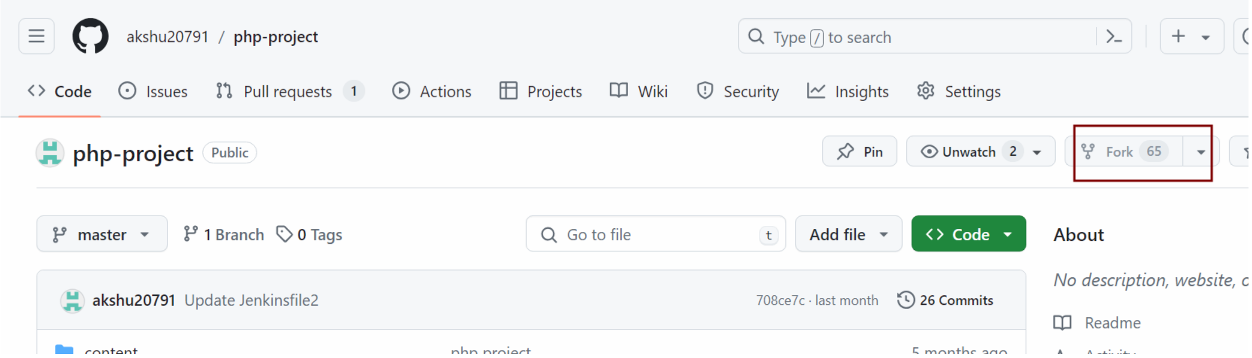
PHP PROJECT

Step1 : FORK THE REPO

https://github.com/akshu20791/php-project

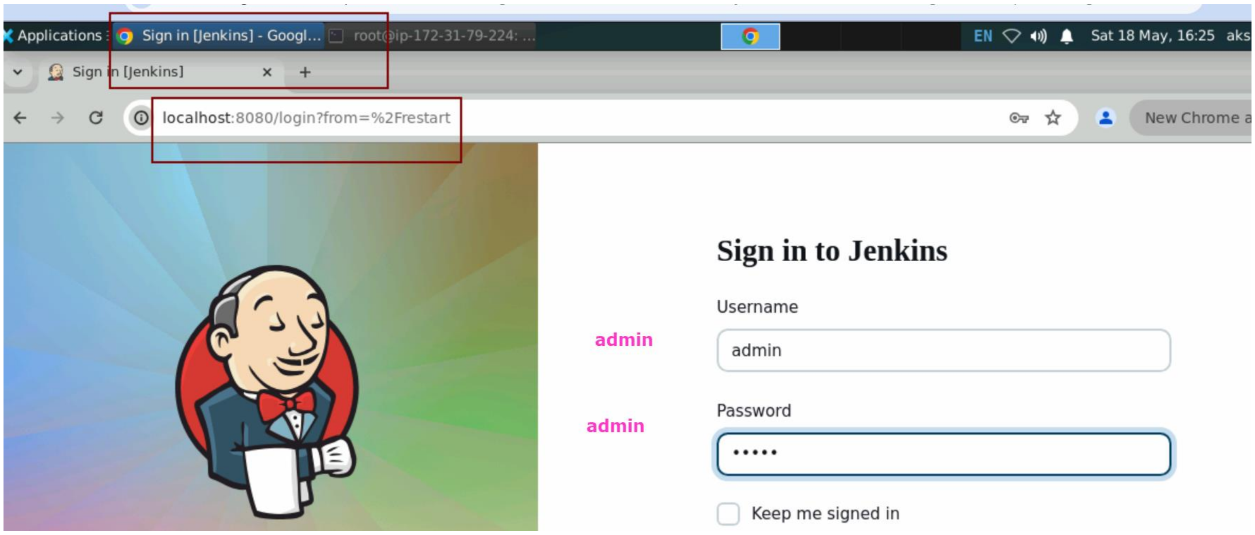


akshu20791 Update Jenkinsfile2

708ce7c · last month

26 Commits

Step 2: Go to Simplilearn machine -> Open Jenkins on port 8080



localhost:8080/login?from=%2Frestart

Sign in to Jenkins

Username

admin

admin

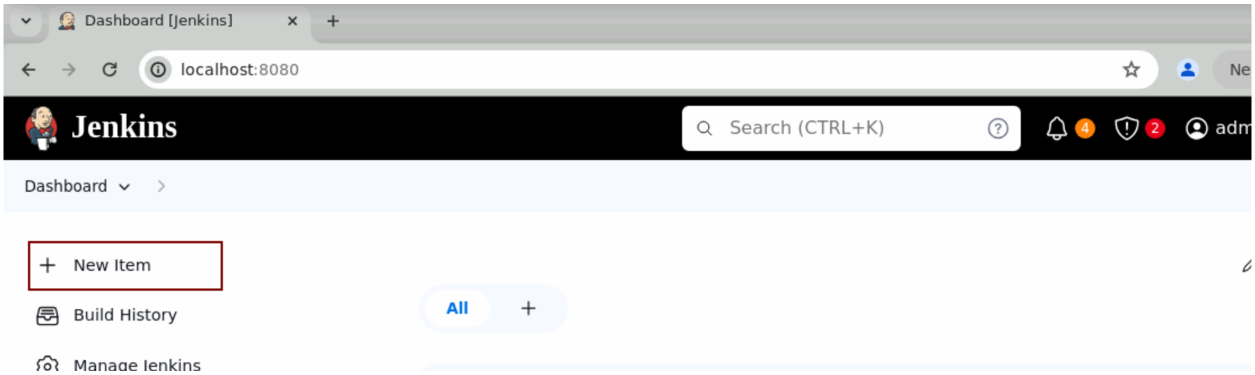
Password

admin

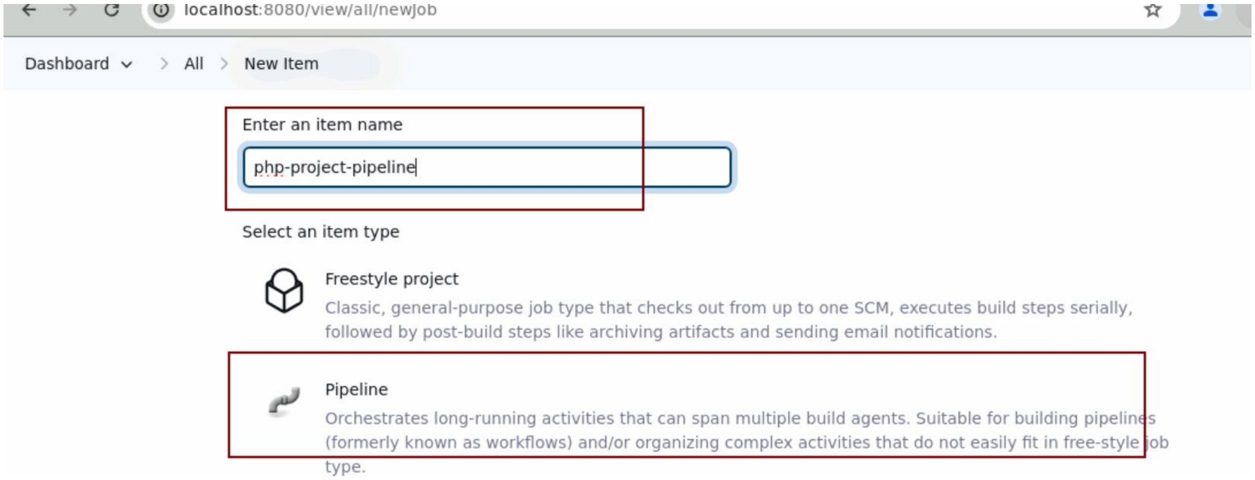
•••••

□ Keep me signed in

Step 3: Lets create a new pipeline



Give some name to the project and also select item type as Pipeline



Enter an item name: php-project-pipeline

Select an item type

Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline

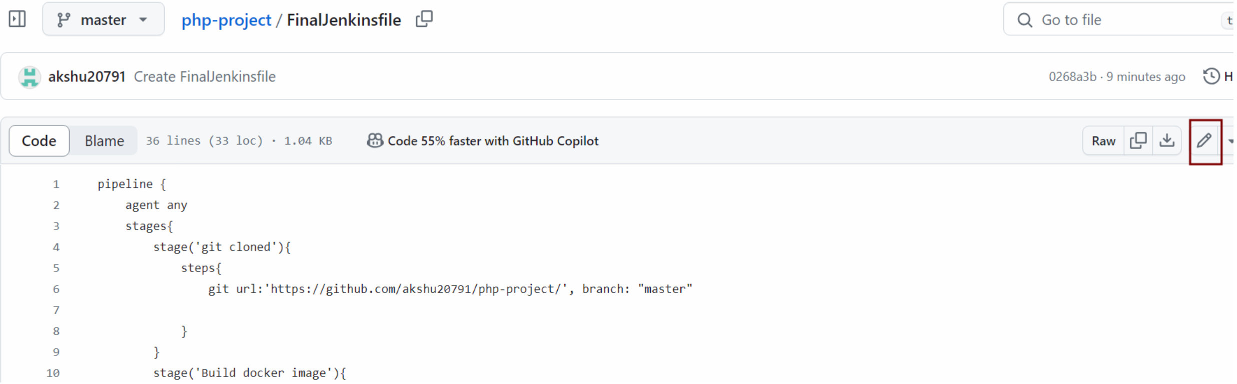
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Click on ok

