

Experiment 06 - Jenkins Pipeline

Roll No.	70
Name	Mayuri Shridatta Yerande
Class	D15-B
Subject	DevOps Lab
LO Mapped	<p>LO1: To understand the fundamentals of DevOps engineering and be fully proficient with DevOps terminologies, concepts, benefits, and deployment options to meet your business requirements</p> <p>LO3: To understand the importance of Jenkins to Build and deploy Software Applications on server environment</p>

Aim: To Build the pipeline of jobs using Maven / Gradle / Ant in Jenkins, create a pipeline script to build and deploy an application over the tomcat server.

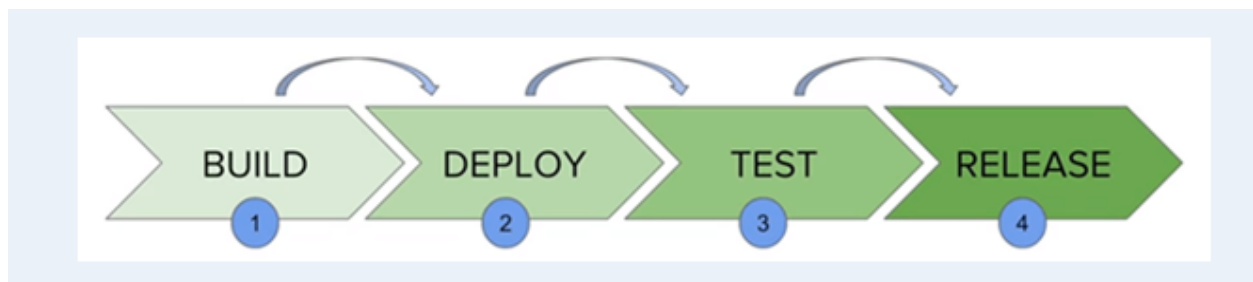
Introduction:

The Jenkins pipeline in CI/CD automates many tasks and makes the CI/CD pipeline tasks reliable, efficient, repeatable, and high in quality.

The connection of jobs in a particular format in the pipeline is done by Jenkins.

It represents the continuous delivery and continuous integration of the jobs in the SDLC and DevOps life cycle management in the pipeline.

The below diagram shows the Jenkins pipeline. This includes continuous integration and continuous delivery of jobs namely the Build, Deploy, Test, and Release. It is done as these jobs are interdependent and forms the pipeline in a particular format. This involves continuous automation from one stage to another to bring down the cost, time, and number of iterations without affecting the quality.



It is also about the version control as the software applications which are being developed goes through the complex processes from the build to release.

The software is developed in a scalable, reliable, and repeatable manner.

Advantages:-

The Jenkins pipeline in CI/CD automates many tasks and makes the CI/CD pipeline tasks reliable, efficient, repeatable, and high in quality.

It has made the process code easier for iterative development with other features such as audit trails, code review, and access control and has sound approval and promotion process by many project members. Multiple jobs from a single project can be managed easily.

Other Advantages are as follows,

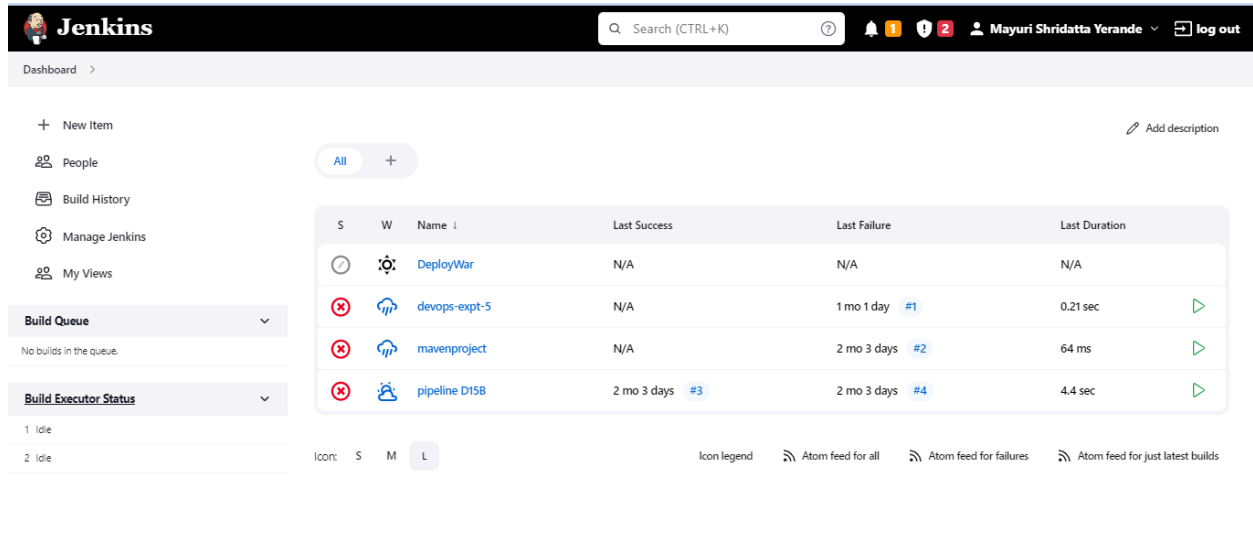
- Review of code on the pipeline is possible
- Make many pipelines automatically for all the branches with a single Jenkins file.
- The Jenkins pipeline can be reviewed
- Single source for the pipeline and be customized for multiple users.
- We can use the Web User interface or Jenkins file directly.

