

k8s.StatefulSet

statefulset(有状态集, 缩写为sts),常用于部署有状态的且需要有序启动的应用程序。

基本概念

statufulset主要用于管理有状态的应用程序工作负载API对象。比如在生产环境中, 可以部署ElasticSearch集群, MongoDB集群, 或者需要持久化的RabbitMQ集群, Redis集群, Kafka集群, 和Zookeeper集群等。

和Deployment类似, 一个Statefulset也同样管理着基于相同容器规范的Pod。不同的是, Statefulset为每一个Pod维护了一个粘性标识。这些Pod是根据相同的标识创建的, 但是不可以互换, 每个Pod都有一个持久的标识符, 在重新调度的时候也会保留, 一般格式为

StatefulSetName-Number.

比如定义一个名字是Redis-Sentinel的Statefulset, 指定创建三个Pod,那么创建处理的Pod名字就是 Redis-Sentinel-0,Redis-Sentinel-1,Redis-Sentinel-2.而StatefulSet创建的Pod一般使用HeadlessService(无头服务)进行通信, 和普通的Service区别在于HeadlessService没有ClusterIP, 它使用的是Endpoint进行相互通信, Headless一般格式为:

statefulset-{0...N-1}.serviceName.namespace.svc.cluster.local.

Headless格式说明:

statefulset-{0...N-1}.serviceName.namespace.svc.cluster.local.

- 1.serviceName为headlessService的名字, 创建statefulset的时候, 必须指定HeadlessService的名称。
- 2.0...N-1为Pod所在的序号, 从0开始到N-1
- 3.statefulsetName为statefulset的名字
4. .cluster.local为 Cluster Domain (集群域)

注意事项

一般Statefulset用于有以下几个或者多个需求的应用程序:

- 1.需要稳定的独一无二的网络标识。
- 2.需要持久化数据。
- 3.需要有序地, 优雅地部署和扩展。
- 4.需要有序的自动滚动更新。

如果应用程序不需要任何稳定的标识符或者有序的部署, 删除, 或者扩展, 应该使用无状态的控制部署应用程序。比如Deployment或者ReplicaSet。

StatefulSet在K8s1.9版本之前是beta资源，在1.5版本之前的任何 k8s版本是没有的。

Pod所用的存储必须由 PersistentVolume Provisioner(持久化卷配置器)根据请求配置 StorageClass，或者由管理员预先配置，当然也可以不配置存储。

为了确保数据安全，删除和缩放StatefulSet不会删除与StatefulSet相关联的卷，可以手动选择性的删除PVC或者PV。

StatefulSet目前使用的HeadlessService(无头服务)负责Pod的网络身份和通信，需要提前创建此服务。

删除一个StatefulSet时，不保证对Pod的终止，要在StatefulSet中实现Pod的有序和正常终止，可以在删除之前将StatefulSet的副本缩减为0。

创建一个StatefulSet资源

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: nginx
5    labels:
6      app: nginx
7  spec:
8    ports:
9      - port: 80
10     name: web
11    clusterIP: None
12    selector:
13      app: nginx
14  ---
15  apiVersion: apps/v1
16  kind: StatefulSet
17  metadata:
18    name: web
19  spec:
20    replicas: 2
21    selector:
22      matchLabels:
23        app: nginx
24    serviceName: "nginx"  template:
```

```

25     metadata:
26       labels:
27         app: nginx
28     spec:
29       containers:
30         - name: nginx
31           image: nginx
32           ports:
33             - containerPort: 80
34             name: web
35

```

基于statefulset.yaml文件创建statefulset。

```

1 → ~ kubectl apply -f ~/Desktop/K8sFiles/statefulset.yaml -n yurisa
2 service/nginx created
3 statefulset.apps/web created

```

```

1 → ~ kubectl get pod -n yurisa
2 NAME      READY   STATUS    RESTARTS   AGE
3 web-0     1/1     Running   0           23s
4 web-1     1/1     Running   0           6s
5 → ~ kubectl get sts -n yurisa
6 NAME      READY   AGE
7 web       2/2     30s
8

```

创建statefulset成功。

查看statefulset的Service

