Experiment 05

Aim:

To deploy an application on an Apache Tomcat server using Jenkins.

Theory:

Jenkins is an open-source automation server that is widely used for **Continuous Integration (CI)** and **Continuous Deployment (CD)**. It helps automate different stages of the software development lifecycle such as pulling code from version control, building it, testing, and deploying it automatically.

In this experiment:

- **Git** → Used as the version control system where the application source code is maintained.
- **Jenkins** → Automates pulling the code from Git, building it (using Maven), and triggering deployment.
- **Maven** → Builds the project and generates a .war file.
- **Apache Tomcat** → Acts as the application server where the WAR file is deployed.

Workflow:

1. Jenkins pulls the latest source code from Git.
2. Jenkins uses Maven to build and package the application.
3. The generated .war file is automatically deployed on the Tomcat server.

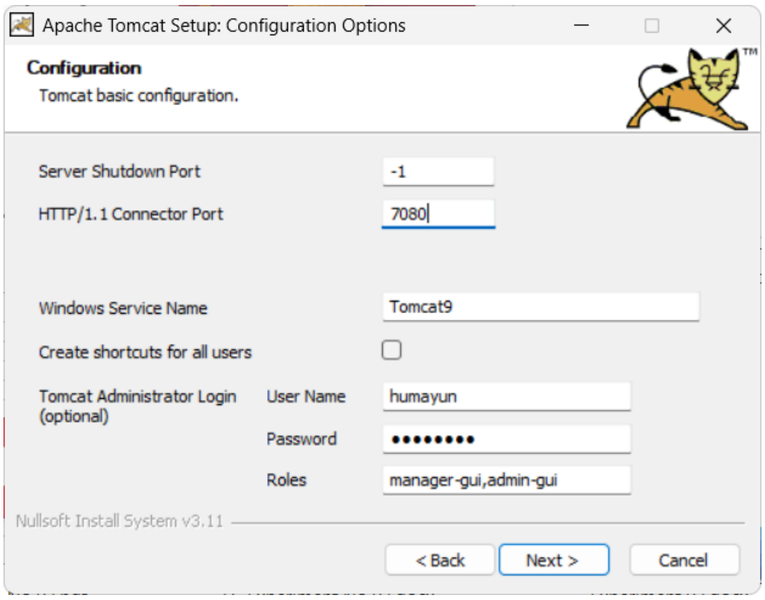
Advantages of this workflow:

- Automated builds and deployments (less manual work).
- Faster development-to-deployment cycle.
- Fewer human errors in deployment.
- Ensures a smooth CI/CD pipeline.

Procedure / Steps:

1. **Install Apache Tomcat**
    - Download from: [Apache Tomcat]

- Install → Next → Agree → Set Port = 7080 → Set Username & Password → Finish.
- Verify installation at: http://localhost:7080/

