K8s lab part 1– Setting up Kubernetes 1.19.2

This guide sets up a single node K8s cluster on VMware workstation. This install guide is loosely based on: <https://phoenixnap.com/kb/how-to-install-kubernetes-on-centos>

**Remember**: Text in red is supposed to be executed, text in blue needs to be pasted in or added. Text in green is output from the system.

1. Assuming you have successfully setup the CentOS 7 VM, now use MobaXterm to connect to the VM through SSH.
2. Install nano and enable syntax highlighting (or use your favorite editor):

yum install -y nano

nano /usr/share/nano/yaml.nanorc

Paste this text in:

# Supports `YAML` files

syntax "YAML" "\.ya?ml$"

header "^(---|===)" "%YAML"

## Keys

color magenta "^\s\*[\$A-Za-z0-9\_-]+\:"

color brightmagenta "^\s\*@[\$A-Za-z0-9\_-]+\:"

## Values

color white ":\s.+$"

## Booleans

icolor brightcyan " (y|yes|n|no|true|false|on|off)$"

## Numbers

color brightred " [[:digit:]]+(\.[[:digit:]]+)?"

## Arrays

color red "\[" "\]" ":\s+[|>]" "^\s\*- "

## Reserved

color green "(^| )!!(binary|bool|float|int|map|null|omap|seq|set|str) "

## Comments

color brightwhite "#.\*$"

## Errors

color ,red ":\w.+$"

color ,red ":'.+$"

color ,red ":".+$"

color ,red "\s+$"

## Non closed quote

color ,red "['\"][^['\"]]\*$"

## Closed quotes

color yellow "['\"].\*['\"]"

## Equal sign

color brightgreen ":( |$)"

finally apply the profile:

find /usr/share/nano/ -iname "\*.nanorc" -exec echo include {} \; >> ~/.nanorc

1. Configure the hostname if you didn’t already during the VM install and configure the hosts file:

hostnamectl set-hostname k8s-controller

nano /etc/hosts

192.168.x.x k8s-controller

1. Install docker:

yum check-update

yum install -y yum-utils device-mapper-persistent-data lvm2

yum-config-manager --add-repo <https://download.docker.com/linux/centos/docker-ce.repo>

yum install -y docker

docker --version

systemctl start docker

systemctl enable docker

systemctl status docker

1. Install Kubernetes:

sudo nano /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64

enabled=1

gpgcheck=1

repo\_gpgcheck=1

gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg

yum install -y kubelet-1.19.2

yum install -y kubeadm-1.19.2

systemctl start kubelet

systemctl enable kubelet

systemctl status kubelet (Check “active” and “enabled” to make sure the kubelet is started)

1. Shut down firewalld:

systemctl disable firewalld

systemctl stop firewalld

1. Update/Create Iptables settings:

nano /etc/sysctl.d/k8s.conf

net.bridge.bridge-nf-call-ip6tables = 1

net.bridge.bridge-nf-call-iptables = 1

sysctl --system

1. Disable SELinux:

setenforce 0

sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config

1. Disable swap:

sed -i '/swap/d' /etc/fstab

swapoff -a

1. Initialize your k8s cluster and get flannel going. Ignore the firewalld warning as we have set that up properly.

kubeadm init --pod-network-cidr=10.244.0.0/16

mkdir -p $HOME/.kube

cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

chown $(id -u):$(id -g) $HOME/.kube/config

kubectl apply -f <https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml>

1. Verify k8s is ready to go:

kubectl get pods --all-namespaces

*NAMESPACE NAME READY STATUS RESTARTS AGE*

*kube-system coredns-5644d7b6d9-9fczp 1/1 Running 0 12m*

*kube-system coredns-5644d7b6d9-pjdqv 1/1 Running 0 12m*

*kube-system etcd-controller-node 1/1 Running 0 11m*

*kube-system kube-apiserver-controller-node 1/1 Running 0 11m*

*kube-system kube-controller-manager-controller-node 1/1 Running 0 11m*

*kube-system kube-flannel-ds-7w4fm 1/1 Running 0 39s*

*kube-system kube-proxy-sv8rn 1/1 Running 0 12m*

*kube-system kube-scheduler-controller-node 1/1 Running 0 11m*

1. Install Bash completion to make your life easier. **Close and restart the SSH session after this**:

yum install -y bash-completion

source <(kubectl completion bash)

echo "source <(kubectl completion bash)" >> ~/.bashrc

1. Allow pods to be scheduled to run on the controller node itself by untainting “NoSchedule”:

kubectl taint node k8s-controller node-role.kubernetes.io/master:NoSchedule-

1. Reboot the node:

shutdown -r now

1. After the reboot make sure Kubernetes is up and running. If you get “x.x.x.x:6443 was refused” or “TLS handshake timeout”, just wait for the kubelet service to start:

kubectl get nodes

NAME STATUS ROLES AGE VERSION

k8s-controller Ready master 8m3s v1.19.2

Download the yaml files to ~/yaml and deploy your first pods using kubectl apply -f ~/yaml/nginx-deploy.yml. Want to play more with kubectl? Check <https://kubernetes.io/docs/reference/kubectl/cheatsheet/>