==========================================================================

Integrating docker in CI/CD pipeline:

-------------------------------------

Git-> github-> jenkins->docker host

|

maven

**step 1) setup docker host:**

->setup linux instance

-> install docker

-> start docker services

-> basic docker commands

---------

->setup linux instance in AWS

create an EC2 instance which will act as docker host

create inbound rule at allow 8080 port

connect to ec2 instance

install docker

**amazon linux:**

-------------

**Installation Steps**

Install docker and start docker services

yum install docker -y

docker --version

# start docker services

service docker start

service docker status

**Create a user called dockeradmin**

useradd dockeradmin

passwd dockeradmin

**add a user to docker group to manage docker**

usermod -aG docker dockeradmin

Validation test

Create a tomcat docker container by pulling a docker image from the public docker registry

docker run -d --name test-tomcat-server -p 8090:8080 tomcat:latest

Ref: centos: <https://docs.docker.com/engine/install/centos/>

#install-using-the-repository

**create a docker container:**

-------------------------

docker run -d -name tomcat-container -p 8081:8080 tomcat

open ports if you create instance in AWS

**login to container:**

docker exec -it tomcat-container bash

cd /usr/local/tomcat

cd webapps

ls

cd ..

cd webapps.dist

**copy everything from webapps directory**

cp -R \* ../webapps

*as these changes are temporary , if container is restarted data will not be persistent*

**to check this stop the container**

# docker stop <container name>

**create new container and see if you were able to access the page**

docker run -d -name tomcat2 -p 8082:8080 tomcat:latest

=============================================

**create docker file to fix the above issue:**

**steps to create docker file**

install tomcat on centos

install java

create /opt/tomcat directory

change work directory to /opt/tomcat

download tomcat packages

extract tar.gz file

rename to tomcat directory

tell to docker that it runs on port 8080

start tomcat services

**tomcat docker file**

FROM centos

RUN yum install java -y

RUN mkdir /opt/tomcat/

WORKDIR /opt/tomcat

ADD https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.54/bin/apache-tomcat-9.0.54.tar.gz /opt/tomcat

RUN tar -xvfz apache-tomcat-9.0.54.tar.gz

RUN mv apache-tomcat-9.0.54/\* /opt/tomcat

EXPOSE 8080

CMD ["/opt/tomcat/bin/catalina.sh", "run"]

**build the image:** docker built . -t mytomcat

# docker images --- to check the created images

**run the container**

docker run -d --name mytomcat -p 8083:8080 mytomcat

docker ps

**hit the url and see if we were able to access**

remove the docker file which we created above

=========================================================================

**Docker file to update the contents in container**

vi Dockerfile

FROM tomcat:latest

RUN cp -R /usr/local/tomcat/webapps.dist/\* /usr/local/tomcat/webapps

docker build . -t mytomcat

docker run -d --name mytomcat-server-2 -p 8085:8080 mytomcat

=================================================================================

**Integrate docker with jenkins**

------------------------------  
*high level steps:*

create a dockeradmin user

install "publish over ssh" plugin

add dockerhost to jenkins "configure systems"

----------

login to docker host:

cat /etc/passwd

cat /etc/groups

docker:x:x:

useradd dockeradmin

passwd dockeradmin

id dockeradmin

**add dockeradmin user to docker group**

#usermod -aG docker dockeradmin

to get password prompt while login to server using "dockeradmin" user, do below changes because by default ec2 instances doesnt allow password based authentication

vi /etc/ssh/sshd\_config

uncomment PasswordAuthentication yes

comment PasswordAuthentication no

service sshd reload

-----------

**To install plugin:**

login to jenkins

mange jenkins-> manage plugins -> available tab-> install "publish over ssh " plugin

**To configure:**

loginto jenkins

manage jenkins -> configure system -> SSH servers ->

update NAME, Hostname/ipaddress, username,- > advanced -> password based authentication ->

apply and save

--------------------------

**jenkins job to build and copy arifacts on docker host**

login to jenkins

new item -> BuildandDeployContainer-> copy from Build and deployjob -> validate all config -> post build actions -> send build artifacts over ssh ->

SSH server -> docker host -> source files-> webapp/target/\*.war -> remove prefix: webapp/target -> remotedirectory: -> apply and save

war file will be copied into docker host in users home path (/home/dockeradmin)

----------------------------------------------------------

**loginto docker host**

cd /opt

mkdir docker

chown dockeradmin:dockeradmin docker

**Docker file consists of creating container using tomcat image**

mv Dockerfile /opt/docker

**update jenkins job config**

mention the path //opt//docker in job

apply and save

----------------------

**update the docker file**

FROM tomcat:latest

RUN cp -R /usr/local/tomcat/webapps.dist/\* /usr/local/tomcat/webapps

COPY ./\*.war /usr/local/tomcat/webapps

docker build . -t mytomcat:v1

docker images

run the below command to start container

docker run -d --name tomcatv1 -p 8086:8080 tomcat:v1

hit the url : http://webapp

---------------------------------------------

**automate build and deploy on docker container**

**update existing job buildand deployon container**

postbuild actions

cd /opt/docker;

docker build . -t regappv1

docker run -d --name registerapp -p 8087:8080 regappv1

apply and save

----------------------------------------

docker container prune -a - to delete all stopped conatiners images

docker images

run build now or update index page using git to trigger the job

-----------------------------------------

cd /opt/docker;

docker build . -t regappv1

docker stop registerapp

docker rm registerapp

docker run -d --name registerapp -p 8087:8080 regappv1

build the job again

-----------------------------------------------

=>Instead of executing the commands we will ansible server to deploy a container

git - github - jenkins - ansible - docker

|

- maven

-------------------------------------