Step 4: Go to the github repo which you have forked -> Open the https://github.com/akshu20791/php-project/blob/master/FinalJenkinsfile (FinalJenkinsfile)

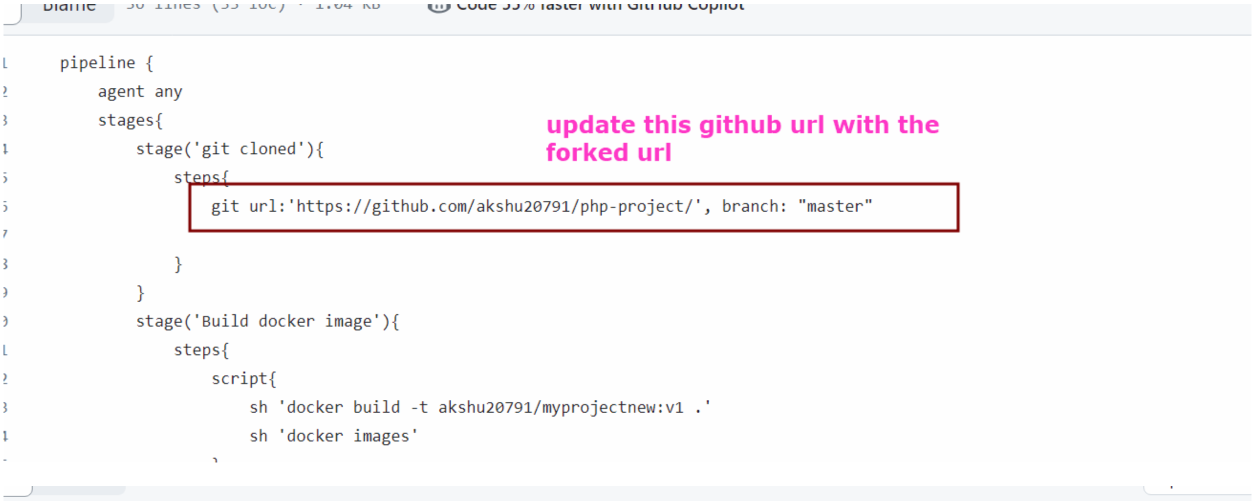
(if you don't find this file .create the file in your repo and copy the content in your file from my repo)

Open FinalJenkinsfile -> click on pencil button on the right corner

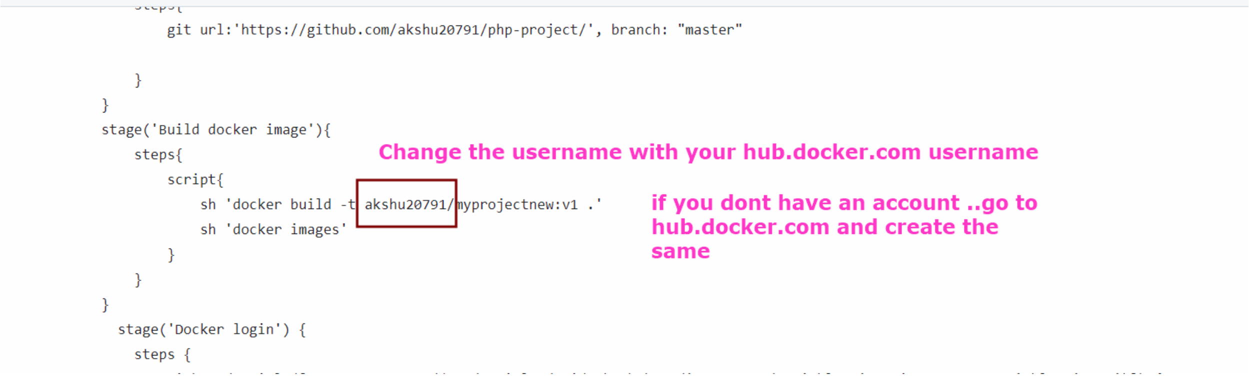

master
php-project / FinalJenkinsfile
Go to file
akshu20791 Create FinalJenkinsfile
0268a3b · 9 minutes ago
Code
Blame
36 lines (33 loc) · 1.04 KB
Code 55% faster with GitHub Copilot
Raw

```
pipeline {
    agent any
    stages{
        stage('git cloned'){
            steps{
                git url:'https://github.com/akshu20791/php-project/', branch: "master"

            }
        }
        stage('Build docker image'){
```

Now we need to make some changes in the FinalJenkinsfile

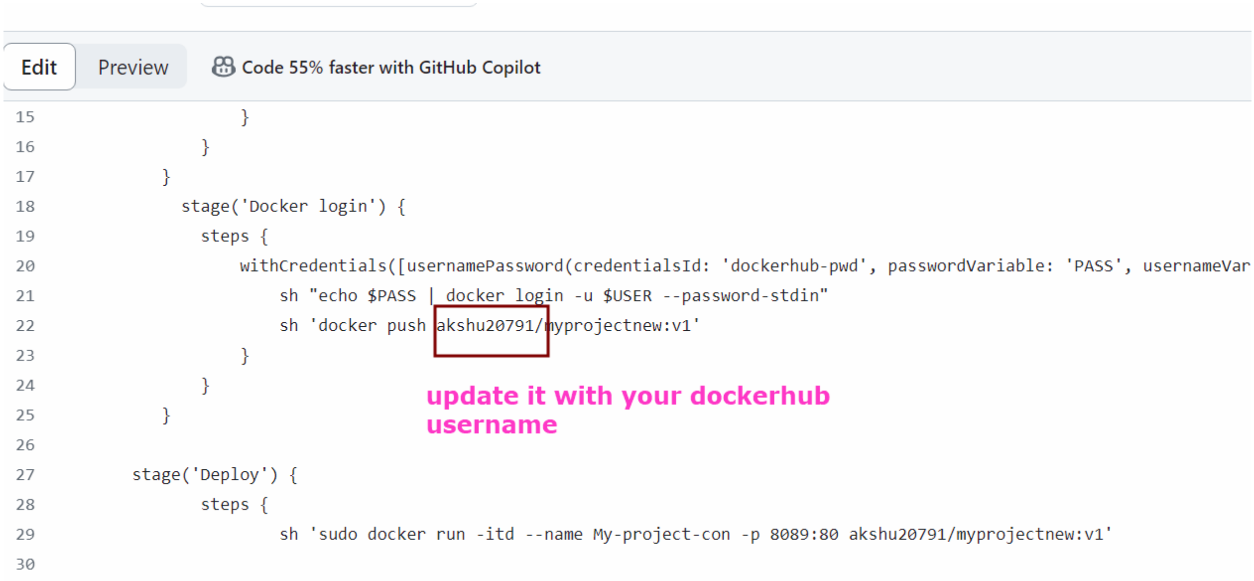

update this github url with the forked url

```
pipeline {
    agent any
    stages{
        stage('git cloned'){
            steps{
                git url:'https://github.com/akshu20791/php-project/', branch: "master"

            }
        }
        stage('Build docker image'){
            steps{
                script{
                    sh 'docker build -t akshu20791/myprojectnew:v1 .'
                    sh 'docker images'
```


Change the username with your hub.docker.com username

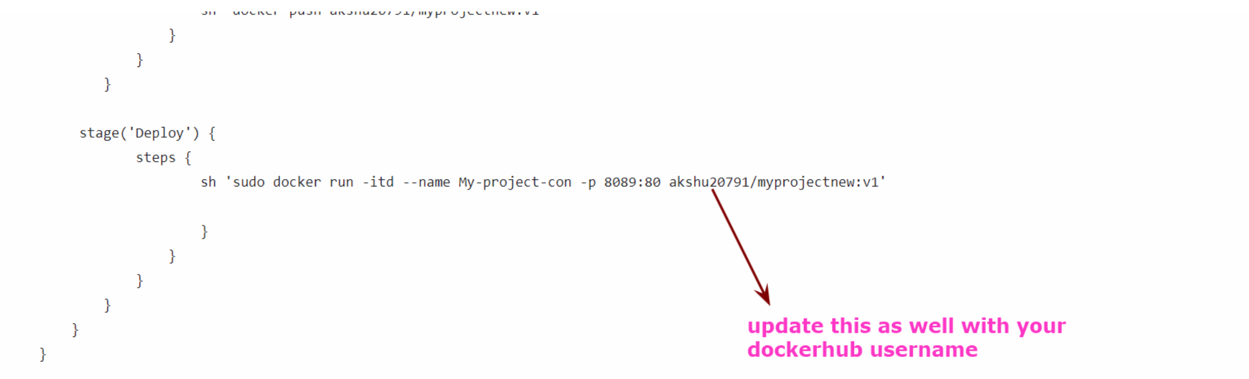
if you dont have an account ..go to hub.docker.com and create the same

```
                git url:'https://github.com/akshu20791/php-project/', branch: "master"

            }
        }
        stage('Build docker image'){
            steps{
                script{
                    sh 'docker build -t akshu20791/myprojectnew:v1 .'
                    sh 'docker images'
                }
            }
        }
          stage('Docker login') {
            steps {
```



```
                    }
                }
            }
              stage('Docker login') {
                steps {
                    withCredentials([usernamePassword(credentialsId: 'dockerhub-pwd', passwordVariable: 'PASS', usernameVar
                        sh "echo $PASS | docker login -u $USER --password-stdin"
                        sh 'docker push akshu20791/myprojectnew:v1'
                    }
                }
            }

         stage('Deploy') {
                steps {
                        sh 'sudo docker run -itd --name My-project-con -p 8089:80 akshu20791/myprojectnew:v1'