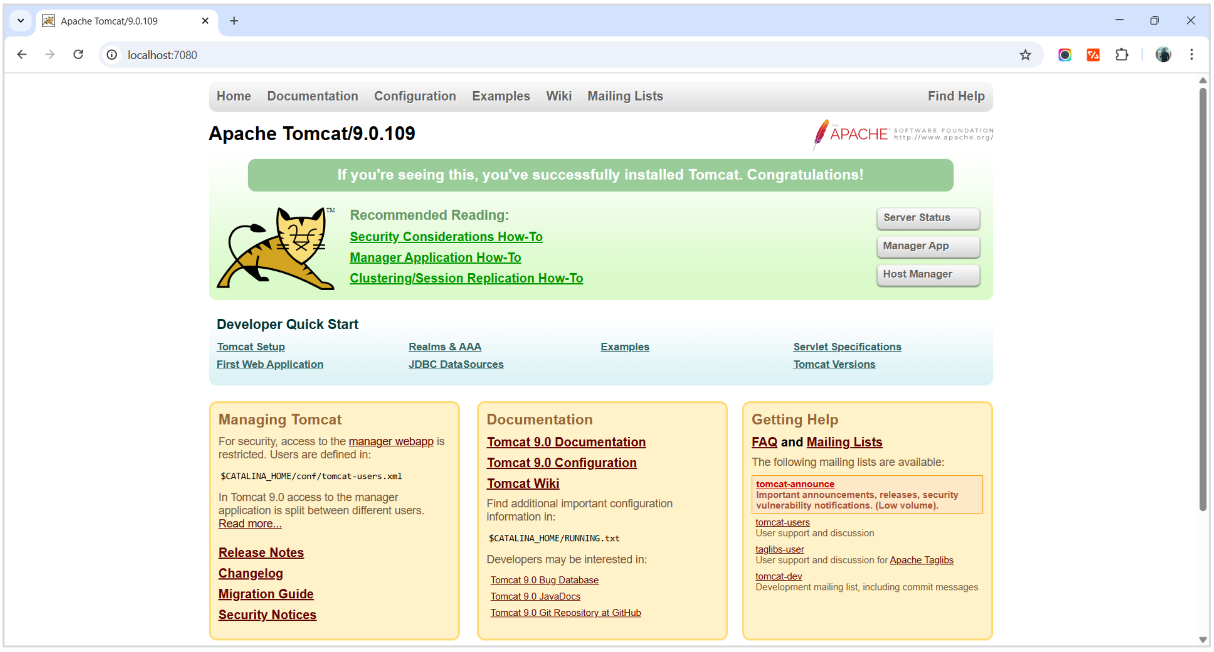

Apache Tomcat Setup: Configuration Options

Configuration
Tomcat basic configuration.

Server Shutdown Port		-1
HTTP/1.1 Connector Port		7080
Windows Service Name		Tomcat9
Create shortcuts for all users		☐
Tomcat Administrator Login (optional)	User Name	humayun
	Password	••••••••
	Roles	manager-gui,admin-gui

Nullsoft Install System v3.11

Check if tomcat is running: https://localhost:7080/



Apache Tomcat/9.0.109

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations How-To
Manager Application How-To
Clustering/Session Replication How-To

Developer Quick Start
Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat
For security, access to the manager webapp is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 9.0 access to the manager application is split between different users. Read more...

Release Notes
Changelog
Migration Guide
Security Notices

Documentation
Tomcat 9.0 Documentation
Tomcat 9.0 Configuration
Tomcat Wiki
Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:
Tomcat 9.0 Bug Database
Tomcat 9.0 JavaDocs
Tomcat 9.0 Git Repository at GitHub

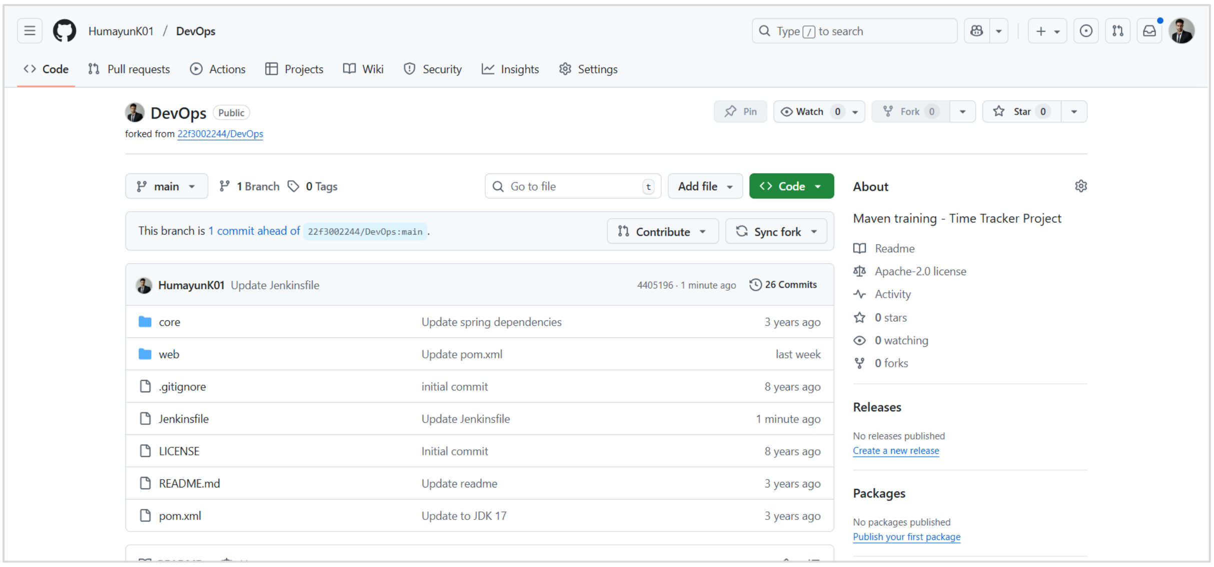
Getting Help
FAQ and Mailing Lists
The following mailing lists are available:
tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).
tomcat-users
User support and discussion
taglibs-user
User support and discussion for Apache Taglibs
tomcat-dev
Development mailing list, including commit messages

