



链滴

kubeadmin 部署完整 K8s-1.15.3 版本的集群的过程

作者: [IwasawaMasami](#)

原文链接: <https://ld246.com/article/1579074241748>

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>

<h3 id="准备工作">准备工作</h3>

<p>实验平台: Vmware pro 15

操作系统镜像: CentOS-7-x86_64-DVD-1804</p>

<p>1.通过虚拟机创建一个模板主机默认最小化安装, 配置如下: </p>

<blockquote>

<p>处理器内核总数: 2 个

内存: 4GB

硬盘大小: 40GB ,60GB

网络模式: NAT 模式, 仅主机模式</p>

</blockquote>

<p></p>

<p>这里有一个配置表可供参考:

| 角色 | CPU 数量 | 内存大小 | 主机名 | 磁盘 |

| master | >=2c | >=2G | master | sda、sdb |

| worker | >=2c | >=2G | node1 | sda、sdb |

| worker | >=2c | >=2G | node2 | sda、sdb |</p>

<p>2.完成安装后, 安装基础软件包</p>

<blockquote>

<p>yum -y install net-tools wget vim</p>

</blockquote>

<p>3.从模板主机克隆三个操作系统, 并分别命名为 Master, node-1, node-2,连接上 ssh 终端。 <p>

<p></p>

<h3 id="准备部署">准备部署</h3>

<h4 id="在三个节点上分别做以下几个步骤-">在三个节点上分别做以下几个步骤: </h4>

<p>1.先分别更改节点的 hostname 为 Master, node-1, node-2</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> hostnamectl set-hostname master
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> hostnamectl set-ostname node-1
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> hostnamectl set-ostname node-2
```

```
</span></span></code></pre>
```

<p>2.编辑节点机的 /etc/hosts 文件, 添加域名解析</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> cat &lt;&lt;EOF &gt;&gt;/etc/hosts
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> 172.22.8.141 mast
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> 172.22.8.142 nod
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> 172.22.8.143 nod
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> EOF
```

```
</span></span></code></pre>
```

<p>3.关闭节点防火墙</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> systemctl stop firewalld
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> systemctl disable f
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> setenforce 0
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">sed -i "s/^SELINU
=enforcing/SELINUX=disabled/g" /etc/selinux/config
</span></span></code></pre>
<p>4.关闭节点交换分区</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">swapoff -a
</span></span><span class="highlight-line"><span class="highlight-cl">sed -i 's/.*/.*/
&amp;/' /etc/fstab
</span></span></code></pre>
<p>5.开启节点 ipv4 转发</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">cat &gt; /etc/sysctl.d/k8s.conf &lt;&lt;EOF
</span></span><span class="highlight-line"><span class="highlight-cl">net.bridge.bridge
nf-call-ip6tables = 1
</span></span><span class="highlight-line"><span class="highlight-cl">net.bridge.bridge
nf-call-iptables = 1
</span></span><span class="highlight-line"><span class="highlight-cl">EOF
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">sysctl --system
</span></span></code></pre>
<p>6.配置节点国内 yum 源</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">yum install -y wget
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">wget -O /etc/yum.
epos.d/CentOS-Base.repo http://mirrors.cloud.tencent.com/repo/centos7_base.repo
</span></span><span class="highlight-line"><span class="highlight-cl">wget -O /etc/yum.
epos.d/epel.repo http://mirrors.cloud.tencent.com/repo/epel-7.repo
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">yum clean all &a
p;&amp; yum makecache
</span></span></code></pre>
<p>7.配置节点国内 Kubernetes 源</p>
<p><strong>请注意这一步很重要，很多时候 kubeadmin 无法 init 就是因为 GFW 的原因，所以
我们配置完下面 Kubernetes 源之后，kubeadmin 就可以从国内 pull 镜像了。</strong></p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">cat &lt;&lt;EOF &gt; /etc/yum.repos.d/kubernetes.repo
</span></span><span class="highlight-line"><span class="highlight-cl">[kubernetes]
</span></span><span class="highlight-line"><span class="highlight-cl">name=Kubernetes
</span></span><span class="highlight-line"><span class="highlight-cl">baseurl=https://m
rrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86_64/
</span></span><span class="highlight-line"><span class="highlight-cl">enabled=1
</span></span><span class="highlight-line"><span class="highlight-cl">gpgcheck=1
</span></span><span class="highlight-line"><span class="highlight-cl">repo_gpgcheck=1
</span></span><span class="highlight-line"><span class="highlight-cl">gpgkey=https://m
rrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg https://mirrors.aliyun.com/kubernetes/y
m/doc/rpm-package-key.gpg
</span></span><span class="highlight-line"><span class="highlight-cl">EOF
</span></span></code></pre>
<p>8.配置节点 docker 源</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">wget https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo -O /etc/yum.repo
.d/docker-ce.repo
</span></span></code></pre>
```

<h3 id="开始部署">开始部署</h3>

<h4 id="在三个节点上进行以下几个步骤-">在三个节点上进行以下几个步骤: </h4>

<p>1.安装并启动节点的 docker</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">yum install -y docker-ce-18.06.1.ce-3.el7
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">systemctl enable
```

```
</span></span></code></pre>
```

