**Kubernetes Installation**

Kubernetes is a container management technology developed in Google lab to manage containerized applications in different kind of environments such as physical, virtual, and cloud infrastructure. It is an open source system which helps in creating and managing containerization of application

**Prerequisite:**

1. Red Hat Enterprise Linux (version 7.0 or above) machine

**Steps:**

1. Launch RHEL 7.3 in AWS
2. Connect the virtual machine using putty
3. Login the machine as: ec2-user

**Commands to execute:**

Docker should be installed in both master and node machines (for docRef. Docker Installation.txt)

**Master Machine**

1) Navigate to the repository

**cd /etc/yum.repos.d**

2) save the file

**vi kubernetes.repo**

**[kubernetes]**

**name=Kubernetes**

**baseurl=http://yum.kubernetes.io/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**](https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg)

3) Install kubelet,kubeadm,kubectl

**yum install -y kubelet-1.6.1-0 kubeadm-1.6.1-0 kubectl-1.6.1-0 kubernetes-cni**

4) **systemctl enable kubelet**

**systemctl start kubelet**

5) unset proxy (if any)

**unset http\_proxy**

**unset https\_proxy**

6) on the master node –init kubeadm

**kubeadm init --apiserver-advertise-address=<your MasterIp > --apiserver-bind-port=8080**

**OUTPUT :**

[root@zlt15174 yum.repos.d]# kubeadm init --apiserver-advertise-address=135.167.156.51 --apiserver-bind-port=8080

[kubeadm] WARNING: kubeadm is in beta, please do not use it for production clusters.

[init] Using Kubernetes version: v1.6.0

[init] Using Authorization mode: RBAC

[preflight] Running pre-flight checks

[preflight] Starting the kubelet service

[certificates] Generated CA certificate and key.

[certificates] Generated API server certificate and key.

[certificates] API Server serving cert is signed for DNS names [zlt15174 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 135.167.156.51]

[certificates] Generated API server kubelet client certificate and key.

[certificates] Generated service account token signing key and public key.

[certificates] Generated front-proxy CA certificate and key.

[certificates] Generated front-proxy client certificate and key.

[certificates] Valid certificates and keys now exist in "/etc/kubernetes/pki"

[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/admin.conf"

[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/kubelet.conf"

[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/controller-manager.conf"

[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/scheduler.conf"

[apiclient] Created API client, waiting for the control plane to become ready

[apiclient] All control plane components are healthy after 15.044053 seconds

[apiclient] Waiting for at least one node to register

[apiclient] First node has registered after 5.003322 seconds

[token] Using token: 1b9d68.1bb44aba3de83330

[apiconfig] Created RBAC rules

[addons] Created essential addon: kube-proxy

[addons] Created essential addon: kube-dns

Your Kubernetes master has initialized successfully!

To start using your cluster, you need to run (as a regular user):

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

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

export KUBECONFIG=$HOME/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:

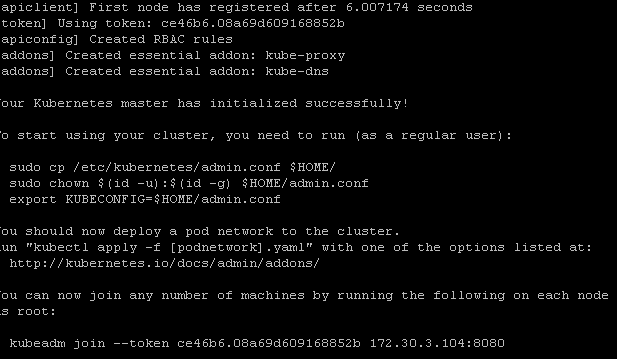
http://kubernetes.io/docs/admin/addons/

You can now join any number of machines by running the following on each node

as root:

kubeadm join --token 1b9d68.1bb44aba3de83330 135.167.156.51:8080

[root@zlt15174 yum.repos.d]#



**Node Machine**

1) Navigate to the repository

**cd /etc/yum.repos.d**

2) save the file

**vi kubernetes.repo**

**[kubernetes]**

**name=Kubernetes**

**baseurl=http://yum.kubernetes.io/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**](https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg)

3) Install kubelet,kubeadm,kubectl

**yum install -y kubelet-1.6.1-0 kubeadm-1.6.1-0 kubectl-1.6.1-0 kubernetes-cni**

4) **systemctl enable kubelet**

**systemctl start kubelet**

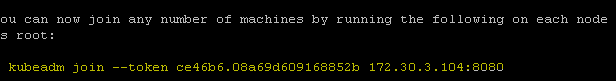
5) unset proxy (if any)

**unset http\_proxy**

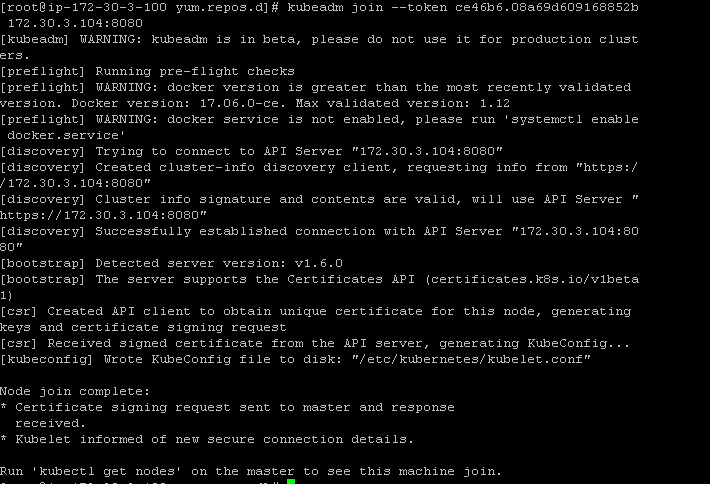
**unset https\_proxy**

6) Join Node with the Master

**Copy the token which is displayed while executing init command**



7) Master and slave nodes joined



We can join “n” no of nodes by executing these commands

**DashBoard:**

**Master :**

1) **wget https://git.io/weave-kube-1.6**

**wget** [**https://rawgit.com/kubernetes/dashboard/master/src/deploy/kubernetes-**](https://rawgit.com/kubernetes/dashboard/master/src/deploy/kubernetes-)**dashboard.yaml**

2) **kubectl apply -f weave-kube-1.6**

**kubectl apply -f kubernetes-dashboard.yaml** // THIS HAS TO BE DONE ON SESSION WHICH DOESNOT HAS PROXY SET.

3) NOW

**kubectl proxy --api-prefix=/ --address='135.167.156.51' --accept-hosts='135.167.156.51' --port=7080 &**

4) **kubectl create clusterrolebinding add-on-cluster-admin --clusterrole=cluster-admin --serviceaccount=kube-system:default**

5) ACCESS ON BROWSER:

**http://135.167.156.51:7080/ui**