2. Install Jenkins

- Download from: Jenkins
- Run Jenkins and access it at: http://localhost:8080/

3. Create or Clone a Maven Project on GitHub

- Example repo: https://github.com/HumayunK01/DevOps


HumayunK01 / DevOps

DevOps Public
forked from 22f3002244/DevOps

main · 1 Branch · 0 Tags

This branch is 1 commit ahead of 22f3002244/DevOps:main.

HumayunK01 Update Jenkinsfile · 4405196 · 1 minute ago · 26 Commits

core	Update spring dependencies	3 years ago
web	Update pom.xml	last week
.gitignore	initial commit	8 years ago
Jenkinsfile	Update Jenkinsfile	1 minute ago
LICENSE	Initial commit	8 years ago
README.md	Update readme	3 years ago
pom.xml	Update to JDK 17	3 years ago

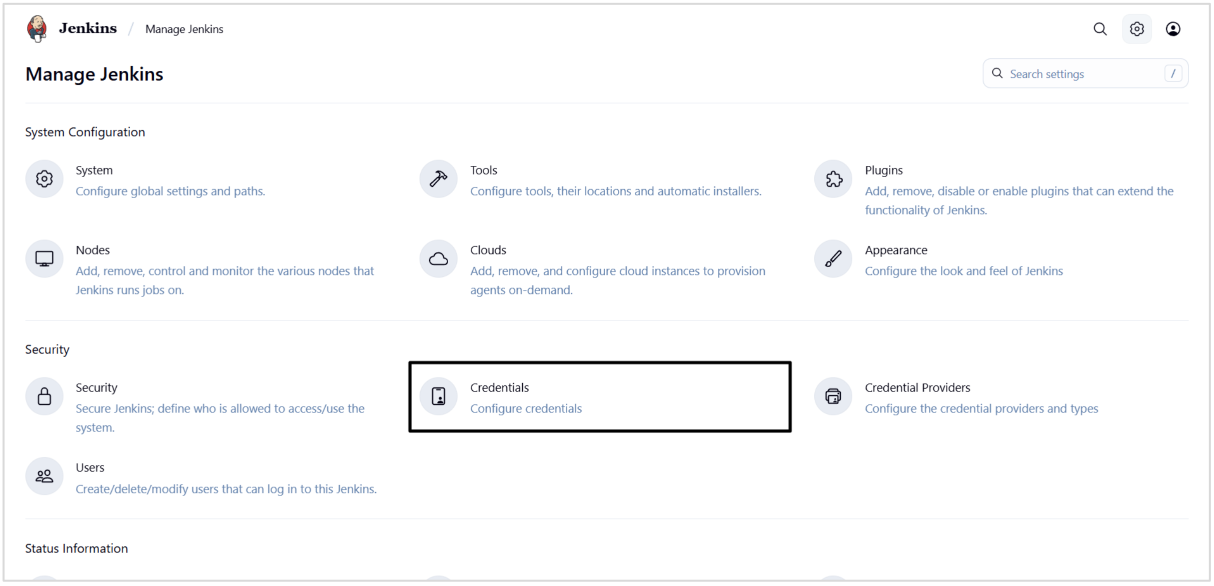
About
Maven training - Time Tracker Project
Readme
Apache-2.0 license
Activity
0 stars
0 watching
0 forks

