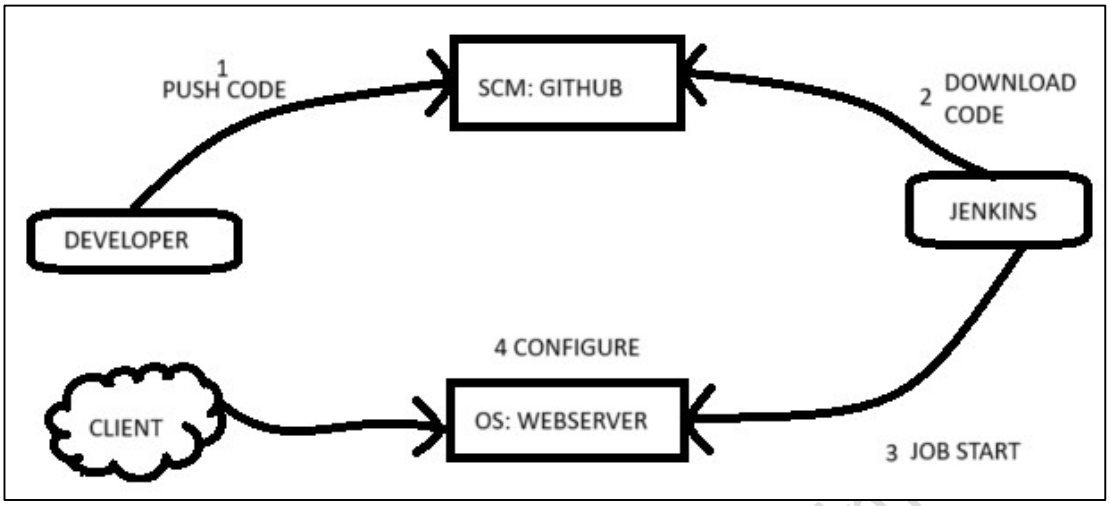
Web Server Configure By Automation



Once upon a time in a digital world, there was a clever developer named Alex. Alex was working on a very important project, creating amazing programs on a computer. Now, to keep all their code safe and organized, Alex wanted to use something special called a centralized repository system. This system would store all the code in one place, like a big virtual library for code.

The place Alex chose for this special library was a famous one called GitHub. GitHub was like a magical vault where code could be stored securely. So, whenever Alex finished writing some new code, they would carefully put it inside the GitHub vault. This was like saving a precious treasure in a secret box.

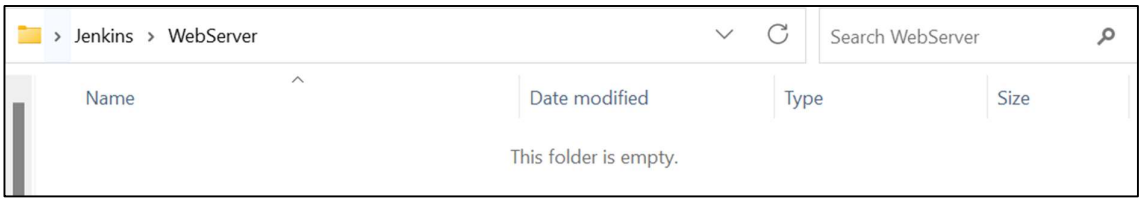
But Alex didn't want to just save their code and forget about it. They wanted to make sure everything was working perfectly. So, they had a helper named Jenkins, who was like a diligent assistant. Whenever Alex put new code into the GitHub vault, Jenkins would wake up and go check GitHub.

Jenkins had a special power – it could take the code from the GitHub vault and bring it to a special place on the computer. This place was called the Document Root, which was like the main folder where the website lived. Imagine it as the heart of the website, where all the important files were kept.

The Document Root lived in a secret location on the computer, known as "/var/www/html." It was a safe and cozy spot where all the website's files could rest. So, when Jenkins fetched the code from GitHub, it carefully placed it in the Document Root. It was like Jenkins was taking the treasure from the vault and putting it in the heart of the website.