```

1 → ~ kubectl get svc -n yurisa

```


9	NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	
	NOMINATED	NODE	READINESS	GATES				
10	busybox	1/1	Running	87 (44m ago)	3d15h	10.244.1.28	k8s.node1	<none>
		<none>						
11	web-0	1/1	Running	1 (3d16h ago)	6d17h	10.244.1.27	k8s.node1	<none>
		<none>						
12	web-1	1/1	Running	1 (3d16h ago)	6d17h	10.244.2.28	k8s.node2	<none>
		<none>						
13	web-2	1/1	Running	1 (3d16h ago)	6d16h	10.244.2.29	k8s.node2	<none>
		<none>						
14	web-3	1/1	Running	0	30s	10.244.1.29	k8s.node1	<none>
		<none>						

在同Namespace启动一个用于观察的Pod

busyBox yaml配置文件如下:

```
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: busybox
5    namespace: yurisa
6  spec:
7    containers:
8      - name: busybox
9        image: busybox:1.28
10       command:
11         - sleep
12         - "3600"
13       imagePullPolicy: IfNotPresent
14     restartPolicy: Always
```

BUG

busybox的版本超过1.28之后, nslookup命令在k8s中无法正常工作。

启动Busybox,并解析StatusfulSet的地址

```
1  → ~ kubectl apply -f ~/Desktop/K8sFiles/busybox.yaml -n yurisa
```

```

2 pod/busybox created
3
4 → ~ kubectl get pod -n yurisa -owide
5 NAME          READY   STATUS    RESTARTS   AGE   IP            NODE          <none>
5 NOMINATED     NODE    READINESS GATES
6 busybox       1/1     Running   0           14s   10.244.1.31   k8s.node1    <none>
7 web-0         1/1     Running   1 (3d16h ago) 6d17h 10.244.1.27   k8s.node1    <none>
8 web-1         1/1     Running   1 (3d16h ago) 6d17h 10.244.2.28   k8s.node2    <none>
9 web-2         1/1     Running   0           80s   10.244.2.30   k8s.node2    <none>

```

进入busybox pod ,使用nslookup 查看statusfulset 地址

```

1 → ~ kubectl exec -it busybox -n yurisa -- sh
2 / # whoami
3 root
4 / # hostname
5 busybox
6 / # nslookup kubernetes.default
7 Server:      10.96.0.10
8 Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local
9
10 Name:        kubernetes.default
11 Address 1: 10.96.0.1 kubernetes.default.svc.cluster.local
12 / # nslookup nginx.yurisa
13 Server:      10.96.0.10
14 Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local
15
16 Name:        nginx.yurisa
17 Address 1: 10.244.2.30 web-2.nginx.yurisa.svc.cluster.local
18 Address 2: 10.244.2.28 web-1.nginx.yurisa.svc.cluster.local
19 Address 3: 10.244.1.27 web-0.nginx.yurisa.svc.cluster.local
20 / #
21 / # nslookup web-0.nginx
22 Server:      10.96.0.10
23 Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local
24

```

```

25 Name:      web-0.nginx
26 Address 1: 10.244.1.27 web-0.nginx.yurisa.svc.cluster.local
27 / # nslookup web-0.nginx.yurisa.cluster.local # 地址写错了, 少了svc没法被正确解析
28 Server:    10.96.0.10
29 Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local
30
31 nslookup: can't resolve 'web-0.nginx.yurisa.cluster.local'
32 / # nslookup web-0.nginx.yurisa.svc.cluster.local
33 Server:    10.96.0.10
34 Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local
35
36 Name:      web-0.nginx.yurisa.svc.cluster.local
37 Address 1: 10.244.1.27 web-0.nginx.yurisa.svc.cluster.local
38

```

访问 nginx