Releases
No releases published
Create a new release

Packages
No packages published
Publish your first package

4. Configure Credentials in Jenkins

- Go to **Manage Jenkins → Credentials → System → Global Credentials**


Jenkins / Manage Jenkins

Manage Jenkins

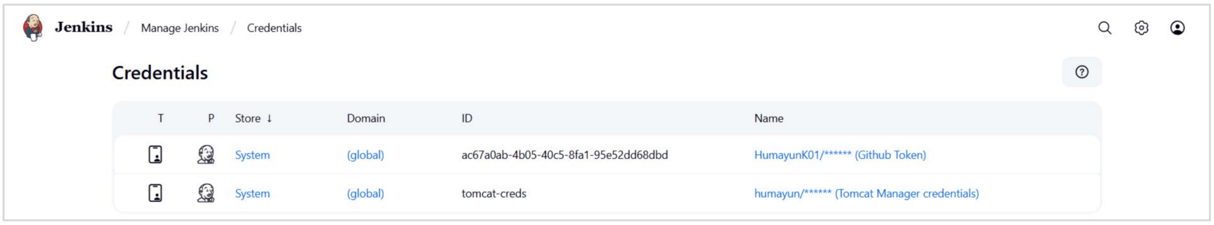
System Configuration

System — Configure global settings and paths.
Tools — Configure tools, their locations and automatic installers.
Plugins — Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
Nodes — Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
Clouds — Add, remove, and configure cloud instances to provision agents on-demand.
Appearance — Configure the look and feel of Jenkins

Security

Security — Secure Jenkins; define who is allowed to access/use the system.
Credentials — Configure credentials
Credential Providers — Configure the credential providers and types
Users — Create/delete/modify users that can log in to this Jenkins.

Status Information

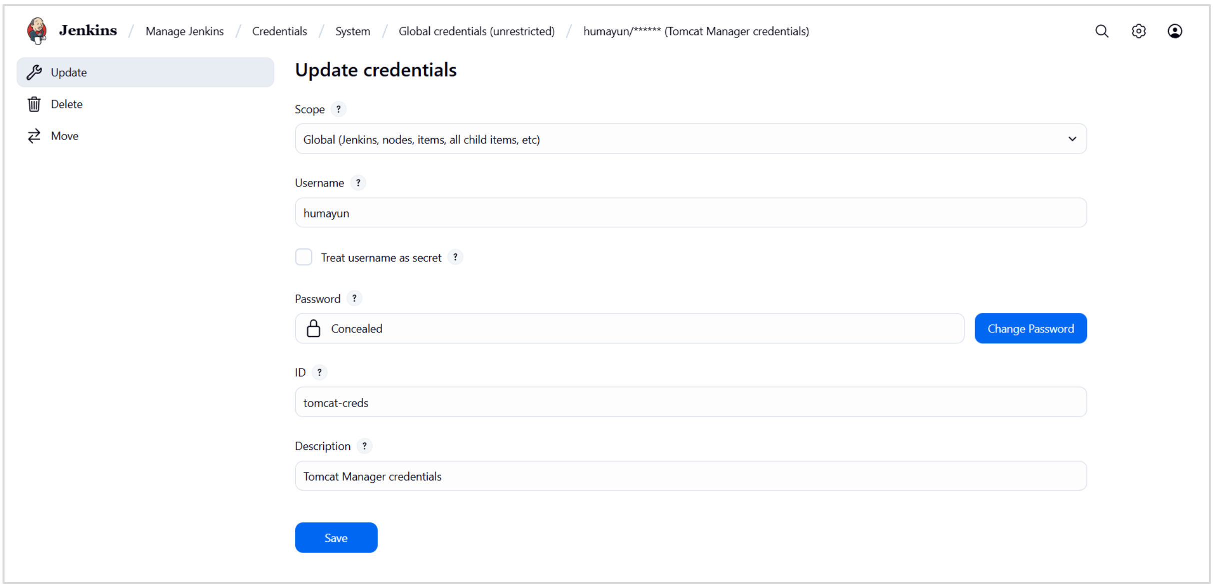

Jenkins / Manage Jenkins / Credentials

Credentials

T	P	Store ↓	Domain	ID	Name
		System	(global)	ac67a0ab-4b05-40c5-8fa1-95e52dd68dbd	HumayunK01/****** (Github Token)
		System	(global)	tomcat-creds	humayun/****** (Tomcat Manager credentials)

