How to use Webhooks in Jenkins

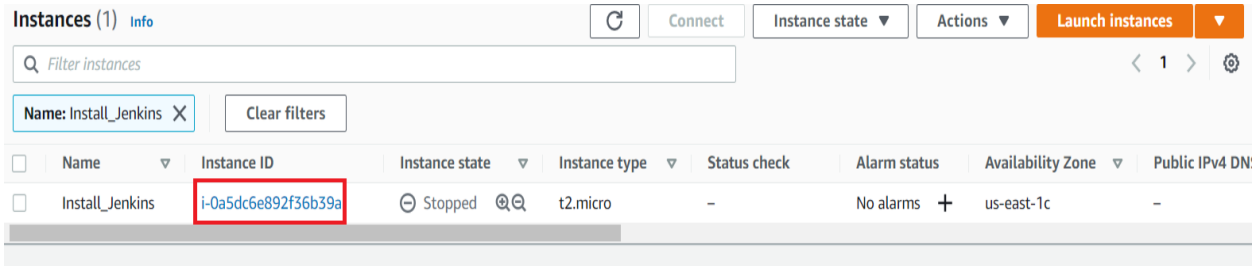
This document covers the instructions on how to integrate Jenkins with Git using Webhooks to trigger continuous downloading whenever any change is made in the code. Furthermore, this document is divided into the following three parts:

1. Opening port no 8080 for GitHub
2. Setting up WebHooks
3. Creating a freestyle job and triggering WebHooks

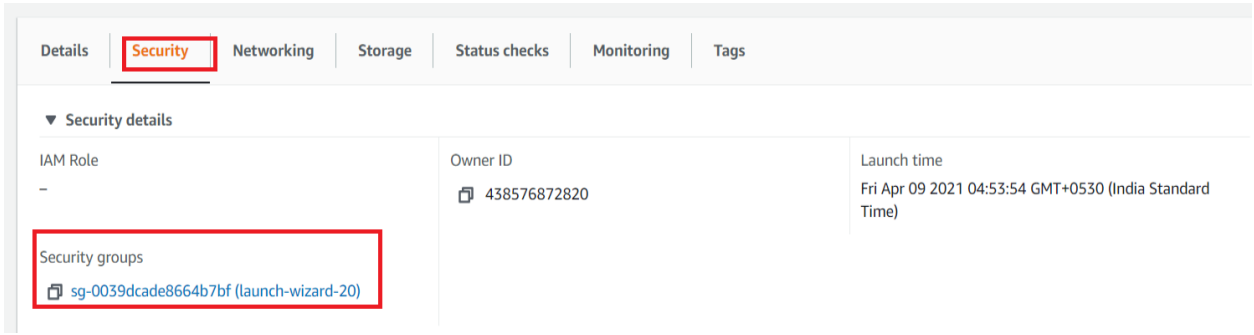
Opening port no 8080 for GitHub

So far, we have been setting port 8080 to **Myip**, but for working with webhooks, we need to open port 8080 for all traffic in the security groups. So, let's take a look at the steps involved.

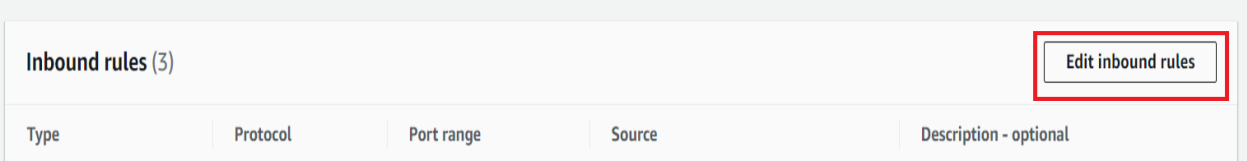
1. Go to your instance over the AWS dashboard as shown.



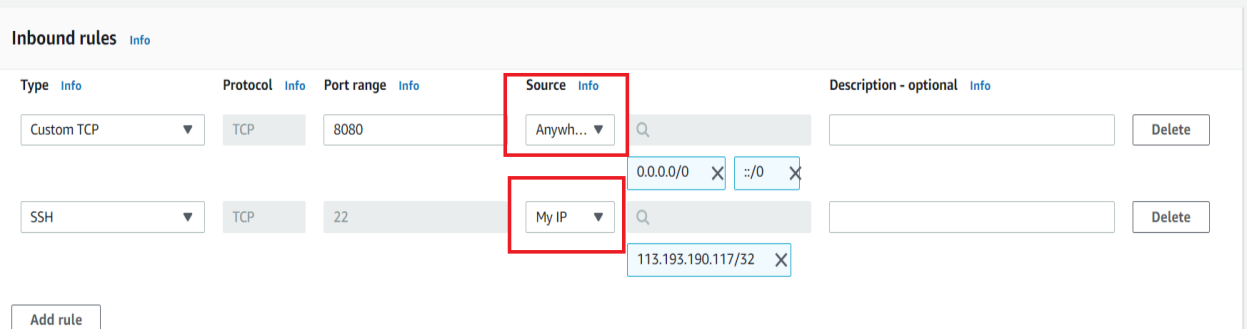
2. Then, click on **Instance Id**.
3. Next, click on **Security** and then select **Security groups**.



