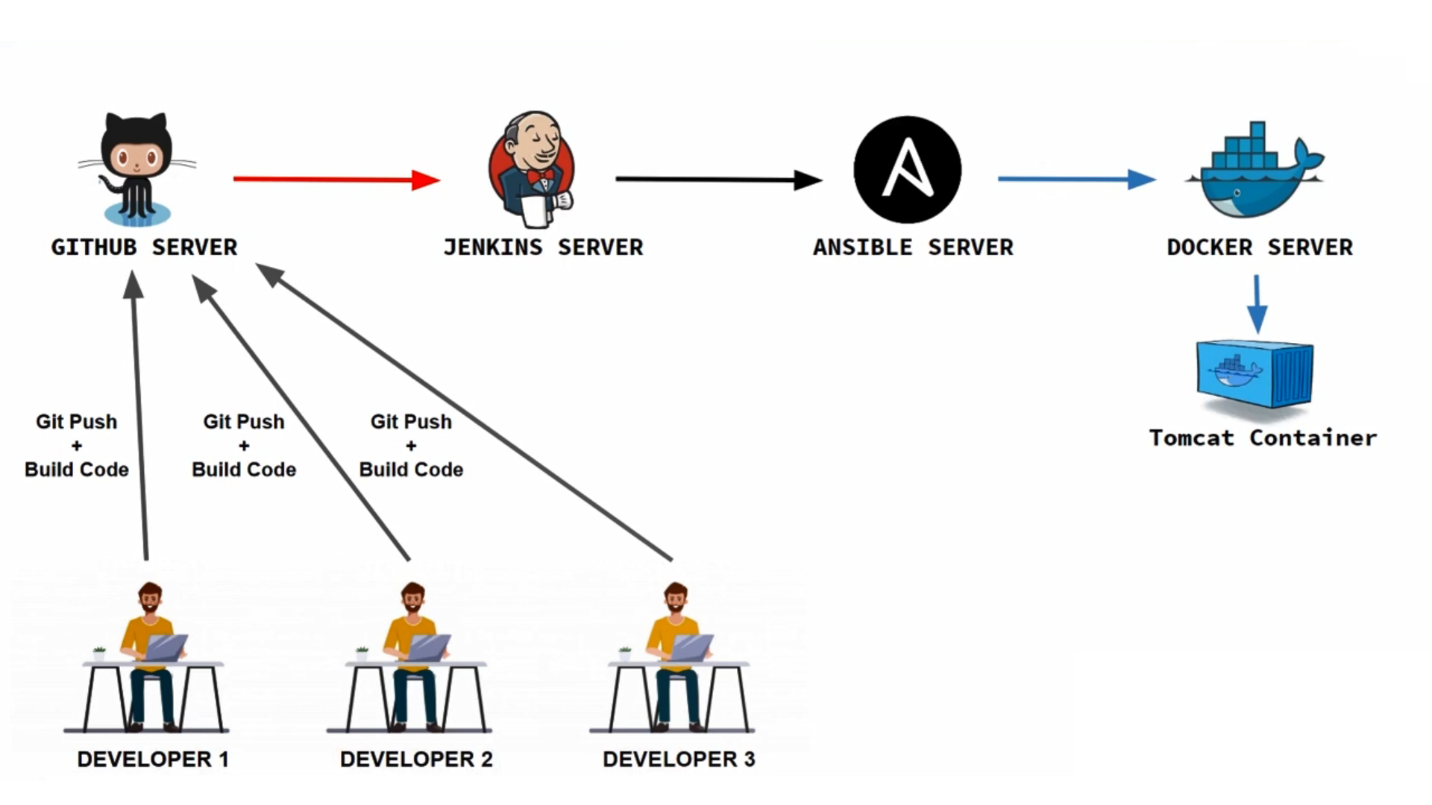
CICD CONFIGRATION – Using Git,ansible,Jenkins,docker,tomcat.



Requirement 3 Servers (AMI Linux 2):

1. Jenkins\_Server 192.168.1.6
2. Ansible\_Server
3. Docker\_Server

**Login on Jenkins\_Server:**

1. **Set hostname**

#hostnamectl set-hostname jenkins

#hostname

#bash

1. **Install java**

#yum update -y

#amazon-linux-extras install java-openjdk11 -y

For AL2023

sudo yum install java-11-amazon-corretto-headless

sudo yum install java-11-amazon-corretto

sudo yum install java-11-amazon-corretto-devel

sudo yum install java-11-amazon-corretto-jmods

#java -version

1. **Create tomcat user and group**

#groupadd --system tomcat

#useradd -d /usr/share/tomcat -r -s /bin/false -g tomcat tomcat

1. **Install Tomcat 9 on Amazon Linux 2**

#wget https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.63/bin/apache-tomcat-9.0.63.tar.gz

