



大家都在搜....



下载APP

开源软件

问答

动弹

博


打赏 

评论 

收藏 

点赞 

分享文章

微博 

QQ 

微信 











# 浅析Kubernetes StatefulSet 顶 原 荐



WaltonWang 发布于 03/14 00:02 字数 2764 阅读 522 收藏 5 点赞 1 评论 0

Kubernetes StatefulSet

## StatefulSet和Deployment的区别

“Deployment用于部署无状态服务，StatefulSet用来部署有状态服务”。

具体的，什么场景需要使用StatefulSet呢？官方给出的建议是，如果你部署的应用满足以下一个或多个部署需求，则建议使用StatefulSet。

- **稳定的**、唯一的网络标识。
- **稳定的**、持久的存储。
- **有序的**、优雅的部署和伸缩。
- **有序的**、优雅的删除和停止。
- **有序的**、自动的滚动更新。

**稳定的**主要是针对Pod发生re-schedule后仍然要保持之前的网络标识和持久化存储。这里所说的网络标识包括hostname、集群内DNS中该Pod对应的A Record，并不能保证Pod re-schedule之后IP不变。要想保持Pod IP不变，我们可以借助稳定的Pod hostname定制IPAM获取固定的Pod IP。借助StatefulSet的 稳定的唯一的网络标识 特性，我们能比较轻松的实现Pod的固定IP需求，然后如果使用Deployment，那么将会复杂的多，你需要考虑滚动更新的过程中的参数控制(maxSurge、maxUnavailable)、每个应用的IP池预留造成的IP浪费等等问题。

因此，我想再加一个StatefulSet的使用场景：

- 实现固定的Pod IP方案, 可以优先考虑基于StatefulSet;

## 最佳实践

- StatefulSet对应Pod的存储最好通过StorageClass来动态创建: 每个Pod都会根据StatefulSet中定义的VolumeClaimTemplate来创建一个对应的PVC, 然后PVS通过StorageClass自动创建对应的PV, 并挂载给Pod。所以这种方式, 需要你事先创建好对应的StorageClass。当然, 你也可以通过预先由管理员手动创建好对应的PV, 只要能保证自动创建的PVC能和这些PV匹配上。
- 为了数据安全, 当删除StatefulSet中Pods或者对StatefulSet进行缩容时, Kubernetes并不会自动删除StatefulSet对应的PV, 而且这些PV默认也不能被其他PVC Bound。当你确认数据无用之后再手动去删除PV的时候, 数据是否删除取决于PV的ReclaimPolicy配置。Reclaim Policy支持以下三种:
  - Retain, 意味着需要你手动清理;
  - Recycle, 等同于rm -rf /thevolume/\*
  - Delete, 默认值, 依赖于后端的存储系统自己实现。

注意:

- 目前只有NFS和HostPath支持Recycle;
  - EBS,GCE PD, Azure Disk, Openstack Cinder支持Delete。
- 
- 请小心删除StatefulSet对应的PVC, 首先确保Pods已经完全Terminate, 然后确定不需要Volume中的数据后, 再考虑删除PV。因为删除PVC可能触发对应PV的自动删除, 并根据StorageClass中的reclaimPolicy配置可能造成volume中的数据丢失。
  - 因为部署的是有状态应用, 我们需要自己创建对应的Headless Service, 注意Label要和StatefulSet中Pods的Label匹配。Kubernetes会为该Headless Service创建对应SRV Records, 包含所有的后端Pods, KubeDNS会通过Round Robin算法进行选择。
  - 在Kubernetes 1.8+中, 你必须保证StatefulSet的spec.selector能匹配.spec.template.metadata.labels, 否则会导致StatefulSet创建失败。在Kubernetes 1.8之前, StatefulSet的spec.selector如果没指定则默认会等同于.spec.template.metadata.labels。
  - 对StatefulSet进行缩容前, 你需要确认对应的Pods都是Ready的, 否则即使你触发了缩容操作, Kubernetes也不会真的进行缩容操作。

## 如何理解稳定的网络标识

StatefulSet中反复强调的“稳定的网络标识”，主要指Pods的hostname以及对应的DNS Records。

- **HostName**：StatefulSet的Pods的hostname按照这种格式生成：`$(statefulset name)-$(ordinal)`，`ordinal`从0 ~ N-1 (N为期望副本数)。
  - StatefulSet Controller在创建pods时，会给pod加上一个pod name label：`statefulset.kubernetes.io/pod-name`，然后设置到Pod的pod name和hostname中。
  - pod name label有啥用呢？我们可以创建独立的Service匹配到这个指定的pod，然后方便我们单独对这个pod进行debug等处理。
- **DNS Records**：
  - Headless Service的DNS解析：`$(service name).$(namespace).svc.cluster.local` 通过DNS RR解析到后端其中一个Pod。SRV Records只包含对应的Running and Ready的Pods，不Ready的Pods不会在对应的SRV Records中。
  - Pod的DNS解析：`$(hostname).$(service name).$(namespace).svc.cluster.local` 解析到对应hostname的Pod。

