

9.使用Minikube搭建K8s环境

1.什么是Minikube

2.安装Docker

3.安装minikube

4.启动Minikube

5.安装kubectl

上一节我们了解了如何发布镜像到镜像仓库，这一节来学习如何使用Minikube配置一个Kubernetes集群

在将应用程序打包到镜像里并且将镜像上传到镜像仓库后，我们就可以将应用部署到k8s集群里了。

在部署之前，需要先配置好k8s集群。配置一个完整的、多节点的k8s集群还是比较复杂的，特别是在不熟悉Linux和网络管理的情况下。

正常安装k8s需要跨多个物理机或者虚拟机，并且需要正确地设置网络，以便k8s集群中运行的所有容器能够在一个扁平的网络内互相通信。

本文先介绍如何使用Minikube来搭建一个Kubernetes单节点环境。

1.什么是Minikube

Minikube提供了一种非常简单快捷的方式来安装单节点Kubernetes环境。通过它，我们可以测试k8s的大部分功能、可以在本地开发应用程序、可以构建一个本地的沙箱环境。但是不能将其用于生产环境，因为它并不能完整地提供K8s的功能，比如不支持扩容也不能共享等。

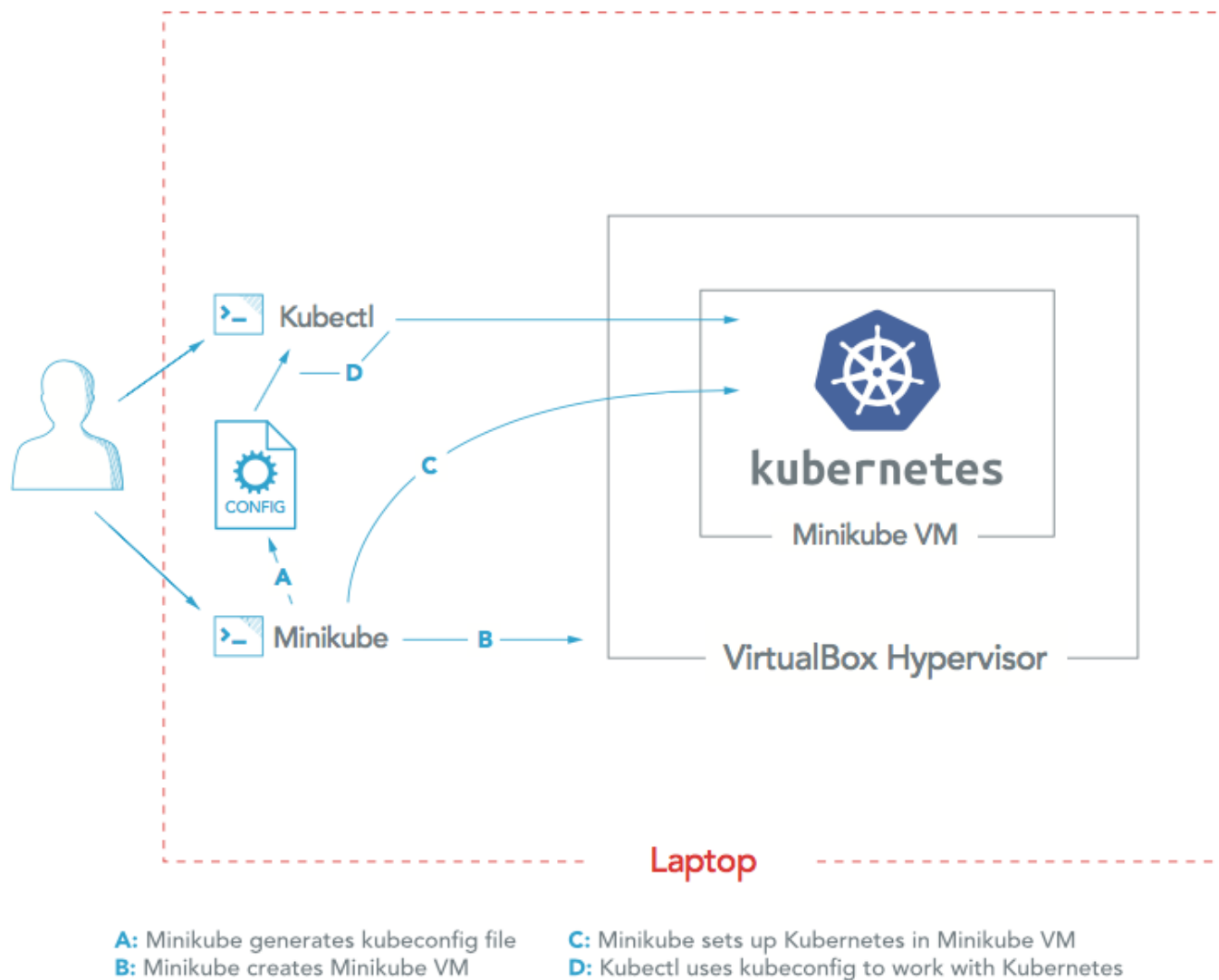
Minikube支持的Kubernetes功能如下：

- DNS
- NodePorts（可使用“minikube service”命令来管理）
- ConfigMaps和Secrets
- 仪表板（Dashboards, minikube dashboard）
- 容器运行时：Docker、CRI-Orkt、containerd
- Enabling CNI（容器网络接口）
- Ingress
- LoadBalancer（负载均衡，可以使用“minikube tunnel”命令来启用）
- Multi-cluster（多集群，可以使用“minikube start -p <name>”命令来启用）

- Persistent Volumes
- RBAC
- 通过命令配置apiserver和kubele

一般情况下，开发人员都是在自己的手提电脑上使用Minikube来第一次接触Kubernetes的。通过使用Minikube，一个单节点的k8s环境（不能称之为集群）就以VM的形式部署到本地了。Minikube支持大多数操作系统（Linux、Windows和OSX），支持多种虚拟化技术（Virtualbox、KVM、xhyve、VMware Fusion、Hyper-V）。

下图展示了在单个主机上设置Minikube的一些细节：



2.安装Docker

这个在之前发布的《6.运行一个简单的容器》文章中讲解过，此处忽略。

3.安装minikube

本文中我们使用阿里云提供的最新版的minikube。

curl -Lo minikube <https://kubernetes.oss-cn-hangzhou.aliyuncs.com/minikube/releases/v1.13.0/minikube-linux-amd64>

```
[root@dhr-demo ~]# curl -Lo minikube https://kubernetes.oss-cn-hangzhou.aliyuncs.com/minikube/releases/v1.13.0/minikube-linux-amd64
