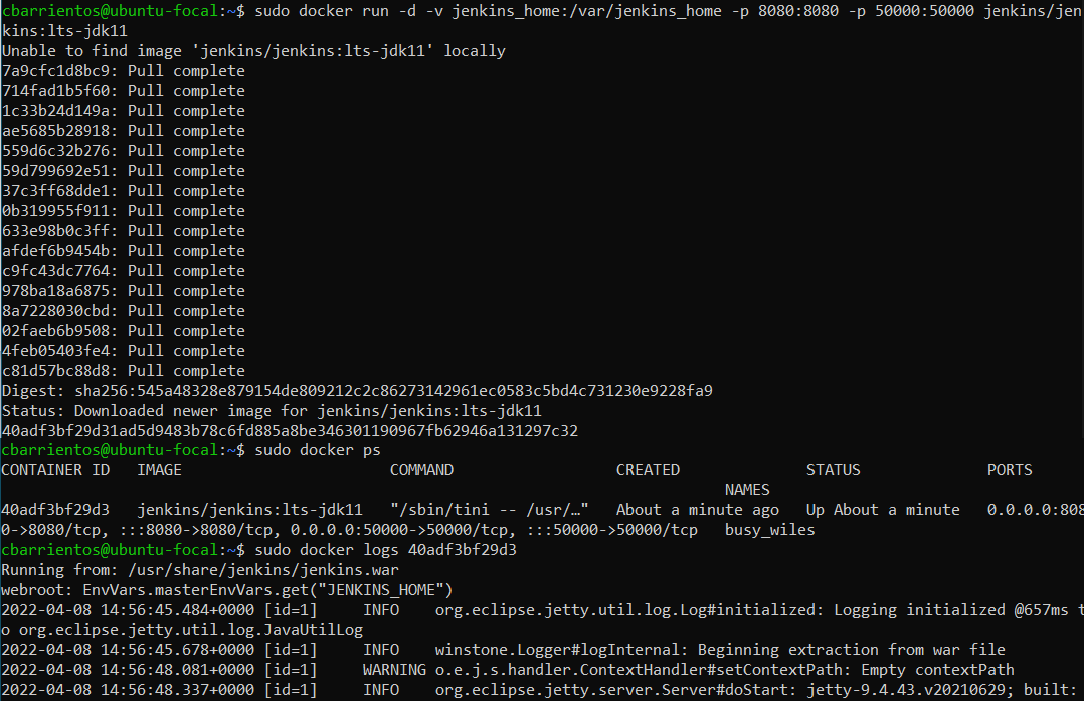
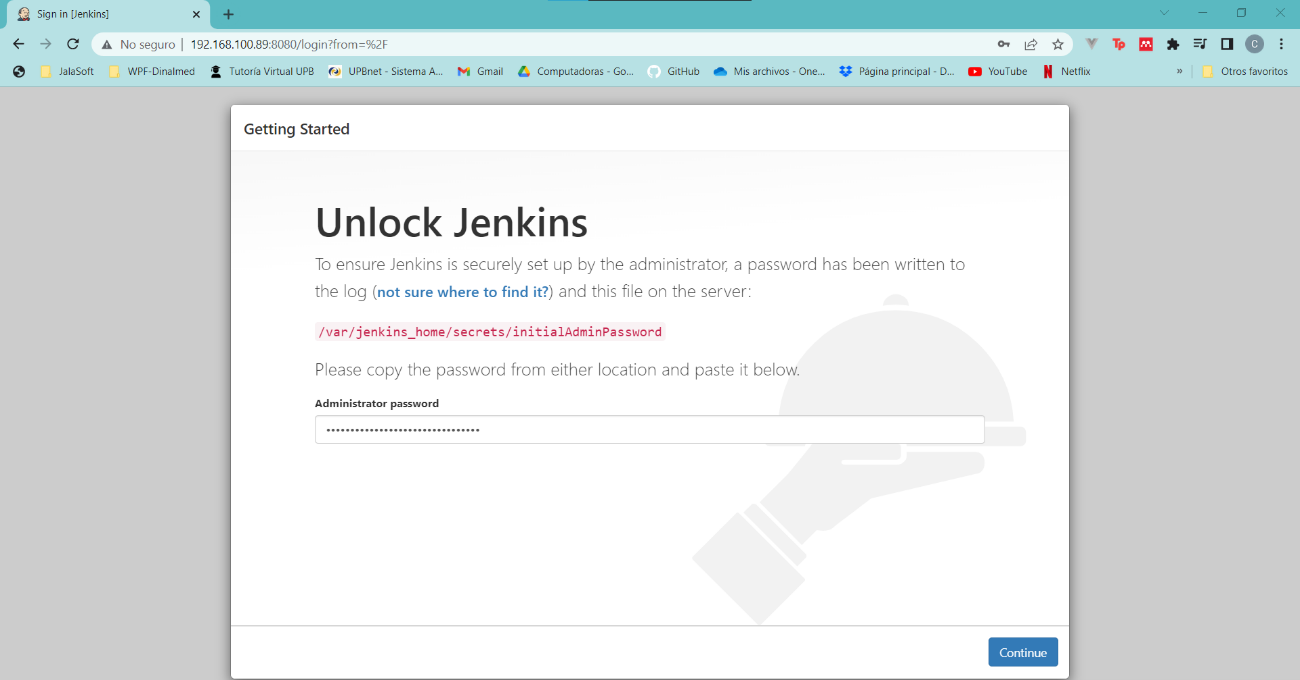
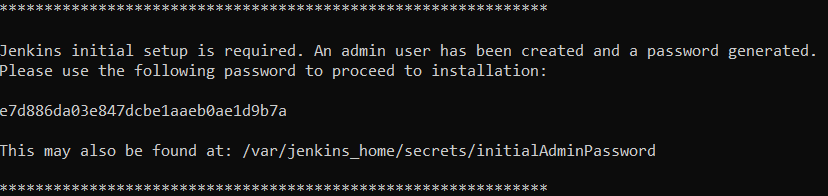
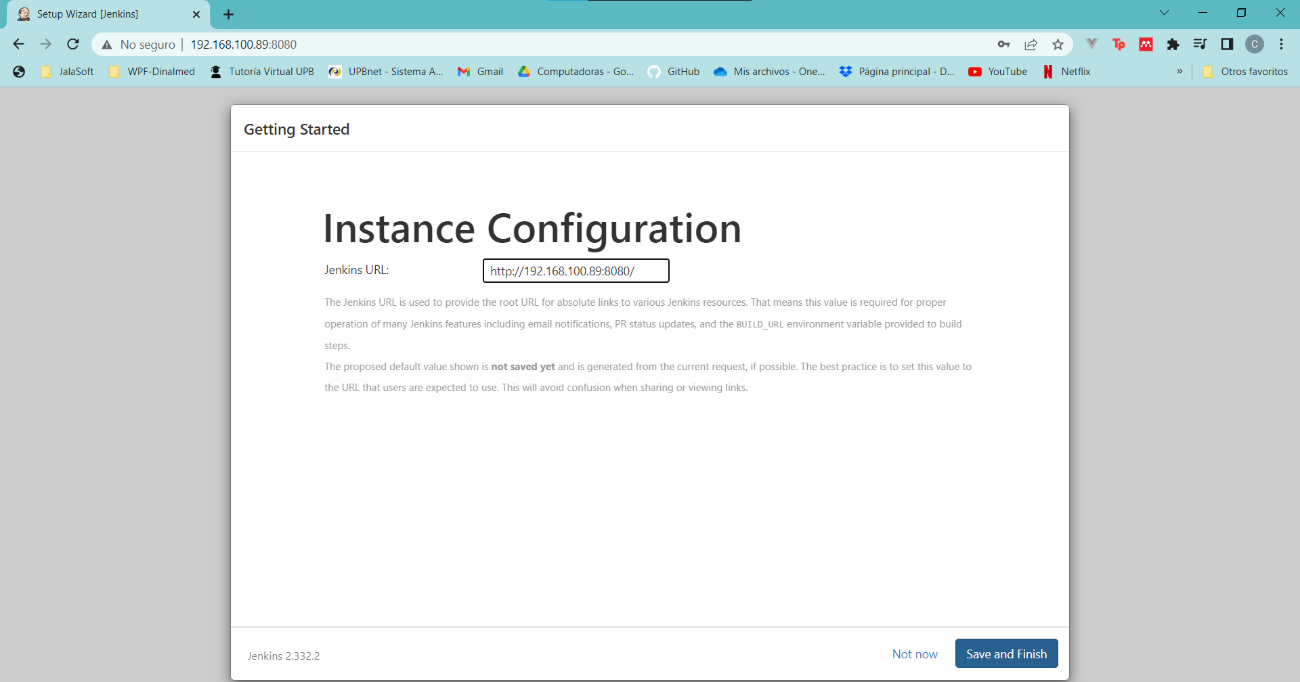
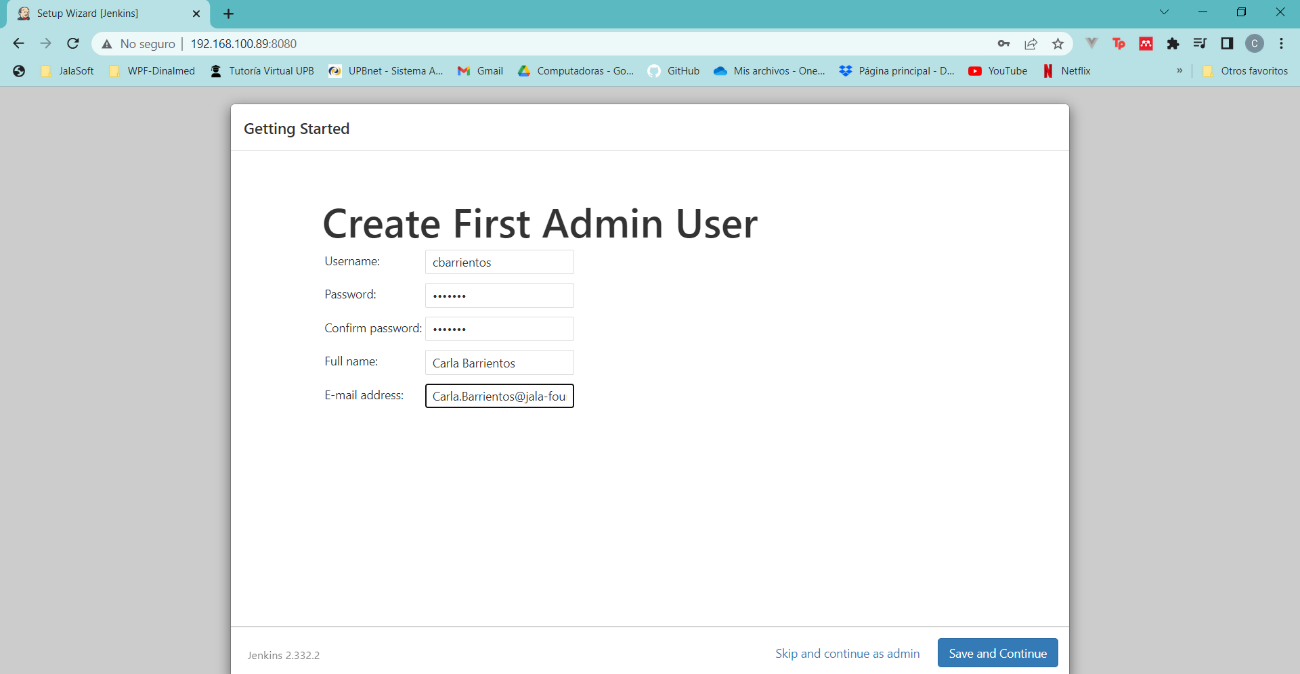
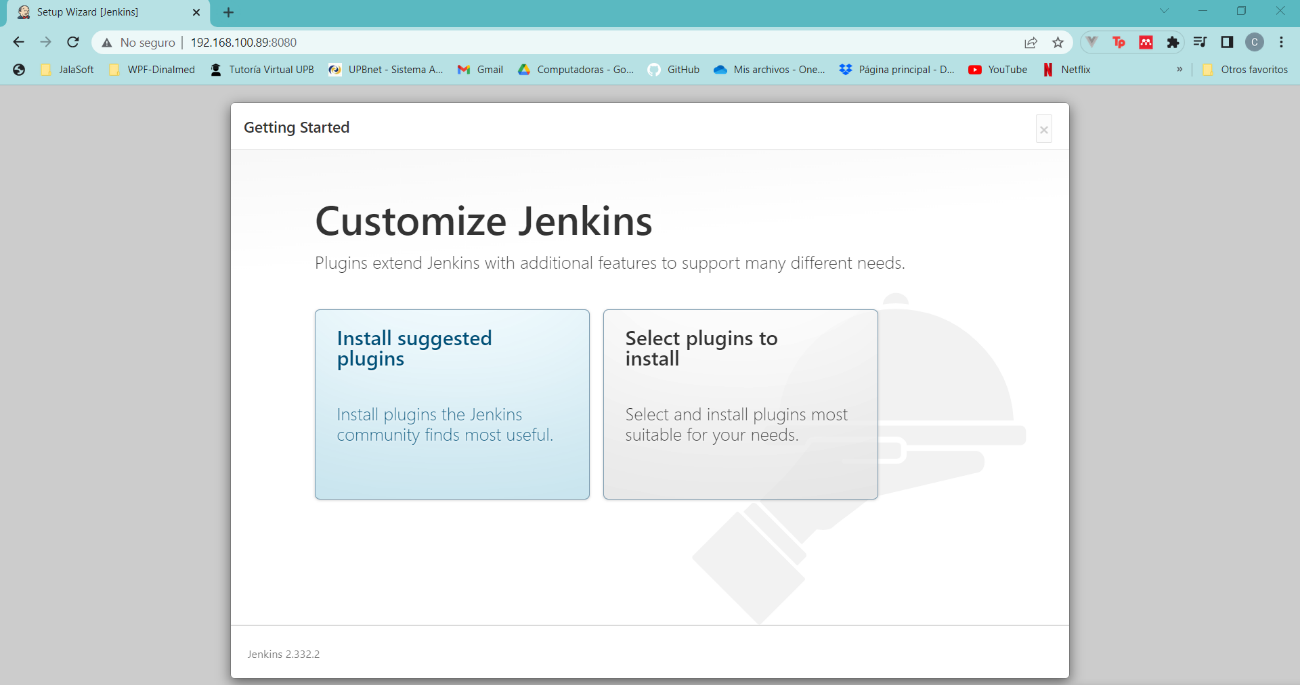
**CI Configuration on Jenkins**

