

DAP4.0 – DevOps PoC-2

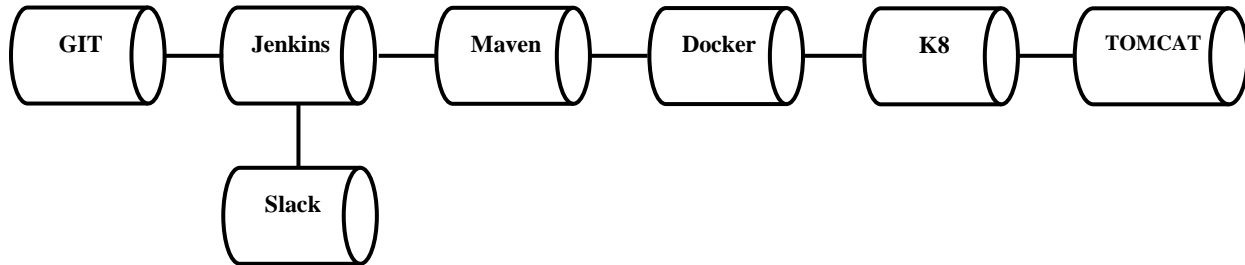
Employee Id: 353882
Employee Name: Srinivas Inampudi
Program: DAP4.0 – DevOps
Mentor Name: Anindya Bhattacharya
Use Case: Containerization and Orchestration

Table of Contents

DAP4.0 – DevOps PoC-2	2
1. Docker Kubernetes Pipeline Diagram	2
2. Docker Hub Account	2
3. Create GIT HUB Account/Repo	2
4. Jenkins Integration	3
Reuse Jenkins setup from PoC-1	3
Install Plug-ins that are required for Docker and Kubernetes Integration	3
K8 integration with Jenkins	4
Install Docker and Kubernetes on Jenkins Server	6
5. Docker Kubernetes Pipeline Job in Jenkins	6
6. Validation Screen Shots	11

