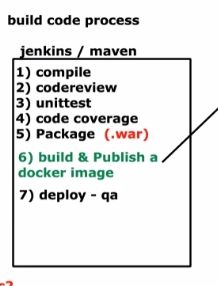
19 - 14 Feb 2023 (CD with Jenkins)

14 February 2023

20:36

Agenda:

CD with Jenkins



Step by step for CD usign Jenkins using docker and ansible

<https://github.com/lerndevops/labs/blob/master/cicd-flow/README.md>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Guide to CREATE A CICD Pipeline on Jenkins Using Docker & Ansible**  **Step1 -- Pre-Requisite**  Create a VM (on any cloud OR Virtual Box) with 1CPU 2GB RAM  login to the VM Install below tools  **Step2 -- Install Java & Jenkins**  sudo wget <https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installJenkins.sh> -P /tmp sudo chmod 755 /tmp/installJenkins.sh sudo bash /tmp/installJenkins.sh  **Step3 -- Install Maven**  sudo wget <https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installMaven.sh> -P /tmp sudo chmod 755 /tmp/installMaven.sh sudo bash /tmp/installMaven.sh  **Step4 -- Install Docker**  sudo wget <https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installDocker.sh> -P /tmp sudo chmod 755 /tmp/installDocker.sh sudo bash /tmp/installDocker.sh  **Step5 -- Install Ansible**  sudo wget <https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installAnsible.sh> -P /tmp sudo chmod 755 /tmp/installAnsible.sh sudo bash /tmp/installAnsible.sh  # modify the ansible config file to ensure disable host key checking  vi /etc/ansible/ansible.cfg  # uncomment this to disable SSH key host checking host\_key\_checking = False  **Step6 -- Login to Jenkins UI**  **hit**[**http://IP:8080**](http://IP:8080)**in browser ## incase of cloud please use Public IP ensure the Port is allowed to access**          enter `initialAdminPassword` the page to login ( cat /var/lib/jenkins/secrets/initialAdminPassword )  click on `Install Suggested Plugins`                  continue next and finish the setup.  **Step7 -- Install reqired Plugins (Install from Jenkins UI)**  install all these from Jenkins UI  Manage Jenkins --> manage plugins -- Available -- search & install the below          1) warnings NG          2) jacoco          3) Junit          4) Build Pipeline          5) Docker Piepeline  **Step8 -- Create Credentials (Setup from Jenkins UI)**  Manage Jenkins --> Manage Credentials ==> Stores scoped to Jenkins - global ==> Add Credentials          --> kind: username with password          --> scope: Global         --> username: <enter your docker hub id>         --> password: <enter your docker hub password>          --> ID: DOCKER\_HUB\_LOGIN          --> Description: DOCKER\_HUB\_LOGIN  Jenkins -> Manage Credential  Kind -> docker hub username  Tread usernam as secret  Pwd          **Step9 -- Configure JAVA - MAVEN - Git (Setup from Jenkins UI)**  Java configuration in Jenkins console                   Manage Jenkins --> Global Tool Configuration --> JDK --> Add JDK                 Name: myjava ( can be any string )                 JAVA\_HOME: /path/to/javahome ( ex: /usr/lib/jvm/java-8-openjdk-amd64 )  Maven Configuration in Jenkins console                  Manage Jenkins --> Global Tool Configuration --> Maven --> Add Maven                 Name: maven3.6 ( can be any string )                 MAVEN\_HOME: /path/to/mavenhome ( ex: /opt/apache-maven-3.6.5 )  Git Configuration in Jenkins console                  Manage Jenkins --> Global Tool Configuration --> Git --> Add Git                 Name: git ( can be any string )                 MAVEN\_HOME: /path/to/githome ( ex: /usr/bin/git )    **Step10 -- configure Jenkins with Docker - from Jenkin Server CLI**  by default Jenkins process runs with Jenkins User, which mean any jenkins Jobs we run from jenkins console will be running jenkins user on Jenkins machine  we need to configure Jenkins user can run the docker commands by adding jenkins user to docker group  sudo usermod -aG docker jenkins  **restart the Jenkins Service**  sudo service jenkins restart  validate, run docker command with jenkins          su - jenkins ## switch to jenkins user         docker ps ## to list any containers running         docker pull nginx ## pull a docker image  **if the above commands execute without any error then we configured jenkins user properly**    **Step11 -- Setup Deployment Environments**  **setup atleast one docker swarm / kubernetes cluster**  create two VMs for QA, Install the required toos & setup the kubernetes cluster, one master - one worker node          **Step12 -- Setup Ansible Inventory on Jenkins machine using CLI**  vi /tmp/inv   enter your servers in groups like qa & prod   sudo chmod 755 /tmp/inv  sudo chown jenkins:jenkins /tmp/inv    // look at the sample inventory file under <https://raw.githubusercontent.com/lerndevops/PetClinic/master/deploy/inv>     Note: ensure to put only manager IPs in inventory file -- DO NOT PUT NODE Ips    From Jenkins server    **Step 13: Now Let's start creating CICD Pipeline Using Pipeline As Code Script**  **Jenkins ( home page )**  --> Click on New Item from left menu   --> Enter an item name: CICD-Pipeline   --> Choose: Pipeline   --> Click: ok  **insdie job parameters as below**  --> Click on Pipeline (TAB) on top  --> Definition (drop down): Pipeline Script from SCM --> SCM (drop down): Git --> Repositories --> Repositories URL --> <https://github.com/lerndevops/PetClinic> --> leave the other values Default for this Demo --> Script Path: cicd.gvy          --> Note: Script is already availabe at <https://github.com/lerndevops/PetClinic/blob/master/cicd.gvy> --> Click on Save  --> Build Now from left Menu    Pipeline as a code      Add 2 more stages:    <https://github.com/lerndevops/samplejavaapp>  <<pac.gvy.txt>>  Create Pipeline Job   |  |  | | --- | --- | | Before | After CI CD Pipeline step | |  | Docker hub      From Kube server | |  |  |     Docker command generation                          From <<https://github.com/lerndevops/labs/blob/master/cicd-flow/README.md>> |  |