```

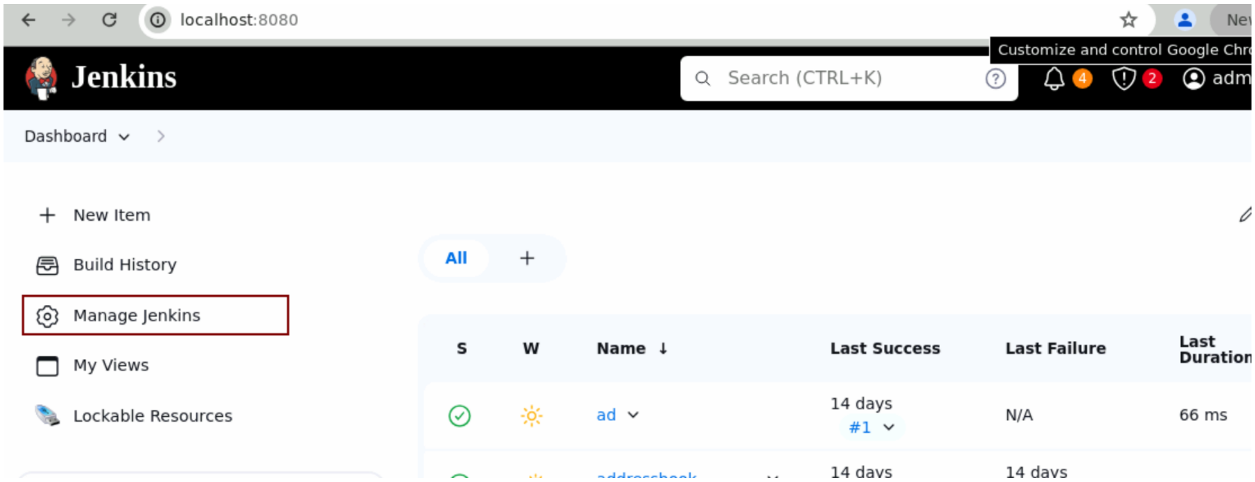
update it with your dockerhub username



```
                        sh 'docker push akshu20791/myprojectnew:v1'
                    }
                }
            }

        stage('Deploy') {
               steps {
                       sh 'sudo docker run -itd --name My-project-con -p 8089:80 akshu20791/myprojectnew:v1'

                       }
                   }
               }
           }
       }
   }
```

update this as well with your dockerhub username

Step 5: Go to Jenkins -> Manage Jenkins -> Credentials -> Click on global Credentials



localhost:8080

Customize and control Google Chr

Jenkins

Search (CTRL+K)

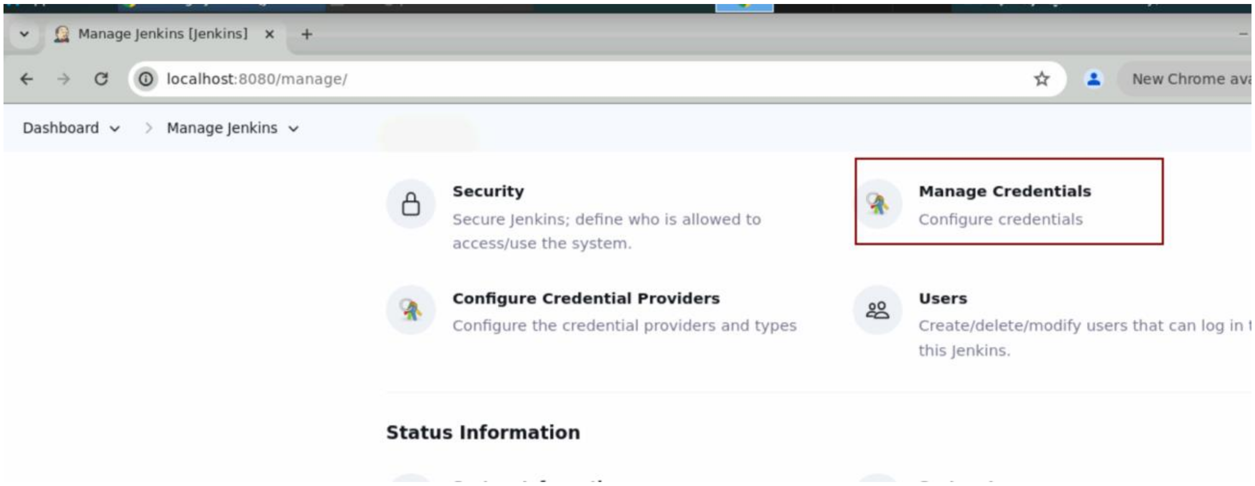
Dashboard

- New Item
- Build History
- Manage Jenkins
- My Views
- Lockable Resources

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	ad	14 days #1	N/A	66 ms
✓	☀	addresscheck	14 days	14 days	

Click on manage credentials



Manage Jenkins [Jenkins]

localhost:8080/manage/

Dashboard > Manage Jenkins

Security
Secure Jenkins; define who is allowed to access/use the system.

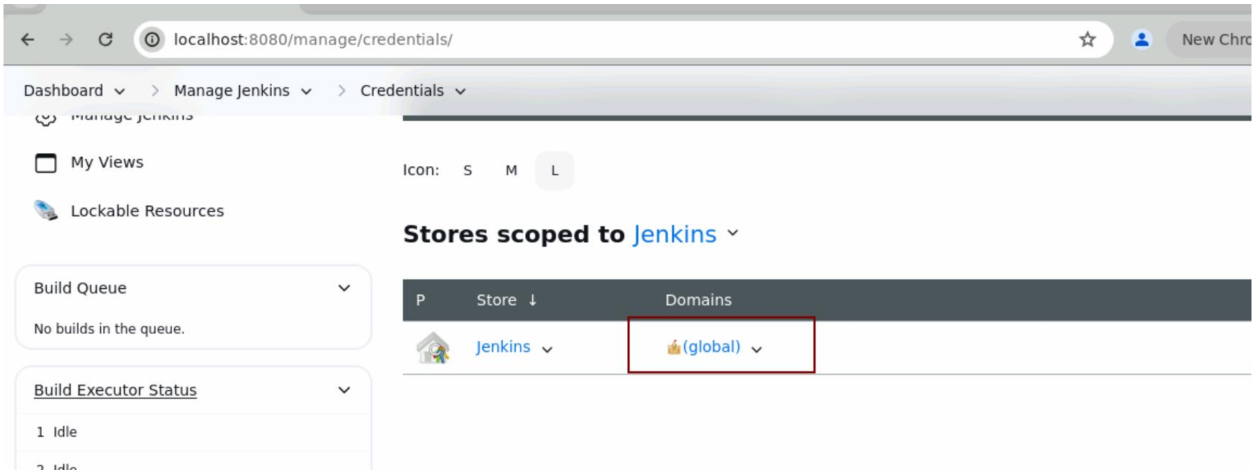
Manage Credentials
Configure credentials

Configure Credential Providers
Configure the credential providers and types

Users
Create/delete/modify users that can log in t this Jenkins.

Status Information

Click on global



localhost:8080/manage/credentials/

Dashboard > Manage Jenkins > Credentials

My Views

Lockable Resources

Build Queue

No builds in the queue.

Build Executor Status

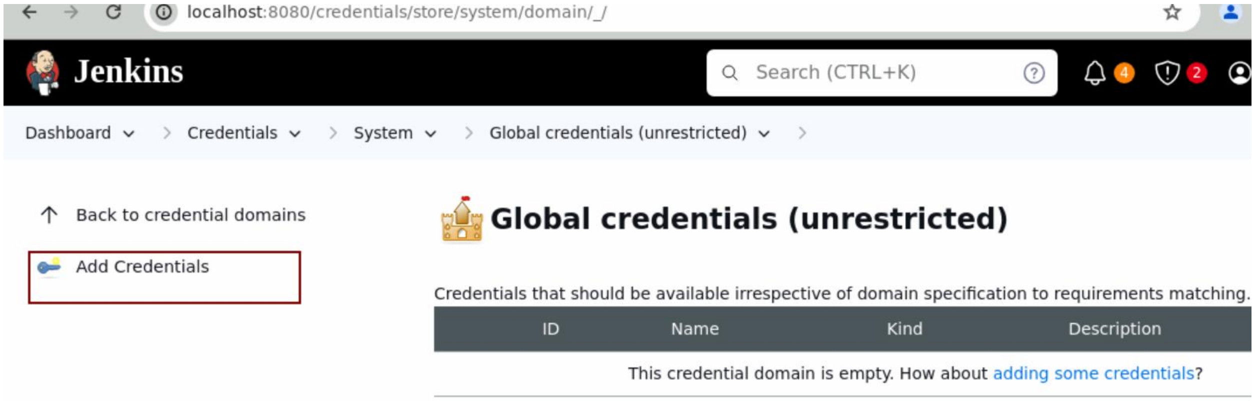
1 Idle

Icon: S M L

Stores scoped to Jenkins

P	Store ↓	Domains
	Jenkins	(global)

