DEPLOYMENTS

About deployment:-

Deployment is very important and being used in production. the hierarchy is deployment <-- replicatset <-- pods

basically, developers are creating new code every day, so latest version will be deployed for testing on top or upgrade of existing pod can be done by using deployment only. here replicaset will not help, so without downtime if you want to upgrade pod, then deployment is the option will be used for update or upgrade.in production single pod can be created but it's doesn't have high availability when it got crash, so here replicaset will help to bring the pod in high availability, it's called as self-healing method, so when it comes to production deployment will be in top of replicaset which will help to upgrade version or roll back according to requirement.

Actual readme:-

A Deployment controller provides declarative updates for Pods and ReplicaSets.

You describe a desired state in a Deployment object, 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.

Use Case

The following are typical use cases for Deployments:

1) Create a Deployment to rollout a ReplicaSet. The ReplicaSet creates Pods in the background. Check the status of the rollout to see if it succeeds or not.

2) Declare the new state of the Pods by updating the PodTemplateSpec of the Deployment. A new ReplicaSet is created and the Deployment manages moving the Pods from the old ReplicaSet to the new one at a controlled rate. Each new ReplicaSet updates the revision of the Deployment.

3) Rollback to an earlier Deployment revision if the current state of the Deployment is not stable. Each rollback updates the revision of the Deployment

##Create a deploy manifest yml##

#kubectl explain deploy |egrep "KIND|VERSION"

```
[root@ansikube mainfest]# kubectl explain deploy |egrep "KIND|VERSION"
KIND:     Deployment
VERSION:  extensions/v1beta1
[root@ansikube mainfest]#
```

#cat deploy1.yml

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9
        ports:
        - containerPort: 80
```

#kubectl get deploy
kubectl apply -f deploy1.yml –record
kubectl get deploy

```
[root@ansikube manifest]# kubectl get deploy
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           27s
[root@ansikube manifest]#
```

kubectl get replicaset

```
[root@ansikube manifest]# kubectl get replicaset
NAME                          DESIRED   CURRENT   READY   AGE
nginx-deployment-76bf4969df   3         3         3       2m15s
[root@ansikube manifest]#
```

kubectl describe deploy nginx-deployment

```
[root@ansikube mainfest]# kubectl describe deploy nginx-deployment
Name:                   nginx-deployment
Namespace:              default
CreationTimestamp:      Fri, 01 Nov 2019 05:31:46 +0000
Labels:                 app=nginx
Annotations:            deployment.kubernetes.io/revision: 1
                        kubectl.kubernetes.io/last-applied-configuration:
                          {"apiVersion":"extensions/v1beta1","kind":"Deployment","met
deployment","na...
Selector:               app=nginx
Replicas:               3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  1 max unavailable, 1 max surge
OldReplicaSets:  <none>
NewReplicaSet:   nginx-deployment-76bf4969df (3/3 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  2m49s  deployment-controller  Scaled up replica set nginx-deployment-76bf4969df to 3
```

kubectl get pods

```
[root@ansikube manifest]# kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
nginx-deployment-76bf4969df-m9qvv   1/1     Running   0          3m54s
nginx-deployment-76bf4969df-mw6fd   1/1     Running   0          3m54s
nginx-deployment-76bf4969df-zrt78   1/1     Running   0          3m54s
[root@ansikube manifest]#
```

kubectl exec nginx-deployment-76bf4969df-m9qvv -- nginx -v