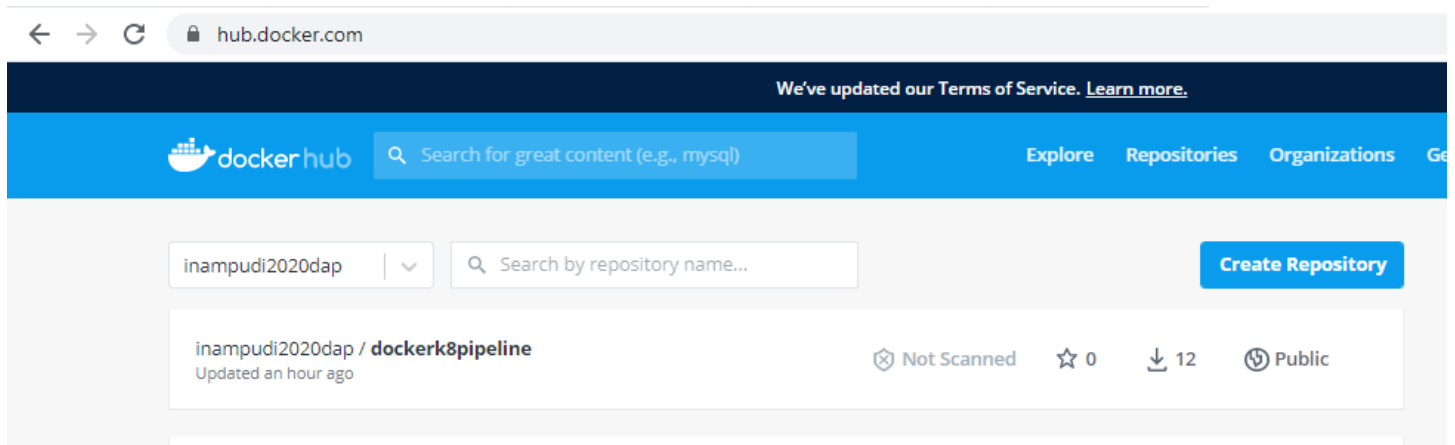
DAP4.0 – DevOps PoC-2

1. Docker Kubernetes Pipeline Diagram



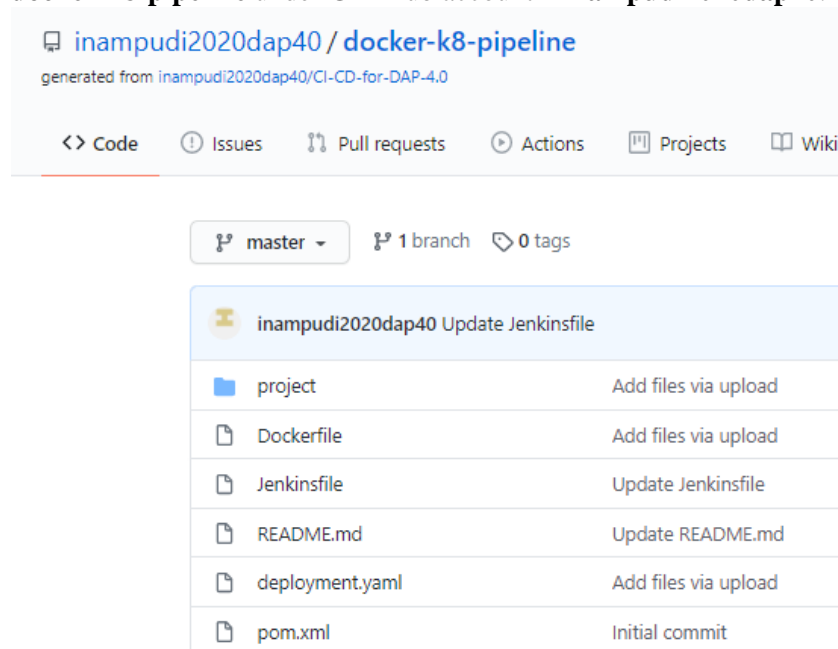
2. Docker Hub Account

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

The screenshot shows the 'Available' tab of the Jenkins Plugins page. The 'Install' column has an upward arrow. The 'Name' column lists the following plugins:

- Docker Commons**: ☐ api-plugin docker Library plugins (for use by other plugins)
Provides the common shared functionality for various Docker-related plugins.
- Docker Pipeline**: ☒ Deployment DevOps docker pipeline
Build and use Docker containers from pipelines.
- Docker API**: ☐ api-plugin docker
This plugin provides [docker-java](#) API for other plugins.
This plugin is up for adoption! We are looking for new maintainers. Visit our [Ado](#)
- Docker**: ☒ Cloud Providers Cluster Management and Distributed Build docker
This plugin integrates Jenkins with [Docker](#)

At the bottom, there are two buttons: 'Install without restart' and 'Download now and install after restart'. To the right, it says 'Update information obtaini'.

The screenshot shows the 'Available' tab of the Jenkins Plugins page. A search bar at the top contains 'gcloud'. The 'Install' column has an upward arrow. The 'Name' column lists the following plugin:

- GCloud SDK**: ☒ Build Tools Build Wrappers
GCloud SDK Plugin allows the invocation of the gcloud CLI as a job step.

At the bottom, there are two buttons: 'Install without restart' and 'Download now and install after restart'. To the right, it says 'Update information obtainec'.

Q google kubernetes

Updates Available Installed Advanced

Install ↑ Name

Kubernetes

☐ agent Cloud Providers Cluster Management and Distributed Build kubernetes

This plugin integrates Jenkins with [Kubernetes](#)

Google Kubernetes Engine

☒ google-cloud kubernetes

This plugin allows Jenkins to deploy build artifacts to a Kubernetes cluster running on

Install without restart Download now and install after restart Update information obtained: 1

K8 integration with Jenkins

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

Cluster basics

NAME

dap-k8-cluster

Location type

☒ Zonal

☐ Regional

Zone

us-central1-c

Generate a Service account for Jenkins and K8 integration

dap-k8-cluster Search products and resources

Service accounts + CREATE SERVICE ACCOUNT DELETE SHOW INFO PANEL

Service accounts for project "dap-k8-cluster"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. [Learn more about service accounts.](#)

Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accounts entirely. [Learn more about service account organization policies.](#)

Email	Status	Name ↑	Description	Key ID	Key creation date	Actions
417111371903-compute@developer.gserviceaccount.com	✓	Compute Engine default service account		No keys		⋮

Generate a JSON Key and download it to local

Create private key for "Compute Engine default service account"

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

Key type

☒ JSON

Recommended

☐ P12

For backward compatibility with code using the P12 format

CANCEL

CREATE

Private key saved to your computer



dap-k8-cluster-d66c14644534.json allows access to your cloud resources, so store it securely. [Learn more](#)

CLOSE

Email	Status	Name ↑	Key ID	Key creation date
417111371903-compute@developer.gserviceaccount.com	✓	Compute Engine default service account	d66c14644534123d429bf20b99bfd38ba57fd3ac	Oct 10, 2020

Upload the key to Jenkins

Kind

Project Name

☒ JSON key

JSON key File dap-k8-clus...644534.json

☐ P12 key

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: 26ff363bcf3b3f5a00498ac43694b1c7d9ce16c
+ sudo -E sh -c apt-get update -qq >/dev/null
+ sudo -E sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-https
+ 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

The screenshot shows the Jenkins configuration interface for a pipeline job named 'jenkins-docker-k8-pipeline'. The 'General' tab is selected. The description field contains 'jenkins-docker-k8-pipeline'. Below this, the 'Discard old builds' option is checked. Under the 'Strategy' section, 'Log Rotation' is selected. The 'Days to keep builds' is set to 30, and the 'Max # of builds to keep' is set to 10. There are also links for '[Plain text]' and 'Preview'.

General Build Triggers Advanced Project Options **Pipeline**

Definition Pipeline script from SCM

SCM Git

Repositories

Repository URL <https://github.com/inampudi2020dap40/c>

Credentials inampudi2020dap40/***** (qithl)

Add Advanced... Add Repository

Branches to build

Branch Specifier (blank for 'any') */master

Add Branch

Repository browser (Auto)

Additional Behaviours Add

Script Path Jenkinsfile

Save Apply

Add below files to GIT Repo

Dockerfile
Jenkinsfile
deployment.yaml

master docker-k8-pipeline / Dockerfile

inampudi2020dap40 Add files via upload

1 contributor

3 lines (3 sloc) 154 Bytes

```

1 from tomcat:8.0.20-jre8
2 RUN mkdir /usr/local/tomcat/webapps/myapp
3 COPY project/target/project-4.0-DAP.war /usr/local/tomcat/webapps/project-4.0-DAP.war

```

master

docker-k8-pipeline / deployment.yaml





inampudi2020dap40 Add files via upload

1 contributor

34 lines (34 sloc) | 609 Bytes

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: tomcatpod
5  spec:
6    replicas: 3
7    selector:
8      matchLabels:
9        app: tomcatpod
10   template:
11     metadata:
12       labels:
13         app: tomcatpod
14     spec:
15       containers:
16       - name: tomcatpod
17         image: inampudi2020dap/dockerk8pipeline:tagversion
18       ports:
19       - containerPort: 8080
```


 inampudi2020dap40 Update Jenkinsfile 1 contributor

76 lines (72 sloc) | 2.71 KB

