

End user will write dockerfile -> push to github -> Jenkins will start building docker images based on dockerfile
 -> once image build Jenkins will push docker image to docker hub -> Jenkins will create container based on latest image in web app server

part1.txt - Notepad

File Edit Format View Help

1st DevOps Project

CI & CD of docker based application using Jenkins Pipeline scripts(Groovy DSL)

Reference to My other Jenkins Pipeline Videos

steps:-

- 1.launch 2 ec2 (jenkins+Webapp)
- 2.ensure docker installed on both server
- 3.install jenkins on jenkins server
- 4.create pipeline & mention stages
 - a) git checkout
 - b) build docker Image
 - c) push docker image to dockerHub
 - d) Run container on webapp_server by pulling the same image from dockerhub
- 5.rectify if any issue comes

Create 2 ec2 instances : 1 for Jenkins , 1 for webapp

Install docker in both server .

```
116 sudo su
117 sudo yum update -y
118 sudo yum install docker.io
119 sudo amazon-linux-extras install docker -y
120 sudo yum install -y yum-utils
121 sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
122 sudo yum install docker-ce docker-ce-cli containerd.io
123 sudo systemctl start docker
124 sudo docker run hello-world
125 docker ps -a
```

```
126 sudo usermod -aG docker ec2-user
127 docker ps -a
```

```
[ec2-user@ip-172-31-40-214 ~]$ sudo systemctl start docker
[ec2-user@ip-172-31-40-214 ~]$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS          NAMES
43de6fe07bda   hello-world    "/hello"                21 minutes ago   Exited (0) 21 minutes ago          romantic_stonebraker
[ec2-user@ip-172-31-40-214 ~]$
```

Plugin required : ssh agent

```
/etc/systemd/jenkins not readable. Perhaps you forgot 'sudo' :
[ec2-user@ip-172-31-40-214 ~]$ sudo service jenkins start
Starting jenkins (via systemctl): [ OK ]
[ec2-user@ip-172-31-40-214 ~]$
```

Create new pipeline : pipeline – Add pipeline script –

CICD_DOCKER_BUILD_WEBAPP

» Required field



Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.



Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

```
&tcpus=&cpusetmems=&cpushares=0&dockerfile=Dockerfile&labels=%7B%7D&memory=0&memswap=0&networkmode=default&r
l": dial unix /var/run/docker.sock: connect: permission denied
```

```
root@ip-172-31-29-203: /home/ubuntu
```

```
root@ip-172-31-29-203:/home/ubuntu# chmod 777 /var/run/docker.sock
```

To resolve above error .

```
[ec2-user@ip-172-31-40-214 ~]$ chmod 777 /var/run/docker.sock
chmod: changing permissions of '/var/run/docker.sock': Operation not permitted
[ec2-user@ip-172-31-40-214 ~]$ sudo su -
Last login: Thu Apr  6 06:32:30 UTC 2023 on pts/0
[root@ip-172-31-40-214 ~]# chmod 777 /var/run/docker.sock
[root@ip-172-31-40-214 ~]# exit
logout
[ec2-user@ip-172-31-40-214 ~]$
```

```
ERROR: invalid tag "CICD_DOCKER_BUILD_WEBAPP:v1.4": repository name must be lowercase
```

```
[Pipeline] }
```

Do changes in pipeline script and run .

Once docker build image done .

```

[ec2-user@ip-172-31-40-214 ~]$ docker images
REPOSITORY                                TAG      IMAGE ID      CREATED        SIZE
cicd_docker_build_webapp                 v1.8     6f87587c9cdf  2 minutes ago  301MB
sravtar/cicd_docker_build_webapp         latest   6f87587c9cdf  2 minutes ago  301MB
sravtar/cicd_docker_build_webapp         v1.8     6f87587c9cdf  2 minutes ago  301MB
cicd_docker_build_webapp                 v1.7     3bc608636fde  3 minutes ago  309MB
hello-world                             latest   feb5d9fea6a5  18 months ago  13.3kB
[ec2-user@ip-172-31-40-214 ~]$

```

Images will be present in server .

