

 [sandeep-varma8029](#) / [645\\_HW2\\_SurveyApp](#) Public[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [main](#) ▾

...

[645\\_HW2\\_SurveyApp](#) / [README.md](#)[sandeep-varma8029](#) Update README.md History 1 contributor

221 lines (156 sloc) | 11.4 KB

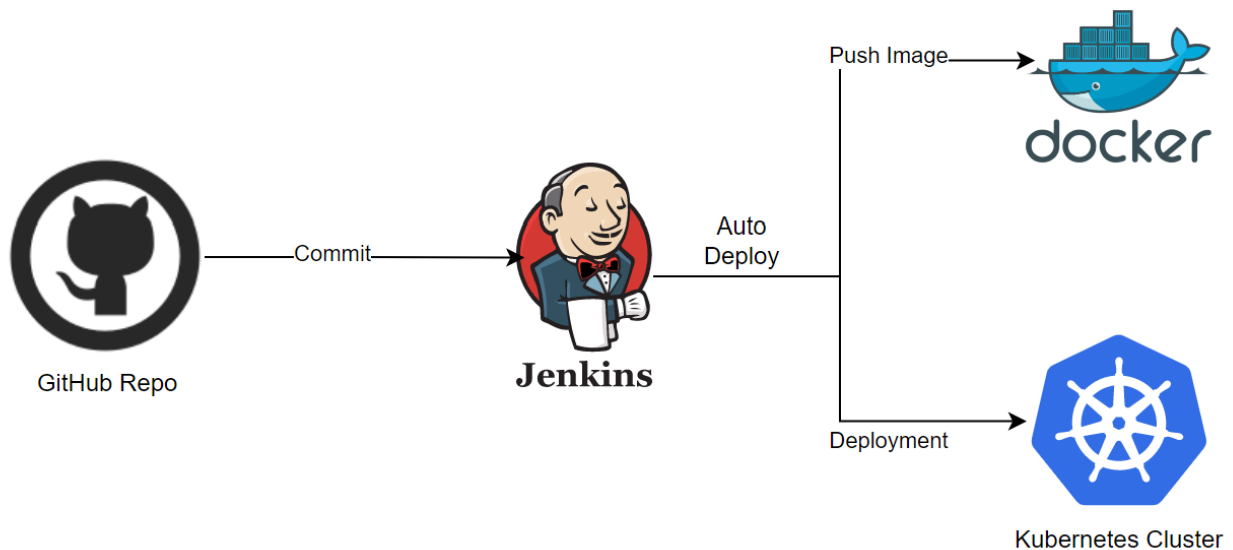
...

# 645 HW2 Survey Web Application using CI/CD Pipeline

GitHub URL : [https://github.com/sandeep-varma8029/645\\_HW2\\_SurveyApp](https://github.com/sandeep-varma8029/645_HW2_SurveyApp)

This Assignment involves establishing a CI/CD pipeline that includes a Git source code repository of the survey application developed in HW1 Assignment, and Jenkins for automated build and for the automated deployment of your application on Kubernetes. This is achieved by building and deploying a Docker Image to GCP Kubernetes cluster with the help of a Jenkins Pipeline. The Git Repo Folder has the source code of the Web App from eclipse, Dockerfile, Jenkinsfile, deployment.yaml and the HW1 Survey Application's war file. The below image shows an overview of the implementation

# CI/CD Pipeline



The basic Requirements Include:

1. A Google Cloud Platform (GCP) Account
2. Jenkins
3. Docker
4. Docker Hub Account
5. GCP Kubernetes Cluster

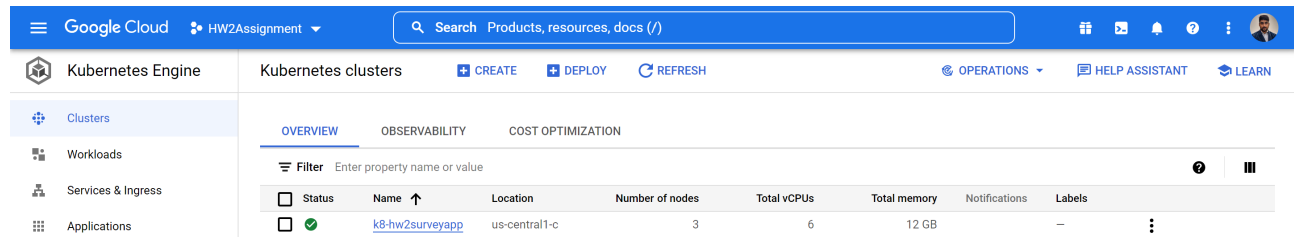
## Step 1: GCP Setup

First, we need to setup our GCP to have Jenkins and Kubernetes. To get started:

- Create a GCP account
- Create a new project
- Enter your project details such as name and location
- Click on create, you should see the created project on the dashboard

## Step 2: Setup Kubernetes on GCP

- Click on Kubernetes Engine
- Click on cluster, then click on create
- Click on configure under standard mode
- Enter your cluster name and click on create
- Enable Kubernetes Engine API



The screenshot shows the Google Cloud console interface for Kubernetes Engine. The left sidebar has 'Kubernetes Engine' selected. The main area shows 'Kubernetes clusters' with tabs for OVERVIEW, OBSERVABILITY, and COST OPTIMIZATION. A table lists one cluster named 'k8-hw2surveyapp' with 3 nodes, 6 vCPUs, and 12 GB memory.

Status	Name	Location	Number of nodes	Total vCPUs	Total memory	Notifications	Labels
<input checked="" type="checkbox"/>	<a href="#">k8-hw2surveyapp</a>	us-central1-c	3	6	12 GB	—	⋮

### Step 3: Setting up Jenkins on GCP

The next step is to create Jenkins's instance on GCP.

- Click on the hamburger menu
- Click on compute engine
- Click on VM instance
- Click on create instance
- Enter your Jenkins instance details
- Enter the name of the instance as Jenkins
- Enable firewall to allow HTTP and HTTPS traffic
- You can configure the Jenkins instance depending on your needs
- Click on create, You should see the Jenkins instance in the list of VM instances

Google Cloud HW2Assignment

Compute Engine

Virtual machines

- VM instances
- Instance templates
- Sole-tenant nodes
- Machine images
- TPUs
- Committed use discounts
- Migrate to Virtual Machines

Storage

- Disks
- Snapshots
- Images

Instance groups

- Instance groups
- Health checks

VM Manager

- OS patch management
- OS configuration managem...

Bare Metal Solution

- Servers
- Networks
- Volumes
- NFS Shares

Settings

- Metadata
- Zones
- Network endpoint groups
- Operations
- Security scans
- Settings

Marketplace

Release Notes

jenkins

EDIT RESET CREATE MACHINE IMAGE CREATE SIMILAR START / RESUME STOP SUSPEND DELETE

DETAILS OBSERVABILITY OS INFO SCREENSHOT

SSH CONNECT TO SERIAL CONSOLE

Connecting to serial ports is disabled

Logs

Cloud Logging  
Serial port 1 (console)  
SHOW MORE

Basic information

Name	jenkins
Instance Id	6014152732506347848
Description	None
Type	Instance
Status	Running
Creation time	Oct 20, 2022, 8:28:26 PM UTC-04:00
Zone	us-west4-b
Instance template	None
In use by	None
Reservations	Automatically choose
Labels	None
Tags	—
Deletion protection	Disabled
Confidential VM service	Disabled
Preserved state size	0 GB

Machine configuration

Machine type	e2-medium
CPU platform	Intel Broadwell
Architecture	x86_64
vCPUs to core ratio	—
Custom visible cores	—
Display device	Disabled Enable to use screen capturing and recording tools
GPUs	None

Networking

Public DNS PTR Record	None
Total egress bandwidth tier	—
NIC type	—

VIEW IN NETWORK TOPOLOGY

Firewalls

HTTP traffic	On
HTTPS traffic	On

Network tags

http-server https-server

Network interfaces

Name	Network	Subnetwork	Primary internal IP address	Alias IP ranges	Stack Type	External IP address	Network
nic0	default	default	10.182.0.2		IPv4	34.125.138.200 (Ephemeral)	Premium

There are various ways in which we can access the Jenkins instance. For this application, we are using browser window

Install Jenkins using the browser window. Check Jenkins Installation on Debian at <https://www.jenkins.io/doc/book/installing/linux/#debianubuntu>. The external IP is the accessible IP address for the Jenkins instance. For example, `http://<jenkins_external_ip>:8080`

