Steps to Set Up Kubernetes Clusters environment on CentOS 7

overview

The systematic configuration of a Kubernetes cluster. The installation process takes place on CentOS 7, with virtualization platform on Oracle VirtualBox.

Step 1

System Prerequirements

Download & Install centOS 7

↳ Create & Lauch a master node.
Create & Lauch a Working node.

System requirements for master & working nodes

↳
- Above 1700 mb - RAM
- 2 CPU processor

Step 2

SYSTEM UPDATATIONS

Use the following commands.

Here, we'll update the machine.

```
$ sudo yum update -y
```

Here, we'll add the content and change the system host name.

```
$sudo vim /etc/hostname
```

content : k8-master

Here, we'll add the content and change the system host file for all the nodes

```
$vim /etc/hosts
```

content :

```
192.168.55.10  k8-master
192.168.55.20  K8-worker1
```

Note: perform all these updatations on master & working nodes.

Step 3

Installing Pakages & Updatations

Here, we'll switch off the swap for all the nodes.

```
$ sudo swapoff -a
```

Here, we'll stop firewall for all nodes.

```
$ sudo systemctl stop firewalld
```

To permanently disable the firewall:

```
$ sudo systemctl disable firewalld
```

Here, we'll download some packages. on all the nodes

```
$sudo yum install -y wget
$sudo yum install -y git
```

Note: Update and install these packages on the master and working nodes.

Step 4

Installing container runtime

Here, we'll set up the docker repository.

```
$ sudo yum install -y yum-utils
$ sudo yum-config-manager --add-repo
https://download.docker.com/linux/centos/docker-ce.repo
```

Here, we'll Install containerd

```
$ Sudo yum install  containerd.io
```

Here, we will verify if the containerized service has been activated or not.

```
$ Sudo Systemctl status containerd
```

Here, we will start the containerized service.

```
$ Sudo Systemctl start containerd
```

Note: In this case, container runtime service will be our only tool. Installing Docker is not necessary because the containerd service is sufficient for managing Kubernetes clusters.

Step 5

Installing Kubernetes

Here, we'll We will install the kubeadm, kubectl, and kubelet packages, which are necessary for Kubernete.

- **Kubeadm:** is a tool used to build Kubernetes (K8s) clusters.
- **Kubectl:** is a command line tool that allows you to run commands against Kubernetes clusters.
- **Kubelet:** is a key component of Kubernetes that helps manage containers and orchestrate them within a cluster.

Here, we will Set SELinux to permissive mode(for centos)

```
$ sudo setenforce 0
$ sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/'
/etc/selinux/config
```

Here, we will Add the Kubernetes yum repository(for centos)

```
# This overwrites any existing configuration in /etc/yum.repos.d/kubernetes.repo
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.29/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.29/rpm/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
```

Note: All packages for Kubernetes 1.29 are available in this repository; to get packages for other Kubernetes minor versions, modify the URL to reflect the relevant minor version. This repository is limited to operating systems based on Red Hat

Here, we will Install kubelet, kubeadm and kubectl.

```
$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
```

Here, we will Enable the kubelet service before running kubeadm.

```
$ sudo systemctl enable --now kubelet
```

Here, we will init cluster.

```
$ sudo kubeadm init
```

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

```
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Alternatively, if you are the root user, you can run.

```
export KUBECONFIG=/etc/kubernetes/admin.conf
```

If the Join command token is expired or you forgot:

```
kubeadm token create --print-join-command
```

Note: Once you execute the last command, it will run but will give you some issues. I will be providing methods to fix those errors. Go through Github Link.