

DevOps Foundations: Version Control and CI/CD with Jenkins

Course-End Project 1: Insured Assurance

Objective

To create a GitHub Actions CI/CD pipeline workflow for invoking the deployment of a Java application as a Jenkins job using Tomcat Apache

Real-time scenario:

Insured Assurance, a leading global insurance provider based in the US, offers a range of products including home, health, car, and life insurance. The company is transitioning to a DevOps architecture and aims to automate code builds and deployments across various environments. To meet this need, it has adopted GitHub Actions for code checkout, building, and testing automation and Jenkins for continuous deployment. As a DevOps engineer at Insured Assurance, you are tasked with implementing a CI/CD pipeline using GitHub Actions and Jenkins.

Tasks

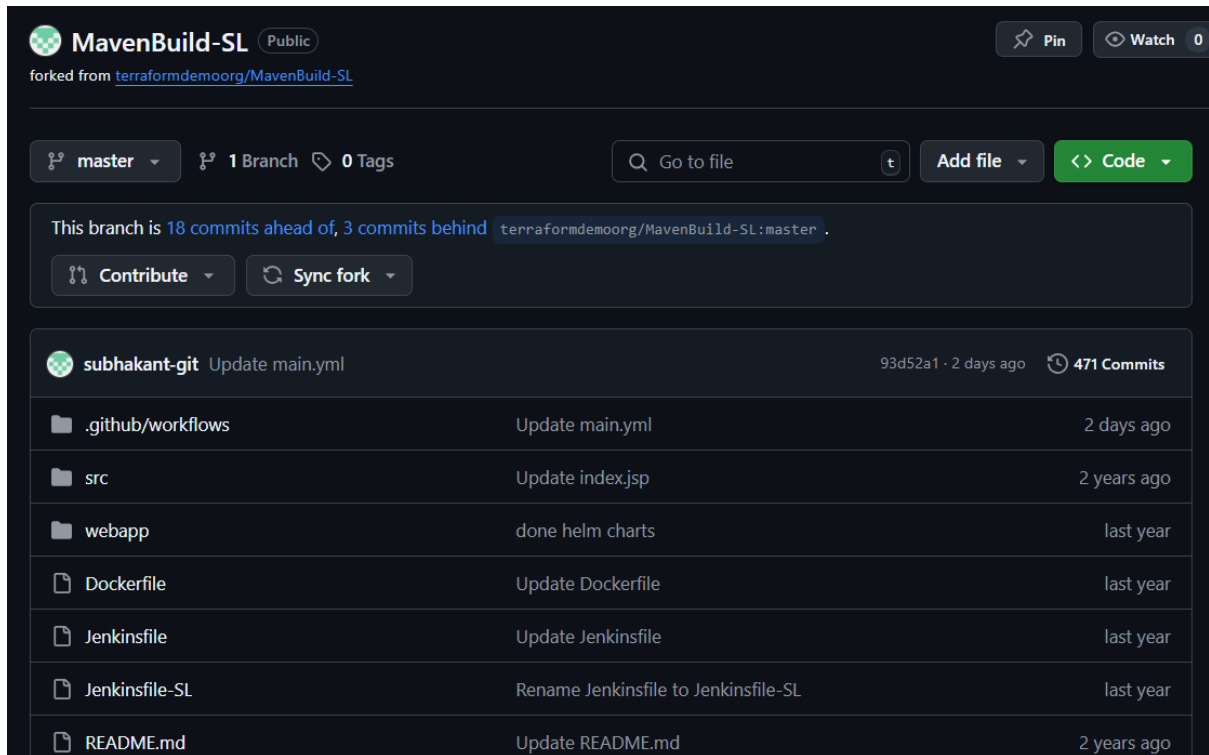
The following tasks outline the process of implementing CI/CD using GitHub Actions and Jenkins:

1. Create a code repository on GitHub
2. Create a GitHub Actions pipeline to perform continuous integration
3. Configure Tomcat Apache for automated code deployment
4. Integrate the GitHub Actions pipeline to invoke the Jenkins pipeline
5. Invoke pipeline to validate automated deployment

