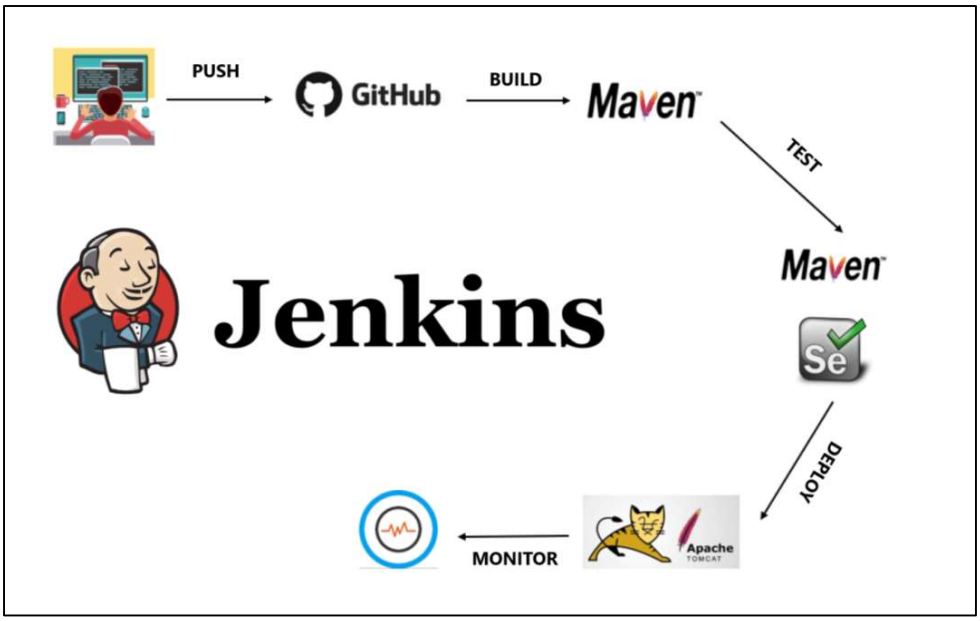
Experiment No : 05

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

Theory :

Jenkins is an open-source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. It is a server-based system that runs in servlet containers such as **Apache Tomcat.**



Maven is a build automation tool used primarily for Java projects. Maven can also be used to build and manage projects written in C#, Ruby, Scala, and other languages. The Maven project is hosted by the Apache Software Foundation, where it was formerly part of the Jakarta Project.

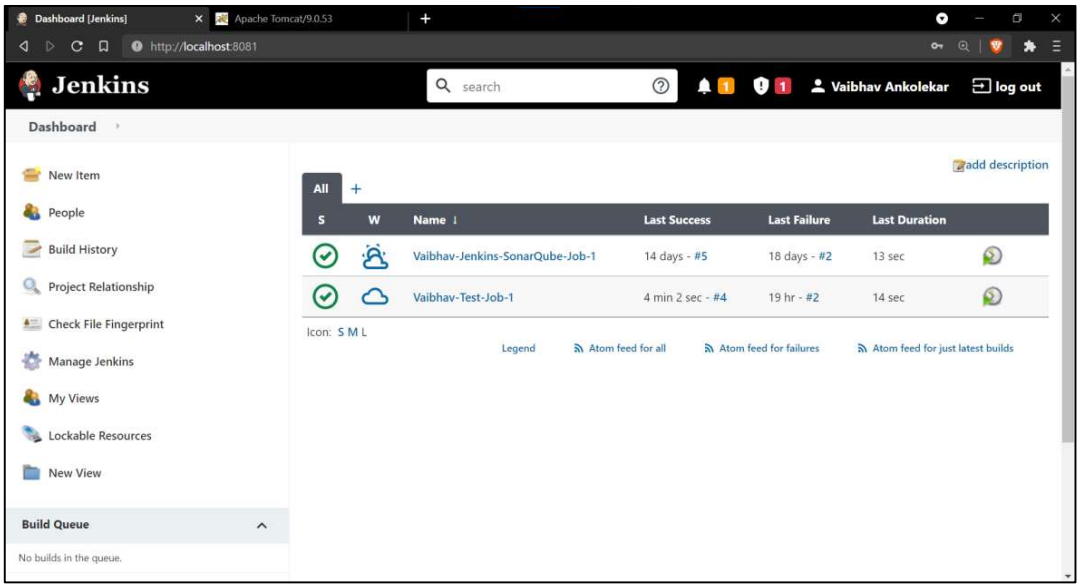
Hence, we can consider Apache Maven is a clear winner in the Jenkins vs Maven comparison. One of the reasons for this being is that Jenkins packages deploys with the Jenkins Maven plug-in. While on the surface level, Jenkins looks like it is doing most of the work, it's actually the Maven plug-in that does all of the heavy-duty work. Maven defines project structure, dependencies, build, and test management. Using pom.xml, we can configure dependencies needed for building testing and running code.

