### 2.2.6 Deployment

虽然ReplicaSet可以确保在任何给定时间运行的Pod副本达到指定的数量，但是Deployment（部署）是一个更高级的概念，它管理ReplicaSet并为Pod和ReplicaSet提供声明性更新以及许多其他有用的功能，所以建议在实际使用中，使用Deployment代替ReplicaSet。

如果在Deployment对象中描述了所需的状态，Deployment控制器就会以可控制的速率将实际状态更改为期望状态。也可以在Deployment中创建新的ReplicaSet，或者删除现有的Deployment并使用新的Deployment部署所用的资源。

2.2.6.1 创建Deployment

创建一个Deployment文件，并命名为dc-nginx.yaml，用于部署三个Nginx Pod：

apiVersion: apps/v1

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

示例解析：

* nginx-deployment：Deployment的名称。
* replicas： 创建Pod的副本数。
* selector：定义Deployment如何找到要管理的Pod，与template的label（标签）对应。
* template字段包含以下字段：
* app: nginx使用label（标签）标记Pod
* spec：表示Pod运行一个名字为nginx的容器。
* image：运行此Pod使用的镜像
* Port：容器用于发送和接收流量的端口

使用kubectl create创建此Deployment：

[root@k8s-master01 2.2.8.1]# kubectl create -f dc-nginx.yaml

deployment.apps/nginx-deployment created

使用kubectl get或者kubectl describe查看此Deployment：

[root@k8s-master01 2.2.8.1]# kubectl get deploy

NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE

nginx-deployment 3 3 3 1 60s

其中：

* NAME：集群中Deployment的名称。
* DESIRED：应用程序副本数。
* CURRENT：当前正在运行的副本数。
* UP-TO-DATE：显示已达到期望状态的被更新的副本数。
* AVAILABLE：显示用户可以使用的应用程序副本数，当前为1，因为部分Pod仍在创建过程中。
* AGE：显示应用程序运行的时间。

查看此时Deployment rollout的状态：

[root@k8s-master01 2.2.8.1]# kubectl rollout status deployment/nginx-deployment

deployment "nginx-deployment" successfully rolled out

再次查看此Deployment：

[root@k8s-master01 2.2.8.1]# kubectl get deploy

NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE

nginx-deployment 3 3 3 3 11m

查看此Deployment创建的ReplicaSet：

[root@k8s-master01 2.2.8.1]# kubectl get rs

NAME DESIRED CURRENT READY AGE

nginx-deployment-5c689d88bb 3 3 3 12m

查看此Deployment创建的Pod：

[root@k8s-master01 2.2.8.1]# kubectl get pods --show-labels

NAME READY STATUS RESTARTS AGE LABELS

nginx-deployment-5c689d88bb-6b95k 1/1 Running 0 13m app=nginx,pod-template-hash=5c689d88bb

nginx-deployment-5c689d88bb-9z5z2 1/1 Running 0 13m app=nginx,pod-template-hash=5c689d88bb

nginx-deployment-5c689d88bb-jc8hr 1/1 Running 0 13m app=nginx,pod-template-hash=5c689d88bb

2.2.6.2 更新Deployment

一般对应用程序升级或者版本迭代时，会通过Deployment对Pod进行滚动更新。

假如更新Nginx Pod的image使用nginx:1.9.1：

[root@k8s-master01 2.2.8.1]# kubectl set image deployment nginx-deployment nginx=nginx:1.9.1 --record

deployment.extensions/nginx-deployment image updated

当然也可以直接编辑Deployment，效果相同：

[root@k8s-master01 2.2.8.1]# kubectl edit deployment.v1.apps/nginx-deployment

deployment.apps/nginx-deployment edited

使用kubectl rollout status查看更新状态：

[root@k8s-master01 2.2.8.1]# kubectl rollout status deployment.v1.apps/nginx-deployment

Waiting for deployment "nginx-deployment" rollout to finish: 1 out of 3 new replicas have been updated...

Waiting for deployment "nginx-deployment" rollout to finish: 2 out of 3 new replicas have been updated...

Waiting for deployment "nginx-deployment" rollout to finish: 2 out of 3 new replicas have been updated...

Waiting for deployment "nginx-deployment" rollout to finish: 2 out of 3 new replicas have been updated...

