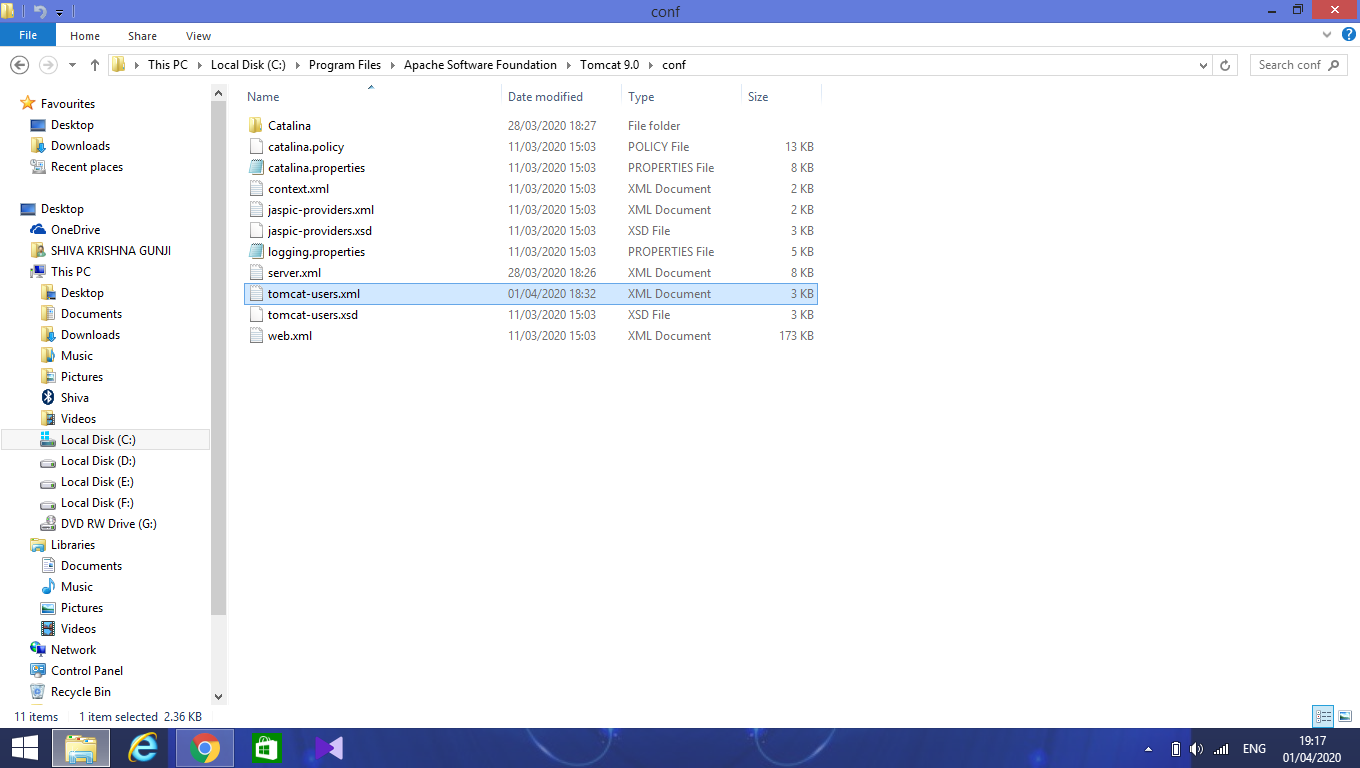
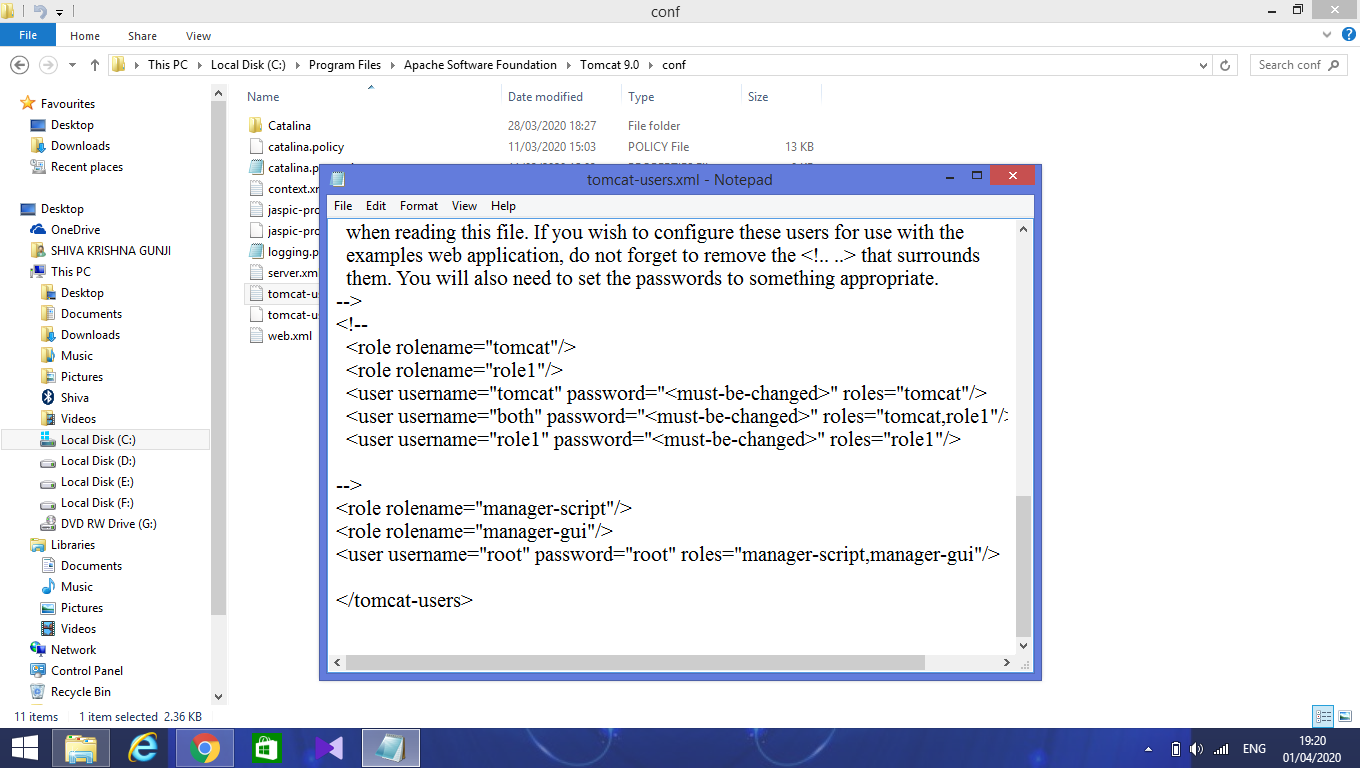
Deploy war file to tomcat container through pipeline script