Click on Add credentials



localhost:8080/credentials/store/system/domain/_/

Jenkins

Search (CTRL+K)

Dashboard > Credentials > System > Global credentials (unrestricted)

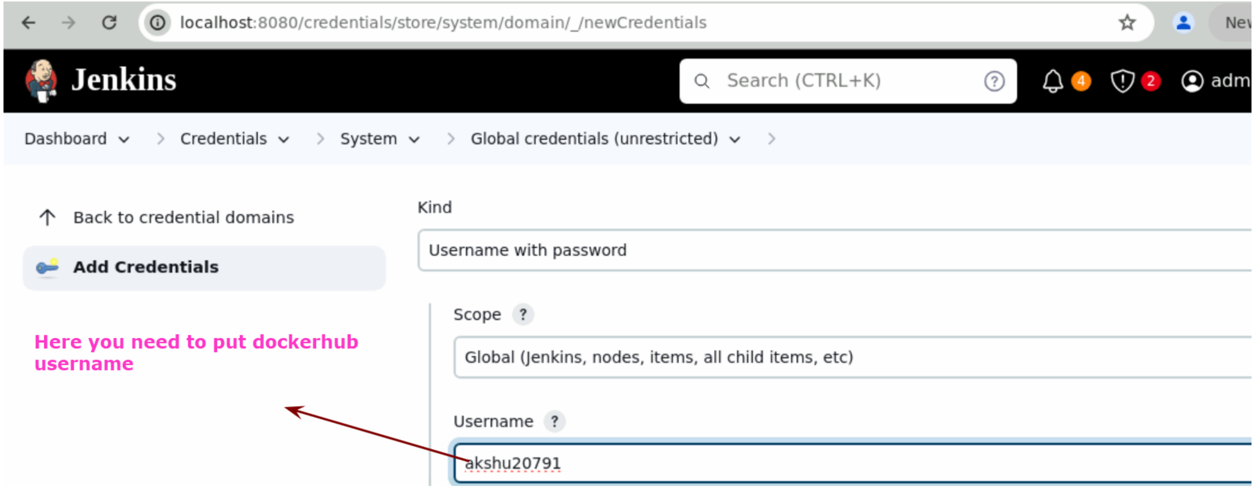
Back to credential domains

Add Credentials

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
This credential domain is empty. How about adding some credentials?			



localhost:8080/credentials/store/system/domain/_/newCredentials

Jenkins

Search (CTRL+K)

Dashboard > Credentials > System > Global credentials (unrestricted)

Back to credential domains

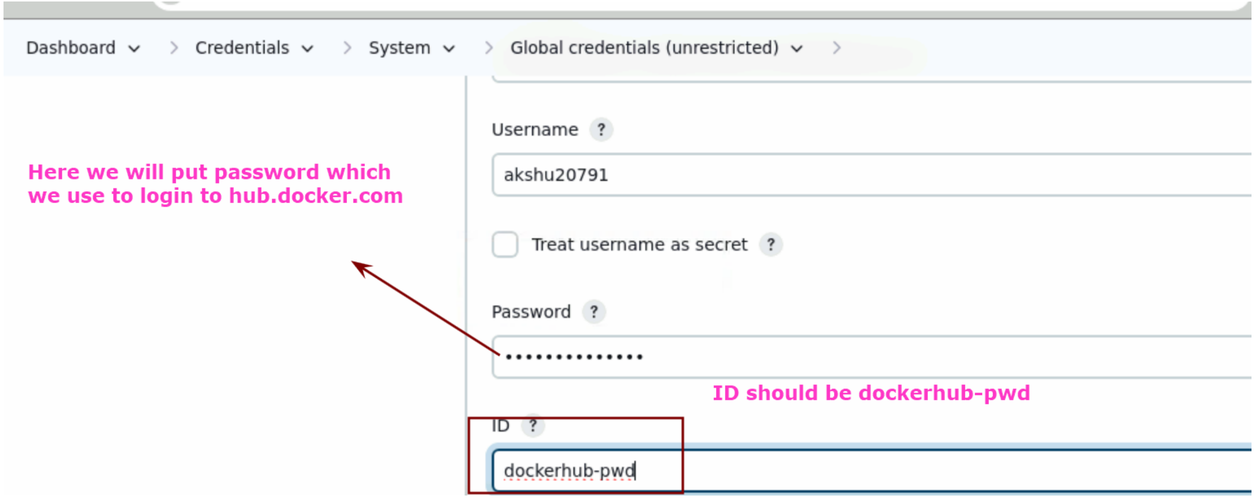
Add Credentials

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: akshu20791

Here you need to put dockerhub username



Username ?

akshu20791

□ Treat username as secret ?

Password ?

••••••••••••••

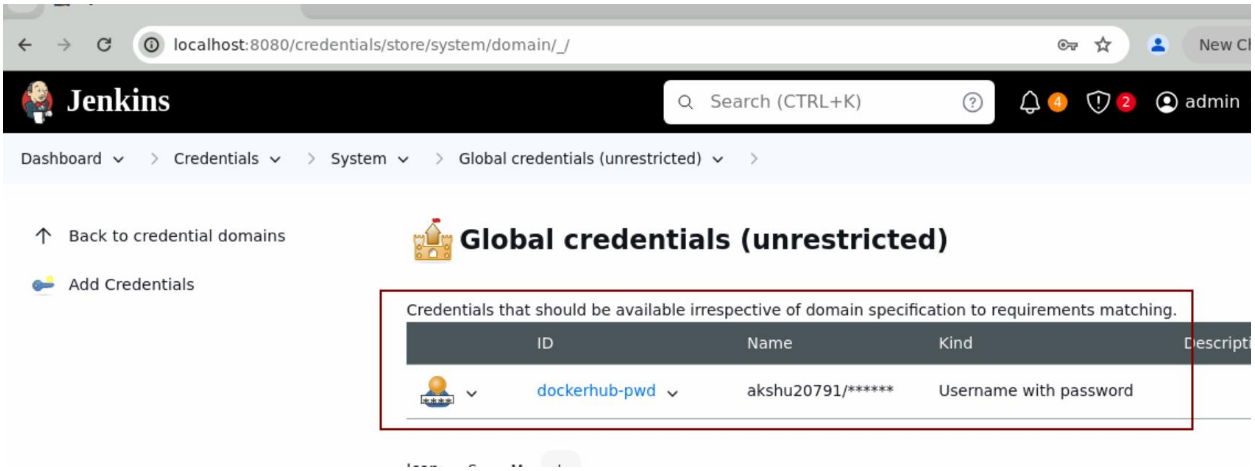
ID ?

dockerhub-pwd

Here we will put password which we use to login to hub.docker.com

ID should be dockerhub-pwd

Click on ok

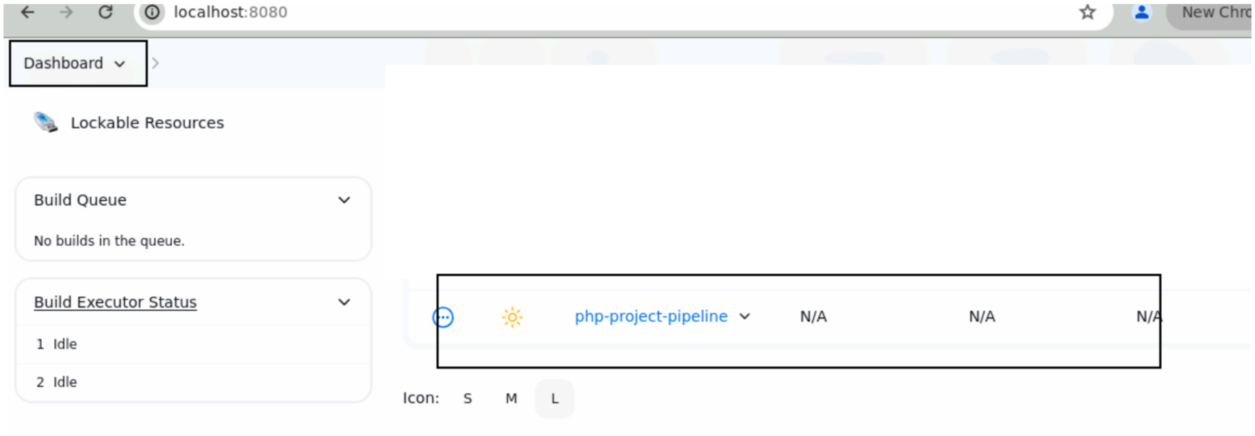

localhost:8080/credentials/store/system/domain/_/
New Cl
Jenkins
Search (CTRL+K)
4
2
admin
Dashboard
Credentials
System
Global credentials (unrestricted)
Back to credential domains
Add Credentials

Global credentials (unrestricted)