Build Tool:

- The Build Now Tool is used to create the pipeline.
- The following example shows a successful build of a pipeline created with a one-line script that uses the “echo” step to output the phrase, “hello from pipeline:”
- Started by user anonymous [Pipeline] echo
- hello from Pipeline
- [Pipeline] End of Pipeline
- Finished: SUCCESS

Pipeline Jobs:

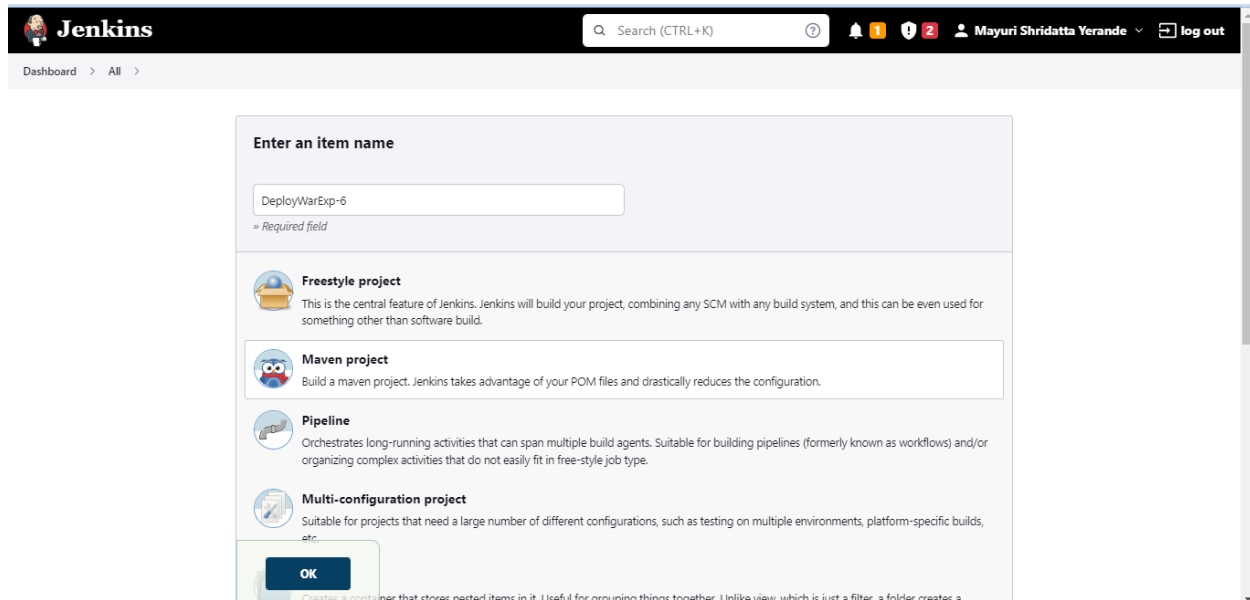
- Log into Jenkins and select ‘New item’ from the dashboard



The screenshot shows the Jenkins Dashboard. The top navigation bar includes the Jenkins logo, a search bar, and user information (Mayuri Shridatta Yerande). The left sidebar contains links for 'New item', 'People', 'Build History', 'Manage Jenkins', and 'My Views'. The main content area displays a table of pipeline jobs. The table has columns for 'S' (Status), 'W' (Icon), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. The jobs listed are 'DeployWar', 'devops-expt-5', 'mavenproject', and 'pipeline D15B'. The 'devops-expt-5' job is highlighted with a green play button icon, indicating it is the current selection.

