps/nginx-deploy created
[node1 lab03-creating-deployment-3replicas-nginx]$ kubectl get deploy
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deploy   0/3     3            0           6s
```

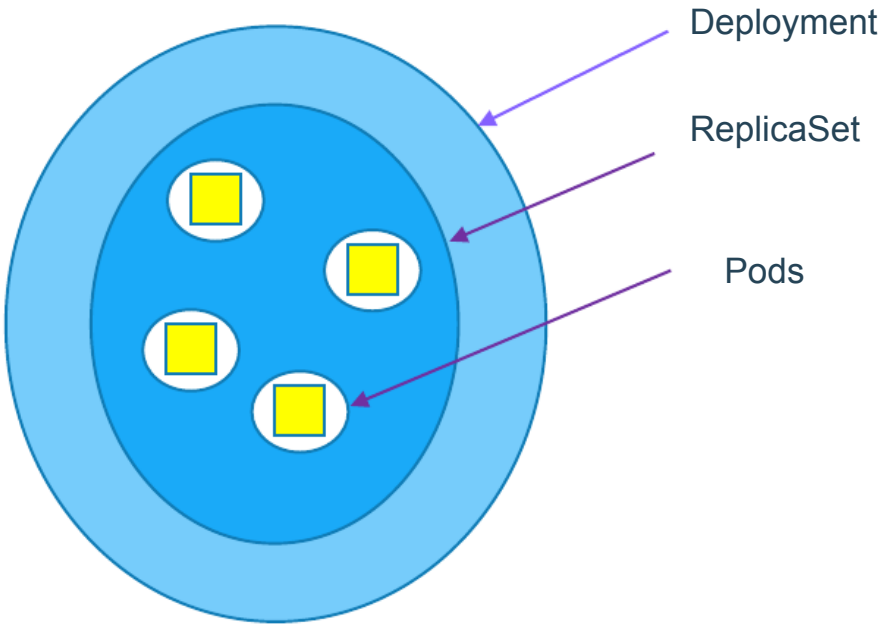
```
[node1 lab03-creating-deployment-3replicas-nginx]$ kubectl get deploy -o wide
NAME           READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES   SELECTOR
nginx-deploy   0/3     3            0           16s   nginx        nginx    app=nginx-app
[node1 lab03-creating-deployment-3replicas-nginx]$ kubectl get deploy -o wide
NAME           READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES   SELECTOR
nginx-deploy   3/3     3            3           57s   nginx        nginx    app=nginx-app
```

Deployment => Pods + ReplicaSet

```
[node1 lab03-creating-deployment-3replicas-nginx]$ kubectl get po,rs,deploy
NAME                                 READY   STATUS    RESTARTS   AGE
pod/nginx-deploy-c9d474fc-lhz9p      1/1     Running   0          2m25s
pod/nginx-deploy-c9d474fc-v8xwg      1/1     Running   0          2m25s
pod/nginx-deploy-c9d474fc-vx4cm      1/1     Running   0          2m25s

NAME                                       DESIRED   CURRENT   READY   AGE
replicaset.extensions/nginx-deploy-c9d474fc   3         3         3       2m25s

NAME                                  READY   UP-TO-DATE   AVAILABLE   AGE
deployment.extensions/nginx-deploy    3/3     3            3           2m25s
```



docker

3 Instances of same Nginx Apps running in the form of Pods

```
[node1 lab03-creating-deployment-3replicas-nginx]$ kubectl get po,rs,deploy -o wide
NAME                                READY   STATUS    RESTARTS   AGE     IP          NODE    NOMINATED NODE   RE
ADINESS GATES
pod/nginx-deploy-c9d474fc-lhz9p     1/1     Running   0          4m21s   10.47.0.1   node3   <none>           <n
one>
pod/nginx-deploy-c9d474fc-v8xwg     1/1     Running   0          4m21s   10.44.0.1   node2   <none>           <n
one>
pod/nginx-deploy-c9d474fc-vx4cm     1/1     Running   0          4m21s   10.36.0.1   node5   <none>           <n
one>