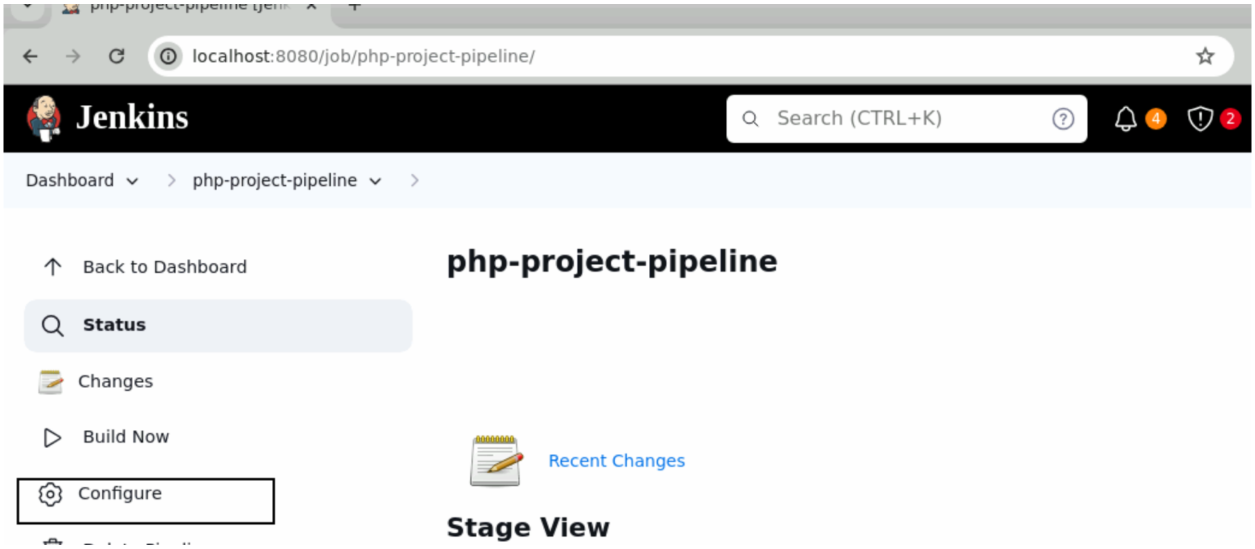
Credentials that should be available irrespective of domain specification to requirements matching.

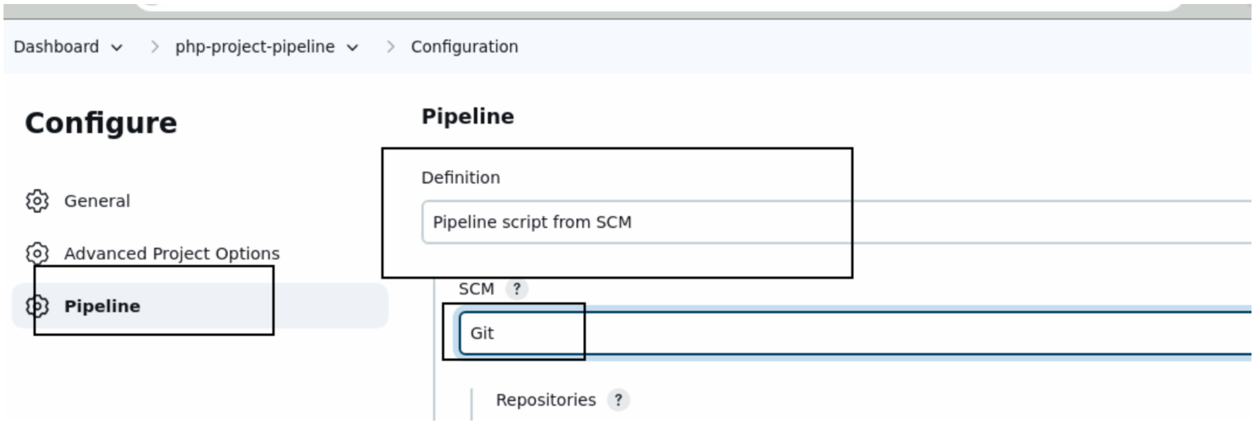
	ID	Name	Kind	Descripti
	dockerhub-pwd	akshu20791/******	Username with password	

Step 6: Now lets configure the pipeline



Click on the pipeline





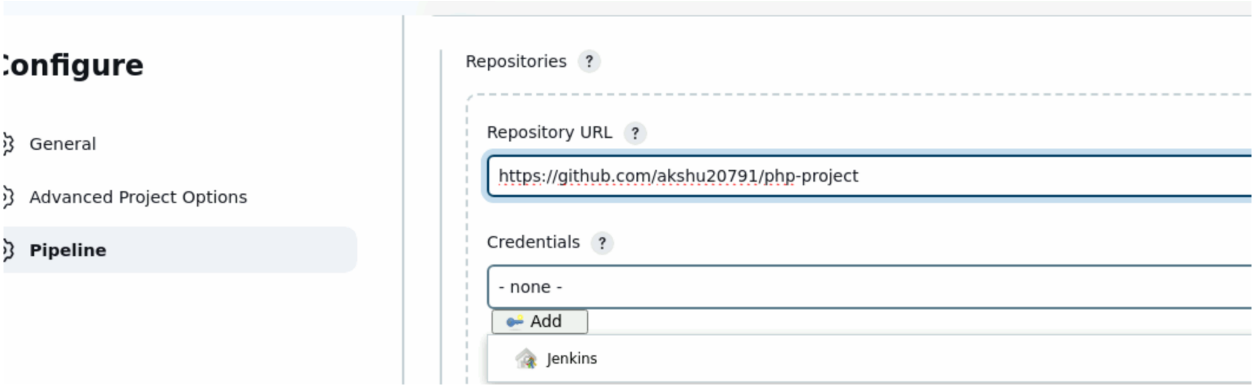
Definition

Pipeline script from SCM

SCM

Git

In repository it should be your forked repository

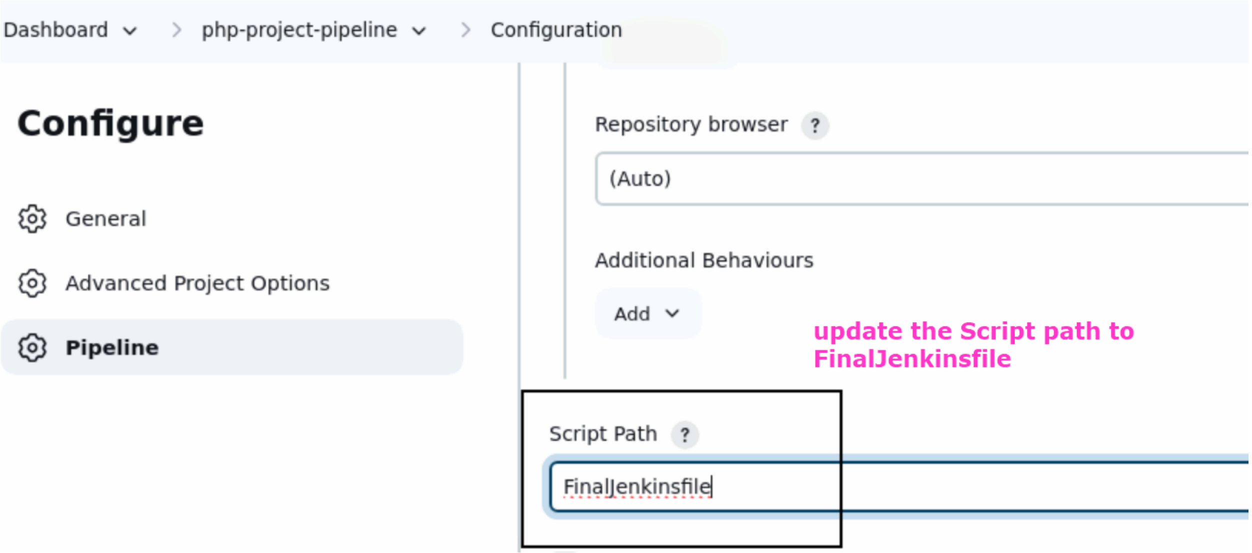


:onfigure

Repositories

Repository URL: https://github.com/akshu20791/php-project

Credentials: - none -



Dashboard > php-project-pipeline > Configuration

Configure

Repository browser: (Auto)

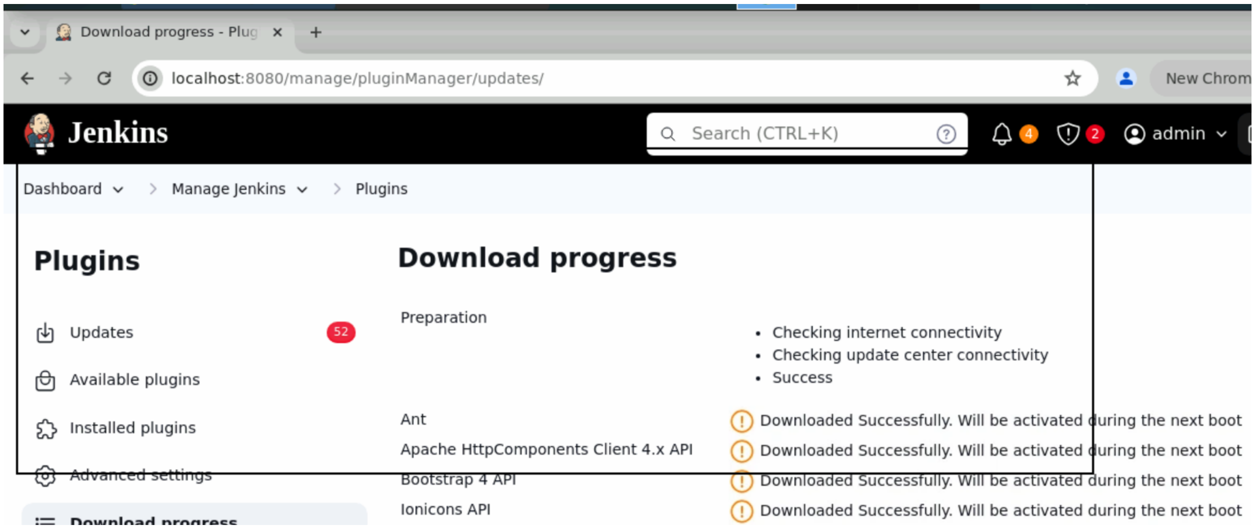
Additional Behaviours

update the Script path to FinalJenkinsfile

Script Path: FinalJenkinsfile

Save

Step 7: Go to manage Jenkins -> Plugins -> updates -> Select all and update all the plugins



