I have put all commands in multiple shell scripts for Debian. Run them as below

Note: Come to this once you have gone through the steps individually.

1. **bash install\_docker.sh**
2. **Use “sudo su -” to run the following commands**

**cat <<EOF > /etc/sysctl.d/k8s.conf**

**net.bridge.bridge-nf-call-ip6tables = 1**

**net.bridge.bridge-nf-call-iptables = 1**

**EOF**

**echo 1 > /proc/sys/net/ipv4/ip\_forward**

1. **bash install\_kubeadm.sh**
2. **Run sudo kubeadm init --pod-network-cidr 10.168.0.0/16 (in master node)**
3. **Run the following in MASTER node**
   1. **mkdir -p $home/.kube**
   2. **sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config**
   3. **sudo chown $(id -u):$(id -g) $HOME/.kube/config**
4. **bash install\_nw\_plugin.sh (only on MASTER)**

# Step 1 – Install Docker Runtime (on 3 servers/machines/VMs)

Note: These commands are for RHEL 7 only. If you are using any other operating system then check the documentation pertaining to that OS to get the commands to install docker, kubeadm, kubelet and kubectl

**sudo yum -y upgrade**

**sudo yum install -y epel-release**

**sudo yum install -y yum-utils device-mapper-persistent-data-lvm2**

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

**sudo yum install -y docker**

**sudo systemctl start docker.service**

**sudo systemctl enable docker.service**

# For Debian

**sudo apt-get update**

**sudo apt-get install apt-transport-https ca-certificates curl gnupg2 software-properties-common**

**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/debian**](https://download.docker.com/linux/debian) **$(lsb\_release -cs) stable"**

**sudo apt-get update**

**sudo apt-get install docker-ce docker-ce-cli containerd.io**

# Step 2 – Install Kubeadm, Kubelet, Kubectl **(on all Nodes)**

**It is better to follow this link** - https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/

* Edit the file “/etc/yum.repos.d/kubernetes.repo”. Copy and paste the following lines (better to type the lines)

**[kubernetes]**

**name=Kubernetes**

**baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64**

**enabled=1**

**gpgcheck=1**

**repo\_gpgcheck=1**

**gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg**

**exclude=kube\***

* Set SELinux in permissive mode and turn swap off. Run the following commands

**sudo setenforce 0**

**sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config**

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

**sudo systemctl enable --now kubelet**

**sudo swapoff -a**

* Install kubeadm, kubelet and kubectl and enable kubelet

**sudo yum install -y kubeadm kubelet kubectl --disableexcludes=Kubernetes**

**sudo systemctl enable --now kubelet**

* Following steps are needed only for RHEL7. Edit the file /etc/sysctl.d/k8s.conf and paste the following lines

sudo vi /etc/sysctl.d/k8s.conf

**net.bridge.bridge-nf-call-ip6tables = 1**

**net.bridge.bridge-nf-call-iptables = 1**

**\*\* Run the following commands \*\***

**sudo sysctl --system**

**sudo modprobe br\_netfilter**

**lsmod | grep br\_netfilter ##(this should give you some output)**

# For Debian

Run as root – do a “sudo su - ”