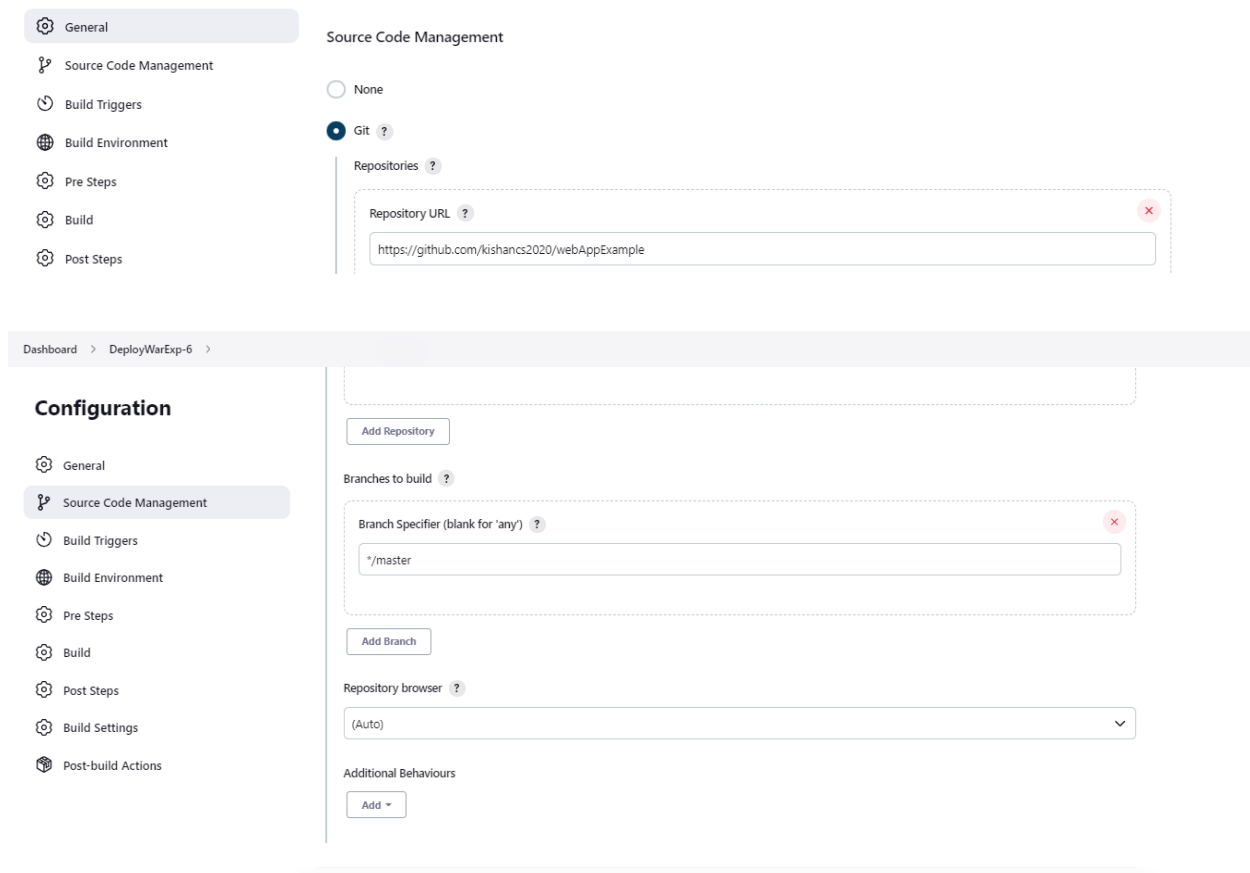
S	W	Name	Last Success	Last Failure	Last Duration
		DeployWar	N/A	N/A	N/A
		devops-expt-5	N/A	1 mo 1 day #1	0.21 sec
		mavenproject	N/A	2 mo 3 days #2	64 ms
		pipeline D15B	2 mo 3 days #3	2 mo 3 days #4	4.4 sec

- Next, enter a name for your pipeline and select ‘maven’ project. Click on ‘ok’ to proceed proceed



The image shows the Jenkins 'New Item' configuration page. At the top, there's a search bar and user information. The main section is titled 'Enter an item name' and contains a text input field with 'DeployWarExp-6'. Below this, there are four project type options: 'Freestyle project', 'Maven project', 'Pipeline', and 'Multi-configuration project'. Each option has a brief description. At the bottom, there is an 'OK' button.

