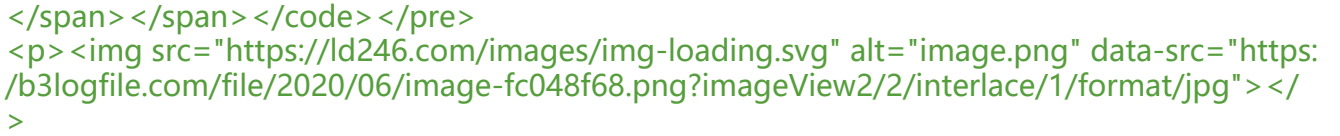
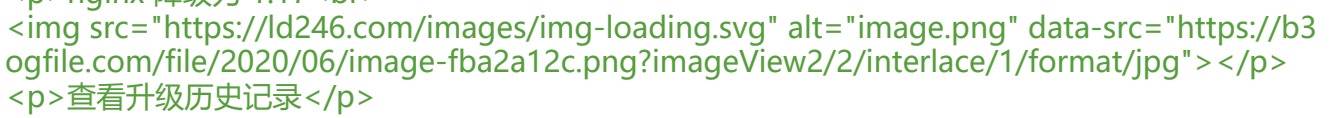
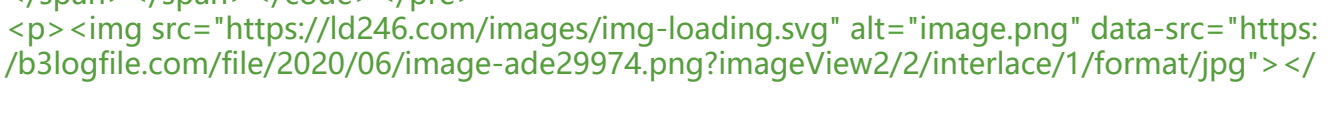
kubernetes 应用升级、弹性伸缩、回滚、删除

作者：Leif160519
原文链接：https://ld246.com/article/1591854978052
来源网站：链滴

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/bing/20200319.jpg?imageView2/1/w/960/h/540/interlace/1/q/100">
<h2 id="一-准备工作">一、准备工作</h2>
创建一个 deployment，使用 nginx 镜像，版本为 1.16 并暴露应用
<pre><code class="highlight-chroma"><span class="highlight
cl">kubectl create deployment web --image=nginx:1.16
kubectl expose d
ployment web --port=80 --target-port=80 --type=NodePort
</code></pre>
查看应用暴露的端口：
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@k8s-master ~]# kubectl get svc
NAME   TYPE
    CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
web   NodePo
t   10.107.162.235   <none>   80:30892/TCP   21s
</code></pre>
在浏览器中检查 nginx 版本：
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:
/b3logfile.com/file/2020/06/image-ade29974.png?imageView2/2/interlace/1/format/jpg"></
>
<h2 id="二-应用升级">二、应用升级</h2>
将 nginx 版本升级到 1.17
<pre><code class="highlight-chroma"><span class="highlight
cl">kubectl set image deployment web nginx=nginx:1.17
</code></pre>
参数解释：

<code>nginx=nginx:1.17</code> 中前者 <code>nginx</code> 代表容器名，后者代表镜
名称，容器名可以在 <code>yaml</code> 中查看

查看升级状态：
<pre><code class="highlight-chroma"><span class="highlight
cl">kubectl rollout status deployment web
</code></pre>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:
/b3logfile.com/file/2020/06/image-fba2a12c.png?imageView2/2/interlace/1/format/jpg"></
>
<h2 id="三-应用回滚">三、应用回滚</h2>
查看应用部署历史记录(默认保留 <code>10</code> 条，也可在 <code>yaml</code> 中查
设置，与副本集 <code>replicas</code> 参数同级，名称为 <code>revisionHistoryLimit</code
):
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@k8s-master ~]# kubectl rollout history deployment
web deployment.
pps/web
REVISION CHAN
E-CAUSE
1   <none&amp
t;
2   <none&amp
t;
</code></pre>
其中，版本 1 代表之前部署的 1.16 版本，版本 2 为 1.17 版本，但是这里的变更记录里面却什

都没有，我们可以使用 <code>--record=true</code> 参数来记录每次升级的详细记录.
将 nginx 升级到 1.18：
<pre><code class="highlight-chroma"><span class="highlight
cl">kubectl set image deployment web nginx=nginx:1.18 --record=true
</code></pre>

/b3logfile.com/file/2020/06/image-fc048f68.png?imageView2/2/interlace/1/format/jpg"></
>
查看更新历史记录：
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@k8s-master ~]# kubectl rollout history deployment web
deployment.apps
web
REVISION CHAN
E-CAUSE
1   <none&
t;
2   <none&
t;
3   kubectl set
mage deployment web nginx=nginx:1.18 --record=true
</code></pre>
添加 <code>--record=true</code> 命令后会将升级的命令记录下来。
<h3 id="3-1-回滚到上一个版本">3.1 回滚到上一个版本</h3>
<pre><code class="highlight-chroma"><span class="highlight
cl">kubectl rollout undo deployment web
</code></pre>
nginx 降级为 1.17

ogfile.com/file/2020/06/image-fba2a12c.png?imageView2/2/interlace/1/format/jpg">
查看升级历史记录
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@k8s-master ~]# kubectl rollout history deployment web
deployment.apps
web
REVISION CHAN
E-CAUSE
1   <none&
t;
3   kubectl set
mage deployment web nginx=nginx:1.18 --record=true
4   <none&
t;
</code></pre>
由此可见，每次的升级都会将之前的版本号重命名为新的版本号，所以版本 1 是 1.16 ，版本 3 是
1.18 ，版本 4 是 1.17
<h3 id="3-2-回滚到指定版本">3.2 回滚到指定版本</h3>
若继续回滚到上个版本，那也只能回滚到 18 版本，若想回滚到 1.16.则需要指定版本号进行回
操作：
<pre><code class="highlight-chroma"><span class="highlight
cl">kubectl rollout undo deployment web --to-revision=1
</code></pre>

