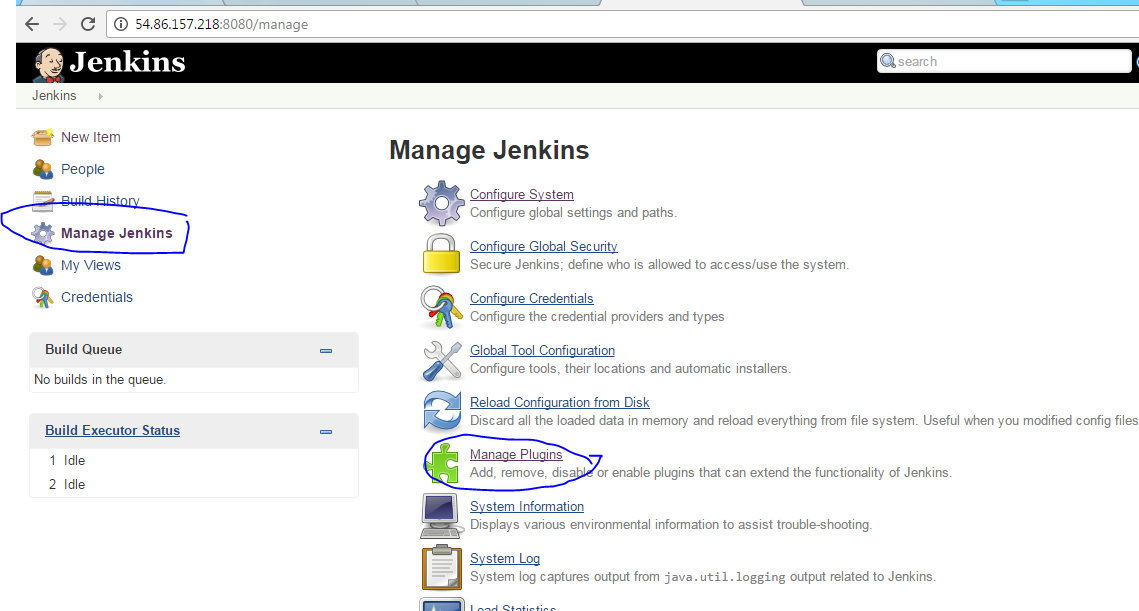
Go to manage Jenkins, followed by manage plugins.

Now select the available tab, then search for and install the **Github plugin**.



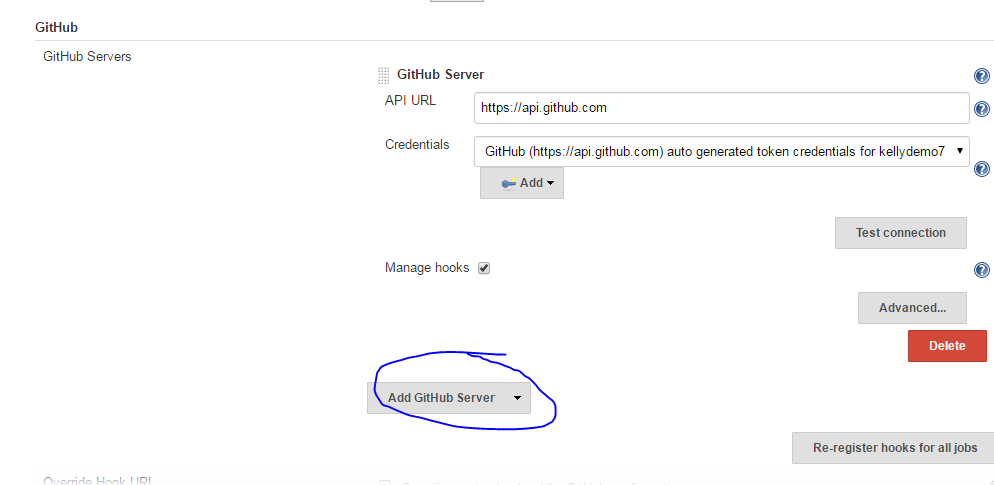
As always, you’ll need to wait for **Jenkins** to restart, but it’s worth it, as this plugin is SUPER useful for setting up web hooks.

