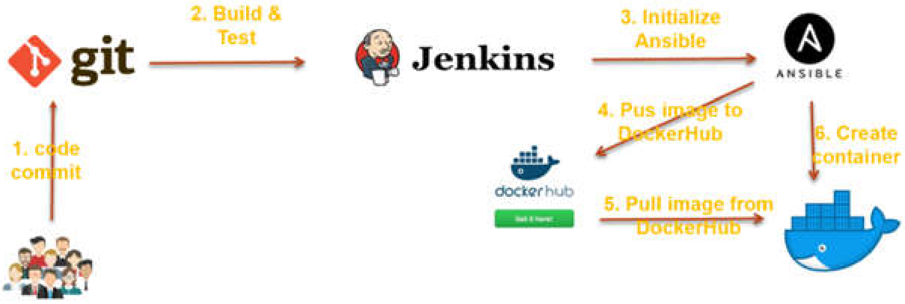
Devops Project-4- By using Git, Jenkins, Ansible, DockerHub, Docker to DEPLOY on a docker container by Raj Gupta



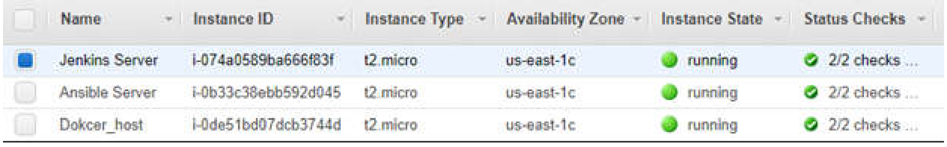
In *part-01* we create Docker image on ansible server through Jenkins job and pushed it onto DockerHub.

1. Docker should be installed on ansible server
2. Should login to "docker hub" on ansible server
3. Docker admin user should be part of `docker` group

In *Part-02* we create *create_docker_container.yml* playbook. this get intiated by jenkins job, run by ansible and exected on dokcer_host

In *Part-03* we try to improvise to store docker images previous version
So for we used latest docker image to build a container, but what happens if latest version is not working? One easiest solution is, maintaining version for each build. This can be achieved by using environment variables.

Take 3 EC2 REHL server as below for Jenkins server, Ansible Server and Docker host



□	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
■	Jenkins Server	i-074a0589ba666f83f	t2.micro	us-east-1c	● running	✓ 2/2 checks ...
□	Ansible Server	i-0b33c38ebb592d045	t2.micro	us-east-1c	● running	✓ 2/2 checks ...
□	Dokcer_host	i-0de51bd07dcb3744d	t2.micro	us-east-1c	● running	✓ 2/2 checks ...

On Jenkins server: - Do the same setup for Jenkins server as project 1. (install jenkins, git,, maven and java on REHL Jenkins server)

On Ansible Server install ansible as below:-

```
rpm -Uvh http://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
yum install ansible
ansible –version
```

adding client to ansible master

```
cd /etc/ansible
vi hosts
```

then add all clients private ip in this file on top and save it

```
[web]
172.31.38.44      ...like this
```

then to check the master and client past below in master

```
[root@ip-172-31-18-161 ~]# ansible -m ping all
```

On Dokcer_host server: - Do the same setup as project 3

To install Docker on RHEL server give the below command

```
rpm -ivh ftp://ftp.icm.edu.pl/vol/rzm3/linux-slc/centos/7.1.1503/extras/x86_64/Packages/container-selinux-2.9-4.el7.noarch.rpm
sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
sudo yum install docker-ce
docker --version
service docker start
service docker status
usermod -aG docker dockeradmin
```

----- Docker admin user should be part of `docker` group

Note: --Docker must be install in both Ansible Server(master) and Docker host Server(client) also below command must be run in both the server(master and client)

```
[root@ip-172-31-89-34 docker]# docker login --username=rajguptaaws --password=aurangabad
```

Install below 2 plugging in Jenkins
publish over ssh and Deploy to container

then create a job in Jenkins and give the below in different tab: -

Source Code Management:

- Repository : https://github.com/rajkumargupta14/hello-world.git
- Branches to build : `*/master`

Build:

- Root POM:`pom.xml`
- Goals and options : `clean install package`

`Before doing below step create the entry for ansible_server in` **Manage Jenkins-→**Configure System-→ **Publish over SSH-→ then click on ADD and fill the below**

For more details check the project 3

Note crate below in ansible server

```
[root@ip-172-31-40-233 ~]# mkdir /opt/docker
```

Post Steps

- *Send files or execute commands over SSH*
  - Name: `ansible_server`
  - Source files : `webapp/target/*.war`
  - Remove prefix : `webapp/target`
  - Remote directory : `//opt//docker`
- *Send files or execute commands over SSH*
  - Name: `ansible_server`
  - Source files : `Dockerfile`
  - Remote directory : `//opt//docker`
  - Exec Command:

```
cd /opt/docker
docker build -t raj_demo4 .
docker tag raj_demo4 rajguptaaws/raj_demo4
docker push rajguptaaws/raj_demo4
docker rmi raj_demo4 rajguptaaws/raj_demo4
```

1. Login to Docker host and check images and containers. (no images and containers)

```
[dockeradmin@ip-172-31-40-233 ~]$ docker images

REPOSITORY        TAG             IMAGE ID         CREATED          SIZE

[dockeradmin@ip-172-31-40-233 ~]$ docker ps

CONTAINER ID       IMAGE             COMMAND           CREATED           STATUS           PORTS            NAMES
```

2. login to docker hub and check. shouldn't find images with for raj_demo4

3. Execute Jenkins job

4. check images in Docker hub. Now you could able to see new images pushed to rajguptaaws Docker_Hub

   Note:- if you get access issue then run the below command

```
[ec2-user@ip-172-31-35-164 ~]$ sudo -i
[root@ip-172-31-35-164 ~]# cd /opt/docker
[root@ip-172-31-35-164 docker]# chown -R dockeradmin:dockeradmin /opt/docker

[root@ip-172-31-35-164 docker]#
```

Part-02 : Deploy Containers

In the ansible server

```
[root@ip-172-31-35-164 ~]# cd /opt
[root@ip-172-31-35-164 opt]# mkdir playbooks
[root@ip-172-31-35-164 opt]# ls
containerd  docker  playbooks
[root@ip-172-31-35-164 opt]# cd playbooks/
[root@ip-172-31-35-164 playbooks]# pwd
/opt/playbooks
[root@ip-172-31-35-164 playbooks]# su - dockeradmin
Last login: Fri Feb  1 11:59:39 UTC 2019 on pts/1
[dockeradmin@ip-172-31-35-164 ~]$ cd /opt
[dockeradmin@ip-172-31-35-164 opt]$ ls
containerd  docker  playbooks
```

```
[dockeradmin@ip-172-31-35-164 opt]$ sudo chown dockeradmin: dockeradmin playbooks
```

```
[dockeradmin@ip-172-31-35-164 opt]$ cd playbooks/

[dockeradmin@ip-172-31-35-164 playbooks]$ sudo vi create_docker_container.yml
```

Then copy the below code

```
- hosts: all
  tasks:
  - name: stop previous version docker
    shell: docker stop raj_demo4
  - name: remove stopped container
    shell: docker rm -f raj_demo4
  - name: remove docker images
    shell: docker image rm -f rajguptaaws/raj_demo4
  - name: create docker image

    shell: docker run -d --name raj_demo4 -p 8090:8080 rajguptaaws/raj_demo4
```

if you get the permission issue then run the below command in ansible server

```
[dockeradmin@ip-172-31-35-164 opt]$ sudo chown dockeradmin: dockeradmin playbooks
[dockeradmin@ip-172-31-35-164 opt]$ cd playbooks/
[dockeradmin@ip-172-31-35-164 playbooks]$ ls -ld
drwxr-xr-x. 2 dockeradmin dockeradmin 78 Feb  1 12:37 .
```

to run playbook manually use below command

```
[dockeradmin@ip-172-31-35-164 playbooks]$ ansible-playbook -v create_docker_container.yml
```

Now Add this script to Jenkins job.