The above URL will not be available, we need to add a firewall rule that will allow port 8080. To do this

- Click on Jenkins instance
- Scroll down to network tag, click on default under network header
- Click on firewall rules
- Click on add firewall rule
- Enter details of the firewall rule
- Type Jenkins as name
- Type any preferred description
- Select All instances in the network as target
- Type 0.0.0.0/0 as Source IPv4 ranges
- Select Specified protocols and ports
- Check tcp and enter 8080 as the value
- Click on save

Google Cloud

HW2Assignment

Search

VPC network

Firewall rule details

EDIT

DELETE

VPC networks

IP addresses

Bring your own IP

Firewall

Routes

VPC network peering

Shared VPC

Serverless VPC access

Packet mirroring

jenkins

Logs ?

Off

[view in Logs Explorer](#)

Network

default

Priority

1000

Direction

Ingress

Action on match

Allow

Source filters

IP ranges

0.0.0.0/0

Protocols and ports

tcp:8080

Enforcement

Enabled

Insights

None

Hit count monitoring ?

—

Applicable to instances

http://<jenkins\_external\_ip>:8080 will be accessible, install the necessary plugins and you will be able to login to Jenkins The simplest and most common way of installing plugins is through the Manage Jenkins > Manage Plugins view, available to administrators of a Jenkins environment. Under the Available tab, plugins available for download from the configured Update Center can be searched and considered. Important plugins to be installed:

Google Kubernetes Engine Plugin  
Kubernetes

Kubernetes CLI Plugin  
Kubernetes Credentials Provider  
Google Container Registry Auth Plugin  
Google Container Registry Auth Plugin  
Docker Pipeline  
Git client plugin  
Git plugin  
GitHub  
GitHub API Plugin  
Github App Kubernetes Credentials  
Github App Kubernetes Credentials  
GitHub Branch Source Plugin  
Pipeline: GitHub Groovy Libraries



# Welcome to Jenkins!

☐

Keep me signed in

Sign in

Storing Credentials in Jenkins: We will use the Jenkins credentials store for the pipeline to communicate with the Kubernetes cluster and the Docker Hub registry

Navigate to the "Dashboard > Manage Jenkins > Manage Credentials" menu item.  
Select the "System" sub-menu and the "Global credentials" domain.  
Click the "Add credentials" link.

Select the "Username with password" credential type and enter your Docker Hub



username and password in the corresponding fields.

Set the "ID" field to dockerID.

Click "OK" to save the changes.

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

sandeepvarma99

☐

Treat username as secret ?

Password ?



Concealed

Change Password

ID ?

docker-cred

Description ?

docker-cred

Save

## To Add GCP Kubernetes Cluster Credentials:

Navigate to the "IAM & admin -> Service accounts" page and create a new service account.

For this Assignment, I have created a service account with the name "Compute Engine default service account".

Assign it the "Kubernetes Engine Admin" role.

Next, create a new JSON key for the service account.

Download and save this key.

Google Cloud HW2Assignment Search Products, resources, docs (/)

IAM & Admin Compute Engine default service account

DETAILS PERMISSIONS KEYS METRICS LOGS

**Keys**

Service account keys could pose a security risk if compromised. We recommend you avoid downloading service account keys and inst accounts on Google Cloud [here](#).

Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using [organization policies](#).  
[Learn more about setting organization policies for service accounts](#)

ADD KEY ▾

- Create new key
- Upload existing key

Key	Key creation date	Key expiration date	
7d420fb3dee570f0947fca74175e527dae8cb29d	Oct 22, 2022	Dec 31, 9999	🗑

On Jenkins Click the "Add credentials" link.

Select the "Google Service Account from private key" credential type and set the project name (which doubles as the credential identifier) to my project name.

Select the "JSON key" radio button and upload the JSON key obtained in the previous step in Setting up the GCP Account.

Click "OK" to save the changes.

Project Name ?

hw2assignment

☒ JSON key

JSON key File ?

