**一、简介**

Minikube是单机版的Kubernetes集群，能够在我们本地的个人电脑（包含Windows、macOS、Linux系统）运行Kubernetes集群，让我们进行日常的开发工作

官方地址：<https://kubernetes.io/docs/tasks/tools/install-minikube/>

**二、安装步骤**

1、检查是否开启虚拟化，运行以下命令验证输出是否为非空

$ grep -E --color 'vmx|svm' /proc/cpuinfo

2、安装docker （可以参考之前博客）

3、安装kubectl

$ cat <<EOF > /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86\_64/

enabled=1

gpgcheck=1

repo\_gpgcheck=1

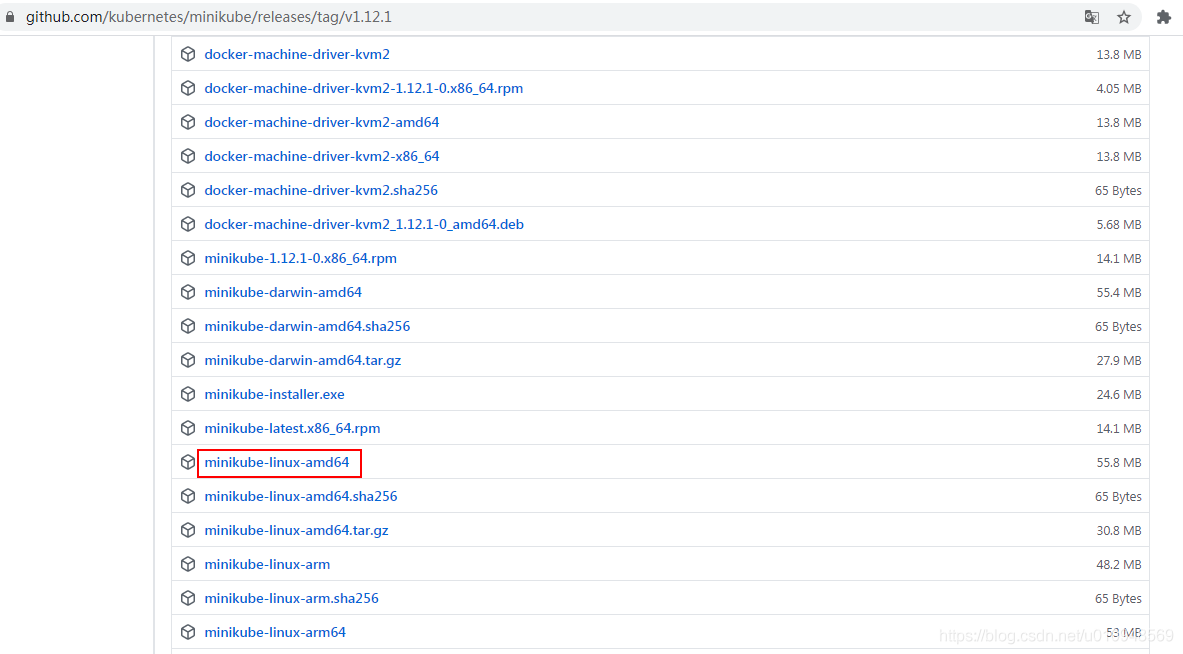
gpgkey=https://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg https://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg

EOF

$ yum install kubectl

4、安装minikube

下载地址：<https://github.com/kubernetes/minikube/tags>



$ mv minikube /usr/local/bin

$ chmod +x minikube

5、启动minikube

$ minikube start --driver=none --memory=8192 --cpus=4 --kubernetes-version=v1.17.0 --image-repository=registry.aliyuncs.com/google\_containers

\* minikube v1.12.1 on Centos 7.6.1810 (kvm/amd64)

\* Kubernetes 1.18.3 is now available. If you would like to upgrade, specify: --kubernetes-version=v1.18.3

\* Using the none driver based on existing profile

! The 'none' driver does not respect the --cpus flag

! The 'none' driver does not respect the --memory flag

! The 'none' driver does not respect the --cpus flag

! The 'none' driver does not respect the --memory flag

\* Starting control plane node minikube in cluster minikube

\* Restarting existing none bare metal machine for "minikube" ...