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100 52.6M  100 52.6M    0     0  11.1M      0  0:00:04  0:00:04 --:--:-- 11.1M
[root@dhr-demo ~]#
```

执上面的命令下载minikube，会在~目录下生成一个没有执行权限的minikube文件

```
[root@dhr-demo ~]# ll
total 53964
-rw----- 1 root root 1371 Oct 17 2019 anaconda-ks.cfg
-rw-r--r-- 1 root root 55253742 Nov 6 00:27 minikube
drwxr-xr-x 2 root root 38 Nov 5 00:17 testui
[root@dhr-demo ~]#
```

添加执行权限并将其加到PATH路径下：

chmod +x minikube

sudo mv minikube /usr/local/bin

4.启动Minikube

创建docker组：

\$ sudo groupadd docker

添加新用户，并加到docker组中：

useradd david

passwd david

sudo usermod -aG docker david && newgrp docker

切换到新用户：

su david

启动minikube：

minikube start

```
[david@dhr-demo root]$ minikube start
* minikube v1.13.0 on Centos 7.8.2003
* Automatically selected the docker driver
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Creating docker container (CPUs=2, Memory=2900MB) ...
* Preparing Kubernetes v1.19.0 on Docker 19.03.8 ...
  > kubectrl.sha256: 65 B / 65 B [-----] 100.00% ? p/s 0s
  > kubelet.sha256: 65 B / 65 B [-----] 100.00% ? p/s 0s
  > kubeadm.sha256: 65 B / 65 B [-----] 100.00% ? p/s 0s
  > kubeadm: 37.30 MiB / 37.30 MiB [-----] 100.00% 16.81 MiB p/s 3s
  > kubectrl: 41.01 MiB / 41.01 MiB [-----] 100.00% 12.77 MiB p/s 3s
  > kubelet: 104.88 MiB / 104.88 MiB [-----] 100.00% 13.04 MiB p/s 9s
* Verifying Kubernetes components...
* Enabled addons: default-storageclass, storage-provisioner
* Done! kubectrl is now configured to use "minikube" by default
[david@dhr-demo root]$
```