Configure tomcat user,

Open tomcat-users.xml which is at location conf/tomcat-users.xml



Add a new user with a manager-script role. So, that it will allow the deployment of war files from the pipeline script as follows.

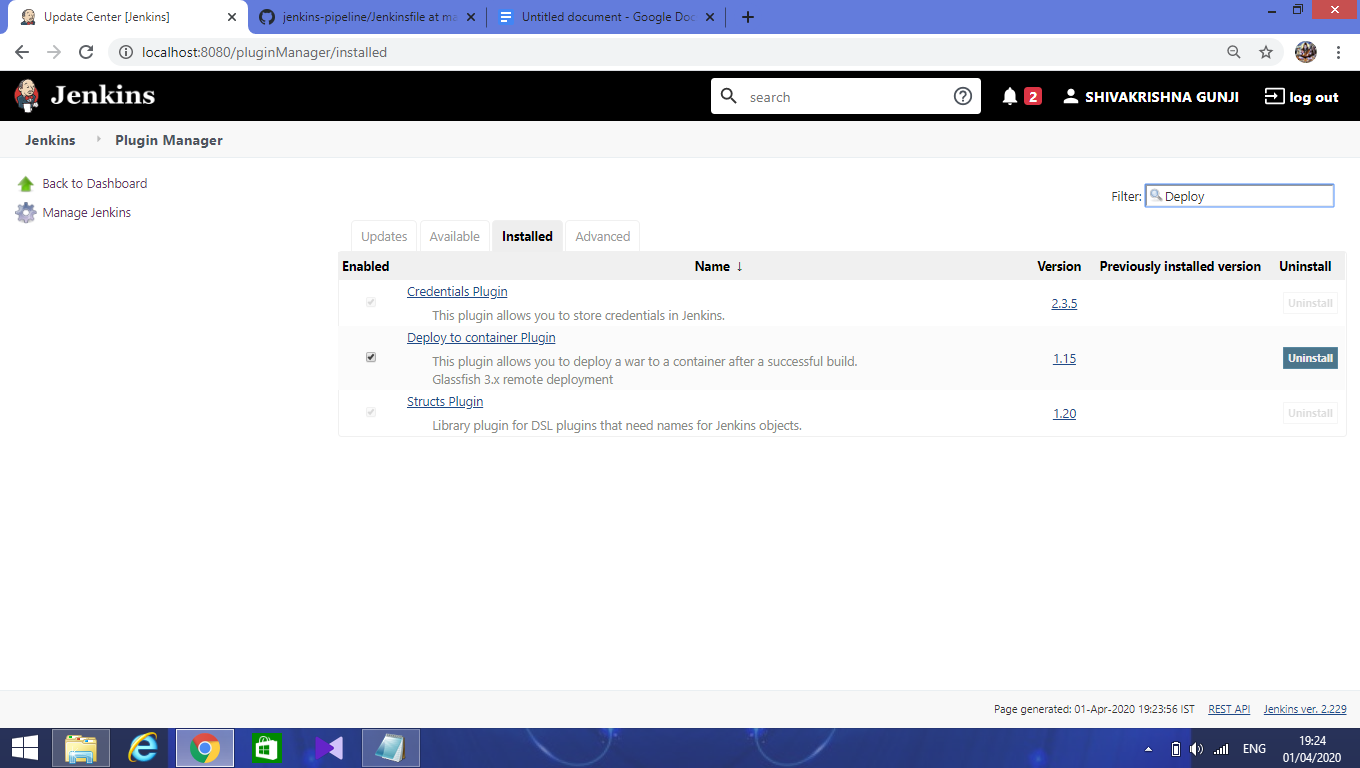


After creating the user, open jenkins dashboard and install plugin **Deploy to Container**.

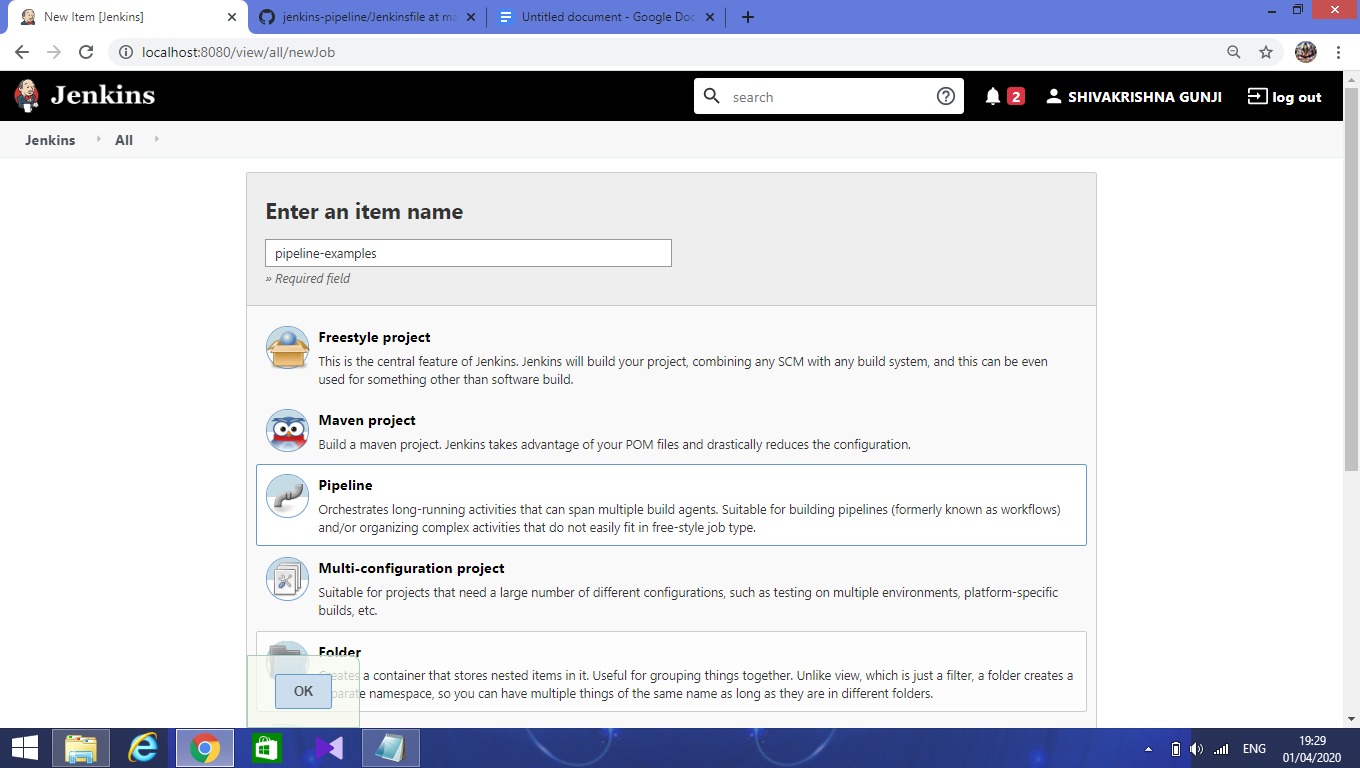
Under manage jenkins->plugin manager->available. Search for deploy.