- Add credentials:
  - Username (humayun)
  - Password (********)
  - ID = tomcat-creds

- Description = Tomcat Manager credentials



Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestricted) / humayun/****** (Tomcat Manager credentials)

Update
Delete
Move

Update credentials

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

Username: humayun

Treat username as secret

Password: Concealed

Change Password

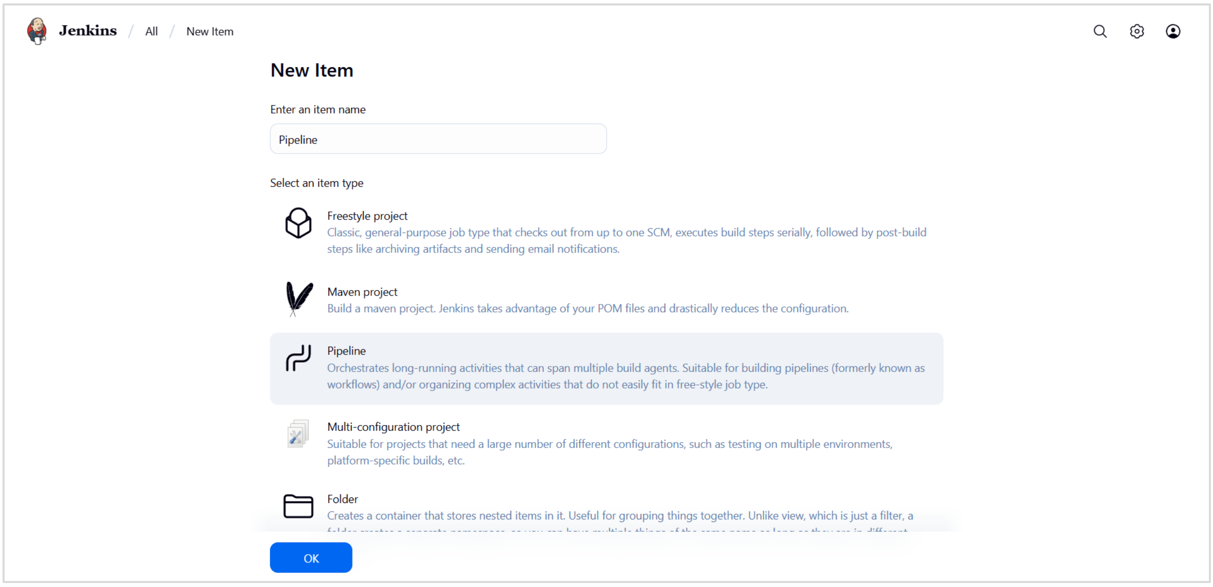
ID: tomcat-creds

Description: Tomcat Manager credentials

Save

5. Create a New Pipeline in Jenkins

- Go to **New Item → Pipeline**



Jenkins / All / New Item

New Item

Enter an item name: Pipeline

Select an item type

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.