5. 安装kubectl

kubectl是Kubernetes的命令行工具，通过它我们可以向Kubernetes集群执行命令，还可以部署应用、监管集群资源以及查看日志。

尽量使用最新版本的kubectl以避免一些莫名其妙的问题。

使用阿里云提供的版本：

```
curl -Lo kubectl http://kubernetes.oss-cn-hangzhou.aliyuncs.com/kubernetes-release/release/v1.19.0/bin/linux/amd64/kubectl
```

```
[root@dhr-demo ~]# curl -Lo kubectl http://kubernetes.oss-cn-hangzhou.aliyuncs.com/kubernetes-release/release/v1.19.0/bin/linux/amd64/kubectl
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 41.0M  100 41.0M    0     0  15.2M    0  0:00:02  0:00:02 --:--:-- 15.2M
```

添加执行权限并将kubectl加到PATH路径下：

```
chmod +x ./kubectl
```

```
sudo mv ./kubectl /usr/local/bin
```

使用kubectl查看k8s集群的状态：

```
kubectl cluster-info
```

```
[david@dhr-demo root]$ kubectl cluster-info
Kubernetes master is running at https://172.17.0.4:8443
kubeDNS is running at https://172.17.0.4:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
[david@dhr-demo root]$
```

可以看到k8s“集群”已经启动起来了。

运行minikube ssh命令进入到Minikube VM。然后使用top命令查看VM中运行的进程，可以看到很多k8s组件：

```
[david@dhr-demo root]$ minikube ssh
docker@minikube:~$ pwd
/home/docker
docker@minikube:~$ top
top - 08:26:53 up 340 days, 16 min, 0 users, load average: 0.67, 0.62, 0.46
Tasks: 38 total, 1 running, 37 sleeping, 0 stopped, 0 zombie
%Cpu(s): 4.6 us, 1.7 sy, 0.0 ni, 92.8 id, 0.2 wa, 0.0 hi, 0.2 si, 0.6 st
MiB Mem : 11995.7 total, 268.8 free, 6564.2 used, 5162.7 buff/cache
MiB Swap: 4096.0 total, 3995.7 free, 100.2 used, 5044.9 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME     COMMAND
 2268 root        20   0 1096256 336544 42256 S   9.0   2.7 75:30.4 kube-apiserver
 2681 root        20   0 1924224 75384 35076 S   8.3   0.6 63:27.0 kubelet
 2358 root        20   0 10.1g 60480 13120 S   3.3   0.5 24:07.3 etcd
 2265 root        20   0 811556 72596 31972 S   2.3   0.6 23:17.2 kube-controller
 382 root        20   0 1598872 79096 28420 S   1.7   0.6 17:59.7 dockerd
 2235 root        20   0 746900 33548 17268 S   1.0   0.3 3:16.8 kube-scheduler
 3203 root        20   0 747904 25808 15100 S   0.7   0.2 4:24.9 coredns
 146 root        20   0 2151944 41304 16160 S   0.3   0.3 1:23.7 containerd
 3308 root        20   0 734208 20536 10280 S   0.3   0.2 2:42.9 storage-provisi
 1 root        20   0 22096 8188 4800 S   0.0   0.1 0:01.9 systemd
 128 root       19  -1 29160 8064 6708 S   0.0   0.1 0:01.8 systemd-journal
 150 root        20   0 12152 3796 3008 S   0.0   0.0 0:00.0 sshd
 1985 root        20   0 110344 3732 2656 S   0.0   0.0 0:03.0 containerd-shim
```