1. Login into Git Hub and create a Repository. Follow below steps:

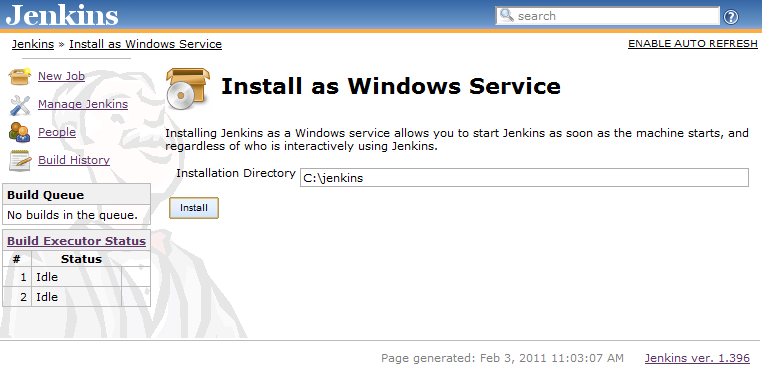
* Sign in to your account.
* Click Create Repository button.
* Choose owner for Repo.
* Give Repo a name and a short description.
* Choose public button, as private is paid.

1. Install and configure Jenkins to run as a windows service:

* Download Jenkins.war in machine from url: <http://jenkins-ci.org/>
* Launch it by executing command java –jar jenkins.war.
* Connect to Jenkins using url: http://<hostname>:8080/.
* Look for the "Install as Windows Service" link in the "Manage Jenkins" page.



* Clicking this link shows you the installation screen.



* Choose the directory where installation would be done.



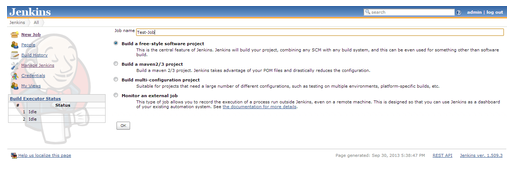
* Click yes button.

1. Install and configure GitHub Plugin on Jenkins.

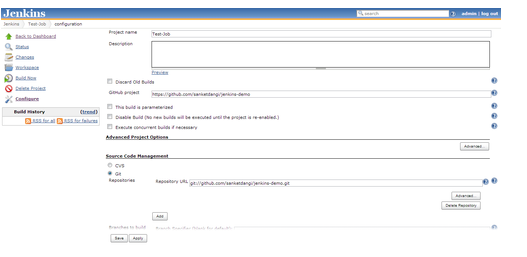
* To install GitHub plugin, navigate to your Jenkins Dashboard, click on “Manage Jenkins” and select “Manage Plugins”.
* On the plugins page, select “Available” tab and search for “GitHub plugin” under “External Site/Tool Integrations”.
* Install the plugin. This will restart Jenkins server.
* Once done, navigate back to dashboard for “Manage Jenkins” and select “Configure System”. Under that, you might observe that Git is not installed.
* Install Git Hub and provide your username and email details for user account.
* Give path of git.exe (C:\Program Files (x86)\Git\bin);

1. Create Jenkins job and configure Git Hub Repository:

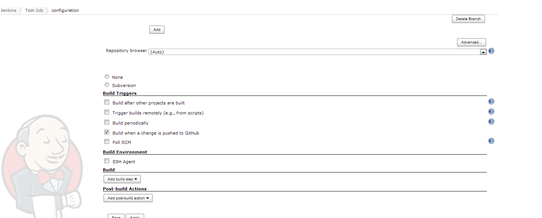
* On Jenkins dashboard, click on “New Job” and provide name to “Build a free style software project” job.



* Provide the path of GitHub repository under “GitHub Project” and provide “repository URL” for “GIT” under “Source Code Management”.



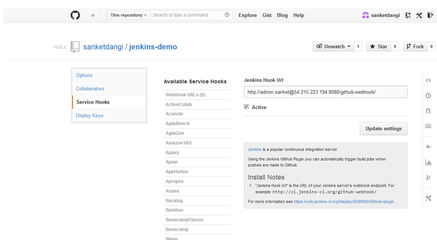
* Lower down the page, don’t forget to enable “Build when a change is pushed to GitHub”.



Now we have configured Jenkis job in such a way that whenever a change is committed to Git Hub Repo-this will trigger the build process on Jenkins.

1. Configure Git Service Hook: To trigger the build process on Jenkins, we need to configure service hook on GitHub. In order to achieve this, navigate to GitHub repository settings and configure Jenkins Hook URL for GitHub plugin.

The URL format : http://<jenkins-username>:<jenkins-password>@<Elastic-IP-Address>:8080/github-webhook/





1. Trigger a build when a change is published to GitHub:

To use this feature:

Step 1. Go to the global configuration and choose the mode in which Jenkins manages post-receive hooks in your GitHub repositories.

•Automatic Mode: in this mode, Jenkins will automatically add/remove hook URLs to GitHub based on the project configuration in the background. You'll specify GitHub user names and passwords so that Jenkins can login as you to do this, and yes, we need your password and not API token because GitHub doesn't have an API for managing post-commit hooks.

•Manual Mode: in this mode, you'll be responsible for registering the hook URLs to GitHub. Click the icon to see the URL in Jenkins that receives the post-commit POSTs.

Step 2. Open "Manage Jenkins > Configure Global Security" page and make sure that "Grant READ permissions for /github-webhook" is enabled in the "GitHub Authorization Settings" section

Step 3. Once that configuration is done, go to the project config of each job you want triggered automatically and simply check "Build when a change is pushed to GitHub" under "Build Triggers". With this, every new push to the repository automatically triggers a new build.

Note that there's only one URL and it receives all post-receive POSTs for all your repositories. The server side of this URL is smart enough to figure out which projects need to be triggered, based on the submission.

1. References:

<http://jenkins-ci.org/>

<http://sanketdangi.com/post/62740311628/integrate-jenkins-github-trigger-build-process>

<https://wiki.jenkins-ci.org/display/JENKINS/Installing+Jenkins+as+a+Windows+service>

<https://help.github.com/articles/create-a-repo>