Now we need to push images to docker hub .

Generate pipeline script to push docker image to docker hub – with creds – bind creds to variable . – add secret text . – creds username with password .

sample step

withCredentials: Bind credentials to variables

withCredentials ?

Secret values are masked on a best-effort basis to prevent help for details and usage guidelines.

Bindings

Add ▲

Filter

G


- Certificate
- Git Username and Password
- SSH User Private Key
- Secret ZIP file
- Secret file
- Secret text


Once image pushed to dockerhub , we can remove from Jenkins .

Add in pipeline script .


sravtar / cicc_docker_build_webapp

Description





This repository does not have a description 

 Last pushed: a few seconds ago

Tags

 IMAGE INSIGHTS INACTIVE
[Activate](#)

This repository contains 2 tag(s).

Tag	OS	Type	Pulled	Pushed
 latest		Image	---	a few seconds ago
 v1.9		Image	---	a few seconds ago

[See all](#)

[Go to Advanced Image Management](#)

Image pushed to dockerhub .

Now Jenkins will ssh to webapp server using sshagent plugin and create container .

Create ec2 for webapp .

Install docker .

- 1 sudo apt-get update
- 2 sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common
- 3 curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
- 4 sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu \$(lsb_release -cs) stable"
- 5 sudo apt-get update
- 6 sudo apt-get install docker-ce docker-ce-cli containerd.io
- 7 sudo docker run hello-world
- 8 sudo docker ps
- 9 sudo docker ps -a
- 10 history

Create pipeline script for sshagent to webapp .

Sample Step

sshagent: SSH Agent

sshagent ?

ubuntu (WEBAPP) ▼

?

+ Add

☐ Ignore missing credentials ?

Generate Pipeline Script

```
sshagent(["WEBAPP"]) {  
  // some block  
}
```

Ssh username and private key .

Once pipeline success .

Image will be deployed in webapp .

```
ubuntu@ip-172-31-38-131:~$ docker ps -a  
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES  
f5cec6c1032b   sravtar/cicd_docker_build_webapp   "/usr/sbin/httpd -D ..." About an hour ago Up About an hour 22/tcp, 0.0.0.0:9000->80/tcp, :::9000->80/tcp scriptedcontainer  
156a4f961cd1   hello-world                         "/hello"                 2 hours ago   Exited (0) 2 hours ago                               focused_brahmagupta  
ubuntu@ip-172-31-38-131:~$
```

As we can see container is running .

We can access it using 9000 port .

launch-wizard-8 sg-088c29cb95c9d177c launch-wizard-8 created 2023-03-14T14:02:49.386Z vpc-0493d4c3ea8528d4d

Owner 240588757964 Inbound rules count 5 Permission entries Outbound rules count 1 Permission entry

Inbound rules Outbound rules Tags

You can now check network connectivity with Reachability Analyzer [Run Reachability Analyzer](#)

Inbound rules (5) [Manage tags](#) [Edit inbound rules](#)

Filter security group rules

<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range
<input type="checkbox"/>	-	sgr-014be9da3e80afdad	IPv4	SSH	TCP	22
<input type="checkbox"/>	-	sgr-015cd8e8ae47dc387	IPv4	HTTPS	TCP	443
<input type="checkbox"/>	-	sgr-0ef568505e2701075	IPv4	HTTP	TCP	80
<input type="checkbox"/>	-	sgr-0926da396134f84c5	IPv4	Custom TCP	TCP	9000
<input type="checkbox"/>	-	sgr-00245cae8198a91...	IPv4	Custom TCP	TCP	8080

Allow inbound rule in security group for 9000 port .

SHINE

Welcome Template Stock

Lorem ipsum dolor sit amet, ed do eiusmod tempor incididunt.

Site is accessible .

Now if we rerun pipeline it will fail as same container name already exists so we have to make changes in pipeline script .

So container is deployed to webapp using docker image .