* **Install Jenkins on VM machine**
* Install java using following command

sudo apt-get update

sudo apt-get install openjdk-8-jre-headless -y

* Install Jenkins

wget -q -O - <https://pkg.jenkins.io/debian/jenkins-ci.org.key> | sudo apt-key add -

sudo sh -c 'echo deb [http://pkg.jenkins.io/debian-stable binary/](http://pkg.jenkins.io/debian-stable%20binary/) > /etc/apt/sources.list.d/jenkins.list'

sudo apt-get update

sudo apt-get install jenkins -y

* Right after we install Jenkins, we can open a browser and go to [http://your\_ip\_address:8080](http://your_ip_address:8080/)
* Unlock jenkins:

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

* Create GKE cluster in GCP:
* Create Service account of jenkins inside Some namesapece (Here I have used namespace jenkins-sa)

kubectl create serviceaccount jenkins -n jenkins-sa

* Create Role

cat > jenkins-role.yaml << EOF

---

kind: Role

apiVersion: rbac.authorization.k8s.io/v1beta1

metadata:

name: jenkins

rules:

- apiGroups: [""]

resources: ["pods"]

verbs: ["create","delete","get","list","patch","update","watch"]

- apiGroups: [""]

resources: ["pods/exec"]

verbs: ["create","delete","get","list","patch","update","watch"]

- apiGroups: [""]

resources: ["pods/log"]

verbs: ["get","list","watch"]

- apiGroups: [""]

resources: ["secrets"]

verbs: ["get"]

EOF

kubectl apply -f jenkins-role.yaml -n jenkins-sa

* Create Role Binding

cat > jenkins-rolebinding.yaml << EOF

---

apiVersion: rbac.authorization.k8s.io/v1beta1

kind: RoleBinding

metadata:

name: jenkins

namespace: jenkins-sa

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: Role

name: jenkins

subjects:

- kind: ServiceAccount

name: jenkins

EOF

kubectl apply -f jenkins-rolebinding.yaml -n jenkins-sa

--------------------------------------------------------------------------------

\*\*\* or use following to create service acc role and role binding with admin access :

kubectl create namespace jenkins-sa && kubectl create serviceaccount jenkins --namespace=jenkins-sa && kubectl describe secret $(kubectl describe serviceaccount jenkins --namespace=jenkins-sa | grep Token | awk '{print $2}') --namespace=jenkins-sa && kubectl create rolebinding jenkins-admin-binding --clusterrole=admin --serviceaccount=jenkins:jenkins --namespace=jenkins-sa

(or refer <https://stackoverflow.com/questions/58410484/how-to-configure-permissions-on-jenkins-kubernetes-plugin-to-access-remote-gke-c>)

\*\*this will also create token which will save as secret text in credentials.

\*\*\*

* Install Kubernetes plugin in jenkins

Jenkins -> Manage Jenkins ->Manage Plugin -> Kubernetes (under available section)



* Configure Pluging

Jenkins -> Manage Jenkins ->configure system -> (in the bottom)select cloud kubernetes



* Cloud setup:
* Add kubernetes url generated by following command

kubectl config view --minify | grep server

* Add Kubernetes server certificate key generated by following command

kubectl get secret $(kubectl get sa jenkins -n jenkins-sa -o jsonpath={.secrets[0].name}) -n jenkins-sa -o jsonpath={.data.'ca\.crt'} | base64 --decode

* Add kubernetes namespace as mentioned while creating service account of jenkins(jenkins-sa)
* In creadentials go to

Add -> jenkins -> In dropdown select ‘secret text’

Add secret text generated by following command

kubectl get secret $(kubectl get sa jenkins -n jenkins-sa -o jsonpath={.secrets[0].name}) -n jenkins-sa -o jsonpath={.data.token} | base64 --decode

* Select that credential and click on Test Connection



It must have to show Connection test successful.

* Add Jenkins Url and Jenkins Tunnel as shown:



* Click on Add Pod Label

Give label and Value as follow:



You can give any label as you want

* Click on Add Pod Template:

Give name and label as shown:



You can also give namespace if not given it will start slave pod on default namespace.

* Click On Add Container -> Container template

Add the details as shown below



We can change the Docker Image according to our use.

* Now click on Apply and Save
* You are now ready to use Jenkins master-slave setup on GKE.

Note: You need to set TCP port for inbound agents under Jenkins -> [**Configure Global Security**](http://35.222.91.237:8080/configureSecurity/) -> TCP port for inbound as shown :

