**Branch :- Computer Science and Engineering Class :- III Year**

**Subject :- C-Skill Lab-IV Sem :- VI**

**Teacher Manual**

**PRACTICAL NO. 10**

**Aim:** Setup continuous integration with Git & Jenkins.

**Theory:**

One of the basic steps of implementing CI/CD is integrating [SCM (Source Control Management)](https://www.blazemeter.com/blog/scm-version-control) tool with CI tool. This saves time and keeps project updated all the time. One of the most popular and valuable SCM tools is GitHub.

**What is GitHub?**

GitHub is a Git-based repository host, commonly used for open-source projects. GitHub enables code collaboration, hosting, and versioning.

**What is Jenkins?**

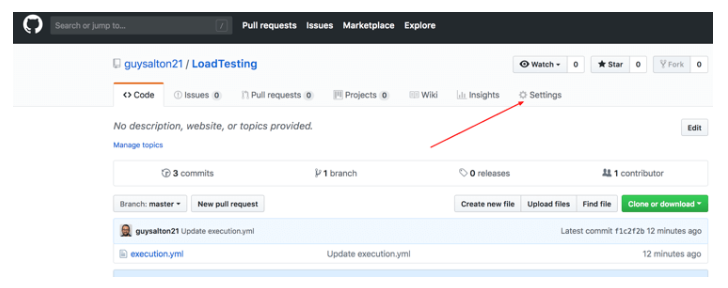
Jenkins is an open-source Continuous Integration and Continuous Deployment (CI/CD) tool for automating the software development life cycle (SDLC). With [Jenkins testing](https://www.blazemeter.com/solutions/jenkins), teams can automate the building, testing, and deploying of code.

**Why Integrate GitHub + Jenkins?**

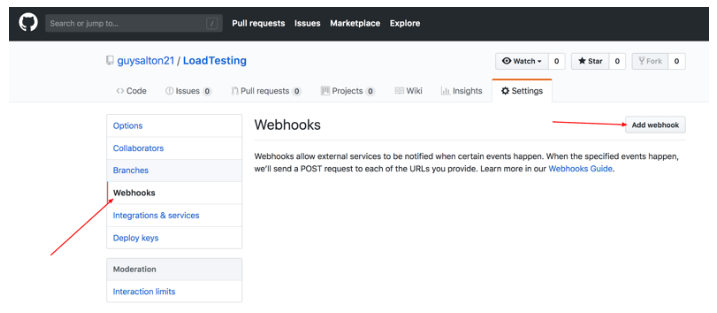
A Jenkins integration with GitHub improves the efficiency of building, testing, and deploying code.

**Steps**:

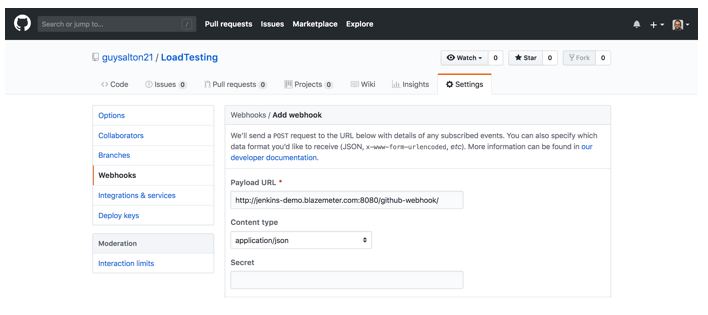
Step 1: Go to GitHub repository and click on ‘Settings’.



Step 2: Click on Webhooks and then click on ‘Add webhook’.



Step 3: In the ‘Payload URL’ field, paste your Jenkins environment URL. At the end of this URL add /github-webhook/. In the ‘Content type’ select: ‘application/json’ and leave the ‘Secret’ field empty.

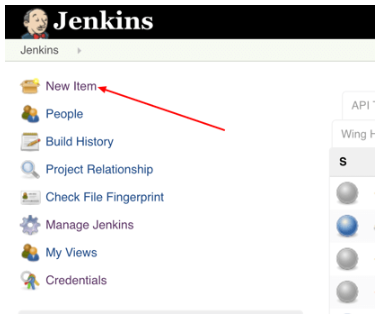


Step 4: In the page ‘Which events would you like to trigger this webhook?’ choose ‘Let me select individual events.’ Then, check ‘Pull Requests’ and ‘Pushes’. At the end of this option, make sure that the ‘Active’ option is checked and click on ‘Add webhook’.

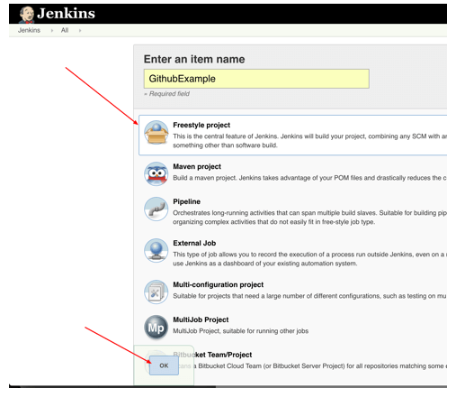


Configuring Jenkins

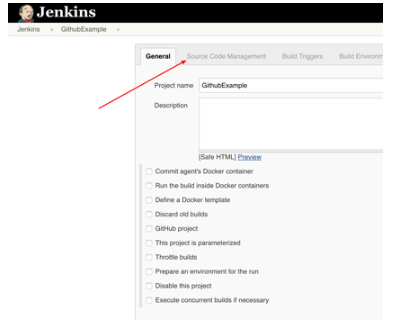
Step 5:  In Jenkins, click on ‘New Item’ to create a new project.