- Next if the source code is from a github repository paste the URL of the repository in the source code management section after selecting git
Github repository that I used: <https://github.com/kishancs2020/webAppExample>



The image shows the Jenkins 'Source Code Management' configuration page. On the left, there's a sidebar with navigation links: 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Pre Steps', 'Build', and 'Post Steps'. The main section is titled 'Source Code Management' and contains a radio button for 'None' and a selected radio button for 'Git'. Below 'Git', there's a 'Repositories' section with a 'Repository URL' input field containing 'https://github.com/kishancs2020/webAppExample'. At the bottom, there's a 'Branches to build' section with a 'Branch Specifier (blank for 'any')' input field containing '*/master'. There are also buttons for 'Add Repository', 'Add Branch', and 'Repository browser'.

Dashboard > DeployWarExp-6 >

Configuration

- General
- Source Code Management
- Build Triggers**
- Build Environment
- Pre Steps
- Build
- Post Steps
- Build Settings
- Post-build Actions

Build Triggers

- ☒ Build whenever a SNAPSHOT dependency is built ?
- ☐ Schedule build when some upstream has no successful builds ?
- ☐ Trigger builds remotely (e.g., from scripts) ?
- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☐ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?

Build Environment

- ☐ Delete workspace before build starts
- ☐ Use secret text(s) or file(s) ?
- ☐ Add timestamps to the Console Output
- ☐ Inspect build log for published Gradle build scans
- ☐ Terminate a build if it's stuck
- ☐ With Ant ?

Save **Apply**

- In the "goals and options" section, write "clean package".

localhost:8080/job/DeployWarExp-6/configure

Dashboard > DeployWarExp-6 >

Configuration

Add pre-build step ▾

- General
- Source Code Management
- Build Triggers
- Build Environment
- Pre Steps**
- Build
- Post Steps
- Build Settings
- Post-build Actions

Build

Root POM ?

pom.xml

Goals and options ?

clean package

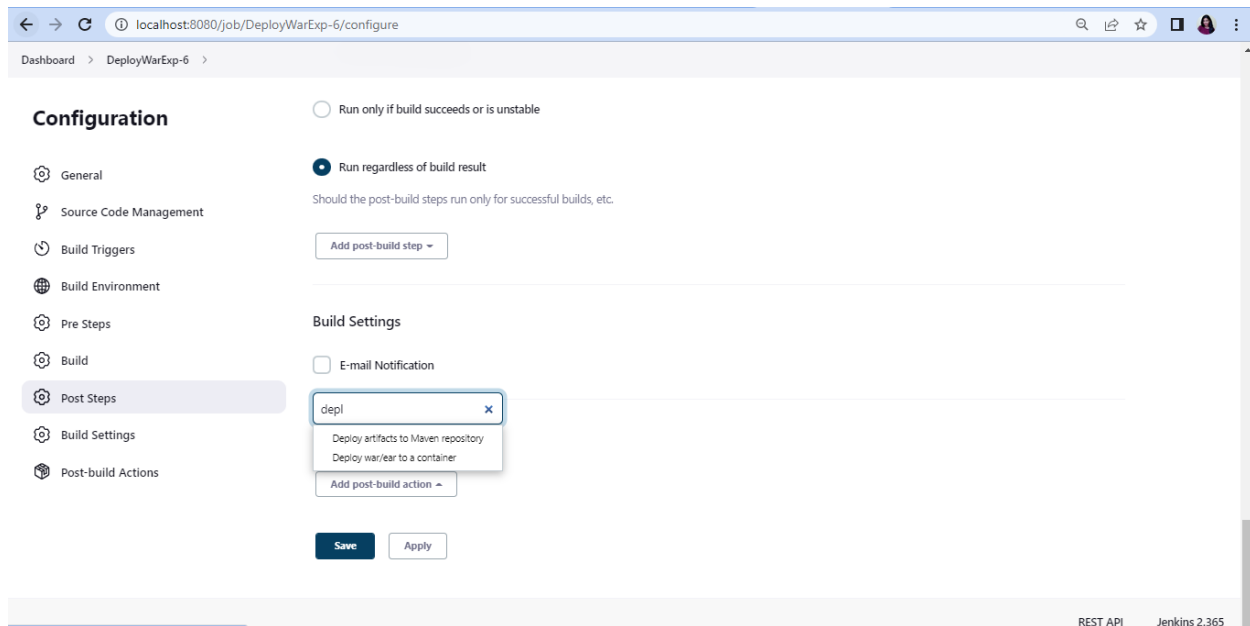
Advanced...

