

《每天 5 分钟玩转 Kubernetes》读书笔记

作者: [Leif160519](#)

原文链接: <https://ld246.com/article/1569316185143>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>

</p>
<p>下载链接: https://leif.fun/downloads/ooks/k8s/每天 5 分钟玩转 Kubernetes.pdf</p>
<p>百度云: 链接:https://pan.ba du.com/s/1eUJzQVGFuMXM3SwGjcFw 密码:47bp</p>
<h2 id="2019年09月24日">2019 年 09 月 24 日</h2>
<h2 id="第1章-先把Kubernetes跑起来">第 1 章 先把 Kubernetes 跑起来</h2>
<h3 id="1-2-创建kubernetes集群">1.2 创建 kubernetes 集群</h3>
<p>查看集群信息</p>
<pre><code class="highlight-chroma">kubectl cluster-info
</code></pre>
<p></>
<h3 id="1-3-部署应用">1.3 部署应用</h3>
<p>Deployment:可以理解为应用
Pod:是容器的集合, 通常会将紧密相关的一组容器放到一个 pod 中, 同一个 pod 中的所有容器共享 I 地址和 Port 空间, 也就是说他们在一个 network namespace 中.
是 K8S 调度的最小单位, 同一个 pod 中的容器始终被一起调度.</p>
<p>查看当前 pod</p>
<pre><code class="highlight-chroma">kubectl get pods
</code></pre>
<p></>
<h3 id="1-4-访问应用">1.4 访问应用</h3>
<p>查看应用被映射到节点的哪个端口</p>
<pre><code class="highlight-chroma">kubectl get services
</code></pre>
<p></>
<p>service: 暂时理解为端口映射</p>
<h3 id="1-5-scale应用">1.5 scale 应用</h3>
<p>查看副本数</p>
<pre><code class="highlight-chroma">kubectl get deployments
</code></pre>
<p></>
<p>增加副本数 (扩容) </p>
<pre><code class="highlight-chroma">kubectl scale deployments/kubernetes-bootcamp --replicas=3
</code></pre>
<p></>

>

<p>通过 <code>kubectl get pods</code> 可以看到当前 Pod 增加到了三个

 </p>

<p>使用 <code>curl</code> 命令访问应用可以看到负载均衡的效果

 </p>

<p>删除一个副本（扩容） </p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">kubectl scale deployments/kubernetes-bootcamp --replicas=2</span></span></code></pre>
```

<p> </p>

>

<p>将 v1 升级到 v2</p> ``` <code class="highlight-chroma">kubectl set image deployments/kubernetes-bootcamp kubernetes-bootcamp=jocatlin/kubernetes-bootcamp:v2</code></pre> ``` <p> </p> > <p>可以看出 v1 的 pod 被删除，同时启用了新的 v2 pod
 </p> <p>将 v2 降级到 v1</p> ``` <code class="highlight-chroma">kubectl rollout undo deployments/kubernetes-bootcamp</code></pre> ``` <p> </p> > <p> </p> > <p>1.Cluster: 是计算、存储和网络资源的集合</p> <p>2.Master: 是 Cluster 的大脑，主要负责调度，可以运行多个 Master 来实现高可用</p> <p>3.Node:职责是运行容器应用。Node 由 Master 管理，Node 负责监控并汇报容器的状态，同根据 Master 的要求管理容器的生命周期。如果 Cluster 只有一个主机，那么它既是 Master 也是 Node。 </p> <p>4.Pod:是 K8S 的最小工作单元。每个 Pod 包含一个或多个容器。Pod 中的容器会作为一个整体由 Master 调度到一个 Node 上运行。 </p> <p>5.Controller:K8S 通过 Controller 来管理 Pod。Controller 中定义了 Pod 的部署特性，比如有多少个副本、在什么样的 Node 上运行等。K8S 提供了多种 Controller: </p> <p>Deployment: 可以管理 Pod 的多个副本并确保 Pod 按照期望的状态运行</p><p>ReplicaSet: 实现了 Pod 的多副本管理。使用 Deployment 会自动创建 ReplicaSet，Deployment 通过 ReplicaSet 来管理 Pod 的多个副本，一般不直接使用 ReplicaSet。 </p>

<p>DaemonSet: 用于每隔 Node 最多只运行一个 Pod 副本的场景。DaemonSet 通常用于运行 da
mon。 </p>

<p>StatefulSet: 能够保证 Pod 的每个副本在整个生命周期中是不变的, 而其他 Controller 不提
这个功能。当某个 Pod 发生故障需要删除并重新启动时, Pod 的名称会发生变化, 同时 StatefulSet
会保证副本按照固定顺序启动、更新或者删除。 </p>

<p>Job: 用于运行结束就删除的应用, 而其他 Controller 中的 Pod 通常是长期持续运行。 </p>

<p>6.Service: K8S Service 定义了外界访问一组特定 Pod 的方式。Service 有自己的 IP 和端口, Se
vice 为 Pod 提供了负载均衡。K8S 运行容器 (Pod) 与访问容器 (Pod) 这两个任分别由 Controller
和 Service 执行</p>

<p>7.Namespace: 可以将物理的 Cluster 逻辑上划分成多个虚拟 Cluster, 每个 Cluster 就是一个
amespace、不同的 Namespace 里的资源是完全隔离的。K8S 默认创建了两个 Namespace。 </p>

<p><
p>

<p>default: 创建资源时如果不指定, 将被放到这个 Namespace 中</p>

<p>kube-system: K8S 自己创建的系统资源将放到这个 Namespace 中。 </p>