/b3logfile.com/file/2020/06/image-ade29974.png?imageView2/2/interlace/1/format/jpg"></

>
查看更新历史记录：
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@k8s-master ~]# kubectl rollout history deployment web
deployment.apps
web
REVISION CHAN
E-CAUSE
3   kubectl set
mage deployment web nginx=nginx:1.18 --record=true
4   <none&
t;
5   <none&
t;
</code></pre>
其实，deployment 利用 rs 做副本管理、回滚，kubelet 会将应用执行一个滚动更新的操作，
体流程如下：
假设 web 副本集为 3，则

1.创建新的 rs 并且 scale up=1
2.将旧的 rs 缩容 scale down=2
3.将新的 rs scale up=2
4.将旧的 rs 缩容 scale down=1
5.将新的 rs 扩容 scale up=3
6.将旧的 rs 缩容 scale down=0 （但并不会删除 rs）

<pre><code class="highlight-chroma"><span class="highlight
cl">[root@k8s-master ~]# kubectl get rs
NAME
  DESIRED   CURRENT   READY   AGE
web-5c987fcb9f
        1   1   1   63m
web-669588cb76
         0   0   0   57m
web-74b9fcbbc7
        0   0   0   27m
</code></pre>
<h2 id="四-应用扩容">四、应用扩容</h2>
将 web 应用扩容副本集为 3
<pre><code class="highlight-chroma"><span class="highlight
cl">kubectl scale deployment web --replicas=3
</code></pre>
查看副本集：
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@k8s-master ~]# kubectl get rs
NAME
DESIRED   CURRENT   READY   AGE
web-5c987fcb9f
  3   3   3   89m
web-669588cb76
    0   0   0   83m
web-74b9fcbbc7
    0   0   0   53m
</code></pre>

<h3 id="4-1-弹性伸缩">4.1 弹性伸缩</h3>
设置 web 应用副本数为 3-10
<pre><code class="highlight-chroma"><span class="highlight
cl">kubectl autoscale deployment web --min=3 --max=10
</code></pre>
查看 hpa
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@k8s-master ~]# kubectl get hpa
NAME   REFER
NCE   TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
web   Deployment
web   <unknown>/80%   3   10   3   20s
</code></pre>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:
/b3logfile.com/file/2020/06/image-bf1d0ac0.png?imageView2/2/interlace/1/format/jpg"></
>
可以看到，获取的资源前面显示 unknown，且时间信息里出现了 <code>failed to get cpu util
zation:missing request for cpu</code>,这是因为我们创建的 pod 对象没有添加 request 资源声
，这样导致 hpa 读取不到 cpu 指标信息，所以要想让 hpa 生效，则对应的 pod 资源中必须添加 req
ests 资源声明:
在线编辑资源（会使用默认编辑器打开）：
<pre><code class="highlight-chroma"><span class="highlight
cl">kubectl edit deployment web
</code></pre>
利用查找功能，找到 resources，修改成以下内容(设置资源占用的 cpu 和内存大小的上限和下限
：
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:
/b3logfile.com/file/2020/06/image-c4ff1d04.png?imageView2/2/interlace/1/format/jpg"></
>
<pre><code class="highlight-chroma"><span class="highlight
cl">   resources:
          requests:
            cpu: 100m
            memory: 5
0Mi
          limits:
            cpu: 200m
            memory: 5
0Mi
</code></pre>
修改完成保存后触发滚动更新，之后继续查看 hpa：
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@k8s-master ~]# kubectl get hpa
NAME   REFER
NCE   TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
web   Deploy
ent/web   0%/80%   3   10   3   6m25s
</code></pre>
当 cpu 占比超过 80% 时自动扩容，当低于 80% 时自动缩容。
<blockquote>
注意：

<code>hpa</code>(<code>horizontal pod autoscaler</code>：Pod 水平伸缩)已经将内
指标禁用，原因是 cpu 分配的资源可控，内存不可控。

弹性伸缩依靠某些硬件指标去进行控制，故依赖 <code>metrics-server</code>。
默认 cpu 占比为 80%，若设置其他值，则需要添加 <code>--cpu-percent=<占比></code> 参数
hpa 的冷却时间为，扩容 3 分钟，缩容 5 分钟

</blockquote>
<h2 id="五-应用删除">五、应用删除</h2>
rs 会周期性检查当前 pod 是否按照预设的副本集去运行，若副本集少于预设值，则会自动拉起 pd，若副本集多余预设值，则删除多余的 pod，若想彻底删除 pod，则只需要删除 deployment 即：
<pre><code class="highlight-chroma">kubectl delete deployment web
kubectl delete svc web
</code></pre>