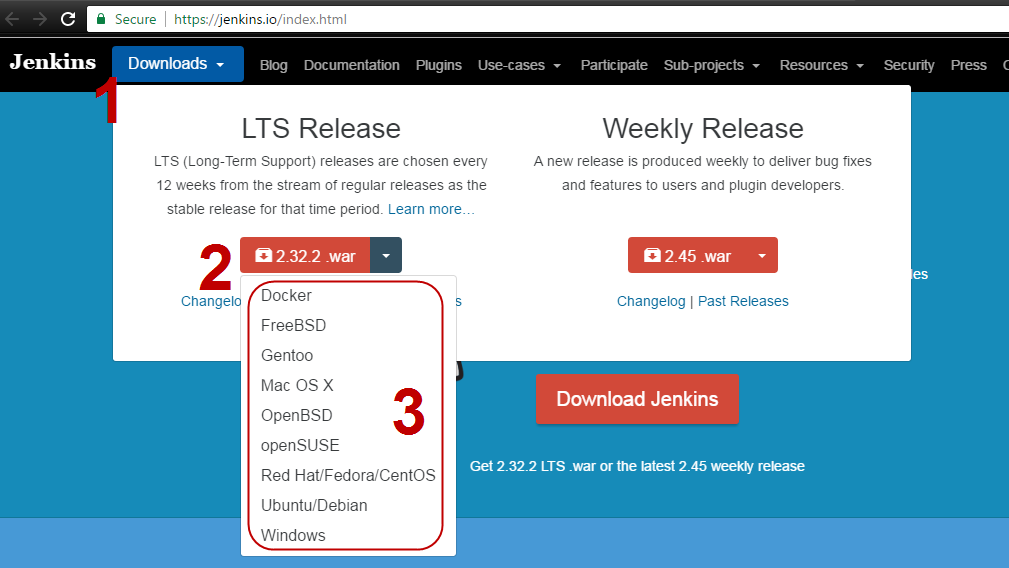
**How to create and configure a Jenkins Server**

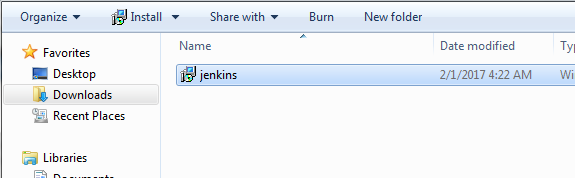
On this document you can find all the necessary information related to the preparation of a Jenkins Server.

* Installation
* Install and configure GitHub Plugins
* Enable Security, Permissions and User Accounts
* Creating Jobs
* Working with GitHub Hooks

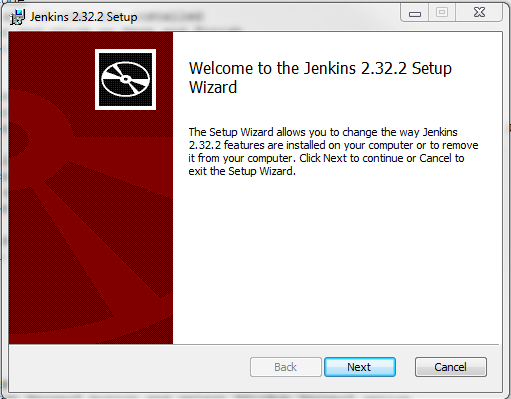
1. **Installation,** in order to install Jenkins we need follow below steps:
   1. Download latest stable build of Jenkins Server from [https://jenkins.io/](http://jenkins-ci.org/)



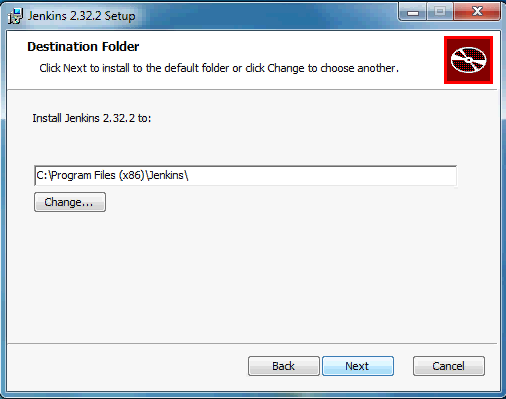
* 1. Extract the content of the installer and execute the **jenkins.msi** file



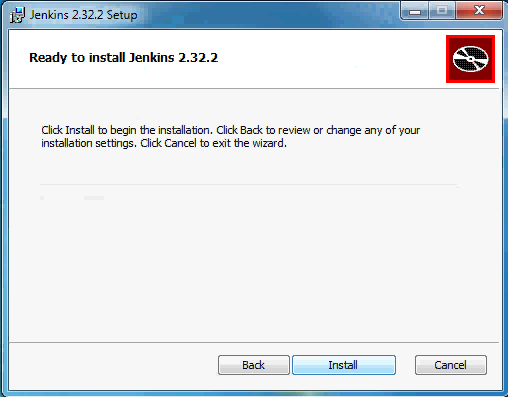
* 1. Click **Next** on Welcome window



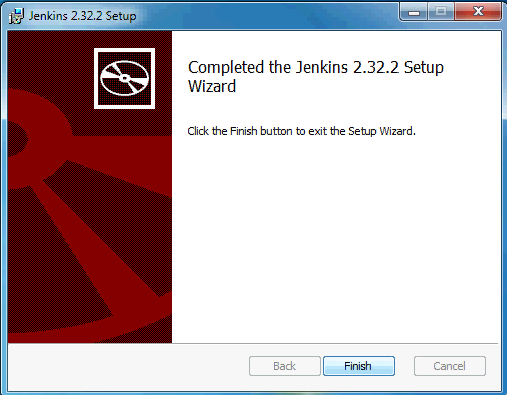
* 1. Select the installation path and click **Next**