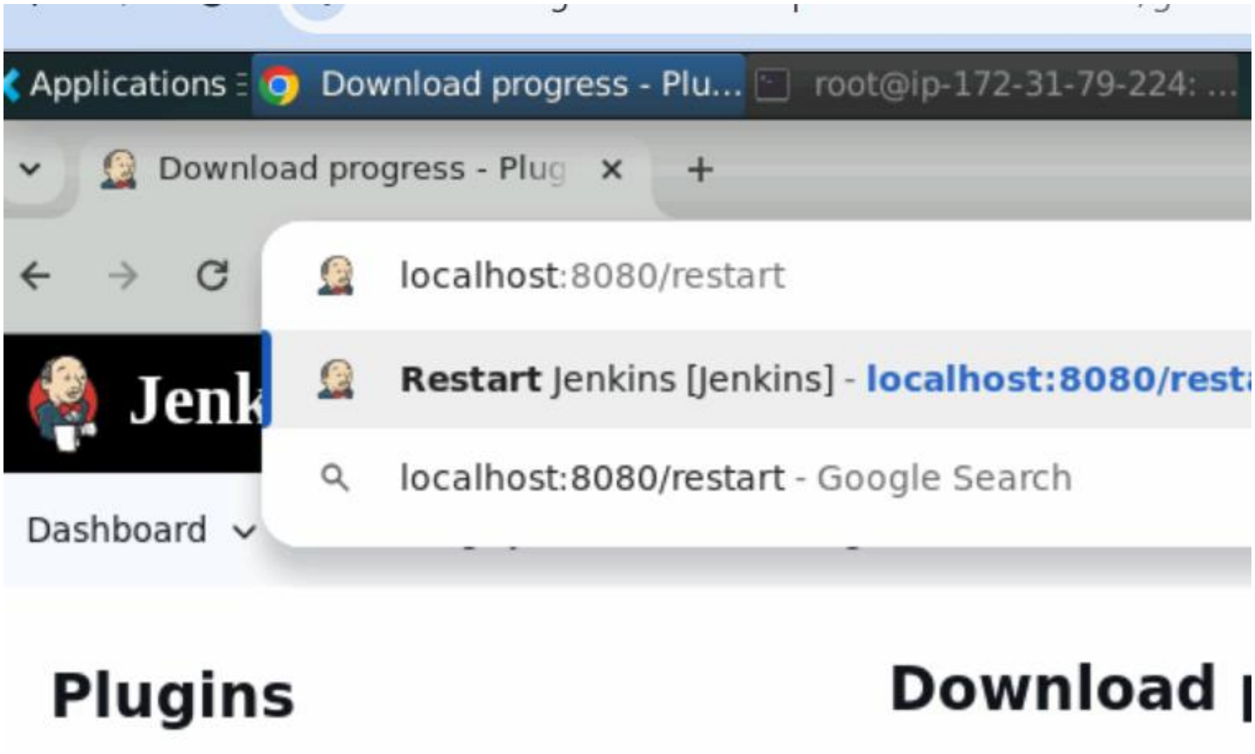
Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Ant — Downloaded Successfully. Will be activated during the next boot
Apache HttpComponents Client 4.x API — Downloaded Successfully. Will be activated during the next boot
Bootstrap 4 API — Downloaded Successfully. Will be activated during the next boot
Ionicons API — Downloaded Successfully. Will be activated during the next boot

Lets restart the Jenkins



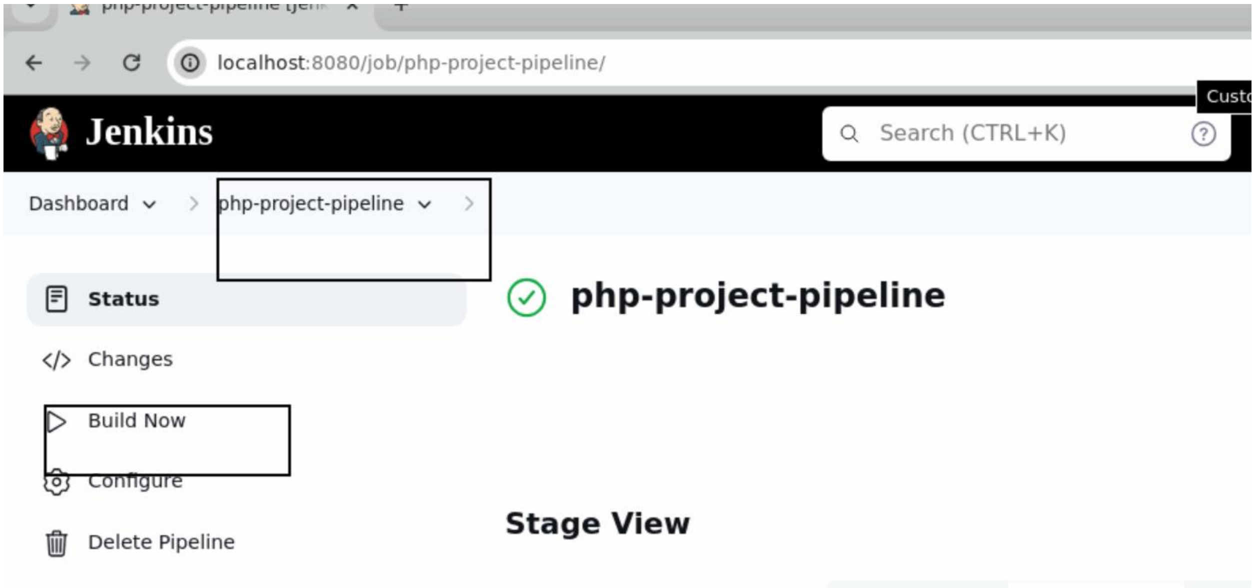
localhost:8080/restart

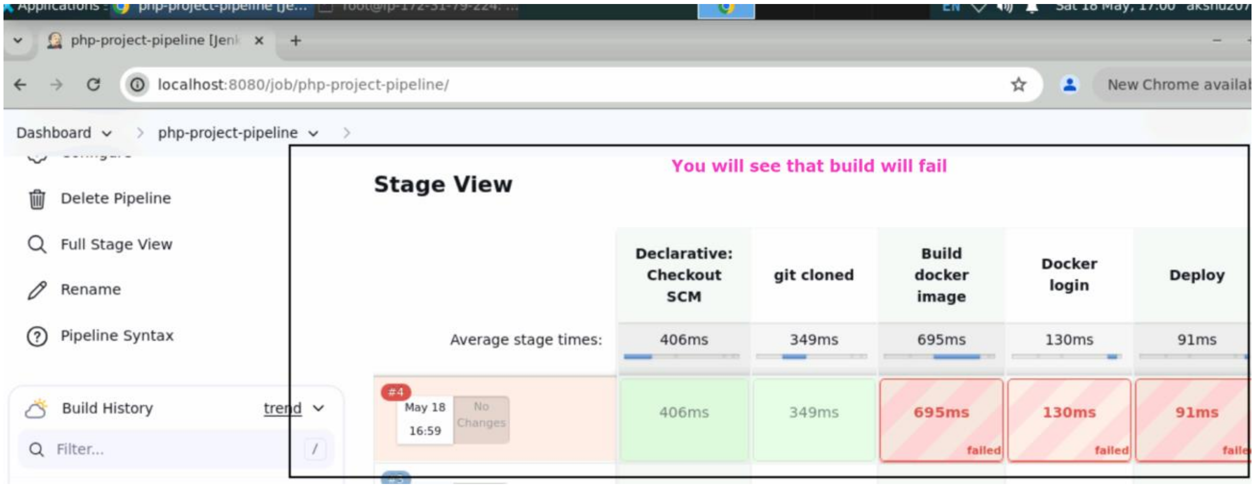
Restart Jenkins [Jenkins] - localhost:8080/resta

