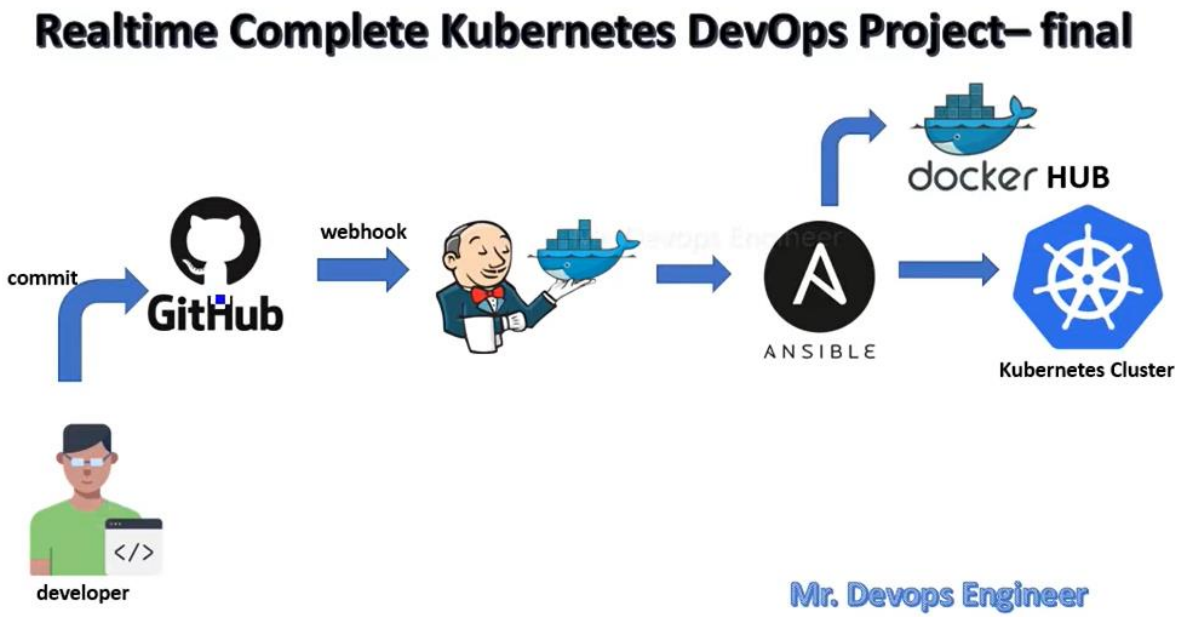


Now ansible will run ansible.yml playbook ,

service.yml and deployment.yml will run on Kubernetes node which will be triggered from ansible .

Kubernetes cluster will pull latest image from docker and will build container using image .
container will be accessible to user from ip port which we will expose .

In deployment.yml use same image name as the one in docker hub .
We will not provide any tag so that it will use latest tag image .
Containerport to specify what port we are exposing for container .
IN service.yml we can define port details targetport is same as conatinerport. node port will also be specified
We will be using service type as load balancer

In ansible.yml use Kubernetes Ip as host.
Now Jenkins will transfer service.yml, deployment.yml and ansible.yml file also to ansible server .
Now let's make ssh connection between ansible and Kubernetes server , so that ansible can run playbook on Kubernetes node as remote host .

Install minikube .
Update password for user ubuntu in kubernetes server :

```
ubuntu@ip-172-31-12-252:~$ sudo su
root@ip-172-31-12-252:/home/ubuntu# passwd ubuntu
New password:
Retype new password:
passwd: password updated successfully
root@ip-172-31-12-252:/home/ubuntu#
```

Set password for root and ubuntu user in kubernetes and enable root login also . -

```
root@ip-172-31-21-216:/home/ubuntu# passwd ubuntu
New password:
Retype new password:
passwd: password updated successfully
root@ip-172-31-21-216:/home/ubuntu# passwd root
New password:
Retype new password:
passwd: password updated successfully
root@ip-172-31-21-216:/home/ubuntu#
```

Enable root login :
Vi /etc/ssh/sshd_config – in this file

```
#LoginGraceTime 2m
PermitRootLogin yes
```

Enable password authentication :

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
```

Restart server : service sshd restart

Set passwordless between ansible and Kubernetes so that we can integrate in pipeline .

Generate ssh key for ansible server – ssh-keygen