Prerequisite :

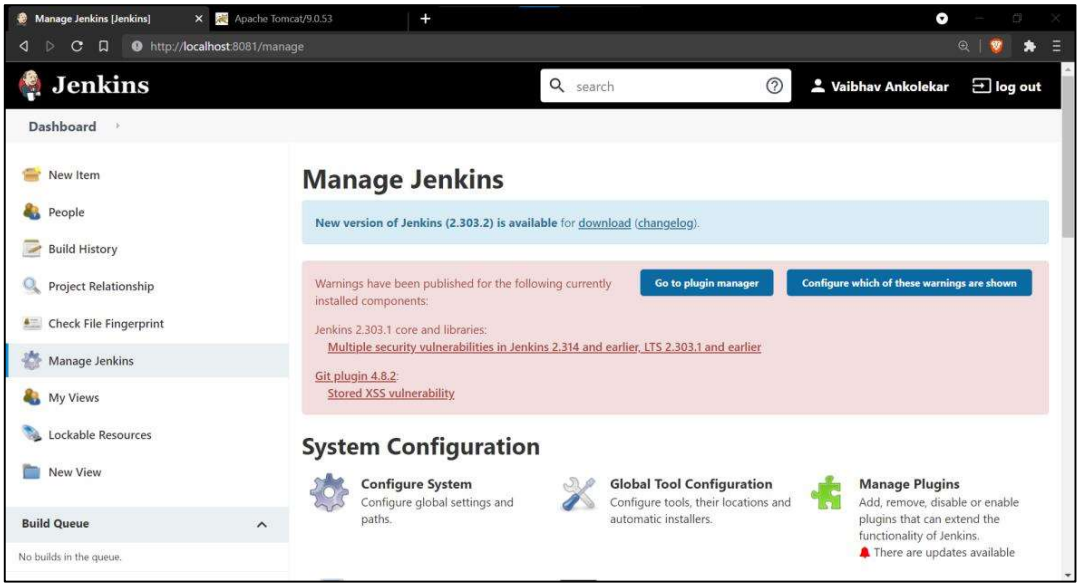
- Install Jenkins, Tomcat, Maven in your machine
- Create a user with admin privileges in Tomcat Server.
- Create a Web application and upload it to Github.

Procedure :

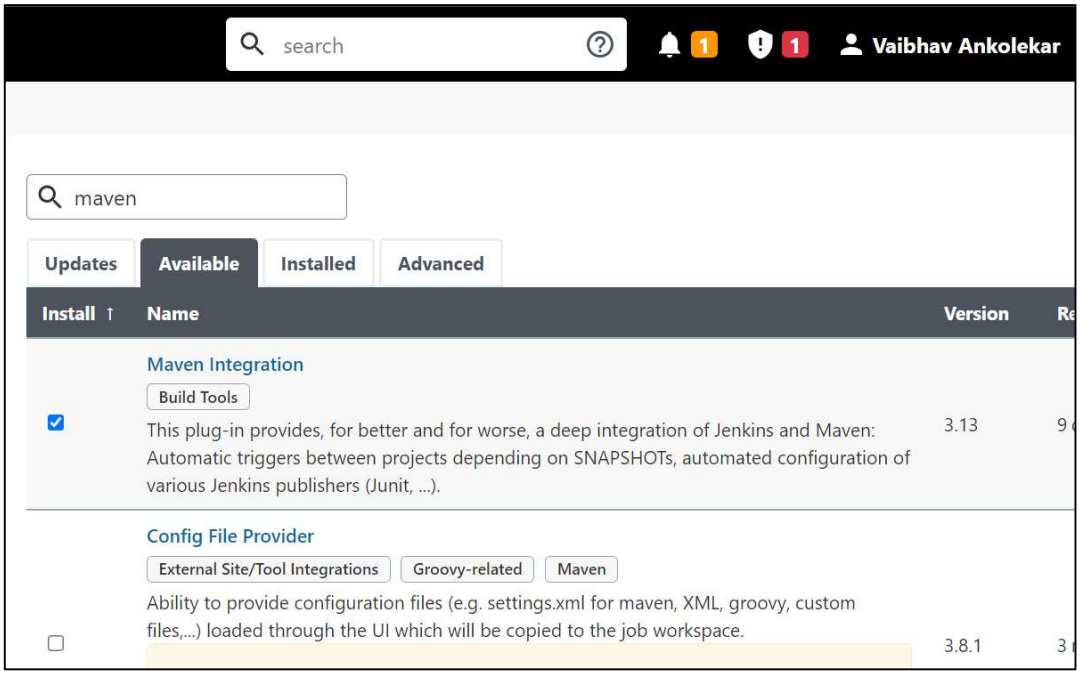
1) In your Jenkins, go to Manage Jenkins