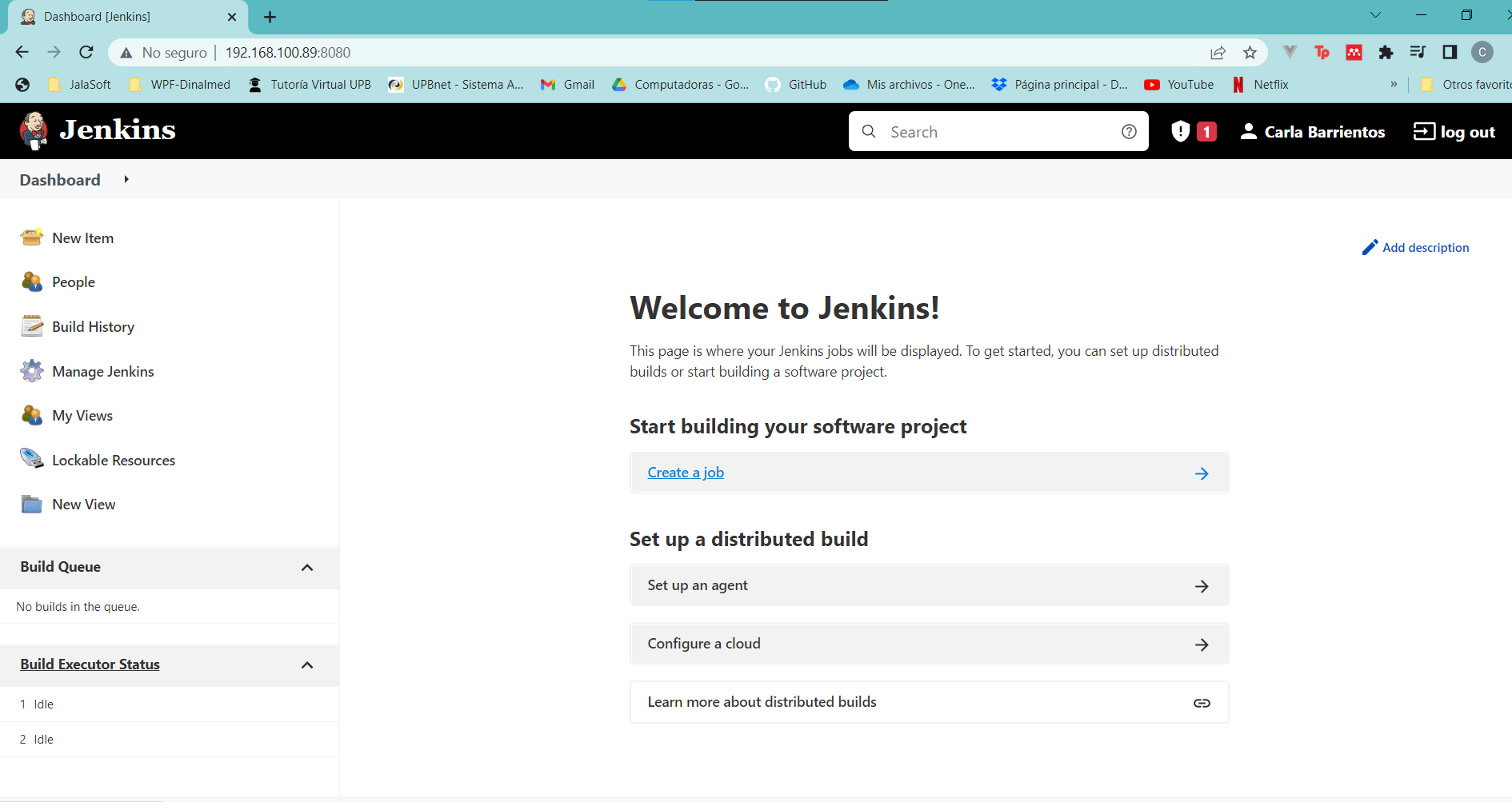
First, we need to start Jenkins in a Docker container, to do so we run the commands shown in the next image:

sudo docker run -d -v jenkins\_home:/var/jenkins\_home -p 8080:8080 -p 50000:50000 jenkins/jenkins:lts-jdk11

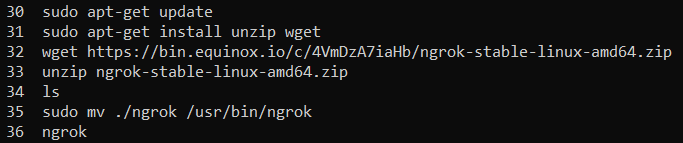


After running this container, we can access the logs to get the administrator password to start configuring Jenkins on the web browser: 

