## Deployment概念

用于部署无状态的服务，这个最常用的控制器。一般用于管理维护企业内部无状态的微服务，比如configserver、zuul、springboot。他可以管理多个副本的Pod实现无缝迁移、自动扩容缩容、自动灾难恢复、一键回滚等功能。

## 创建一个Deployment

手动创建：kubectl create deployment nginx --image=nginx:1.15.2

从文件创建：

# cat nginx-deploy.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

annotations:

deployment.kubernetes.io/revision: "1"

creationTimestamp: "2020-09-19T02:41:11Z"

generation: 1

labels:

app: nginx

name: nginx

namespace: default

spec:

progressDeadlineSeconds: 600

replicas: 2 #副本数

revisionHistoryLimit: 10 # 历史记录保留的个数

selector:

matchLabels:

app: nginx

strategy:

rollingUpdate:

maxSurge: 25%

maxUnavailable: 25%

type: RollingUpdate

template:

metadata:

creationTimestamp: null

labels:

app: nginx

spec:

containers:

- image: nginx:1.15.2

imagePullPolicy: IfNotPresent

name: nginx

resources: {}

terminationMessagePath: /dev/termination-log

terminationMessagePolicy: File

dnsPolicy: ClusterFirst

restartPolicy: Always

schedulerName: default-scheduler

securityContext: {}

terminationGracePeriodSeconds: 30

状态解析：

# kubectl get deploy -owide

NAME READY UP-TO-DATE AVAILABLE AGE CONTAINERS IMAGES SELECTOR

nginx 2/2 2 2 9m29s nginx nginx:1.15.2 app=nginx

* NAME： Deployment名称
* READY：Pod的状态，已经Ready的个数
* UP-TO-DATE：已经达到期望状态的被更新的副本数
* AVAILABLE：已经可以用的副本数
* AGE：显示应用程序运行的时间
* CONTAINERS：容器名称
* IMAGES：容器的镜像
* SELECTOR：管理的Pod的标签

## Deployment的更新

更改deployment的镜像并记录：

kubectl set image deploy nginx nginx=nginx:1.15.3 –record

查看更新过程：

# **kubectl rollout status deploy nginx**

deployment "nginx" successfully rolled out

[root@k8s-master01 ~]# kubectl rollout status deploy nginx

Waiting for deployment "nginx" rollout to finish: 1 out of 2 new replicas have been updated...

Waiting for deployment "nginx" rollout to finish: 1 out of 2 new replicas have been updated...

Waiting for deployment "nginx" rollout to finish: 1 out of 2 new replicas have been updated...

Waiting for deployment "nginx" rollout to finish: 1 old replicas are pending termination...

Waiting for deployment "nginx" rollout to finish: 1 old replicas are pending termination...

deployment "nginx" successfully rolled out

或者使用describe查看：

# kubectl describe deploy nginx

Normal ScalingReplicaSet 25m deployment-controller Scaled up replica set nginx-66bbc9fdc5 to 1

Normal ScalingReplicaSet 18m (x2 over 23m) deployment-controller Scaled up replica set nginx-66bbc9fdc5 to 2

Normal ScalingReplicaSet 7m7s deployment-controller Scaled up replica set nginx-5dfc8689c6 to 1

Normal ScalingReplicaSet 6m28s (x2 over 23m) deployment-controller Scaled down replica set nginx-66bbc9fdc5 to 1

Normal ScalingReplicaSet 6m27s deployment-controller Scaled up replica set nginx-5dfc8689c6 to 2

Normal ScalingReplicaSet 5m58s deployment-controller Scaled down replica set nginx-66bbc9fdc5 to 0

Normal ScalingReplicaSet 4m19s deployment-controller Scaled up replica set nginx-6cdd5dd489 to 1

Normal ScalingReplicaSet 3m44s deployment-controller Scaled down replica set nginx-5dfc8689c6 to 1

Normal ScalingReplicaSet 3m44s deployment-controller Scaled up replica set nginx-6cdd5dd489 to 2

Normal ScalingReplicaSet 3m6s deployment-controller Scaled down replica set nginx-5dfc8689c6 to 0

## Deployment回滚

# 执行更新操作

[root@k8s-master01 ~]# **kubectl set image deploy nginx** nginx=nginx:787977da --record

