**Install Kubernetes Cluster**

In this demo we will be establishing a demo on how to install Kubernetes Cluster with two servers out of which one server will be master server and other one will be slave machine.

This demo will be divided in mainly three components:

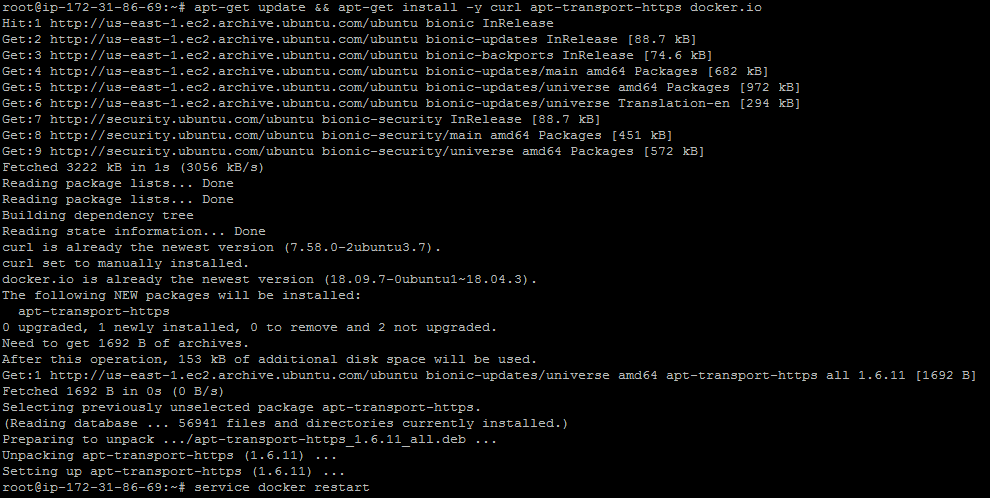
1. Installing prerequisites packages and Installing Kubernetes
2. Configuring Kubernetes Master
3. Configuring Kubernetes Slave

**Installing prerequisites packages and Installing Kubernetes**

Before proceeding with Kubernetes installation we need to install some prerequisites packages on both servers. Please follow below set of commands to instal.

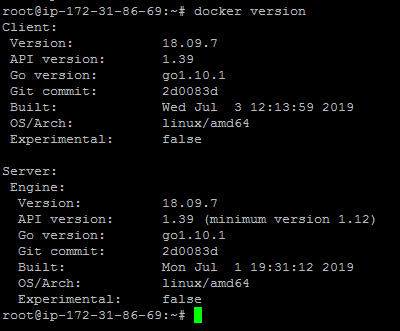
apt-get update && apt-get install -y curl apt-transport-https docker.io

service docker restart



Once Docker is installed we can validate if Docker is installed perfectly or not. We can use docker version command to validate Docker.

docker version



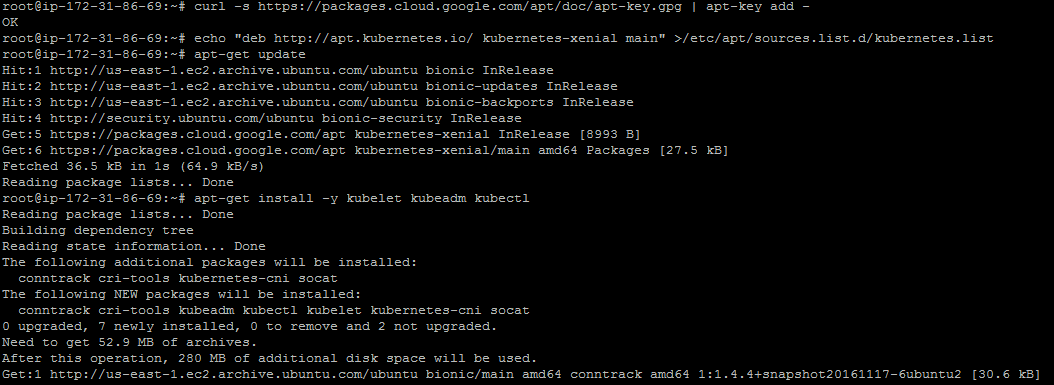
Once docker installation is done we can install and configure Kubernetes using below procedure.

curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -

echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" >/etc/apt/sources.list.d/kubernetes.list

apt-get update

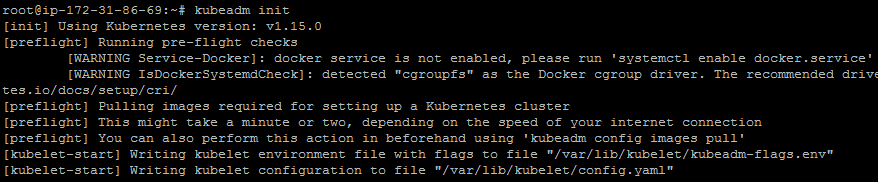
apt-get install -y kubelet kubeadm kubectl



**Configuring Kubernetes Master**

Once Kubernetes is installed then we can configure Kubernetes master on one of the server so that we can master instance up and running. Follow below commands on Kubernetes Master node.