```
1  pipeline {
2      agent any
3      environment {
4          registry = "inampudi2020dap/dockerk8pipeline"
5          registryCredential = 'docker'
6          dockerImage = ''
7          PROJECT_ID = 'dap-k8-cluster'
8          CLUSTER_NAME = 'dap-k8-cluster'
9          LOCATION = 'us-central1-c'
10         CREDENTIALS_ID = 'gcp-k8-key'
11     }
12
13     stages {
14         stage('Scm Checkout') {
15             steps {
16                 checkout scm
17             }
18         }
19     }
20 }
```

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

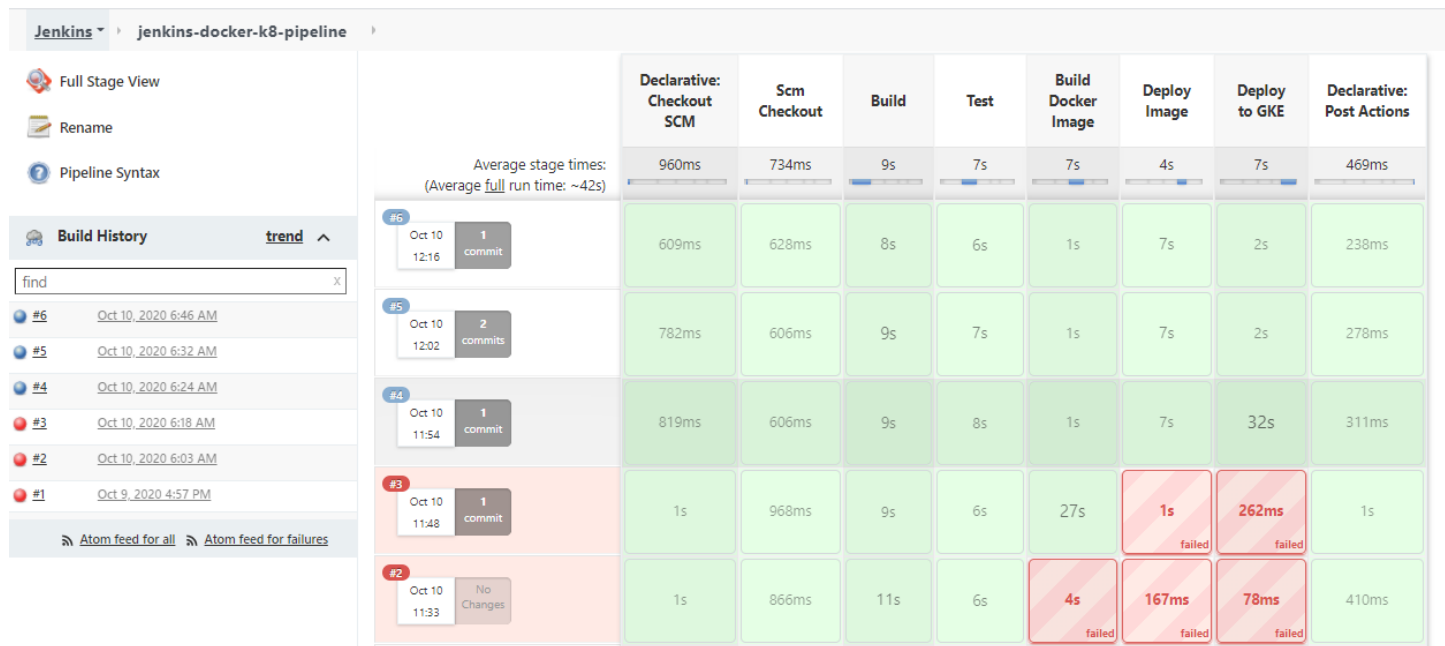
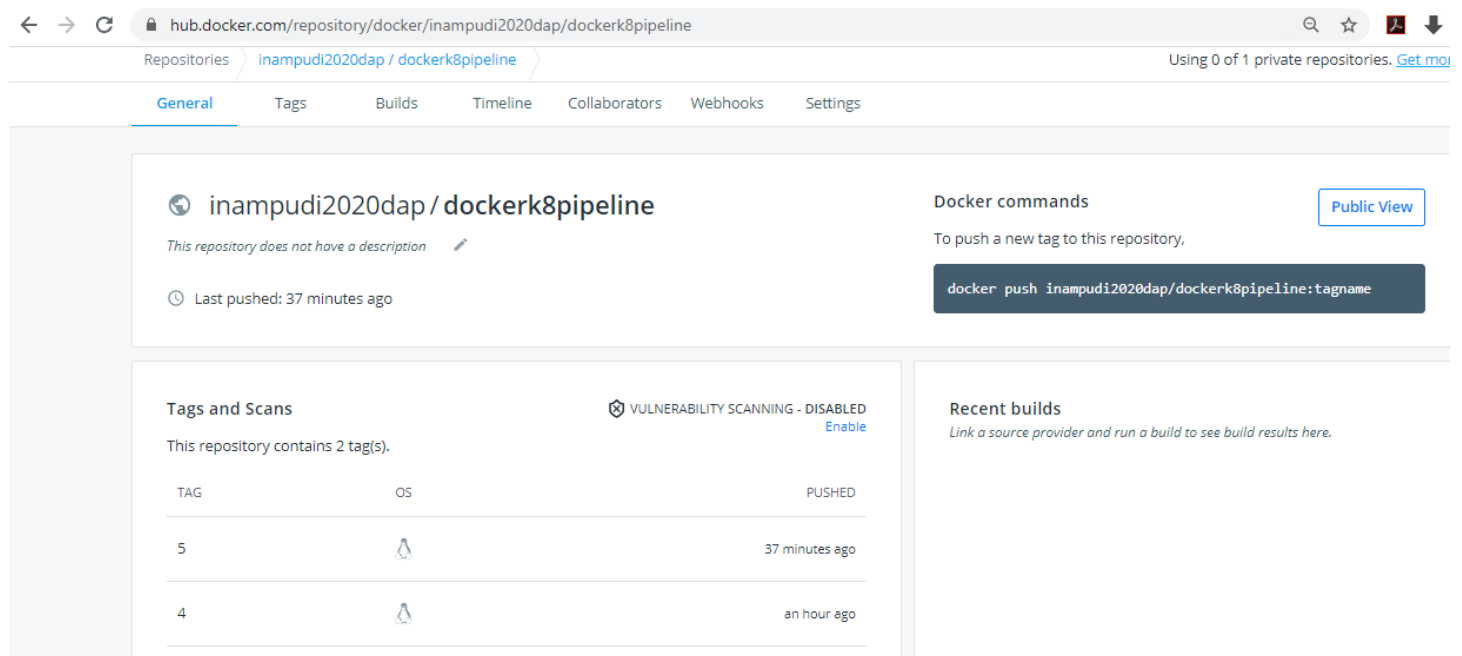
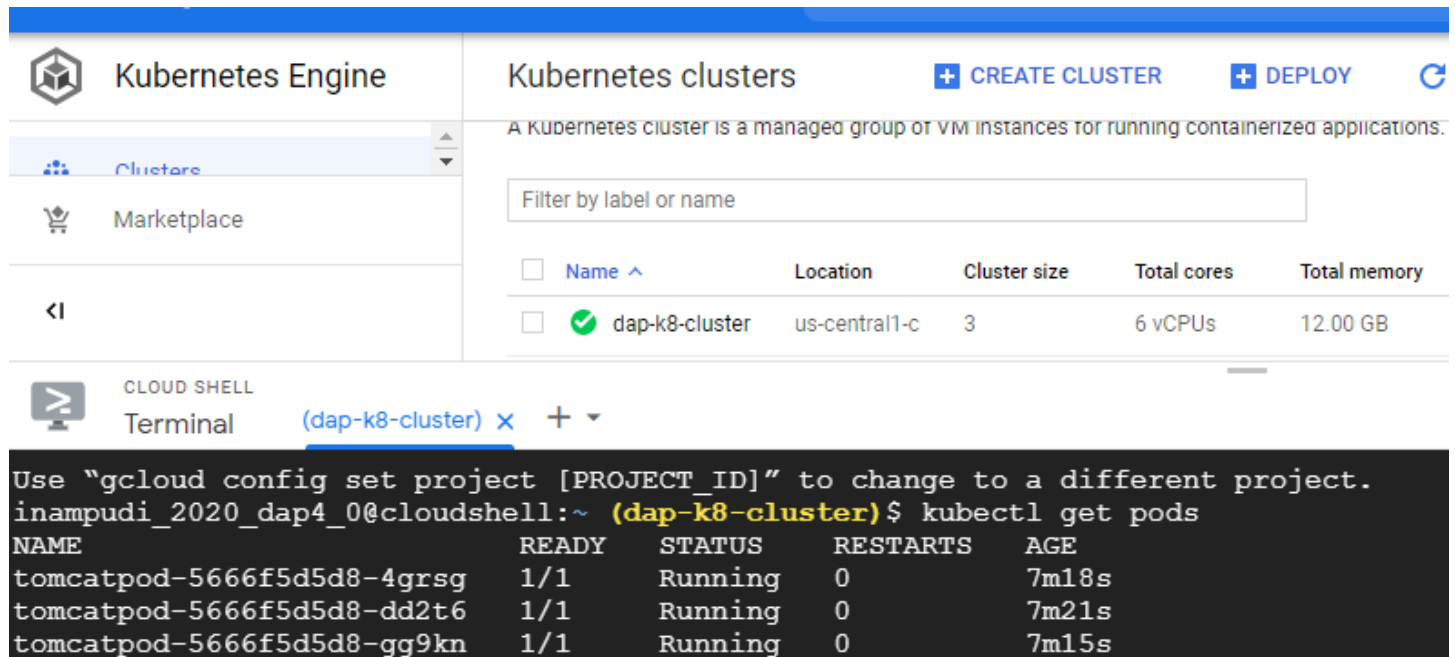


Image has been pushed to Docker



PODs have been created in K8 cluster



Kubernetes Engine

Kubernetes clusters

A Kubernetes cluster is a managed group of VM instances for running containerized applications.

Filter by label or name

<input type="checkbox"/> Name ^	Location	Cluster size	Total cores	Total memory
<input checked="" type="checkbox"/> dap-k8-cluster	us-central1-c	3	6 vCPUs	12.00 GB

CLOUD SHELL

Terminal (dap-k8-cluster) x +