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

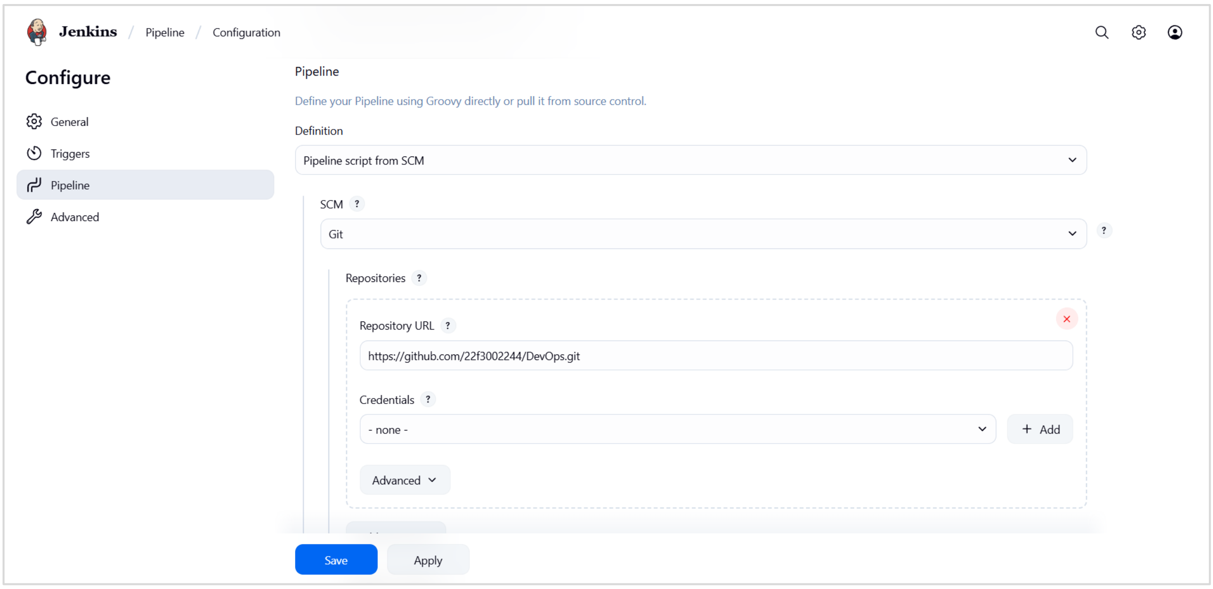
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

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

OK

- Under **Pipeline Configuration**:
  - Choose **Pipeline Script from SCM**
  - Select **Git** as SCM
  - Add repository link and credentials (if private)
  - Set branch as */main
  - Click **Apply & Save**



Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition: Pipeline script from SCM

