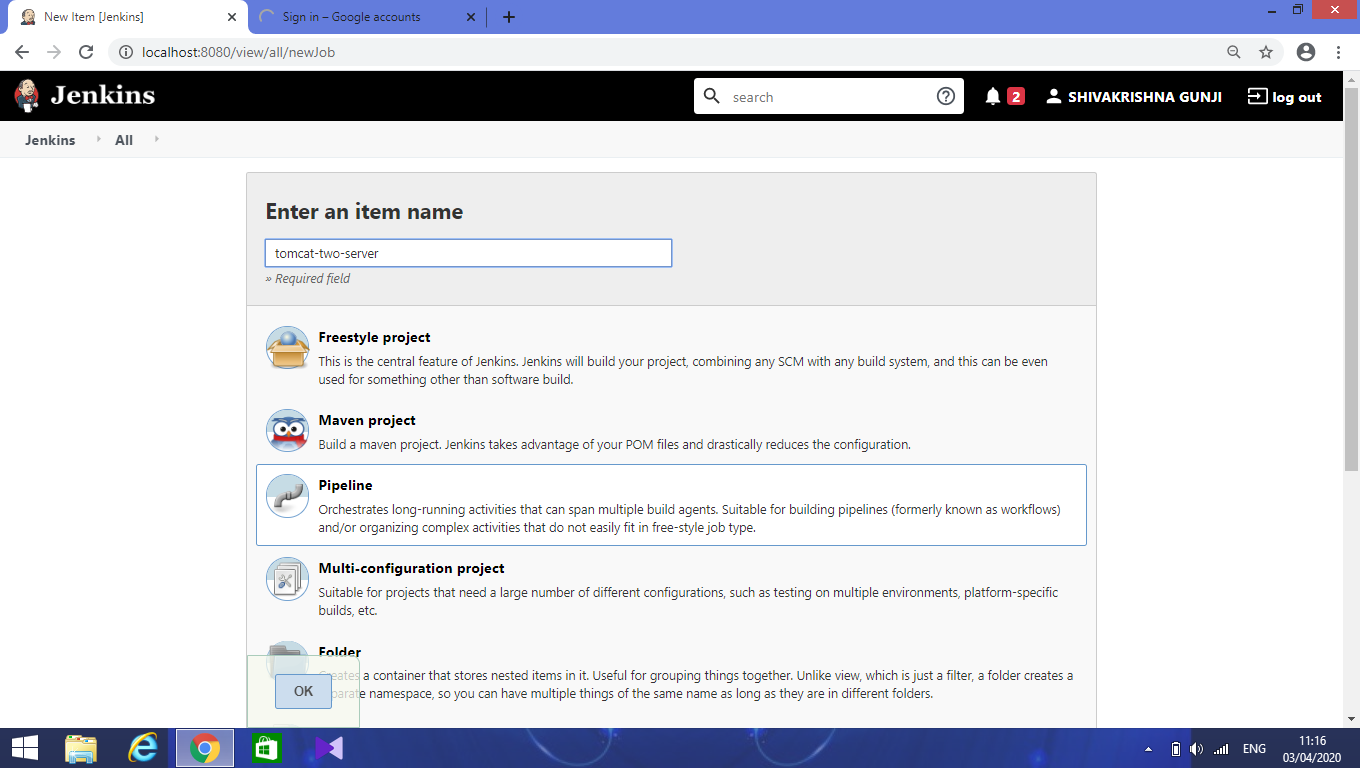
Create a pipeline job,



Enter a name for the job and select **pipeline** in the list.

Click ok.

Next, in the pipeline section configure the following values.