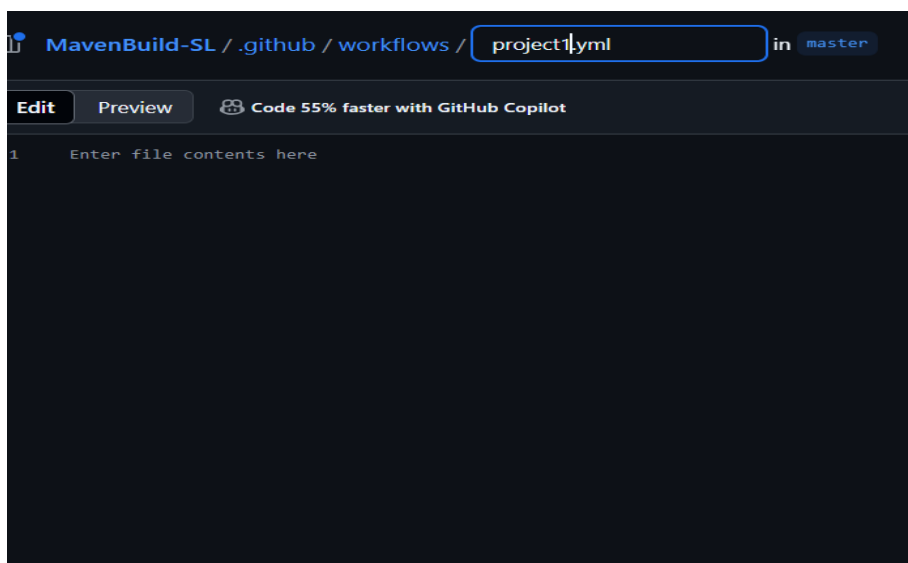
Solution

Task 1: Create a code repository on GitHub Forking repository from

<https://github.com/demo1orgtoday/MavenBuild-SL.git>



Task 2: Create a GitHub Actions workflow to perform continuous integration




Subhakanta Mishra
Subhamishra.in@gmail.com


Initial GitHub Action Code:

```
Edit Preview Code 55% faster with GitHub Copilot


1  name: Jenkins-EndProject1-CI
2  on:
3    #push: # run when there is commit to repo
4    workflow_dispatch: # run manually
5  jobs:
6    CICDjob:
7      runs-on: ubuntu-latest
8      steps:
9        - name: Clone the repo on ubuntu server
10         uses: actions/checkout@v4
11        - name: Install Java and maven on ubuntu server
12         uses: actions/setup-java@v4
13         with:
14           distribution: 'temurin'
15           java-version: '11'
16           cache: 'maven'
17        - name: Build the code
18         run: mvn package
```