Post Steps

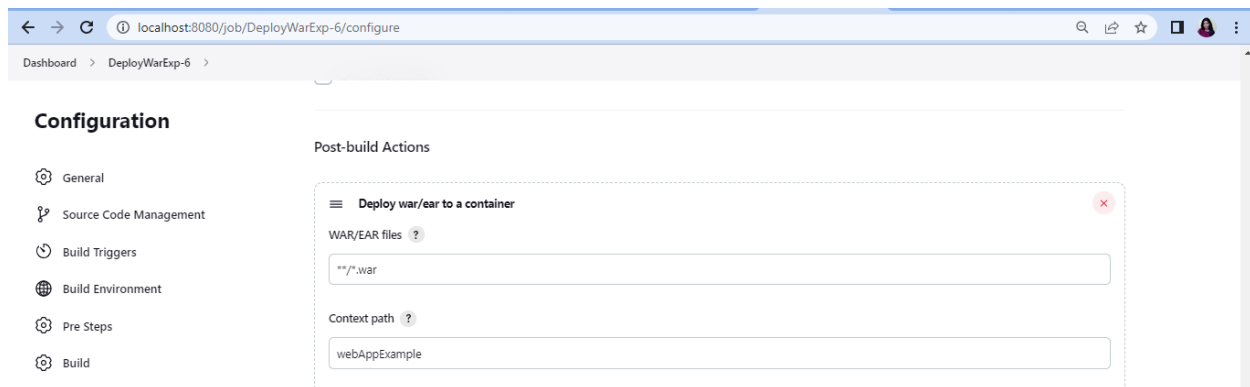
- ☐ Run only if build succeeds
- ☐ Run only if build succeeds or is unstable

Save **Apply**

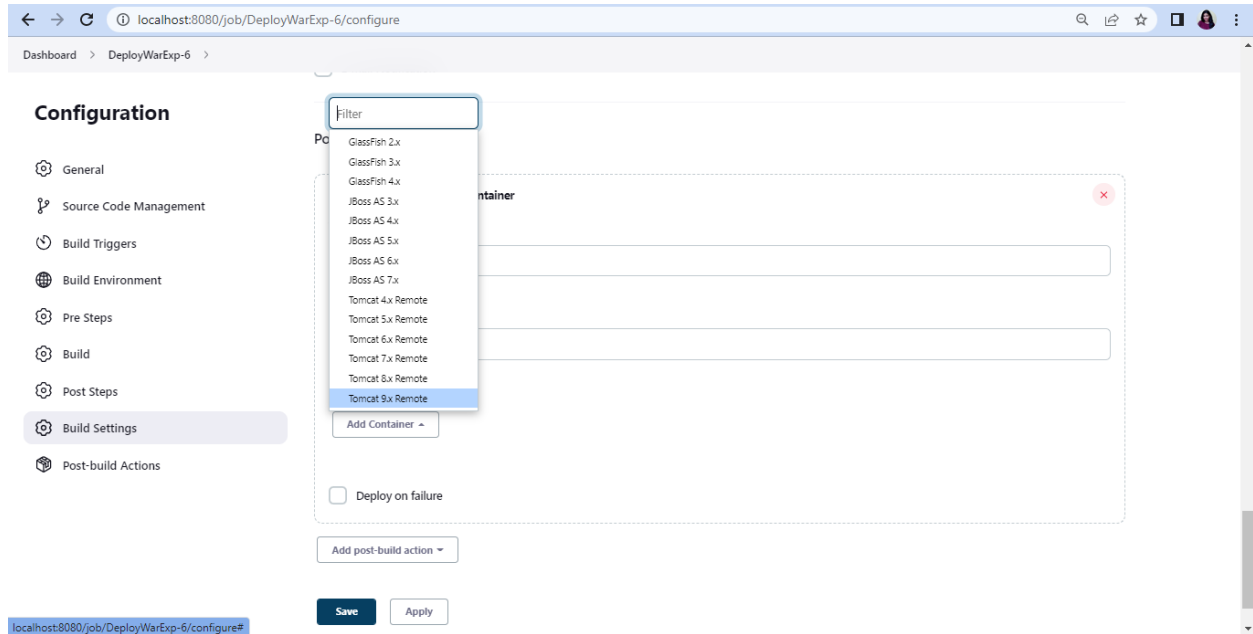
- Next in the post build actions choose deploy war/ear to a container and enter “**/*.war” in the input box