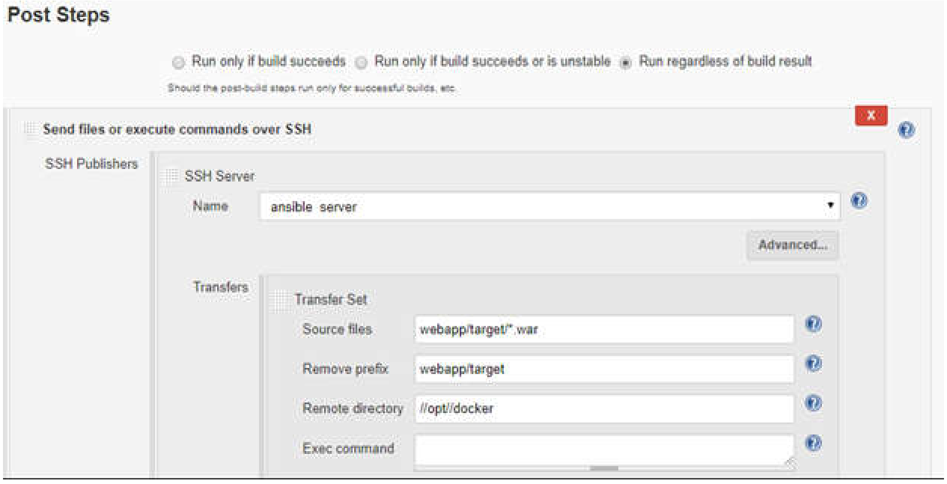
- Chose "*configure*" to modify your jenkins job.
  - *Under post build actions*

- Send files or execute commands over SSH
  - Exec Command:

```
cd /opt/playbooks
ansible-playbook -v create_docker_container.yml
```

1. Execute Jenkins job.
2. You could see a new container on your docker host. can able access it from browser on port 8090

http://54.152.73.16:8090/webapp/ -----Take the public ip of docker host



Post Steps

○ Run only if build succeeds ○ Run only if build succeeds or is unstable ● Run regardless of build result

Should the post-build steps run only for successful builds, etc.

Send files or execute commands over SSH

SSH Publishers

SSH Server

Name: ansible server

Transfers

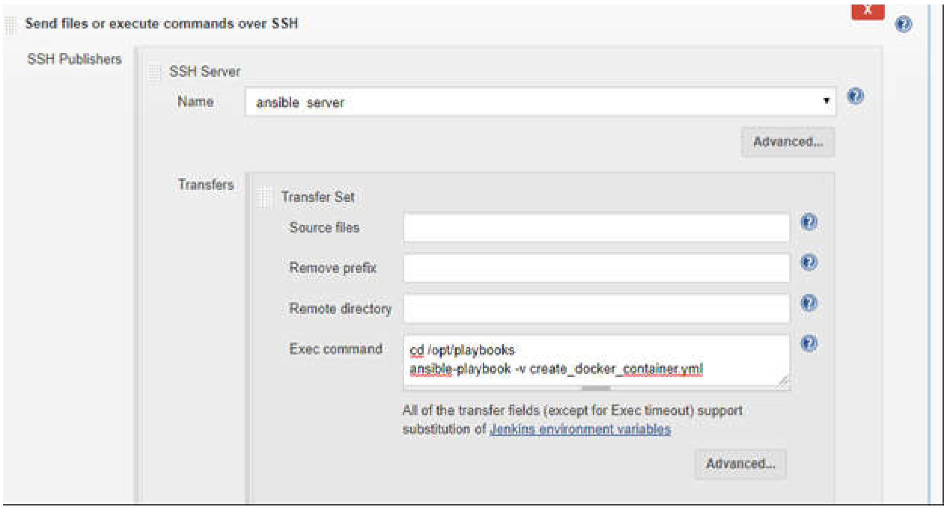
Transfer Set

Source files: webapp/target/*.war

Remove prefix: webapp/target

Remote directory: //opt//docker

Exec command:



Send files or execute commands over SSH

SSH Publishers

SSH Server

Name: ansible server

Transfers

Transfer Set

Source files:

Remove prefix:

Remote directory:

Exec command:

```
cd /opt/playbooks
ansible-playbook -v create_docker_container.yml
```