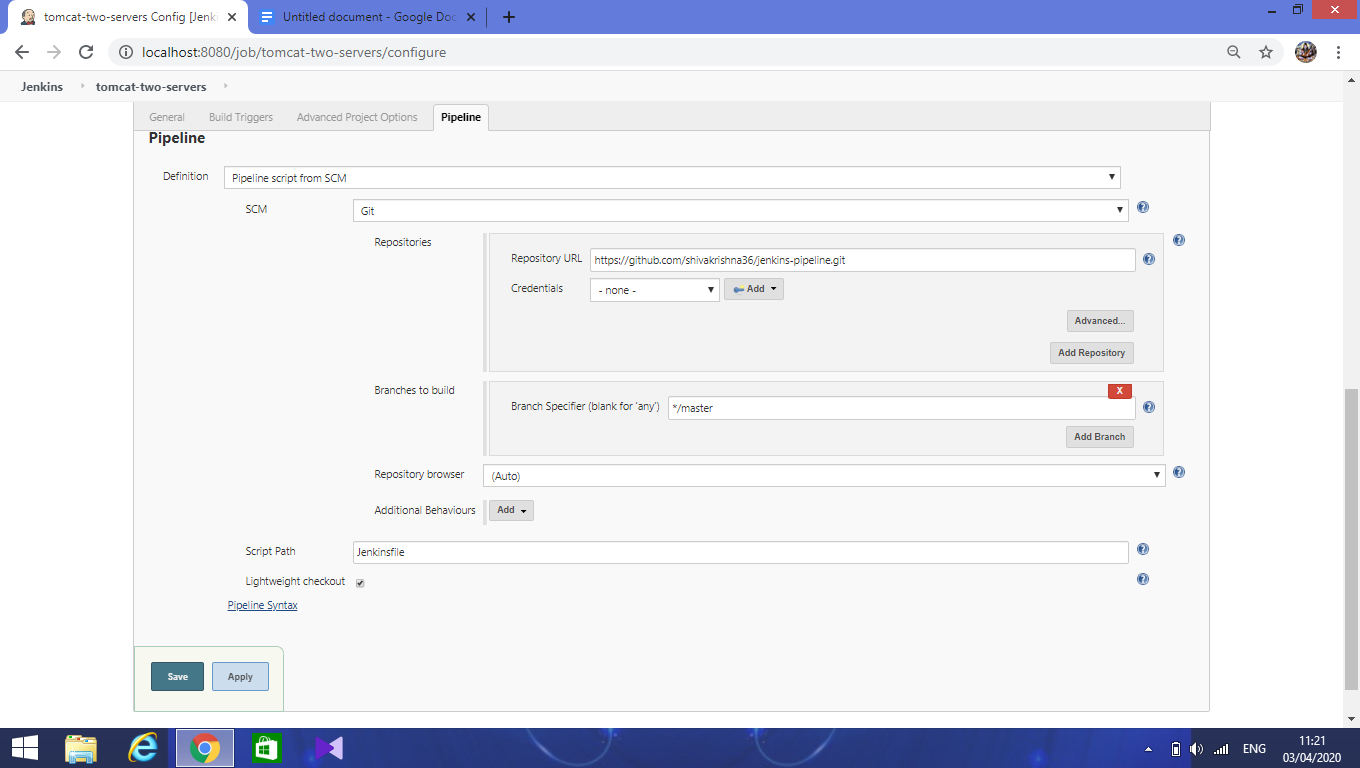
In definition, **select pipeline script from scm**.

In SCM, select **Git**.

In Repository url, enter the git repository url of the maven project which has a jenkins file.

Credentials not needed for public repository.