```
ubuntu@ip-172-31-48-163:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa): /home/ubuntu/.ssh/id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:+kb23KlPICFlCaII3njL7COjMuM5R8C/9346RXa5V3A ubuntu@ip-172-31-48-163
The key's randomart image is:
+---[RSA 3072]----+
|.   . ..o.       |
|o.o. . o.    . E |
|oo.o  . .   . o  |
| o+ .  . + o   . |
|  o+    S o . .  |
|  .o   .oo o .   |
| o.o. .o.o .o.   |
|=.+o.. o..o.o    |
|+=o . o== .o.    |
+----[SHA256]-----+
```

Copy public key of ansible server to Kubernetes server .

ssh-copy-id remoteuser@remoteserver

```
ubuntu@ip-172-31-48-163:~$ ssh-copy-id ubuntu@54.242.193.200
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ubuntu/.ssh/id_rsa.pub"
The authenticity of host '54.242.193.200 (54.242.193.200)' can't be established.
ED25519 key fingerprint is SHA256:kalUQumlfdG35txsatllsUClZEbJQtoAZhTTg+ouu3Q.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ubuntu@54.242.193.200's password:
```

Passwordless connection done .
Now create inventory file and add Kubernetes ip as node .

```
ubuntu@ip-172-31-48-163:~/ansible$ cat hosts
[node]
54.242.193.200
ubuntu@ip-172-31-48-163:~/ansible$
```

Test ansible ping to Kubernetes node ,

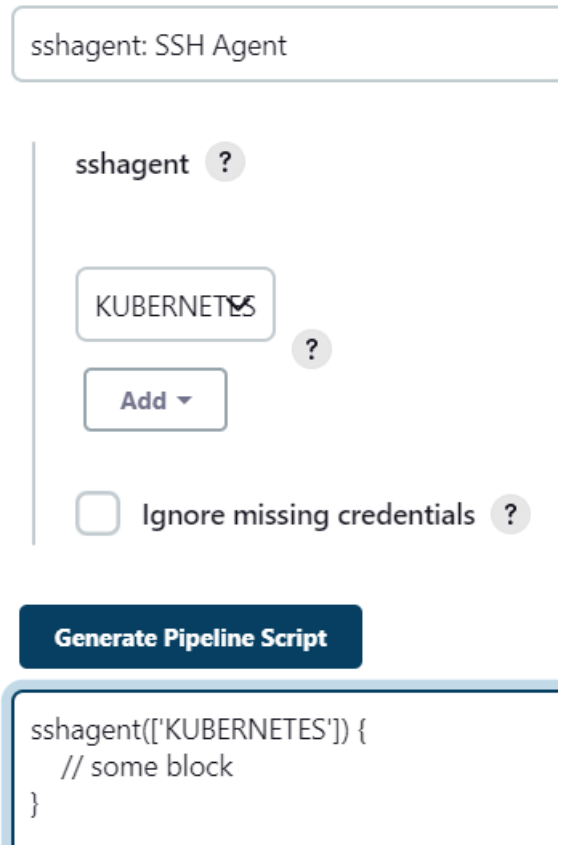
```
ubuntu@ip-172-31-48-163:~/ansible$ ansible -i hosts -m ping node
54.242.193.200 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

It is successful .

Transfer service.yml and deployment.yml to Kubernetes server .

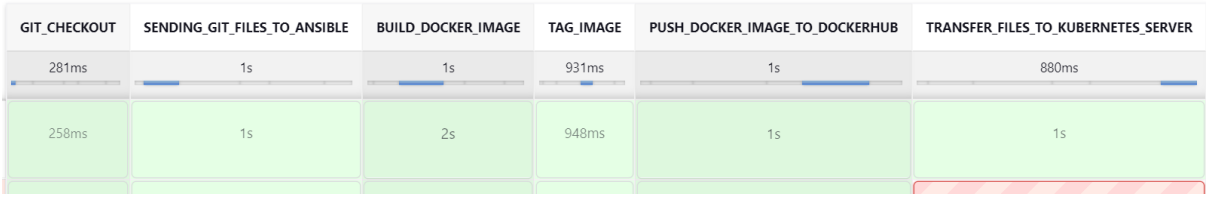
STAGE 6 : Transfer file from Jenkins to Kubernetes server .

Generate pipeline syntax by adding kubernets node as ssh agent in similar way as we did for ansible .



