Create CICD Pipeline to build and deploy java application using plain maven build and export war file to GCS Bucket. Deploy the War file from bucket to Server

Step1: Create a Compute Instance VM with Default - Service Account (SA) and Default VPC (in VPC firewall Add-fire wall rule: All-Ports – Any Where).

Step2: Goto IAM – Select the Default Compute Engine- SA - Assign Roles: Storage Admin & Storage Object User & Generate the **JSON Key**

Step3: Login to VM – Instance SSH – Install Maven-Jenkins-java17- Git and Tomcat.

Copy and paste the below scripts for Installation Process:

RUN MANUALLY:

- **Installing your desired maven version in particular path:**
- Download Maven:

wget https://apache.osuosl.org/maven/maven-3/3.9.5/binaries/apache-maven-3.9.5-bin.tar.gz

- **Extract the archive:** tar xzvf apache-maven-3.9.5-bin.tar.gz

- **Move Maven to /opt directory (optional):** sudo mv apache-maven-3.9.5 /opt

- **Set environment variables:** vim ~/.bashrc (copy paste below command)

```
export M2_HOME=/opt/apache-maven-3.9.5

export PATH=$M2_HOME/bin:$PATH
```

- **Run This Command:** source ~/.bashrc
- **Verify installation:** mvn --version

Jenkins & Git - Installation:

Vim jenkins.sh (Copy & Paste the below script in this file):

```
 sudo apt update
Sudo apt install git -y
sudo apt install openjdk-17-jre -y
  curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \
    /usr/share/keyrings/jenkins-keyring.asc > /dev/null
  echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
    https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
    /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins –y
```

To Run The Script : sh jenkins.sh

Tomcat Installation:

Vim tomcat.sh (Copy & Paste the below script in this file):

```
sudo apt install default-jdk -y
wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.88/bin/apache-tomcat-9.0.88.tar.gz
tar -zxvf apache-tomcat-9.0.88.tar.gz
sed -i '56  a\<role rolename="manager-gui"/>' apache-tomcat-9.0.88/conf/tomcat-users.xml
sed -i '57  a\<role rolename="manager-script"/>' apache-tomcat-9.0.88/conf/tomcat-users.xml
sed -i '58  a\<user username="tomcat" password="tomcat" roles="manager-gui, manager-script"/>' apache-tomcat-9.0.88/conf/tomcat-users.xml
sed -i '59  a\</tomcat-users>' apache-tomcat-9.0.88/conf/tomcat-users.xml
sed -i '56d' apache-tomcat-9.0.88/conf/tomcat-users.xml

sed -i '21d' apache-tomcat-9.0.88/webapps/manager/META-INF/context.xml

sed -i '22d'  apache-tomcat-9.0.88/webapps/manager/META-INF/context.xml

sh apache-tomcat-9.0.88/bin/startup.sh
```

To Run Script : sh jenkins.sh

NOTE: In the script for Login given: PASSWORD=USERNAME=tomcat

Step4: After Installation of Jenkins and Tomcat – Change Port Number for Jenkins.

Follow the below script to change the port number:

vim script.sh (copy paste below code and run script)

```
#Changing port from 8080 to 8082

sudo sed -i 's/Environment="JENKINS_PORT=8080"/Environment="JENKINS_PORT=8081"/' /usr/lib/systemd/system/jenkins.service

sudo systemctl daemon-reload

sudo systemctl restart jenkins.service
```

To Run the script: sh script.sh

Step5: Now login to the Jenkins Console with the use of External IP and Jenkins Port Number

Step6: Setup Your Jenkins Console & create a Pipeline Job.

Setp7: Goto Manage Jenkins -> Plugins -> Install **Google Cloud Storage & Deploy to Container**

Step8: Manage Jenkins -> Credentials -> Global -> Add Credentials -> kind (Google Service Account from Private Key) -> project ID -> select JSON Key.

Step9: Now write the below Pipeline-Script:

```
pipeline {
  agent any
  tools {
    maven 'maven' // Use the name configured in Jenkins
  }
  stages {
    stage('checkout') {
      steps {
        git 'https://github.com/summu97/monolythic-project.git'
      }
    }
    stage('package') {
      steps {
        sh 'mvn clean package'
      }
    }
    stage('storage') {
      steps {
        googleStorageUpload bucket: 'gs://YOUR_BUCKET_NAME/', credentialsId: 'YOUR_JSON_KEY_CREDENTIAL_ID', pattern: 'target/*.war'
      }
    }
    stage ('Download from GCS') {
      steps {
        googleStorageDownload bucketUri: ' YOUR_gsutil_URI ', credentialsId: 'YOUR_JSON_KEY_CREDENTIAL_ID', localDirectory: '.'
 // to gen
      }
    }
    stage('Deploy') {
      steps {
        deploy adapters: [
          tomcat9(
            credentialsId: 'YOUR_TOMCAT_CREDENTIAL_ID',
            path: '',
            url: 'TOMCAT_SERVER_URL'
            )
            ],
            contextPath: 'tomcat-netflix',
            war: 'target/NETFLIX-1.2.2.war'
      }
    }
  }
}
```

❖ ANOTHER APPROACH TO STORE WAR FILE TO STORAGE & THEN DOWNLOAD AND DEPLOY TO TOMCAT:

- **In VM Instance Terminal - Set password to jenkins-Change permissions to sudoers file:**
- chmod 640 /etc/sudoers
- vim /etc/sudoers (add below line and :wq)

- jenkins   ALL=(ALL:ALL) ALL

- chmod 400 /etc/sudoers

In Jenkins UI: Add above Jenkins password in credentials: Kind(secret text)

```
pipeline {
    agent any
    tools {
        maven 'maven' // Use the name configured in Jenkins
    }
        environment {
        MY_PASSWORD = credentials('Paste-jenkins-password-ID')
    }
    stages {
        stage('checkout') {
            steps {
                git 'https://github.com/summu97/monolythic-project.git'
            }
        }
        stage('package') {
            steps {
                sh 'mvn clean package'
            }
        }
        stage('storage') {
            steps {
                googleStorageUpload bucket: 'gs://YOUR_BUCKET_NAME', credentialsId: 'Paste-json-key-ID', pattern: 'target/*.war'
            }
        }
        stage('Copying war') {
            steps {
                sh 'gsutil cp gs://YOUR_BUCKET_NAME/target/NETFLIX-1.2.2.war .'
            }
        }
        stage('Deploy to container') {
            steps {
                sh '''
                    echo $MY_PASSWORD | sudo -S  mv  NETFLIX-1.2.2.war /root/apache-tomcat-9.0.88/webapps
                '''
            }
        }
    }
}
```