**Installation and configuration with centos with docker -1 master and 2 worker**

* Many Kubernetes installation tools, we will use minikube, kubeadm
* Deploy a server with minimum required specs - 2vCPU/2GiB RAM minimum
* On root user:
  + Disable swap (swapoff -a)
  + Disable all the firewall (systemctl disable firewalld)
  + Disable selinux ( setenforce 0 | vim /etc/sysconfig/selinux )
* Create a new user, set sudo and gain user access

**Introduction: Configuring system for sudo -**

* Sudo setup
  + Create a new user/password
  + vim /etc/sudoers.d/student
  + student ALL=(ALL) ALL
  + chmod 440 student
  + Login to new user
* PATH variable setup on the user
  + Allow search in /sbin and /usr/bin
  + Open .bashrc and add:
    - PATH=$PATH:/usr/sbin:/sbin
  + Save and logout/login

**Basics of Kubernetes**

* Fast changing project site: <https://kubernetes.io/>
* Source code on Github: <https://github.com/kubernetes/kubernetes/>
* Feature and Issue tracking: <https://github.com/kubernetes/kubernetes/issues>

**Kubeadm (Boot strap utility)**

* Setup centOS 8 - 3 machine - Master and 2 worker node.
* Set hostname for all 3 nodes and hosts file entry, sudo, selinux, firewall
* Set Docker and Kubeadm repo

**For docker**

sudo yum install -y yum-utils

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

sudo yum install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

**For K8**

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

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

sudo systemctl enable --now kubelet

* Install docker, kubeadm-1.29\*, kubectl-1.29\*, kubelet-1.29\*
* Start/enable docker and kubelet service, kubectl service and kubeadm service
* kubeadm init --pod-network-cidr=10.244.0.0/16 -> Setup master, & pre-config
* Config setup - /etc/kubernetes/admin.conf -> $HOME/.kube/config

**ERRORS**

When resources are less or command not installed proper

A close-up of a text

Description automatically generated

**Command reference**

* kubectl get node
* rm $HOME/.kube/config
* kubectl get nodes --kubeconfig=/etc/kubernetes/admin.conf (super-user config file)
* cp -i /etc/kubernetes/admin.conf $HOME/.kube/config (Master)
* netstat -tlpn | grep -w 6443 (API server port number)
* kubeadm token list
* kubeadm token create
* kubeadm join API\_SERVER\_IP:PORT --token <> --discovery-token-ca-crt-hash sha:<[here](https://take.ms/G2aZav)>
* Kubeadm token create --print-join-command (If you miss join command)
* netstat -tlpn | grep kube-proxy (10249)

**FILE REFRENCE**

* Master:
  + - /etc/kubernetes/pki/apiserver.crt;
    - /etc/kubernetes/pki/ca.crt
    - /etc/kubernetes/manifests
    - $HOME/.kube/cache/discovery/$host:port/v1 … (multiple kind of files)
* Worker:
  + - /var/lib/kubelet/config.yaml
    - /var/lib/kubelet/pka/kubelet.crt
    - /var/lib/kubelet/kubelet.key
* CNI -> Container network interface plugin (SDN)
* Worker nodes not ready till now as container network is not defined yet
* Software switch is allocate IP to container and C1-C2 interaction via routing table