```

1 / # wget web-1.nginx
2 Connecting to web-1.nginx (10.244.2.28:80)
3 index.html      100%
  |*****|
  |*****| 615    0:00:00 ETA
4 / # cat index.html
5 <!DOCTYPE html>
6 <html>
7 <head>
8 <title>Welcome to nginx!</title>
9 ..... 省略

```

StatusfulSet扩容缩容过程

- a. 对于包含 N 个 副本的 StatefulSet, 当部署 Pod 时, 它们是依次创建的, 顺序为 0..N-1。即先从0开始创建, 然后依次往下。
- b. 在将扩缩操作应用到 Pod 之前, 它前面的所有 Pod 必须是 Running 和 Ready 状态。即在创建第N个Pod的过程中, 存在0~N-1个Pod发生故障, 应该先等待0~N-1Pod变为正常(Ready或是Running)才能继续创建第N个Pod。
- c. 当删除 Pod 时, 它们是逆序终止的, 顺序为 N-1..0。即删除和创建是反着来的过程, 先从N开始, 直接0结束整个删除过程。

- d. 在一个 Pod 终止之前，**所有的继任者必须完全关闭**。即删除过程是有序的同时，必选保证第N个Pod完全关闭才会去关闭第N-1个。

statusfulset扩容过程:

```
1  ~ kubectl get pod -n yurisa -w
2  NAME      READY   STATUS    RESTARTS      AGE
3  web-0     1/1     Running   1 (3d16h ago)  6d18h    # 当前只有一个Pod
4
5  web-1     0/1     Pending   0              0s
6  web-1     0/1     Pending   0              0s
7  web-1     0/1     ContainerCreating 0              0s
8  web-1     1/1     Running   0              17s      # web-1 Pod 创建成功
9
10
11 web-2     0/1     Pending   0              0s
12 web-2     0/1     Pending   0              0s
13 web-2     0/1     ContainerCreating 0              0s      # web-2 Pod正在创建，创建
    过程成执行 `kubectl delete pod web-0 -n yurisa`
14
15 web-0     1/1     Terminating 1 (3d16h ago)  6d18h    # 由于执行删除操作，web-0
    Pod正在终止
16 web-0     0/1     Terminating 1 (3d16h ago)  6d18h
17 web-0     0/1     Terminating 1 (3d16h ago)  6d18h
18 web-0     0/1     Terminating 1 (3d16h ago)  6d18h
19
20 web-0     0/1     Pending   0              0s      # web-0由于statusfulset控
    制器管理，所有web-0又开始创建
21 web-0     0/1     Pending   0              0s
22 web-0     0/1     ContainerCreating 0              0s
23 web-0     1/1     Running   0              1s      # web-0创建成功
24
25 web-2     1/1     Running   0              16s      # 当web-0创建成功之后 web-
    2才成功
26
27 web-3     0/1     Pending   0              0s      # web-3开始创建
28 web-3     0/1     Pending   0              0s
29 web-3     0/1     ContainerCreating 0              0s
30 web-3     1/1     Running   0              16s      # web-3创建成功
31
```



```

32 web-4 0/1 Pending 0 0s # web-4开始创建
33 web-4 0/1 Pending 0 0s
34 web-4 0/1 ContainerCreating 0 0s # web-4 正在创建，创建过程中执行 `kubectl delete pod web-0 -n yurisa`
35
36 web-0 1/1 Terminating 0 28s # 由于执行删除操作，web-0 Pod正在终止
37 web-0 0/1 Terminating 0 30s
38 web-0 0/1 Terminating 0 30s
39 web-0 0/1 Terminating 0 30s
40
41 web-0 0/1 Pending 0 0s # web-0由于statusfulset控制器管理，所有web-0又开始创建
42 web-0 0/1 Pending 0 0s
43 web-0 0/1 ContainerCreating 0 0s
44 web-4 1/1 Running 0 16s # web-4 创建成功 此时web-0是Ready状态，所以web-4可以继续创建，因此web-4的日志先打印出来
45 web-0 1/1 Running 0 28s # web-0 创建成功
46

```

statusfulset缩容过程:

- i. 缩容过程中删除的Pod是在缩容列表里面，则delete的Pod不会被k8s重启。(详见第一种)
- ii. 缩容过程中被删除的Pod不是在缩容列表里面，则删除过程中会先重新启动被删除的Pod,直到Pod的状态变成(Ready or Running)状态才会接着继续往下删除。(详见第二种)

```

1 # 第一种:
2 → ~ kubectl get pod -n yurisa -w
3 NAME      READY   STATUS    RESTARTS   AGE
4 web-0     1/1     Running   0           28m
5 web-1     1/1     Running   0           27m
6 web-2     1/1     Running   0           29m
7 web-3     1/1     Running   0           28m
8 web-4     1/1     Running   0           28m # 从web 0-4,一共5个Pod
9
10 web-4    1/1     Terminating 0           29m
11
12 web-1    1/1     Terminating 0           28m # 执行 delete pod web-1
13

```