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">检测docker是否成功安装, 出现下面显示就说明成功安装了
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">docker --version
```

```
</span></span></code></pre>
```

<blockquote>

<p>Docker version 18.06.1-ce, build e68fc7a</p>

</blockquote>

<p>2.安装并启动节点的 kubeadm、kubectl 工具</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">yum install -y kubelet kubeadm kubectl
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">systemctl enable
```

```
</span></span></code></pre>
```

<blockquote>

<p>Kubelet 负责与其他节点集群通信, 并进行本节点 Pod 和容器生命周期的管理。

Kubeadm 是 Kubernetes 的自动化部

工具, 降低了部署难度, 提高效率。

Kubectl 是 Kubernetes 集群管理工具。</p>

</blockquote>

<h3 id="Kubeadmin在master节点进行Kubernetes集群初始化">Kubeadmin 在 master 节点进行 Kubernetes 集群初始化</h3>

<p>:heart: 只在 master 节点上进行! </p>

<p>请注意将 api-server 的地址改成你的 master 节点的地址</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">kubeadm init --kubernetes-version=1.15.3 \
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">--apiserver-advert
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">se-address=172.22.8.141 \
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">--image-repositor
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">registry.aliyuncs.com/google_containers \
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">--service-cidr=10.
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">.0.0/16 \
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">--pod-network-ci
```

```
</span></span></code></pre>
```

<p>3.当你的集群初始化成功后返回如下信息:

</p>

<p>先别急着高兴, 把最后生成部分内容记录下来, 此内容需要在其它节点加入 Kubernetes 集群时

行, 如下: </p>

<p>4.配置 kubectl 工具</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">mkdir -p /root/.kube
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">cp /etc/kubernet
/admin.conf /root/.kube/config
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">查看主节点状态是
OK:
</span></span><span class="highlight-line"><span class="highlight-cl">kubectl get nodes
</span></span><span class="highlight-line"><span class="highlight-cl">kubectl get cs
</span></span></code></pre>
<p>5.将其他两个节点加入集群,分别在另外两个节点上使用刚刚在最后部分记录下的命令: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">kubeadm join 172.22.8.141:6443 --token o5weqe.nwmt7dbn8jghuvkx \
</span></span><span class="highlight-line"><span class="highlight-cl"> --discovery-tok
n-ca-cert-hash sha256:c872d1c4b4919f0dcd00fbd2efcfd6564593342be9716eea12e31e14c9
3e6b
</span></span></code></pre>
<p>6.部署节点的 flannel 网络 (其实还可以使用 <a href="https://ld246.com/forward?goto=htt
s%3A%2F%2Fwww.projectcalico.org%2F" target="_blank" rel="nofollow ugc">calico</a> 网
进行安装,这里就不做演示了) </p>
<p>当我们成功将两个节点加入后,还应该选择一种网络插件对集群进行安装: <br>
使用 <code>kubectl</code> 工具可以让我们快速部署 flannel 网络</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/a70459be0084506e4e
919aa1c114638878db11b/Documentation/kube-flannel.yml
</span></span></code></pre>
<p>验证 flannel 是否安装成功,在 master 上使用 <code>kubectl get pods -n kube-system -o wi
e</code> 命令</p>
<p>只要出现三个 flannelpod 分别在三个节点上处于 running 状态,像这样:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">kube-flannel-ds-amd64-6jmcm    1/1    Running    0      88s    172.22.8.143    node-2
</span></span><span class="highlight-line"><span class="highlight-cl">kube-flannel-ds-
md64-89rrf    1/1    Running    0      88s    172.22.8.141    master
</span></span><span class="highlight-line"><span class="highlight-cl">kube-flannel-ds-
md64-fqqqt    1/1    Running    0      88s    172.22.8.142    node-1
</span></span></code></pre>
<p>以及 coredns 也处于 running 状态: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">coredns-bccdc95cf-ghwp6      1/1    Running    0      12m    10.244.0.2     master
</span></span><span class="highlight-line"><span class="highlight-cl">coredns-bccdc95c
-glhn2      1/1    Running    0      12m    10.244.0.3     master
</span></span></code></pre>
<p>否则就会出现诸如此类的错误: <br>
使用 <code>kubectl get nodes</code> 出现节点 <code>NotReady</code> 状态: <br>
</p>
<p>使用命令 <code>kubectl get pods -n kube-system -o wide</code> 出现 coredns pending
的状态: <br>
</p>
<h4 id="如果这些都没有问题的话-那么k8s的基础部署就已经成功部署了-">如果这些都没有问题的
, 那么 k8s 的基础部署就已经成功部署了。</h4>
<h3 id="-可选做-部署Dashboard-图形化界面-">【可选做】部署 Dashboard (图形化界面) </h3>
```

