How To Install Vanilla Kuberntes on Centos 7

Introduction

This guide will describe the process of setting up a vanilla Kubernetes (K8s) cluster on a single Dell server using Kernel-based Virtual Machines (KVM). The goal is to create a fully functional, minimalistic Kubernetes environment that can serve as a robust test or development cluster.

Why Use KVM on CentOS 7?

- **Virtualization**: KVM provides a type-1 (bare-metal) hypervisor, which means it operates directly on the hardware of your Dell server, offering better performance and resource utilization compared to type-2 hypervisors. This is ideal for creating isolated environments where each virtual machine acts as a node in your Kubernetes cluster.
- **CentOS 7**: Known for its stability and compatibility with enterprise-level applications, CentOS 7 remains a preferred choice among system administrators for setting up robust server environments. Its long-term support and wide community backing ensure that you have access to necessary documentation and community help.
- **Test/Dev Environment**: Installing Kubernetes in a virtualized setup allows for flexibility in testing different configurations, experimenting with new features, or simulating production environments without the risk of affecting real production systems.

Key Benefits of This Setup:

- **Isolation**: Each virtual machine can be configured with different resources or network settings, providing an excellent environment for testing various scenarios or application behaviors.
- **Resource Management**: KVM on a Dell server allows for fine-tuned control over resource allocation, which is perfect for simulating different workloads or testing scalability.
- **Learning and Experimentation**: For those learning Kubernetes, this setup offers a hands-on approach to understanding cluster

management, networking, and container orchestration without the cost of cloud services.

What This Guide Covers:

- **Docker Installation**: Installing Docker as the container runtime for Kubernetes.
- **Kubernetes Components**: Installing `kubeadm`, `kubelet`, and `kubectl` on your nodes.
- **Cluster Initialization**: Bootstrapping your Kubernetes master node and joining worker nodes.
- **Networking**: Configuring networking for your VMs to ensure proper communication within the cluster.
- **Post-Installation**: Verifying your cluster's health and basic operations.

Prerequisites

- A Dell server with sufficient CPU, memory, and disk resources (at least 16 GB RAM and 4 CPUs recommended) with Centos 7 installed. Note: for the host machine the OS system does not have to be Centos it can be any Linux distribution that supports KVM.
- KVM and associated tools (libvirt, virt-manager, and virsh) installed on the server.
- Centos 7 OS installed on two KVM VMs one for the master and one for the worker nodes.
- Basic familiarity with Linux command-line operations.

Here the official sites for installing Centos7 and KVM. However, there are plenty of good tutorials available on the Web.
Centos Install Guide
KVM How to

Step-by-Step Instructions

Steps 1 thorugh 10 are to be performed on master node and all worker nodes!

1. Disable selinux and firewalld

Note, if you need a firewall running must open the ports required for all applications. For example, the API server uses port 6443. Click on the below link for the ports required.
https://kubernetes.io/docs/reference/networking/ports-and-protocols/

```
setenforce 0
sed -i --follow-symlinks 's/SELINUX=enforcing/SELINUX=di
systemctl stop firewalld
systemctl disable firewalld
modprobe br_netfilter
echo '1' > /proc/sys/net/bridge/bridge-nf-call-iptables
```

2. Disable swap

```
sudo swapoff -a
```

3. Install Kubernetes repository

```
cat <<EOF> /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=<https://packages.cloud.google.com/yum/repos/kub
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=<https://packages.cloud.google.com/yum/doc/yum-ke
EOF
```

4. Install Kubernetes

Configure kubernetes repo

```
cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kube
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key
EOF
```

5. Install kubectl, kubeadm and kubectl

```
yum update
yum install -y kubelet kubeadm kubectl
systemctl enable kubelet
systemctl start kubelet
```

6. Install Docker

If you need to unistall an old verison of docker.

```
sudo yum remove docker \
                  docker-client \
                  docker-client-latest \
                  docker-common \
                  docker-latest \
                  docker-latest-logrotate \
                  docker-logrotate \
                  docker-engine
```

7. Configure docker repository

```
yum install -y yum-utils

yum-config-manager \
    --add-repo \
    https://download.docker.com/linux/centos/docker-ce.r
```

8. Install docker docker compose and containerd

```
yum install docker-ce docker-ce-cli containerd.io docker
```

Start and enable docker

```
systemctl start docker
systemctl enable docker
```

9.Ensure docker is running

```
docker run hello-world
```

10. Remove config.toml file

You must remove this file otherwise when the cluster is initialized it will fail.

```
cd /etc/ containerd/
rm config.toml
systemctl start containerd
systemctl enable containerd
```

These next steps are to be performed on the Master Node only!

```
kubeadm init
```

This will take a few minutes. At the end of the intialization you should see the message.

```
Your Kubernetes control-plane has initialized successful

To start using your cluster, you need to run the followi

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

You should now deploy a Pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the
  /docs/concepts/cluster-administration/addons/

You can now join any number of machines by running the f
  kubeadm join <control-plane-host>:<control-plane-port>
```

For example on your cluster replace your local info such as:

```
kubeadm join --token 7c0767.e8467d0b00930f2e 172.18.10.8
```

Installing a Pod Network

The final step is to install a pod network application for intra cluster communication, in this case Calico. Download the Calico networking manifest for the Kubernetes API datastore.

```
curl https://raw.githubusercontent.com/projectcalico/cal
```

Apply the calico yaml to the cluster.

```
kubectl apply -f calico.yaml
kubectl get pods -n kube-system
```

Verifying Installation

Form the master node verify your nodes are in the ready state using the kubectl get nodes command.

```
demo@training-kube-master$ kubectl get nodes -o wide
```

If you are interested in a video demonstration on how to perform basic interactions with pods and nodes click on the link below. Note this is a partial video for demonstration purposes only.

Sample Work K8s lab

It will cover:

- **Viewing the cluster's current state** – Understanding what's running in the cluster.
- **Creating and managing Pods** – The basic units of deployment in Kubernetes.
- **Scaling applications** – Adjusting the number of running instances of an application.
- **Exploring Kubernetes objects** – Learning about services, deployments, and namespaces.