Select **Deploy to container** plugin and click on **install without restart**.

After successful install restart the jenkins.



After restarting jenkins, create a job for pipeline and configure as follows:



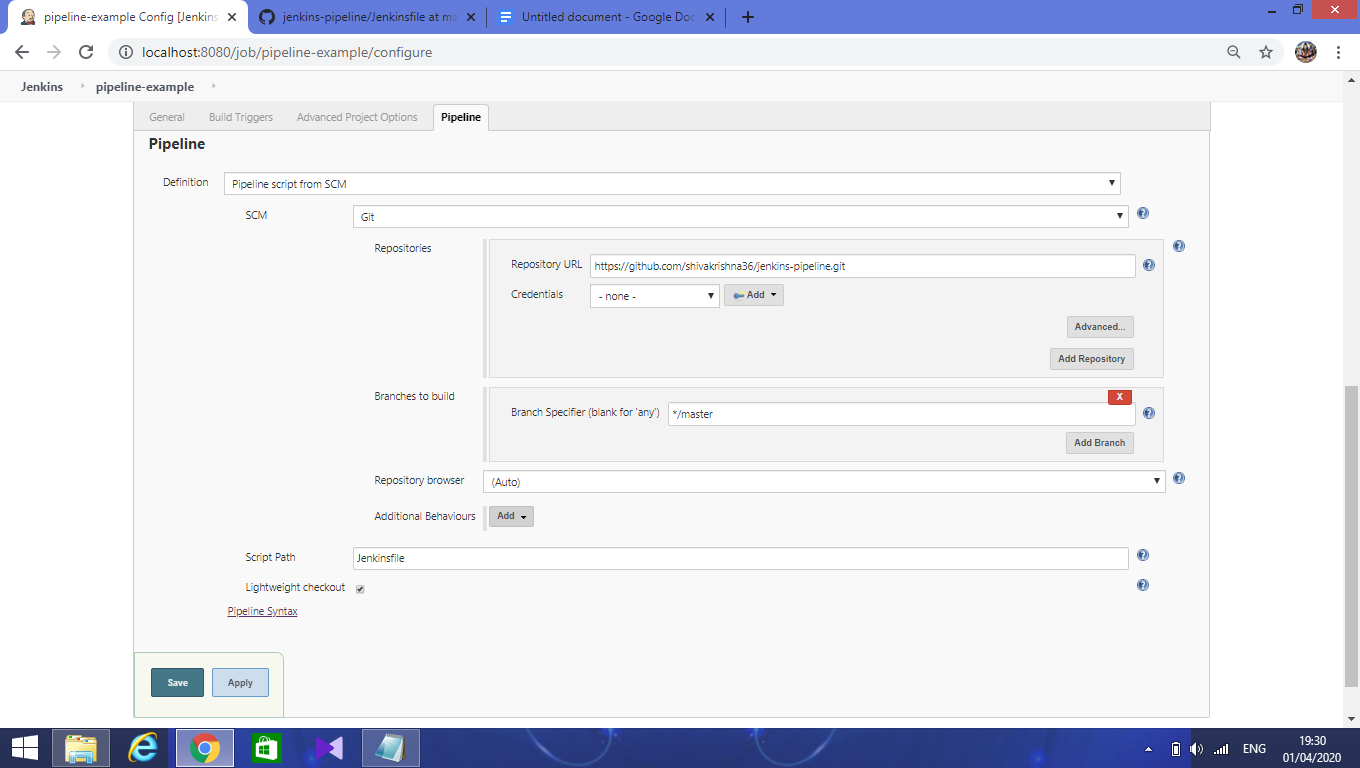
Next, under the **pipeline** section configure github details for your maven project.

Enter the repository url of github.

