INSTALLING KUBEADM

Wednesday, January 4, 2023 1:20 PM

1) Check the mac address of the network interfaces(ip link,ifconfig -a)
2) The product muuid using (sudo cat /sys/class/dmi/id/product_uuid)
3) Use netcat to see the required ports are open (nc 127.0.0.1 6443)(127.0.0.53 to 8.8.8.8 in nameserver in/etc/resolv.conf(sudo vi /etc/resolv.conf)(any functional dns server ip's)
4) Container runtime

Runtime	Path to Unix domain socket
Containerd	unix:///var/run/containerd/containerd.sock npipe:////./pipe/containerd-containerd (windows)
CRI-O	unix:///var/run/crio/crio.sock
Docker Engine (using cri-dockerd)	unix:///var/run/cri-dockerd.sock npipe:////./pipe/cri-dockerd (windows)

5) You will install these packages on all of your machines:

   kubeadm: the command to bootstrap the cluster.

   kubelet: the component that runs on all of the machines in your cluster and does things like starting pods and containers.

   kubectl: the command line util to talk to your cluster.

sudo apt-get update
sudo apt-get install -y apt-transport-https ca-certificates curl

sudo curl -fsSLo /etc/apt/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg (/etc/hostname is the hostname /etc/hosts should have the 127.0.0.1 loclahost\n127.0.0.1 node1)

echo "deb [signed-by=/etc/apt/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list

sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl

For red hat based version , without package manager see https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/

Without package manager:(if above gives errors)

CNI_PLUGINS_VERSION="v1.1.1"
ARCH="amd64"
DEST="/opt/cni/bin"
sudo mkdir -p "$DEST"
curl -L "https://github.com/containernetworking/plugins/releases/download/${CNI_PLUGINS_VERSION}/cni-plugins-linux-${ARCH}-${CNI_PLUGINS_VERSION}.tgz" | sudo tar -C "$DEST" -xz

```
DOWNLOAD_DIR="/usr/local/bin"
sudo mkdir -p "$DOWNLOAD_DIR"

CRICTL_VERSION="v1.25.0"
ARCH="amd64"
curl -L "https://github.com/kubernetes-sigs/cri-tools/releases/download/${CRICTL_VERSION}/crictl-${CRICTL_VERSION}-linux-${ARCH}.tar.gz" | sudo tar -C $DOWNLOAD_DIR -xz

RELEASE="$(curl -sSL https://dl.k8s.io/release/stable.txt)"
ARCH="amd64"
cd $DOWNLOAD_DIR
sudo curl -L --remote-name-all https://dl.k8s.io/release/${RELEASE}/bin/linux/${ARCH}/{kubeadm,kubelet}
sudo chmod +x {kubeadm,kubelet}

RELEASE_VERSION="v0.4.0"
curl -sSL "https://raw.githubusercontent.com/kubernetes/release/${RELEASE_VERSION}/cmd/kubepkg/templates/latest/deb/kubelet/lib/systemd/system/kubelet.service" | sed "s:/usr/bin:${DOWNLOAD_DIR}:g" | sudo tee /etc/systemd/system/kubelet.service
sudo mkdir -p /etc/systemd/system/kubelet.service.d
curl -sSL "https://raw.githubusercontent.com/kubernetes/release/${RELEASE_VERSION}/cmd/kubepkg/templates/latest/deb/kubeadm/10-kubeadm.conf" | sed "s:/usr/bin:${DOWNLOAD_DIR}:g" | sudo tee /etc/systemd/system/kubelet.service.d/10-kubeadm.conf

systemctl enable --now kubelet ( the vm should be password configured)
```

6) Both the container runtime and the kubelet have a property called "cgroup driver", which is important for the management of cgroups on Linux machines.

Matching the container runtime and kubelet cgroup drivers is required or otherwise the kubelet process will fail.