localhost:8080/restart - Google Search

Relogin to Jenkins after restart

Step 8: Go back to the pipeline and click on Build now



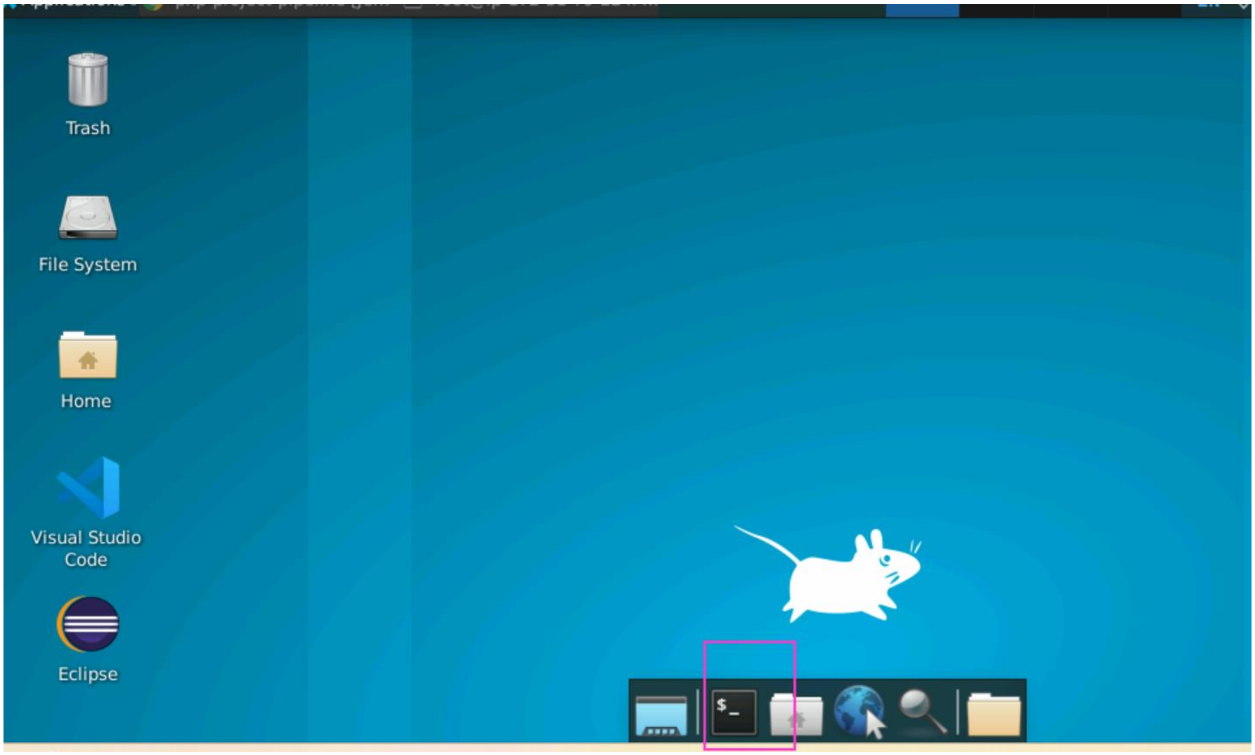


You will see that build will fail

Stage View

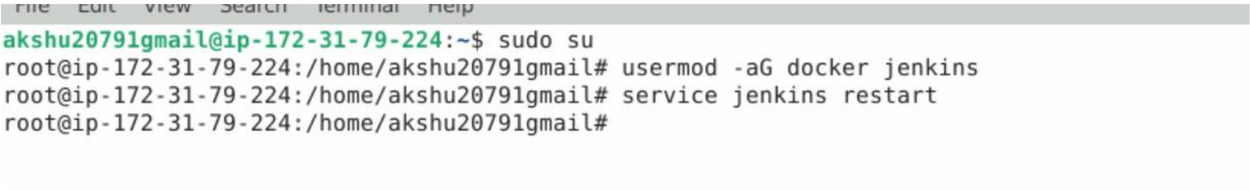
	Declarative: Checkout SCM	git cloned	Build docker image	Docker login	Deploy
Average stage times:	406ms	349ms	695ms	130ms	91ms
#4 May 18 16:59 No Changes	406ms	349ms	695ms failed	130ms failed	91ms faile

Step 10: Now open the machine terminal . The Jenkins does not have permission to access docker. We now have to give Jenkins to perform the same



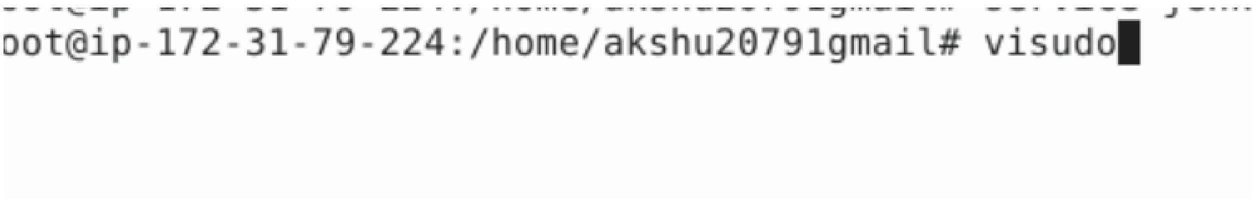
# sudo su

# usermod -aG docker Jenkins



```
akshu20791gmail@ip-172-31-79-224:~$ sudo su
root@ip-172-31-79-224:/home/akshu20791gmail# usermod -aG docker jenkins
root@ip-172-31-79-224:/home/akshu20791gmail# service jenkins restart
root@ip-172-31-79-224:/home/akshu20791gmail#
```

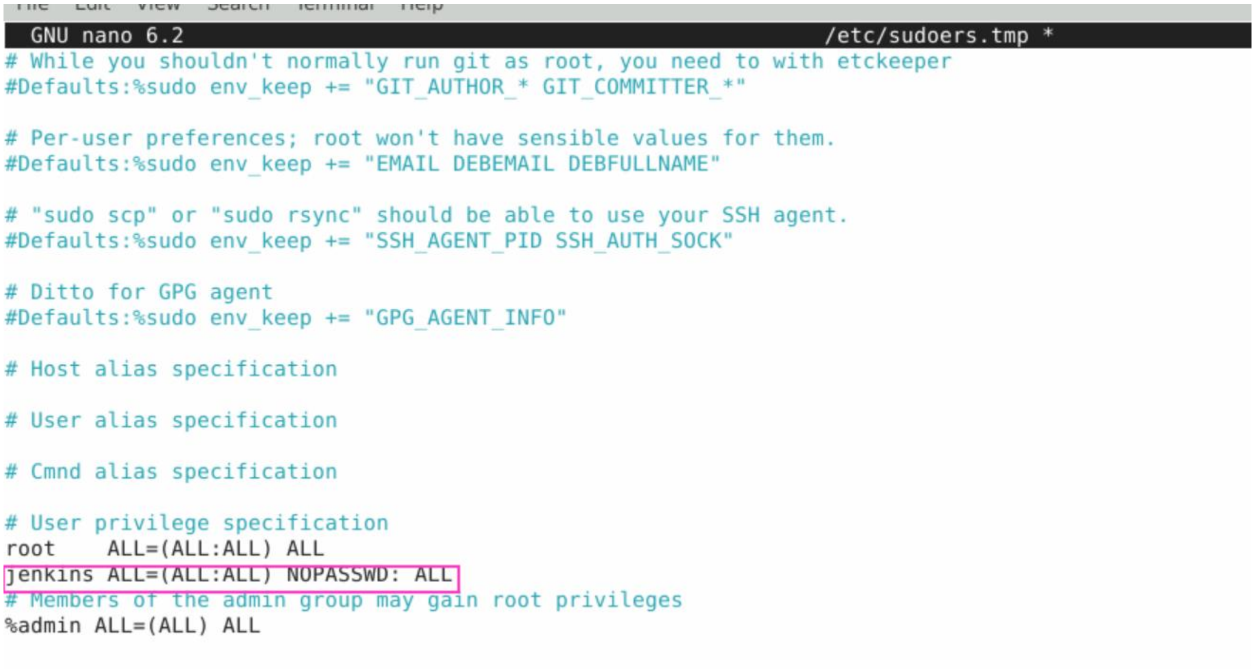
# visudo



```
oot@ip-172-31-79-224:/home/akshu20791gmail# visudo
```

add the line

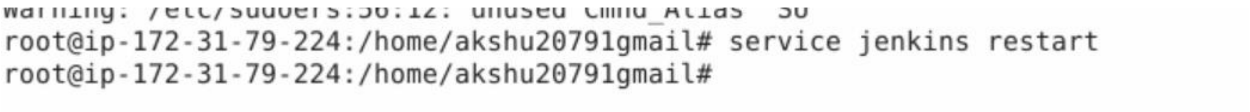
# jenkins ALL=(ALL:ALL) NOPASSWD: ALL



```
  GNU nano 6.2                                                /etc/sudoers.tmp *
# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"

# Per-user preferences; root won't have sensible values for them.
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
jenkins ALL=(ALL:ALL) NOPASSWD: ALL
# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
```

