Docker, Kubernetes, VMware Tanzu and RedHat OCP
13Oct2022
Day07

Kubernetes Workloads II

Workloads

https://kubernetes.io/docs/concepts/workloads/
A workload is an application running on Kubernetes.
1-Pod
2-replicaset
3-deployment

ex
create **redispod** on **cust1** namespace
namespce
kubectl get namespace
kubectl create namespace cust1
or
kubectl create namespace cust1 -o yaml --dry-run=client >cust1.yaml
kubectl create namespace cust1 -o yaml --dry-run=client
kubectl create namespace cust1
Pod
kubectl run redispod --image redis --namespace cust1
kubectl run redispod --image redis --namespace cust1 -o yaml –dry-run=client
kubectl run redispod --image redis --namespace cust1 -o yaml –dry-run=client >redispod.yaml

ReplicaSet

https://kubernetes.io/docs/concepts/workloads/controllers/replicaset/
A ReplicaSet's purpose is to maintain a stable set of replica Pods running at any given time.
kubectl get pods -A

Labels and Selectors

https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/
Labels are key/value pairs that are attached to objects, such as pods. Labels are intended to be used to specify identifying attributes of objects that are meaningful and relevant to users, but do not directly imply semantics to the core system.
labels
key1:value1

Create
```
# vim  redisreplica.yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: redisreplica
  labels:
    app: redisrepl
spec:
  replicas: 3
  selector:
    matchLabels:
      app: redis
  template:
    metadata:
      labels:
        app: redis
    spec:
      containers:
      - name: rediscnt
        image: redis
:wq!
# kubectl apply -f redisreplica.yaml
# kubectl get replicasets.apps
# kubectl get rs
NAME           DESIRED   CURRENT   READY   AGE
redisreplica   3         3         3       5m21s

# kubectl get pods -o wide

# kubectl delete pod redisreplica-hhg8s --force --grace-period=0

# kubectl get rs
NAME           DESIRED   CURRENT   READY   AGE
redisreplica   3         3         2       6m56s

# kubectl get rs
NAME           DESIRED   CURRENT   READY   AGE
redisreplica   3         3         3       7m6s

# kubectl delete rs  redisreplica --force --grace-period=0
```

Mohammad Ali Naghval
ali@prodevans.com

Deployments

https://kubernetes.io/docs/concepts/workloads/controllers/deployment/

A Deployment provides declarative updates for Pods and ReplicaSets.

describe a desired state in a Deployment, and the Deployment Controller changes the actual state to the desired state at a controlled rate.

Create

```
# kubectl api-resources | grep -i "deployment"
deployments                 deploy      apps/v1                        true        Deployment
# kubectl create deployment nginxdpl --image nginx --replicas 3
# kubectl create deployment nginxdpl --image nginx --replicas 3 -o yaml --dry-run=client >nginxdpl.yaml
# cat nginxdpl.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: nginxdpl                       ->deployment's label
  name: nginxdpl
spec:
  replicas: 3
  selector:
    matchLabels:
      app: pd
  template:
    metadata:
      labels:
        app: pd
    spec:
      containers:
      - image: nginx
        name: nginxcnt
# kubectl create -f nginxdpl.yaml --dry-run=client
# kubectl create -f nginxdpl.yaml
# watch kubectl get deployments.apps
or
# kubectl get deployments.apps -w
# kubectl get deployments.apps
```

create deployment in other namespace

```
# kubectl create -f nginxdpl.yaml --namespace cust1
# vim nginxdpl.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: nginxdpl
  name: nginxdpl
  namespace: cust1
spec:
:wq!
```

Deployment Operations

get

```
# kubectl get deploy
# kubectl get deploy -o wide
# kubectl get pods
# kubectl get pods -o wide
```

describe

-pod

```
# kubectl describe pod nginxdpl-64c8ff5d79-vmpt9
```

-deployment

```
# kubectl describe deployments.apps nginxdpl
```

modify deployment parameters

```
# kubectl describe pod nginxdpl-64c8ff5d79-c99k4 | grep -i "image"                    ->pod detail
    Image:          nginx
# kubectl get pods                                                                   ->number of pods
# kubectl get deployments.apps nginxdpl -o yaml | grep -i "replica"
  replicas: 3
```

-number of replicas

```
# kubectl scale deployment nginxdpl --replicas 6
# kubectl get pods
# kubectl scale deployment nginxdpl --replicas 1
# kubectl get pods
or
# kubectl edit deployments.apps nginxdpl
spec:
  progressDeadlineSeconds: 600
  replicas: 4
:wq!
# kubectl get pods
```

Mohammad Ali Naghval
ali@prodevans.com

-change image version

```
# kubectl describe pod nginxdpl-64c8ff5d79-c7wcz | grep -i "image"
   Image:         nginx
```

Cli

```
# kubectl set image deployment nginxdpl nginxcnt=nginx:1.19
```

or

```
# kubectl edit deployments.apps nginxdpl
   spec:
     containers:
     - image: nginx:1.18
:wq!
```

or

```
# kubectl get deployments.apps nginxdpl -o yaml >abc.yaml
# vim abc.yaml
   spec:
     containers:
     - image: nginx:1.17
:wq!
# kubectl replace -f abc.txt
```

NOTE: this deployment generated by 'create' command then should update through 'replace' command.

Backup Deployment

```
# kubectl get deployments.apps nginxdpl -o yaml >nginxdpll.yaml
```

Delete Deployment

```
# kubectl delete deployments.apps nginxdpl --force --grace-period=0
```

Restore Deployment

```
# kubectl create -f nginxdpll.yaml
```

deployment Rollout

rollout gives us more flexibility on daemonset, deployment, statefulset

```
# kubectl rollout
history  pause   restart  resume   status   undo

# kubectl rollout history deployment nginxdpl
deployment.apps/nginxdpl
REVISION  CHANGE-CAUSE
1         <none>
2         <none>
3         <none>
4         <none>
```

```
# kubectl rollout history deployment nginxdpl --revision 1
# kubectl rollout history deployment nginxdpl --revision 2
# kubectl set image deployment nginxdpl nginxcnt=nginx2.1 --record
kubectl rollout history deployment nginxdpl
deployment.apps/nginxdpl
REVISION  CHANGE-CAUSE
1         <none>
2         <none>
3         <none>
4         <none>
5         <none>
6         kubectl set image deployment nginxdpl nginxcnt=nginx2.1 --record=true
# kubectl rollout status deployment nginxdpl
```

-restart Deployment

```
# kubectl rollout restart deployment nginxdpl
```

deployment Log

```
# kubectl get deployments.apps
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
nginxdpl  3/4     2            3           18m
# kubectl logs deployments/nginxdpl
```

Mohammad Ali Naghval
ali@prodevans.com