In the Script path, enter the file name **Jenkinsfile.**



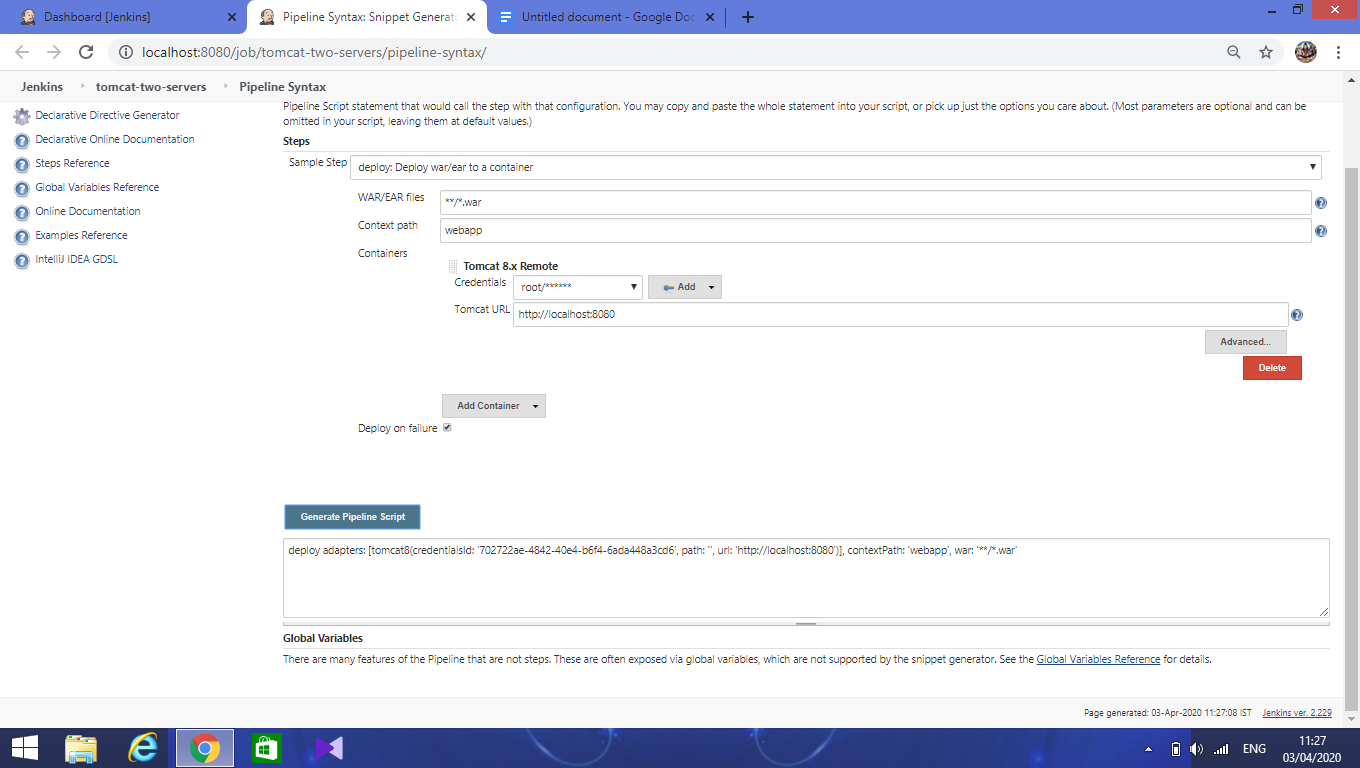
Click on **pipeline syntax** for generating the syntax for deploying the war into the container.

Here, select **deploy** option from the list.