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

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

deployment "nginx-deployment" successfully rolled out

查看ReplicaSet：

[root@k8s-master01 2.2.8.1]# kubectl get rs

NAME DESIRED CURRENT READY AGE

nginx-deployment-5c689d88bb 0 0 0 34m

nginx-deployment-6987cdb55b 3 3 3 5m14s

通过describe查看Deployment的详细信息：

[root@k8s-master01 2.2.8.1]# kubectl describe deploy nginx-deployment

Name: nginx-deployment

Namespace: default

CreationTimestamp: Thu, 24 Jan 2019 15:15:15 +0800

Labels: app=nginx

Annotations: deployment.kubernetes.io/revision: 2

kubernetes.io/change-cause: kubectl set image deployment nginx-deployment nginx=nginx:1.9.1 --record=true

Selector: app=nginx

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

Containers:

nginx:

Image: nginx:1.9.1

Port: 80/TCP

Host Port: 0/TCP

Environment: <none>

Mounts: <none>

Volumes: <none>

Conditions:

Type Status Reason

---- ------ ------

Available True MinimumReplicasAvailable

Progressing True NewReplicaSetAvailable

OldReplicaSets: <none>

NewReplicaSet: nginx-deployment-6987cdb55b (3/3 replicas created)

Events:

Type Reason Age From Message

---- ------ ---- ---- -------

Normal ScalingReplicaSet 36m deployment-controller Scaled up replica set nginx-deployment-5c689d88bb to 3

Normal ScalingReplicaSet 7m16s deployment-controller Scaled up replica set nginx-deployment-6987cdb55b to 1

Normal ScalingReplicaSet 5m18s deployment-controller Scaled down replica set nginx-deployment-5c689d88bb to 2

Normal ScalingReplicaSet 5m18s deployment-controller Scaled up replica set nginx-deployment-6987cdb55b to 2

Normal ScalingReplicaSet 4m35s deployment-controller Scaled down replica set nginx-deployment-5c689d88bb to 1

Normal ScalingReplicaSet 4m34s deployment-controller Scaled up replica set nginx-deployment-6987cdb55b to 3

Normal ScalingReplicaSet 3m30s deployment-controller Scaled down replica set nginx-deployment-5c689d88bb to 0

在describe中可以看出，第一次创建时，它创建了一个名为nginx-deployment-5c689d88bb的ReplicaSet，并直接将其扩展为3个副本。更新部署时，它创建了一个新的ReplicaSet，命名为nginx-deployment-6987cdb55b，并将其副本数扩展为1，然后将旧的ReplicaSet缩小为2，这样至少可以有2个Pod可用，最多创建了4个Pod。以此类推，使用相同的滚动更新策略向上和向下扩展新旧ReplicaSet，最终新的ReplicaSet可以拥有3个副本，并将旧的ReplicaSet缩小为0。

2.2.6.3 回滚Deployment

当新版本不稳定时，可以对其进行回滚操作，默认情况下，所有Deployment的rollout历史都保留在系统中，可以随时回滚。

假设我们又进行了几次更新：

[root@k8s-master01 2.2.8.1]# kubectl set image deployment nginx-deployment nginx=dotbalo/canary:v1 --record

[root@k8s-master01 2.2.8.1]# kubectl set image deployment nginx-deployment nginx=dotbalo/canary:v2 --record

使用kubectl rollout history查看部署历史：

[root@k8s-master01 2.2.8.1]# kubectl rollout history deployment/nginx-deployment

deployment.extensions/nginx-deployment

REVISION CHANGE-CAUSE

1 <none>

2 kubectl set image deployment nginx-deployment nginx=nginx:1.9.1 --record=true

3 kubectl set image deployment nginx-deployment nginx=dotbalo/canary:v1 --record=true

4 kubectl set image deployment nginx-deployment nginx=dotbalo/canary:v2 --record=true

查看Deployment某次更新的详细信息，使用--revision指定版本号：

[root@k8s-master01 2.2.8.1]# kubectl rollout history deployment.v1.apps/nginx-deployment --revision=3

deployment.apps/nginx-deployment with revision #3

Pod Template:

Labels: app=nginx

pod-template-hash=645959bf6b

Annotations: kubernetes.io/change-cause: kubectl set image deployment nginx-deployment nginx=dotbalo/canary:v1 --record=true

Containers:

nginx:

Image: dotbalo/canary:v1

Port: 80/TCP

Host Port: 0/TCP

Environment: <none>

Mounts: <none>

Volumes: <none>

使用kubectl rollout undo回滚到上一个版本：

[root@k8s-master01 2.2.8.1]# kubectl rollout undo deployment.v1.apps/nginx-deployment

deployment.apps/nginx-deployment

再次查看更新历史，发现REVISION5回到了canary:v1：

[root@k8s-master01 2.2.8.1]# kubectl rollout history deployment/nginx-deployment

deployment.extensions/nginx-deployment

REVISION CHANGE-CAUSE

1 <none>

2 kubectl set image deployment nginx-deployment nginx=nginx:1.9.1 --record=true

4 kubectl set image deployment nginx-deployment nginx=dotbalo/canary:v2 --record=true

5 kubectl set image deployment nginx-deployment nginx=dotbalo/canary:v1 --record=true