SCM: Git

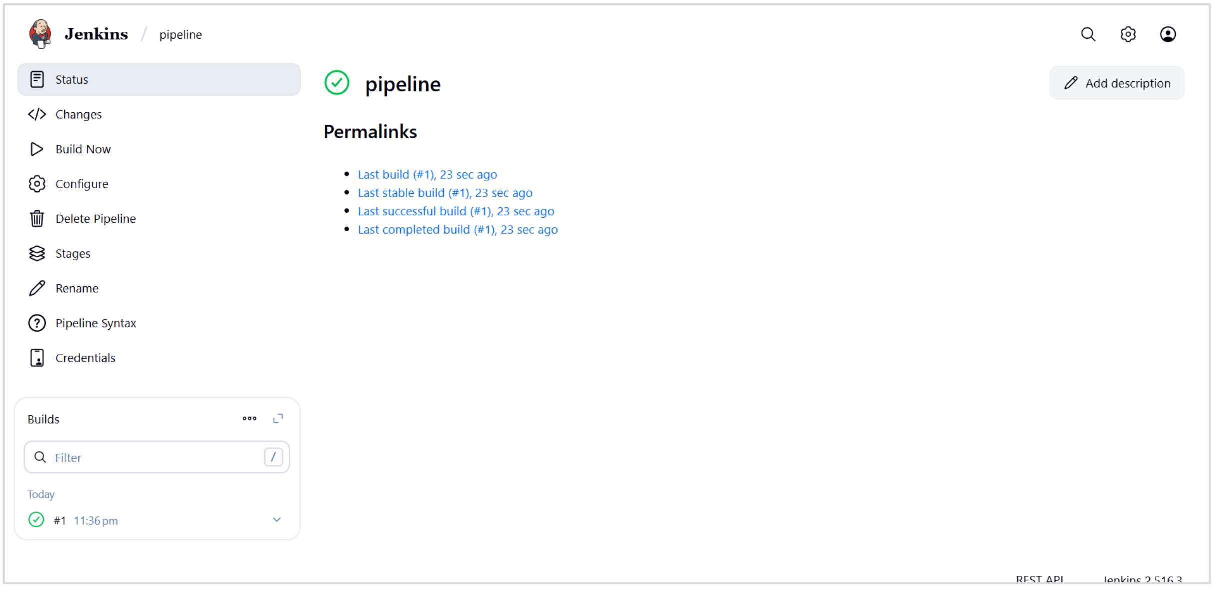
Repositories

Repository URL: https://github.com/22f3002244/DevOps.git

Credentials: - none -

6. Build the Project

- Click **Build Now**
- Verify that a .war file is generated inside:
- C:\ProgramData\Jenkins\.jenkins\workspace\pipeline\web\target\*.war



pipeline

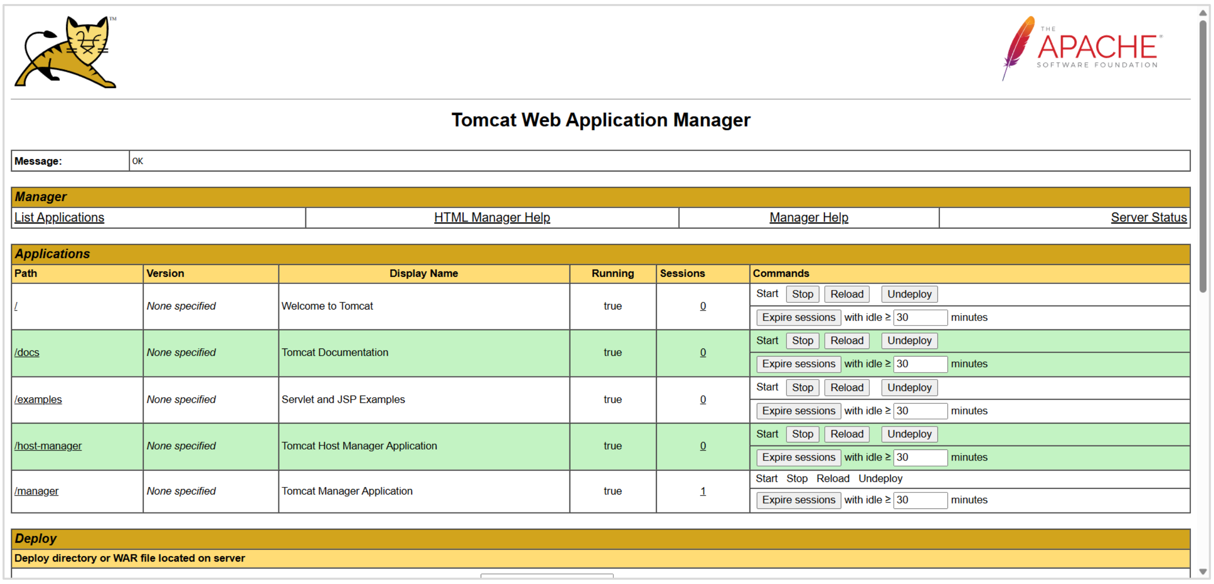
Permalinks

- Last build (#1), 23 sec ago
- Last stable build (#1), 23 sec ago
- Last successful build (#1), 23 sec ago
- Last completed build (#1), 23 sec ago