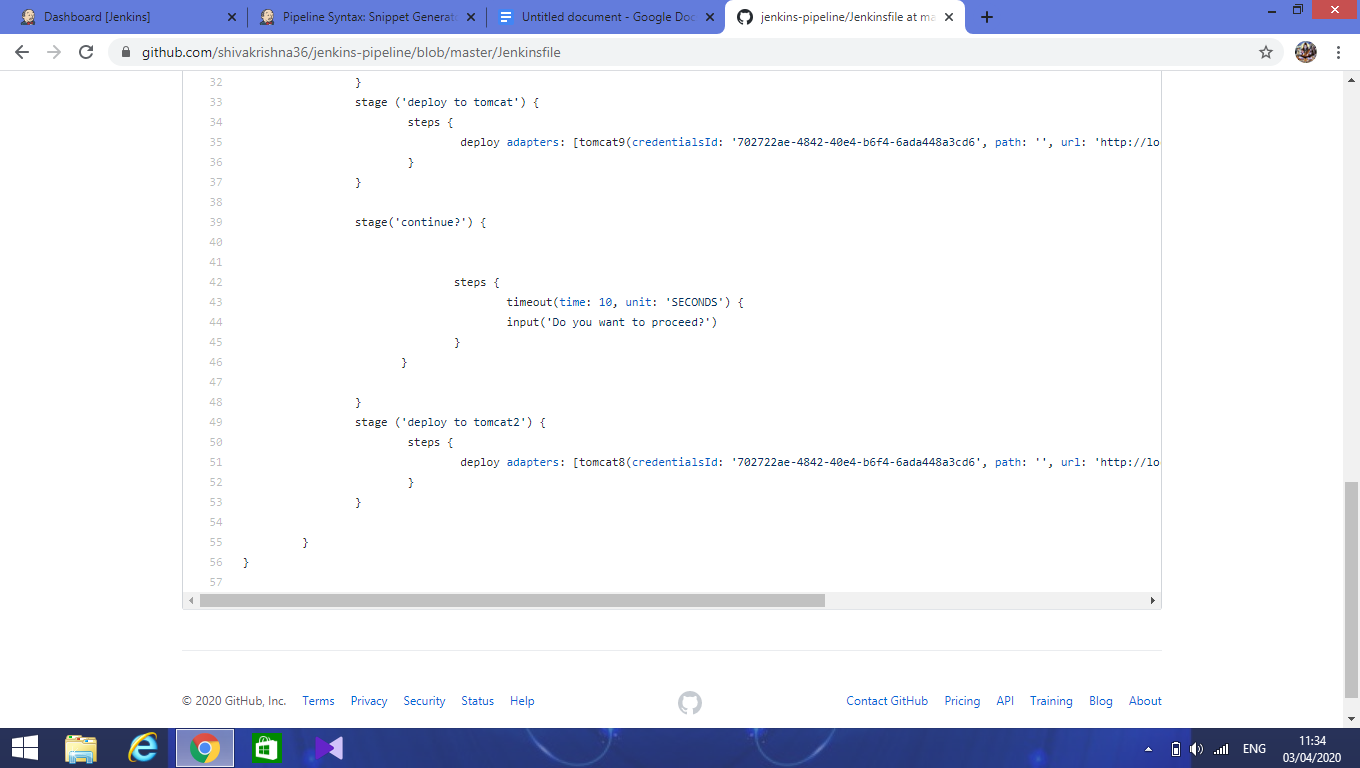
Context path is name of war file. Select the tomcat version from the list.

Then, provide the tomcat user credentials and enter tomcat url.

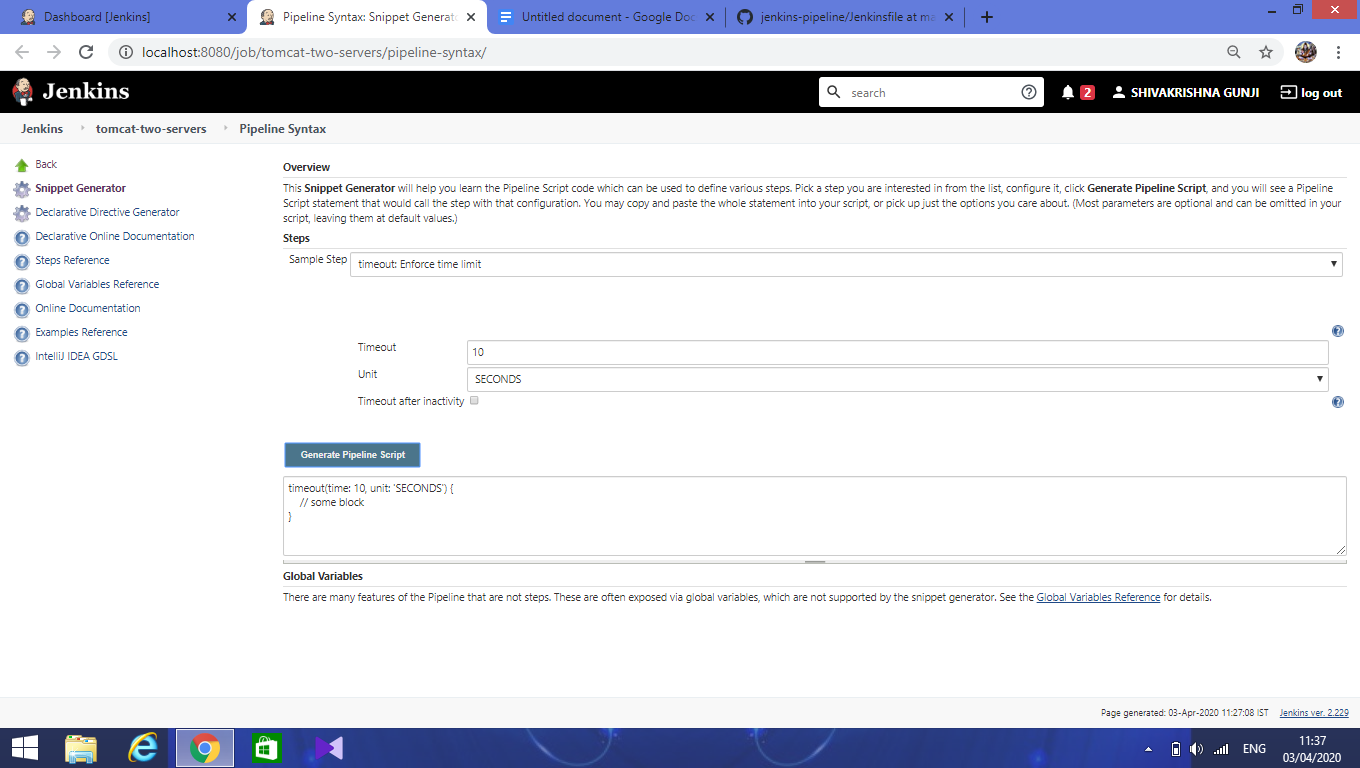
Click on generate script and copy that in Jenkinsfile.