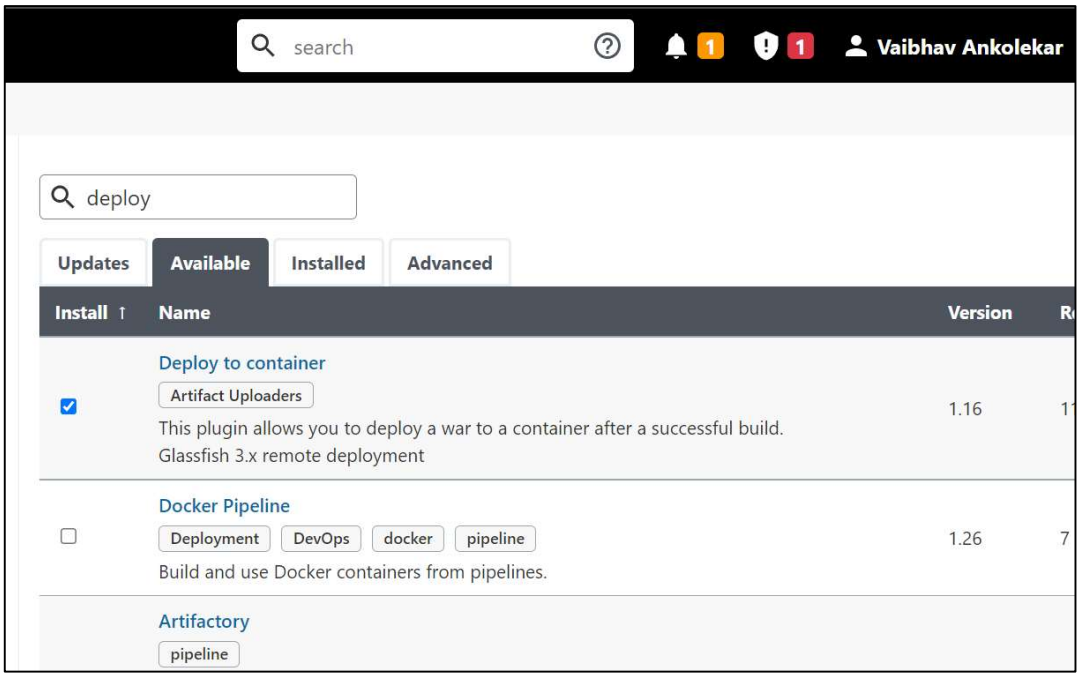
2) Now go to Manage Plugins



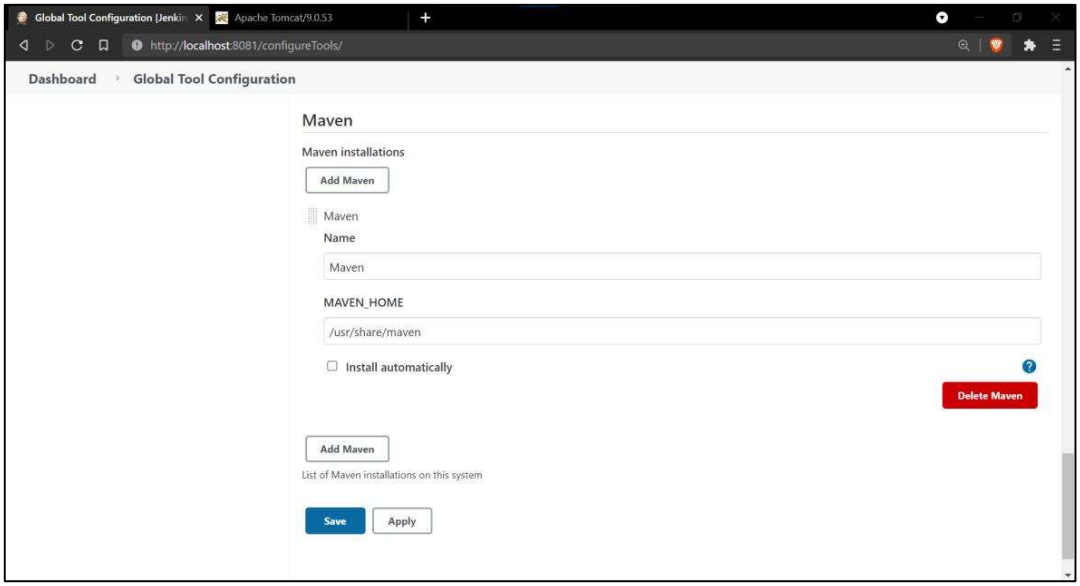
3) Install Maven Integration plugin.