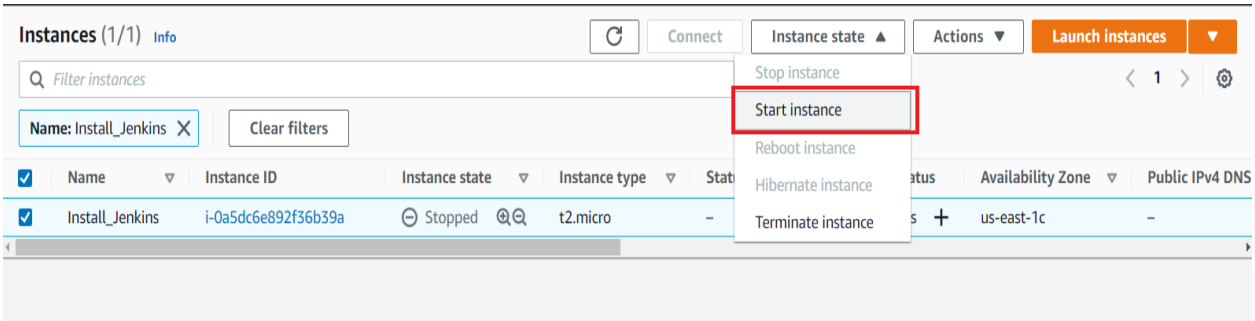
4. Click on **Edit inbound rules**.



5. Then, for port 8080, select **Anywhere** in the **Source**.
6. For port 22, it needs to be **My Ip**.



7. After that, click on **Save Rules**.
8. Then, come back to your instance dashboard from where you started.
9. Select **Your Instance**.
10. Click on **Instance State** and then select **Start Instance**.

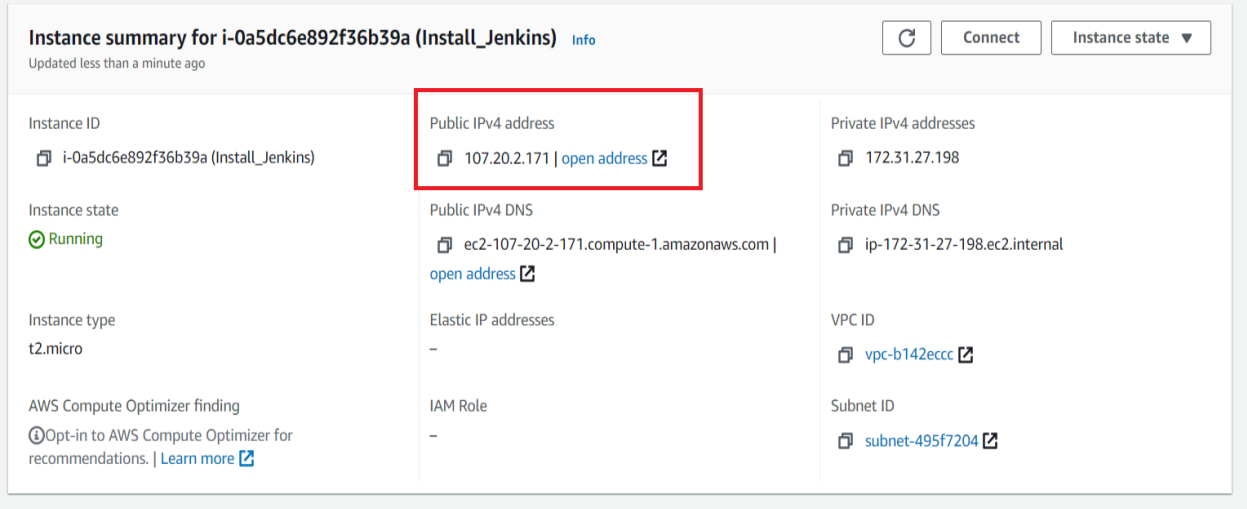


11. Once your machine comes in the running condition, log in to your machine using putty and go to the terminal.
12. Then, type the command **sudo service jenkins start.**
13. Run the **sudo service jenkins status** to check whether Jenkins has started.

```
ubuntu@ip-172-31-27-198:~$ sudo service jenkins start
ubuntu@ip-172-31-27-198:~$ sudo service jenkins status
● jenkins.service - LSB: Start Jenkins at boot time
     Loaded: loaded (/etc/init.d/jenkins; generated)
     Active: active (exited) since Fri 2021-04-09 00:32:43 UTC; 6min ago
       Docs: man:systemd-sysv-generator(8)
    Process: 424 ExecStart=/etc/init.d/jenkins start (code=exited, status=

Apr 09 00:32:40 ip-172-31-27-198 systemd[1]: Starting LSB: Start Jenkins a
Apr 09 00:32:41 ip-172-31-27-198 jenkins[424]: Correct java version found
Apr 09 00:32:41 ip-172-31-27-198 jenkins[424]:  * Starting Jenkins Automat
Apr 09 00:32:42 ip-172-31-27-198 su[605]: (to jenkins) root on none
Apr 09 00:32:42 ip-172-31-27-198 su[605]: pam_unix(su-l:session): session
Apr 09 00:32:42 ip-172-31-27-198 su[605]: pam_unix(su-l:session): session
Apr 09 00:32:43 ip-172-31-27-198 jenkins[424]:    ...done.
Apr 09 00:32:43 ip-172-31-27-198 systemd[1]: Started LSB: Start Jenkins at
ubuntu@ip-172-31-27-198:~$
```