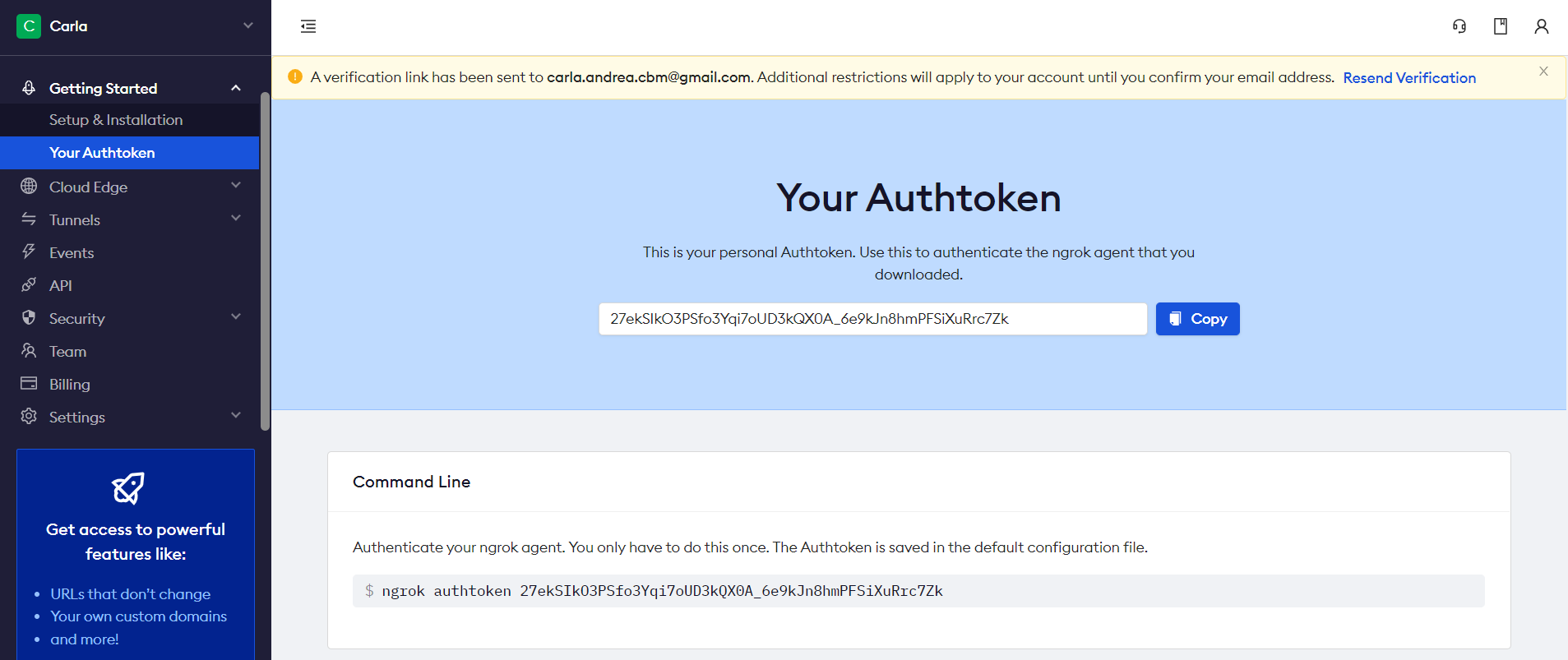
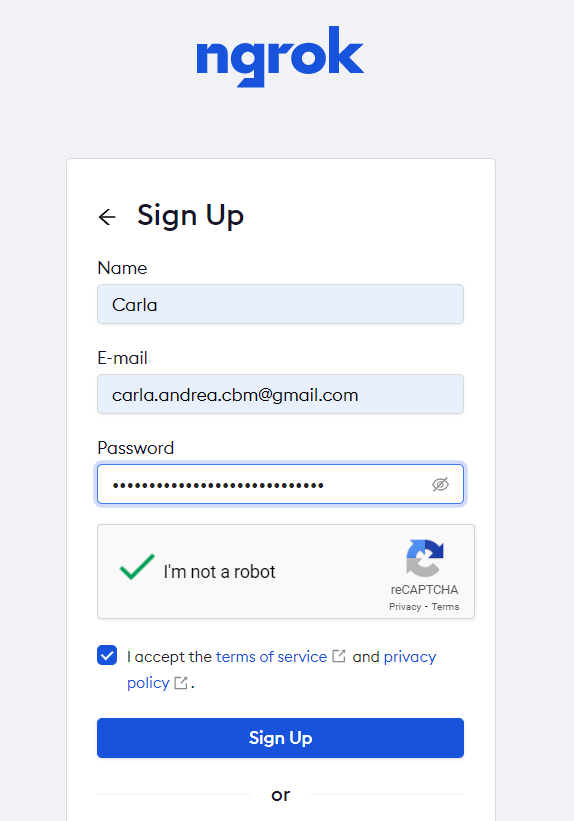
Next we follow the configuration menu, we select to install suggested plugins, create the first admin userand check that the URL of the Jenkins instance is correct:

Then we select the option to create a new job, in this case we named it test-pipeline and selected the *Multibranch Pipeline* option:

To configure our GitHub repo with our Jenkins server we need to configure webhooks, to do so first we need to install ngrok to use our server locally with an URL provided by ngrok.

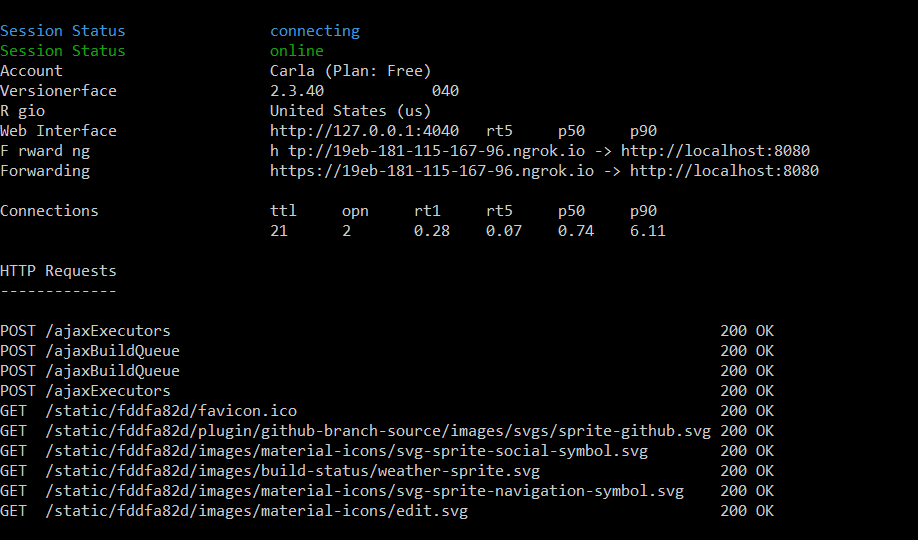


Then we need to register in ngrok to get our auth token and confirm the confirmation email:



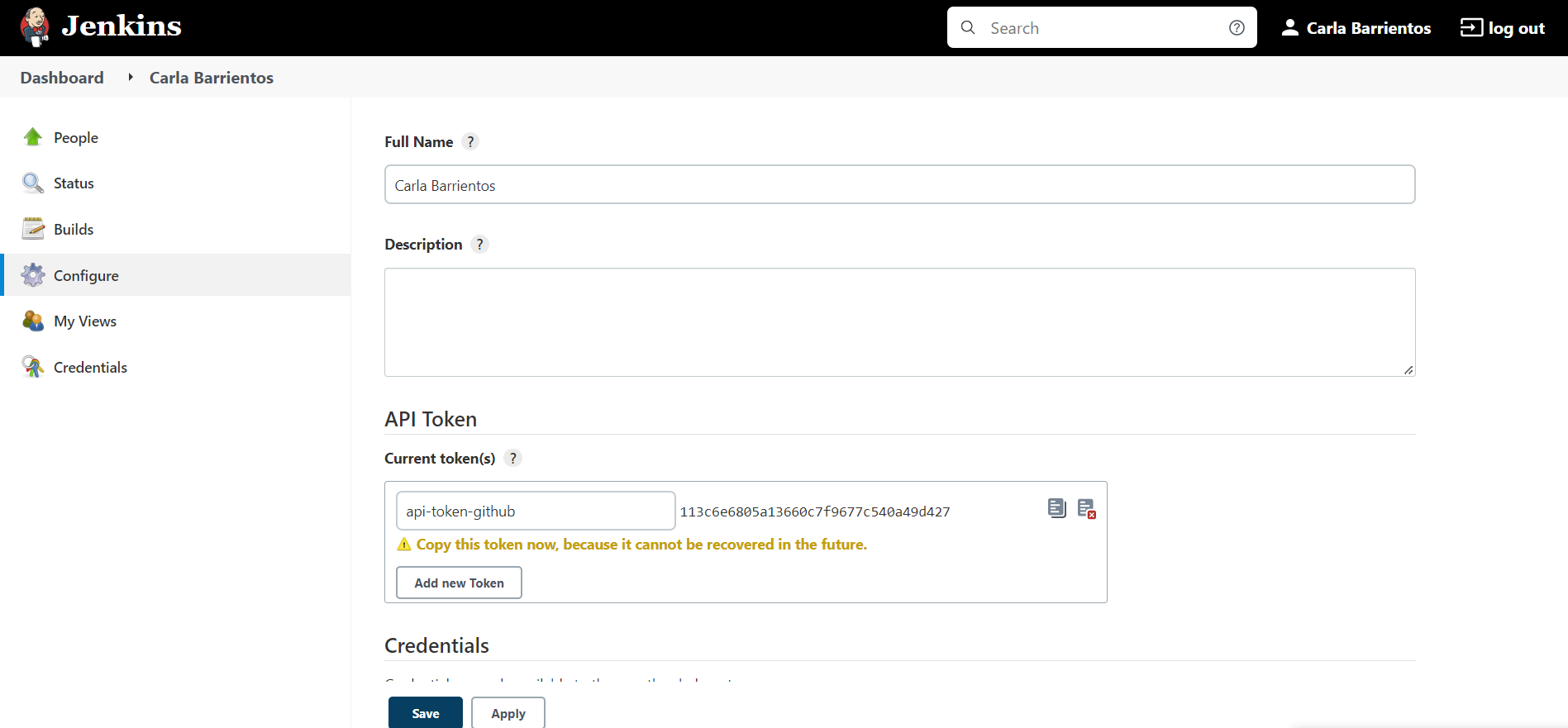
We run the command shown to validate the authorization token and then we excecute the next command:

$ ngrok http 8080

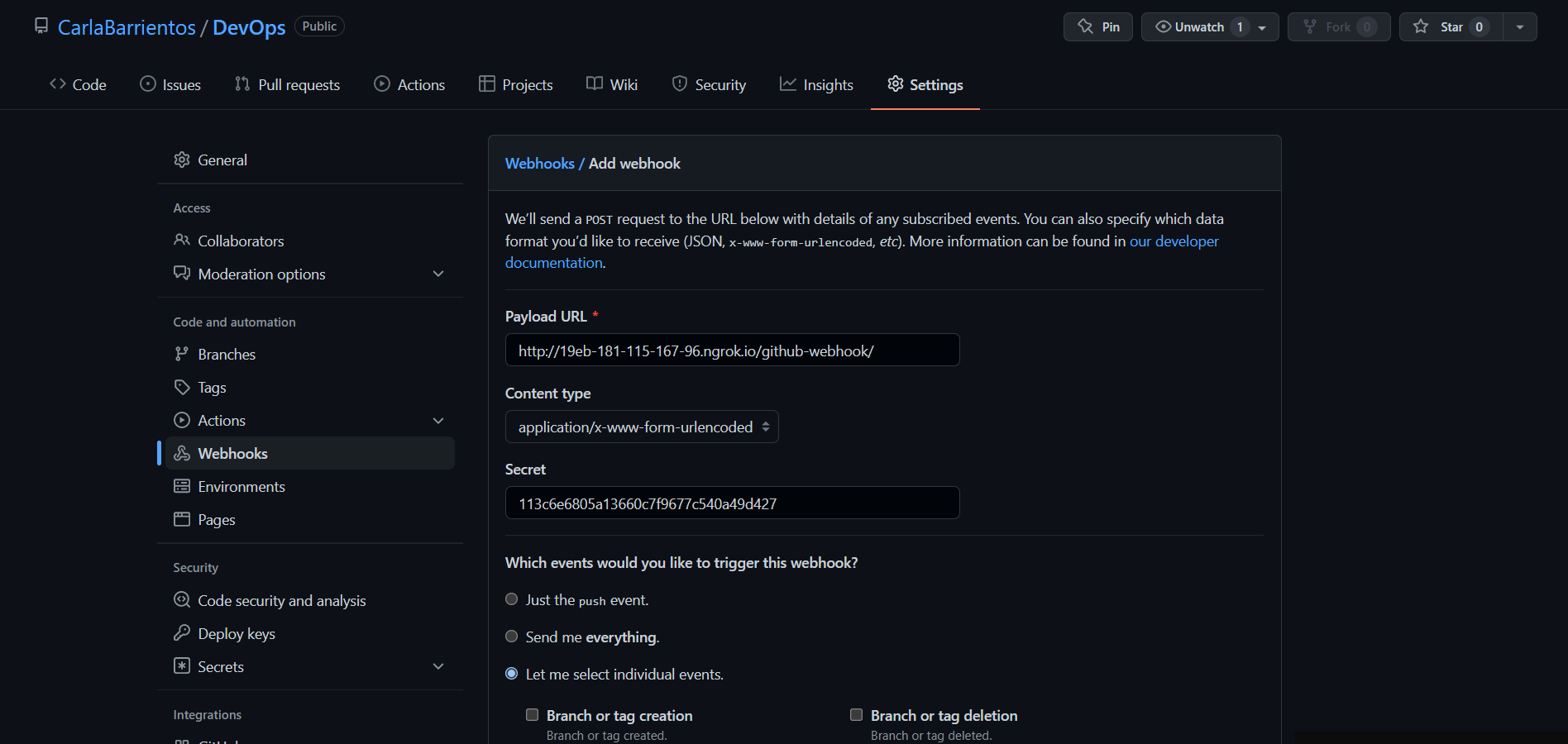


Now we can access Jenkins web page using the provided URL: <http://19eb-181-115-167-96.ngrok.io/>.

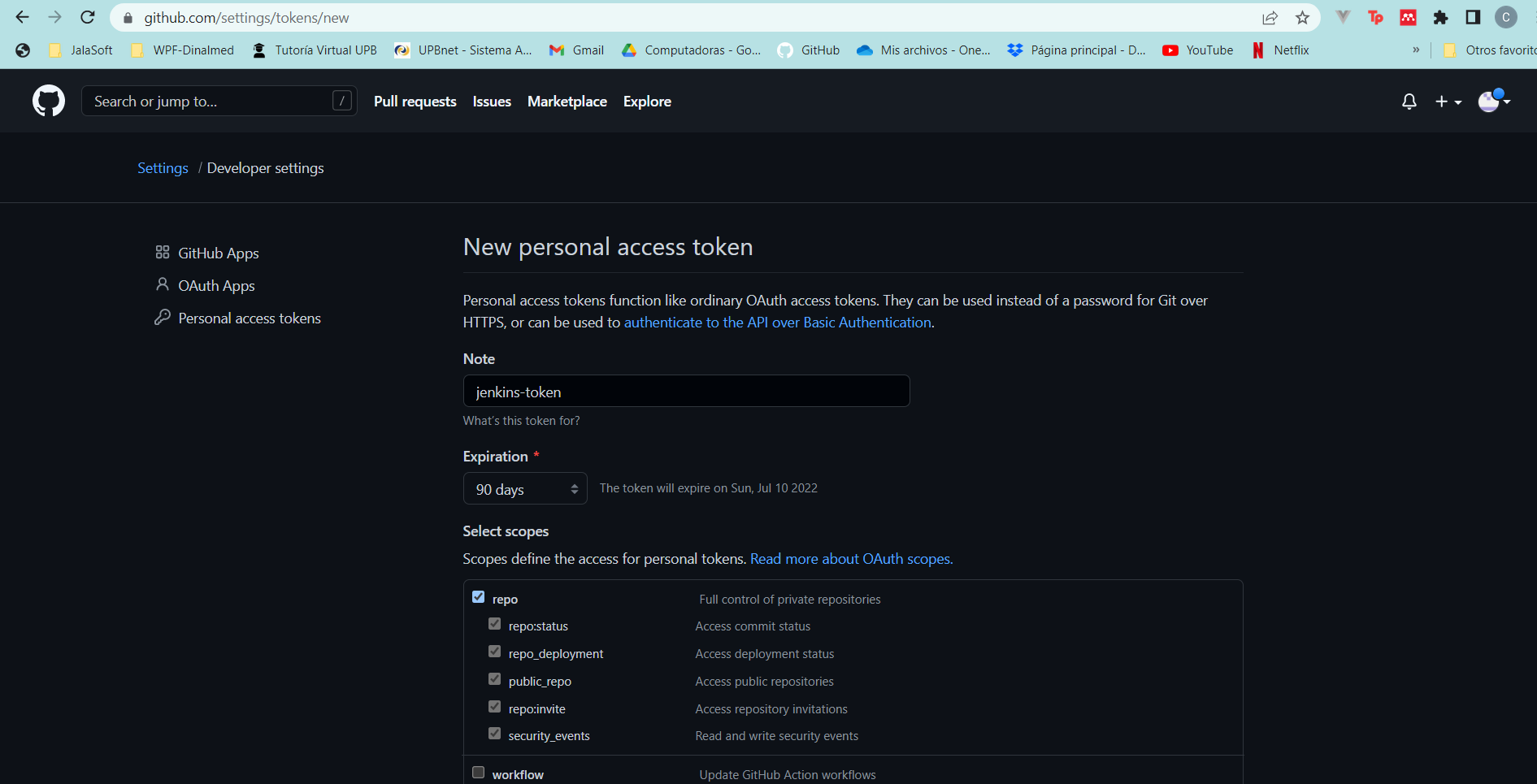
The next step is to configure GitHub webhooks, we create a token in Jenkins for GitHub going to our account -> confogurations and clicking on “Add new Token”, we have to save the generated token because it can not be recovered in the future:

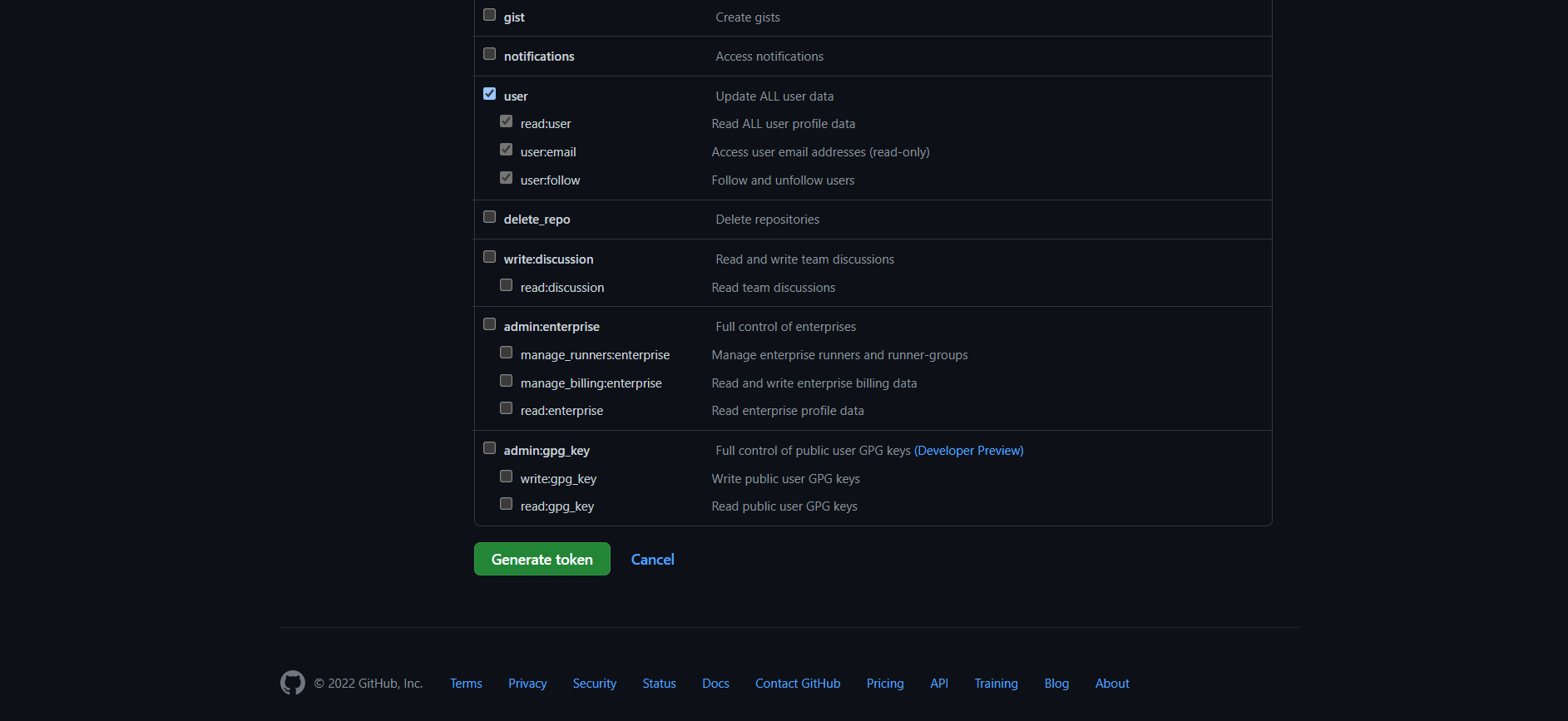


Then we go to our GitHub repo settings and select Webhooks option, we fill the Payload URL with the URL generated by ngrok adding /github-webhook ate the end and we copy the generated token in the last step:

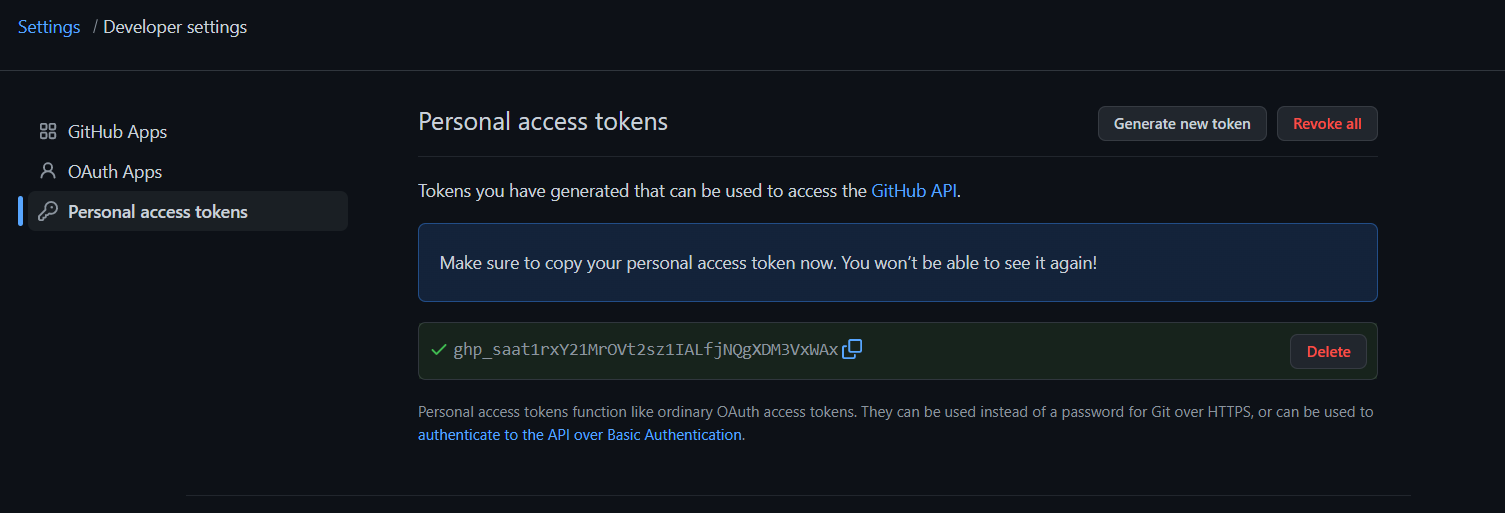


Next step is to add a new personal access token to our GitHub account, here is necessary to check repo and user options.