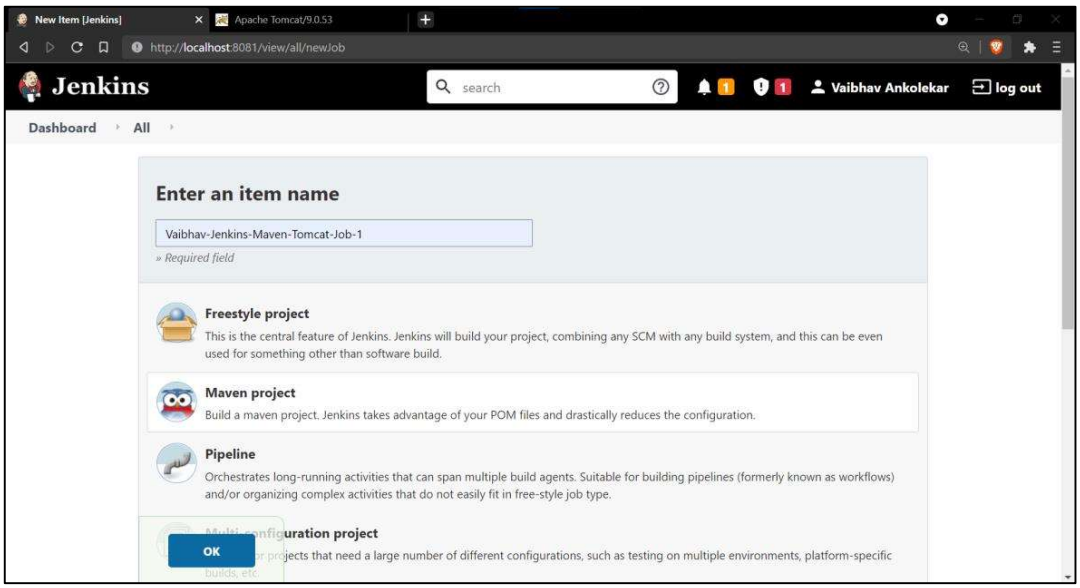
4) Install Deploy to container plugin.



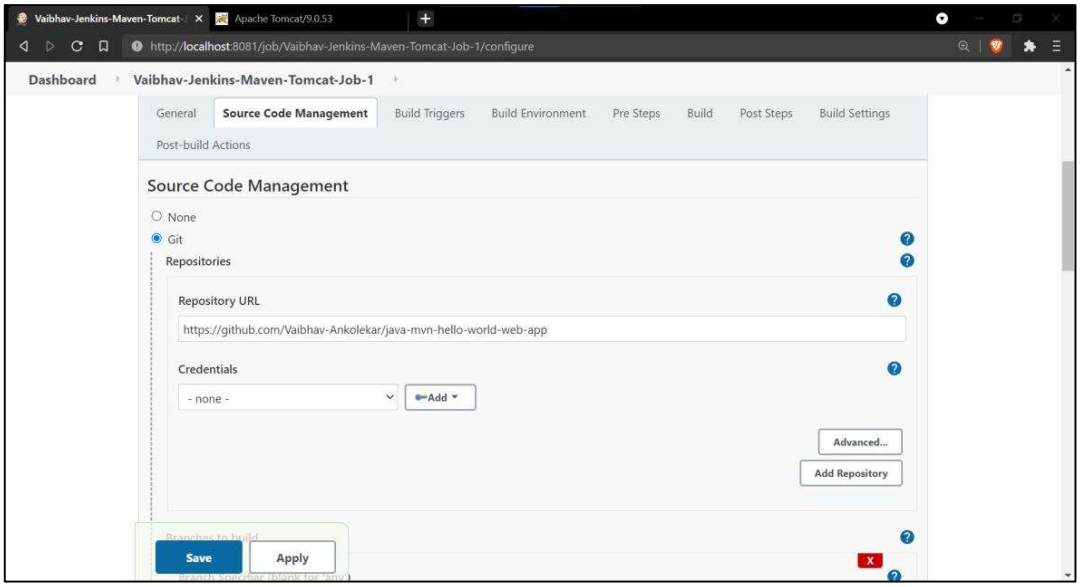
5) Go to Global Tool Configuration and add Maven. Give suitable name and the home directory of maven.