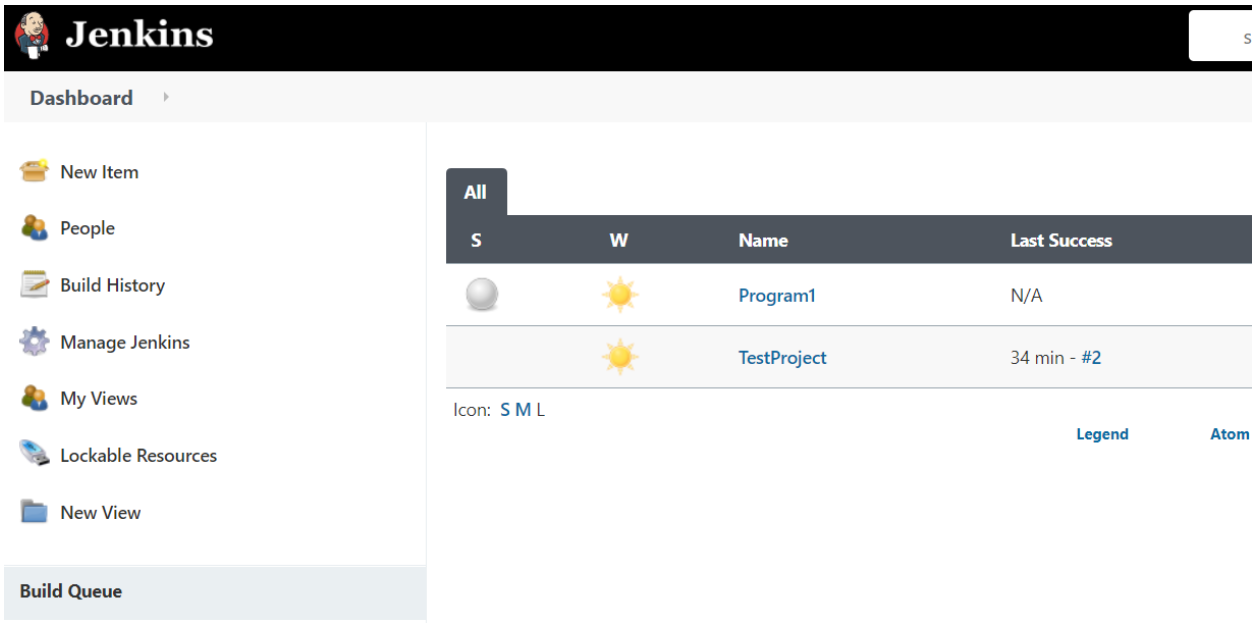
14. Now, to access Jenkins, copy the public URL of your Ubuntu instance.



15. Then, go to the browser and paste **<url:8080>** and press enter.

16. After that, provide the username and password.

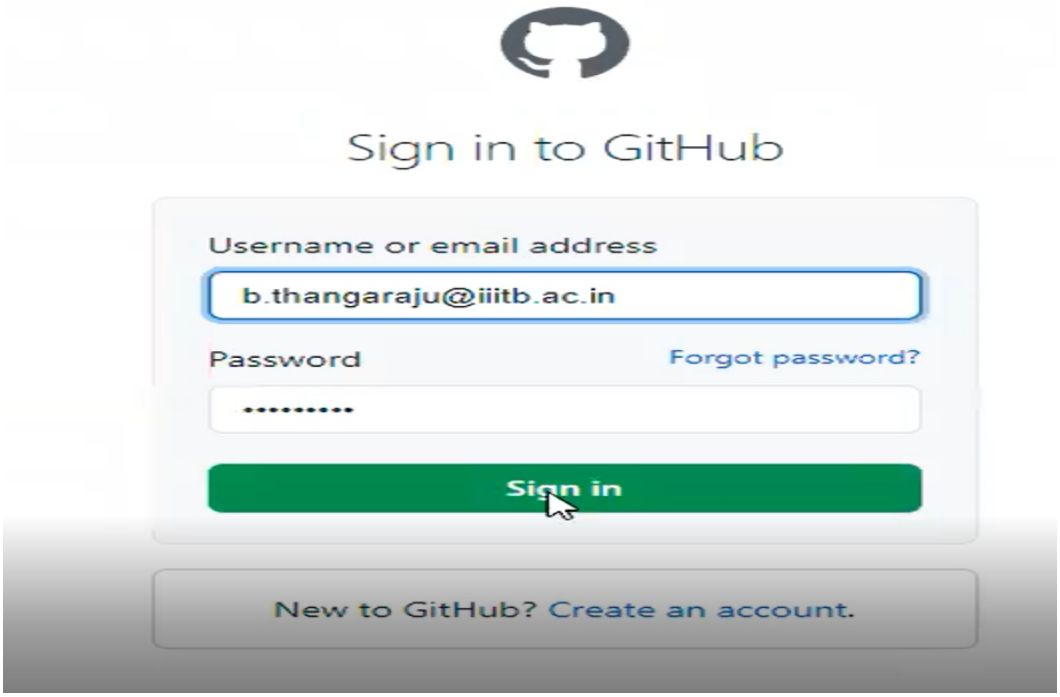
17. Then, you will be directed to the Jenkins dashboard.