Pipeline script :

```
}
    stage('TRANSFER_FILES_TO_KUBERNETES_SERVER'){
        sshagent(['KUBERNETES']) {
            sh 'ssh -T -o StrictHostKeyChecking=no ubuntu@54.242.193.200'
            sh 'scp /var/lib/jenkins/workspace/KUBERNETES_DEPLOYMENT_PIPELINE/* ubuntu@54.242.193.200:/home/ubuntu'
        }
    }
}
```



we can delete local images from docker after images pushed to docker hub – update in push to docker hub stage of pipeline .

```
}
stage('PUSH_DOCKER_IMAGE_TO_DOCKERHUB'){
    sshagent(['ANSIBLE']) {
    withCredentials([string(credentialsId: 'DOCKERHUB', variable: 'DOCKERHUB')]) {
        sh '''
        ssh -T -o StrictHostKeyChecking=no ubuntu@18.204.10.209 docker login -u sravtar -p Sapna@yadav2612
        JOB_SMALL=$(echo "$JOB_NAME" | tr '[:upper:]' '[:lower:]')
        ssh -T -o StrictHostKeyChecking=no ubuntu@18.204.10.209 docker image push sravtar/$JOB_SMALL:v1.$BUILD_ID
        ssh -T -o StrictHostKeyChecking=no ubuntu@18.204.10.209 docker image push sravtar/$JOB_SMALL:v1.latest
        ssh -T -o StrictHostKeyChecking=no ubuntu@18.204.10.209 docker image rm sravtar/$JOB_SMALL:v1.$BUILD_ID sravtar/$JOB_SMALL:v1.la
        '''
}
```

Now add stage to execute playbook in ansible server

STAGE 7 :

Add service.yml , deployment.yml and ansible.yml file to github .

Deployment.yml	Create Deployment.yml	now
Dockerfile	Update Dockerfile	yesterday
README.md	Update README.md	5 days ago
Service.yml	Create Service.yml	2 minutes ago
ansible.yml	Create ansible.yml	3 minutes ago

Update image name , app name .

```
    }
    stage('KUBERNETES_DEPLOYMENT_USING_ANSIBLE'){
        sshagent(['ANSIBLE']) {
            sh 'ssh -T -o StrictHostKeyChecking=no ubuntu@18.204.10.209 cd /home/ubuntu/ansible'
            sh 'ssh -T -o StrictHostKeyChecking=no ubuntu@18.204.10.209 ansible-playbook -i hosts ansible.yml'
        }
    }
}
```

Pipeline script .

GIT_CHECKOUT	SENDING_GIT_FILES_TO_ANSIBLE	BUILD_DOCKER_IMAGE	TAG_IMAGE	PUSH_DOCKER_IMAGE_TO_DOCKERHUB	TRANSFER_FILES_TO_KUBERNETES_SERVER	KUBERNETES_DEPLOYMENT_USIN
299ms	1s	1s	926ms	1s	1s	1s
378ms	1s	2s	935ms	1s	1s	1s

Pipeline is success .

```
1  - hosts: all
2    become: true
3    tasks:
4       - name: delete old deployment
5         command: kubectl delete -f /home/ubuntu/Deployment.yml
6       - name: delete old service
7         command: kubectl delete -f /home/ubuntu/Service.yml
8       - name: create new deployment
9         command: kubectl apply -f /home/ubuntu/Deployment.yml
10      - name: create new service
11        command: kubectl apply -f /home/ubuntu/Service.yml
```

In ansible.yml remove delete scripts and add it after first deployment .
With ansible playbook it was not running so I changed pipeline script .

```
ubuntu@ip-172-31-21-216:~$ kubectl get pods
NAME                     READY   STATUS    RESTARTS   AGE
sapna-74cd9d86b9-8xc7w   1/1     Running   0          9m7s
sapna-74cd9d86b9-t25mj   1/1     Running   0          9m7s
ubuntu@ip-172-31-21-216:~$ kubectl get svc
NAME         TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
kubernetes   ClusterIP      10.96.0.1        <none>        443/TCP          94m
sapna        LoadBalancer   10.101.241.209   <pending>     8080:31200/TCP   4m41s
ubuntu@ip-172-31-21-216:~$
```

Service is getting mapped to Kubernetes ip and 31200 port , enable in security group .

Once done check in Kubernetes server
Kubectl get all
Kubectl get pods
Kubectl get svc

Also if we push new docker file then also old service.yml and deployment will be used for creating new pod since we have not added any code to delete these files from Kubernetes server and ansible server .

So update ansible.yml to delete old service and deployment files .
And now new image will be run as pod in Kubernetes and service will be accessible .