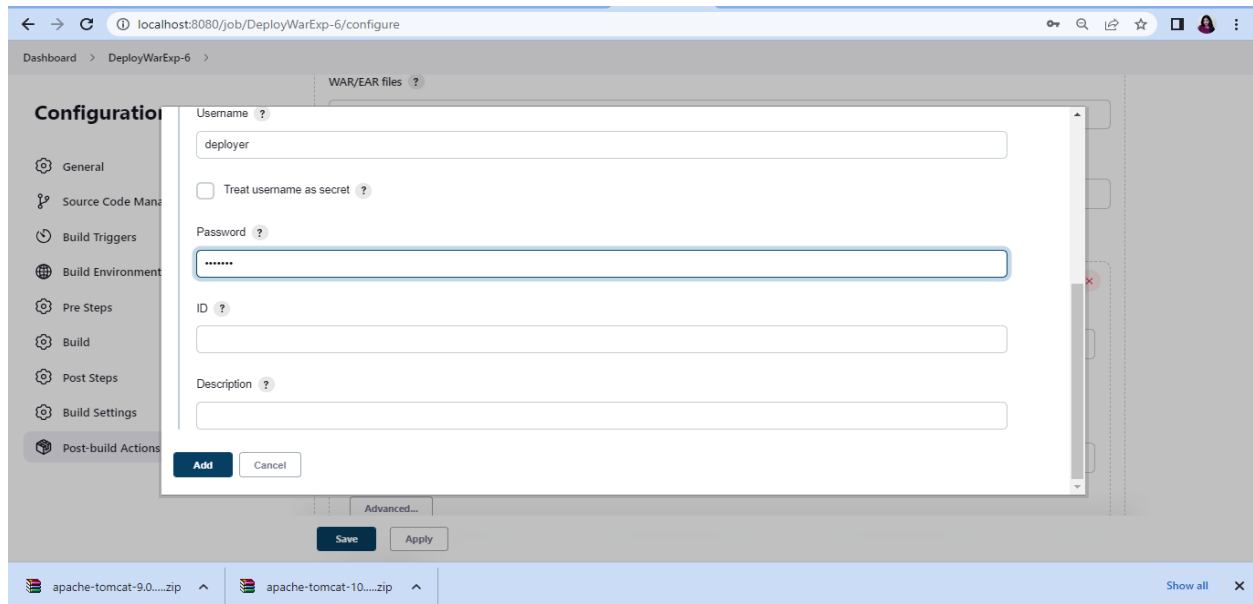
- Add war/ear files name as “**/*.war”
- Context Path name as “webAppExample”