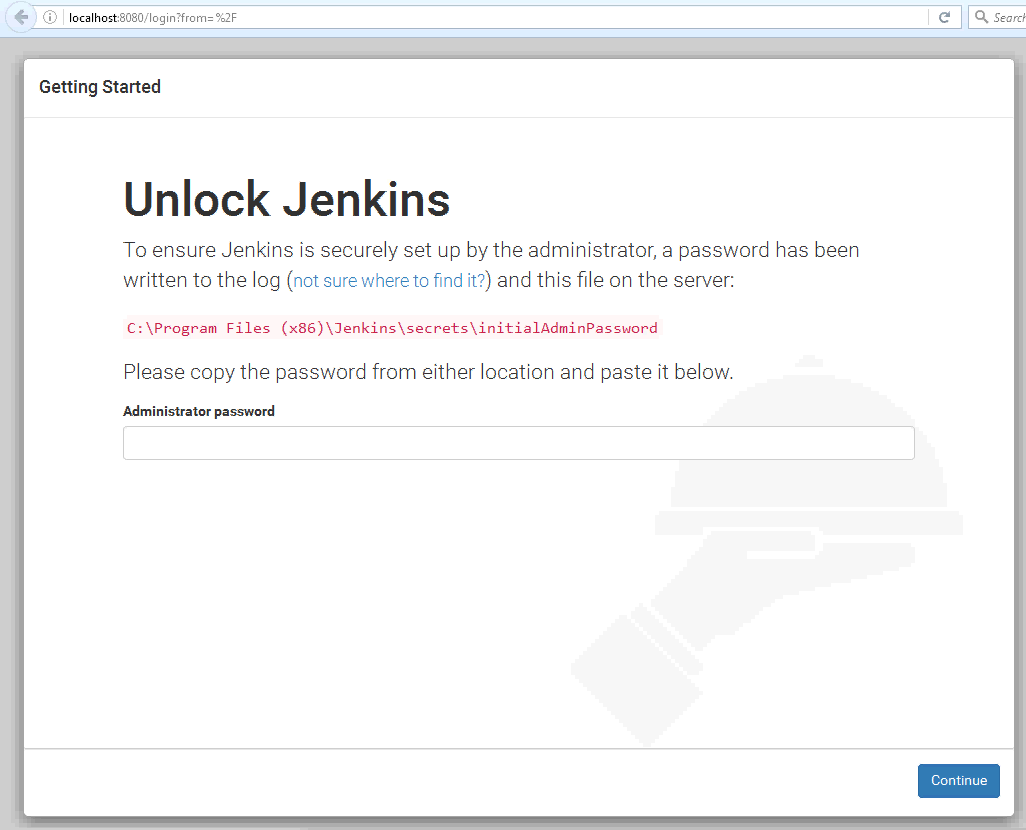
* 1. Click **Install** and wait until the installation finished



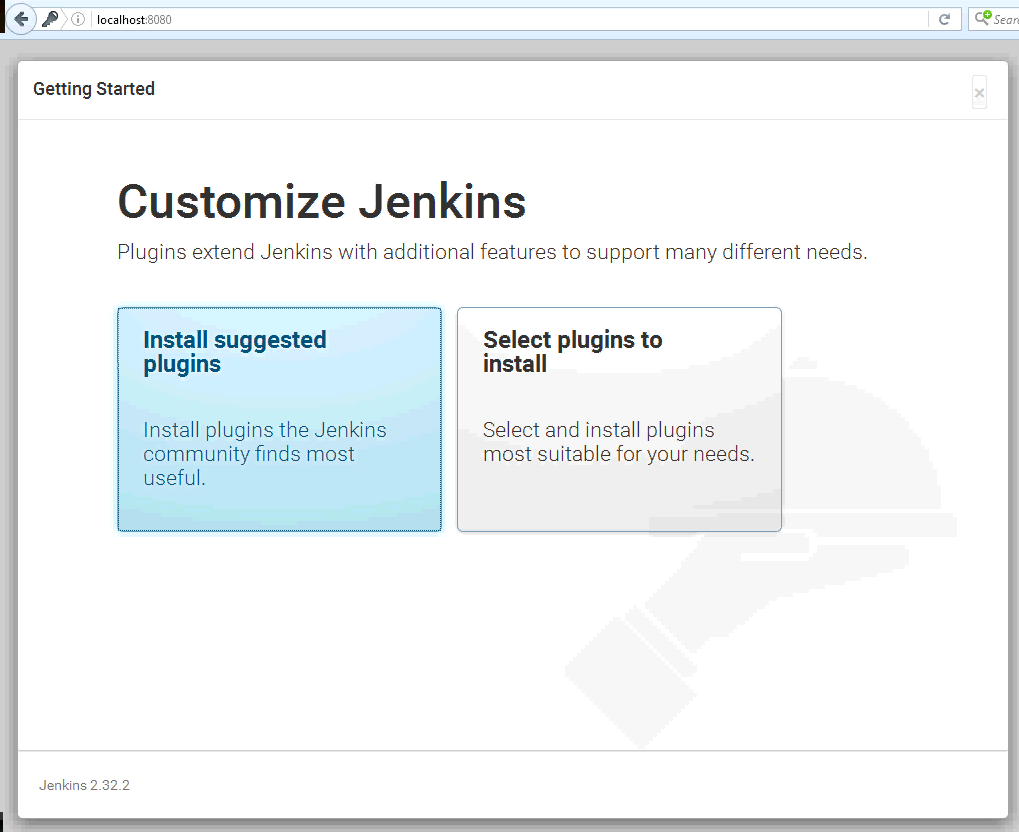
* 1. Click **Finish** button to complete the installation