## 如何理解稳定的持久化存储

- 每个Pod对应一个PVC，PVC的名称是这样组成的：`$(volumeClaimTemplates.name)-$(pod's hostname)`，跟对应的Pod是一一对应的。
- 当Pod发生re-schedule（其实是recreate）后，它所对应的PVC所Bound的PV仍然会自动的挂载到新的Pod中。
- Kubernetes会按照VolumeClaimTemplate创建N(N为期望副本数)个PVC，由PVCs根据指定的StorageClass自动创建PVs。
- 当通过级联删除StatefulSet时并不会自动删除对应的PVCs，所以PVC需要手动删除。
- 当通过级联删除StatefulSet或者直接删除对应Pods时，对应的PVs并不会自动删除。需要你手动的去删除PV。

## 部署和伸缩时与Deployment的区别

- 当部署有N个副本的StatefulSet应用时，严格按照index从0到N-1的递增顺序创建，下一个Pod创建必须是前一个Pod Ready为前提。
- 当删除有N个副本的StatefulSet应用时，严格按照index从N-1到0的递减顺序删除，下一个Pod删除必须是前一个Pod shutdown并完全删除为前提。
- 当扩容StatefulSet应用时，每新增一个Pod必须是前一个Pod Ready为前提。
- 当缩容StatefulSet应用时，没删除一个Pod必须是前一个Pod shutdown并成功删除为前提。
- 注意StatefulSet的pod.Spec.TerminationGracePeriodSeconds不要设置为0。



## Node网络异常等情况下该如何处理

- 正常情况下，StatefulSet Controller会保证集群内同一namespace下不会出现多个相同network identity的StatefulSet Pods。
- 如果集群内出现以上情况，那么有可能导致该有状态应用不能正常工作、甚至出现数据丢失等致命问题。

那么什么情况下会导致出现同一namespace下会出现多个相同network identity的StatefulSet Pods呢？我们考虑下Node出现网络Unreachable的情况：

- 如果你使用Kubernetes 1.5之前的版本，当Node Condition是NetworkUnavailable时，node controller会强制从apiserver中删除这个Node上的这些pods对象，这时StatefulSet Controller就会自动在其他Ready Nodes上recreate同identity的Pods。这样做其实风险是很大的，可能会导致有一段时间有多个相同network identity的StatefulSet Pods，可能会导致该有状态应用不能正常工作。所以尽量不要在Kubernetes 1.5之前的版本中使用StatefulSet，或者你明确知道这个风险并且无视它。
- 如果你使用Kubernetes 1.5+的版本，当Node Condition是NetworkUnavailable时，node controller不会强制从apiserver中删除这个Node上的这些pods对象，这些pods的state在apiserver中被标记为 `Terminating` 或者 `Unknown`，因此StatefulSet Controller并不会在其他Node上再recreate同identity的Pods。当你确定了这个Node上的StatefulSet Pods shutdown或者无法和该StatefulSet的其他Pods网络不同时，接下来就需要强制删除apiserver中这些 unreachable pods object，然后StatefulSet Controller就能在其他Ready Nodes上recreate同identity的Pods，使得StatefulSet继续健康工作。

那么在Kubernetes 1.5+中，如何强制从apiserver中删除该StatefulSet pods呢？有如下三种方法：

- 如果Node永久的无法连接网络或者关机了，意味着能确定这个Node上的Pods无法与其他Pods通信了，不会对StatefulSet应用的可用性造成影响，那么建议手动从apiserver中删除该NetworkUnavailable的Node，Kubernetes会自动从apiserver中删除它上面的Pods object。
- 如果Node是因为集群网络脑裂导致的，则建议去检查网络问题并成功恢复，因为Pods state已经是 `Terminating` 或者 `Unknown`，所以kubelet从apiserver中获取到这个信息后就会自动删除这些Pods。
- 其他情况才考虑直接手动从apiserver中删除这些Pods，因为这时你无法确定对应的Pods是否已经shutdown或者对StatefulSet应用无影响，强制删除后就可能导致出现同一namespace下有多个相同network identity的StatefulSet Pods，所以尽量不要使用这种方法。

- `kubectl delete pods <pod> --grace-period=0 --force`

小知识：当前Node Condition有以下6种：

Node Condition	Description
OutOfDisk	True if there is insufficient free space on the node for adding new pods, otherwise False
Ready	True if the node is healthy and ready to accept pods, False if the node is not healthy and is not accepting pods, and Unknown if the node controller has not heard from the node in the last 40 seconds
MemoryPressure	True if pressure exists on the node memory – that is, if the node memory is low; otherwise False
DiskPressure	True if pressure exists on the disk size – that is, if the disk capacity is low; otherwise False
NetworkUnavailable	True if the network for the node is not correctly configured, otherwise False
ConfigOK	True if the kubelet is correctly configured, otherwise False