And that's how Alex's special setup worked! With the magic of GitHub and the help of Jenkins, every piece of code found its home in the Document Root. Alex could now work on their project without worrying about losing any code. And whenever they added new treasures to the GitHub vault, Jenkins would make sure they found their way to the Document Root, ready to shine on the website for the world to see.

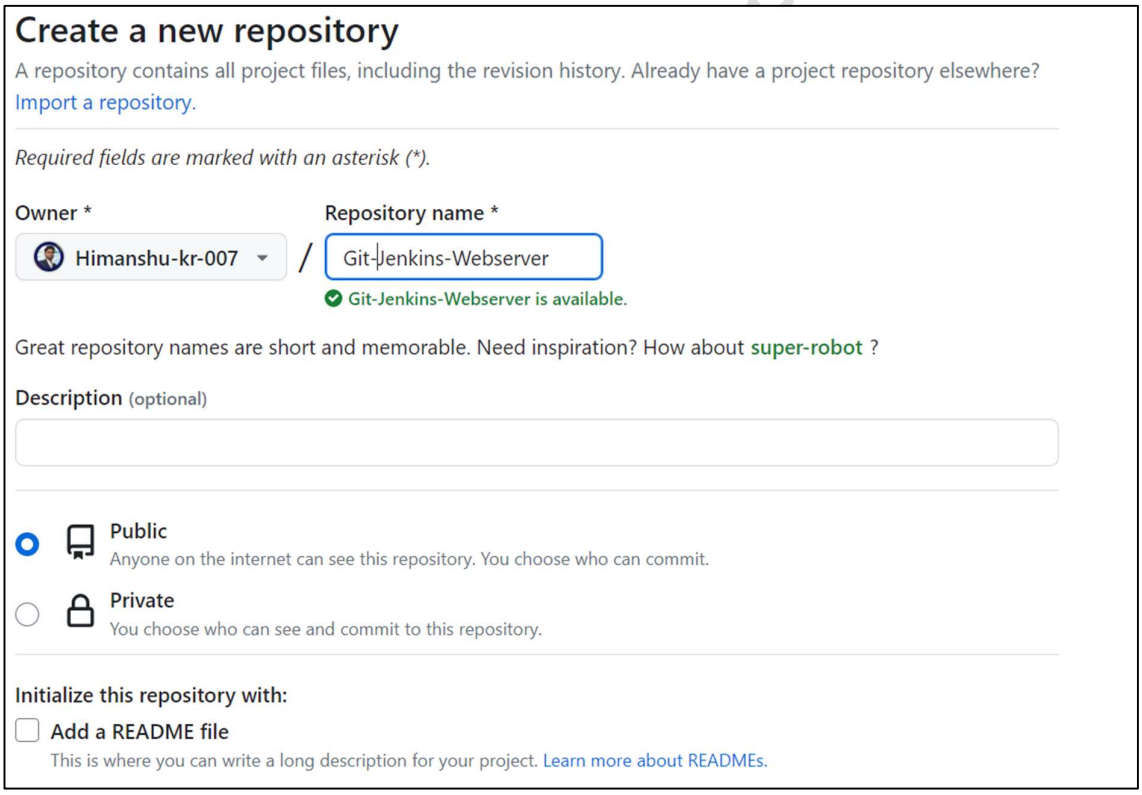
I have Created an Empty Repository and named it a Web Server. Now, Opening the terminal Git Bash.



Initializing the Git Repository

```
Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer
$ ls

Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer
$ git init
Initialized empty Git repository in C:/Users/Himanshu Kumar/Desktop/Jenkins/WebS
erver/.git/
```

In My Git Hub Account, I have created one repository where I am going to store the webpage from my Local System.



I have created one webpage with the name of index.html.