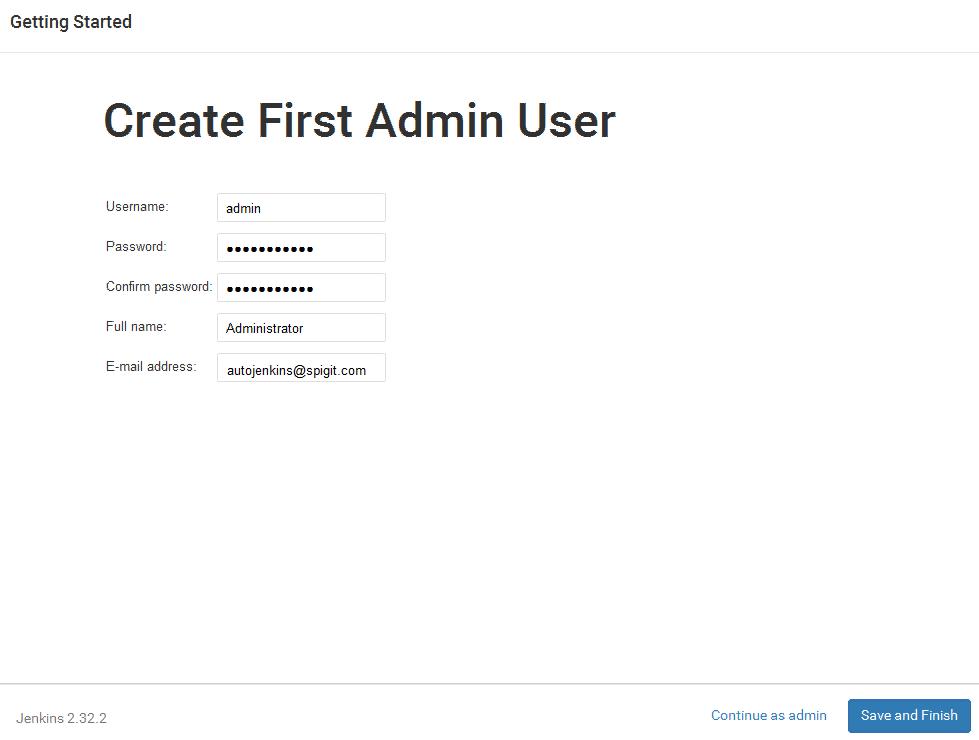
* 1. After the installation wait until the browser open Jenkins Getting Started section
  2. Follow the instructions in order to unlock Jenkins through the security code



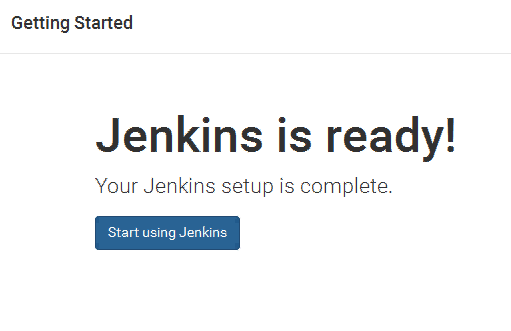
* 1. Click on **install suggested plugins** and wait until all of them are installed



* 1. Complete the information about Admin User and click on "Save and Finish" button

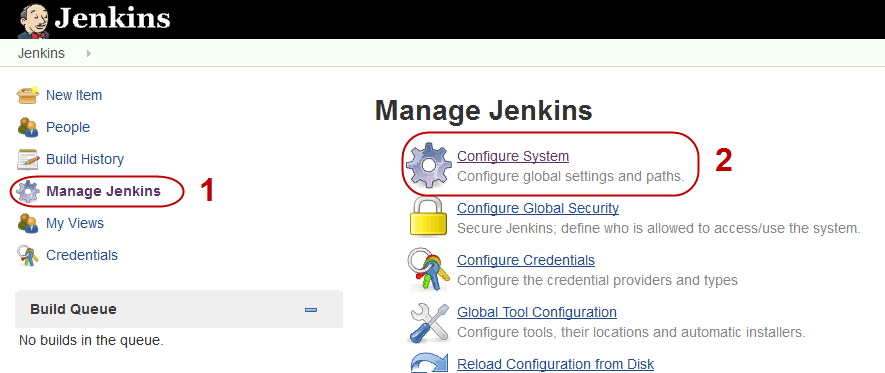


* 1. Finally click "Start using Jenkins" button

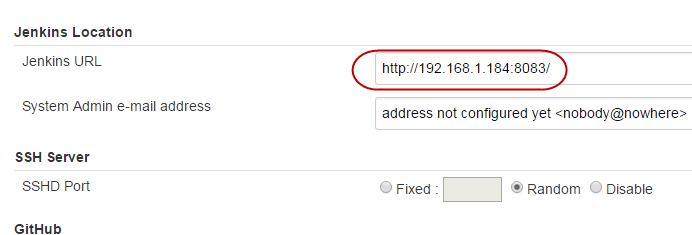


Further you need to configure the Jenkins Location URL

1. Go to Jenkins Dashboard <http://localhost:8080> and navigate to "**Manage Jenkins > Configure System**" option



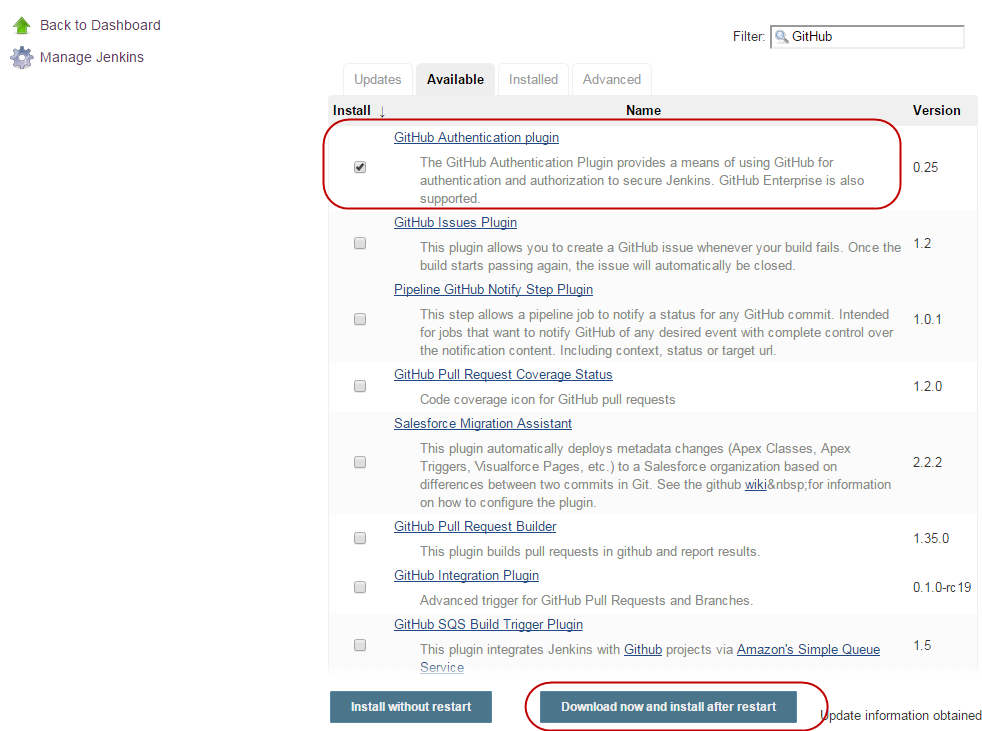
1. Go to **Jenkins Location** section and modify the **Jenkins URL** from ***localhost,*** to ***your Jenkins Server Hostname or IP Address*** (to be used by the Jenkins Nodes)