Choose File No file chosen

(Or reuse previous file: hw2assignment-366123-7d420fb3dee5.json)

☐ P12 key

Save

## Step 4: Creating a Docker Container

. Make sure you have docker installed on your machine, install from

<https://docs.docker.com/engine/install/ubuntu/>

. Create a file called Dockerfile. Docker requires the file to be called 'Dockerfile'

. Build the HW1 Survey Web App project on eclipse and put the war file in the same folder as the Dockerfile, Note that 'HW1\_Survey\_from' is the display name of the Tomcat application

. In the DockerFile, use the FROM command to get the initial image for the build,

We want to run the war file in Tomcat so we should use the tomcat image:

```
FROM tomcat:8.5.47-jdk8-openjdk
```

. Next, we need to drop the war file in the webapps folder:

```
COPY ./HW1_Survey_form.war /usr/local/tomcat/webapps
```

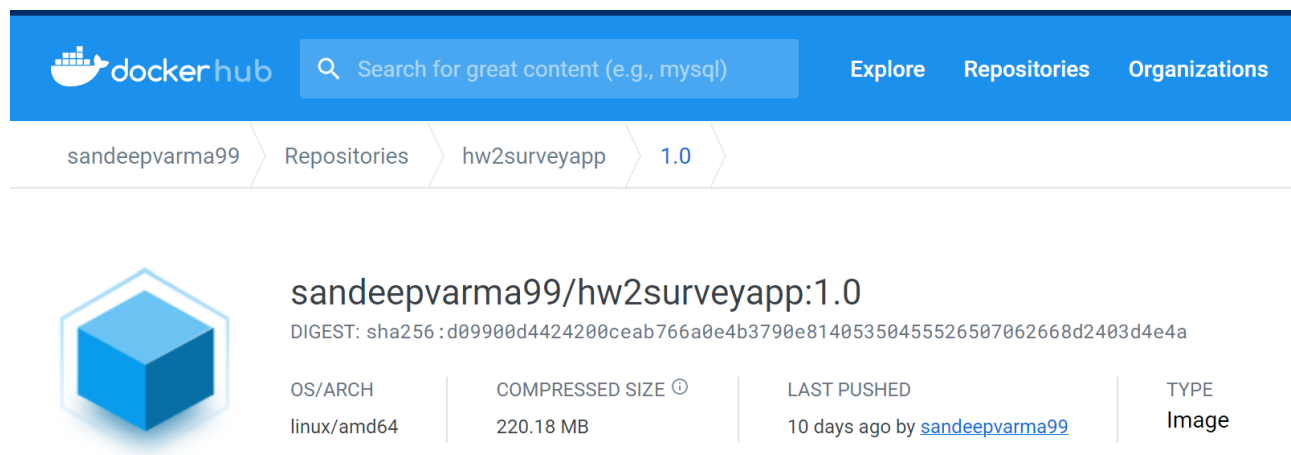
. On the command line, use this command: 'docker build -t hw2surveyapp:1.0' You can use whatever name and tag you want

. Verify that the image is properly working by running 'docker run -it -p 8080 hw2surveyapp:1.0' and open a browser at [http://localhost:8080/HW1\\_Survey\\_form](http://localhost:8080/HW1_Survey_form)

. On the command line, login to docker using 'docker login -u <your username>'

. Change the name of you image to be <your username on dockerhub>/<name of the app>:<image tag> using the docker tag command. In my case it is: 'docker tag hw2surveyapp:1.0 sandeepvarma99/hw2surveyapp:1.0'

. Verify that your image is on Docker Hub, Your image should be accessible from the internet



The screenshot shows the Docker Hub interface. At the top is a blue navigation bar with the Docker Hub logo, a search bar, and links for Explore, Repositories, and Organizations. Below the navigation bar is a breadcrumb trail: sandeepvarma99 > Repositories > hw2surveyapp > 1.0. The main content area displays the repository details for sandeepvarma99/hw2surveyapp:1.0. It includes a blue cube icon, the repository name, a long digest string, and a table with four columns: OS/ARCH, COMPRESSED SIZE, LAST PUSHED, and TYPE. The table shows the image is for linux/amd64, is 220.18 MB, was pushed 10 days ago by sandeepvarma99, and is of type Image.

