JUNE 11, 2024

DevOps Classroom Notes – 11/June/2024

**Kubernetes Setup**

* Kubeadm:
  + This is setup where need to have physical or virtual machines with connectivity between them ready i.e. we can use this setup on-premises as well as virtual machines
* Managed k8s:
  + This is k8s as a service offered by cloud providers

**kubedadm**

* [Refer Here](https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/) for the documentation

**Setup**

* Create two vms in any cloud with network connectivity between them and also the ports mentioned over here open [Refer Here](https://kubernetes.io/docs/reference/networking/ports-and-protocols/)
* Install docker on both machines (container runtime)
* Install kubeadm,kubelet, kubectl on all nodes [Refer Here](https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/#installing-kubeadm-kubelet-and-kubectl)

sudo apt-get update

# apt-transport-https may be a dummy package; if so, you can skip that package

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

curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.30/deb/ /' | 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

* Configuring CRI runtime

wget https://github.com/Mirantis/cri-dockerd/releases/download/v0.3.14/cri-dockerd\_0.3.14.3-0.ubuntu-jammy\_amd64.deb

sudo dpkg -i cri-dockerd\_0.3.14.3-0.ubuntu-jammy\_amd64.deb

* Choose your master (node-1) become a root user

kubeadm init --cri-socket unix:///var/run/cri-dockerd.sock

* When executed this command will give the output as shown below

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

You should now deploy a pod network to the cluster.

Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:

https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 10.0.0.5:6443 --token f9zz3v.zlm2l3ki8jmiso9z \

--discovery-token-ca-cert-hash sha256:0f95d518a316f089a5ce3b74079e08421e454ec0b6c7ecadb645d5e11690af81

* reference installation [Refer Here](https://directdevops.blog/2024/01/23/devops-classroom-notes-23-jan-2024/)
* [Refer Here](https://kubernetes.io/docs/reference/kubectl/quick-reference/) for kubectl cheatsheet

**Share this:**