```

14 web-4 0/1 Terminating 0 29m
15 web-4 0/1 Terminating 0 29m
16 web-4 0/1 Terminating 0 29m # web-4先终止
17
18 web-3 1/1 Terminating 0 29m
19 web-3 0/1 Terminating 0 29m
20 web-3 0/1 Terminating 0 29m
21 web-3 0/1 Terminating 0 29m # web-3再终止
22
23 web-2 1/1 Terminating 0 29m # web-2再终止
24
25 web-1 0/1 Terminating 0 28m
26 web-1 0/1 Terminating 0 28m
27 web-1 0/1 Terminating 0 28m # web-1再终止
28
29 web-2 0/1 Terminating 0 29m
30 web-2 0/1 Terminating 0 29m
31 web-2 0/1 Terminating 0 29m # web-2完全终止

```

```

1 # 第二种
2 → ~ kubectl get pod -n yurisa -w
3 NAME      READY   STATUS    RESTARTS   AGE
4 web-0     1/1     Running   0           39m
5 web-1     1/1     Running   0           74s
6 web-2     1/1     Running   0           58s
7 web-3     1/1     Running   0           42s
8 web-4     1/1     Running   0           40s # web0-4 一共5个Pod
9
10 web-4     1/1     Terminating 0           55s # web-4最先被终止
11 web-1     1/1     Terminating 0           90s
12 web-4     0/1     Terminating 0           57s
13 web-4     0/1     Terminating 0           57s
14 web-4     0/1     Terminating 0           57s
15
16 web-1     0/1     Terminating 0           92s # web-4终止过程中, 执行`kubectl delete
    pod web-1 -n yurisa`
17 web-1     0/1     Terminating 0           92s

```

```

18 web-1 0/1 Terminating 0 92s # web-1被终止
19 web-1 0/1 Pending 0 0s # 由于web-1不在此次的缩容列表的范围内, k8s
    又重新创建了web-1
20 web-1 0/1 Pending 0 0s
21 web-1 0/1 ContainerCreating 0 0s
22 web-1 1/1 Running 0 17s # web-1创建成功
23
24 web-3 1/1 Terminating 0 77s # 按顺序执行, web-3终止
25 web-3 0/1 Terminating 0 79s
26 web-3 0/1 Terminating 0 79s
27 web-3 0/1 Terminating 0 79s
28 web-2 1/1 Terminating 0 95s # 按顺序执行, web-2终止
29 web-2 0/1 Terminating 0 97s
30 web-2 0/1 Terminating 0 97s
31 web-2 0/1 Terminating 0 97s
32

```

StatusfulSet更新策略

StatefulSet 的 `.spec.updateStrategy` 字段让你可以配置和禁用掉自动滚动更新 Pod 的容器、标签、资源请求或限制、以及注解。有两个允许的值：

OnDelete

当 StatefulSet 的 `.spec.updateStrategy.type` 设置为 `OnDelete` 时，它的控制器将不会自动更新 StatefulSet 中的 Pod。用户必须手动删除 Pod 以便让控制器创建新的 Pod，以此来对 StatefulSet 的 `.spec.template` 的变动作出反应。

RollingUpdate

`RollingUpdate` 更新策略对 StatefulSet 中的 Pod 执行自动的滚动更新。这是默认的更新策略。

```

1  updateStrategy:
2    rollingUpdate:
3      partition: 0 # 小于这个值的Pod不会被更新到, 假设有5个Pod, 即web0-4, 当前值为2, 则只
    会更新 web-4, web-3, web-2
4      type: RollingUpdate
5
6  updateStrategy:
7    type: OnDelete

```

StatusfulSet更新策略之 RollingUpdate

当statusfulset的更新策略为rollingupdate时。

更新操作为逆序更新，会先更新最后一个，其次在更新前一个。示例如下：