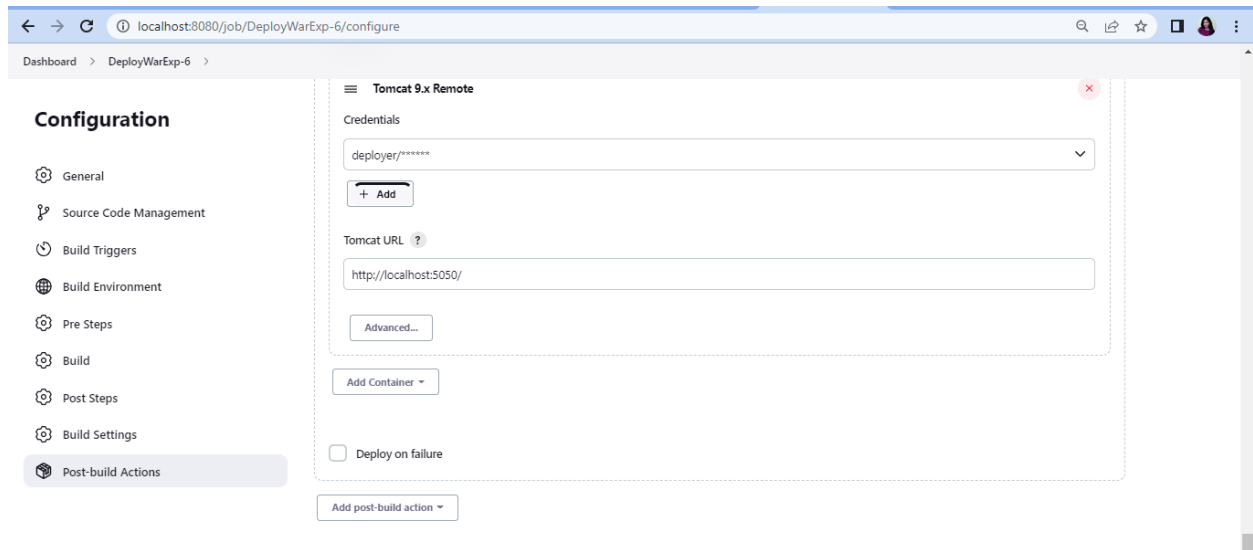
- Choose “tomcat 9x”



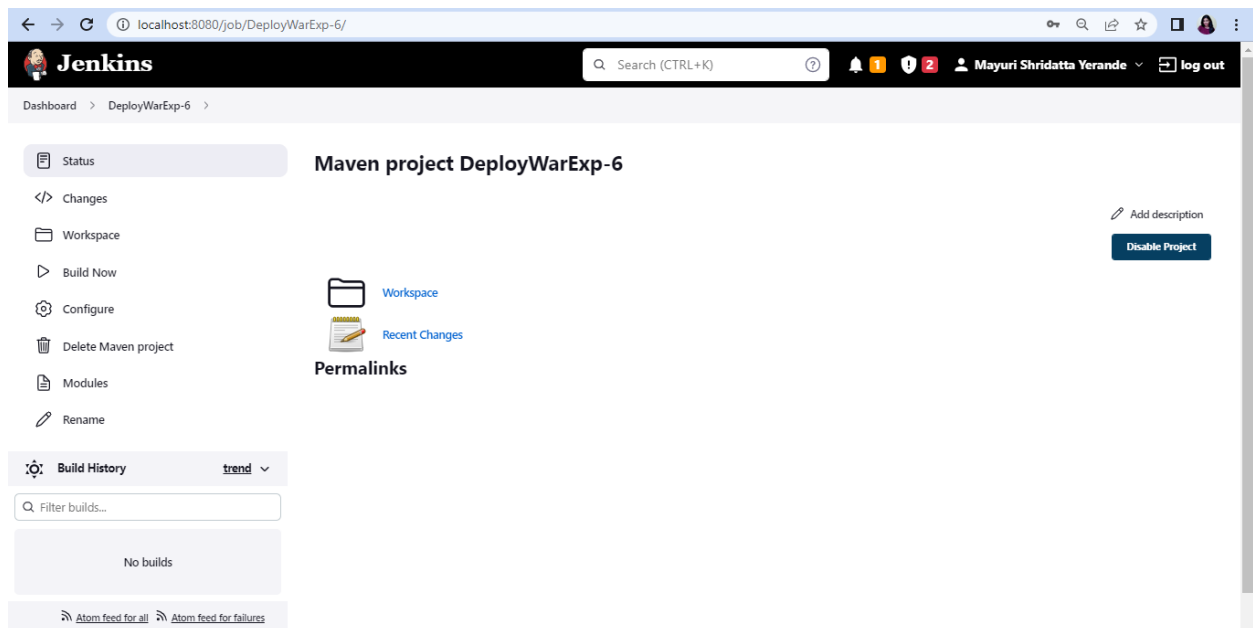
- Add the username and password which was added into “tomcat users file”
- Add this file in tomcat user first:
<user username="deployer" password="yourpassword" roles="manager-script" />