使用--to-revision参数回到指定版本：

[root@k8s-master01 2.2.8.1]# kubectl rollout undo deployment/nginx-deployment --to-revision=2

deployment.extensions/nginx-deployment

2.2.6.4 扩展Deployment

当公司访问量变大，三个Pod已无法支撑业务时，可以对其进行扩展。

使用kubectl scale动态调整Pod的副本数，比如增加Pod为5个：

[root@k8s-master01 2.2.8.1]# kubectl scale deployment.v1.apps/nginx-deployment --replicas=5

deployment.apps/nginx-deployment scaled

查看Pod，此时Pod已经变成了5个：

[root@k8s-master01 2.2.8.1]# kubectl get po

NAME READY STATUS RESTARTS AGE

nginx-deployment-5f89547d9c-5r56b 1/1 Running 0 90s

nginx-deployment-5f89547d9c-htmn7 1/1 Running 0 25s

nginx-deployment-5f89547d9c-nwxs2 1/1 Running 0 99s

nginx-deployment-5f89547d9c-rpwlg 1/1 Running 0 25s

nginx-deployment-5f89547d9c-vlr5p 1/1 Running 0 95s

2.2.6.5 暂停和恢复Deployment更新

Deployment支持暂停更新，用于对Deployment进行多次修改操作。

使用kubectl rollout pause暂停Deployment更新：

[root@k8s-master01 2.2.8.1]# kubectl rollout pause deployment/nginx-deployment

deployment.extensions/nginx-deployment paused

然后对Deployment进行相关更新操作，比如更新镜像，然后对其资源进行限制：

[root@k8s-master01 2.2.8.1]# kubectl set image deployment.v1.apps/nginx-deployment nginx=nginx:1.9.1

deployment.apps/nginx-deployment image updated

[root@k8s-master01 2.2.8.1]# kubectl set resources deployment.v1.apps/nginx-deployment -c=nginx --limits=cpu=200m,memory=512Mi

deployment.apps/nginx-deployment resource requirements updated

通过rollout history可以看到没有新的更新：

[root@k8s-master01 2.2.8.1]# kubectl rollout history deployment.v1.apps/nginx-deployment

deployment.apps/nginx-deployment

REVISION CHANGE-CAUSE

1 <none>

5 kubectl set image deployment nginx-deployment nginx=dotbalo/canary:v1 --record=true

7 kubectl set image deployment nginx-deployment nginx=dotbalo/canary:v2 --record=true

8 kubectl set image deployment nginx-deployment nginx=dotbalo/canary:v2 --record=true

使用kubectl rollout resume恢复Deployment更新：

[root@k8s-master01 2.2.8.1]# kubectl rollout resume deployment.v1.apps/nginx-deployment

deployment.apps/nginx-deployment resumed

可以查看到恢复更新的Deployment创建了一个新的RS（复制集）：

[root@k8s-master01 2.2.8.1]# kubectl get rs

NAME DESIRED CURRENT READY AGE

nginx-deployment-57895845b8 5 5 4 11s

可以查看Deployment的image（镜像）已经变为nginx:1.9.1

[root@k8s-master01 2.2.8.1]# kubectl describe deploy nginx-deployment

Name: nginx-deployment

Namespace: default

CreationTimestamp: Thu, 24 Jan 2019 15:15:15 +0800

Labels: app=nginx

Annotations: deployment.kubernetes.io/revision: 9

kubernetes.io/change-cause: kubectl set image deployment nginx-deployment nginx=dotbalo/canary:v2 --record=true

Selector: app=nginx

Replicas: 5 desired | 5 updated | 5 total | 5 available | 0 unavailable

StrategyType: RollingUpdate

MinReadySeconds: 0

RollingUpdateStrategy: 25% max unavailable, 25% max surge

Pod Template:

Labels: app=nginx

Containers:

nginx:

Image: nginx:1.9.1

Port: 80/TCP

Host Port: 0/TCP

2.2.6.6 更新Deployment的注意事项

清理策略：

在默认情况下，revision保留10个旧的ReplicaSet，其余的将在后台进行垃圾回收，可以在.spec.revisionHistoryLimit设置保留ReplicaSet的个数。当设置为0时，不保留历史记录。

更新策略：

* .spec.strategy.type==Recreate，表示重建，先删掉旧的Pod再创建新的Pod。
* .spec.strategy.type==RollingUpdate，表示滚动更新，可以指定maxUnavailable和maxSurge来控制滚动更新过程。
* .spec.strategy.rollingUpdate.maxUnavailable，指定在回滚更新时最大不可用的Pod数量，可选字段，默认为25%，可以设置为数字或百分比，如果maxSurge为0，则该值不能为0。
* .spec.strategy.rollingUpdate.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maxUnavailable为0，则该值不能为0。