1. **Use tar command line tool to extract downloaded archive.**

#tar -xvzf apache-tomcat-9.0.63.tar.gz -C /usr/share/

1. **Create Symlink to the folder /usr/share/tomcat. This is for easy updates.**

#ln -s /usr/share/apache-tomcat-9.0.63/ /usr/share/tomcat

1. **Update folder permissions**

#chown -R tomcat:tomcat /usr/share/tomcat

#chown -R tomcat:tomcat /usr/share/apache-tomcat-9.0.63/

1. **Create Tomcat Systemd service**

Run following code to create tomcat service in a single command on terminal

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

#tee /etc/systemd/system/tomcat.service<<EOF

[Unit]

Description=Tomcat Server

After=syslog.targetnetwork.target

[Service]

Type=forking

User=tomcat

Group=tomcat

Environment=JAVA\_HOME=/usr/lib/jvm/jre

Environment='JAVA\_OPTS=-Djava.awt.headless=true'

Environment=CATALINA\_HOME=/usr/share/tomcat

Environment=CATALINA\_BASE=/usr/share/tomcat

Environment=CATALINA\_PID=/usr/share/tomcat/temp/tomcat.pid

Environment='CATALINA\_OPTS=-Xms512M -Xmx1024M'

ExecStart=/usr/share/tomcat/bin/catalina.sh start

ExecStop=/usr/share/tomcat/bin/catalina.sh stop

[Install]

WantedBy=multi-user.target

EOF

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

1. **Enable and start tomcat service:**

#systemctl daemon-reload

#systemctl start tomcat

#systemctl enable tomcat

#systemctl status tomcat

1. **Allow Tomcat TCP port 8080 if you have firewalld service enabled. (Note – For Centos/RHEL8/9)**

#firewall-cmd --permanent --add-port=8080/tcp

#firewall-cmd --reload

1. **Configure Tomcat Authentication**

We have to edit Tomcat configuration file to enable Admin and Manager UI roles.

#vim /usr/share/tomcat/conf/tomcat-users.xml

Add below lines before closing with </tomcat-users>

<role rolename="admin-gui"/>

<role rolename="manager-gui"/>

<user username="admin" password="111" fullName="Administrator" roles="admin-gui,manager-gui"/>

**:wq**

1. **Configure Apache web server as a proxy for Tomcat server. First install httpd package.**

#yum install httpd -y

1. **Create VirtualHost file for Tomcat Admin web interface:**

#vim /etc/httpd/conf.d/tomcat\_manager.conf

<VirtualHost \*:80>

ServerAdmin root@localhost

ServerName tomcat.example.com

DefaultType text/html

ProxyRequests off

ProxyPreserveHost On

ProxyPass / http://localhost:8080/

ProxyPassReverse / http://localhost:8080/

</VirtualHost>

<VirtualHost \*:80>

ServerName ajp.example.com

ProxyRequests Off

ProxyPass / ajp://localhost:8009/

ProxyPassReverse / ajp://localhost:8009/

</VirtualHost>

**:wq**

1. **If SELinux is enabled run the following commands:**

#setsebool -P httpd\_can\_network\_connect 1

#setsebool -P httpd\_can\_network\_relay 1

#setsebool -P httpd\_graceful\_shutdown 1

#setsebool -P nis\_enabled 1

1. **Restart and enable httpd service:**

#systemctl restart httpd

#systemctl enable httpd

#systemctl status httpd

1. **Open Web Browser and open address**

<http://192.168.1.111>

login tomcat with proper authentication

username: admin

password: 111

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

1. **Open terminal and configure jenkins**

#cd /usr/share/tomcat/

#ls

#cd webapps

1. **Download jenkins war file**

# wget <https://get.jenkins.io/war-stable/2.414.3/jenkins.war>

**Restart tomcat service and open jenkins**

#systemctl restart tomcat

1. **Open Web Browser and open address**

http://192.168.1.24/jenkins

1. **Login into jenkins**

go to linux get default password using following command

#cat /usr/share/tomcat/.jenkins/secrets/initialAdminPassword

Getting Started - Customize Jenkins – Install suggested plugins.

1. **Set jenkins new credentials**

username: admin

password: 111

**Login in Jenkin Dashboard.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***Take Snapshot (Jenkins Working)**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. **Set java path**

**Verify Version**

#java -version

Update path in bash profile

#cd

#vim .bash\_profile

JAVA\_HOME=/usr/lib/jvm/java-11-amazon-corretto.x86\_64

PATH=$PATH:$JAVA\_HOME:$HOME/bin

:wq

#echo $PATH

#source .bash\_profile

#echo $PATH

1. **Install Maven Amazon Linux AMI2**

Following are the set of commands need to be executed sequentially to install maven.