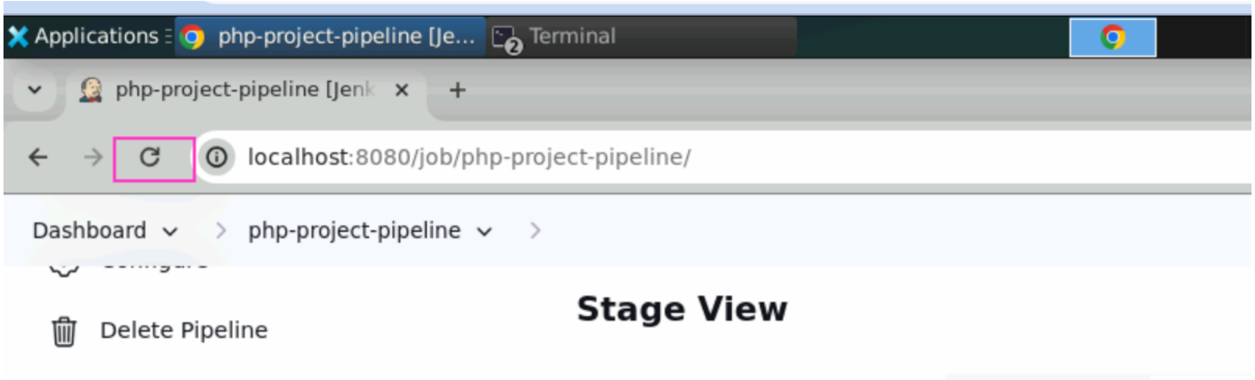
To come out of this file press ctrl X -> y -> enter

# service jenkins restart



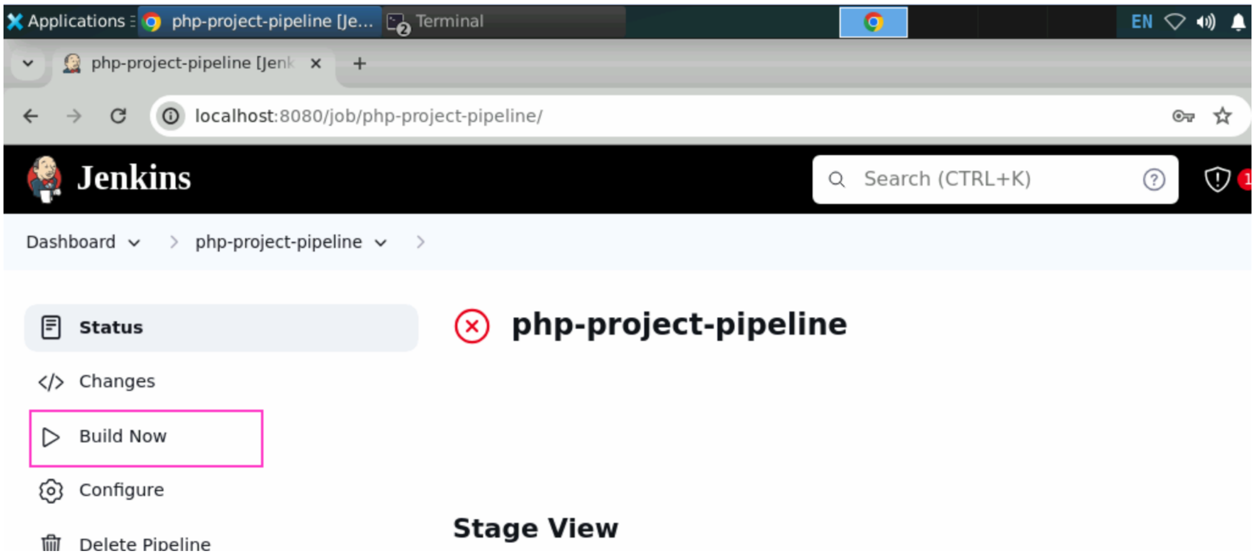
```
warning: /etc/sudoers:56:12: unused Cmnd_Alias "SU"
root@ip-172-31-79-224:/home/akshu20791gmail# service jenkins restart
root@ip-172-31-79-224:/home/akshu20791gmail#
```

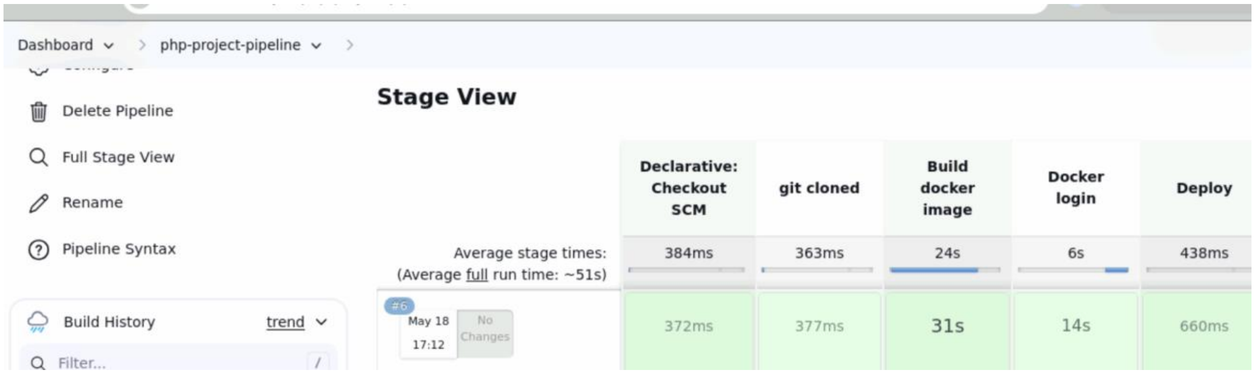
Step11 : Refresh Jenkins


Applications
php-project-pipeline [Je...
Terminal
php-project-pipeline [Jenk
localhost:8080/job/php-project-pipeline/
Dashboard
php-project-pipeline
Configure
Delete Pipeline
Stage View

And login again

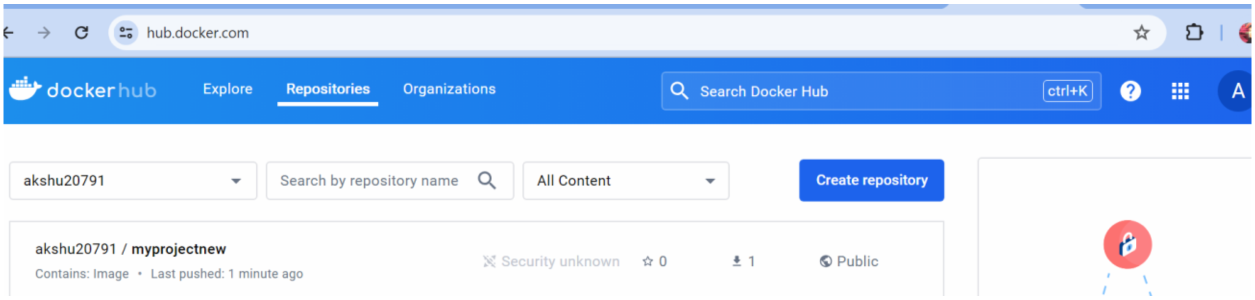
Step 12: Go back to your pipeline and build the JenkinsPipeline again





	Declarative: Checkout SCM	git cloned	Build docker image	Docker login	Deploy
Average stage times: (Average full run time: ~51s)	384ms	363ms	24s	6s	438ms
#6 May 18 17:12 No Changes	372ms	377ms	31s	14s	660ms

Now go to hub.docker.com and check if the image is pushed to dockerhub or not



akshu20791 / myprojectnew

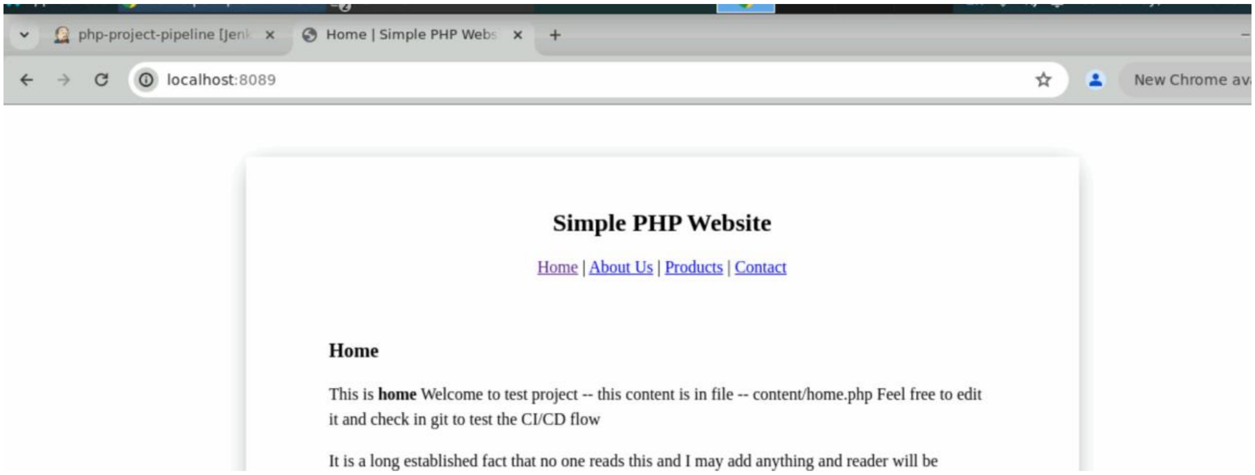
Contains: Image • Last pushed: 1 minute ago

Security unknown ☆ 0 ⭳ 1 Public

(Ensure that you are logged into hub.docker.com)

Finally, we will check if our application is live or not

Open a new tab in chrome of the machine and write localhost:8089



Simple PHP Website

Home | About Us | Products | Contact

Home

This is **home** Welcome to test project -- this content is in file -- content/home.php Feel free to edit it and check in git to test the CI/CD flow

It is a long established fact that no one reads this and I may add anything and reader will be

###