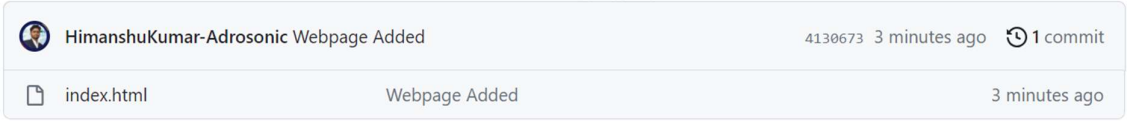
```
Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer (master)
$ ls
index.html
```

After that I have pushed this code into my Git Hub Account.

```
Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer (master)
$ ls
index.html

Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer (master)
$ git add .

Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer (master)
$ git commit -m "Webpage Added"
[master (root-commit) 4130673] Webpage Added
 1 file changed, 28 insertions(+)
 create mode 100644 index.html

Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer (master)
$ git branch -M main

Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer (main)
$ git remote add origin https://github.com/Himanshu-kr-007/Git-Jenkins-Webserver.git
```

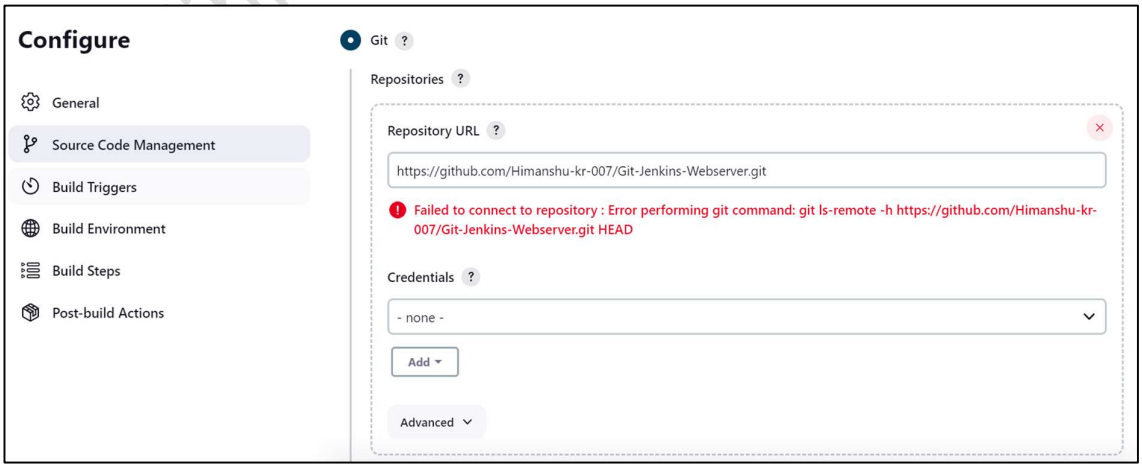
```
Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer (main)
$ git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 493 bytes | 164.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Himanshu-kr-007/Git-Jenkins-Webserver.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'
```

And I can see this on my Git Hub Account.



In the Jenkins I am going to create one job with the name of Git-Jenkins-Webserver and Here I am going to mention my Git Hub URL where My code is stored.

In the SCM section I selected Git and then mentioned the URL of my repository. But here I am getting the error. This is because Jenkins try to connect with my Git Hub by using git command, but in my OS I don't have git command installed



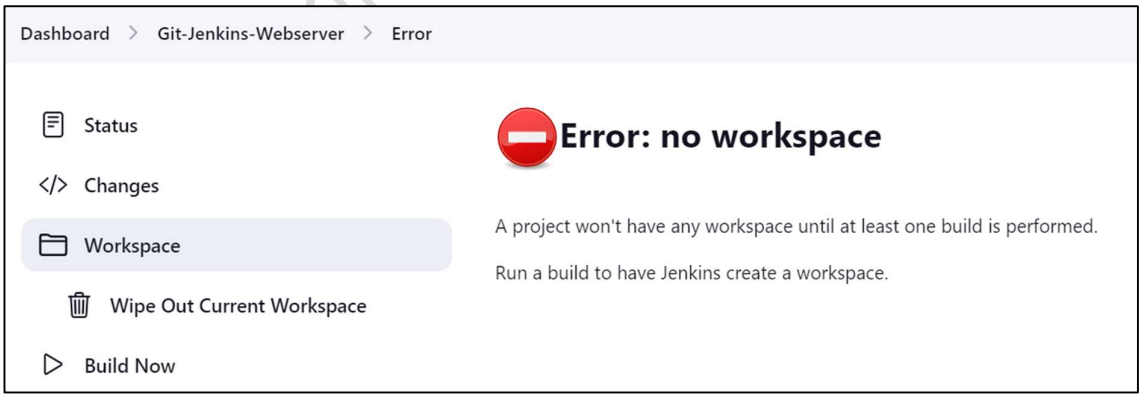
```
[ec2-user@ip-172-31-32-78 ~]$ git
-bash: git: command not found
[ec2-user@ip-172-31-32-78 ~]$
```

