**Upgrade the Control plane**

* Drain the node kubectl drain master --ignore-daemonsets --force
* Verify the kubeadm Version kubeadm version
* Upgrade apt-get update && apt-get install -y --allow-change-held-packages kubeadm=1.22.2-00
* Plan the upgrade kubeadm upgrade plan
* Plan the upgrade to a specific version  kubeadm upgrade plan v1.22.2
* upgrade the control plan components  kubeadm upgrade apply v1.22.2
* Upgrade the Kubelet and Kubectl on the control plane apt-get install -y kubelet=1.22.2-00 kubectl=1.22.2-00
* Restart the kubelet sudo systemctl daemon-reload && sudo systemctl restart kubelet
* Verify the nodes kubectl get nodes
* uncordon the control plan node  kubectl uncordon master

**Upgrade the WorkerNode**

* Drain the worker node (Note: use kubectl drain wokernode, command from master) kubectl drain worker-1
* Upgrade Kubeadm on worker node 1: ```apt-get update && apt-get install -y kubeadm=1.22.2-00``
* Upgrade the kubelete configuraiton on worker node  kubeadm upgrade node
* Upgrade kubectl and kubelet components on worker node1 apt-get install -y kubelet=1.22.2-00 kubectl=1.22.2-00
* Restart the kubelet sudo systemctl daemon-reload && sudo systemctl restart kubelet
* Verify the nodes kubectl get nodes
* uncordon the Worker node (Note: You need to login to master to execute the uncordon command) kubectl uncordon worker-1

## Create a private GKE Cluster

gcloud config set compute/zone us-central1-a

# Create a private cluster

gcloud beta container clusters create my-private-cluster \

--enable-private-nodes \

--master-ipv4-cidr 172.16.0.16/28 \

--enable-ip-alias \

--machine-type "e2-small" \

--num-nodes "1" \

--disk-size "50"

--create-subnetwork ""

# Verify a new subnet should be created in the default VPC

## Create a public GKE Cluster

# Create a public cluster

gcloud beta container clusters create my-public-cluster \

--machine-type "e2-small" \

--num-nodes "1" \

--disk-size "50"

## Create a VM - Jump server to connect to the gke cluster

# Creata a JumpServer

gcloud compute instances create jump-server --zone us-central1-a --scopes 'https://www.googleapis.com/auth/cloud-platform'

## Run the following to Authorize your external address range,

gcloud container clusters update my-private-cluster \

--enable-master-authorized-networks \

--master-authorized-networks {JUMPSERVER\_PUBLIC\_IP]/32

# Login to jumpserver

gcloud compute ssh jump-server --zone us-central1-a

#install kubectl

sudo apt-get install kubectl

# Confgure the config file using gcloud command

gcloud container clusters get-credentials my-private-cluster --zone us-central1-a

#kubectl get nodes

# Create a deployment to test the access to private gke from customers

kubectl create deploy siva-nginx --image nginx

#Expose the deploy to access

kubectl expose deploy siva-nginx --port 80 --type LoadBalancer