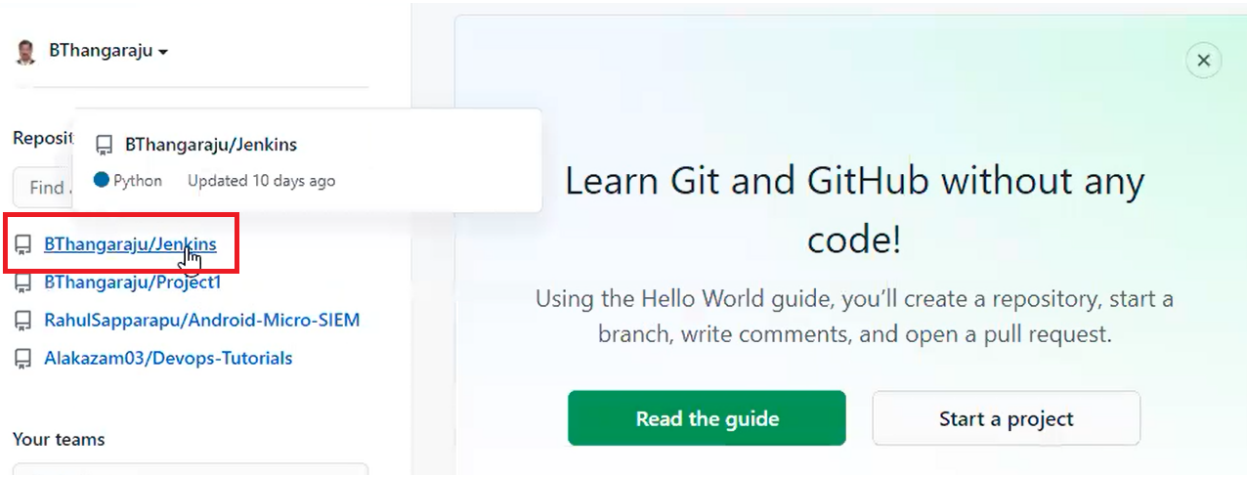
Setting up WebHooks

So, let's take a look at the steps involved in setting up WebHooks in Jenkins.

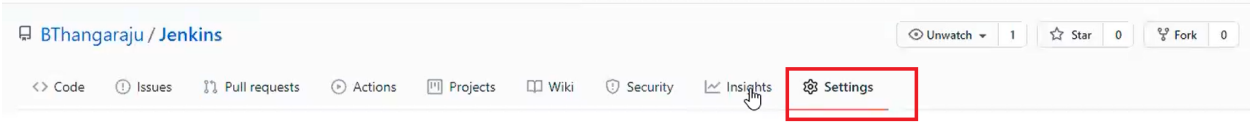
1. Sign in to the GitHub account.



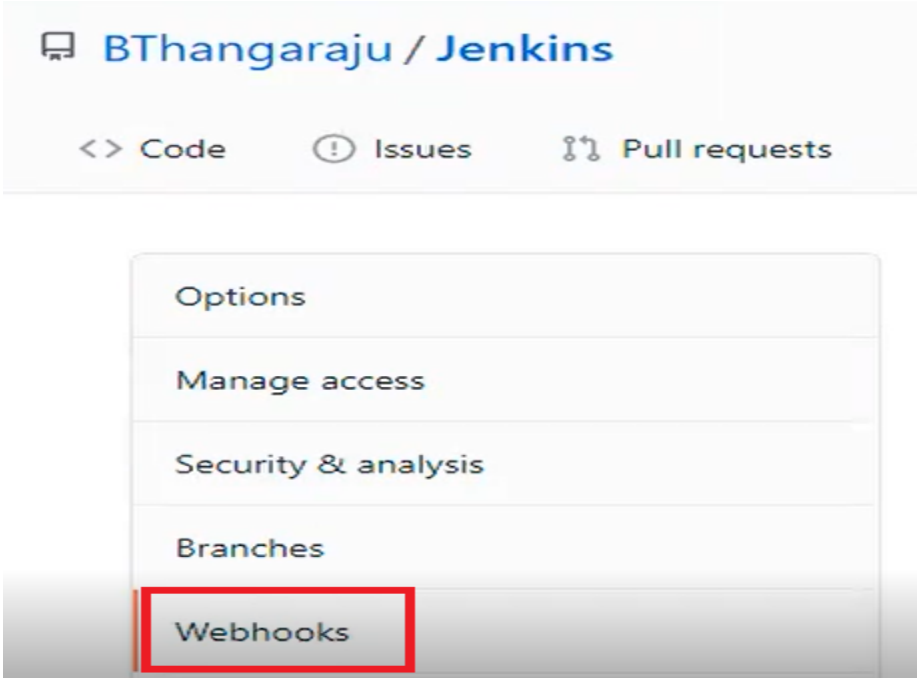
2. Now, select the repository where you have kept your code.



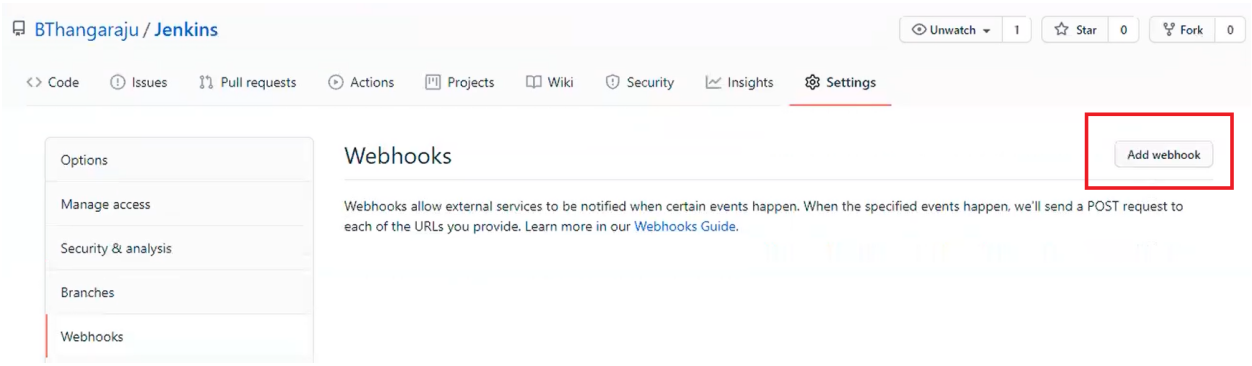
3. Then, click on **Settings**.



