

# DevOps Foundations: Version Control and CI/CD with Jenkins



## Course-End Project





## **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



# Problem Statement and Motivation



## 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.

# Industry Relevance

The following tools used in this project serve specific purposes within the industry:

1. **Jenkins:** It is an open-source automation server commonly used in the software development industry for building, testing, and deploying projects. It supports continuous integration and continuous delivery (CI/CD) practices, enabling teams to increase efficiency, detect issues early, and accelerate software releases.
2. **GitHub Actions:** It is a CI/CD platform that streamlines the code integration and deployment processes by automating the software development process directly within GitHub. This makes it invaluable for developers to manage and deploy code updates efficiently.
3. **Tomcat Apache:** It is an open-source web server and servlet container that supports Java servlets and JSPs, making it a staple in web application development and deployment for organizations relying on Java technologies.



# 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





# Project References

- **Task 1:** Lesson 02
- **Task 2:** Lesson 10
- **Task 3:** Lesson 07
- **Task 4:** Lesson 10
- **Task 5:** Lessons 08 and 10





# Output Screenshots

Maven workflow file configured in GitHub Actions:

The screenshot shows the GitHub interface for a repository named 'Maven-Build'. The top navigation bar includes links for Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main content area displays the file structure of the repository, specifically the '.github/workflows' directory. A file named 'maven.yml' is highlighted with a red box, indicating it is the selected file. The commit history for this file is shown, with the most recent commit by 'Shree697' titled 'Create maven.yml'.

Repository: **Maven-Build**

Branch: **master**

File path: **.github/workflows/**

Commit history:

Name	Last commit message	Last commit
..		
maven.yml	Create maven.yml	Shree697

# Output Screenshots

Tomcat Apache setup:

```
root@ip-172-31-31-214: ~  
File Edit View Search Terminal Help  
ravitulsianisim@ip-172-31-31-214:~$ sudo su -  
root@ip-172-31-31-214:~# apt -y update  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease  
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [11  
9 kB]  
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]  
Hit:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Hit:5 https://download.docker.com/linux/ubuntu jammy InRelease  
Ign:6 https://pkg.jenkins.io/debian-stable binary/ InRelease  
Hit:7 https://pkg.jenkins.io/debian-stable binary/ Release  
Hit:8 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stabl  
e:/v1.28/deb InRelease  
Fetched 229 kB in 1s (340 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
60 packages can be upgraded. Run 'apt list --upgradable' to see them.  
root@ip-172-31-31-214:~# apt -y install tomcat9 tomcat9-admin  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
tomcat9 is already the newest version (9.0.58-1ubuntu0.1).  
tomcat9-admin is already the newest version (9.0.58-1ubuntu0.1).  
0 upgraded, 0 newly installed, 0 to remove and 60 not upgraded.
```

# Output Screenshots

SSH pipeline configuration in **maven.yml** workflow file:

The screenshot displays the GitHub Actions workflow editor for the file `maven.yml` in the `master` branch. The interface includes a top bar with the file path, a "Commit changes..." button (highlighted with a red box), and a sidebar with a "Marketplace" section.

The workflow file content is as follows:

```
33 # Optional: Uploads the full dependency graph to GitHub to improve the quality of Dependabot al
34 - name: Update dependency graph
35   uses: advanced-security/maven-dependency-submission-action@571e99aab1055c2e71a1e2309b9691de18
36
37   - name: scp ssh pipelines
38     uses: cross-the-world/ssh-scp-ssh-pipelines@latest
39     with:
40       host: ${ secrets.HOST }
41       user: ${ secrets.USERNAME }
42       pass: ${ secrets.PASSWORD }
43       port: ${ secrets.PORT }
44       scp: |
45         ./target/*.war => /tmp/
46       last_ssh: |
47         ls -lart /tmp
48
```

The sidebar on the right shows the "Marketplace" section with a search bar and featured actions:

- Cache** (4.3k stars): By actions. Cache artifacts like dependencies and build outputs to improve workflow execution time.
- Setup Node.js environment** (3.6k stars): By actions. Setup a Node.js environment by adding

# Output Screenshots

GitHub repository secrets:

Security

🔍 Code security and analysis

🔑 Deploy keys

🔒 Secrets and variables

⬆

📁 Actions

📁 Codespaces

📁 Dependabot














Integrations

🗨️ GitHub Apps

✉️ Email notifications

Repository secrets

New repository secret

Name 	Last updated	
 HOST	4 minutes ago	 
 PASSWORD	1 minute ago	 
 PORT	now	 
 USERNAME	2 minutes ago	 


# Output Screenshots


Freestyle job creation in Jenkins for continuous deployment:


**Enter an item name**

Test\_GitHubActions\_job

» Required field

**Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

**Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

**Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known

OK

# Output Screenshots

Workflow configuration to integrate Jenkins and GitHub Actions:

The screenshot displays the GitHub Actions workflow editor interface. On the left, the 'Files' sidebar shows the repository structure, including the '.github/workflows' directory and the 'maven.yml' file being edited. The main editor area shows the workflow configuration for 'Maven-Build' in the 'master' branch. The configuration includes a 'host' field, a 'user' field, a 'pass' field, a 'port' field, and a 'scp' block. The 'scp' block contains a command to copy the target directory to a temporary location. Below this, a 'name' field is set to 'Invoke Jenkins Deployment Job', and the 'uses' field is set to 'appleboy/jenkins-action@master'. The 'with' block contains a 'url' field, a 'user' field, a 'token' field, and a 'job' field. The 'Commit changes...' button is highlighted with a red box. The right sidebar shows the 'Marketplace' section with a search bar and featured actions like 'Cache' and 'Setup Node.js environment'.

```
40   host: ${ secrets.HOST }}
41   user: ${ secrets.USERNAME }}
42   pass: ${ secrets.PASSWORD }}
43   port: ${ secrets.PORT }}
44   scp: |
45     ./target/*.war => /tmp/
46   last_ssh: |
47     ls -lart /tmp
48
49   - name: Invoke Jenkins Deployment Job
50     uses: appleboy/jenkins-action@master
51     with:
52       url: "http://104.198.139.137:8080"
53       user: "admin"
54       token: ${ secrets.JENKINS_TOKEN }}
55       job: "TomcatDeployment"
```

Use **Control + Shift + m** to toggle the **tab** key moving focus. Alternatively, use **esc** then **tab** to move to the next interactive element on the page.

# Output Screenshots

Jenkins job being invoked from the GitHub Actions pipeline for continuous deployment:

```
✓ Invoke Jenkins Deployment Job 0s

1 ▼ Run appleboy/jenkins-action@master
2   with:
3     url: http://***:8080
4     user: admin
5     token: ***
6     job: TomcatDeployment
7   env:
8     JAVA_HOME: /opt/hostedtoolcache/Java_Temurin-Hotspot_jdk/17.0.10-7/x64
9     JAVA_HOME_17_X64: /opt/hostedtoolcache/Java_Temurin-Hotspot_jdk/17.0.10-7/x64
10  /usr/bin/docker run --name b519c0c1d7c619f1114fc480944d876b9e7937_1b9369 --label b519c0 --workdir /github/workspace --rm -e "JAVA_HOME" -e "JAVA_HOME_17_X64"
    -e "INPUT_URL" -e "INPUT_USER" -e "INPUT_TOKEN" -e "INPUT_JOB" -e "HOME" -e "GITHUB_JOB" -e "GITHUB_REF" -e "GITHUB_SHA" -e "GITHUB_REPOSITORY" -e
    "GITHUB_REPOSITORY_OWNER" -e "GITHUB_REPOSITORY_OWNER_ID" -e "GITHUB_RUN_ID" -e "GITHUB_RUN_NUMBER" -e "GITHUB_RETENTION_DAYS" -e "GITHUB_RUN_ATTEMPT" -e
    "GITHUB_REPOSITORY_ID" -e "GITHUB_ACTOR_ID" -e "GITHUB_ACTOR" -e "GITHUB_TRIGGERING_ACTOR" -e "GITHUB_WORKFLOW" -e "GITHUB_HEAD_REF" -e "GITHUB_BASE_REF" -e
    "GITHUB_EVENT_NAME" -e "GITHUB_SERVER_URL" -e "GITHUB_API_URL" -e "GITHUB_GRAPHQL_URL" -e "GITHUB_REF_NAME" -e "GITHUB_REF_PROTECTED" -e "GITHUB_REF_TYPE" -e
    "GITHUB_WORKFLOW_REF" -e "GITHUB_WORKFLOW_SHA" -e "GITHUB_WORKSPACE" -e "GITHUB_ACTION" -e "GITHUB_EVENT_PATH" -e "GITHUB_ACTION_REPOSITORY" -e
    "GITHUB_ACTION_REF" -e "GITHUB_PATH" -e "GITHUB_ENV" -e "GITHUB_STEP_SUMMARY" -e "GITHUB_STATE"
```



# Output Screenshots

Deployment of the Java application on Tomcat by invoking Jenkins job from a GitHub Actions pipeline:

Dashboard > TomcatDeployment >

Status

Changes

Workspace

Build Now

Configure

Delete Project

Rename

Build History

trend ▾

Filter...

#1

May 7, 2024, 2:44 AM

Atom feed for all Atom feed for failures

✓ TomcatDeployment

Permalinks

- Last build (#1), 13 sec ago
- Last stable build (#1), 13 sec ago
- Last successful build (#1), 13 sec ago
- Last completed build (#1), 13 sec ago



**Thank you**