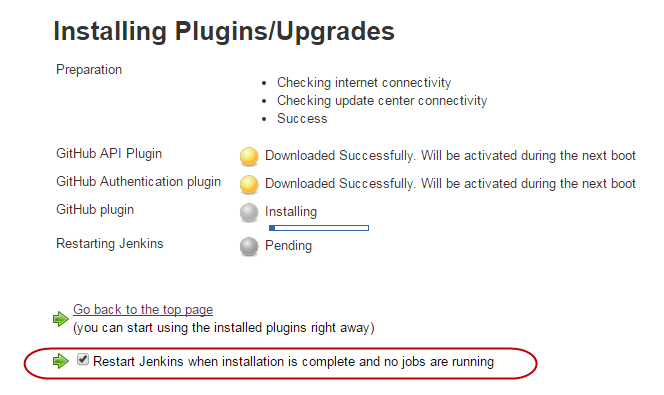
1. **Install And Configure GitHub Plugins**, in order to have access to Githubwe will need to install and configure some extra plugins.
   1. **Install GutHub Plugins**
      1. Go to Jenkins Dashboard <http://localhost:8080> and navigate to **"Manage Jenkins > Manage Plugins"** option

****

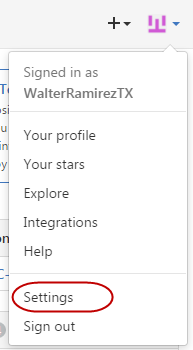
* + 1. Click on **Available** tab, search by following plugins and click on **Download now and install after restart** button
       1. GitHub Plugin
       2. GitHub API Plugin
       3. GitHub Authentication Plugin



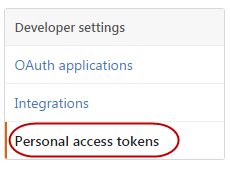
* + 1. On the update Center page, check **Restart Jenkins when installation os complete and no jobs are running** and wait until the plugins installation has been complete



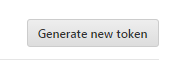
* 1. **Configure GitHub Plugin**
     1. **GitHub token**, before to configure GitHub Plugin you will need to create an Access Token from github webPage
        1. Go to **GitHub.com** and Login with you account
        2. Click on your profile on the top right corner and select "**Settings**" option



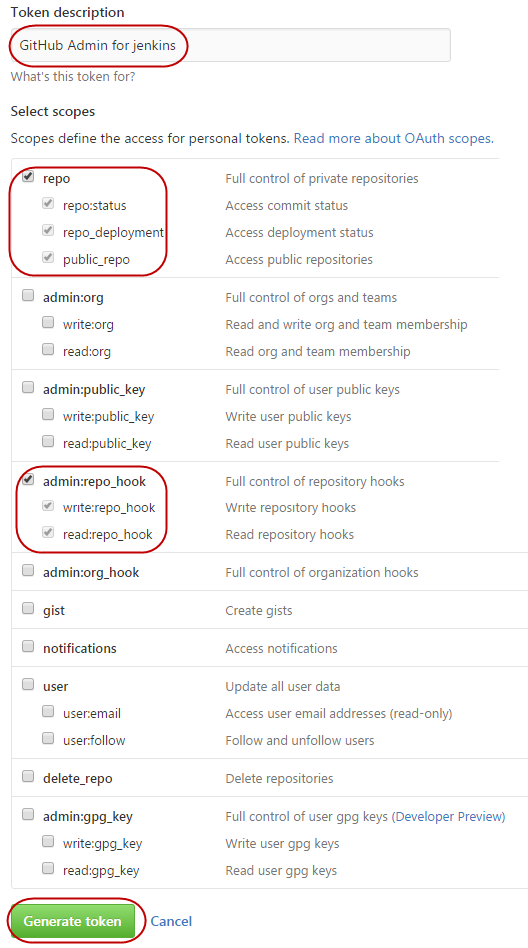
* + - 1. Click on "**Personal Access Token**" link on the left panel under the "**Developer settings**" section



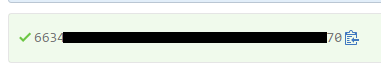
* + - 1. Click on "**Generate new token**" button



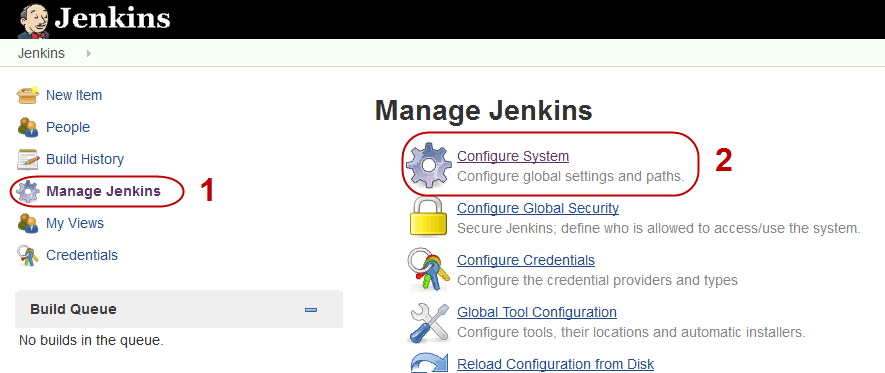
* + - 1. Re-enter your Github password if needed and continue
      2. Create a new token adding a description and selecting below scopes:
         * admin:repo\_hook
         * repo
         * repo:status
      3. Click on "Generate Token"