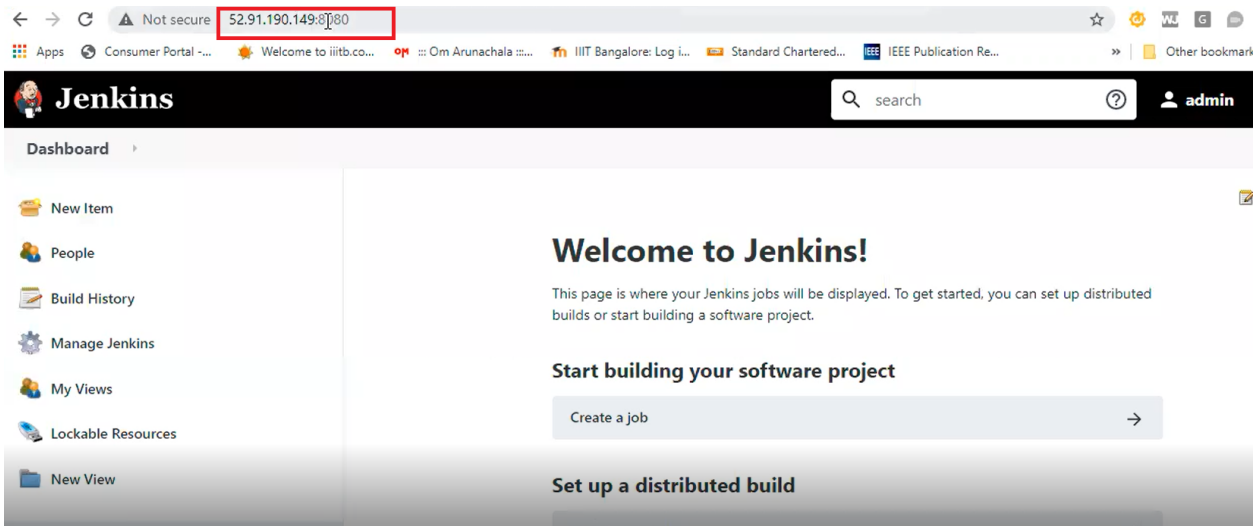
4. Then, select **Webhooks** from the menu on the left.



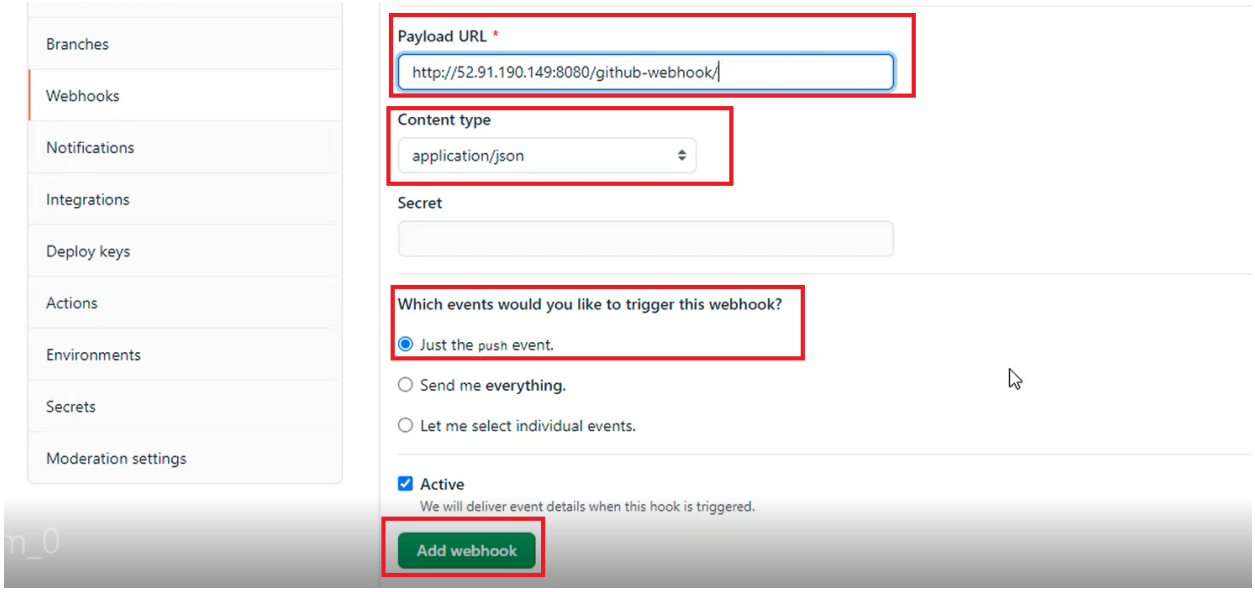
5. Then, click on **Add Webhooks** as shown in the screenshot given below.



6. We need to provide a **Payload URL**. So, go to the Jenkins dashboard and copy the URL as shown in the screenshot given below.