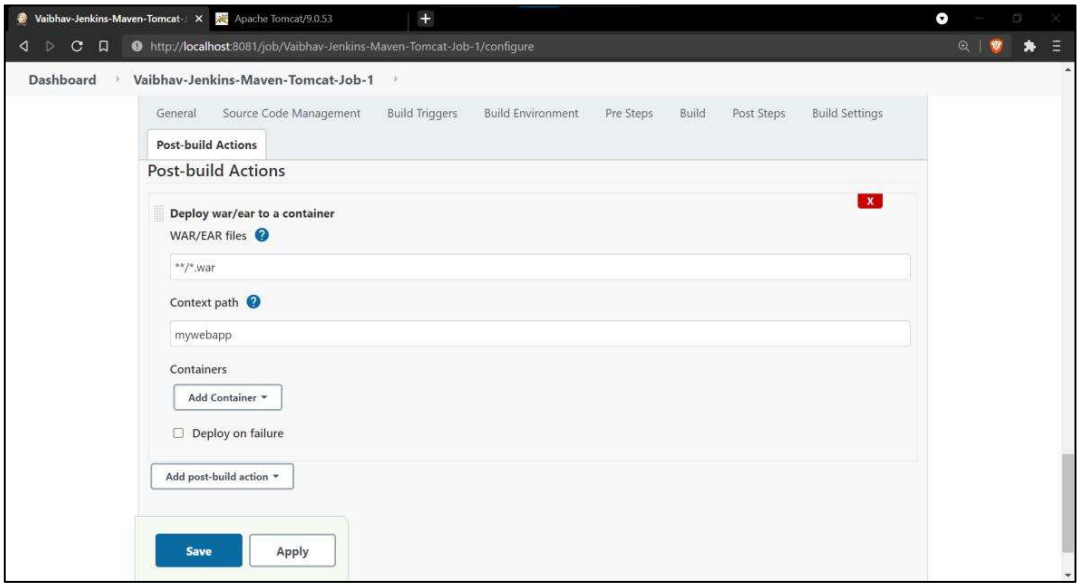
6) Now Go to Dashboard, and click on New Item. Enter Job Name and select Maven Project.



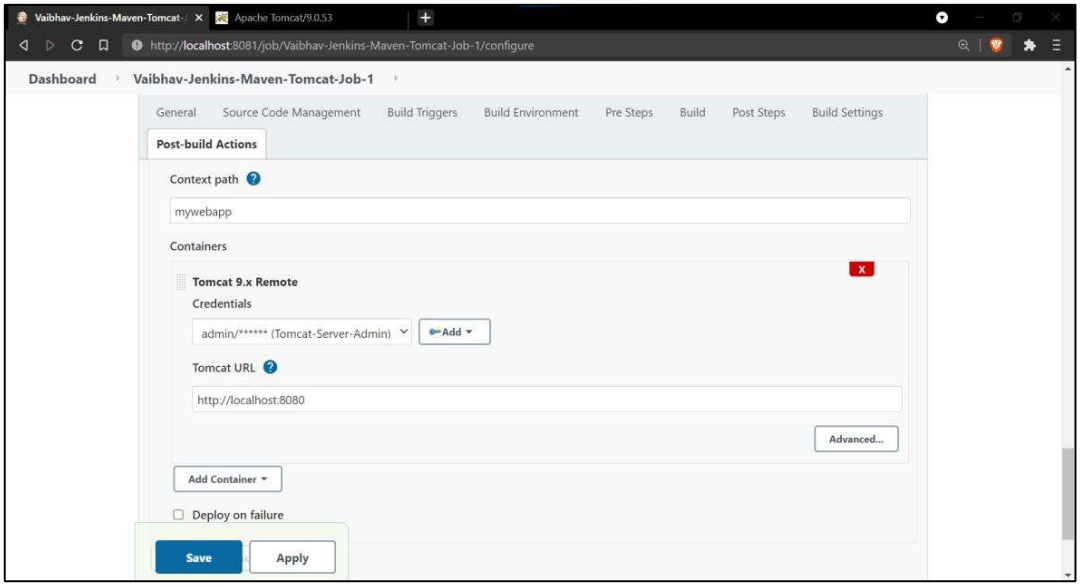
7) In Source Code Management, select Git and paste your Github repo url.