OS/ARCH	COMPRESSED SIZE	LAST PUSHED	TYPE
linux/amd64	220.18 MB	10 days ago by <a href="#">sandeepvarma99</a>	Image

## Step 5: Create deployment file for Kubernetes and JenkinsFile for the Jenkins pipeline

A deployment.yaml file in the repository is added which defines how the built container should be deployed on Kubernetes. The definition pulls the built container from Docker Hub and creates a new deployment with it in your Kubernetes cluster running on 3 pods. It also creates a LoadBalancer service so that the deployment can be accessed from outside the cluster.

Finally, create a pipeline script named Jenkinsfile (also added in the repo). This is the script Jenkins will use to build and deploy the application. What the Jenkins pipeline file does are as follows;

- Pull from your repository
- Dockerize the application
- Push the image to docker hub
- Update the deployment YAML file with the build number
- Deploy to GKE

## Step 6: Jenkins Pipeline Setup

We are using Jenkins as our CI/CD

- Open Jenkins on GCP
- Click on New Item
- Enter Jenkins job name
- Under 'Build Triggers' select 'GitHub hook trigger for GITScm polling' and 'Poll SCM'
- Select Pipeline and ok
- Click on pipeline
- Select Pipeline script from SCM as definition
- Select git as SCM
- Enter your repository url
- Enter Jenkinsfile in the script path
- Click on apply and save

Jenkins

Dashboard > Hw2Assign >

Configure

General

Advanced Project Options

Pipeline

General

Enabled

Description

[Plain text] [Preview](#)

☐ Discard old builds [?](#)

☐ Do not allow concurrent builds

☐ Do not allow the pipeline to resume if the controller restarts

☐ GitHub project

☐ Pipeline speed/durability override [?](#)

☐ Preserve stashes from completed builds [?](#)

☐ This project is parameterized [?](#)

☐ Throttle builds [?](#)

Build Triggers

☐ Bitbucket webhook trigger [?](#)

☐ Build after other projects are built [?](#)

☐ Build periodically [?](#)

☒ GitHub hook trigger for GITScm polling [?](#)

☒ Poll SCM [?](#)

Schedule [?](#)

No schedules so will only run due to SCM changes if triggered by a post-commit hook

☐ Ignore post-commit hooks [?](#)

☐ Quiet period [?](#)

☐ Trigger builds remotely (e.g., from scripts) [?](#)

Advanced Project Options

Advanced...

Pipeline

Definition

Pipeline script from SCM

SCM [?](#)

Git

Repositories [?](#)

Repository URL [?](#)

https://github.com/sandeep-varma8029/645\_HW2\_SurveyApp.git

Credentials [?](#)

none

+ Add

Advanced...

Add Repository

Branches to build [?](#)

Branch Specifier (blank for 'any') [?](#)

\*/main

Add Branch

Repository browser [?](#)

(Auto)

Additional Behaviours

Add

Script Path [?](#)

jenkinsfile

☒ Lightweight checkout [?](#)

[Pipeline Syntax](#)

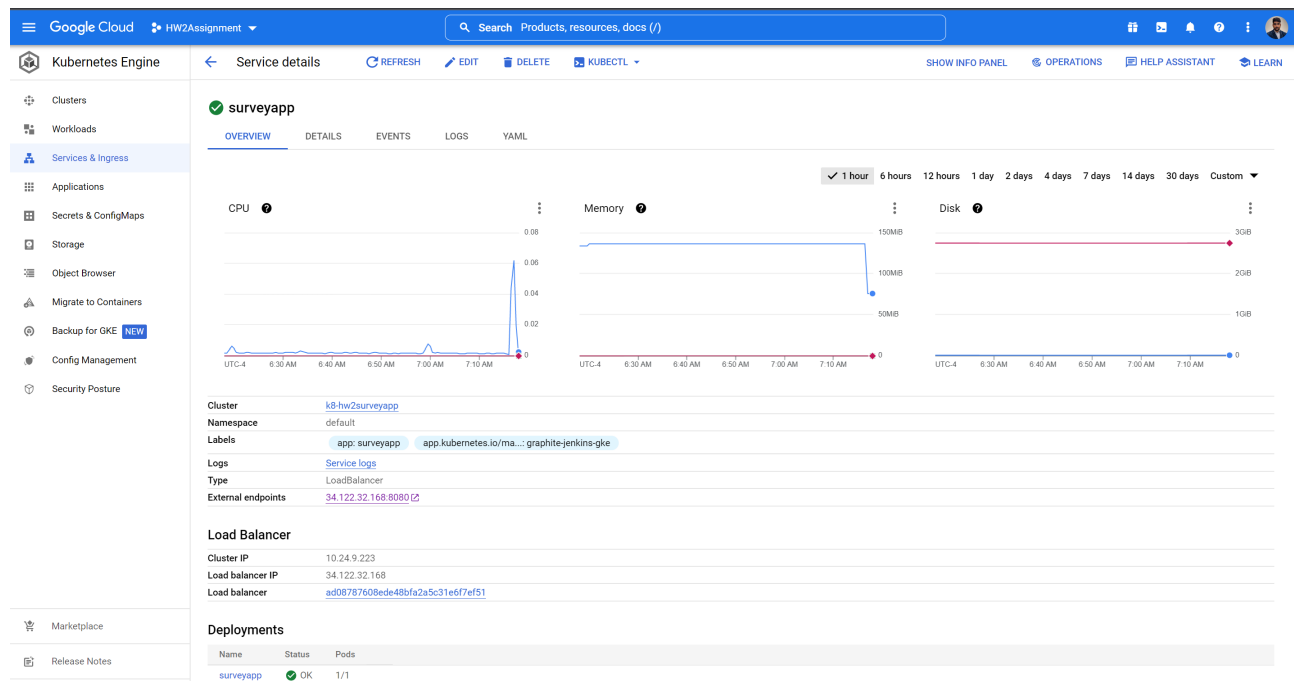
Save

Apply

https://github.com/sandeep-varma8029/645\_HW2\_SurveyApp/blob/main/README.md

13/16





The application can be assessed at [http://34.122.32.168:8080/HW1\\_Survey\\_form/](http://34.122.32.168:8080/HW1_Survey_form/)

# GMU Student Survey Form

\* First Name

Enter your First name

\* Last Name

Enter your Last name

\* Street address

Street address

\* City

city

\* State

State

\* Zip Code

Zip Code

\* Telephone Number

+1 xxx-xxx-xxxx

AWS URL of the SurveyPage : <http://studentsurveyapp.s3-website-us-east-1.amazonaws.com/>

AWS URL of Sai Sandeep Varma Mudundi HomePage: <http://personalwebsitehw1.s3-website-us-east-1.amazonaws.com/>

AWS Homepage URL of Sai Anjaneya Sowrab Meduri( G01270421): <http://swe645-msa.s3-website-us-east-1.amazonaws.com/>

AWS Homepage URL of Rajeev Priyatam Panchadula( G01333080): <http://rajeevpartone.s3-website-us-east-1.amazonaws.com/>

AWS Homepage URL of Lakshmi Guttikonda(G01334433): <http://cs645ha.com.s3-website-us-east-1.amazonaws.com/>

Project Demo Link :

[https://drive.google.com/file/d/1EqZemRzBKUokM3IK3q\\_2spCHs0Q79\\_wx/view?usp=share\\_link](https://drive.google.com/file/d/1EqZemRzBKUokM3IK3q_2spCHs0Q79_wx/view?usp=share_link)

Team Details and Contributions:

Sai Sandeep Varma Mudundi (G01352322) - Jenkins CI/CD pipeline creation, Kubernetes Cluster setup, GitHub setup containerization of web app, Jenkinsfile, deployment file creation ,readme , demo video and documentation

Sai Anjaneya Sowrab Meduri( G01270421) - Docker setup,Dockerfile creation, hosting of web application container on kubernetes cluster documentation and containerization

Rajeev Priyatam Panchadula( G01333080) - Git SCM polling,Virtual Machine instance configuration and documentation, setting up git repo war file generation

Lakshmi Guttikonda(G01334433) - Documentation, AWS hosting of homepage, maintaining jenkins plugins and IAM credentials, and web application contribution

References :

[1] : [https://www.linkedin.com/pulse/create-cicd-pipeline-jenkins-google-kubernetes-engine-pralay-debroy?trk=articles\\_directory](https://www.linkedin.com/pulse/create-cicd-pipeline-jenkins-google-kubernetes-engine-pralay-debroy?trk=articles_directory)

[2] : <https://medium.com/@bukunmitanimonure/build-and-deploy-docker-image-to-gcp-kubernetes-cluster-with-jenkins-pipeline-5405a2966a58>