```
1 → ~ kubectl get pod -n yurisa -w
2 NAME      READY   STATUS    RESTARTS   AGE
3 web-0     1/1     Running   0           79m
4 web-1     1/1     Running   0           40m
5 web-2     1/1     Running   0           89s
6 web-3     1/1     Running   0           72s    # web0-3 4 Pod
7
8 web-3     1/1     Terminating 0           4m53s    # web-3 update
9 web-3     0/1     Terminating 0           4m56s
10 web-3    0/1     Terminating 0           4m56s
11 web-3    0/1     Terminating 0           4m56s
12 web-3    0/1     Pending      0           0s
13 web-3    0/1     Pending      0           0s
14 web-3    0/1     ContainerCreating 0           0s
15 web-3    1/1     Running      0           19s
16
17 web-2     1/1     Terminating 0           5m32s    # web-2 update
18 web-2     0/1     Terminating 0           5m33s
19 web-2     0/1     Terminating 0           5m33s
20 web-2     0/1     Terminating 0           5m33s
21 web-2     0/1     Pending      0           0s
22 web-2     0/1     Pending      0           0s
23 web-2     0/1     ContainerCreating 0           0s
24 web-2     1/1     Running      0           19s
25
26 web-1     1/1     Terminating 0           44m    # web-1 update
27 web-1     0/1     Terminating 0           44m
28 web-1     0/1     Terminating 0           44m
29 web-1     0/1     Terminating 0           44m
30 web-1     0/1     Pending      0           0s
31 web-1     0/1     Pending      0           0s
32 web-1     0/1     ContainerCreating 0           0s
33 web-1     1/1     Running      0           3s
```

```

34
35 web-0 1/1 Terminating 0 84m # web-0 update
36 web-0 0/1 Terminating 0 84m
37 web-0 0/1 Terminating 0 84m
38 web-0 0/1 Terminating 0 84m
39 web-0 0/1 Pending 0 0s
40 web-0 0/1 Pending 0 0s
41 web-0 0/1 ContainerCreating 0 0s
42 web-0 1/1 Running 0 18s
43

```

StatusfulSet更新策略之 RollingUpdate且partition

当前有5个Pod,设置partition的值为2

当partition的值设置为0之后，剩下的更新则会继续。

更新操作仅更新了web-4,web-3以及web-2,web-1和web-0并没有被更新了，可以使用此策略实现简单的灰度发布。示例如下：

```

1 → ~ kubectl get pod -n yurisa -w
2 NAME      READY   STATUS    RESTARTS   AGE
3 web-0     1/1     Running   0          8m42s
4 web-1     1/1     Running   0          8m46s
5 web-2     1/1     Running   0          9m5s
6 web-3     1/1     Running   0          9m25s
7 web-4     1/1     Running   0          58s # web0-4 5个Pod
8
9 web-4     1/1     Terminating 0 109s # web-4 update
10 web-4    0/1     Terminating 0 112s
11 web-4    0/1     Terminating 0 112s
12 web-4    0/1     Terminating 0 112s
13 web-4    0/1     Pending      0 0s
14 web-4    0/1     Pending      0 0s
15 web-4    0/1     ContainerCreating 0 0s
16 web-4    1/1     Running      0 42s
17
18 web-3     1/1     Terminating 0 11m # web-3 update
19 web-3     0/1     Terminating 0 11m

```

20	web-3	0/1	Terminating	0	11m	
21	web-3	0/1	Terminating	0	11m	
22	web-3	0/1	Pending	0	0s	
23	web-3	0/1	Pending	0	0s	
24	web-3	0/1	ContainerCreating	0	0s	
25	web-3	1/1	Running	0	18s	
26						
27	web-2	1/1	Terminating	0	11m	# web-2 update 到此次终止
28	web-2	0/1	Terminating	0	11m	
29	web-2	0/1	Terminating	0	11m	
30	web-2	0/1	Terminating	0	11m	
31	web-2	0/1	Pending	0	0s	
32	web-2	0/1	Pending	0	0s	
33	web-2	0/1	ContainerCreating	0	0s	
34	web-2	1/1	Running	0	37s	
35						

版本号异同,说明只更新了 web-4,web-3以及web-2