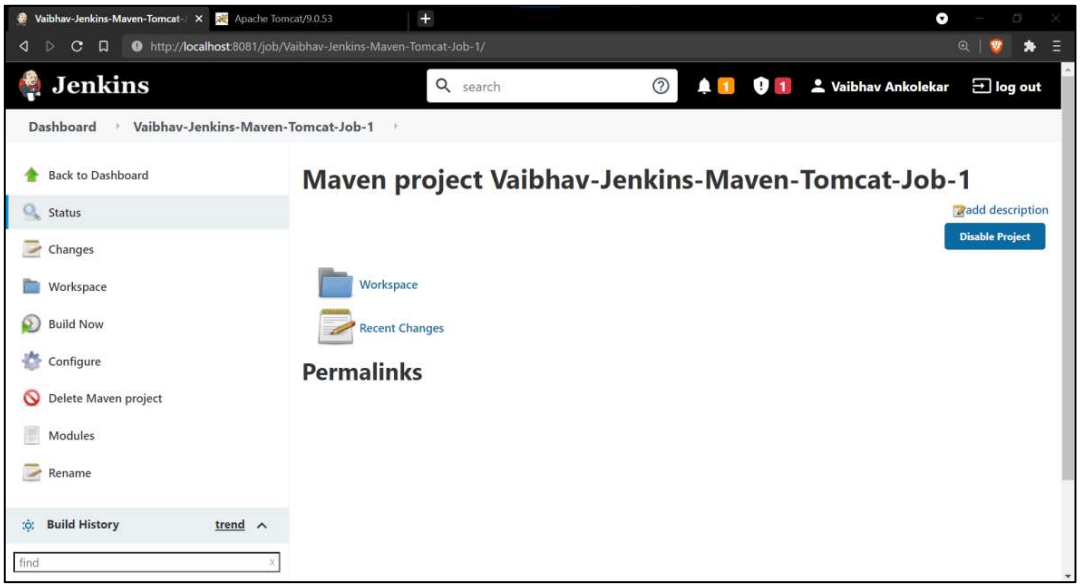
8) In Post-build Actions, click on Add post-build actions and select Deploy war/ear to a container. Enter the war file name which you want to deploy and enter a tomcat app name in Context path. Then click on Add container and select Tomcat version



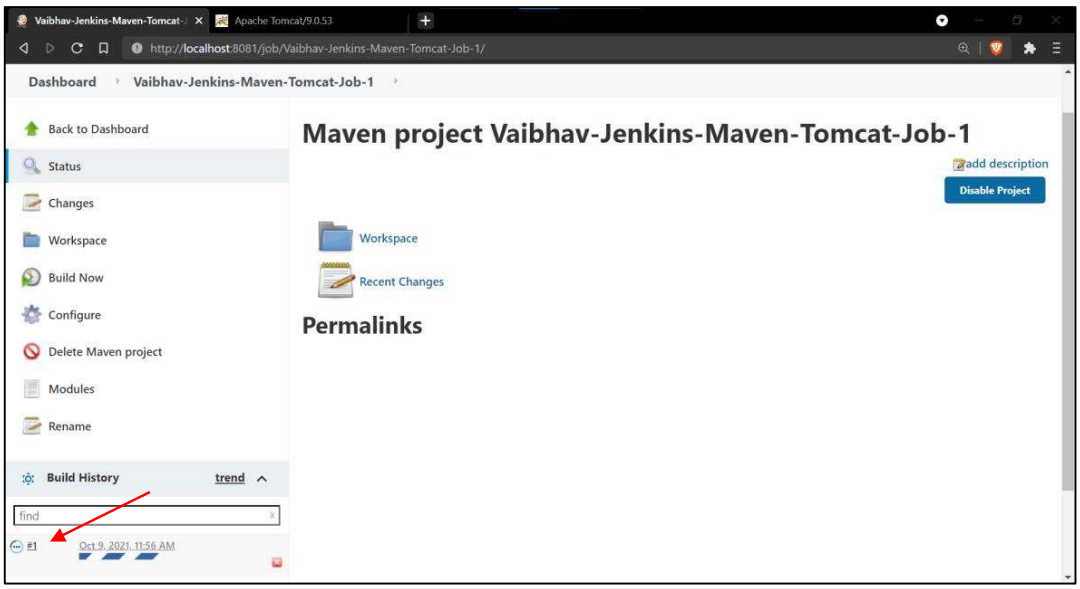
9) Click on Add under credentials and Enter your Tomcat user credentials and click on Create



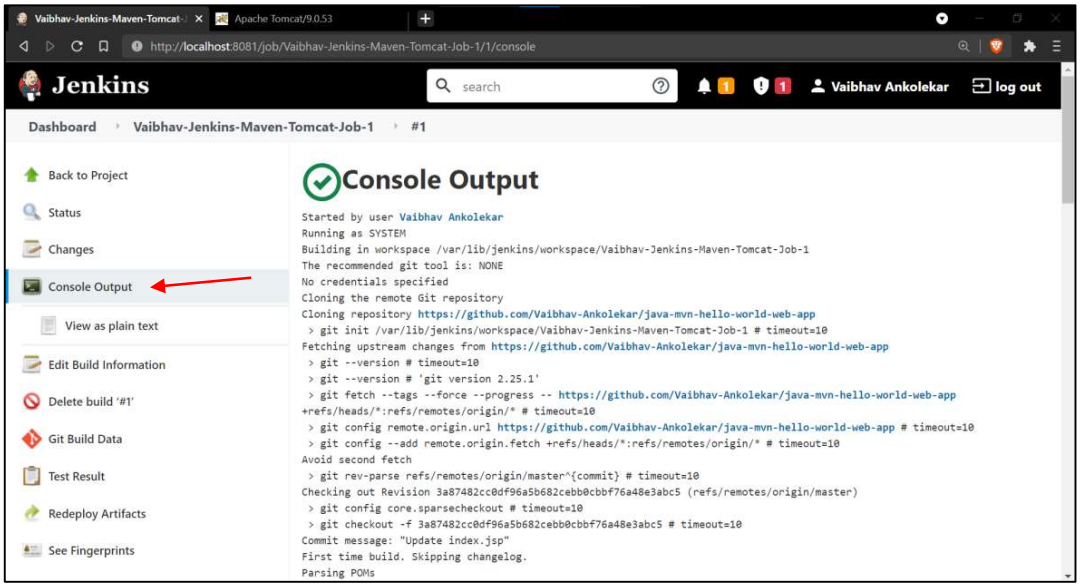
10) Click on Apply and Save. Now the job configuration is completed. Click on Build Now.