For Solving this error, I am downloading the Git command by using yum.

```
[ec2-user@ip-172-31-32-78 ~]$ sudo yum install git -y
Last metadata expiration check: 20:42:41 ago on Thu Aug 17 09:30:52 2023.
Dependencies resolved.
================================================================================================================================
 Package                    Architecture                Version                          Repository                      Size
================================================================================================================================
Installing:
 git                        x86_64                      2.40.1-1.amzn2023.0.1            amazonlinux                     57 k
```

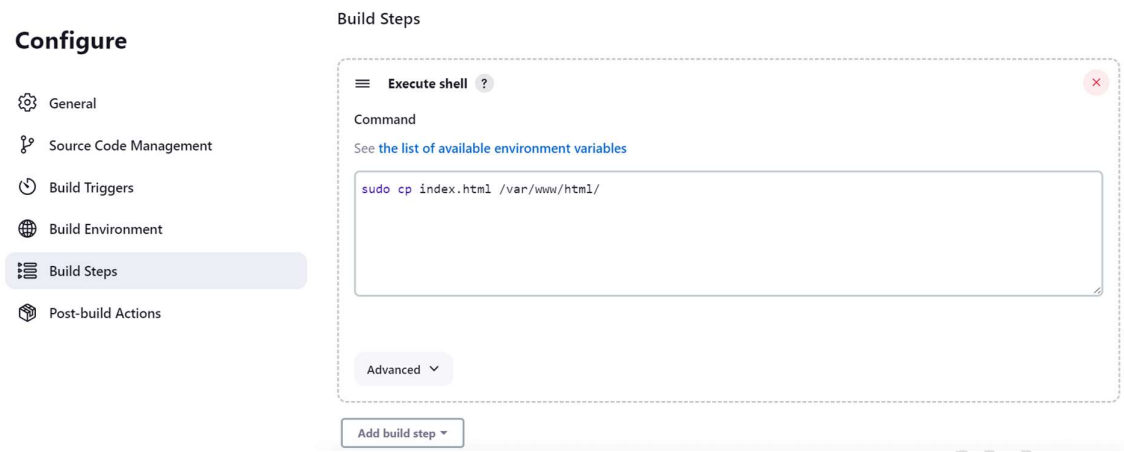
Now Just Cut the URL of repository and paste it again and then no error will come.

Jenkins will go to Git Hub URL and they find the code from the main branch and download the entire code and store in the Workspace for this we need to build the project.