```

1 → ~ kubectl get pod -oyaml -n yurisa | grep image
2   - image: nginx:1.25
3     imagePullPolicy: Always
4     image: docker.io/library/nginx:1.25
5     imageID:
6     docker.io/library/nginx@sha256:08bc36ad52474e528cc1ea3426b5e3f4bad8a130318e3140d6cfe29c8892c7ef
7   - image: nginx:1.25
8     imagePullPolicy: Always
9     image: docker.io/library/nginx:1.25
10    imageID:
11    docker.io/library/nginx@sha256:08bc36ad52474e528cc1ea3426b5e3f4bad8a130318e3140d6cfe29c8892c7ef
12  - image: nginx:1.24
13    imagePullPolicy: Always
14    image: docker.io/library/nginx:1.24
15    imageID:
16    docker.io/library/nginx@sha256:45f5d97d1817508d0cb83c5ae4ab1c6007a4f53b5e193e32aa45126eb4f00fe4
17  - image: nginx:1.24
18    imagePullPolicy: Always

```

```

16     image: docker.io/library/nginx:1.24
17     imageID:
18     docker.io/library/nginx@sha256:45f5d97d1817508d0cb83c5ae4ab1c6007a4f53b5e193e32aa45126e
19     b4f00fe4
20     - image: nginx:1.24
21     imagePullPolicy: Always
22     image: docker.io/library/nginx:1.24
23     imageID:
24     docker.io/library/nginx@sha256:45f5d97d1817508d0cb83c5ae4ab1c6007a4f53b5e193e32aa45126e
25     b4f00fe4

```

当Partition设置为0之后，web-1,web-0会继续更新。示例如下：

```

1  → ~ kubectl get pod -n yurisa -w
2  NAME      READY   STATUS    RESTARTS   AGE
3  web-0     1/1     Running   0           20m
4  web-1     1/1     Running   0           20m
5  web-2     1/1     Running   0           9m47s
6  web-3     1/1     Running   0           10m
7  web-4     1/1     Running   0           10m
8
9  web-1     1/1     Terminating   0           20m
10 web-1     0/1     Terminating   0           20m
11 web-1     0/1     Terminating   0           20m
12 web-1     0/1     Terminating   0           20m
13 web-1     0/1     Pending        0           0s
14 web-1     0/1     Pending        0           0s
15 web-1     0/1     ContainerCreating 0           0s
16 web-1     1/1     Running        0           3s
17
18 web-0     1/1     Terminating   0           20m
19 web-0     0/1     Terminating   0           20m
20 web-0     0/1     Terminating   0           20m
21 web-0     0/1     Terminating   0           20m
22 web-0     0/1     Pending        0           0s
23 web-0     0/1     Pending        0           0s
24 web-0     0/1     ContainerCreating 0           0s

```

```
25 web-0 1/1 Running 0 18s
26
```

检查所有的Pod使用的镜像版本是否一致，如一致，则更新成功。

```
1 → ~ kubectl get pod -oyaml -n yurisa | grep image
2     - image: nginx:1.24
3       imagePullPolicy: Always
4       image: docker.io/library/nginx:1.24
5       imageID:
6 docker.io/library/nginx@sha256:45f5d97d1817508d0cb83c5ae4ab1c6007a4f53b5e193e32aa45126e
7 b4f00fe4
8     - image: nginx:1.24
9       imagePullPolicy: Always
10      image: docker.io/library/nginx:1.24
11      imageID:
12 docker.io/library/nginx@sha256:45f5d97d1817508d0cb83c5ae4ab1c6007a4f53b5e193e32aa45126e
13 b4f00fe4
14     - image: nginx:1.24
15       imagePullPolicy: Always
16       image: docker.io/library/nginx:1.24
17       imageID:
18 docker.io/library/nginx@sha256:45f5d97d1817508d0cb83c5ae4ab1c6007a4f53b5e193e32aa45126e
19 b4f00fe4
20     - image: nginx:1.24
21       imagePullPolicy: Always
22       image: docker.io/library/nginx:1.24
23       imageID:
24 docker.io/library/nginx@sha256:45f5d97d1817508d0cb83c5ae4ab1c6007a4f53b5e193e32aa45126e
25 b4f00fe4
26
```

StatusfulSet更新策略之 OnDelete

当yaml文件发生变化，k8s不会自动更新，只有当pod被删除的时候Pod才会被更新

1.修改更新策略为OnDelete