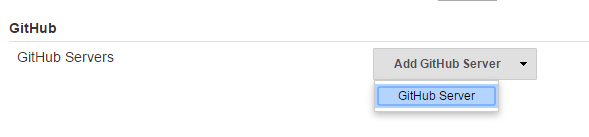
* + - 1. Save the generated token



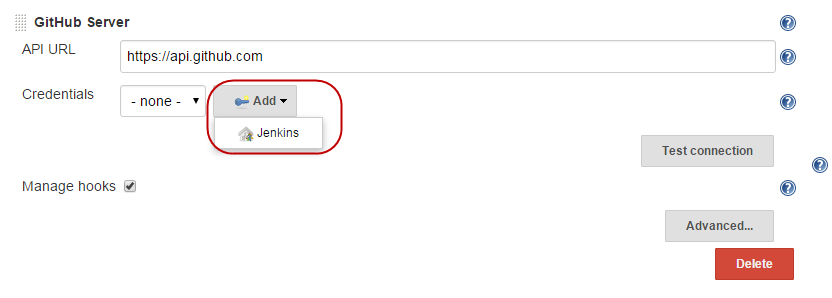
* + 1. **GitHub account**
       1. go to Jenkins Dashboard <http://localhost:8080> and navigate to **"Manage Jenkins > Configure System"** option

****

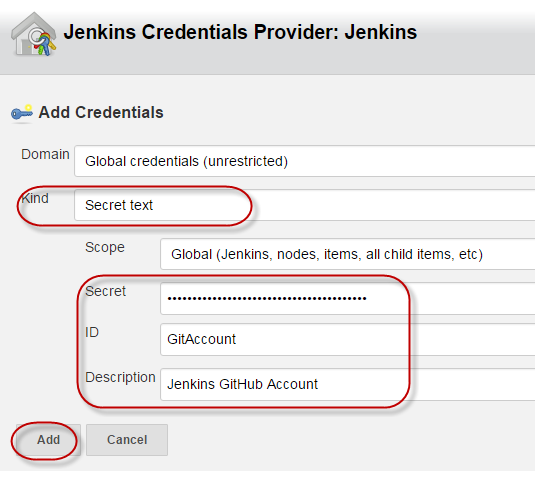
* + - 1. Under **GitHub** section click on "**Add GitHub Server**" button and select "**GitHub Server**" option



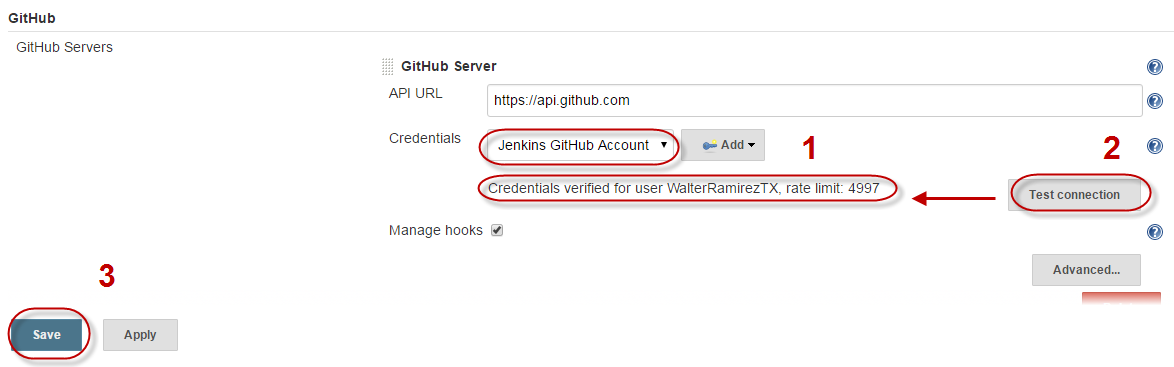
* + - 1. On **Credentials** section click on "**Add**" button and select "**Jenkins**" option



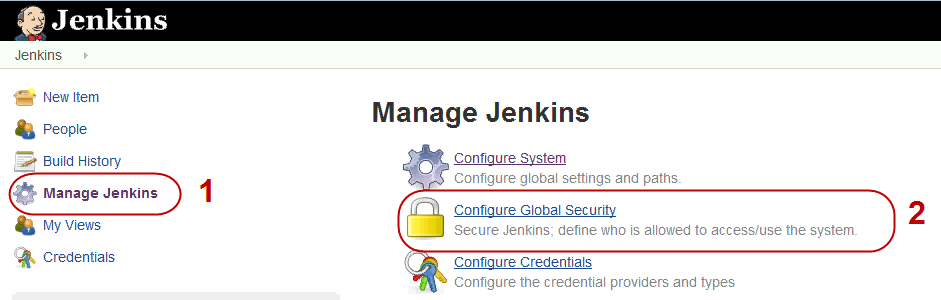
* + - 1. On the new dialog, Select "**Secret Text**" on "**Kind**" section
      2. On "Secret" textbox paste the generated token from Github
      3. Fill the ID and Description Sections and click "Add" Button



* + - 1. Back to the Github server configuration select the new credentials saved previously
      2. Click on "Test connection" button and wait until the test runs successfully
      3. Click "Save" button



1. **Enable Security, Permissions and User Accounts,** in order to add some security on our server we should create some user accounts, below the necessary steps to create the accounts and assign the account permission to each user
   1. Enable Security
      1. Go to Jenkins Dashboard <http://localhost:8080> and navigate to "**Manage Jenkins >** **Configure Global Security"** option



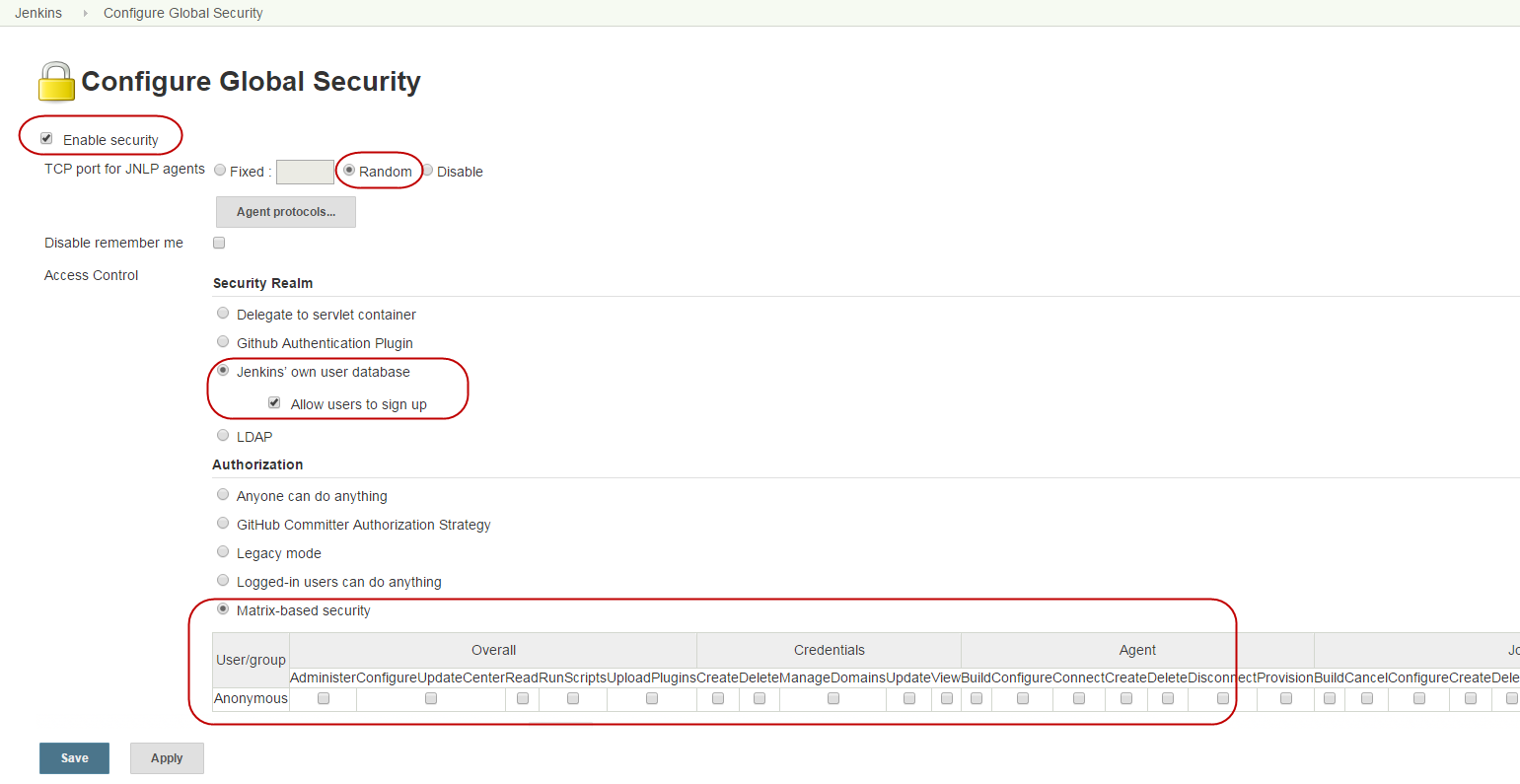
* + 1. Check **Enable Security** checkbox and configure below options:
       - *TCP port for JNLP slave agents* – **Random**
       - *Access Control***:**
         * *Security Realm*–**Jenkins’ own user database**

Leave **Allow users to sign up** checked

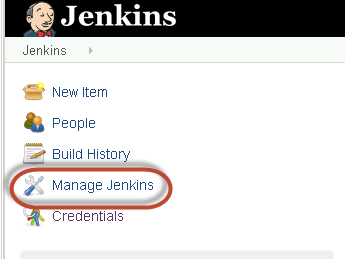
* + - * + *Authorization*- **Matrix-based security**

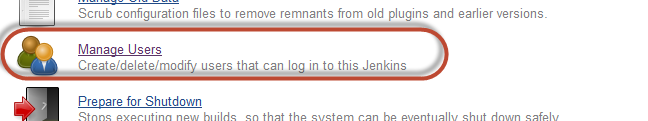
Select the options than you want to allow to the anonymous account (move the mouse over the option if you want to see a description of it)

**NOTE: Don't forget to include your administrator user on this section and give him all the necessary permissions**

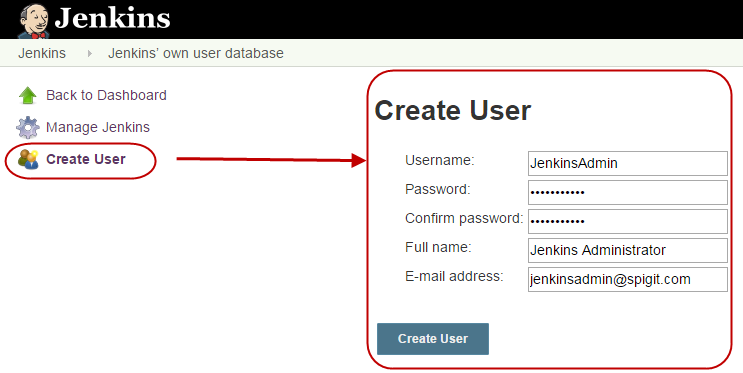


* + 1. Click **Save** button
  1. Adding users
     1. Go to Jenkins Dashboard <http://localhost:8080> and navigate to "**Manage Jenkins >** **Configure Manage Users"** option

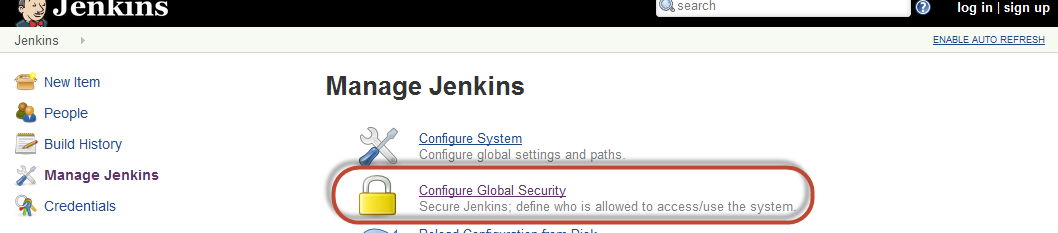




* + 1. Click **Create User** on left panel, then complete all fields and click **Sign Up**

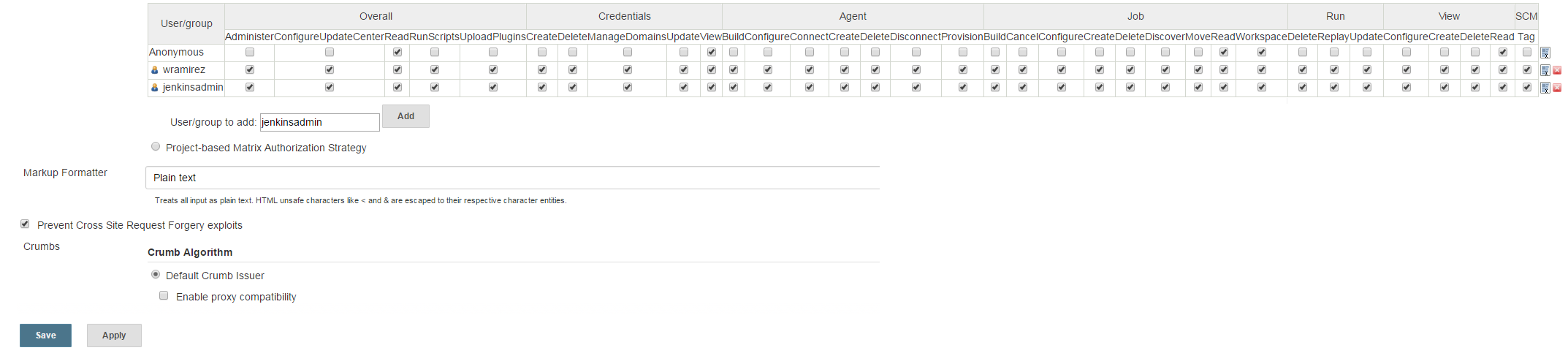


* + 1. Repeat step **iii** and **iv** to add more users
  1. Configure Permissions
     1. Go to Jenkins Dashboard <http://localhost:8080> and navigate to "**Manage Jenkins >** **Configure Global Security"** option

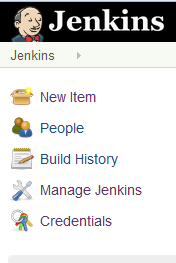


* + 1. Go to **Authorization** section and add all users that you have added
    2. Set the permissions for all new users and click **Save** button

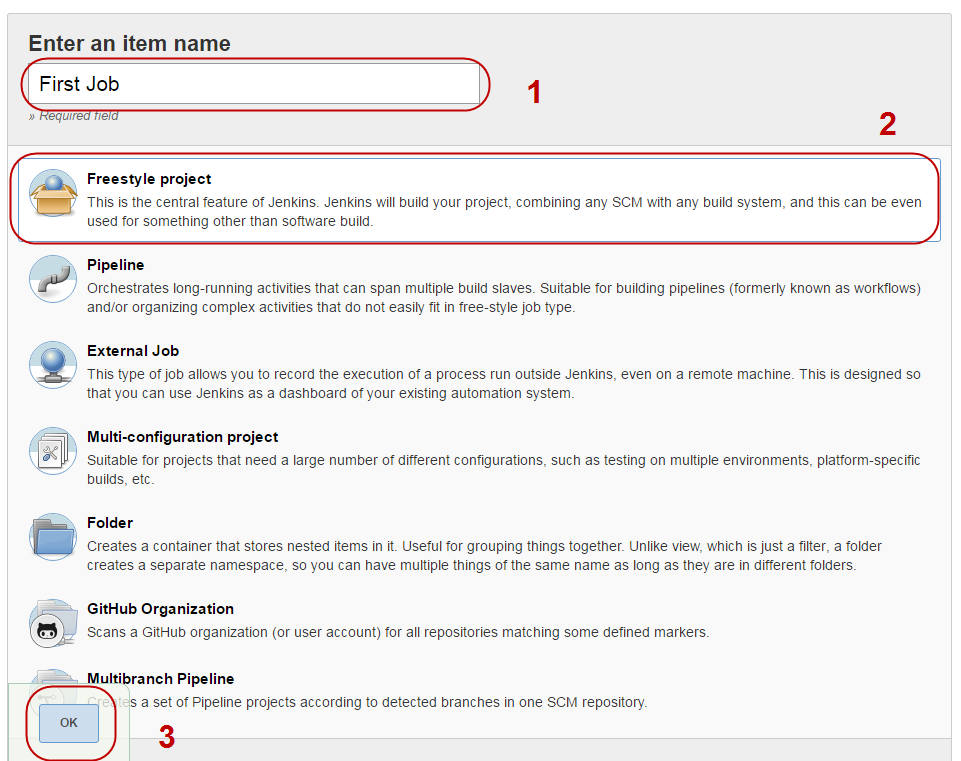
NOTE: Disable the **Administer** permissions for the Anonymous user when you have an specific account for this purpose