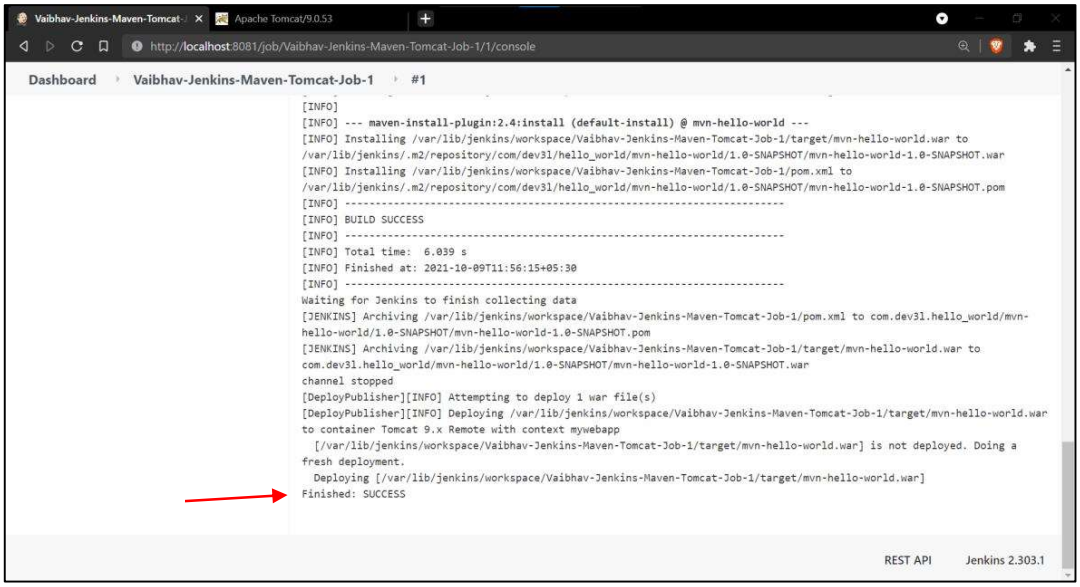
11) The job will start. Click on the job id under Build History.



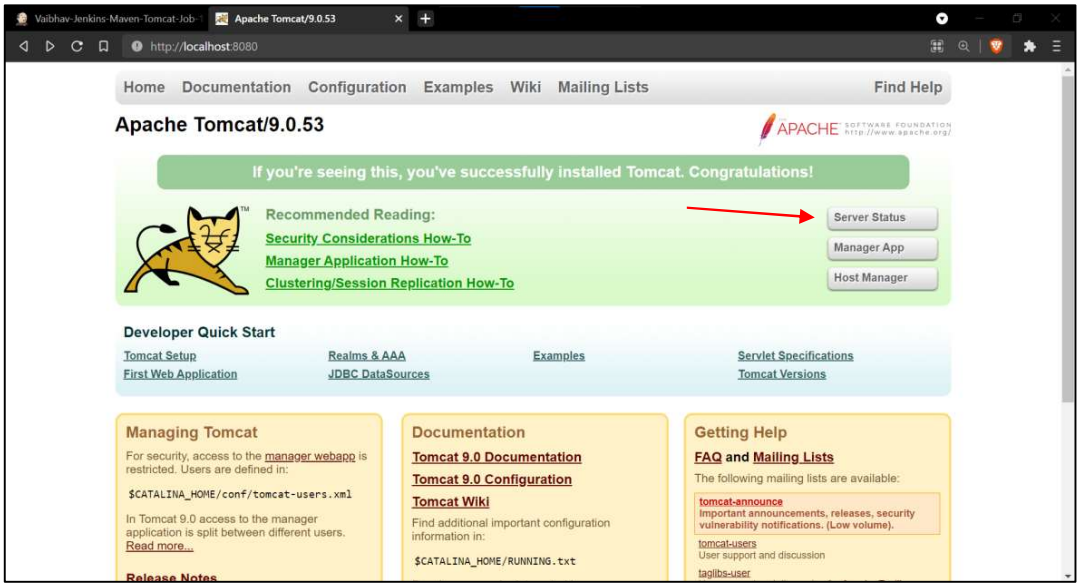
12) Click on Console Output to view the logs.