\* OS release is CentOS Linux 7 (Core)

\* Preparing Kubernetes v1.17.0 on Docker 19.03.12 ...

\* Configuring local host environment ...

\*

! The 'none' driver is designed for experts who need to integrate with an existing VM

\* Most users should use the newer 'docker' driver instead, which does not require root!

\* For more information, see: https://minikube.sigs.k8s.io/docs/reference/drivers/none/

\*

! kubectl and minikube configuration will be stored in /root

! To use kubectl or minikube commands as your own user, you may need to relocate them. For example, to overwrite your own settings, run:

\*

- sudo mv /root/.kube /root/.minikube $HOME

- sudo chown -R $USER $HOME/.kube $HOME/.minikube

\*

\* This can also be done automatically by setting the env var CHANGE\_MINIKUBE\_NONE\_USER=true

\* Verifying Kubernetes components...

\* Enabled addons: default-storageclass, storage-provisioner

\* Done! kubectl is now configured to use "minikube"

测试安装是否成功

$ minikube status

minikube

type: Control Plane

host: Running

kubelet: Running

apiserver: Running

kubeconfig: Configured

$ kubectl get node

NAME STATUS ROLES AGE VERSION

master Ready master 8m24s v1.17.0

**三、其他操作**

1、启动、停止、删除集群

$ minikube start --driver=none --memory=8192 --cpus=4 --kubernetes-version=v1.17.0 --image-repository=registry.aliyuncs.com/google\_containers

$ minikube stop

$ minikube delete

2、列出当前支持的插件

$ minikube addons list

|-----------------------------|----------|--------------|

| ADDON NAME | PROFILE | STATUS |

|-----------------------------|----------|--------------|

| ambassador | minikube | disabled |

| dashboard | minikube | enabled ✅ |

| default-storageclass | minikube | enabled ✅ |

| efk | minikube | disabled |

| freshpod | minikube | disabled |

| gvisor | minikube | disabled |

| helm-tiller | minikube | disabled |

| ingress | minikube | disabled |

| ingress-dns | minikube | disabled |

| istio | minikube | disabled |

| istio-provisioner | minikube | disabled |

| kubevirt | minikube | disabled |

| logviewer | minikube | disabled |

| metallb | minikube | disabled |

| metrics-server | minikube | disabled |

| nvidia-driver-installer | minikube | disabled |

| nvidia-gpu-device-plugin | minikube | disabled |

| olm | minikube | disabled |

| pod-security-policy | minikube | disabled |

| registry | minikube | disabled |

| registry-aliases | minikube | disabled |

| registry-creds | minikube | disabled |

| storage-provisioner | minikube | enabled ✅ |

| storage-provisioner-gluster | minikube | disabled |

|-----------------------------|----------|--------------|

3、启用指定插件，比如efk

$ minikube addons enable efk

4、禁用指定插件，比如efk

$ minikube addons disable efk

5、启动dashboard插件

$ minikube dashboard

[root@control-plane ~]# kubectl get svc --all-namespaces

NAMESPACE NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

default kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 82m

kube-system kube-dns ClusterIP 10.96.0.10 <none> 53/UDP,53/TCP,9153/TCP 82m

kubernetes-dashboard dashboard-metrics-scraper ClusterIP 10.96.231.151 <none> 8000/TCP 71m

kubernetes-dashboard kubernetes-dashboard ClusterIP 10.96.241.149 <none> 80/TCP 71m

#dashboard启动后默认是以ClusterIP方式暴露服务，无法使用浏览器访问，需要做一个代理

$ kubectl proxy --port=8088 --address=192.168.110.149 --accept-hosts='^.\*' > /dev/null 2>&1

#使用浏览器访问

http://192.168.110.149:8088/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/error?namespace=default

6、运行一个实例应用并暴露服务

$ kubectl create deployment my-nginx --image=nginx

$ kubectl expose deployment my-nginx --type=NodePort --port=8080