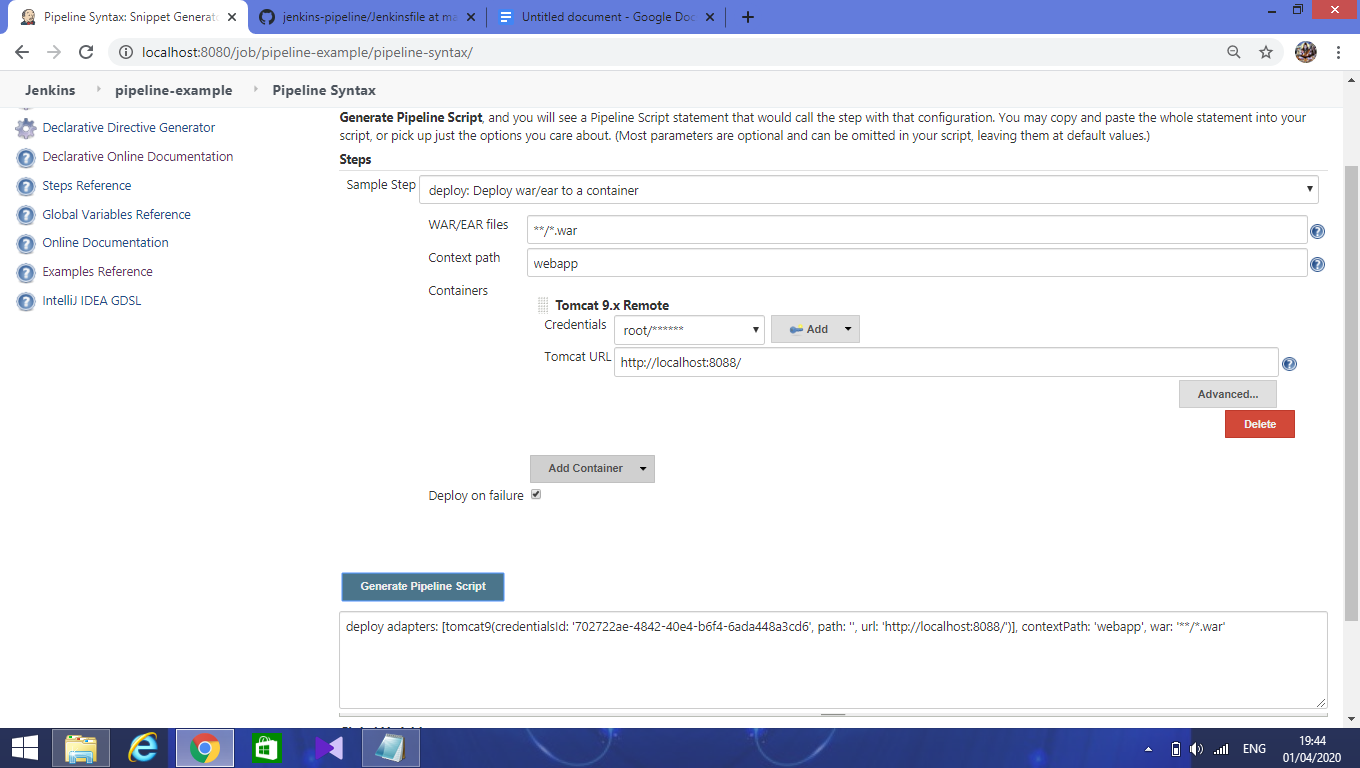
Credentials are not needed for public repositories.

If your project is in other than the master branch, enter that name in **Branches to build.**

Secretpath - which is **Jenkinsfile**.



Click on **pipeline syntax link** for generating the syntax for deploying war file to container.



Here, in the sample step select **deploy** option.