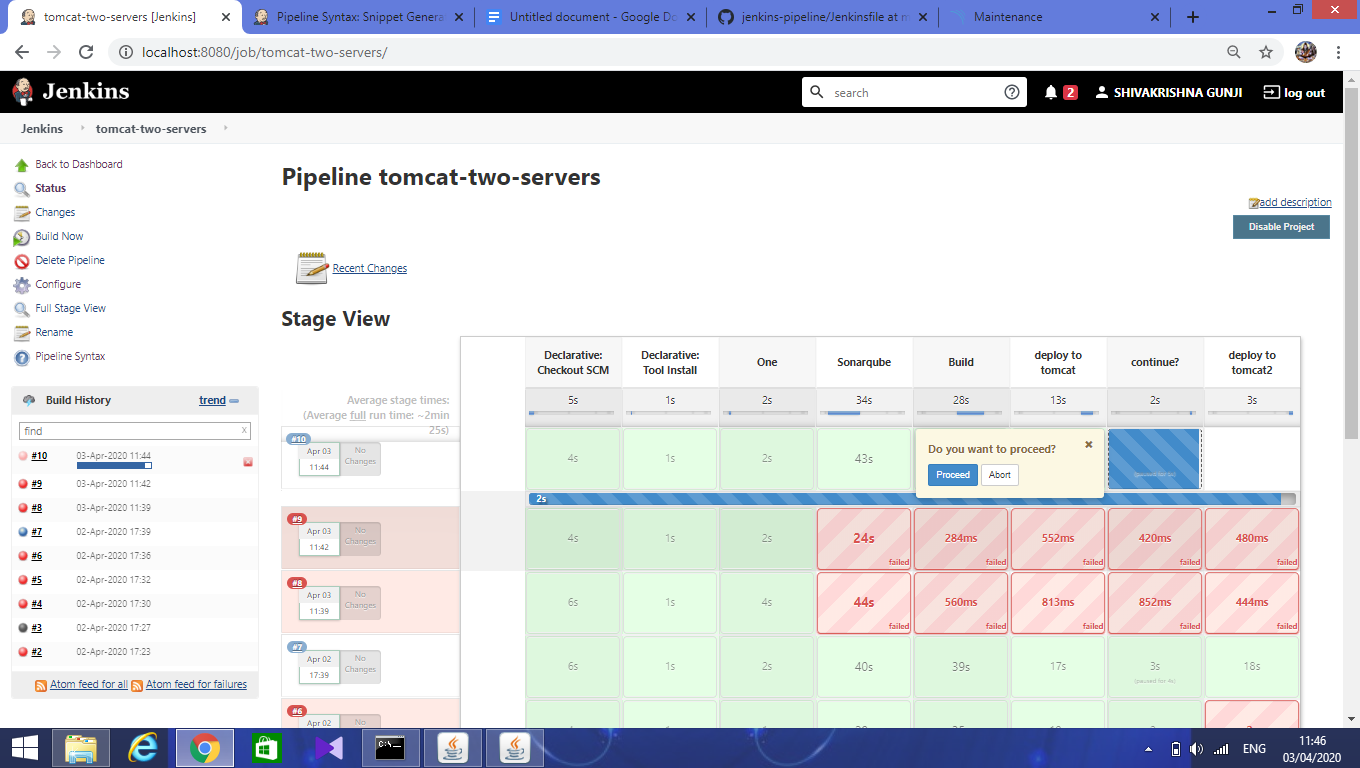
Paste the script in Jenkinsfile and save.



