Practical operation Deployment&ReplicaSet

Author: Heike07

> 参考文章：https://www.yuque.com/wukong-zorrm/qdoy5p/yt6rx9

Knowledge

Deployment是对ReplicaSet和Pod更高级的抽象。

它使Pod拥有多副本，自愈，扩缩容、滚动升级等能力。

ReplicaSet(副本集)是一个Pod的集合。

它可以设置运行Pod的数量，确保任何时间都有指定数量的 Pod 副本在运行。

通常我们不直接使用ReplicaSet，而是在Deployment中声明。

Practical Operation

```
# 创建deployment 并设置 replicas为 3
controlplane $ kubectl create deployment nginx-deployment --image=nginx:1.22 --
replicas=3
deployment.apps/nginx-deployment created
controlplane $ kubectl get pod
NAME                                READY   STATUS              RESTARTS   AGE
nginx-deployment-978469b8b-9jlfj    0/1     ContainerCreating   0          8s
nginx-deployment-978469b8b-dcnk2    0/1     ContainerCreating   0          8s
nginx-deployment-978469b8b-l9gxf    0/1     ContainerCreating   0          8s
# 查看 deployment 详细状态
controlplane $ kubectl describe pod/nginx-deployment-978469b8b-9jlfj
Name:             nginx-deployment-978469b8b-9jlfj
Namespace:        default
Priority:         0
Service Account:  default
Node:             node01/172.30.2.2
Start Time:       Fri, 01 Sep 2023 05:22:12 +0000
Labels:           app=nginx-deployment
                  pod-template-hash=978469b8b
Annotations:      cni.projectcalico.org/containerID:
c78a3862ecd194d6ed1e3dcd44fe9cf1e623c0f890660c40987608dd3d7a97e2
                  cni.projectcalico.org/podIP: 192.168.1.3/32
                  cni.projectcalico.org/podIPs: 192.168.1.3/32
Status:           Running
IP:               192.168.1.3
IPs:
  IP:           192.168.1.3
Controlled By:  ReplicaSet/nginx-deployment-978469b8b
Containers:
```

```
  nginx:
    Container ID:
containerd://6eb9b81965895cf5f960b890c9155618dcf4a208b11c6984a52f893ac8aba9ba
    Image:          nginx:1.22
    Image ID:
docker.io/library/nginx@sha256:fc5f5fb7574755c306aaf88456ebfbe0b006420a184d52b92
3d2f0197108f6b7
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Fri, 01 Sep 2023 05:22:21 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-shkrt
(ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  kube-api-access-shkrt:
    Type:                    Projected (a volume that contains injected data
from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists
for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists
for 300s
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  25s   default-scheduler  Successfully assigned
default/nginx-deployment-978469b8b-9jlfj to node01
  Normal  Pulling    24s   kubelet            Pulling image "nginx:1.22"
  Normal  Pulled     16s   kubelet            Successfully pulled image
"nginx:1.22" in 8.147750899s (8.14782024s including waiting)
  Normal  Created    16s   kubelet            Created container nginx
  Normal  Started    16s   kubelet            Started container nginx
# 查看pod list
controlplane $ kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-978469b8b-9jlfj    1/1     Running   0          31s
nginx-deployment-978469b8b-dcnk2    1/1     Running   0          31s
nginx-deployment-978469b8b-l9gxf    1/1     Running   0          31s
# 查看deployment list
controlplane $ kubectl get deployments.apps
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           60s
# 查看replicaset list
```

```
controlplane $ kubectl get rs
NAME                          DESIRED   CURRENT   READY   AGE
nginx-deployment-978469b8b   3         3         3       68s
controlplane $
controlplane $
controlplane $
# 通过手动调整 replicas 5
controlplane $ kubectl scale deployment/nginx-deployment --replicas=5
deployment.apps/nginx-deployment scaled
controlplane $ kubectl get d
daemonsets.apps   deployments.apps
# 查看 deployment list 发现 生效5
controlplane $ kubectl get deployments.apps
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   5/5     5            5           113s
# 也可以deployment 简写成 deploy
controlplane $ kubectl get deploy
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   5/5     5            5           2m10s
# 设置自动调度 根据cpu状态 75% 设置min 和 max 的 rs 数量
controlplane $ kubectl autoscale deployment/nginx-deployment --min 3 --max 10 --
cpu-percent 75
horizontalpodautoscaler.autoscaling/nginx-deployment autoscaled
# 查看deployment list
controlplane $ kubectl get deployments.apps
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   5/5     5            5           3m10s
# 查看内存状态
controlplane $ free -h
               total        used        free      shared  buff/cache   available
Mem:           1.9Gi       939Mi        83Mi       2.0Mi       960Mi       849Mi