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

Part-03 : Deploy with Version Control Containers

we use 2 variables

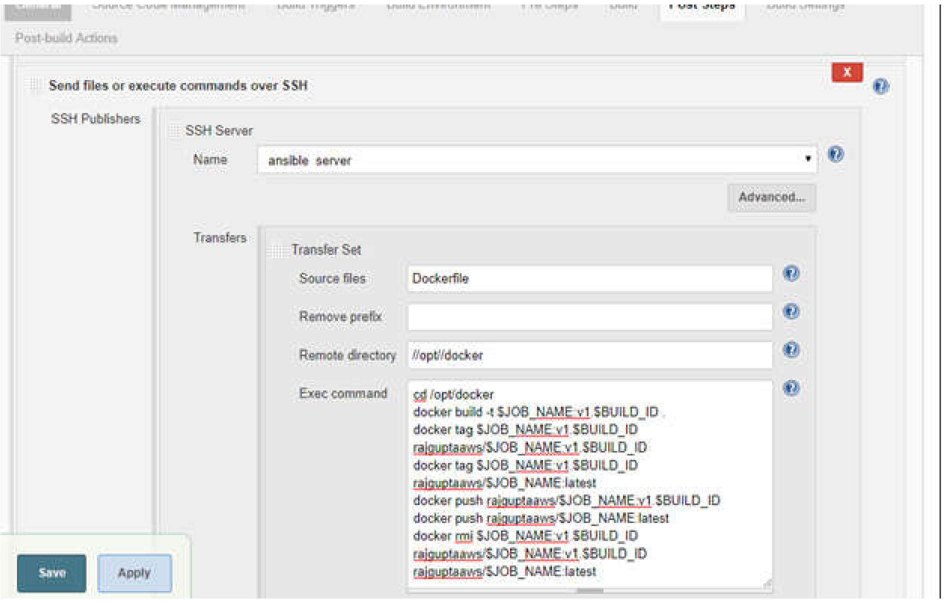
- `BUILD_ID` - The current build id of Jenkins (every time you click on build now it will create new build id like 1,2,3,4,5,6........)
- `JOB_NAME` - Name of the project of this build. This is the name you gave your job when you first set it up like in our case hellow-world

add the below part:-

Send files or execute commands over SSH

Name: ansible_server
Source files : Dockerfile
Remote directory : //opt//docker

```
cd /opt/docker
docker build -t $JOB_NAME:v1.$BUILD_ID .
docker tag $JOB_NAME:v1.$BUILD_ID rajguptaaws/$JOB_NAME:v1.$BUILD_ID
docker tag $JOB_NAME:v1.$BUILD_ID rajguptaaws/$JOB_NAME:latest
docker push rajguptaaws/$JOB_NAME:v1.$BUILD_ID
docker push rajguptaaws/$JOB_NAME:latest
docker rmi $JOB_NAME:v1.$BUILD_ID rajguptaaws/$JOB_NAME:v1.$BUILD_ID
rajguptaaws/$JOB_NAME:latest
```



Send files or execute commands over SSH

SSH Publishers

SSH Server

Name: ansible server

Transfers

Transfer Set

Source files: Dockerfile

Remove prefix:

Remote directory: //opt//docker

Exec command:

```
cd /opt/docker
docker build -t $JOB_NAME:v1.$BUILD_ID .
docker tag $JOB_NAME:v1.$BUILD_ID
rajguptaaws/$JOB_NAME:v1.$BUILD_ID
docker tag $JOB_NAME:v1.$BUILD_ID
rajguptaaws/$JOB_NAME:latest
docker push rajguptaaws/$JOB_NAME:v1.$BUILD_ID
docker push rajguptaaws/$JOB_NAME:latest
docker rmi $JOB_NAME:v1.$BUILD_ID
rajguptaaws/$JOB_NAME:v1.$BUILD_ID
rajguptaaws/$JOB_NAME:latest
```

Now do the changes in ansible code as per below yellow part:

```
- hosts: all
  tasks:
  - name: stop previous version docker
    shell: docker stop raj_demo4
  - name: remove stopped container
    shell: docker rm -f raj_demo4
  - name: remove docker images
    shell: docker image rm -f rajguptaaws/hellow-world:latest
  - name: create docker image

    shell: docker run -d --name raj_demo4 -p 8090:8080 rajguptaaws/hellow-world:latest
```