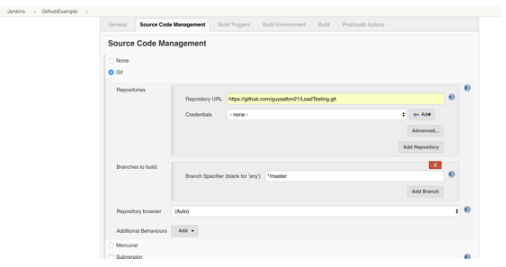
Step 6: Give your project a name, then choose ‘Freestyle project’ and finally, click on ‘OK’.



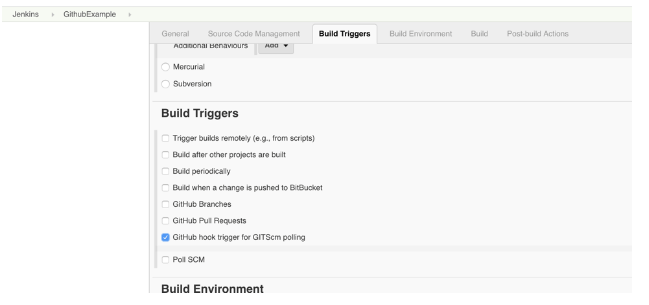
Step 7: Click on the ‘Source Code Management’ tab.



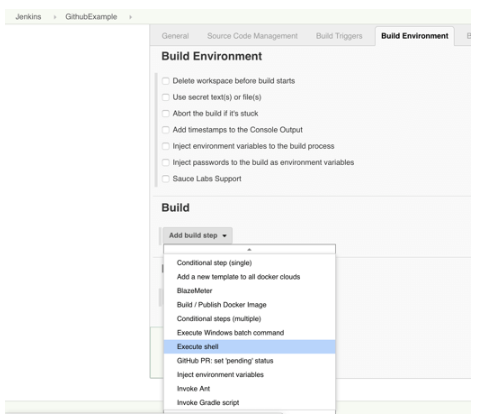
Step 8: Click on Git and paste your GitHub repository URL in the ‘Repository URL’ field.



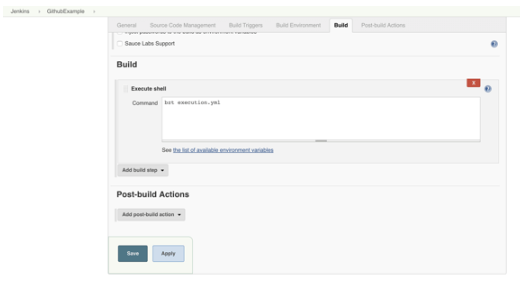
Step 9: Click on the ‘Build Triggers’ tab and then on the ‘GitHub hook trigger for GITScm polling’. Or, choose the trigger of your choice.



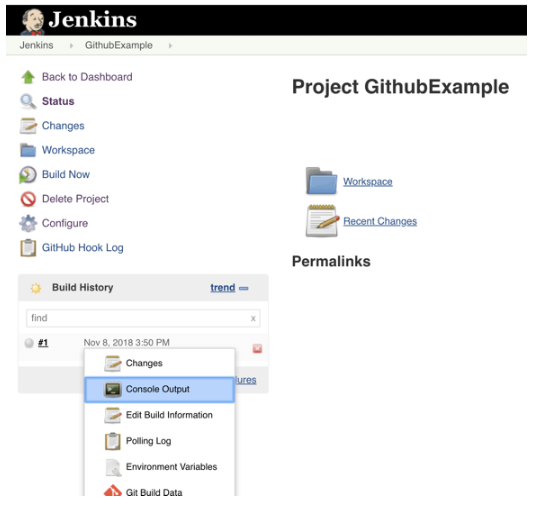
Step 10: Click on the ‘Build’ tab, then click on ‘Add build step’ and choose ‘Execute shell’.



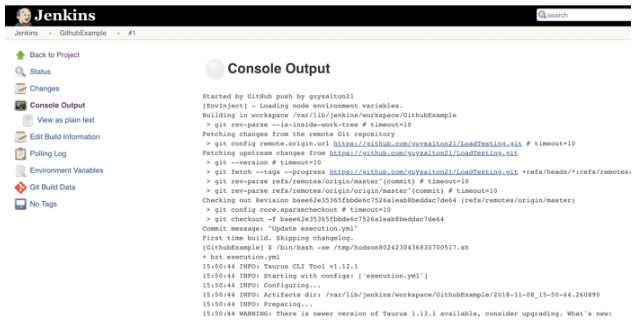
Step 11: To run a python program, simply use the command and click on ‘Save’.



Step 12: Go back to your Jenkins project and you'll see that a new job was triggered automatically from the commit we made at the previous step. Click on the little arrow next to the job and choose ‘Console Output’.



Step 13: You can see that Jenkins was able to pull the Python program and run it!



**Conclusion:** Thus, I have studied about Setup of continuous integration with Git & Jenkins.