<p>还是要在 master 节点上做以下几个步骤：

1.创建下载 Dashboard 的 yaml 文件</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">wget https://raw.githubusercontent.com/kubernetes/dashboard/v1.10.1/src/deploy/recommended/kubernetes-dashboard.yaml</span></span></code></pre>
```

<p>2.使用如下命令直接编辑 kubernetes-dashboard.yaml 文件</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">sed -i 's/k8s.gcr.io/loveone/g' kubernetes-dashboard.yaml</span></span><span class="highlight-line"><span class="highlight-cl">sed -i '/targetPort: a\\\\\\\\\\\\ nodePort: 30001\\n\\\\ type: NodePort' kubernetes-dashboard.yaml</span></span></code></pre>
```

<p>3.部署 Dashboard</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">kubectl create -f kubernetes-dashboard.yaml</span></span></code></pre>
```

<p>4.创建完成后，检查相关服务运行状态</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">kubectl get deployment kubernetes-dashboard -n kube-system</span></span><span class="highlight-line"><span class="highlight-cl">kubectl get pods n kube-system -o wide</span></span><span class="highlight-line"><span class="highlight-cl">kubectl get services -n kube-system</span></span><span class="highlight-line"><span class="highlight-cl">netstat -ntlp|grep 30001</span></span></code></pre>
```

<p>5.创建 Dashboard 的认证令牌</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">kubectl create serviceaccount dashboard-admin -n kube-system</span></span><span class="highlight-line"><span class="highlight-cl">kubectl create clusterrolebinding dashboard-admin --clusterrole=cluster-admin --serviceaccount=kube-system:dashboard-admin</span></span></code></pre>
```

<p>6.查看 Dashboard 的认证令牌:</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">kubectl describe secrets -n kube-system $(kubectl -n kube-system get secret | awk '/dashboard-admin/{print $1}')</span></span></code></pre>
```

<p>打印出来的应该是这个样子：

</p>

<p>7.然后你就可以在 Firefox 浏览器（别用 google）输入 Dashboard 访问地址： https://xxx.xxx.xxx.xxx :

</p>

<p>然后我们就可以在 dashboard 里面以图形化的方式监测各个数据的变化。 </p>

<p>节点的工作状态：

</p>

<p>资源空间的状态：

</p>

<p>pod 的状态：

 dashboard 里通过 yaml 文件部署



结语:

这就是整个k8s的部署过程-当然这只是一个开始-我还在学习的路上-关于k8s后面还有好多多的知识等待被发掘->这就是整个 k8s 的部署过程, 当然这只是一个开始, 我还在学习的路上, 关于 k8s 后面还有好多好多的知识等待被发掘。

部署的参考链接是: [kubernetes 中文区](https://ld246.com/forward?goto=https%3A%2F%2Fwww.kubernetes.org.cn%2F5462.html)