#1 11:36pm

7. Deploy the Application on Tomcat

- Check Tomcat Manager at: http://localhost:7080/manager/html



Tomcat Web Application Manager

Message:	OK

Manager

List Applications | HTML Manager Help | Manager Help | Server Status

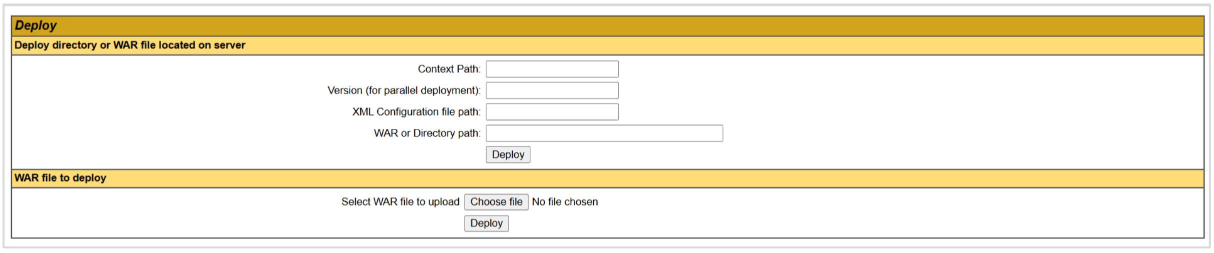
Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy / Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy / Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy / Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy / Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy / Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

- If the .war file is not listed, manually upload it in the Tomcat Manager → "WAR file to deploy."



Deploy

Deploy directory or WAR file located on server

Context Path:	
Version (for parallel deployment):	
XML Configuration file path:	
WAR or Directory path:	

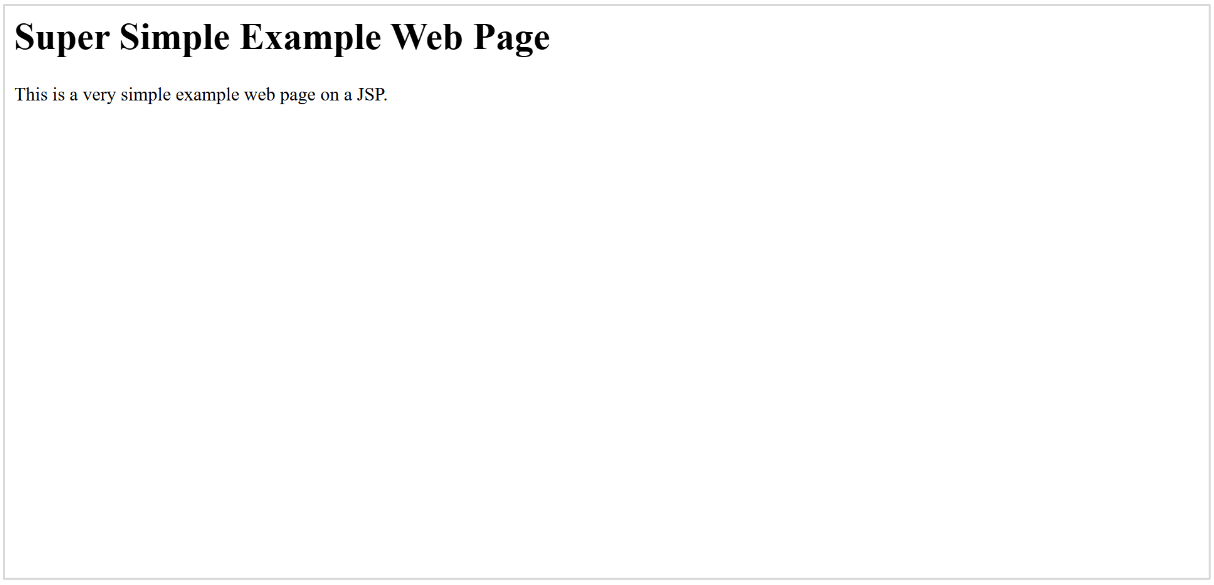
Deploy

WAR file to deploy

Select WAR file to upload Choose file No file chosen

Deploy

- Once deployed, access the app at:
- http://localhost:7080/roshambo.war



Super Simple Example Web Page

This is a very simple example web page on a JSP.

Output:

- Jenkins successfully pulled code from Git.
- Maven generated the .war file inside the target folder.
- The .war file was deployed on Apache Tomcat.
- The application was accessible through the browser at

  http://localhost:7080/roshambo.war

Conclusion:

In this experiment, we implemented a CI/CD pipeline using **Jenkins, Maven, Git, and Apache Tomcat**. The pipeline automated code checkout, build, testing, and deployment. While deployment was generally smooth, occasional issues like a 404 error may occur if the context path or deployment configuration is not set correctly. Overall, this experiment provided practical hands-on experience in DevOps automation, ensuring efficient and error-free deployments.