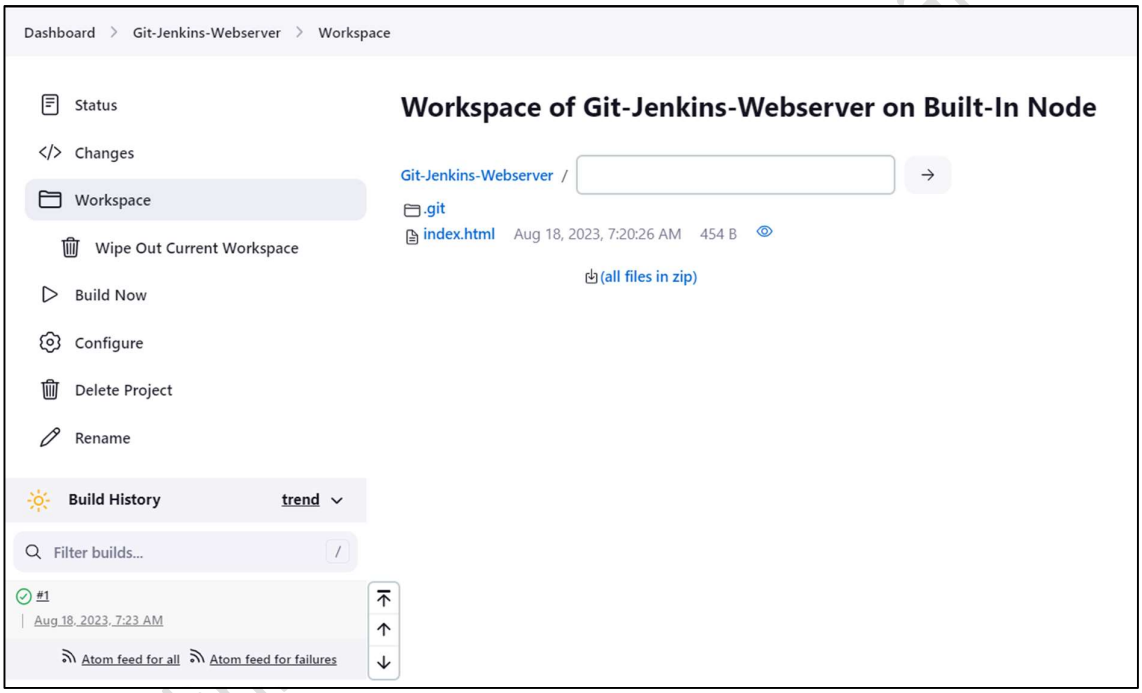


And after downloading the code we need to transfer the code in the Document Root. For this we need to write the command for copying the index.html file and store in /var/www/html.

Then Click on Build Now.



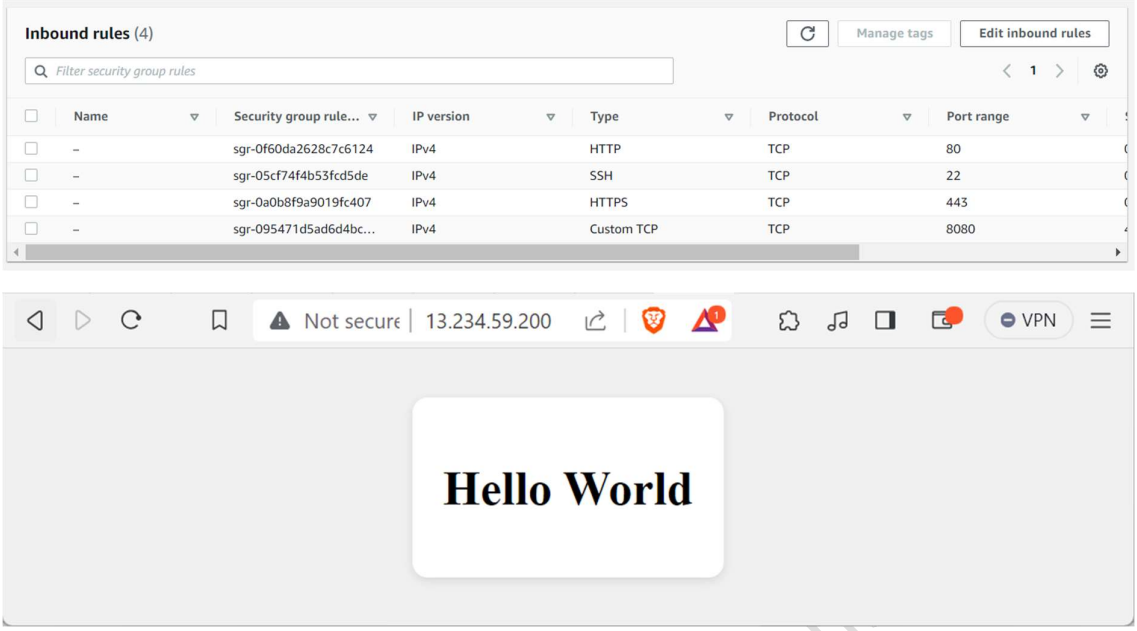
After Build, It will download the Code in My workspace.