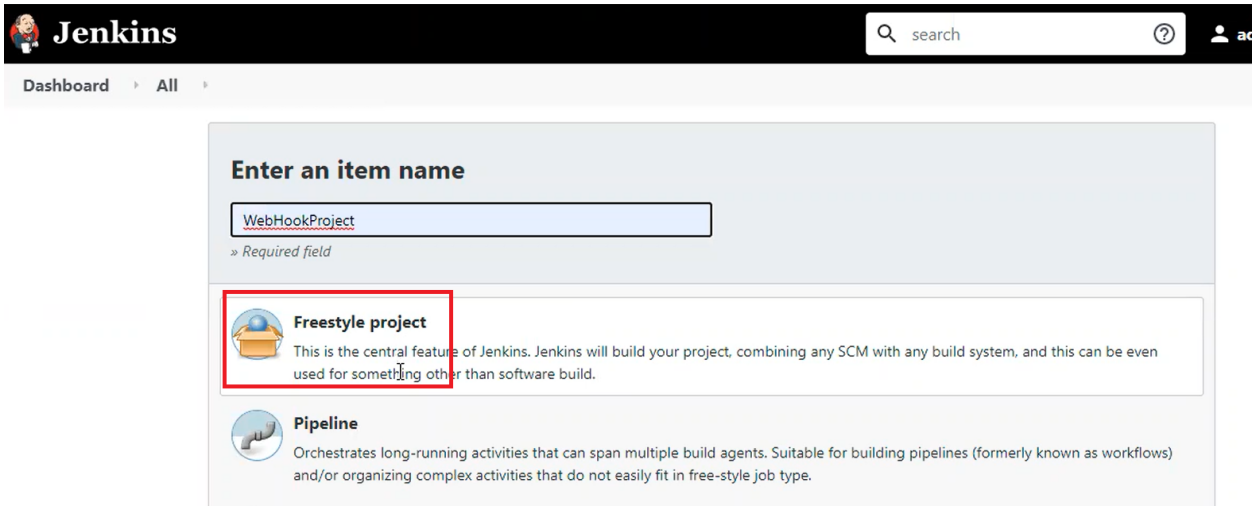
7. Next, come back to GitHub and paste the URL. Remember that you need to append **/github-webhook/** after the URL**.**
8. Then, select the **content type** and select **application/json**.
9. In the **Which events would you like to trigger the webhook?** option, select **Just the push event**.
10. After that, click on **Add Webhook** as shown in the screenshot given below.



Creating a freestyle project and trigger WebHooks

So far, we have set up webhooks. Now, you will learn about the steps that you need to follow to trigger webhooks.

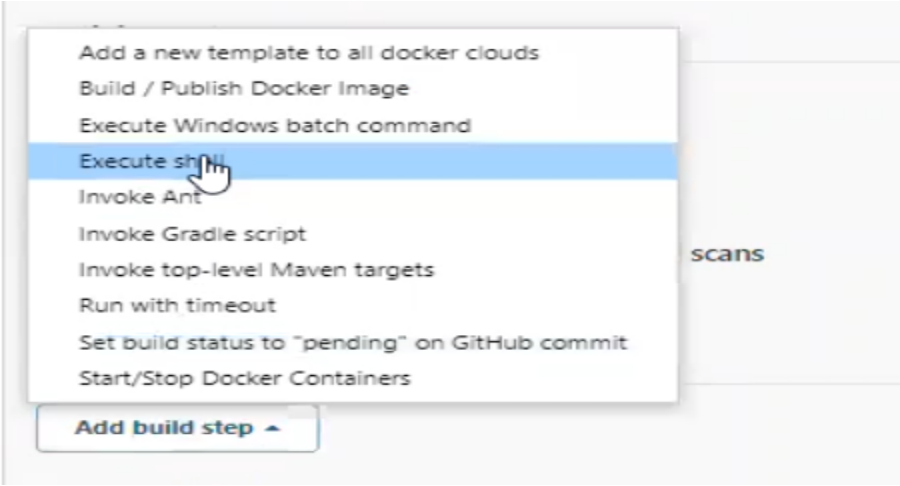
1. Come back to the **Jenkins Dashboard** and create a new freestyle job. Here, we created a job named WebhookProject.



2. In the **Source Code Management**, select **Git** and provide the **Repository URL** as shown in the screenshot given below.

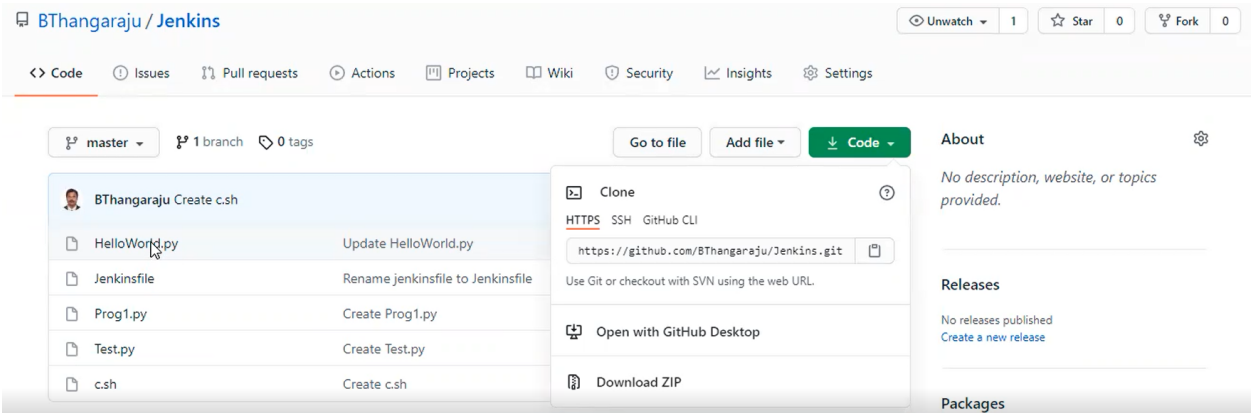
3. Scroll down, and in the **Build Triggers** option, select the **GitHub hook trigger** for the **GitScm polling** option.