```
1  updateStrategy:
2      type: OnDelete
3
4  → ~ kubectl edit sts web -n yurisa
5  statefulset.apps/web edited
```

2.修改template中的image镜像版本:

```
1  spec.template.spec.containers.image:1.24 --> 1.25
2
3  → ~ kubectl edit sts web -n yurisa
4  statefulset.apps/web edited
5
```

3.删除Pod web-4

```
1  → ~ kubectl delete pod web-4 -n yurisa
2  pod "web-4" deleted
3
```

此时,触发Pod更新:

```
1  → ~ kubectl get pod -n yurisa -w
2  NAME      READY   STATUS    RESTARTS   AGE
3  web-0     1/1     Running   0           5m7s
4  web-1     1/1     Running   0           5m11s
5  web-2     1/1     Running   0           15m
6  web-3     1/1     Running   0           15m
7  web-4     1/1     Running   0           16m
8  web-4     1/1     Terminating 0           50m
9  web-4     0/1     Terminating 0           50m
10 web-4     0/1     Terminating 0           50m
11 web-4     0/1     Terminating 0           50m
```

```

12 web-4 0/1 Pending 0 0s
13 web-4 0/1 Pending 0 0s
14 web-4 0/1 ContainerCreating 0 0s
15 web-4 1/1 Running 0 18s
16

```

4.继续删除Pod web-2,再次触发更新

```

1 → ~ kubectl get pod -n yurisa -w
2 NAME      READY   STATUS    RESTARTS   AGE
3 web-0     1/1     Running   0          5m7s
4 web-1     1/1     Running   0          5m11s
5 web-2     1/1     Running   0          15m
6 web-3     1/1     Running   0          15m
7 web-4     1/1     Running   0          16m    # 5个Pod
8
9 web-4     1/1     Terminating 0          50m    # 删除web-4
10 web-4     0/1     Terminating 0          50m
11 web-4     0/1     Terminating 0          50m
12 web-4     0/1     Terminating 0          50m
13 web-4     0/1     Pending      0          0s
14 web-4     0/1     Pending      0          0s
15 web-4     0/1     ContainerCreating 0          0s
16 web-4     1/1     Running      0          18s
17
18 web-2     1/1     Terminating 0          51m    # 删除web-2
19 web-2     0/1     Terminating 0          51m
20 web-2     0/1     Terminating 0          51m
21 web-2     0/1     Terminating 0          51m
22 web-2     0/1     Pending      0          0s
23 web-2     0/1     Pending      0          0s
24 web-2     0/1     ContainerCreating 0          0s
25 web-2     1/1     Running      0          18s
26

```

5.验证删除过程是否成功，通过比较image的版本号即可知道

```

1 → ~ kubectl get pod -oyaml -n yurisa | grep image
2     - image: nginx:1.24
3       imagePullPolicy: Always
4       image: docker.io/library/nginx:1.24
5       imageID:
6     docker.io/library/nginx@sha256:45f5d97d1817508d0cb83c5ae4ab1c6007a4f53b5e193e32aa45126e
7     b4f00fe4
8     - image: nginx:1.24
9       imagePullPolicy: Always
10      image: docker.io/library/nginx:1.24
11      imageID:
12    docker.io/library/nginx@sha256:45f5d97d1817508d0cb83c5ae4ab1c6007a4f53b5e193e32aa45126e
13    b4f00fe4
14    - image: nginx:1.25
15      imagePullPolicy: Always
16      image: docker.io/library/nginx:1.25
17      imageID:
18    docker.io/library/nginx@sha256:08bc36ad52474e528cc1ea3426b5e3f4bad8a130318e3140d6cfe29c
19    8892c7ef
20    - image: nginx:1.24
21      imagePullPolicy: Always
22      image: docker.io/library/nginx:1.24
23      imageID:
24    docker.io/library/nginx@sha256:45f5d97d1817508d0cb83c5ae4ab1c6007a4f53b5e193e32aa45126e
25    b4f00fe4
26    - image: nginx:1.25
27      imagePullPolicy: Always
28      image: docker.io/library/nginx:1.25
29      imageID:
30    docker.io/library/nginx@sha256:08bc36ad52474e528cc1ea3426b5e3f4bad8a130318e3140d6cfe29c
31    8892c7ef

```

StatusfulSet级联删除

1.级联删除

级联删除在删除statusfulset的同时也会删除statusfulset相关的Pod