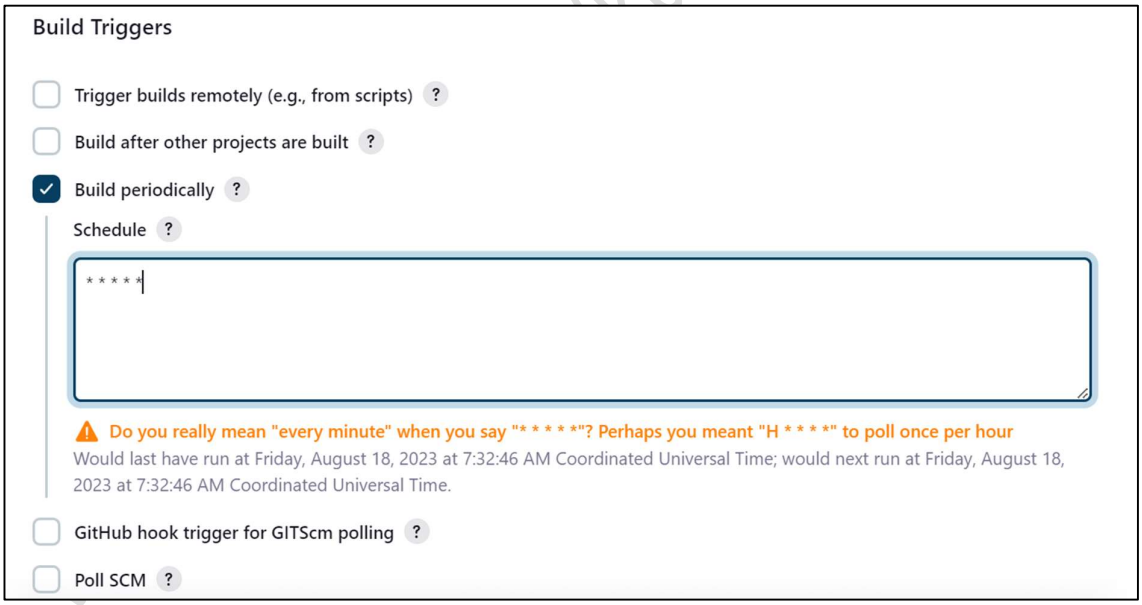
And transfer the code to my Document Root.

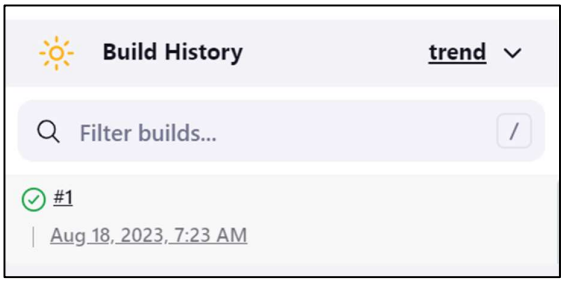
```
[root@ip-172-31-32-78 bin]# cd /var/www/html/
[root@ip-172-31-32-78 html]# ls
index.html
```

Now If I try to visit my webpage then I am getting the error, due to firewall. Because in My OS. I have blocked the inbound rule. From the EC2 -> Select Instance -> Go to Security Section -> Select Security Group -> Then Click on Inbound Rules -> Add HTTP, HTTPS.



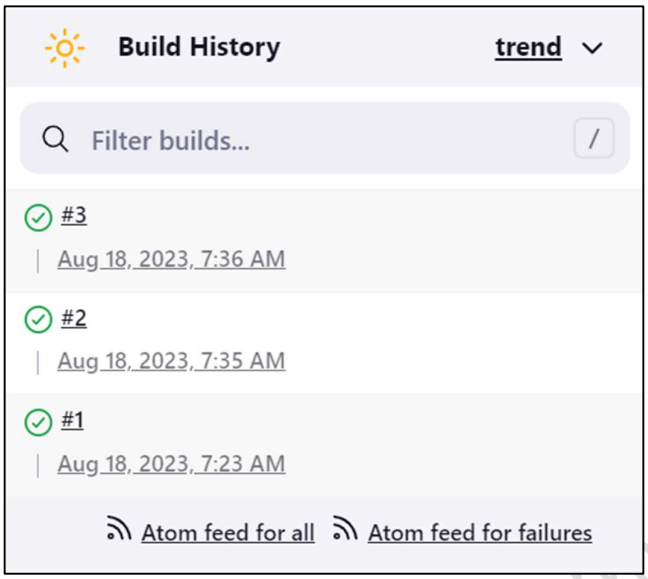
Now Here When Developer Push the code in Git Hub then I want to automatically Jenkins Go to Git Hub Download the Code and paste the Code in Document Root. For this We have Triggers available in Jenkins. One of the Triggers is Build Triggers. It will go to Git Hub Every minute and every time they download the entire code and paste in the Document Root.