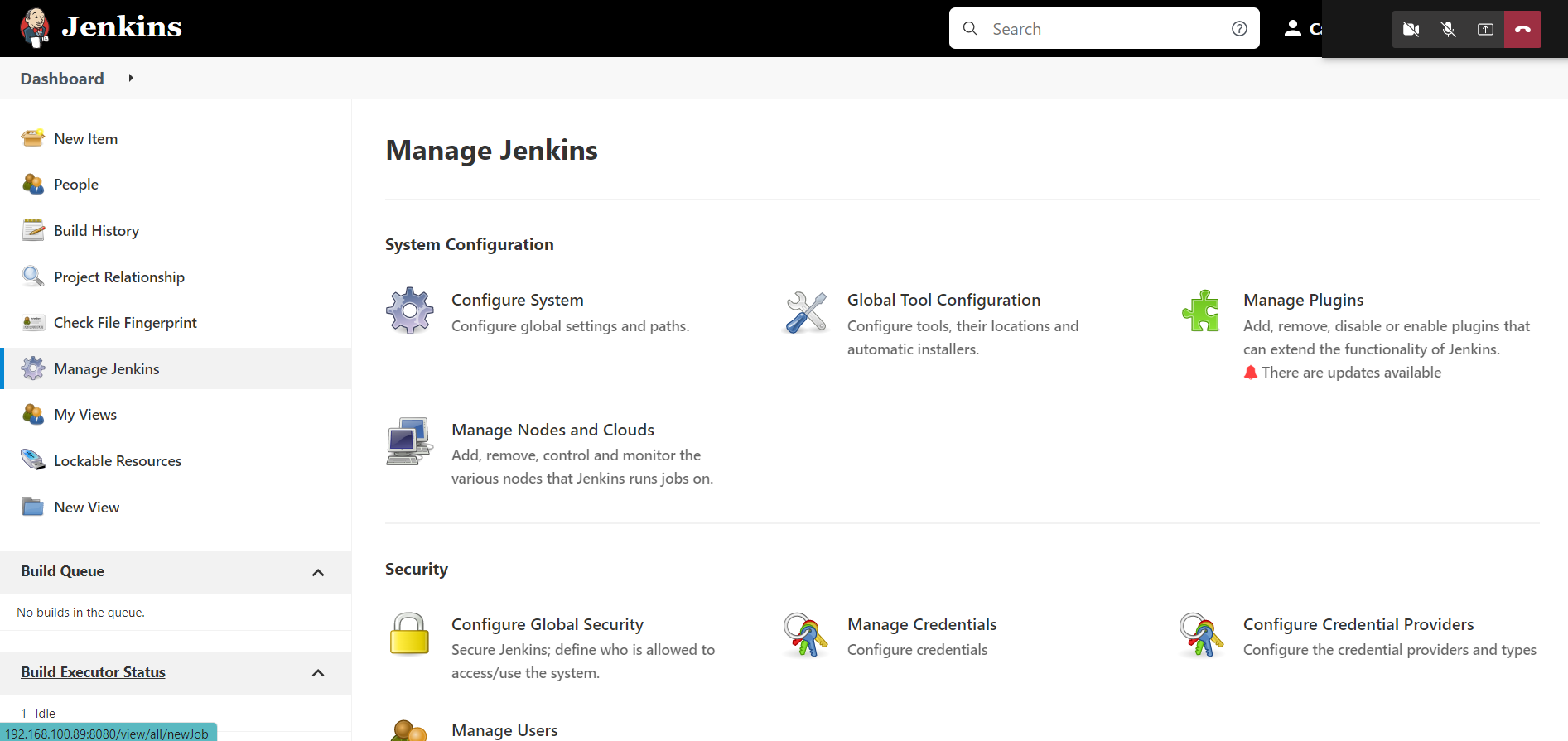




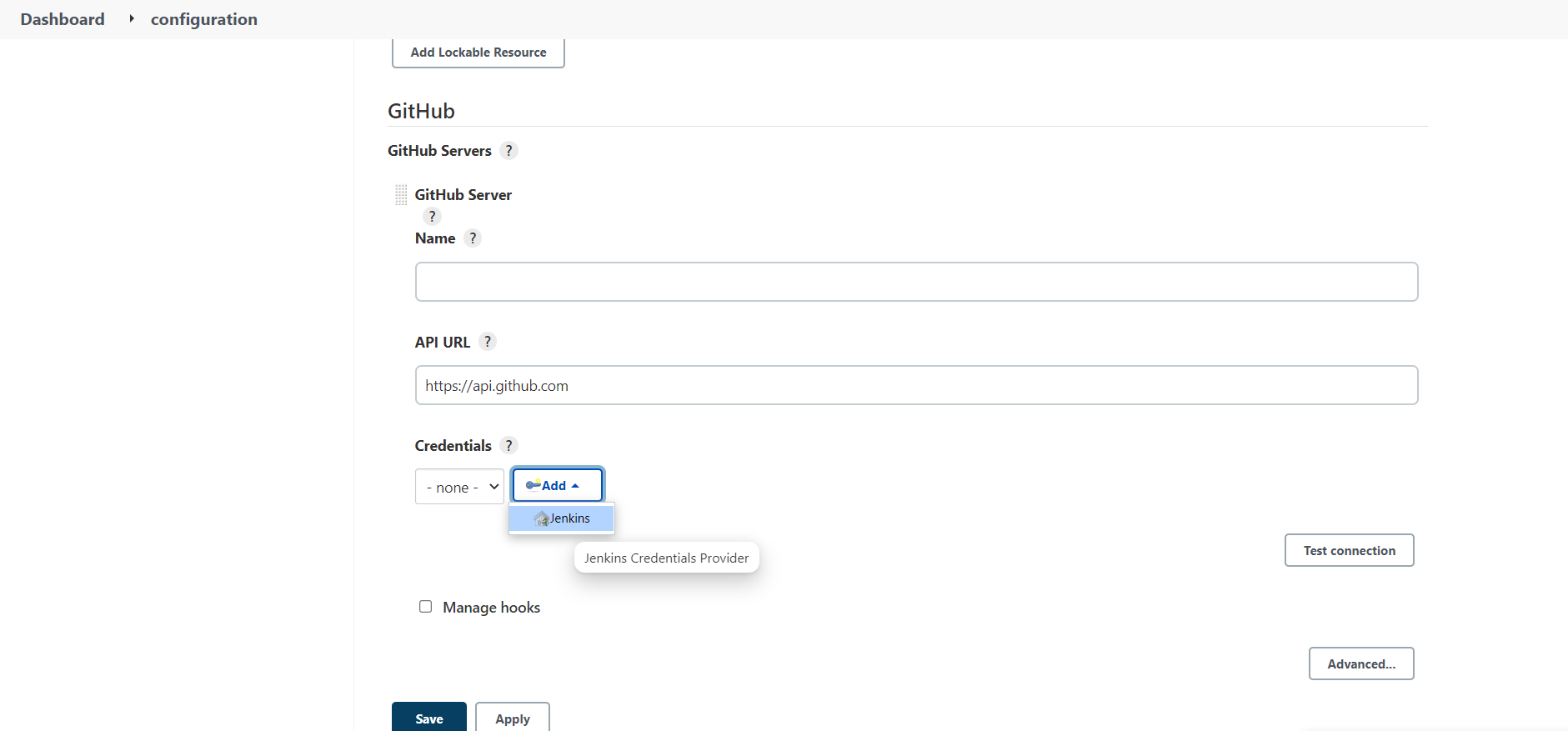
We need to save the personal token generated to access the GitHub API from our Jenkins server:



Then we need to add the GitHub credential generated in Jenkins. To do so, we go to Manage Jenkins -> Configure System