**Configure as follows:**

**WAR/EAR files** - enter \*\*/\*.war.

**Context path** - name of war file.

**Containers** - select tomcat from list and credentials of tomcat user and enter the url of tomcat.

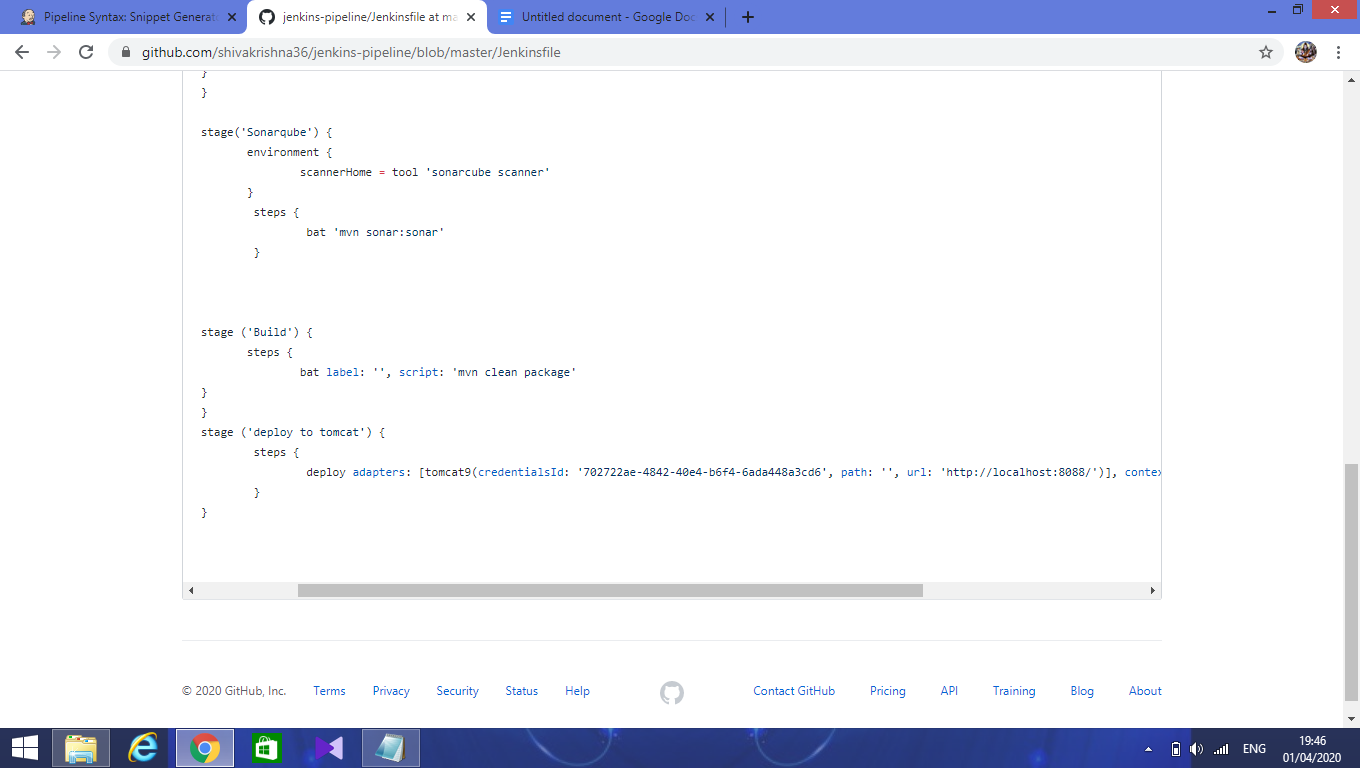
Next, click on generate pipeline script.

And copy that command.