**Configure Webhook Authentication**

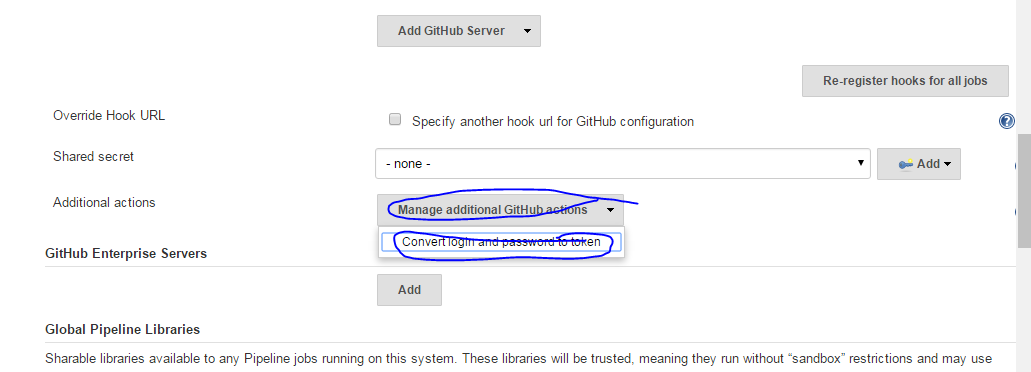
Alright, let’s go to **Manage Jenkins** again, but this time click **Configure System**.

Scroll down to the **Github** section which is usually around the middle of the page.

Right under the **Add GitHub Server**, clicked the **Advanced** button.

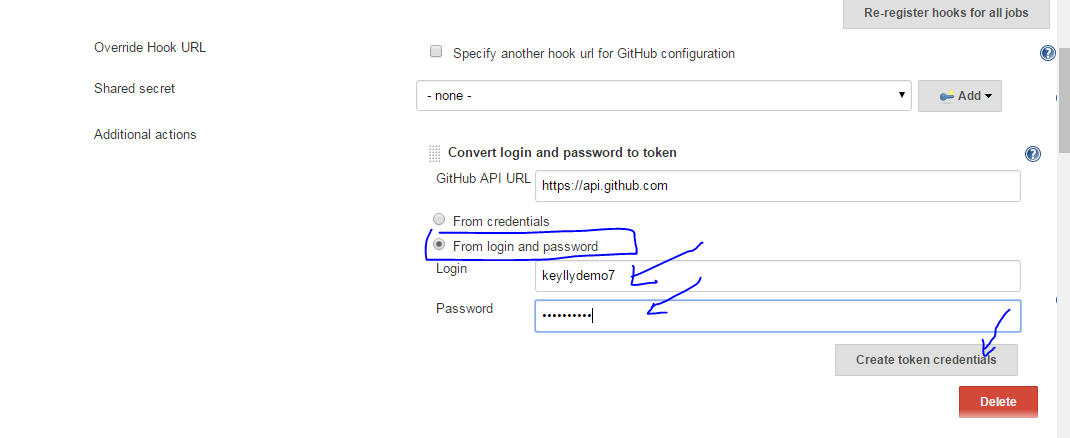


Now click **Manage additional Github Actions**, then select **Convert Login and Password to Token**.



Next select **From Login and Password**.

Type your Github username and password, followed by **Create Token Credentials**.



Well done! If you’re wondering, Jenkins just created a personal access token on Github, which will be used for authenticating with Github! We just have one more step left on this page.

Click **Add Github Server**, then click the **Credentials** drop down. Go ahead and select the token that we just generated, then hit **Save** at the bottom of the page.

**Configure Job for Webhooks**

Now head back to the configuration page for our automated test job.  
Near the top of the page, check the box named **Github Project**. Now copy and paste the URL of your repository into the project URL box.  
Next scroll down to the **Build Triggers** section again, but this time select **Build when a change is pushed to Github**.

Before you hit save though, we have one final step left.

**Configure Github for Webhooks**

Let’s switch from Jenkins to Github.com for moment.

We need to go to the **Settings** for our Github repository, go ahead and click the **Settings** button now.

Next, Click **WebHooks and Services**

Now click the **Add service** button.

Type **Jenkins**, then select the **Jenkins (Github Plugin)**. Now we just need to paste in the URL to our Jenkins server’s webhook.  
By default, it’s going to be http:// followed by the name or IP of your Jenkins server, slash github dash webhook. Make sure active is selected, then hit **Add Service**.

If you’d like, you can now hit **edit** and **test** the service, which we recommend, but it’s completely optional.

Now switch back to Jenkins and save our job! From now on, anytime we create a job using Github, and select **Build when a change is pushed to Github**, a Webhook will automatically be created for us!