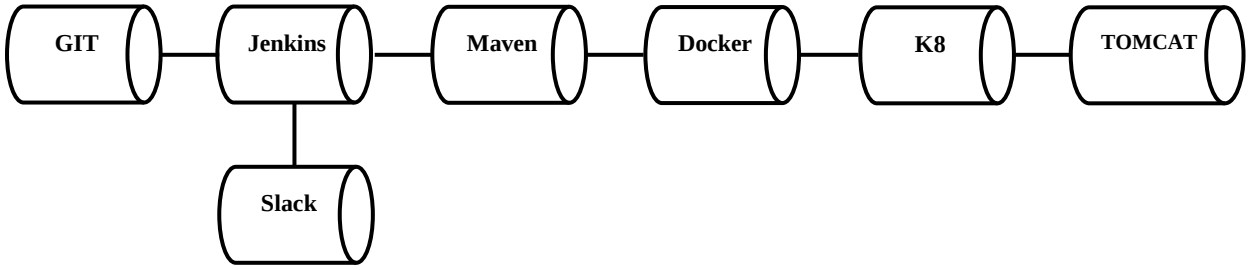
DAP4.0 – DevOps PoC-2

Employee Id:	353882
Employee Name:	Srinivas Inampudi
Program:	DAP4.0 – DevOps
Mentor Name:	Anindya Bhattacharya
Use Case:	Containerization and Orchestration

Table of Contents

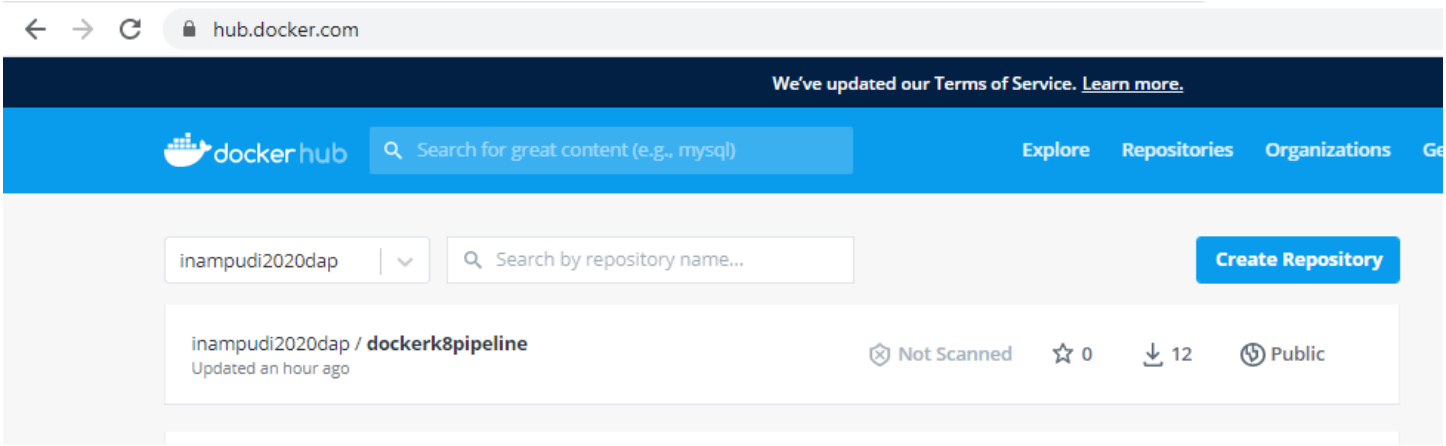
DAP4.0 – DevOps PoC-2

1. Docker Kubernetes Pipeline Diagram



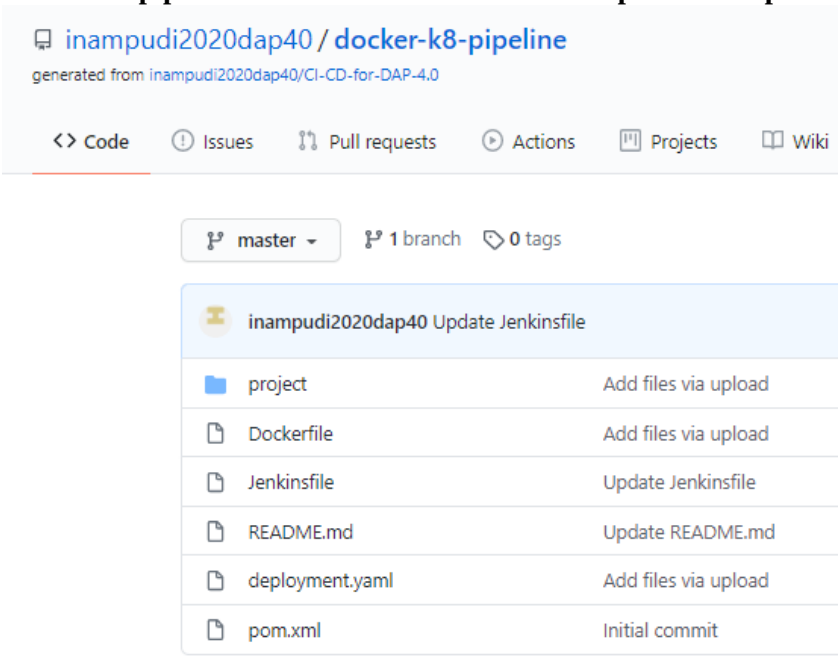
2. Docker Hub Account

Created a Docker Hub account



3. Create GIT HUB Account/Repo

Created a new repository **docker-k8-pipeline** under GIT Hub account – **inampudi2020dap40**.



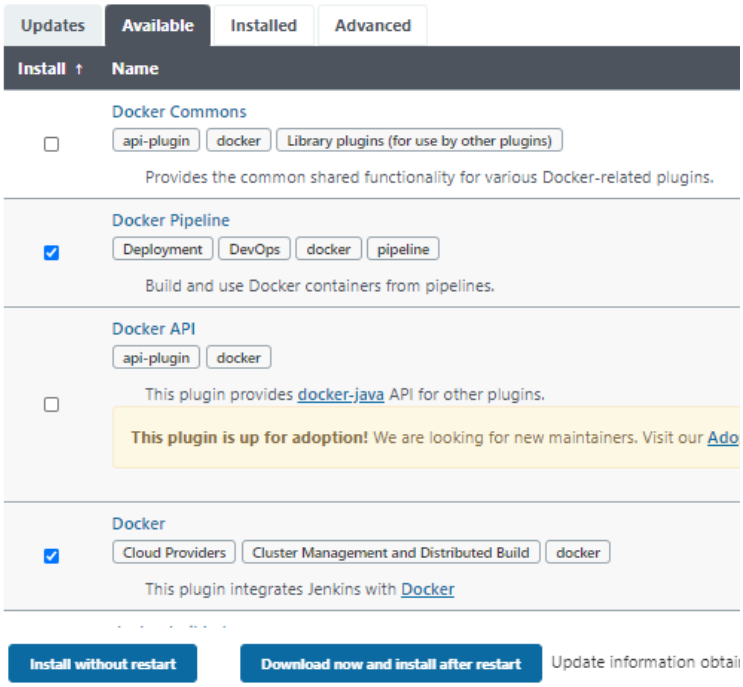
4. Jenkins Integration

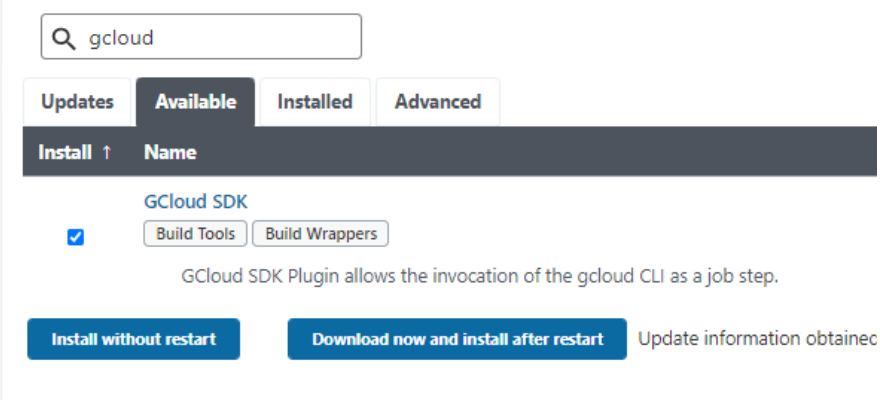
Reuse Jenkins setup from PoC-1

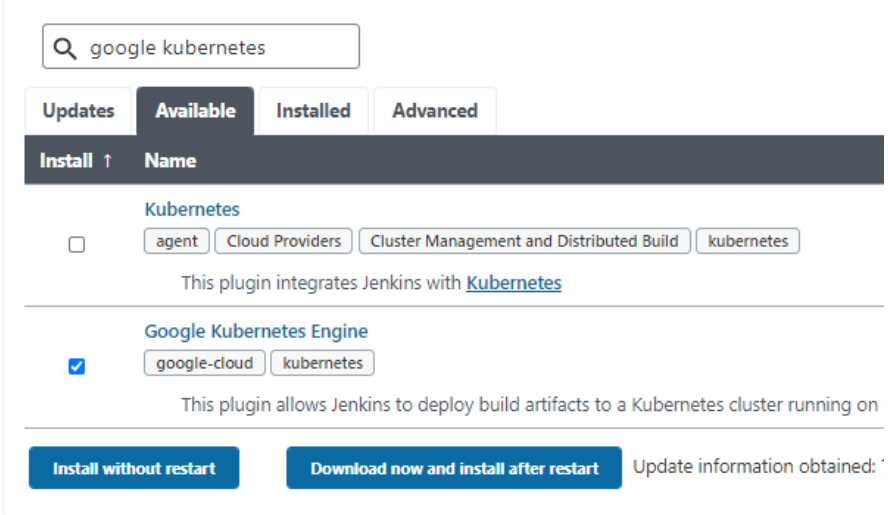
Jenkins URL: http://35.222.150.4:8080/

Install Plug-ins that are required for Docker and Kubernetes Integration

- Install below plug-ins in Jenkins
  - Docker
  - Docker Pipeline
  - GCloud SDK
  - Google Kubernetes Engine

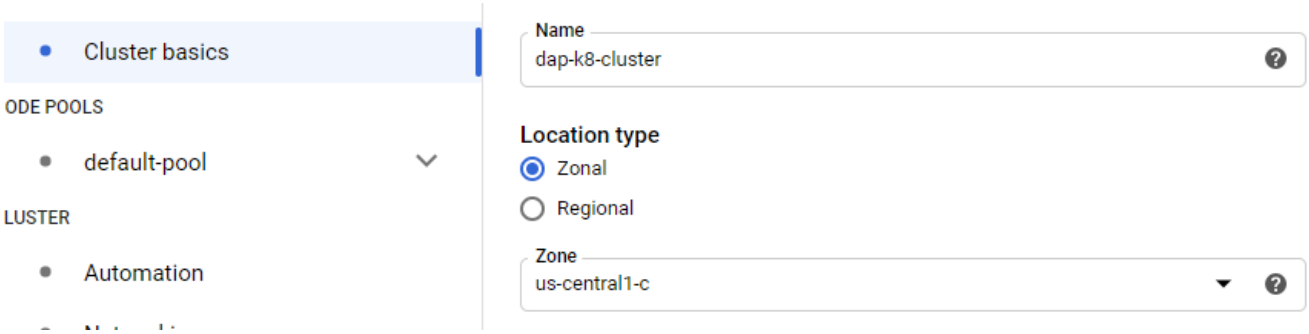




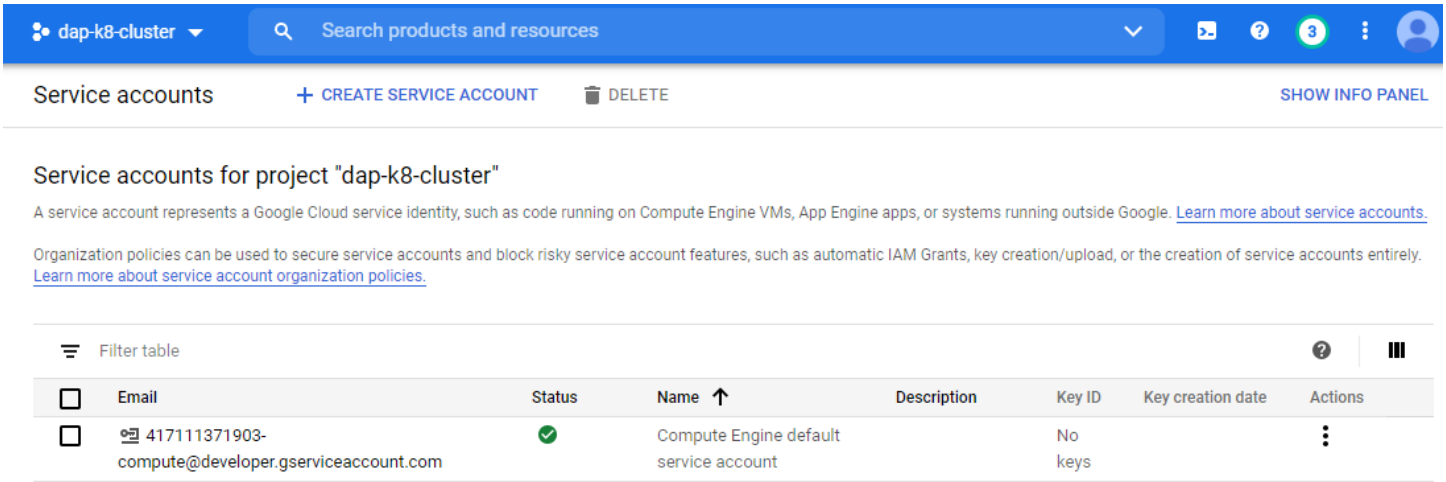


K8 integration with Jenkins

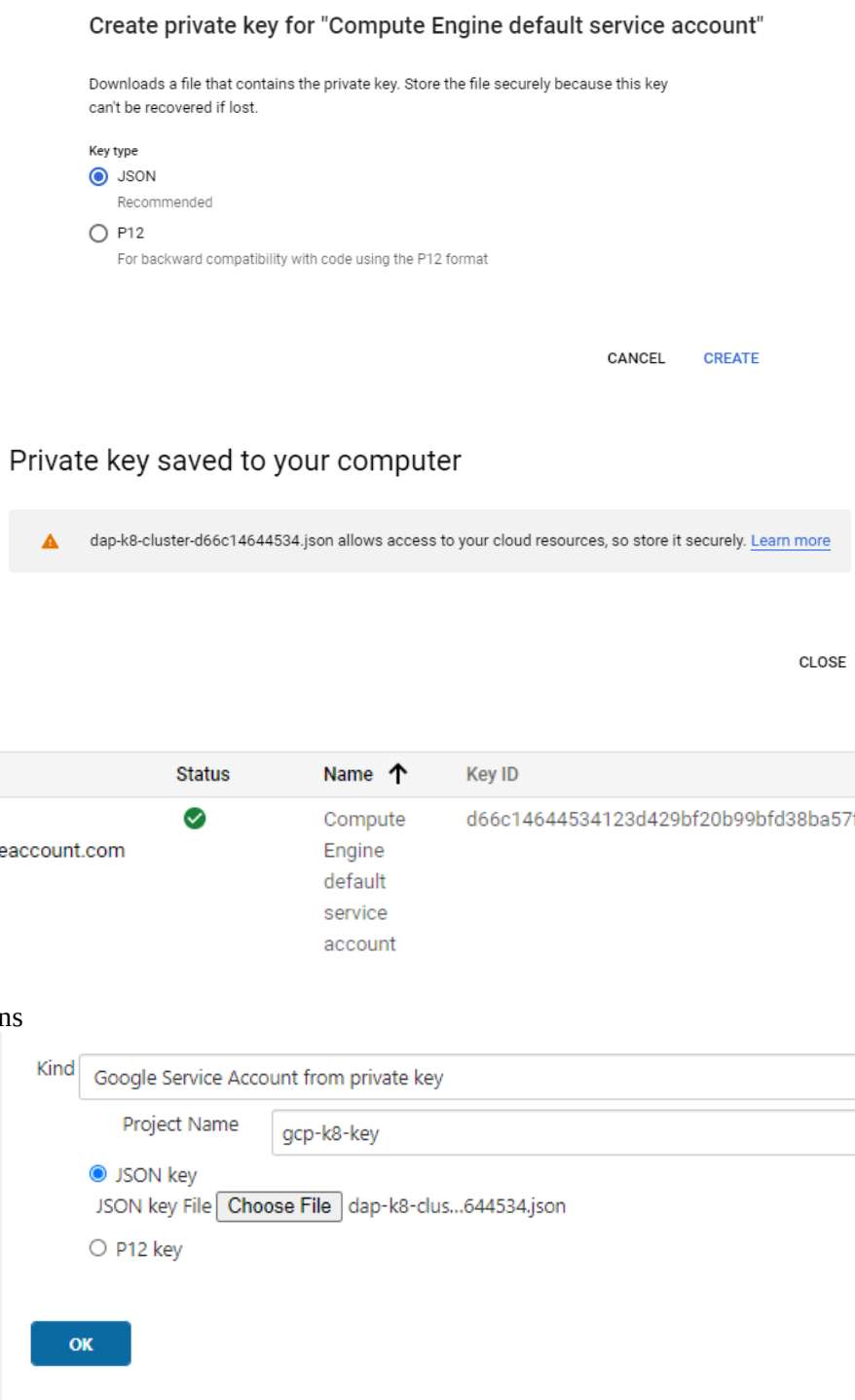
Created a Kubernetes cluster - **`dap-k8-cluster`**



Generate a Service account for Jenkins and K8 integration



Generate a JSON Key and download it to local



Upload the key to Jenkins

Install Docker and Kubernetes on Jenkins Server

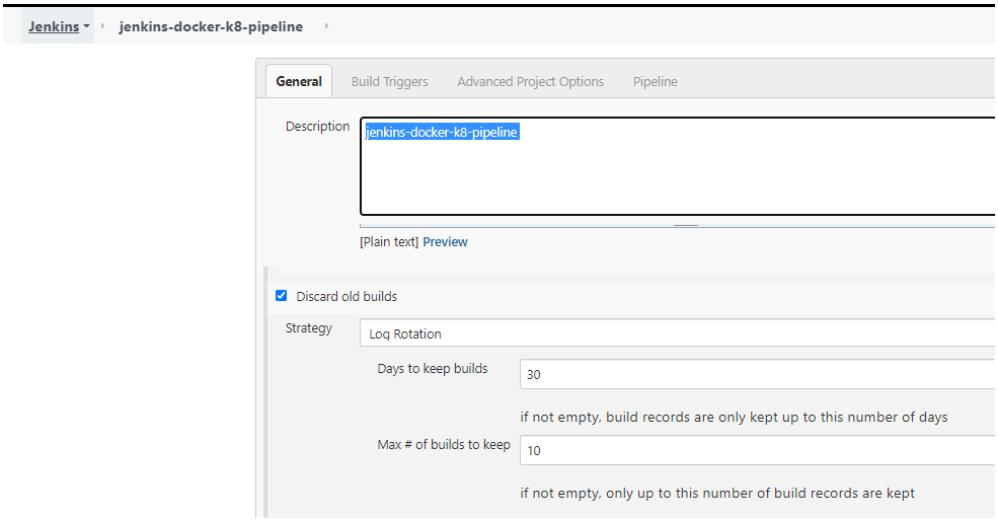
```
inampudi_2020_dap4_0@maven-jenkins-vm:~$ sudo curl -sSL https://get.docker.io/ | sh
# Executing docker install script, commit: 26ff363bcf3b3f5a00498ac43694bf1c7d9ce16c
+ sudo -E sh -c apt-get update -qq >/dev/null
+ sudo -E sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-htt
+ sudo -E sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" | apt-key add
+ sudo -E sh -c echo "deb [arch=amd64] https://download.docker.com/linux/ubuntu xenial
+ sudo -E sh -c apt-get update -qq >/dev/null
+ [ -n  ]
+ sudo -E sh -c apt-get install -y -qq --no-install-recommends docker-ce >/dev/null
+ sudo -E sh -c docker version
Client: Docker Engine - Community
 Version:           19.03.13
 API version:       1.40
 Go version:        go1.13.15
 Git commit:        4484c46d9d
 Built:             Wed Sep 16 17:02:59 2020
 OS/Arch:           linux/amd64
 Experimental:      false
```

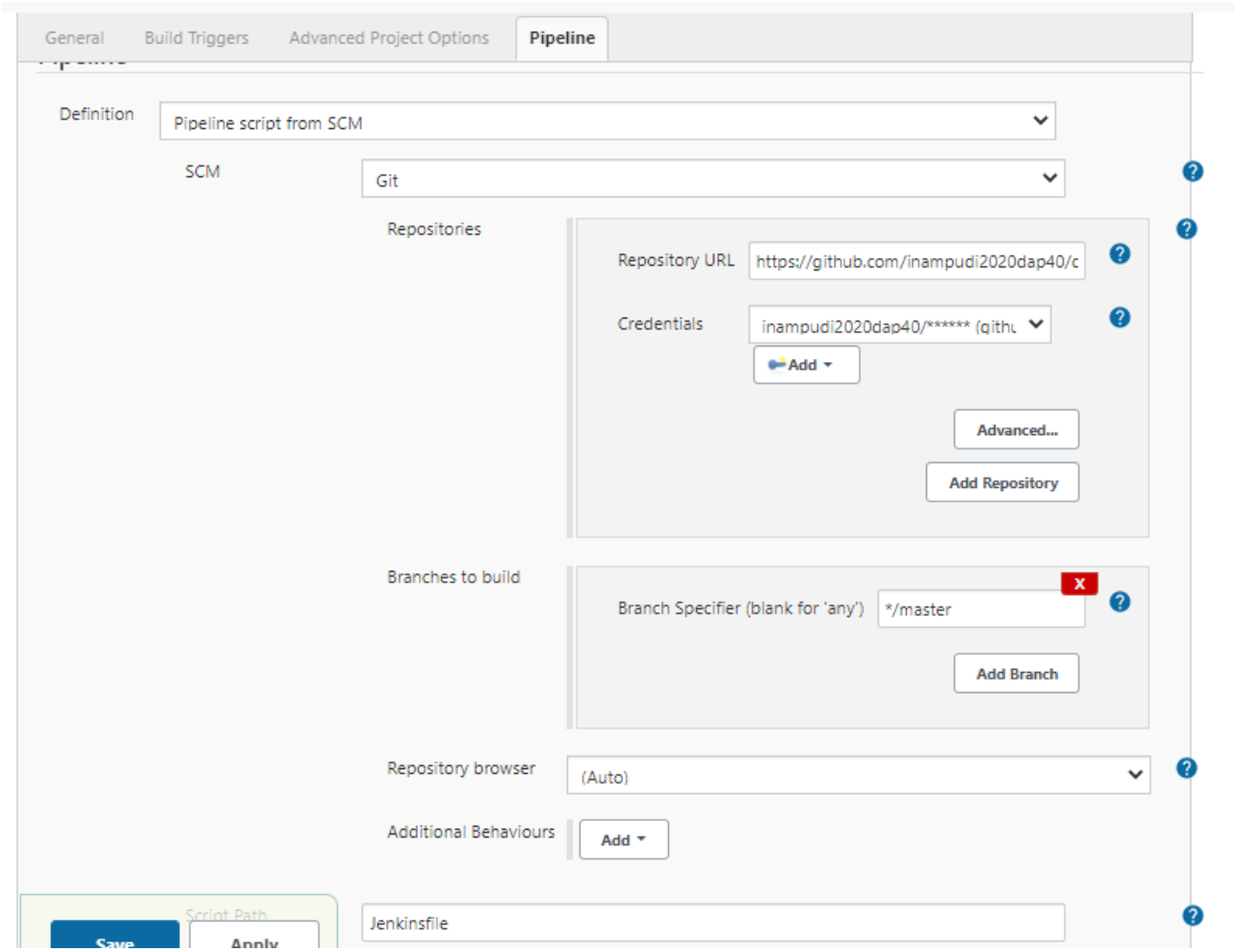
```
inampudi_2020_dap4_0@maven-jenkins-vm:~$ sudo usermod -aG docker inampudi_2020_dap4_0
inampudi_2020_dap4_0@maven-jenkins-vm:~$ sudo usermod -a -G root jenkins
inampudi_2020_dap4_0@maven-jenkins-vm:~$ sudo service jenkins restart
inampudi_2020_dap4_0@maven-jenkins-vm:~$
```

```
inampudi_2020_dap4_0@maven-jenkins-vm:~$ sudo snap install kubectl --classic
2020-10-10T06:11:34Z INFO Waiting for automatic snapd restart...
kubectl 1.19.0 from Canonical✓ installed
inampudi_2020_dap4_0@maven-jenkins-vm:~$ sudo chmod 777 /var/run/docker.
docker.pid   docker.sock
inampudi_2020_dap4_0@maven-jenkins-vm:~$ sudo chmod 777 /var/run/docker.sock
inampudi_2020_dap4_0@maven-jenkins-vm:~$
```

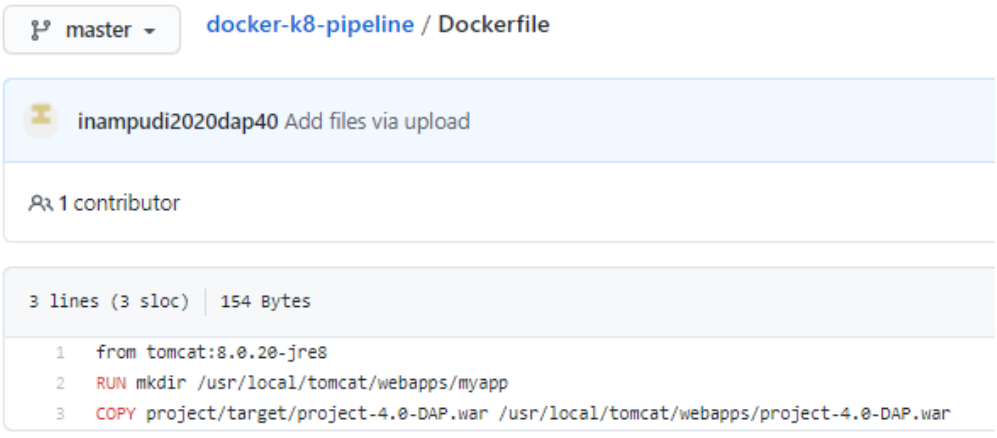
5. Docker Kubernetes Pipeline Job in Jenkins

Created a Pipeline Job - jenkins-docker-k8-pipeline





Add below files to GIT Repo
Dockerfile
Jenkinsfile
deployment.yaml



master ▾ docker-k8-pipeline / deployment.yaml

inampudi2020dap40 Add files via upload

1 contributor

34 lines (34 sloc) | 609 Bytes

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: tomcatpod
spec:
  replicas: 3
  selector:
   matchLabels:
    app: tomcatpod
  template:
    metadata:
      labels:
        app: tomcatpod
    spec:
      containers:
      - name: tomcatpod
        image: inampudi2020dap/dockerk8pipeline:tagversion
        ports:
        - containerPort: 8080
```

master ▾ docker-k8-pipeline / Jenkinsfile

inampudi2020dap40 Update Jenkinsfile

1 contributor

76 lines (72 sloc) | 2.71 KB

```
pipeline {
        agent any
        environment {
            registry = "inampudi2020dap/dockerk8pipeline"
            registryCredential = 'docker'
            dockerImage = ''
                PROJECT_ID = 'dap-k8-cluster'
                CLUSTER_NAME = 'dap-k8-cluster'
                LOCATION = 'us-central1-c'
                CREDENTIALS_ID = 'gcp-k8-key'
        }

            stages {
                   stage('Scm Checkout') {
                        steps {
                     checkout scm
                                }
                   }
```

Docker and Kubernetes Stages

```
        stage('Build Docker Image') {
             steps {
                script {
                   dockerImage = docker.build registry + ":$BUILD_NUMBER"
                   slackSend channel: 'ci-cd-pipeline', color: '#BADA55', message: 'Build Docker Image', teamDomain: 'dap40devops', tokenCredentialId: 'slack'
                }
             }
        }
    stage('Deploy Image') {
        steps{
            script {
            docker.withRegistry( '', registryCredential ) {
                dockerImage.push()
                slackSend channel: 'ci-cd-pipeline', color: '#BADA55', message: 'Deploy Image', teamDomain: 'dap40devops', tokenCredentialId: 'slack'
            }
            }
        }
    }
        stage('Deploy to GKE') {
                steps{
                        echo "Deployment started"
                        sh 'ls -ltr'
                        sh 'pwd'
                        sh "sed -i 's/tagversion/${env.BUILD_ID}/g' deployment.yaml"
                        step([$class: 'KubernetesEngineBuilder', projectId: env.PROJECT_ID,
                              clusterName: env.CLUSTER_NAME, location: env.LOCATION, manifestPattern: 'deployment.yaml',
                              credentialsId: env.CREDENTIALS_ID, verifyDeployments: true])
                        echo "Deployment Finished"
                        slackSend channel: 'ci-cd-pipeline', color: '#BADA55', message: 'Deploy to GKE', teamDomain: 'dap40devops', tokenCredentialId: 'slack'
```

Slack Integration Steps

```
          post {
                   success {
slackSend (color: '#00FF00', message: "SUCCESSFUL: Job '${env.JOB_NAME} [${env.BUILD_NUMBER}]' (${env.BUILD_URL})")
                                }

                   failure {
slackSend (color: '#FF0000', message: "FAILED: Job '${env.JOB_NAME} [${env.BUILD_NUMBER}]' (${env.BUILD_URL})")
                                }
                   }
```

6. Validation Screen Shots

Run the Build and check the stage wise status

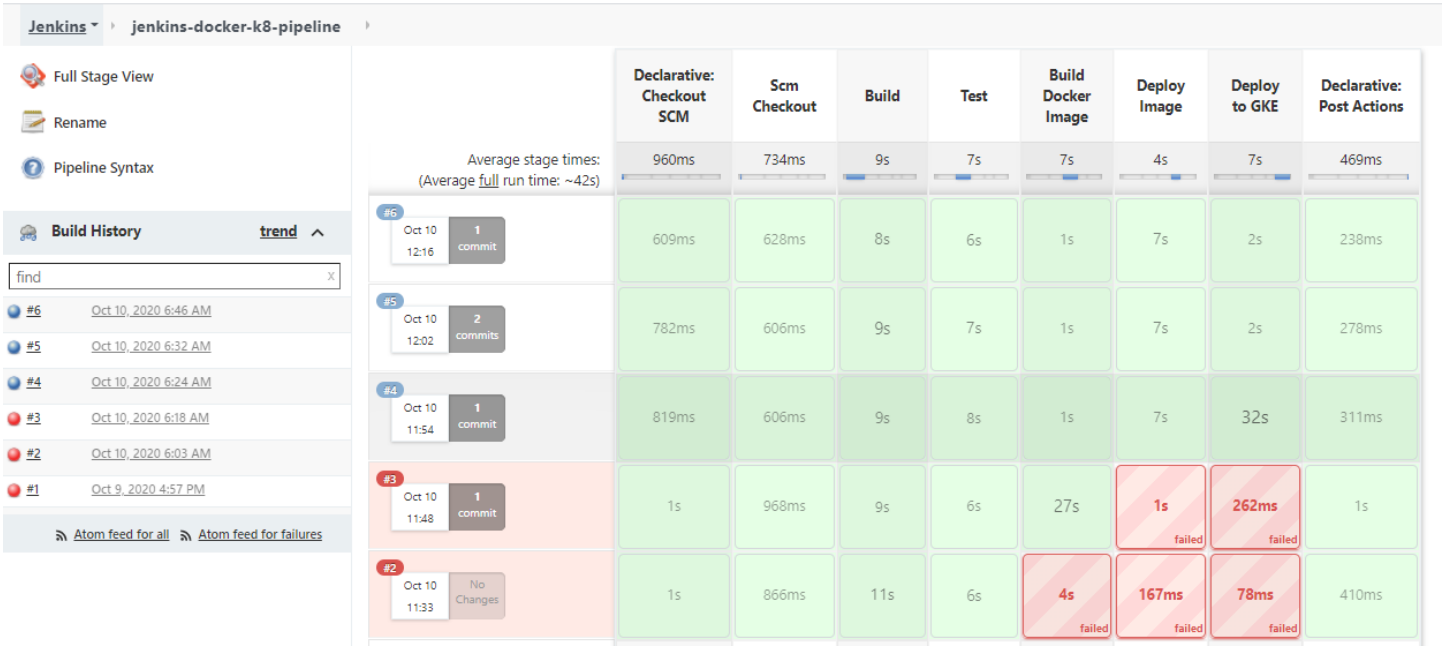
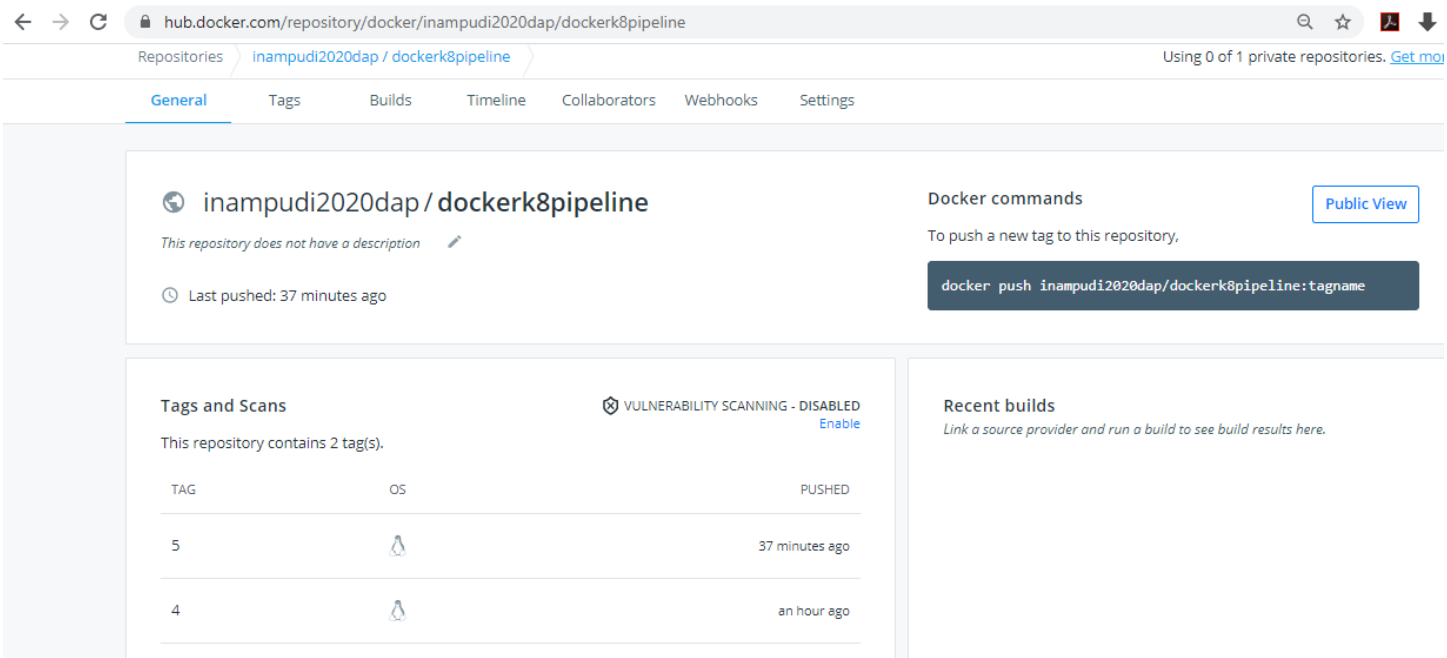
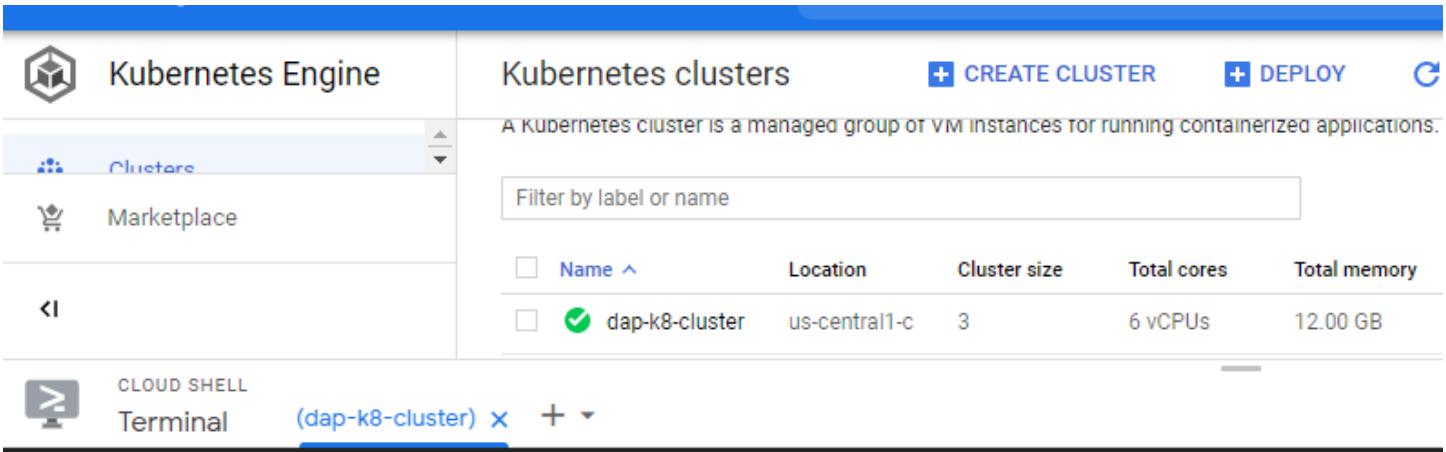


Image has been pushed to Docker



PODs have been created in K8 cluster



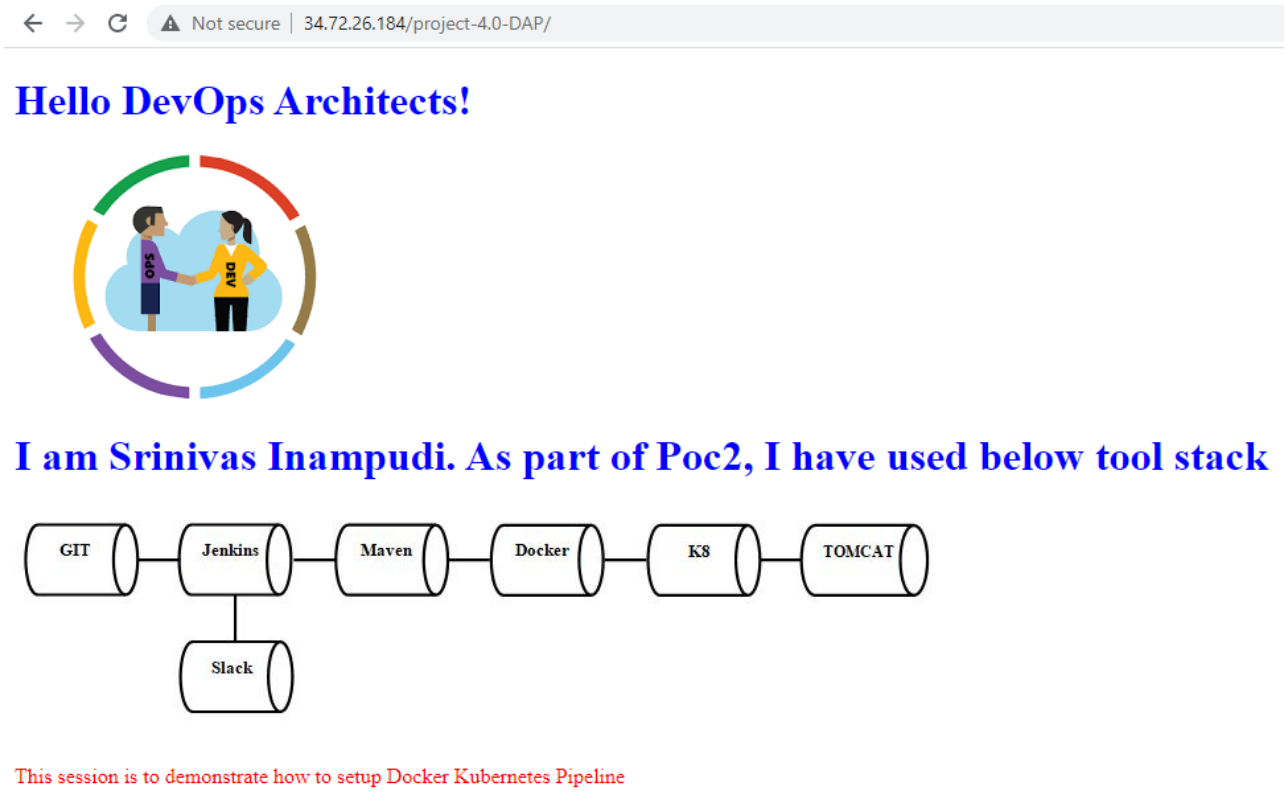

```
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
inampudi_2020_dap4_0@cloudshell:~ (dap-k8-cluster)$ kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
tomcatpod-5666f5d5d8-4grsg    1/1     Running   0          7m18s
tomcatpod-5666f5d5d8-dd2t6    1/1     Running   0          7m21s
tomcatpod-5666f5d5d8-gg9kn    1/1     Running   0          7m15s
```

Get the External IP from the tomcat service

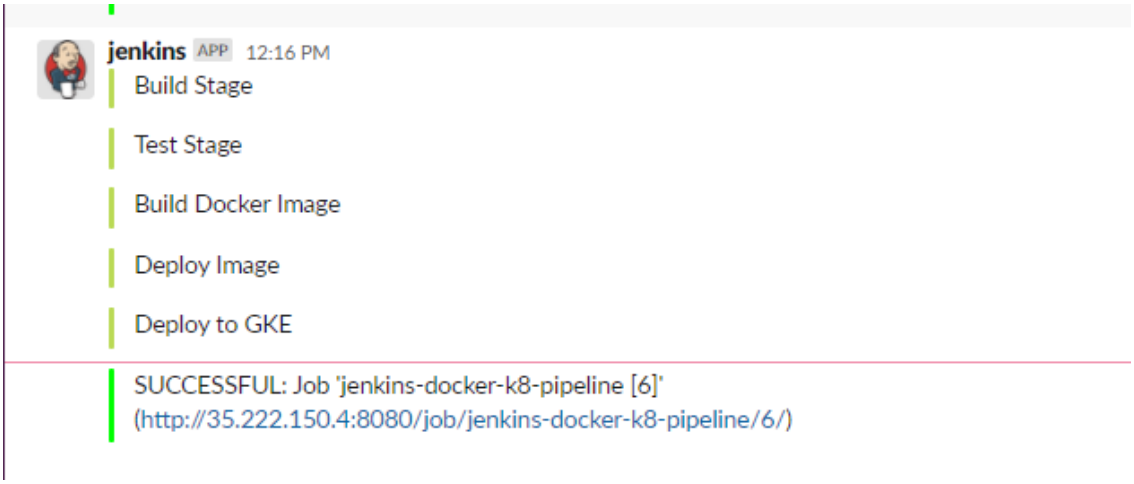
```
inampudi_2020_dap4_0@cloudshell:~ (dap-k8-cluster)$ kubectl get services
NAME            TYPE           CLUSTER-IP     EXTERNAL-IP    PORT(S)        AGE
kubernetes      ClusterIP      10.104.0.1     <none>         443/TCP        10h
tomcatpod-svc   LoadBalancer   10.104.8.130   34.72.26.184   80:32310/TCP   29m
```

```
inampudi_2020_dap4_0@cloudshell:~ (dap-k8-cluster)$ kubectl get all
NAME                              READY   STATUS    RESTARTS   AGE
pod/tomcatpod-5666f5d5d8-4grsg    1/1     Running   0          7m33s
pod/tomcatpod-5666f5d5d8-dd2t6    1/1     Running   0          7m36s
pod/tomcatpod-5666f5d5d8-gg9kn    1/1     Running   0          7m30s

NAME                     TYPE           CLUSTER-IP     EXTERNAL-IP    PORT(S)        AGE
service/kubernetes       ClusterIP      10.104.0.1     <none>         443/TCP        10h
service/tomcatpod-svc    LoadBalancer   10.104.8.130   34.72.26.184   80:32310/TCP   29m

NAME                         READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/tomcatpod    3/3     3            3           29m

NAME                                    DESIRED   CURRENT   READY   AGE
replicaset.apps/tomcatpod-559f876c9c    0         0         0       21m
replicaset.apps/tomcatpod-5666f5d5d8    3         3         3       7m37s
replicaset.apps/tomcatpod-78cb458bb6    0         0         0       29m
```

Deployment has been completed. http://34.72.26.184/project-4.0-DAP/



Slack Messages on the pipeline stages



PoC2 has completed. Thank you