kubeadm init

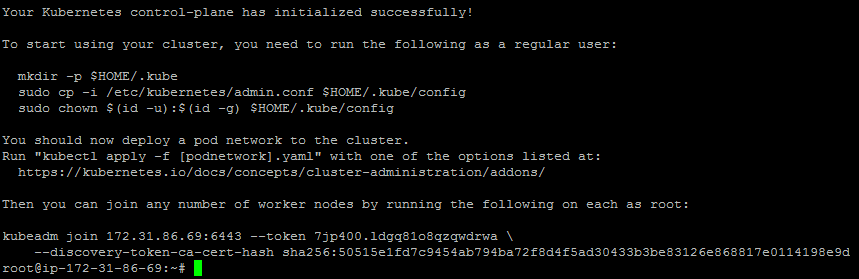


Once Kubernetes is initialized we can proceed with configuring Kubernetes so that we can start using Kubernetes cluster.

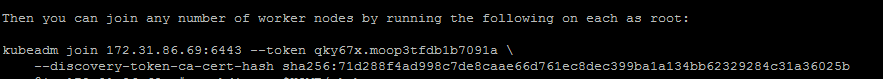
mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config



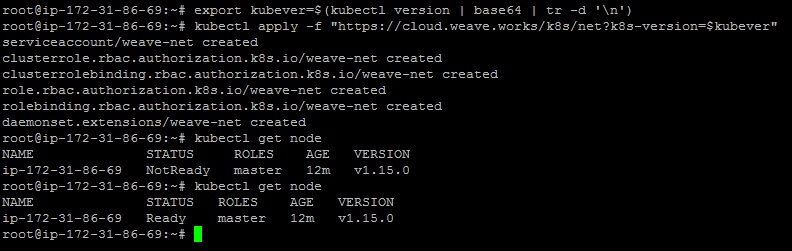
In the above output we will get kubeadm join command which will help us to add any number of worked nodes to our Kubernetes Cluster. Please remember below command to add new nodes to Kubernetes Cluster.



Once we start our cluster we need to deploy weave network to our cluster.

export kubever=$(kubectl version | base64 | tr -d '\n')

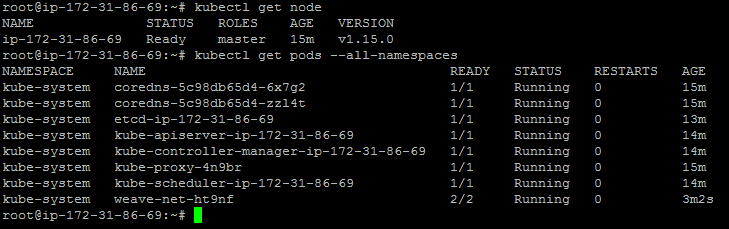
kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$kubever"



With weave network deployment we can validate if our node is up and running or not. This will help us to deploy Docker container to Kubernetes cluster.

kubectl get node

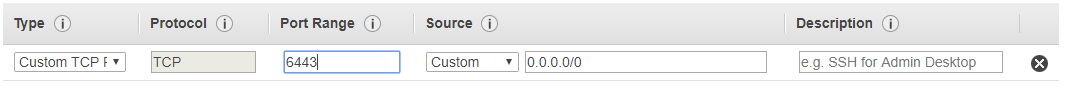
kubectl get pods --all-namespaces



**Configuring Kubernetes Slave**

Now once Kubernetes master is configured we have configure slave machine which can be attached to master server. So in case we want to deploy Kubernetes pod it can be installed easily on any of the slave machine.

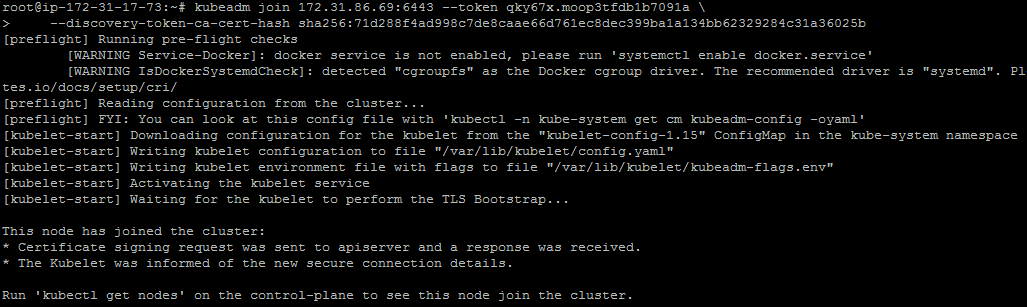
First please allow 6443 port from Kubernetes master so that we can connect to Kubernetes master from slave machine.



Next, you will need to log in to the Slave Node and add it to the Cluster. Remember the join command in the output from the Master Node initialization command and issue it on the Slave Node as shown below:

kubeadm join 172.31.86.69:6443 --token qky67x.moop3tfdb1b7091a \

--discovery-token-ca-cert-hash sha256:71d288f4ad998c7de8caae66d761ec8dec399ba1a134bb62329284c31a36025b



On Kubernetes Master we can node in Ready state once all this setup is done and we can proceed with deploying pods in this Kubernetes Cluster.