## StatefulSet的Pod管理策略

Kubernetes 1.7+, StatefulSet开始支持Pod Management Policy配置，提供以下两种配置：

- **OrderedReady**，StatefulSet的Pod默认管理策略，就是逐个的、顺序的进行部署、删除、伸缩，也是默认的策略。
- **Parallel**，支持并行创建或者删除同一个StatefulSet下面的所有Pods，并不会逐个的、顺序的等待前一个操作确保成功后才进行下一个Pod的处理。其实用这种管理策略的场景非常少。

## StatefulSet的更新策略

StatefulSet的更新策略（由 `.spec.updateStrategy.type` 指定）支持以下两种：

- **OnDelete**，含义同Deployment的OnDelete策略，大家应该很熟悉了，不多介绍。

- **RollingUpdate**，滚动更新过程也跟Deployment大致相同，区别在于：
  - 相当于Deployment的maxSurge=0, maxUnavailable=1（其实StatefulSet是不存在这两个配置的）
  - 滚动更新的过程是有序的（逆序），index从N-1到0逐个依次进行，并且下一个Pod创建必须是前一个Pod Ready为前提，下一个Pod删除必须是前一个Pod shutdown并完全删除为前提。
  - 支持部分实例滚动更新，部分不更新，通过`.spec.updateStrategy.rollingUpdate.partition`来指定一个index分界点。
    - 所有ordinal大于等于partition指定的值的Pods将会进行滚动更新。
    - 所有ordinal小于partition指定的值得Pods将保持不变。即使这些Pods被recreate，也会按照原来的pod template创建，并不会更新到最新的版本。
    - 特殊地，如果partition的值大于StatefulSet的期望副本数N，那么将不会触发任何Pods的滚动更新。

思考：StatefulSet滚动更新时，如果某个Pod更新失败，会怎么办呢？

先卖个关子，下一篇对StatefulSet Controller源码分析时我们再来回答。

© 著作权归作者所有

¥ 打赏

👍 点赞 (1)

☆ 收藏 (5)

➦ 分享

🚩 举报

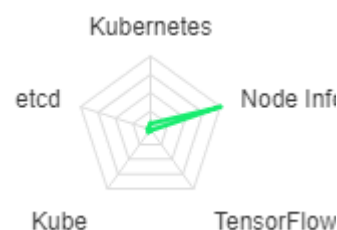


**WaltonWang**

粉丝 166 博文 92 码字总数 187622 作品 0

📍 深圳 🏢 程序员

♡ 关注



相关文章

最新文章

## kubernetes之StatefulSet详解

概述 RC、Deployment、DaemonSetStatefulSet都是面向无状态的服务，它们所管理的Pod的IP、名字，启

停顺序等都是随机的，而StatefulSet是什么？顾名思义，有状态的集合，管理所有有状态的服务...

Flywithmeto 07/10 0 0

## Kubernetes 1.5：支撑生产环境工作负载

如果你一直等待在Kubernetes中尝试运行某分布式数据库，或者在寻找能让有状态和无状态应用的应用中断SLO（服务等级指标）得到保障的途径，Kubernetes 1.5版本有你想要的解决方案。Stateful...

有容云 2016/12/28 15 0

## 2017年最后一次更新，Kubernetes 1.9发布！

【IT168 技术】根据CNCF的最新调查，有 61%的机构正在评估Kubernetes，83%的机构正在使用Kubernetes进行生产。世界上最大的长途拼车社区BlaBlaCar上拥有来自22个...

it168网站 2017/12/25 0 0



## 为什么Kubernetes Operator是游戏规则的改变者？

整个Web开发社区都在为Kubernetes（K8s）而沸腾。毫无疑问，去年的大会和开发人员集会上这都是最热的话题。它并非仅仅是管理容器的工具，实际上，Kubernetes允许用户轻...

Docker 04/11 0 0



崔婧雯

高级软件工程师，DockOne社区金牌

## Kubernetes 1.9.10 和 1.9.11-beta.0 版本发布

Kubernetes 1.9.10 和 1.9.11-beta.0 版本发布了。Kubernetes 是一个开源的，用于管理云平台中多个主机上的容器化的应用，Kubernetes 的目标是让部署容器化的应用简单并且高效（powerful），...

达尔文 08/04 0 0

加载更多

### 开源中国社区

关于我们  
联系我们  
合作伙伴  
Open API

### 在线工具

码云 Gitee.com  
在线工具  
Team@OSC 项目协作平台  
RunJS 在线开发

### 微信公众号



### 开源中国 APP

聚合全网技术文章，根据你的阅读喜好进行个性推荐

下载 APP