**Steps Needed for Installing Kubernetes with Vagrant and Kubeadm**

*Tools Needed*

* Vagrant
* Virtualbox
* Git

*Before you start*

* Create this folder structure in windows under your home directory or anywhere else.
  + K8s
    - master
    - worker1
    - worker2
* Copy the below file in the master folder



* Copy the below file in worker1 folder



* Copy the below file in worker2 folder



* Install vagrant in your system from the below location
  + <https://www.vagrantup.com/downloads>
* Install git and get git bash ready
  + <https://git-scm.com/download/win>
* Open a “command prompt” and issue ipconfig
  + Note the IPv4 address under the “Ethernet Adapter Virtualbox Host-Only Network”
* Changing the Vagrantfile specific to your laptop
  + Based on the ipv4 address from above, change the following line:
    - config.vm.network “private\_network”, ip: “**192.168.56**.30”
    - The highlighted portion should be the same as that in your laptop

**Creating the Master Node**

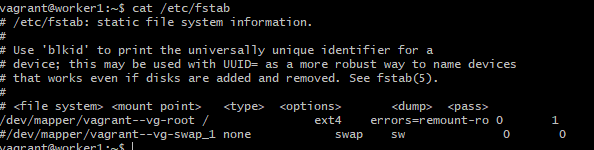
* Go to the “master” folder in git bash
* Issue “**vagrant up”** command
  + The VM will get launched
* Issue “**vagrant ssh”**
  + This will connect to the VM

**Creating the worker1/worker2 nodes**

* Go to the “worker1/worker2” folder in git bash
* Issue “**vagrant up”** command
  + The VM will get launched
* Issue “**vagrant ssh”**
  + This will connect to the VM

**For all the nodes**

* Disable Swap
  + **sudo swapoff -a**
  + **sudo vi /etc/fstab** # comment on swap file system



* Add the Docker Repository on all three servers.
  + **curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –**
  + **sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"**
* Add the Kubernetes repository on all three servers.
  + **curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add –**
  + **cat << EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list**

**deb https://apt.kubernetes.io/ kubernetes-xenial main**

**EOF**

* Install Kubeadm, Kubelet, Kubelet and Kubernetes-cni
  + ***sudo apt-get update \  
     && sudo apt-get install -y kubelet kubeadm kubernetes-cni \ kubectl***
* Install docker
  + **sudo apt-mark hold docker-ce kubelet kubeadm kubectl**
* Enable bridge networking
  + ***echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf***
  + ***sudo sysctl –p /etc/sysctl.conf***
* Fix docker cgroupdriver

sudo mkdir /etc/docker

cat <<EOF | sudo tee /etc/docker/daemon.json

{

"exec-opts": ["native.cgroupdriver=systemd"],

"log-driver": "json-file",

"log-opts": {

"max-size": "100m"

},

"storage-driver": "overlay2"

}

EOF

sudo systemctl enable docker

sudo systemctl daemon-reload

sudo systemctl restart docker

**Do this only in the Master**

**sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --apiserver-advertise-address=<ip-address-mentioned in Vagrantfile of master>**

This will keep  a token which help to join the other nodes. Please keep it save.

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

* Install flannel
  + ***kubectl apply -f*** [***https://github.com/coreos/flannel/raw/master/Documentation/kube-flannel.yml***](https://github.com/coreos/flannel/raw/master/Documentation/kube-flannel.yml)

**Join the other nodes**

* use the output of the kubeadm init command on all worker nodes.

**sudo kubeadm join <Master\_ip):6443 –token <token>–discovery-token-ca-cert-hash <hash)**