deployment.apps/nginx image updated

[root@k8s-master01 ~]# **kubectl get po**

NAME READY STATUS RESTARTS AGE

nginx-6cdd5dd489-lv28z 1/1 Running 0 7m12s

nginx-6cdd5dd489-nqqz7 1/1 Running 0 6m37s

nginx-7d79b96f68-x7t67 0/1 ContainerCreating 0 19s

[root@k8s-master01 ~]# # 查看历史版本

[root@k8s-master01 ~]# **kubectl rollout history deploy nginx**

deployment.apps/nginx

REVISION CHANGE-CAUSE

1 <none>

2 kubectl set image deploy nginx nginx=nginx:1.15.3 --record=true

3 kubectl set image deploy nginx nginx=nginx:1.15.4 --record=true

4 kubectl set image deploy nginx nginx=nginx:787977da --record=true

[root@k8s-master01 ~]# # 回滚到上一个版本

[root@k8s-master01 ~]# **kubectl rollout undo deploy nginx**

deployment.apps/nginx rolled back

[root@k8s-master01 ~]# **kubectl get po**

NAME READY STATUS RESTARTS AGE

nginx-6cdd5dd489-lv28z 1/1 Running 0 9m8s

nginx-6cdd5dd489-nqqz7 1/1 Running 0 8m33s

[root@k8s-master01 ~]# **kubectl get deploy nginx -oyaml | grep nginx**

kubernetes.io/change-cause: kubectl set image deploy nginx nginx=nginx:1.15.4

app: nginx

k:{"name":"nginx"}:

k:{"name":"nginx"}:

name: nginx

selfLink: /apis/apps/v1/namespaces/default/deployments/nginx

app: nginx

app: nginx

- image: nginx:1.15.4

name: nginx

message: ReplicaSet "nginx-6cdd5dd489" has successfully progressed.

[root@k8s-master01 ~]# # 进行多次更新

[root@k8s-master01 ~]# **kubectl set image deploy nginx nginx=nginx:787977da --record**

deployment.apps/nginx image updated

[root@k8s-master01 ~]# **kubectl set image deploy nginx nginx=nginx:787977dadaa --record**

deployment.apps/nginx image updated

[root@k8s-master01 ~]# **kubectl set image deploy nginx nginx=nginx:787977xxxxxdadaa --record**

deployment.apps/nginx image updated

[root@k8s-master01 ~]# **kubectl set image deploy nginx nginx=nginx:787977dadxxxxxdadaa --record**

deployment.apps/nginx image updated

[root@k8s-master01 ~]# # 查看历史记录

[root@k8s-master01 ~]# **kubectl rollout history deploy nginx**

deployment.apps/nginx

REVISION CHANGE-CAUSE

1 <none>

2 kubectl set image deploy nginx nginx=nginx:1.15.3 --record=true

5 kubectl set image deploy nginx nginx=nginx:1.15.4 --record=true

6 kubectl set image deploy nginx nginx=nginx:787977da --record=true

7 kubectl set image deploy nginx nginx=nginx:787977dadaa --record=true

8 kubectl set image deploy nginx nginx=nginx:787977xxxxxdadaa --record=true

9 kubectl set image deploy nginx nginx=nginx:787977dadxxxxxdadaa --record=true

[root@k8s-master01 ~]# # 查看指定版本的详细信息

[root@k8s-master01 ~]# **kubectl rollout history deploy nginx --revision=5**

deployment.apps/nginx with revision #5

Pod Template:

Labels: app=nginx

pod-template-hash=6cdd5dd489

Annotations: kubernetes.io/change-cause: kubectl set image deploy nginx nginx=nginx:1.15.4 --record=true

Containers:

nginx:

Image: nginx:1.15.4

Port: <none>

Host Port: <none>

Environment: <none>

Mounts: <none>

Volumes: <none>

[root@k8s-master01 ~]# # 回滚到执行的版本

[root@k8s-master01 ~]# **kubectl rollout undo deploy nginx --to-revision=5**

deployment.apps/nginx rolled back

[root@k8s-master01 ~]# # 查看deploy状态

[root@k8s-master01 ~]# **kubectl get deploy -oyaml**

## Deployment的暂停和恢复

[root@k8s-master01 ~]# # Deployment 暂停功能

[root@k8s-master01 ~]# kubectl rollout pause deployment nginx

deployment.apps/nginx paused

[root@k8s-master01 ~]# kubectl set image deploy nginx nginx=nginx:1.15.3 --record

deployment.apps/nginx image updated

[root@k8s-master01 ~]# # 进行第二次配置变更

[root@k8s-master01 ~]# # 添加内存CPU配置

[root@k8s-master01 ~]# kubectl set -h

Configure application resources

These commands help you make changes to existing application resources.

Available Commands:

env Update environment variables on a pod template

image Update image of a pod template

resources Update resource requests/limits on objects with pod templates

selector Set the selector on a resource

serviceaccount Update ServiceAccount of a resource

subject Update User, Group or ServiceAccount in a RoleBinding/ClusterRoleBinding

Usage:

kubectl set SUBCOMMAND [options]

Use "kubectl <command> --help" for more information about a given command.

Use "kubectl options" for a list of global command-line options (applies to all commands).