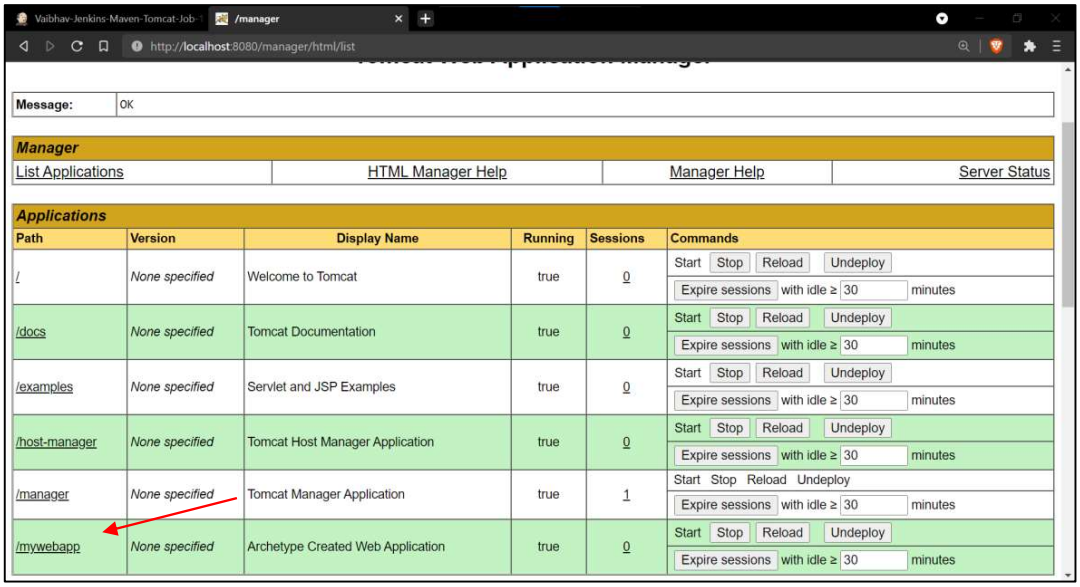
13) Go to bottom to see if the job is successfully completed.



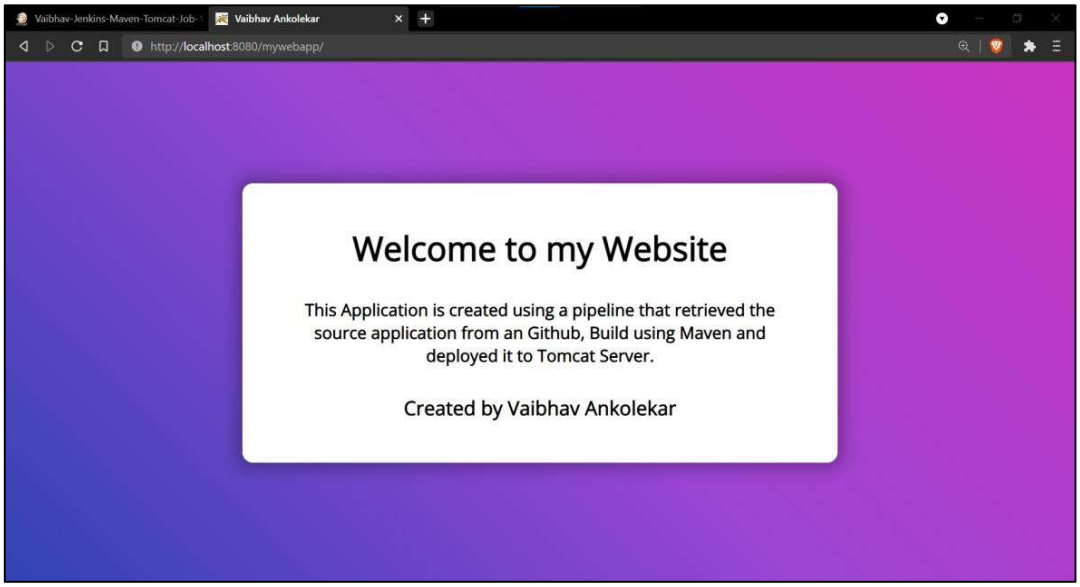
14) Go to Tomcat, and click on Server Status.



15) Scroll down and click on your app name under Applications Path column.



16) Your app is deployed on the Tomcat Server.



Conclusion : Thus, we have successfully build the pipeline of jobs using Maven in Jenkins, and deployed an application over the tomcat server