4. Scroll down, and in the **Build** section, select the **Execute shell** option.

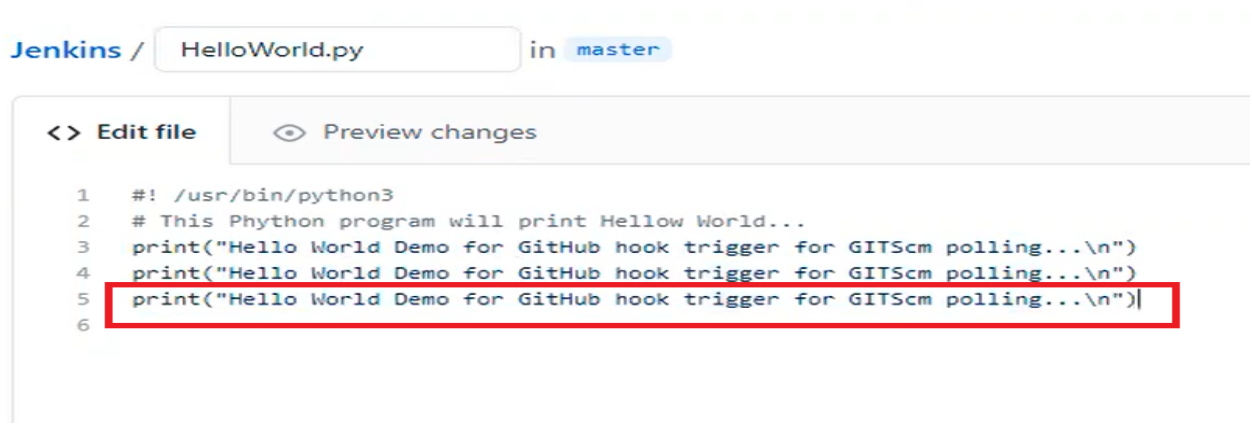


5. Then, write the script that you will execute.

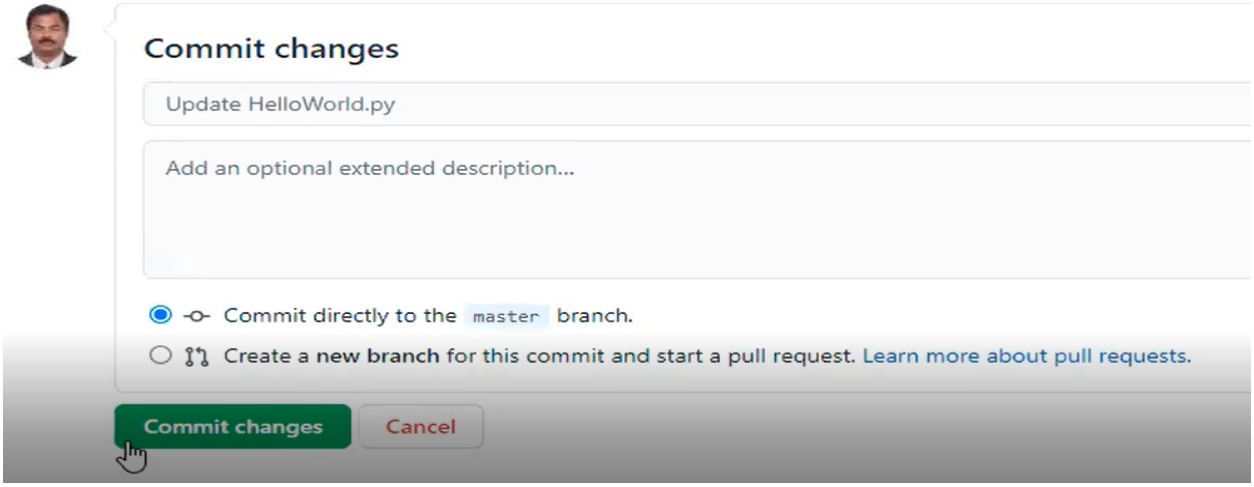
6. Click on **Apply** and then **Save**.
7. To trigger Webhooks, you need to make some changes. For this, go back to the GitHub repository.