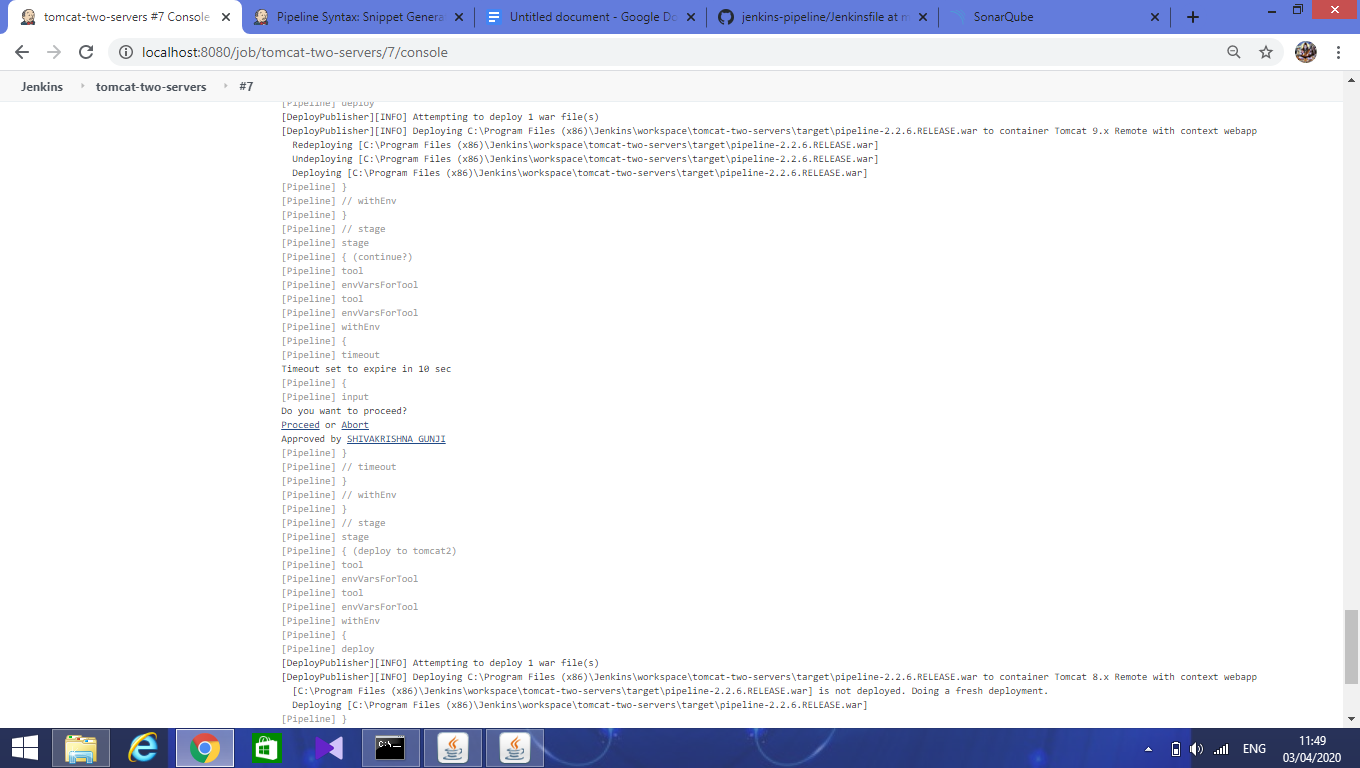
Similarly, we can generate the script for **timeout** and paste in Jenkinsfile.