When I save the settings, initially we can see the total 1 Job we have run. But after one minute we can see that multiple jobs are going to run. But this is not we want.

We want when developer push the code in GitHub then Jenkins will go to GitHub and download the code.

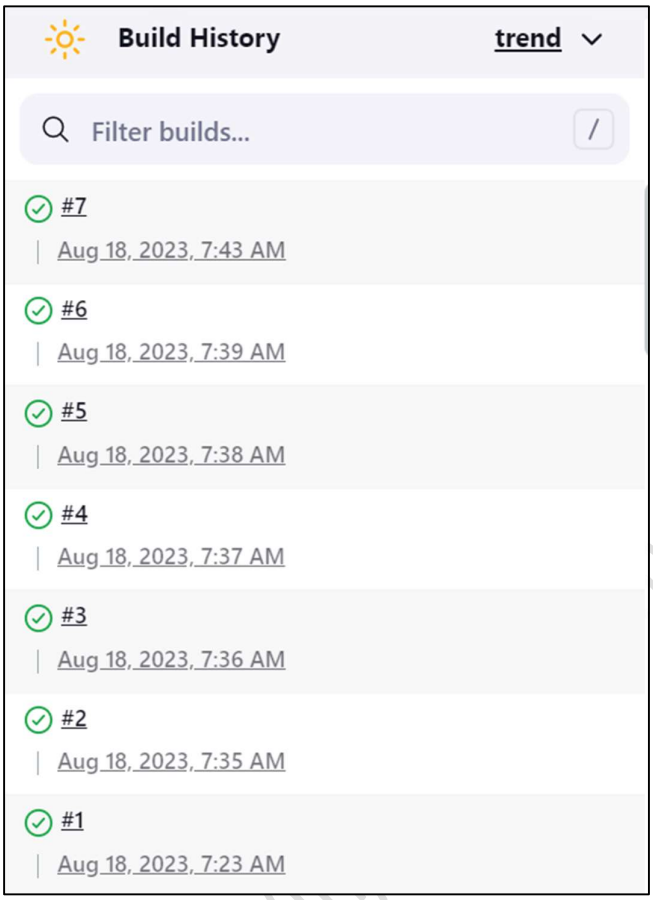


For Solving this problem, we have the Poll SCM available in the triggers. Here by using this, POLL SCM go to GitHub. Check there if any code changes happen then it will download the code and paste in document root. Else it won't do anything.

I have changed the code in my Webpage. And now I am going to push this code in the Git Hub.

```
<body>
  <div class="content">
    <h1>Hello World - Code Changed </h1>
  </div>
</body>
</html>
```

```
Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer (main)
$ git add .

Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer (main)
$ git commit -m "Code Changed 1 Time"
[main 8ad01ef] Code Changed 1 Time
 1 file changed, 1 insertion(+), 1 deletion(-)

Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer (main)
$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 307 bytes | 102.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Himanshu-kr-007/Git-Jenkins-Webserver.git
   4130673..8ad01ef  main -> main
branch 'main' set up to track 'origin/main'.
```



Here We can see the 6 build was happened on 7:39 APM and after that the next build happened on 7:43. Here, they have downloaded the latest code. And replaced with the older content.

We can also do one thing, as soon as the developer commits the code. Then the code will automatically push in the GitHub account.