```
[root@ansikube manifest]# kubectl exec nginx-deployment-76bf4969df-m9qvv -- nginx -v
nginx version: nginx/1.7.9
[root@ansikube manifest]#
```

@@Change the app version and deploy for testing. make sure to specify --record, which will helpful to roll back the version.

#cat deploy1.yml

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.17
        ports:
        - containerPort: 80
```

kubectl apply -f deploy1.yml –record
kubectl get deploy -o wide

```
[root@ansikube manifest]# kubectl get deploy -o wide
NAME               READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES       SELECTOR
nginx-deployment   3/3     3            3           12m   nginx        nginx:1.17   app=nginx
[root@ansikube manifest]#
```

kubectl get pods

```
[root@ansikube manifest]# kubectl get pods
NAME                                READY   STATUS              RESTARTS   AGE
nginx-deployment-76bf4969df-m9qvv   1/1     Running             0          9m3s
nginx-deployment-76bf4969df-mw6fd   1/1     Running             0          9m3s
nginx-deployment-8c5ddb5c-c75gn     0/1     ContainerCreating   0          5s
nginx-deployment-8c5ddb5c-n4mr8     0/1     ContainerCreating   0          5s
[root@ansikube manifest]# kubectl get pods
NAME                                READY   STATUS              RESTARTS   AGE
nginx-deployment-76bf4969df-mw6fd   0/1     Terminating         0          9m9s
nginx-deployment-8c5ddb5c-5d8vh     0/1     ContainerCreating   0          3s
nginx-deployment-8c5ddb5c-c75gn     1/1     Running             0          11s
nginx-deployment-8c5ddb5c-n4mr8     1/1     Running             0          11s
[root@ansikube manifest]# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-8c5ddb5c-5d8vh     1/1     Running   0          11s
nginx-deployment-8c5ddb5c-c75gn     1/1     Running   0          19s
nginx-deployment-8c5ddb5c-n4mr8     1/1     Running   0          19s
```

kubectl exec nginx-deployment-8c5ddb5c-5d8vh -- nginx -v

```
[root@ansikube manifest]# kubectl exec nginx-deployment-8c5ddb5c-5d8vh -- nginx -v
nginx version: nginx/1.17.5
[root@ansikube manifest]#
```

kubectl describe deploy nginx-deployment

```
Events:
  Type    Reason             Age    From                   Message
  ----    ------             ----   ----                   -------
  Normal  ScalingReplicaSet  14m    deployment-controller  Scaled up replica set nginx-deployment-76bf4969df to 3
  Normal  ScalingReplicaSet  5m58s  deployment-controller  Scaled up replica set nginx-deployment-8c5ddb5c to 1
  Normal  ScalingReplicaSet  5m58s  deployment-controller  Scaled down replica set nginx-deployment-76bf4969df to 2
  Normal  ScalingReplicaSet  5m58s  deployment-controller  Scaled up replica set nginx-deployment-8c5ddb5c to 2
  Normal  ScalingReplicaSet  5m50s  deployment-controller  Scaled down replica set nginx-deployment-76bf4969df to 1
  Normal  ScalingReplicaSet  5m50s  deployment-controller  Scaled up replica set nginx-deployment-8c5ddb5c to 3
  Normal  ScalingReplicaSet  5m50s  deployment-controller  Scaled down replica set nginx-deployment-76bf4969df to 0
```

Note:- Now nginx version has been updated on pods, by one after another.

@@ Below command will show you the revision of history, which will help for roll back.

kubectl rollout history deployment nginx-deployment

```
[root@ansikube manifest]# kubectl rollout history deployment nginx-deployment
deployment.extensions/nginx-deployment
REVISION  CHANGE-CAUSE
1         kubectl apply --filename=deploy1.yml --record=true
2         kubectl apply --filename=deploy1.yml --record=true

[root@ansikube manifest]#
```

@Let’s roll back to the older version of nginx.

kubectl rollout undo deploy nginx-deployment --to-revision=1
kubectl get deploy -o wide

```
[root@ansikube manifest]# kubectl get deploy -o wide
NAME               READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES        SELECTOR
nginx-deployment   3/3     3            3           24m   nginx        nginx:1.7.9   app=nginx
[root@ansikube manifest]#
```

kubectl describe deploy nginx-deployment

```
Events:
  Type    Reason             Age                From                   Message
  ----    ------             ----               ----                   -------
  Normal  ScalingReplicaSet  17m                deployment-controller  Scaled up replica set nginx-deployment-8c5ddb5c to 1
  Normal  ScalingReplicaSet  17m                deployment-controller  Scaled down replica set nginx-deployment-76bf4969df to 2
  Normal  ScalingReplicaSet  17m                deployment-controller  Scaled up replica set nginx-deployment-8c5ddb5c to 2
  Normal  ScalingReplicaSet  17m                deployment-controller  Scaled down replica set nginx-deployment-76bf4969df to 0
  Normal  ScalingReplicaSet  17m                deployment-controller  Scaled down replica set nginx-deployment-76bf4969df to 1
  Normal  ScalingReplicaSet  17m                deployment-controller  Scaled up replica set nginx-deployment-8c5ddb5c to 3
  Normal  ScalingReplicaSet  106s               deployment-controller  Scaled up replica set nginx-deployment-76bf4969df to 1
  Normal  ScalingReplicaSet  106s               deployment-controller  Scaled down replica set nginx-deployment-8c5ddb5c to 2
  Normal  ScalingReplicaSet  106s               deployment-controller  Scaled up replica set nginx-deployment-76bf4969df to 2
  Normal  ScalingReplicaSet  105s (x2 over 26m)  deployment-controller  Scaled up replica set nginx-deployment-76bf4969df to 3
  Normal  ScalingReplicaSet  105s               deployment-controller  Scaled down replica set nginx-deployment-8c5ddb5c to 1
  Normal  ScalingReplicaSet  105s               deployment-controller  Scaled down replica set nginx-deployment-8c5ddb5c to 0
```

kubectl get pods
kubectl exec nginx-deployment-76bf4969df-qxwjr -- nginx -v

```
[root@ansikube manifest]# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-76bf4969df-qxwjr   1/1     Running   0          3m3s
nginx-deployment-76bf4969df-tnsl2   1/1     Running   0          3m2s
nginx-deployment-76bf4969df-v8rzw   1/1     Running   0          3m3s
[root@ansikube manifest]# kubectl exec nginx-deployment-76bf4969df-qxwjr -- nginx -v
nginx version: nginx/1.7.9
[root@ansikube manifest]#
```

Note:- nginx version from 1.17.5 to 1.7.9 has been roll backed successfully.

kubectl rollout history deployment nginx-deployment

```
[root@ansikube manifest]# kubectl rollout history deployment nginx-deployment
deployment.extensions/nginx-deployment
REVISION  CHANGE-CAUSE
2         kubectl apply --filename=deploy1.yml --record=true
3         kubectl apply --filename=deploy1.yml --record=true
```

Note:- max 2 revision will have in the deployment history.

STRATEGY

strategy specifies the strategy used to replace old Pods by new ones.
"Recreate" or "RollingUpdate". "RollingUpdate" is the default value.

canary update:- update will on specific client or small portion of clients will update and then will be checked whether if its ok then only it will be pushed to other clients.

Blue-green deployment is nothing, but latest version of app will be deployed when the existing version is running, if everything is stable then switchover to latest version and then decommission the older version of app in backend.

Recreate Deployment

All existing Pods are killed before new one gets created when .spec.strategy.type==Recreate.

Rolling Update Deployment

The Deployment updates Pods in a rolling update fashion when .spec.strategy.type==RollingUpdate. You can specify maxUnavailable and maxSurge to control the rolling update process

OPTIONS

Max Unavailable

maxUnavailable is an optional field that specifies the maximum number of Pods that can be unavailable during the update process. The value can be an absolute number (for example, 5) or a percentage of desired Pods (for example, 10%). The absolute number is calculated from percentage by rounding down. The value cannot be 0 if .spec.strategy.rollingUpdate.maxSurge is 0. The default value is 25%.
For example, when this value is set to 30%, the old ReplicaSet can be scaled down to 70% of desired Pods immediately when the rolling update starts. Once new Pods are ready, old ReplicaSet can be scaled down further, followed by scaling up the new ReplicaSet, ensuring that the total number of Pods available all the times during the update is at least 70% of the desired Pods.

Max Surge

.spec.strategy.rollingUpdate.maxSurge is an optional field that specifies the maximum number of Pods that can be created over the desired number of Pods. The value can be an absolute number (for example, 5) or a percentage of desired Pods (for example, 10%). The value cannot be 0 if MaxUnavailable is 0. The absolute number is calculated from the percentage by rounding up. The default value is 25%.
For example, when this value is set to 30%, the new ReplicaSet can be scaled up immediately when the rolling update starts, such that the total number of old and new Pods does not exceed 130% of desired Pods. Once old Pods have been killed, the new ReplicaSet can be scaled up further, ensuring that the total number of Pods running at any time during the update is at most 130% of desired Pods

@@Create a deploy with strategy :- recreate

cat deploy3.yml

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.15
        ports:
        - containerPort: 80
```

kubectl apply -f deploy3.yml –record
kubectl get deploy -o wide

```
[root@ansikube manifest]# kubectl get deploy -o wide
NAME               READY   UP-TO-DATE   AVAILABLE   AGE    CONTAINERS   IMAGES         SELECTOR
nginx-deployment   3/3     3            3           135m   nginx        nginx:1.15   app=nginx
[root@ansikube manifest]#
```

kubectl get pods
kubectl exec nginx-deployment-5fc86c987f-hh6cv -- nginx -v

```
[root@ansikube manifest]# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-5fc86c987f-hh6cv   1/1     Running   0          10m
nginx-deployment-5fc86c987f-rbv4g   1/1     Running   0          10m
nginx-deployment-5fc86c987f-tm6bd   1/1     Running   0          10m
[root@ansikube manifest]# kubectl exec nginx-deployment-5fc86c987f-hh6cv -- nginx -v
nginx version: nginx/1.15.12
[root@ansikube manifest]#
```

kubectl describe deploy nginx-deployment

```
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        nginx:1.15
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
OldReplicaSets:  <none>
NewReplicaSet:   nginx-deployment-5fc86c987f (3/3 replicas created)
Events:
  Type    Reason             Age                From                   Message
  ----    ------             ----               ----                   -------
  Normal  ScalingReplicaSet  14m                deployment-controller  Scaled down replica set nginx-deployment-5fc86c987f to 0
  Normal  ScalingReplicaSet  14m (x3 over 146m)  deployment-controller  Scaled up replica set nginx-deployment-76bf4969df to 3
  Normal  ScalingReplicaSet  14m (x3 over 137m)  deployment-controller  Scaled down replica set nginx-deployment-76bf4969df to 0
  Normal  ScalingReplicaSet  13m (x2 over 137m)  deployment-controller  Scaled up replica set nginx-deployment-8c5ddb5c to 3
  Normal  ScalingReplicaSet  12m (x2 over 122m)  deployment-controller  Scaled down replica set nginx-deployment-8c5ddb5c to 0
  Normal  ScalingReplicaSet  12m (x2 over 15m)   deployment-controller  Scaled up replica set nginx-deployment-5fc86c987f to 3
```

kubectl rollout history deployment nginx-deployment

```
[root@ansikube manifest]# kubectl rollout history deployment nginx-deployment
deployment.extensions/nginx-deployment
REVISION  CHANGE-CAUSE
5         kubectl apply --filename=deploy1.yml --record=true
6         kubectl apply --filename=deploy1.yml --record=true
7         kubectl apply --filename=deploy3.yml --record=true
```

#kubectl rollout undo deploy nginx-deployment --to-revision=5

#kubectl exec nginx-deployment-76bf4969df-n4k9l -- nginx -v

```
[root@ansikube manifest]# kubectl get deploy -o wide
NAME               READY   UP-TO-DATE   AVAILABLE   AGE    CONTAINERS   IMAGES         SELECTOR
nginx-deployment   3/3     3            3           157m   nginx        nginx:1.7.9    app=nginx
[root@ansikube manifest]#
```

Note:- All existing Pods are killed before the new pods get created when .spec.strategy.type==Recreate.