```

1 → ~ kubectl get sts -n yurisa
2 NAME      READY    AGE
3 web       5/5      6d20h
4
5 → ~ kubectl delete sts web -n yurisa
6 statefulset.apps "web" deleted
7 → ~ kubectl get sts -n yurisa
8 No resources found in yurisa namespace.
9

```

删除监控过程如下:

```

1 → ~ kubectl get pod -n yurisa -w
2 NAME      READY    STATUS      RESTARTS   AGE
3 web-0     1/1      Running     0           59m
4 web-1     1/1      Running     0           59m
5 web-2     1/1      Running     0           18m
6 web-3     1/1      Running     0           69m
7 web-4     1/1      Running     0           19m
8 web-0     1/1      Terminating 0           59m
9 web-1     1/1      Terminating 0           59m
10 web-2     1/1      Terminating 0           18m
11 web-3     1/1      Terminating 0           70m
12 web-4     1/1      Terminating 0           20m
13 web-0     0/1      Terminating 0           59m
14 web-0     0/1      Terminating 0           59m
15 web-0     0/1      Terminating 0           59m
16 web-2     0/1      Terminating 0           18m
17 web-2     0/1      Terminating 0           18m
18 web-2     0/1      Terminating 0           18m
19 web-1     0/1      Terminating 0           59m
20 web-1     0/1      Terminating 0           59m
21 web-1     0/1      Terminating 0           59m
22 web-4     0/1      Terminating 0           20m
23 web-4     0/1      Terminating 0           20m
24 web-4     0/1      Terminating 0           20m
25 web-3     0/1      Terminating 0           70m

```

```
26 web-3 0/1 Terminating 0 70m
27 web-3 0/1 Terminating 0 70m
28
```

2.非级联删除

非级联删除模式下，删除statusfulset并不会删除Pod,此时Pod会变成孤儿Pod，删除之后不会被k8s重建。

非级联删除，设置 `--cascade=false`

```
1 → ~ kubectl get pod -n yurisa # 查看当前Pod
2 NAME      READY   STATUS    RESTARTS   AGE
3 web-0     1/1     Running   0          34s
4 web-1     1/1     Running   0          18s
5
6 → ~ kubectl delete sts web --cascade=false -n yurisa # 采用非级联删除的模式删除
  statusfulset
7 warning: --cascade=false is deprecated (boolean value) and can be replaced with --
  cascade=orphan.
8 statefulset.apps "web" deleted
9
10 → ~ kubectl get pod -n yurisa # 再次查看Pod,发现Pod资源还在，但是此时Pod已经变成了孤儿
   Pod了。
11 NAME      READY   STATUS    RESTARTS   AGE
12 web-0     1/1     Running   0          95s
13 web-1     1/1     Running   0          79s
14
15 → ~ kubectl get sts -n yurisa # 查看statusfulset控制器已经没有了
16 No resources found in yurisa namespace.
17
18 → ~ kubectl delete pod web-0 web-1 -n yurisa # 删除孤儿Pod
19 pod "web-0" deleted
20 pod "web-1" deleted
21
```

删除过程监控如下:

```
1 → ~ kubectl get pod -n yurisa -w
2 NAME      READY   STATUS    RESTARTS   AGE # 当前statusfuleset下的Pod
```

```
3 web-0 1/1 Running 0 45s
4 web-1 1/1 Running 0 29s
5
6 web-0 1/1 Running 0 69s # 执行`kubectl delete sts web --
  cascade=false -n yurisa`仅仅只是删除了statusfulset并未删除Pod
7 web-1 1/1 Running 0 53s
8
9 web-0 1/1 Terminating 0 117s # 执行delete操作删除Pod,Pod此时被真正删
  除
10 web-1 1/1 Terminating 0 101s
11 web-0 0/1 Terminating 0 119s
12 web-0 0/1 Terminating 0 119s
13 web-0 0/1 Terminating 0 119s
14 web-1 0/1 Terminating 0 103s
15 web-1 0/1 Terminating 0 103s
16 web-1 0/1 Terminating 0 103s
17
```