← Jenkins-EndProject1-CI


 **Jenkins-EndProject1-CI #1**


 Summary

Jobs


 **CICDjob**


Run details


 Usage


 Workflow file


CICDjob
succeeded now in 17s


>  Set up job


>  Clone the repo on ubuntu server

>  Install Java and maven on ubuntu server

>  Build the code

>  Post Install Java and maven on ubuntu server

>  Post Clone the repo on ubuntu server

>  Complete job


Subhakanta Mishra
Subhamishra.in@gmail.com

Task 3: Configure Tomcat Apache for automated code deployment

```
root@ip-172-31-43-138:~# apt install tomcat9 tomcat9-admin -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libapr1 libeclipse-jdt-core-java libtcnative-1 libtomcat9-java tomcat9-common
Suggested packages:
  tomcat9-docs tomcat9-examples tomcat9-user
The following NEW packages will be installed:
  libapr1 libeclipse-jdt-core-java libtcnative-1 libtomcat9-java tomcat9 tomcat9-admin tomcat9-common
0 upgraded, 7 newly installed, 0 to remove and 261 not upgraded.
Need to get 12.7 MB of archives.
After this operation, 16.5 MB of additional disk space will be used.
Get:1 http://ap-south-1a.clouds.ports.ubuntu.com/ubuntu-ports jammy-updates/main arm64 libapr1 arm64
Get:2 http://ap-south-1a.clouds.ports.ubuntu.com/ubuntu-ports jammy/universe arm64 libeclipse-jdt-core-
B]
Get:3 http://ap-south-1a.clouds.ports.ubuntu.com/ubuntu-ports jammy-updates/universe arm64 libtomcat9
Get:4 http://ap-south-1a.clouds.ports.ubuntu.com/ubuntu-ports jammy-updates/universe arm64 tomcat9-co
Get:5 http://ap-south-1a.clouds.ports.ubuntu.com/ubuntu-ports jammy-updates/universe arm64 tomcat9 al
Get:6 http://ap-south-1a.clouds.ports.ubuntu.com/ubuntu-ports jammy-updates/universe arm64 tomcat9-ad
Get:7 http://ap-south-1a.clouds.ports.ubuntu.com/ubuntu-ports jammy/universe arm64 libtcnative-1 arm6
Fetched 12.7 MB in 2s (5651 kB/s)
Selecting previously unselected package libapr1:arm64.
(Reading database ... 331510 files and directories, 0 newly installed.)
Preparing to unpack .../0-libapr1_1.7.0-8ubuntu0.22.04.2_arm64.deb ...
```

← → ↺

🔒 📄 http://localhost:9090/manager/html



Tomcat Web Application Manager

Message:

OK

Manager

[List Applications](#)[HTML Manager Help](#)

Applications				
Path	Version	Display Name	Running	Sessions
/	None specified		true	0
/host-manager	None specified	Tomcat Host Manager Application	true	0

Subhakanta Mishra
Subhamishra.in@gmail.com

Added Deploy to container Jenkins plugin

Plugins

Updates

Available plugins

Installed plugins

Advanced settings

Q

deploy

Install	Name ↓
<input checked="" type="checkbox"/>	<div>Deploy to container 1.16</div> <div>Artifact Uploaders</div> <div>This plugin allows you to deploy a war to a container after a successful build. Glassfish 3.x remote deployment</div>
<input type="checkbox"/>	<div>Docker Pipeline 580.vc0c340686b_54</div> <div>pipeline DevOps Deployment docker</div> <div>Build and use Docker containers from pipelines.</div>

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

SSH server

☒ Success

Deploy to container

☒ Success

Loading plugin extensions

☒ Success

→ [Go back to the top page](#)

(you can start using the installed plugins right away)

→ ☐ Restart Jenkins when installation is complete and no jobs are running

Added tomcat user

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

tomcat

☐ Treat username as secret ?

Password ?

●●●●●●●●

ID ?

tomcat-user

Description ?

tomcat-user

Subhakanta Mishra
Subhamishra.in@gmail.com

Task 4: Integrate the GitHub Actions pipeline to invoke the Jenkins pipeline

Added secrets for lab connection

Repository secrets			New repository secret
Name ↕	Last updated		
HOST	3 minutes ago		
JENKINS_TOKEN	2 days ago		
PASSWORD	2 minutes ago		
PORT	now		
SNYK_TOKEN	last week		
USERNAME	1 minute ago		

Updated action with .war copy and trigger for Jenkins job

```
name: Jenkins-EndProject1-CICD
on:
  push: # run when there is commit to repo
  workflow_dispatch: # run manually
jobs:
  CICDjob:
    runs-on: ubuntu-latest
    steps:
      - name: Clone the repo on ubuntu server
        uses: actions/checkout@v4
      - name: Install Java and maven on ubuntu server
        uses: actions/setup-java@v4
        with:
          distribution: 'temurin'
          java-version: '11'
          cache: 'maven'
      - name: Build the code
        run: mvn package
      - name: connect to Lab and deploy code on tomcat9
        uses: cross-the-world/ssh-scp-ssh-pipelines@latest
        with:
          host: ${secrets.HOST}
          user: ${secrets.USERNAME}
          pass: ${secrets.PASSWORD}
          port: ${secrets.PORT}
          connect_timeout: 10s
          first_ssh: |
            sudo chmod 777 /var/lib/tomcat9/webapps
          scp: |
            './target/*.war' => /var/lib/tomcat9/webapps
          last_ssh: |
            sudo systemctl restart tomcat9
      - name: Connect to Lab machine and trigger a Jenkins job
        uses: cross-the-world/ssh-scp-ssh-pipelines@latest
        with:
          host: ${secrets.HOST}
          user: ${secrets.USERNAME}
          pass: ${secrets.PASSWORD}
          port: ${secrets.PORT}
          connect_timeout: 10s
          first_ssh: |
            curl -l -u admin:Root123$ http://localhost:8080/job/JenkinsEndProject2/build?token=tomcatDeploy
```

Subhakanta Mishra
Subhamishra.in@gmail.com
Added maven tool in Jenkins

Maven Installations

Maven installations ^

 Edited

Add Maven

Maven

Name

mymaven

☒ Install automatically ?

Install from Apache

Version

3.9.9

Creating Jenkins pipeline job

JenkinsEndProject1

» Required field



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



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.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

branch Pipeline

Build Triggers

- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☐ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?
- ☐ Quiet period ?
- ☒ Trigger builds remotely (e.g., from scripts) ?

Authentication Token

tomcatDeploy|

Use the following URL to trigger build remotely: JENKINS_URL/job/jenkinsEndProject1/build?token=TOKEN
Optionally append &cause=Cause+Text to provide text that will be included in the recorded build cause.

```
pipeline{
  agent any
  tools{
    maven 'mymaven'
  }
  stages{
    stage('checkout code')
    {
      steps{
        git
      }
    }
    stage('Build and Deploy the Code')
    {
      steps{
        sh 'mvn package'
      }
    }
    post{
      success{
        deploy adapters: [tomcat9(credentialsId: 'tomcat-user', path: '', url:
        'http://localhost:9090')], contextPath: null, war: '**/*.war'
      }
    }
  }
}
```

Task 5: Invoke pipeline to validate automated deployment

```
1 Post Clone the repo on ubuntu server 0s

1 Post job cleanup.
2 /usr/bin/git version
3 git version 2.49.0
4 Temporarily overriding HOME='/home/runner/work/_temp/af9fbbb7-72bd-465b-b73d-802732de4508' before making global git config changes
5 Adding repository directory to the temporary git global config as a safe directory
6 /usr/bin/git config --global --add safe.directory /home/runner/work/MavenBuild-SL/MavenBuild-SL
7 /usr/bin/git config --local --name-only --get-regexp core.sshCommand
8 /usr/bin/git submodule foreach --recursive sh -c "git config --local --name-only --get-regexp 'core.sshCommand' && git config --local --unset-all 'core.sshCommand' || :"
9 /usr/bin/git config --local --name-only --get-regexp http.https://github.com/.extraheader
10 http.https://github.com/.extraheader
11 /usr/bin/git config --local --unset-all http.https://github.com/.extraheader
12 /usr/bin/git submodule foreach --recursive sh -c "git config --local --name-only --get-regexp 'http.https://github.com/.extraheader' && git config --local --unset-all 'http.https://github.com/.extraheader' || :"

1 Complete job 0s

1 Cleaning up orphan processes
```


Dashboard > JenkinsEndProject1 >

Status

</> Changes

▶ Build Now

⚙️ Configure

🗑️ Delete Pipeline

📁 Stages

✎ Rename

❓ Pipeline Syntax

🟢 JenkinsEndProject1

Permalinks

- [Last build \(#2\), 1 min 3 sec ago](#)
- [Last stable build \(#2\), 1 min 3 sec ago](#)
- [Last successful build \(#2\), 1 min 3 sec ago](#)
- [Last failed build \(#1\), 10 min ago](#)
- [Last unsuccessful build \(#1\), 10 min ago](#)
- [Last completed build \(#2\), 1 min 3 sec ago](#)

JenkinsEndProject1 [Jenk: x]Addressbook x +

← → ↺

🔒 http://localhost:9090/addressbook/8

Filter contacts...

First Name	Last Name	Email
George	White	george@white.com
Daniel	Thompson	daniel@thompson.com
Timothy	Jones	timothy@jones.com
Peter	Wilson	peter@wilson.com
Dan	Robinson	dan@robinson.com
Dan	Davis	dan@davis.com
Olivia	Davis	olivia@davis.com
Dan	Smith	dan@smith.com
Daniel	Anderson	daniel@anderson.com
Alice	Thomas	alice@thomas.com