In the Current Directory in my Local OS. Go inside the .git Directory -> then hooks -> Create the file with the name of post-commit.

```
Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer (main)
$ cd .git/

Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer/.git (GIT_DIR!)
$ ls
COMMIT_EDITMSG  HEAD    description  index  logs/     refs/
FETCH_HEAD      config  hooks/       info/  objects/

Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer/.git (GIT_DIR!)
$ cd hooks/

Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer/.git/hooks (GIT_DI
R!)
$ ls
applypatch-msg.sample*      pre-push.sample*
commit-msg.sample*          pre-rebase.sample*
fsmonitor-watchman.sample*  pre-receive.sample*
post-update.sample*         prepare-commit-msg.sample*
pre-applypatch.sample*      push-to-checkout.sample*
pre-commit.sample*          sendemail-validate.sample*
pre-merge-commit.sample*    update.sample*

Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer/.git/hooks (GIT_DI
R!)
$ vim post-commit
```

Inside the file, write the command git push.

```
#!/bin/bash
git push
```

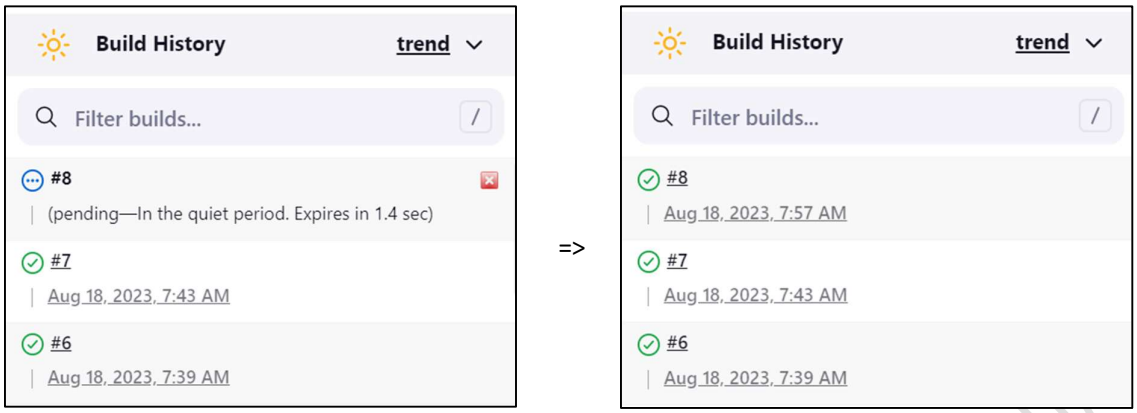
Then Switching the directory where I am writing my code. And changing the code.

```
Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer (main)
$ pwd
/c/Users/Himanshu Kumar/Desktop/Jenkins/WebServer

Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer (main)
$ vim index.html
```

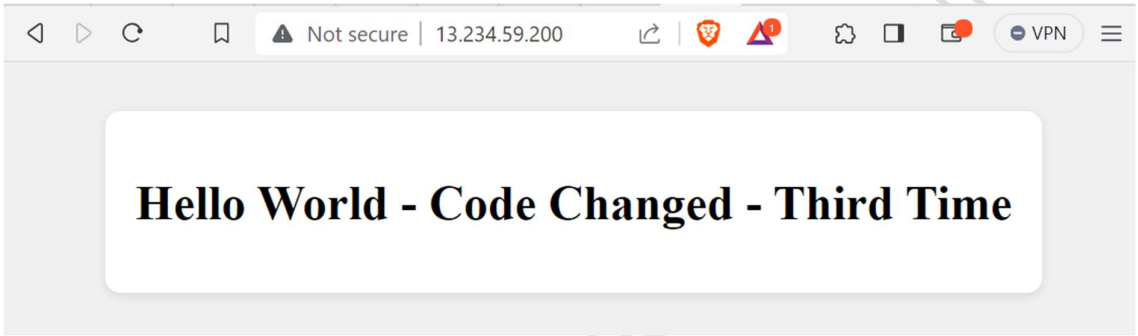
```
<body>
  <div class="content">
    <h1>Hello World - Code Changed - Third Time </h1>
  </div>
</body>
```

Committing the Code.

```
Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer (main)
$ git add .

Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer (main)
$ git commit index.html -m "Third Time Code Changed"
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 307 bytes | 153.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Himanshu-kr-007/Git-Jenkins-Webserver.git
   8ad01ef..2f2187f  main -> main
[main 2f2187f] Third Time Code Changed
 1 file changed, 1 insertion(+), 1 deletion(-)

Himanshu Kumar@AICPL-L128 MINGW64 ~/Desktop/Jenkins/WebServer (main)
$
```



Here In the webpage we can see the code is changed automatically.



THANK YOU