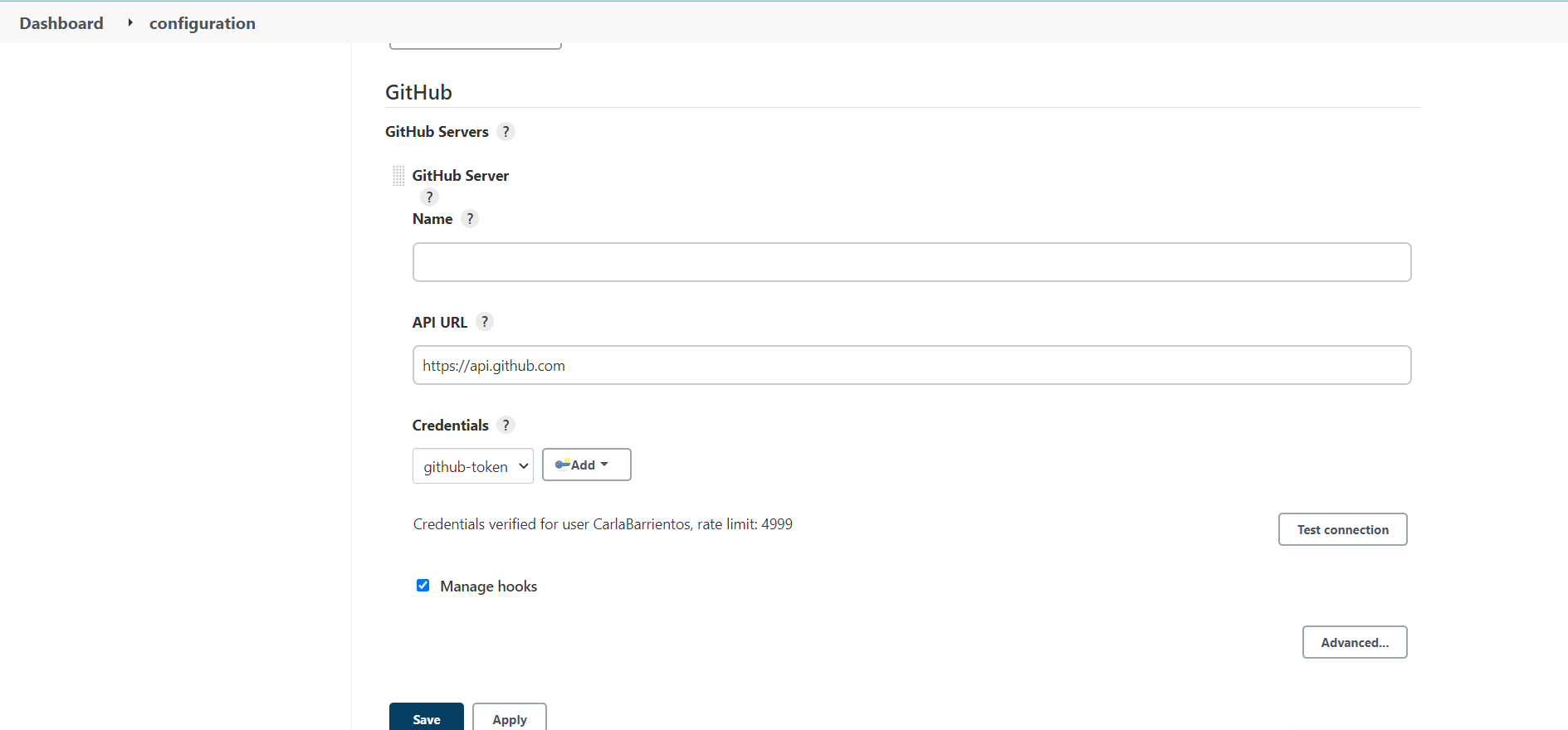
In the GitHub section we need to add a new credential:



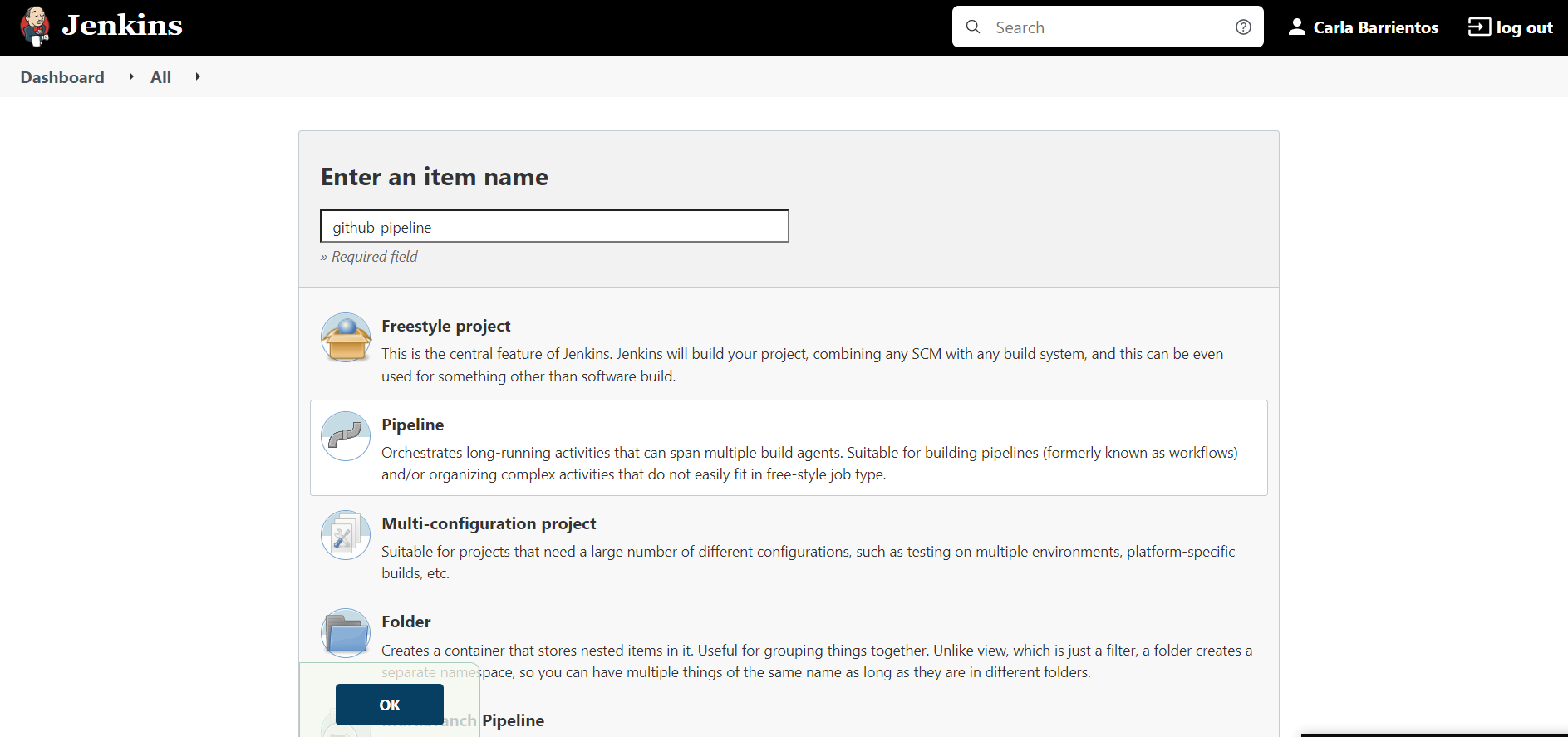
Here we have to copy the token generated in GitHub and write any ID:



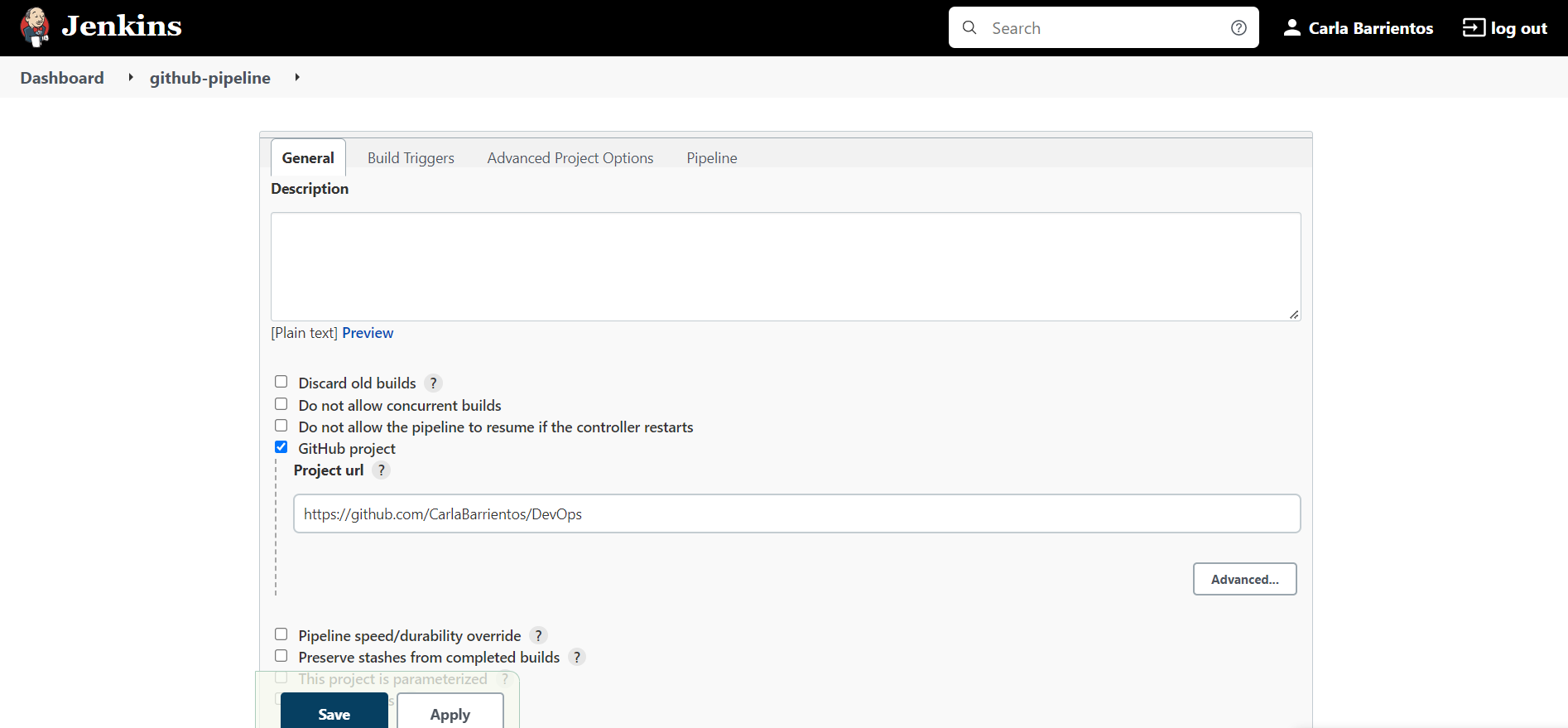
We select the created credential and check the option Manage hooks:



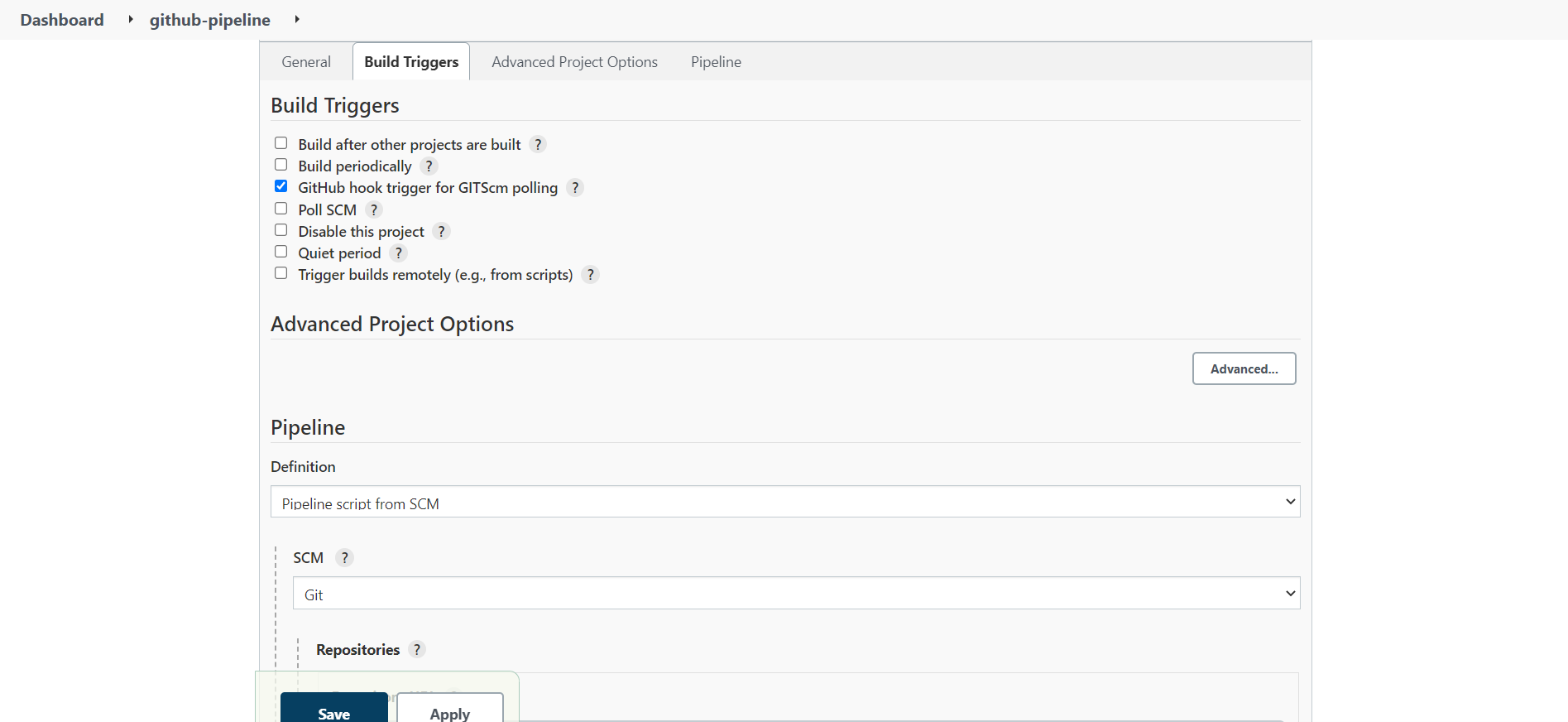
Finally we need to create a new pipeline to configure our repo, we enter the name of our Item and select the Pipeline option



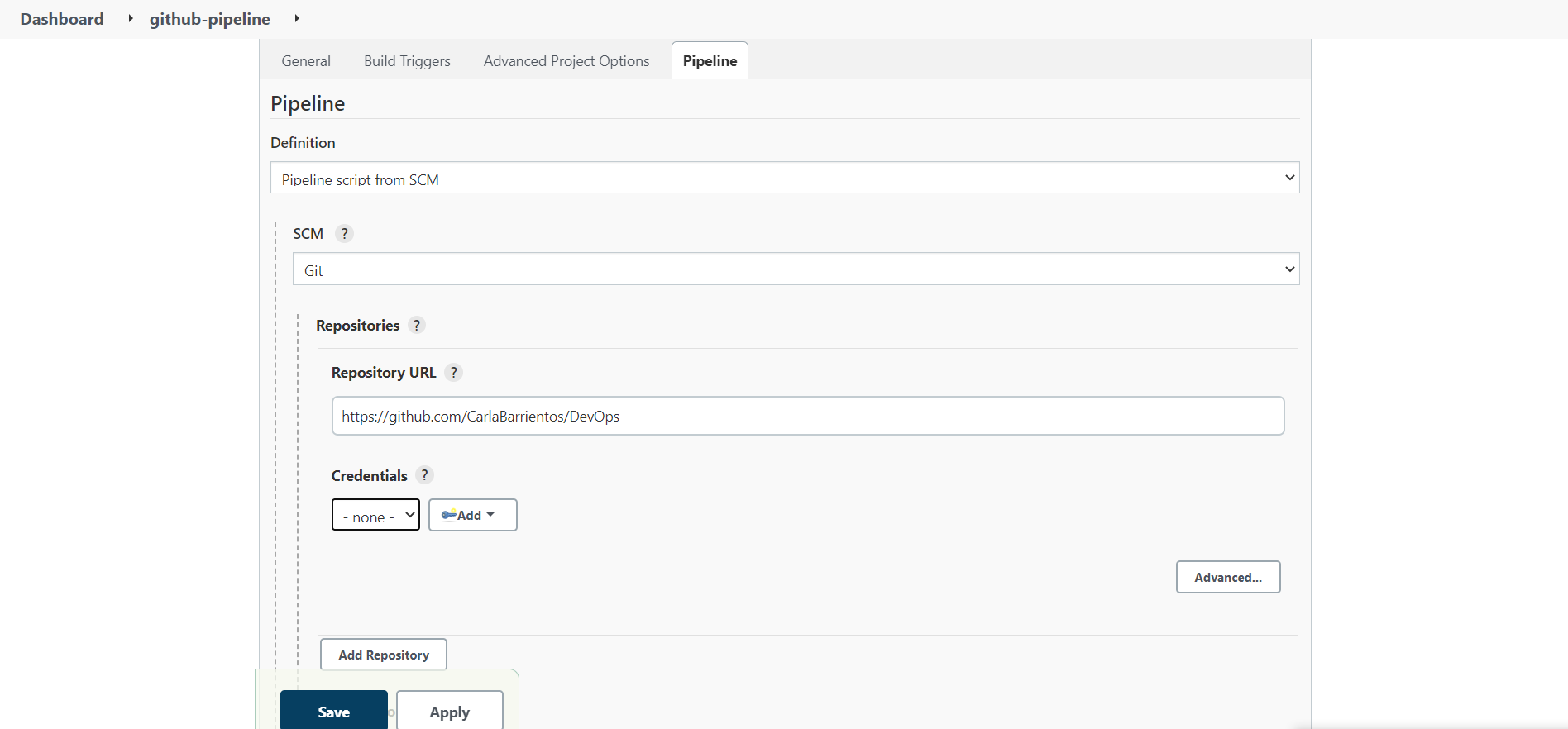
On the general setting we check the GitHub project option and copy the URL of our GitHub repository:



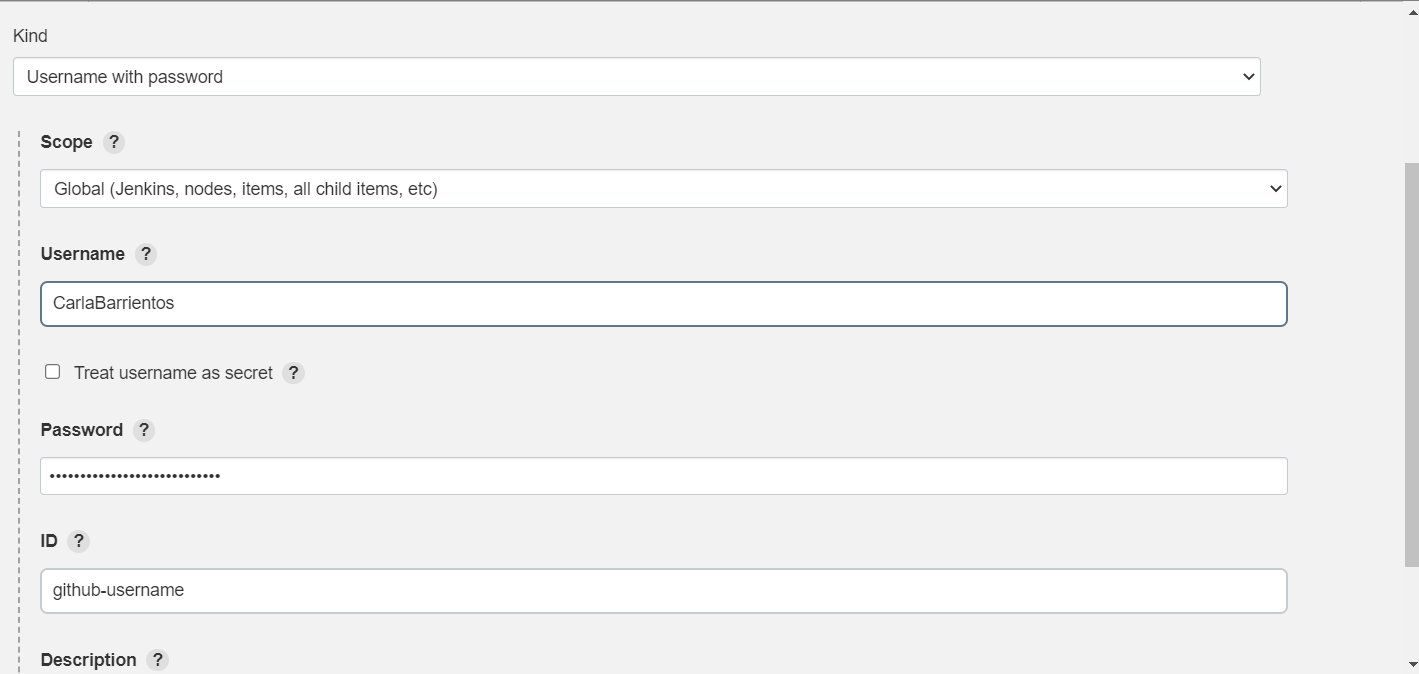
On the Build Triggers section we check GitHub gook trigger for GITScm polling:

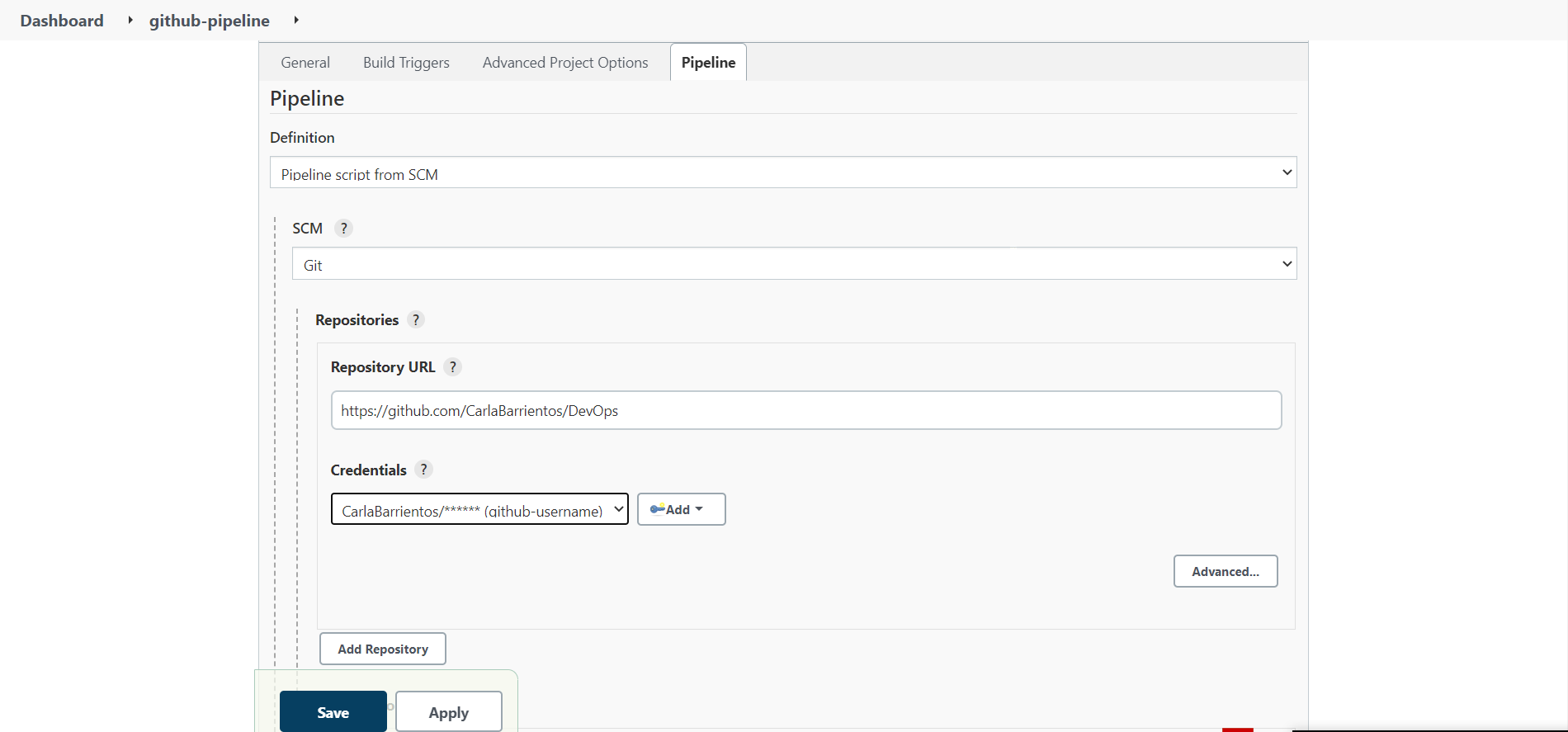


On the Pipeline section we select the Git option, copy the URL of our repo and add new credentials:

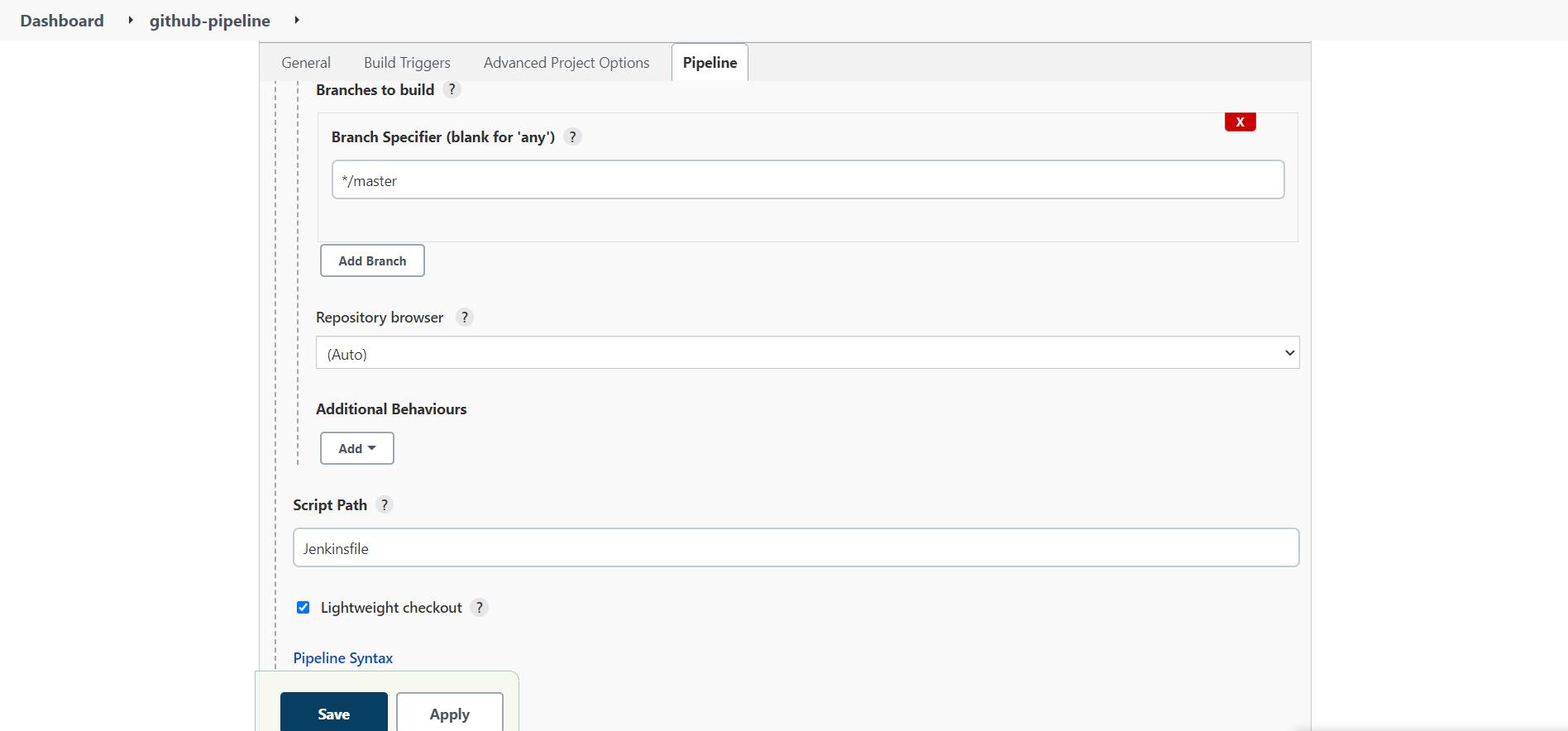


We fill the information with our GitHub credentials and choose any ID:

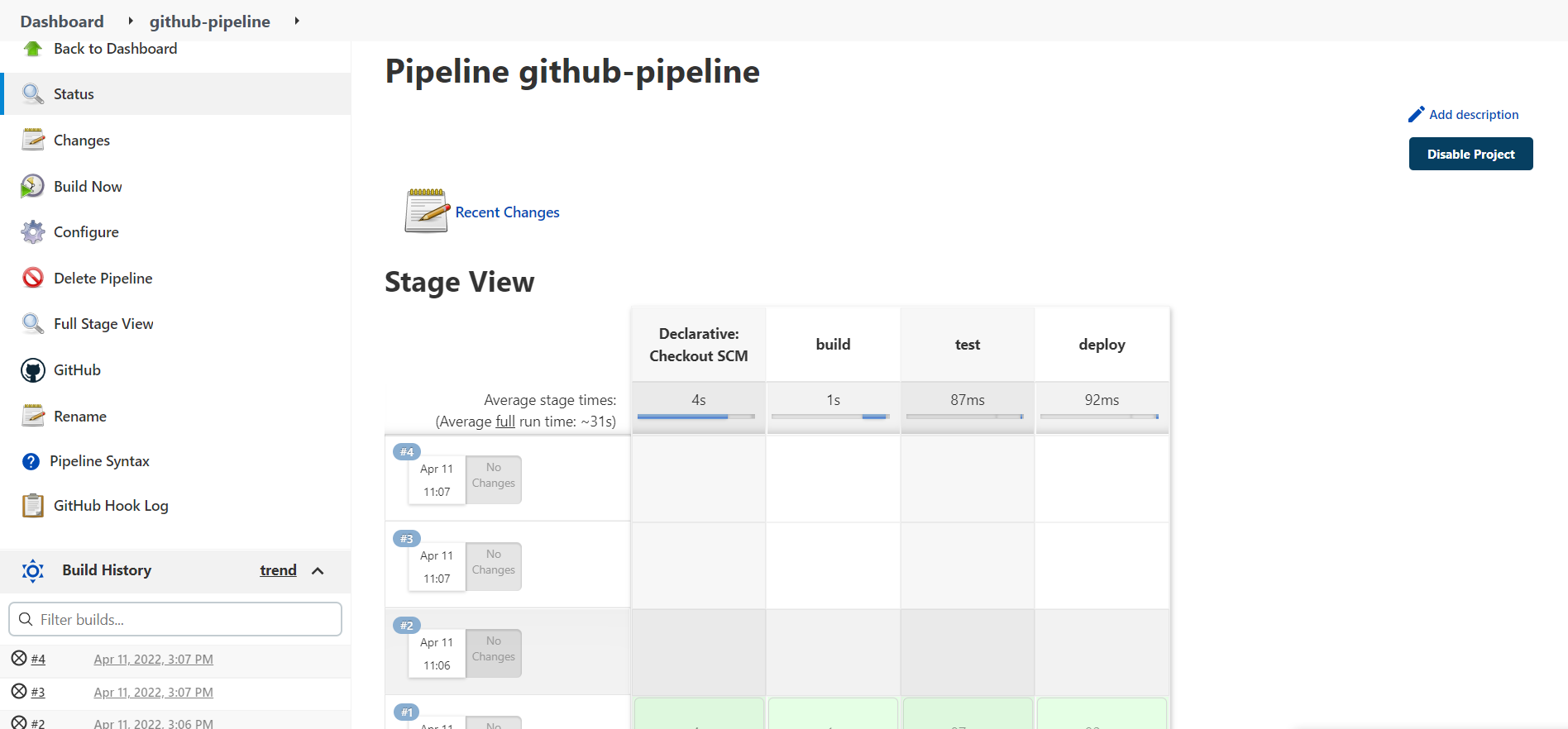




Then we specify the branches that we want to track and where is de Jenkinsfile located:



We can run the pipeline to see if everything is working using “Build now”



In the console option we can see that our Jenkinsfile had been used:

