

Scenario 3: - CI/CD (Jenkins + Git hub + Maven + Ansible + Docker container + kubernetes cluster).

Server 1: - jenkins (Make sure to install Jenkins, Git, Maven, kubectl)

Server 2: - tomcat (Make sure to install docker container, ansible engine)

Step 1: - Create a kubernetes cluster node on GCP (note: - Kubernetes master will be free from google.)

: -Click Kubernetes Engine → Clusters → select 2 nodes

The screenshot shows the Google Cloud Platform console interface. In the left-hand navigation menu, 'Kubernetes Engine' is highlighted with a blue circle. A blue arrow points from this menu item to the 'Clusters' link in the sub-menu that appears on the right. The 'Clusters' link is also circled in blue. Below this, the 'Kubernetes clusters' page is visible, showing a table with one cluster named 'robo'.

Name	Location	Cluster size	Total cores	Total memory	Notifications	Labels
robo	us-central1-a	2	2 vCPUs	7.50 GB		

Step 2:- login to google cloud via gcloud auth login.

```
[root@jenkins ~]# gcloud auth login

You are running on a Google Compute Engine virtual machine.
It is recommended that you use service accounts for authentication.

You can run:

    $ gcloud config set account `ACCOUNT`

to switch accounts if necessary.

Your credentials may be visible to others with access to this
virtual machine. Are you sure you want to authenticate with
your personal account?

Do you want to continue (Y/n)? yes

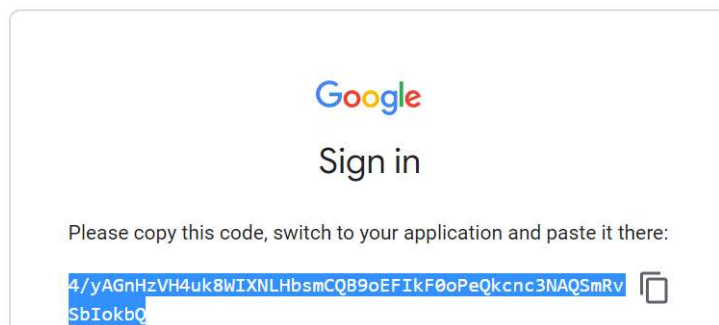
Go to the following link in your browser:

    https://accounts.google.com/o/oauth2/auth?code_challenge=eFZm_r3JvFXJCYkg-IFoDUXtKwP7JeGYPOv7vfCB-ys&prompt=select_account&code_challenge_method=S256&access_type=offline&redirect_uri=urn%3Aietf%3Aawg%3Aoauth%3A2.0%3Aoob&response_type=code&client_id=32555940559.apps.googleusercontent.com&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth

Enter verification code: 4/yAGnHzVH4uk8WIXNLHbsmCQB9oEFIkF0oPeQkcnc3NAQSmRvSbIokbQ
```

← → ↺ accounts.google.com/o/oauth2/approval/v2/approvalnativeapp?auto=false&response=code%3D4%2FyAGnHzVH4uk8WIXNLHbsmCQB9oEFIkF0oPeQkcnc3NAQSmRvSbIokbQ

Apps AWS Free Tier Imported From IE Apps GoToMeet.Me Kubernetes Master... CentOS 7 : Kuberne... Install K



Step 2.1:- Connect to Kubernetes cluster.

```
[root@jenkins ~]# gcloud container clusters get-credentials robo --zone us-central1-a --project caramel-primer-267805
Fetching cluster endpoint and auth data.
kubeconfig entry generated for robo.
[root@jenkins ~]#
```

#yum install kubectl -y

#kubectl get nodes

```
[root@jenkins ~]# kubectl get nodes
NAME                                STATUS    ROLES    AGE    VERSION
gke-robo-default-pool-258eb833-bkmt Ready    <none>    21m    v1.14.10-gke.24
gke-robo-default-pool-258eb833-kmqw Ready    <none>    21m    v1.14.10-gke.24
[root@jenkins ~]#
```

Step 3:- Install ansible engine on tomcat server

#yum install ansible -y

#yum install python -y

pip install --upgrade pip

@@Create ansadm user on both servers (Jenkins and tomcat(ansible) and add ssh key on both servers to make passwordless@@

```
#useradd ansadm && passwd ansadm
```

```
#usermod -G google-sudoers ansadm
```

```
#su - ansadm
```

```
$ssh-keygen
```

```
$ssh-copy-id ansadm@10.128.0.46
```

```
$cd /opt && sudo mkdir docker && sudo chmod -R 777 docker
```

Step 3.1:- on ansible server, create an .ansible.cfg file on ansadm user home folder with below settings.

```
#vim .ansible.cfg
```

```
[ansadm@tomcat ~]$ cat .ansible.cfg
```

```
[defaults]
```

```
inventory = /opt/docker/inventory
```

```
fact_path = /home/ansadm/facts.d/
```

```
forks = 5
```

```
poll_interval = 90
```

```
roles_path = /home/ansadm/playbook/roles
```

```
host_key_checking = False
```

```
deprecation_warnings=False
```

```
[privilege_escalation]
```

```
become=no
```

```
become_method=sudo
```

```
become_user=root
```

```
become_ask_pass=no
```

```
[colors]
```

```
highlight = white
```

```
verbose = blue
```

```
warn = bright purple
```

```
error = red
```

```
debug = dark gray
```

```
deprecate = purple
```

```
skip = cyan
```

```
unreachable = red
```

```
ok = green
```

```
changed = yellow
```

```
diff_add = green
```

```
diff_remove = red
```

```
diff_lines = cyan
```

```
[ansadm@tomcat ~]$
```

```
$ansible --version
```

```
[ansadm@tomcat ~]$ ansible --version
```

```
ansible 2.9.6
```

```
config file = /home/ansadm/.ansible.cfg
```

```
configured module search path = [u'/home/ansadm/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
```

```
ansible python module location = /usr/lib/python2.7/site-packages/ansible
```

```
executable location = /bin/ansible
```

```
python version = 2.7.5 (default, Aug 7 2019, 00:51:29) [GCC 4.8.5 20150623 (Red Hat 4.8.5-39)]
```

```
[ansadm@tomcat ~]$
```

@@now create an inventory file under /opt/docker

```
[ansadm@tomcat docker]$ cat inventory
[local]
10.128.0.45

[kube]
10.128.0.46
[ansadm@tomcat docker]$ 
[ansadm@tomcat docker]$ ansible all -m ping
10.128.0.46 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
10.128.0.45 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
[ansadm@tomcat docker]$
```

@Make sure to have install docker on tomcat server and add ansadm user into docker group

```
[ansadm@tomcat ~]$ id
```

```
uid=1003(ansadm) gid=1004(ansadm) groups=1004(ansadm),994(docker),1000(google-sudoers)
```

```
[ansadm@tomcat ~]$
```

Step 4:-Now in ansible server create a playbooks.

Playbook 1:- create a docker tomcat image with help of Docker file and once build, push the image into docker hub register.

###please make sure to use your docker hub ID, here in this playbook I have used mine (deepu1986). In case you don't have, kindly create it.

```
#cd /opt/docker/
```

```
[ansadm@tomcat docker]$ cat Dockerfile
```

```
FROM tomcat:8-jre8
```

```
LABEL maintainer=Deepan
```

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

```
[ansadm@tomcat docker]$
```

```
# create-simple-devops-image.yml
```

```
---
```

```
- hosts: local
```

```
  become: yes
```

```
tasks:
```

```
- name: create docker image using war file
```

```
  command: docker build -t simple-devops-image:latest .
```

```
args:
  chdir: /opt/docker
- name: create tag to image
  command: docker tag simple-devops-image deepu1986/simple-devops-image

- name: login to docker
  docker_login:
    username: deepu1986
    password: XXXXXXXXXX

- name: push image on to dockerhub
  command: docker push deepu1986/simple-devops-image

- name: remove docker images form ansible server
  command: docker rmi simple-devops-image:latest deepu1986/simple-devops-image
  ignore_errors: yes
```

Playbook2 :- create a playbook to execute kubernetes yml for pod deployment.

```
# cat kubernetes-valaxy-deployment.yml
```

```
---
```

```
- name: Create pods using deployment
  hosts: kube
  become: yes
```

```
tasks:
```

```
- name: create a deployment
  command: kubectl apply -f valaxy-deploy.yml
  args:
    chdir: /opt/docker
```

```
- name: update deployment with new pods if image updated in docker hub
  command: kubectl rollout restart deployment.v1.apps/valaxy-deployment
```

Playbook3: create a playbook for kubernetes service.

```
# cat kubernetes-valaxy-service.yml
```

```
---
```

```
- name: create service for deployment
  hosts: kube
  become: yes
```

```
tasks:
```

```
- name: create a service
  command: kubectl apply -f valaxy-service.yml
  args:
    chdir: /opt/docker
```

Step 5: - add ansible server details on jenkins configure system.

Manage Jenkins → configure system → drag down and select publish over ssh → then add ansible server

SSH Server

Name

ansible

Hostname

tomcat.us-central1-a.c.caramel-primer-267805.internal

Username

ansadm

Remote Directory

☒ Use password authentication, or use a different key

Passphrase / Password

.....

Step 6:-Create a new maven project for CI

Note:- you can fork this GIT repo <https://github.com/yankils/hello-world>

Source Code Management

None

Git

Repositories

Repository URL

https://github.com/deepanredhat/hello-world

Credentials

- none -

Add

Advanced...

Add Repository

Branches to build

Branch Specifier (blank for 'any')

*/master

Add Branch

Build

Root POM

pom.xml

Goals and options

clean install package

Advanced...

@@add ansible command on exec command@@
ansible-playbook /opt/docker/create-simple-devops-image.yml -b

Jenkins » Deploy_on_Kubernetes_CI »

General Maven Info Plugin Configuration Source Code Management Build Triggers Build Environment Pre Steps Build

Post Steps Build Settings **Post-build Actions**

SSH Publishers

SSH Server

Name ansible

Advanced...

Transfers

Transfer Set

Source files webapp/target/*.war

Remove prefix webapp/target

Remote directory /opt/docker

Exec command ansible-playbook /opt/docker/create-simple-devops-image.yml -b

Save Apply

Step 6.1 Create a free style job for CD and add below commands on Exec command.

ansible-playbook /opt/docker/kubernetes-valaxy-deployment.yml -b;
ansible-playbook /opt/docker/kubernetes-valaxy-service.yml -b

Jenkins » Deploy_on_Kubernetes-CD »

General Source Code Management Build Triggers Build Environment **Build** Post-build Actions

Advanced...

Transfers

Transfer Set

Source files

Remove prefix

Remote directory

Exec command ansible-playbook /opt/docker/kubernetes-valaxy-deployment.yml -b;
ansible-playbook /opt/docker/kubernetes-valaxy-service.yml -b

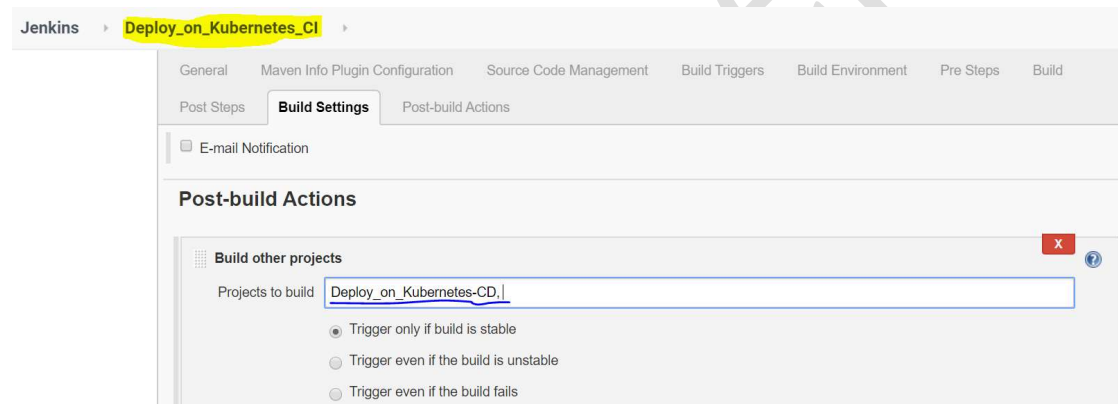
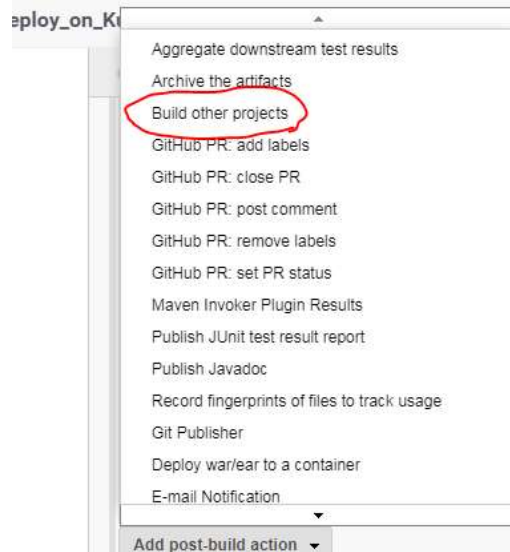
All of the transfer fields (except for Exec timeout) support substitution of [Jenkins environment variables](#)

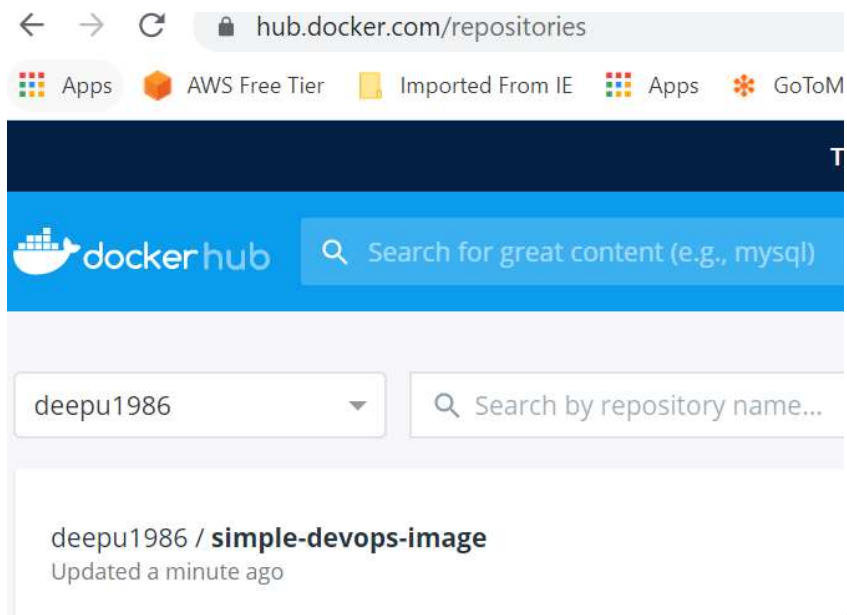
Advanced...

Save Apply

Add Transfer Set

Step 6.2 open on CI project and add CD by select post-build Actions → select build other project → select the CD project.





kubectl get pods
kubectl get service
kubectl get deploy

```
[root@jenkins docker]# kubectl get pods
NAME                                READY    STATUS    RESTARTS   AGE
robo-deployment-7dc74cc96c-7lvqc    1/1      Running   0           17m
robo-deployment-7dc74cc96c-mvl98    1/1      Running   0           17m
[root@jenkins docker]# kubectl get service
NAME            TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
kubernetes      ClusterIP     10.36.0.1     <none>         443/TCP          4h38m
robo-service    LoadBalancer 10.36.6.192   35.202.98.16   8080:31200/TCP   17m
[root@jenkins docker]#
```

@@Open the URL like below and verify the webpage@@



Hello, Welcome to robo DevOps Project !!

Deploying on a kubernetes using ansible for robo2.0

Glad to see you