Create the cluster

First require to install kubeadm to install in system

Step 1 : Change to root:

```
sudo su
```

Step-2 Initial the kubeadm:

```
kubeadm init --pod-network-cidr=192.168.0.0/16
```

Step -3 After a while, you will get following output:

```
Your Kubernetes master has initialized successfully!To start using your
cluster, you need to run the following as a regular user:mkdir -p
$HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/configYou should now deploy a
pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/You can
now join any number of machines by running the following on each node
as root:kubeadm join 10.91.4.105:6443 --token alpia1.8cjc1yfv5ezganq7 --
discovery-token-ca-cert-hash
sha256:3f2da2fa1967b8e974b9097fcdd15c66e0d136db5b1f08b3db7fe45c3e2b790b
```

And `exit`

Copy kubectl config:

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

Install network plugin (Calico):

```
kubectl apply -f https://docs.projectcalico.org/v3.3/getting-
started/kubernetes/installation/hosted/rbac-kdd.yaml
kubectl apply -f https://docs.projectcalico.org/v3.3/getting-
started/kubernetes/installation/hosted/kubernetes-datastore/calico-
networking/1.7/calico.yaml
```

Taint the master node for allowing deployment:

```
kubectl taint nodes --all node-role.kubernetes.io/master-
```

Now let’s try with kubectl:

```
kubectl get node
NAME                STATUS    ROLES     AGE    VERSION
my-kubernete        Ready     master    83m    v1.13.2
```

And deploy a pod:

```
kubectl run hello --image=k8s.gcr.io/echoserver:1.4 --
port=8080
kubectl get pod
NAME                        READY    STATUS     RESTARTS    AGE
hello-5975cd9c9d-5pvsn      1/1      Running    0           120m
```

Step 4: Trouble shooting

If found the core dns is always crashed like that:

```
$ kubectl get pod -n kube-system

NAME                                READY        STATUS                  RESTARTS     AGE
coredns-86c58d9df4-hrp2w             0/1          CrashLoopBackOff        542           1d
coredns-86c58d9df4-ptgsk             0/1          CrashLoopBackOff        543           1d
```

Manually modify or Delete config file --: /etc/resolv.conf

```
nameserver 8.8.4.4
nameserver 8.8.8.8
```

Commands:

Step-1 Master Node Upgrade command

```
→kubectl get nodes

→apt-get upgrade -y kubelet=1.12.0-00

→systemctl restart kubelet

→kubectl get nodes
```

Step-2 Worker node upgrade command :

```
→kubectl drain node1-name
```

So node1- all load move to another nodes

After that you can use worker node upgrade command.

```
→apt-get upgrade -y kubeadm=1.12.0-00

→apt-get upgrade -y kubelet=1.12.0-00

→kubeadm upgrade node config  --kubelet-version v1.12.0

→systemctl restart kubelet
```

Step-3 After that done to convert Uncordon

```
→ kubectl uncordon node-1
```

Same process for All node-2 and node-3 and more nodes