[root@k8s-master01 ~]# kubectl set resources deploy nginx -c nginx --limits=cpu=200m,memory=128Mi --requests=cpu=10m,memory=16Mi

deployment.apps/nginx resource requirements updated

[root@k8s-master01 ~]# kubectl get deploy nginx -oyaml

apiVersion: apps/v1

kind: Deployment

metadata:

annotations:

deployment.kubernetes.io/revision: "11"

kubernetes.io/change-cause: kubectl set image deploy nginx nginx=nginx:1.15.3

--record=true

creationTimestamp: "2020-09-19T02:41:11Z"

generation: 18

labels:

app: nginx

name: nginx

namespace: default

resourceVersion: "2660534"

selfLink: /apis/apps/v1/namespaces/default/deployments/nginx

uid: 1d9253a5-a36c-48cc-aefe-56f95967db66

spec:

paused: true

progressDeadlineSeconds: 600

replicas: 2

revisionHistoryLimit: 10

selector:

matchLabels:

app: nginx

strategy:

rollingUpdate:

maxSurge: 25%

maxUnavailable: 25%

type: RollingUpdate

template:

metadata:

creationTimestamp: null

labels:

app: nginx

spec:

containers:

- image: nginx:1.15.3

imagePullPolicy: IfNotPresent

name: nginx

resources:

limits:

cpu: 200m

memory: 128Mi

requests:

cpu: 10m

memory: 16Mi

terminationMessagePath: /dev/termination-log

terminationMessagePolicy: File

dnsPolicy: ClusterFirst

restartPolicy: Always

schedulerName: default-scheduler

securityContext: {}

terminationGracePeriodSeconds: 30

status:

availableReplicas: 2

conditions:

- lastTransitionTime: "2020-09-19T03:26:50Z"

lastUpdateTime: "2020-09-19T03:26:50Z"

message: Deployment has minimum availability.

reason: MinimumReplicasAvailable

status: "True"

type: Available

- lastTransitionTime: "2020-09-19T03:30:15Z"

lastUpdateTime: "2020-09-19T03:30:15Z"

message: Deployment is paused

reason: DeploymentPaused

status: Unknown

type: Progressing

observedGeneration: 18

readyReplicas: 2

replicas: 2

[root@k8s-master01 ~]# # 查看pod是否被更新

[root@k8s-master01 ~]# kubectl get po

NAME READY STATUS RESTARTS AGE

nginx-6cdd5dd489-lv28z 1/1 Running 0 30m

nginx-6cdd5dd489-nqqz7 1/1 Running 0 30m

[root@k8s-master01 ~]# kubectl rollout resume deploy nginx

deployment.apps/nginx resumed

[root@k8s-master01 ~]# kubectl get rs

NAME DESIRED CURRENT READY AGE

nginx-5475c49ffb 0 0 0 21m

nginx-5dfc8689c6 0 0 0 33m

nginx-66bbc9fdc5 0 0 0 52m

nginx-68db656dd8 1 1 0 15s

nginx-6cdd5dd489 2 2 2 31m

nginx-799b8478d4 0 0 0 21m

nginx-7d79b96f68 0 0 0 24m

nginx-f974656f7 0 0 0 21m

## Deployment注意事项

# kubectl get deploy nginx -oyaml

apiVersion: apps/v1

kind: Deployment

metadata:

annotations:

deployment.kubernetes.io/revision: "12"

kubernetes.io/change-cause: kubectl set image deploy nginx nginx=nginx:1.15.3

--record=true

creationTimestamp: "2020-09-19T02:41:11Z"

generation: 19

labels:

app: nginx

name: nginx

namespace: default

spec:

progressDeadlineSeconds: 600

replicas: 2

revisionHistoryLimit: 10

selector:

matchLabels:

app: nginx

strategy:

rollingUpdate:

maxSurge: 25%

maxUnavailable: 25%

type: RollingUpdate

template:

metadata:

creationTimestamp: null

labels:

app: nginx

spec:

containers:

- image: nginx:1.15.3

imagePullPolicy: IfNotPresent

name: nginx

resources:

limits:

cpu: 200m

memory: 128Mi

requests:

cpu: 10m

memory: 16Mi

terminationMessagePath: /dev/termination-log

terminationMessagePolicy: File

dnsPolicy: ClusterFirst

restartPolicy: Always

schedulerName: default-scheduler

securityContext: {}

terminationGracePeriodSeconds: 30

* .spec.revisionHistoryLimit：设置保留RS旧的revision的个数，设置为0的话，不保留历史数据
* .spec.minReadySeconds：可选参数，指定新创建的Pod在没有任何容器崩溃的情况下视为Ready最小的秒数，默认为0，即一旦被创建就视为可用。
* 滚动更新的策略：
  + - .spec.strategy.type：更新deployment的方式，默认是RollingUpdate
      * RollingUpdate：滚动更新，可以指定maxSurge和maxUnavailable
        + maxUnavailable：指定在回滚或更新时最大不可用的Pod的数量，可选字段，默认25%，可以设置成数字或百分比，如果该值为0，那么maxSurge就不能0
        + maxSurge：可以超过期望值的最大Pod数，可选字段，默认为25%，可以设置成数字或百分比，如果该值为0，那么maxUnavailable不能为0
      * Recreate：重建，先删除旧的Pod，在创建新的Pod