NAME                                         DESIRED   CURRENT   READY   AGE     CONTAINERS   IMAGES   SELECT
OR
replicaset.extensions/nginx-deploy-c9d474fc   3         3         3       4m21s   nginx        nginx    app=ng
inx-app,pod-template-hash=c9d474fc
```

```
[node1 lab03-creating-deployment-3replicas-nginx]$ kubectl get deploy -l app=nginx-app
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deploy   3/3     3            3           7m46s
[node1 lab03-creating-deployment-3replicas-nginx]$
```

3 Instances of same Nginx Apps running in the form of Pods

```
[node1 lab03-creating-deployment-3replicas-nginx]$ kubectl get rs -l app=nginx-app
NAME                      DESIRED   CURRENT   READY   AGE
nginx-deploy-c9d474fc     3         3         3       8m33s
```

Update Deployment

```
[node1 lab03-creating-deployment-3replicas-nginx]$
[node1 lab03-creating-deployment-3replicas-nginx]$ kubectl set image deploy nginx-deploy nginx=nginx:1.9.1
deployment.extensions/nginx-deploy image updated
```

```
CreationTimestamp:      Sat, 13 Jul 2019 18:50:48 +0000
Labels:                 app=nginx-app
Annotations:            deployment.kubernetes.io/revision: 2
Selector:               app=nginx-app
Replicas:               3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx-app
  Containers:
   nginx:
    Image:        nginx:1.9.1
    Port:         80/TCP
    Host Port:    0/TCP
```

3 Instances of same Nginx Apps running in the form of Pods

```
CreationTimestamp:      Sat, 13 Jul 2019 18:50:48 +0000
Labels:                 app=nginx-app
Annotations:            deployment.kubernetes.io/revision: 2
Selector:               app=nginx-app
Replicas:               3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx-app
  Containers:
   nginx:
    Image:        nginx:1.9.1
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
```

```
[node1 lab03-creating-deployment-3replicas-nginx]$ kubectl rollout status deployment/nginx-deploy
deployment "nginx-deploy" successfully rolled out
[node1 lab03-creating-deployment-3replicas-nginx]$
```

Scaling up

```
[node1 lab03-creating-deployment-3replicas-nginx]$ kubectl scale deployment nginx-deploy --replicas=6
deployment.extensions/nginx-deploy scaled
[node1 lab03-creating-deployment-3replicas-nginx]$ kubectl get deploy
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deploy   5/6     6            5           22m
[node1 lab03-creating-deployment-3replicas-nginx]$
```

```
[node1 lab03-creating-deployment-3replicas-nginx]$ kubectl get po
NAME                            READY   STATUS    RESTARTS   AGE
nginx-deploy-5985c6547d-g8nf4   1/1     Running   0          7m38s
nginx-deploy-5985c6547d-jmfc5   1/1     Running   0          8m16s
nginx-deploy-5985c6547d-jnzhh   1/1     Running   0          96s
nginx-deploy-5985c6547d-nbfd8   1/1     Running   0          96s
nginx-deploy-5985c6547d-qr8r6   1/1     Running   0          96s
nginx-deploy-5985c6547d-rvkn6   1/1     Running   0          8m54s
[node1 lab03-creating-deployment-3replicas-nginx]$
```

Listing Pods by Labels

```
[node1 lab03-creating-deployment-3replicas-nginx]$ kubectl get po -l app=nginx-app
NAME                            READY   STATUS    RESTARTS   AGE
nginx-deploy-5985c6547d-g8nf4   1/1     Running   0          8m25s
nginx-deploy-5985c6547d-jmfc5   1/1     Running   0          9m3s
nginx-deploy-5985c6547d-jnzhh   1/1     Running   0          2m23s
nginx-deploy-5985c6547d-nbfd8   1/1     Running   0          2m23s
nginx-deploy-5985c6547d-qr8r6   1/1     Running   0          2m23s
nginx-deploy-5985c6547d-rvkn6   1/1     Running   0          9m41s
[node1 lab03-creating-deployment-3replicas-nginx]$
[node1 lab03-creating-deployment-3replicas-nginx]$
[node1 lab03-creating-deployment-3replicas-nginx]$
```

Thank You