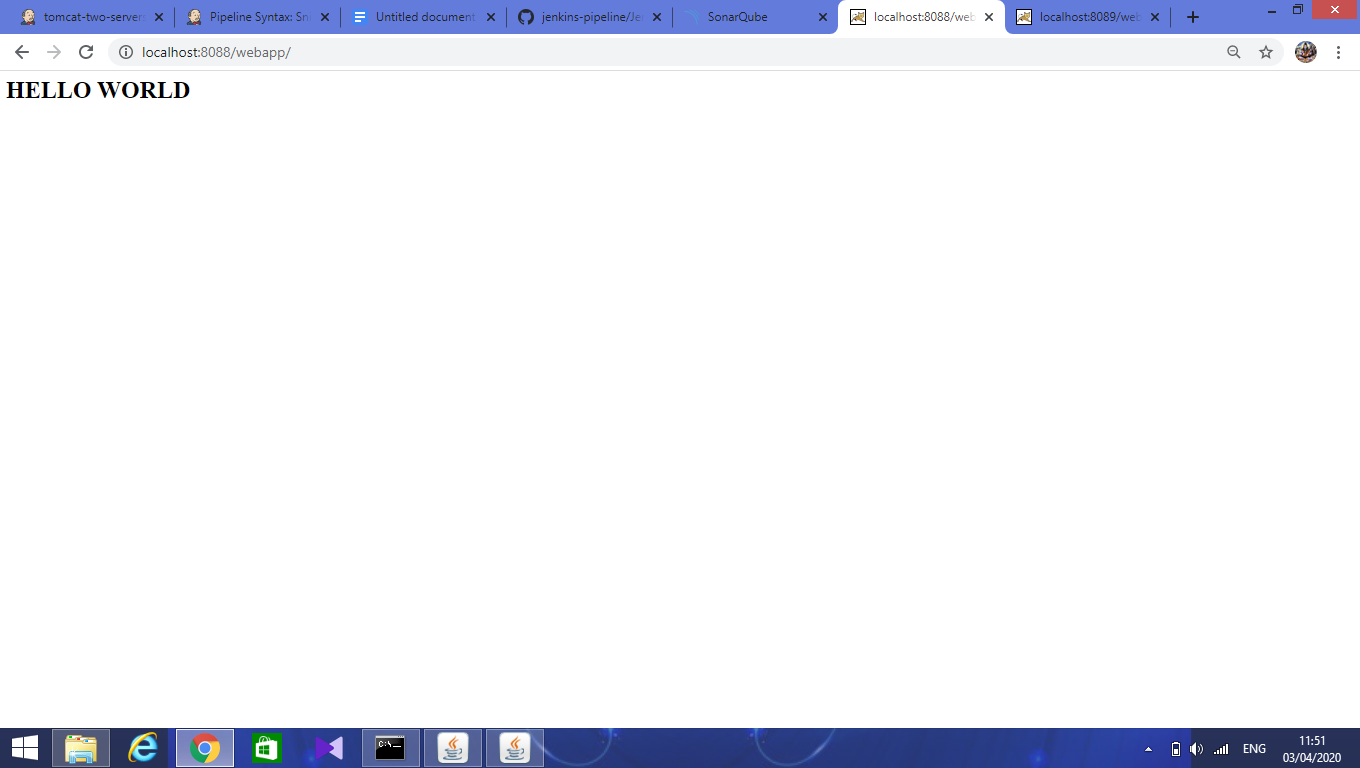


Now, in jenkins dashboard click on build now,



After deploying to the first tomcat server, it will ask to proceed or abort. It will proceed only when allowed by Jenkins admin who started the build. It will abort if the timeout has reached its limit.





After successful build, check the tomcat urls for final output.