- Add the local host port of your tomcat server into “tomcat url” section



- Apply and save project

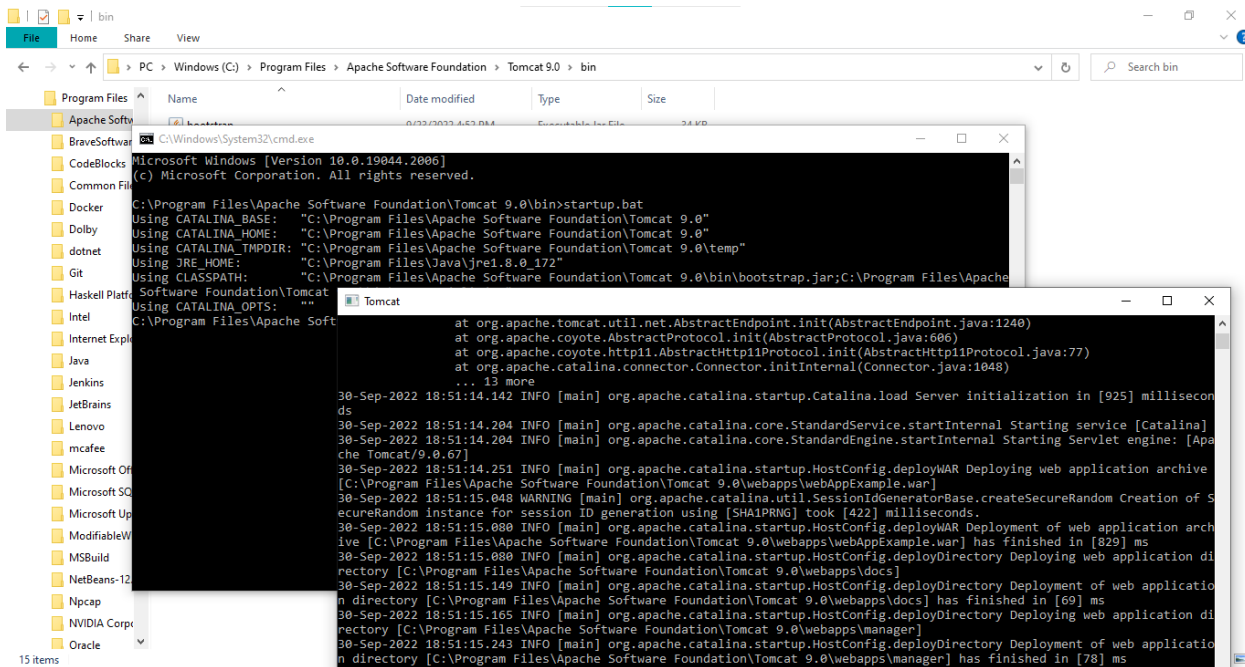


- (Keep running tomcat in your machine)
- Click on build now

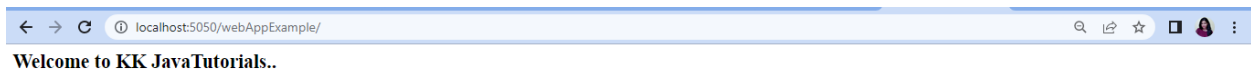
The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and user information for 'Mayuri Shridatta Yerande'. The main content area displays the 'Console Output' for a build named 'DeployWarExp-6'. The output text shows the build process starting with 'git rev-parse', fetching changes from a remote repository, and checking out a specific revision. The build is running as 'SYSTEM' in the workspace 'C:\ProgramData\Jenkins\jenkins\workspace\DeployWarExp-6'. The output also shows the build configuration, including the git tool and the repository URL. The build is currently in a 'Building' state, and the console output is being displayed in real-time.

The screenshot shows a web browser window displaying the console output of a Jenkins build. The browser address bar shows 'localhost:8080/job/DeployWarExp-6/7/console'. The console output text shows the build process, including packaging the webapp, assembling the war project, and copying resources. The build is successful, and the console output shows the build time and the finished time. The output also shows the build configuration, including the git tool and the repository URL. The build is currently in a 'Building' state, and the console output is being displayed in real-time.

- Project has been built successfully.
- Go to the path “C:\Program Files\Apache Software Foundation\Tomcat 9.0\bin”
- And put the following command: “startup.bat”



- Pipeline build is successful and the site is deployed successfully and working



Conclusion We successfully built the pipeline of jobs using Maven in Jenkins and created a pipeline script to build and deploy an application over the Tomcat server.