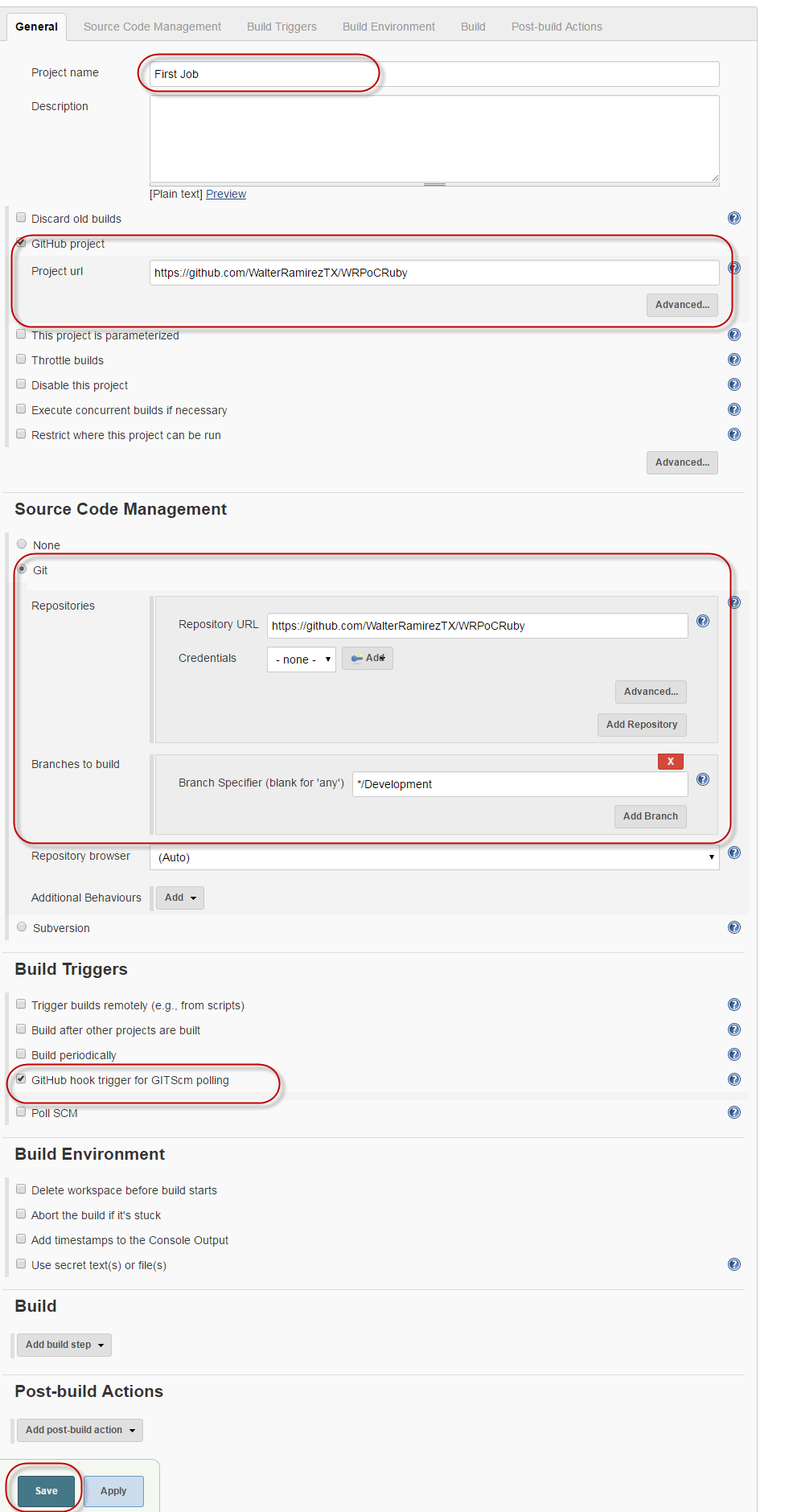
1. **Creating a new Job**, Basically we need to configure Jobs in order to execute any automated process from Jenkins on the slave VMs, below some simple steps in order to create a Jenkins job
   1. Go to Jenkins Dashboard <http://localhost:8080> and click **New Item** link on left panel



* 1. On **Item Name** textbox, set the Job Name
  2. Select the type of Item than you want to create, the most common item is **Freestyle project,** and Click **OK**

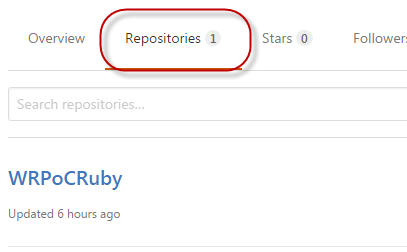
****

* 1. Set the Node Configuration following below parameters
     + **Description:** A little description of the Job
     + **GitHub project**: check this option if you want to execute a Github Project and include the project URL on the textbox
     + **Restrict where this project can be run**: If needed, check this option andset the Node Name of the VM when you want to execute the Job, i.e. ***Win7SP1x64\_001***
     + **Source Code Management** select Git option and include the Repository URL, also you can select a specific branch where the code will be downloaded on **Branch to build** section
     + **Build Triggers**, if you want to execute the job everytime when an event is raised from GitHub, e.g. Push, you need to check **GitHub hook trigger for GITScm polling** checkbox, and configure your GitHub project to raise the Hook from your Git Project, see **Working With GitHub Hooks**
     + **Build,** modify this sectionIf you want to execute actions during Build and/or Post Build

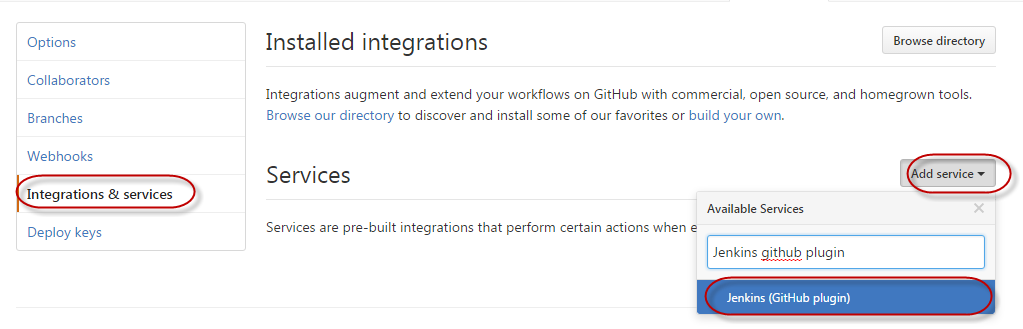
****

* 1. Click **Save** Button

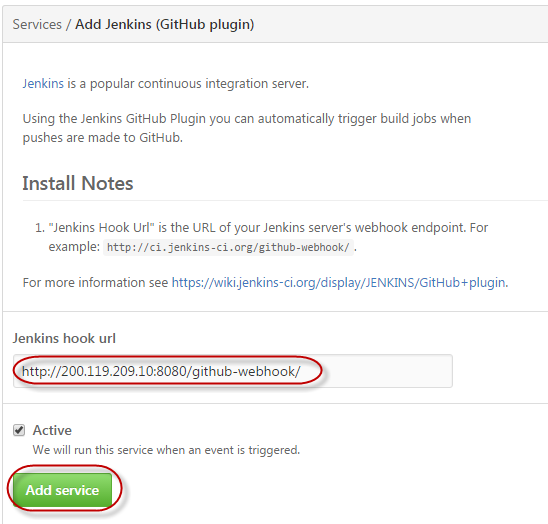
1. **Working with GitHub Hooks**, In order to connect Github with Jenkins, you will need to configure a new Hook on your Github project following below steps:
   1. Go to GitHub.com and Login with you account
   2. Search for the project that you want to create the hook on the **Repository** tab, and click on it



* 1. From the project view, go to **settings** tab
  2. From left menu click on **Integration & services** option, click on **Add Service** button and serch for **Jenkins (GitHub plugin)**



* 1. On the Service window, set the **Jenkins hook url** and click on **Add service** button



Note: if you want to use the IP address for your Jenkins server, make sure to use the **public IP**.

If everything is fine, a new WebHook will be created on the next Push into your repository, further, you can configure it from **Webhooks** section on the left panel.