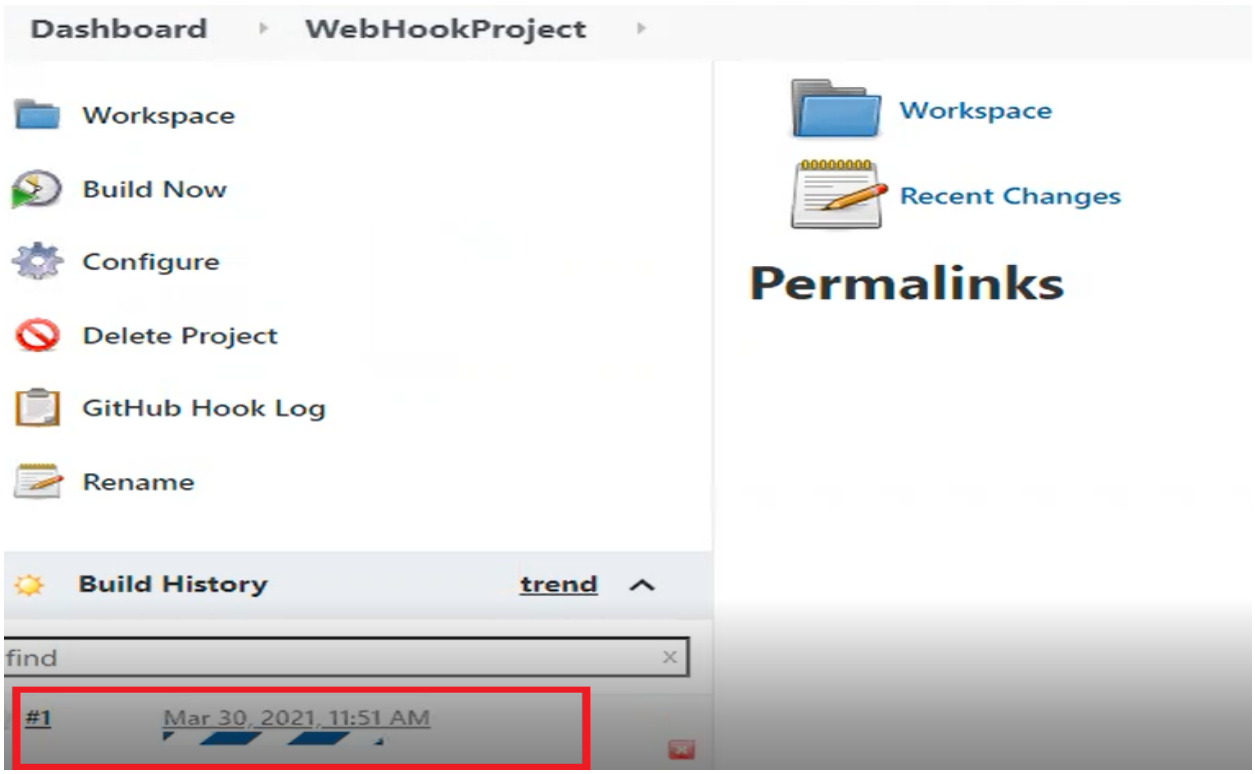
8. Then, edit the file and make some changes. Here, we are adding one more line to the file **HelloWorld.py**.



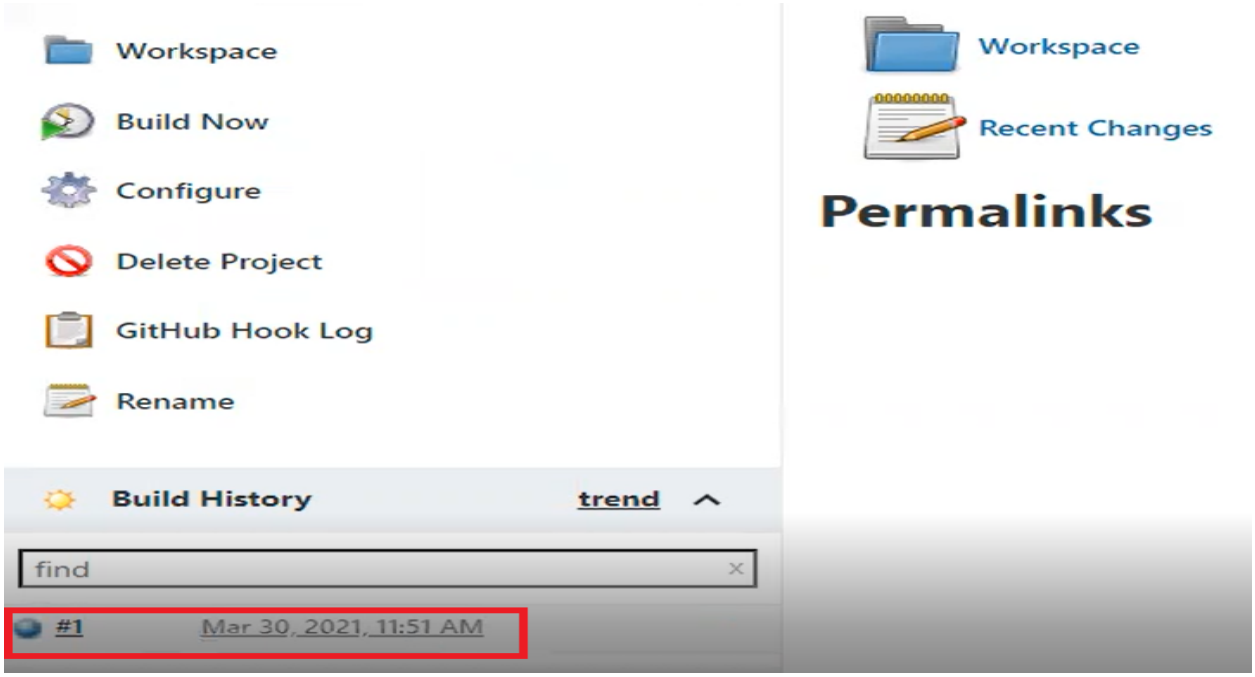
9. Now, scroll down and commit the changes.



10. Then, come back to the **Jenkins Dashboard**. You would see your job getting built as shown in the screenshot given below.



11. Once the project is successfully built, click on the build number to view the log output as shown in the screenshot given below.



11. Once you click on the build, you can view the console output as shown in the screenshot given below.