```
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
```

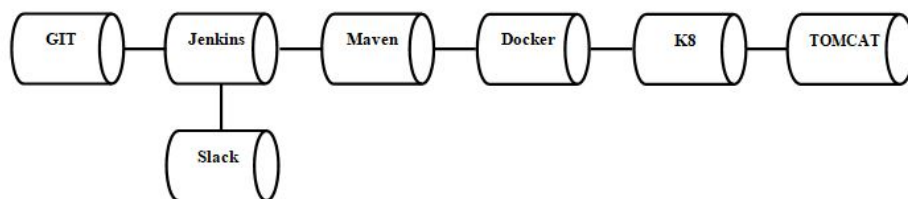
Deployment has been completed. <http://34.72.26.184/project-4.0-DAP/>

← → ↻ ⚠ Not secure | 34.72.26.184/project-4.0-DAP/

Hello DevOps Architects!

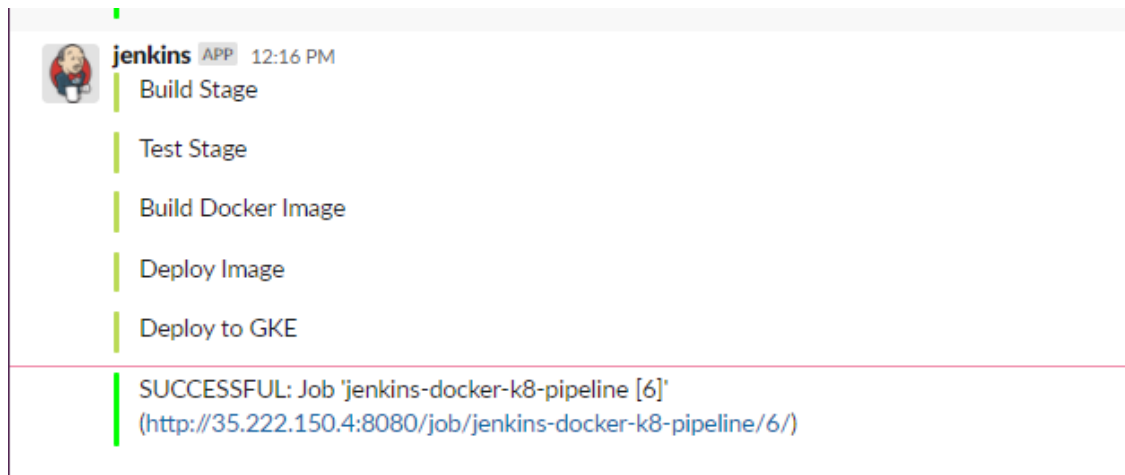


I am Srinivas Inampudi. As part of Poc2, I have used below tool stack



This session is to demonstrate how to setup Docker Kubernetes Pipeline

Slack Messages on the pipeline stages



PoC2 has completed. Thank you