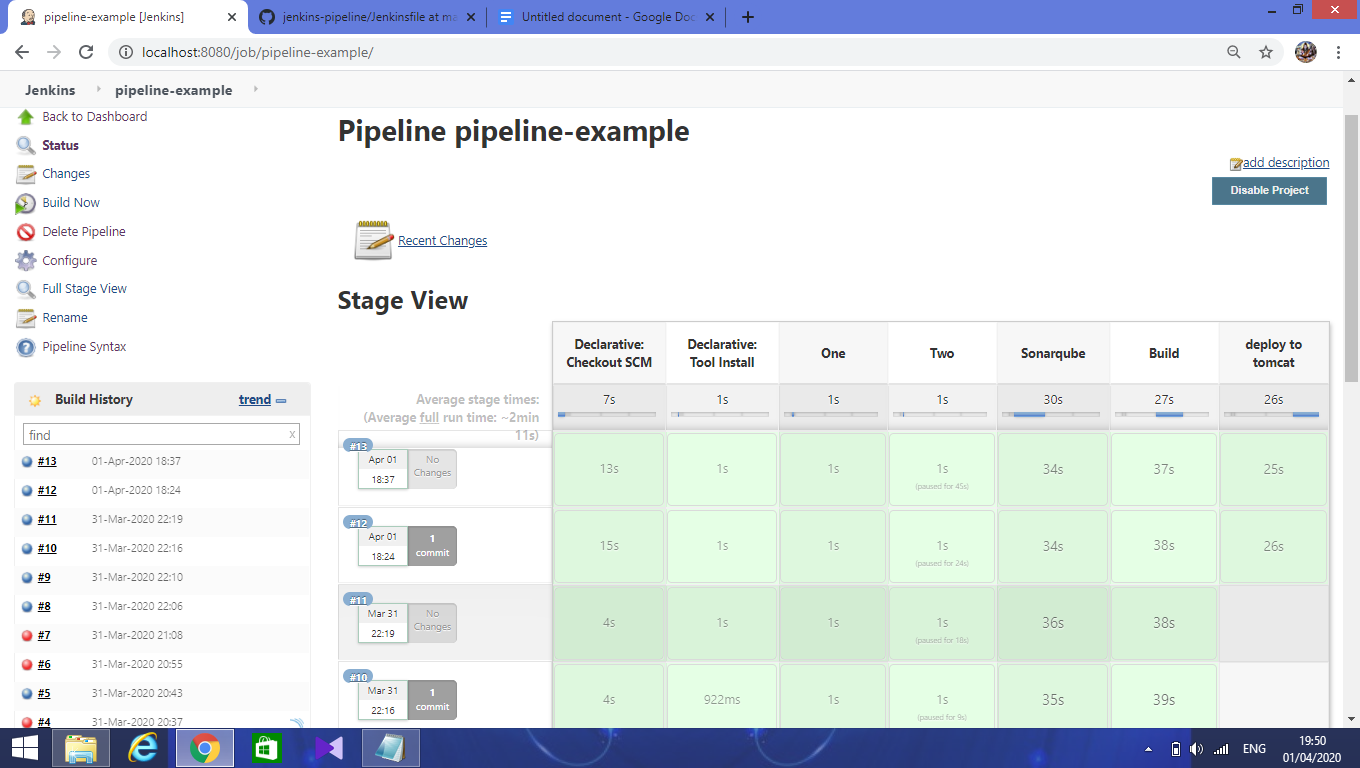
Next, in the Jenkinsfile create a stage for deploying the war file.

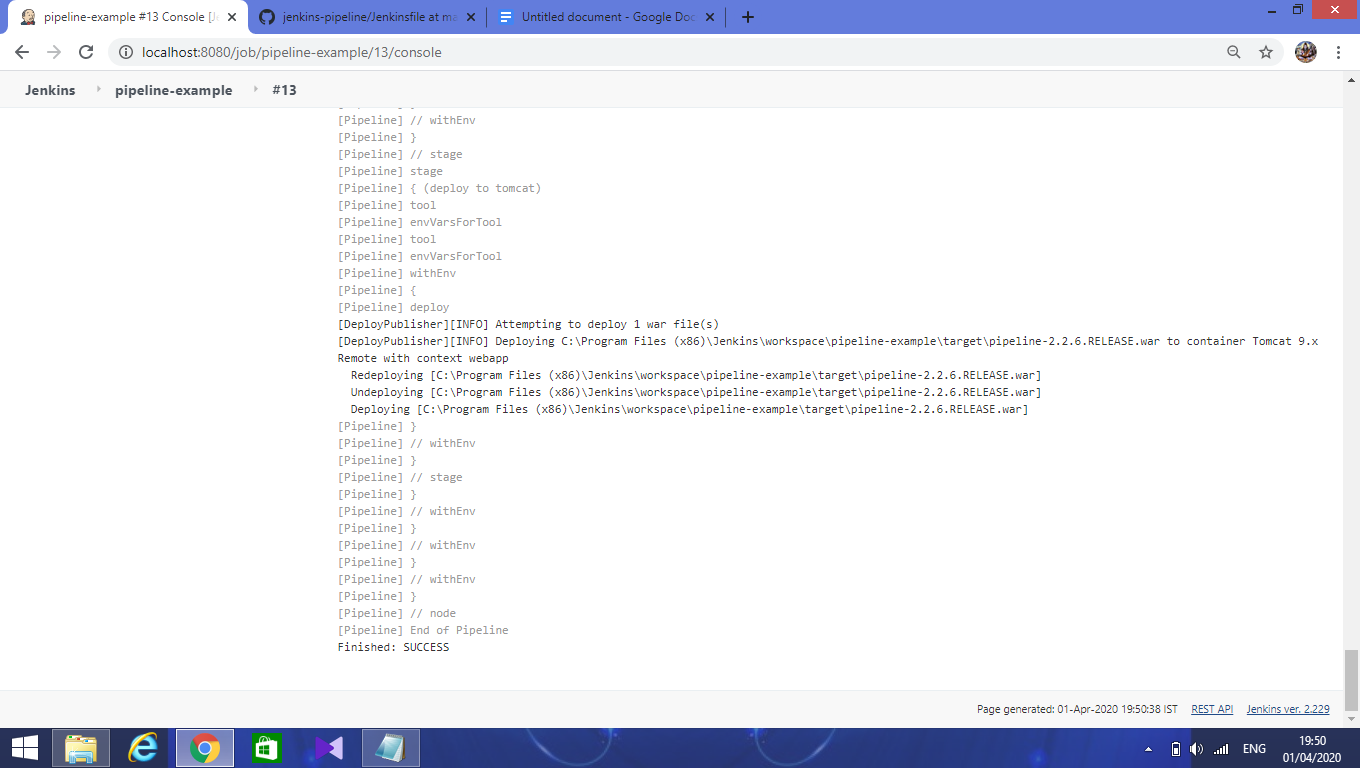
Inside that stage, under steps block paste the command.