#cd /opt/

#wget <https://dlcdn.apache.org/maven/maven-3/3.8.8/binaries/apache-maven-3.8.8-bin.tar.gz>

#tar-xvzf apache-maven-3.8.8-bin.tar.gz

#rm -rvf \*.gz

#mvn –version

1. **Set path for maven**

#cd

#vim .bash\_profile

MAVEN\_HOME=/opt/apache-maven-3.8.8

M2=/opt/apache-maven-3.8.8 /bin/

PATH=$PATH:$JAVA\_HOME:$MAVEN\_HOME:$M2:$HOME/bin

:wq

#source .bash\_profile

#echo $M2

#echo $MAVEN\_HOME

#mvn --version

1. **Install git:**

#yum update -y

#yum install git -y

#git --version

#which git

1. **Open Jenkins**

<http://localhost/jenkins>

Dashboard – Manage Jenkins – Tools – git – Name (default) – path to git execute (/bin/git)

Maven – add maven – Name (mvn) – MAVEN\_HOME(/opt/apache-maven-3.9.0/)– Apply – Save.

Open Jenkins

<http://localhost/jenkins>

Dashboard – New Item – Enter new item name (devops\_pro) – freestyle project – ok – source code management – git – repository url ( <https://github.com/hackwithabhi1/hello-world.git> ) - Build Steps (invoke top-level maven targets ) – Maven version (mvn) – Goals ( clean install package ) – Apply – Save.

Dashboard – dev\_pro – Build Now – console output.

**Steps for ansible server:**

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

**Login in system2.**

#hostnamectl set-hostname ansible

**install ansible**

#yum install ansible -y

#wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm

#yum install epel-release-latest-7.noarch.rpm

#yum install ansible -y

#ansible --version

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

**Create auser user on both system (2 System- Ansible\_Server&Docker\_Server)**

#useradd auser

#passwd auser

111

111

#vim /etc/sudoers

auser ALL=(ALL) NOPASSWD: ALL # ( on line 101)

:wq!

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

**Login as aauser and establishssh based authentication with Docker\_server**

#su auser

$ssh-keygen

$ssh-copy-id auser@192.168.1.113 (client ip)

**update inventory file**

#sudo mkdir /opt/playbook

$sudo vim /opt/playbook/inventory

[docker]

192.168.1.113

:wq

**Create ansible.cfg file**

#sudo vim /opt/playbook/ansible.cfg

[defaults]

inventory=/opt/playbook/inventory

remote\_user=auser

host\_key\_checking=false

[privilege\_escalation]

become=true

become\_user=root

become\_method=sudo

become\_ask\_pass=false

:wq

**Check ansible client (docker server) is reachable through ansible server**

$cd /opt/playbook/

$ansible all -m ping

$su root

Install docker on ansible server

#yum install docker -y

#yum info docker

#systemctl start docker

#systemctl enable docker

#systemctl status docker

#usermod -aG docker auser

#mkdir /opt/docker

#chown -R auser:auser /opt/docker

**Login on Docker-Server1:**

go to ansible client system

#hostnamectl set-hostname docker

#yum install docker -y

#yum info docker

#systemctl start docker

#systemctl enable docker

#systemctl status docker

#usermod -aG docker auser

#mkdir /opt/docker

#chown -R auser:auser /opt/docker

Login ansible server

#suauser

$sudo vim /opt/playbook/create-docker-container.yml

---

- name: create docker container

hosts: all

ignore\_errors: yes

tasks:

- name: stop the running container

shell: docker stop webapp

- name: remove container

shell: docker rm -f webapp

- name: remove container image

shell: docker rmi -f sevenmentor:latest

- name: transfer docker file

copy:

src: /opt/docker/Dockerfile

dest: /opt/docker/

- name: trasfer web appplication

copy:

src: /opt/docker/webapp.war

dest: /opt/docker/

- name: build docker image

shell: cd /opt/docker; docker build -t sevenmentor .

- name: create docker tomcat container

shell: docker run -dt --name webapp -p 8090:8080 sevenmentor:latest

:wq

**Login Jenkins**

Open Jenkins

<http://localhost/jenkins>

Dashboard- manage Jenkins – manage plugins – available – install “publish over ssh” plugins – restart Jenkins.

Dashboard – manage Jenkins – configure system – system –SSH Servers (add ) – Name (Ansible\_Server) Hostname (192.168.1.112) – username (auser) – Advanced - use password authentication or use a different key (111) – Test Configuration – Apply - Save.

Open Dashboad – DevOps\_Project – Configure-

Now click on “Post Build Action” – send build artifacts over SSH – Name (Ansible\_Server) – Source files (webapp/target/\*.war) – Remove prefix (webapp/target/) – Remote directory (//opt/docker)

Add Server

Name (Ansible\_Server) – Source files (Dockerfile) – Remote directory (//opt/docker)

Add Server

Exec command

cd /opt/playbook; ansible-playbook create-docker-container.yml

Save - Apply

Goto docker server

#docker images

#dockerps

#dockerps -a

#ifconfig

Run Project

Go to Jenkins dashboard – run created project (right click and build)

Open web browser –

http://dockerip:8090/webapp