**cat <<EOF > /etc/sysctl.d/k8s.conf**

**net.bridge.bridge-nf-call-ip6tables = 1**

**net.bridge.bridge-nf-call-iptables = 1**

**EOF**

**echo 1 > /proc/sys/net/ipv4/ip\_forward**

**\*\* You can come out of ROOT user now**

**lsmod | grep br\_netfilter**

**sudo apt-get install -y iptables arptables ebtables**

**sudo update-alternatives --set iptables /usr/sbin/iptables-legacy**

**sudo update-alternatives --set ip6tables /usr/sbin/ip6tables-legacy**

**sudo update-alternatives --set arptables /usr/sbin/arptables-legacy**

**sudo update-alternatives --set ebtables /usr/sbin/ebtables-legacy**

**sudo apt-get update && sudo apt-get install -y apt-transport-https curl**

**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**

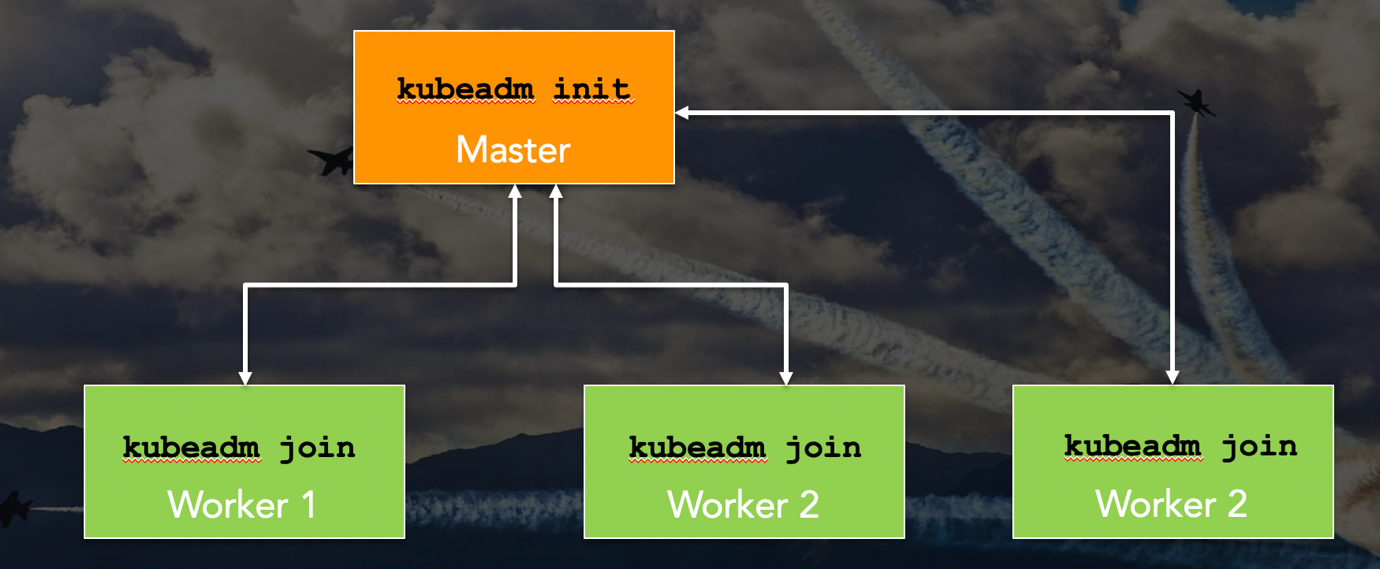
**sudo apt-get update**

**sudo apt-get install -y kubelet kubeadm kubectl**

**sudo apt-mark hold kubelet kubeadm kubectl**

# Step 3 – Set up a cluster using kubeadm

We will create cluster with 2 worker nodes as below.



Start “Docker” service on all the nodes –

**sudo** **systemctl start docker.service**

**sudo systemctl enable docker.service**

Check if Docker service is running or not

**sudo service docker status**

Run commands as mentioned below:

1. On MASTER Node > **sudo kubeadm init --pod-network-cidr 10.168.0.0/16**
2. Run the following commands
   1. **mkdir -p $HOME/.kube**
   2. **sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config**
   3. **sudo chown $(id -u):$(id -g) $HOME/.kube/config**
3. On MASTER Node > This is for Pod network plugin. We will be using WeaveNet for this. Don’t worry too much about it. Run the command –

**kubectl apply -f "**[**https://cloud.weave.works/k8s/net?k8s-version=$(kubectl**](https://www.google.com/url?q=https://cloud.weave.works/k8s/net?k8s-version%3D$(kubectl&sa=D&ust=1524760785241000&usg=AFQjCNF4lxXiYNW0iq1-mOap4Wr7Ei9-YA)**version | base64 | tr -d '\n')"**

The following one is for Flannel. Use Weave as we will use it for some of the demos. Also, Flannel does not have the advanced features.

**kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml**

1. On WORKER Nodes > Whatever output you get with “**sudo kubeadm join …**”

Note: You might have to run the command **sudo sysctl -w net.ipv4.ip\_forward=1** if you get the following error.

W0304 16:58:27.648459 14964 join.go:346] [preflight] WARNING: JoinControlPane.controlPlane settings will be ignored when control-plane flag is not set.

[preflight] Running pre-flight checks

error execution phase preflight: [preflight] Some fatal errors occurred:

[ERROR FileContent--proc-sys-net-ipv4-ip\_forward]: /proc/sys/net/ipv4/ip\_forward contents are not set to 1

[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...`

To see the stack trace of this error execute with --v=5 or higher

1. To check the status of the nodes run **“kubectl get nodes”** on master node.

# Optional Troubleshooting

While running “Kubectl get nodes” command if you get the following error

The connection to the server localhost:8080 was refused - did you specify the right host or port?

Then run the following commands in sequence. These commands are enabling you to work as a K8S admin.

**sudo cp /etc/kubernetes/admin.conf $HOME/**

**sudo chown $(id -u):$(id -g) $HOME/admin.conf**

**export KUBECONFIG=$HOME/admin.conf**

NOTE: In case your Kubernetes cluster does not come up after restarting EC2 instances or Google Compute Engine instances and kubectl command gives error, follow the steps below to fix the issue:

* Start Docker Service on all nodes
* Check if kubectl (kubectl get nodes) command works or not. If it DOES NOT then only follow the steps below.
* Disable swap
* Enable kubelet service on all nodes
* Try kubectl command again