Save it.



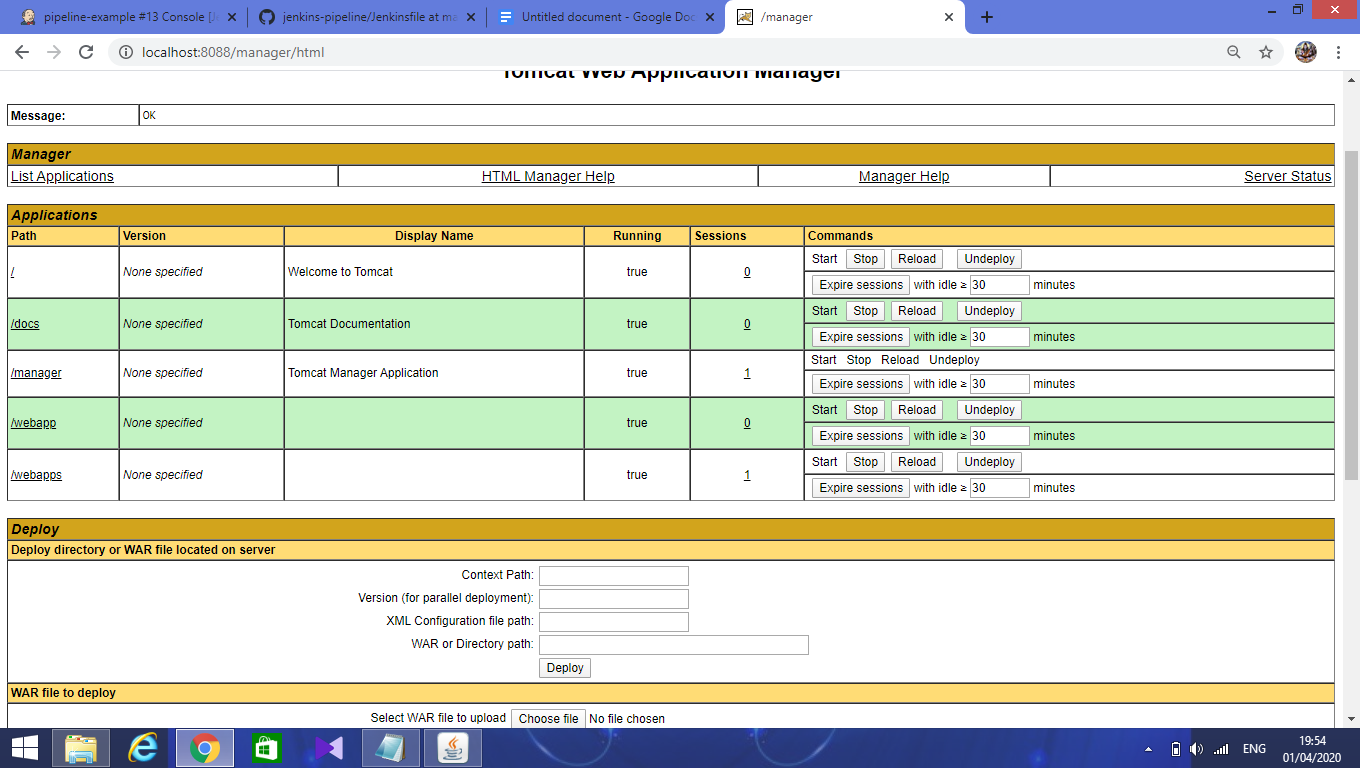
After saving, start the build in jenkins dashboard.





After the successful build goto tomcat dashboard and check war file.

Click on that file and check it.



Click on the webapp it will display home page.