Swap:             0B          0B          0B
# 查看自动调度状态 list
controlplane $ kubectl get hpa
NAME               REFERENCE                     TARGETS          MINPODS
MAXPODS   REPLICAS   AGE
nginx-deployment   Deployment/nginx-deployment   <unknown>/75%   3         10
     5          48s
# 删除自动调度 hpa
controlplane $ kubectl delete hpa nginx-deployment
horizontalpodautoscaler.autoscaling "nginx-deployment" deleted
# 查看hpa list
controlplane $ kubectl get hpa
No resources found in default namespace.
controlplane $
controlplane $
controlplane $
# 查看deployment list 的详细信息
controlplane $ kubectl get deployments.apps/nginx-deployment -o wide
NAME               READY   UP-TO-DATE   AVAILABLE   AGE     CONTAINERS   IMAGES
      SELECTOR
nginx-deployment   5/5     5            5           4m42s   nginx
 nginx:1.22   app=nginx-deployment
# 查看pod list
controlplane $ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-978469b8b-5rqwj   1/1     Running   0          3m14s
nginx-deployment-978469b8b-9jlfj   1/1     Running   0          4m58s
```

```
nginx-deployment-978469b8b-9v2mh   1/1     Running   0          3m14s
nginx-deployment-978469b8b-dcnk2   1/1     Running   0          4m58s
nginx-deployment-978469b8b-l9gxf   1/1     Running   0          4m58s
# 将deployment 的 nginx-deployment 手动修改镜像 为 nginx1.23
controlplane $ kubectl set image deployment/nginx-deployment nginx=nginx:1.23
deployment.apps/nginx-deployment image updated
# 查看rollout 状态 是否生效
controlplane $ kubectl rollout status deployment nginx-deployment
Waiting for deployment "nginx-deployment" rollout to finish: 4 of 5 updated
replicas are available...
deployment "nginx-deployment" successfully rolled out
# 查看rs 也就是修改版本
controlplane $ kubectl get rs
NAME                          DESIRED   CURRENT   READY   AGE
nginx-deployment-66c6c86699   5         5         5       31s
nginx-deployment-978469b8b    0         0         0       6m47s
controlplane $
controlplane $
# 查看 rollout 版本list
controlplane $ kubectl rollout history deployment/nginx-deployment
deployment.apps/nginx-deployment
REVISION  CHANGE-CAUSE
1         <none>
2         <none>
# 查看 rollout 版本list 的详细信息 用id 进行筛选
controlplane $ kubectl rollout history deployment/nginx-deployment --revision 2
deployment.apps/nginx-deployment with revision #2
Pod Template:
  Labels:       app=nginx-deployment
        pod-template-hash=66c6c86699
  Containers:
   nginx:
    Image:      nginx:1.23
    Port:       <none>
    Host Port:  <none>
    Environment:        <none>
    Mounts:     <none>
  Volumes:      <none>
# 查看 rollout 版本list 的详细信息 用id 进行筛选
controlplane $ kubectl rollout history deployment/nginx-deployment --revision 1
deployment.apps/nginx-deployment with revision #1
Pod Template:
  Labels:       app=nginx-deployment
        pod-template-hash=978469b8b
  Containers:
   nginx:
    Image:      nginx:1.22
    Port:       <none>
    Host Port:  <none>
    Environment:        <none>
    Mounts:     <none>
  Volumes:      <none>
# 将版本指定到 版本2 也就是 nginx 1.23 回退
controlplane $ kubectl rollout undo deployment/nginx-deployment --to-revision 2
deployment.apps/nginx-deployment skipped rollback (current template already
matches revision 2)
>> - 这个提示是因为当前的版本已经是版本2了 所以没有办法退回
# 将版本回退到 版本1 也就是 nginx 1.22 回退
```

```
controlplane $ kubectl rollout undo deployment/nginx-deployment --to-revision 1
deployment.apps/nginx-deployment rolled back
controlplane $ kubectl get deployments.apps/nginx-deployment -o wide
NAME               READY   UP-TO-DATE   AVAILABLE   AGE     CONTAINERS   IMAGES
      SELECTOR
nginx-deployment   5/5     5            5           8m58s   nginx
 nginx:1.22   app=nginx-deployment
# 查看pod list
controlplane $ kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-978469b8b-64nbf   1/1     Running   0          40s
nginx-deployment-978469b8b-8jzkp   1/1     Running   0          39s
nginx-deployment-978469b8b-dbzhk   1/1     Running   0          40s
nginx-deployment-978469b8b-nwjrq   1/1     Running   0          38s
nginx-deployment-978469b8b-px9pp   1/1     Running   0          40s
# 删除deployment nginx-deployment
controlplane $ kubectl delete deployments.apps/nginx-deployment
deployment.apps "nginx-deployment" deleted
# 查看pod list
controlplane $ kubectl get pod
No resources found in default namespace.
# 查看node list
controlplane $ kubectl get node
NAME           STATUS   ROLES           AGE   VERSION
controlplane   Ready    control-plane   23d   v1.27.1
node01         Ready    <none>          23d   v1.27.1
controlplane $
# END
```