DevOps Integration using Jenkins Pipeline

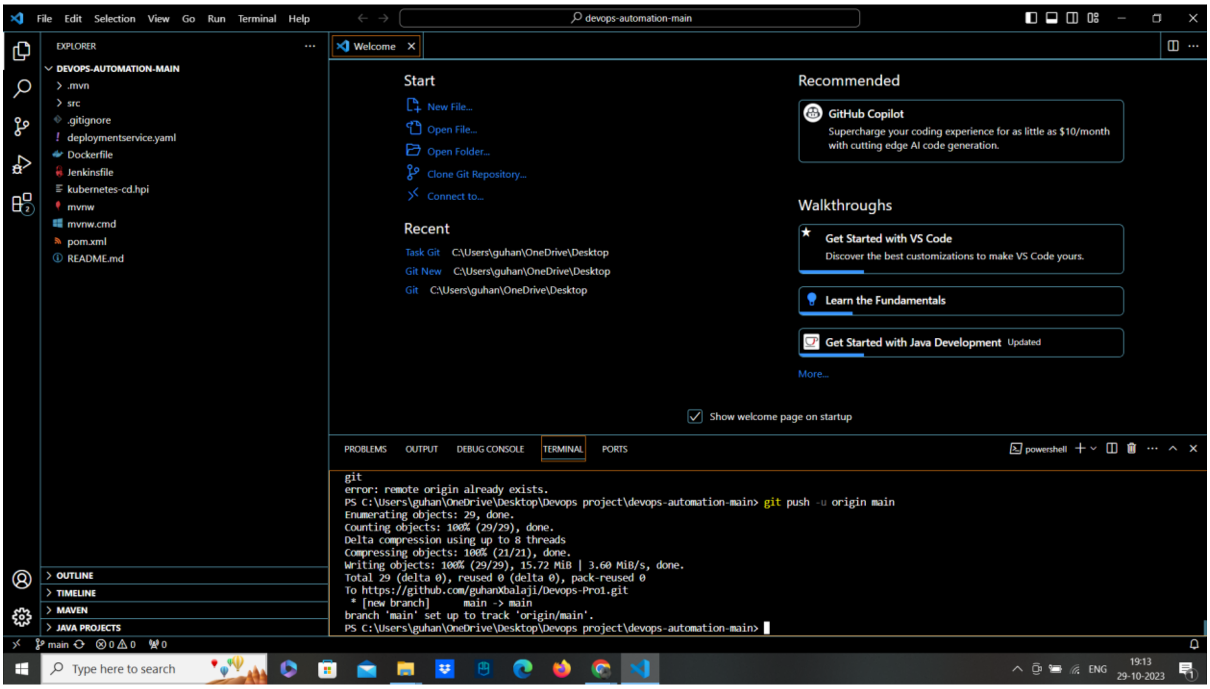
Name: S. Guhan Balaji

In this project we are going to build and push docker image using Jenkins pipeline. Building and pushing a Docker image using a Jenkins pipeline is a common and useful task for automating the deployment process.

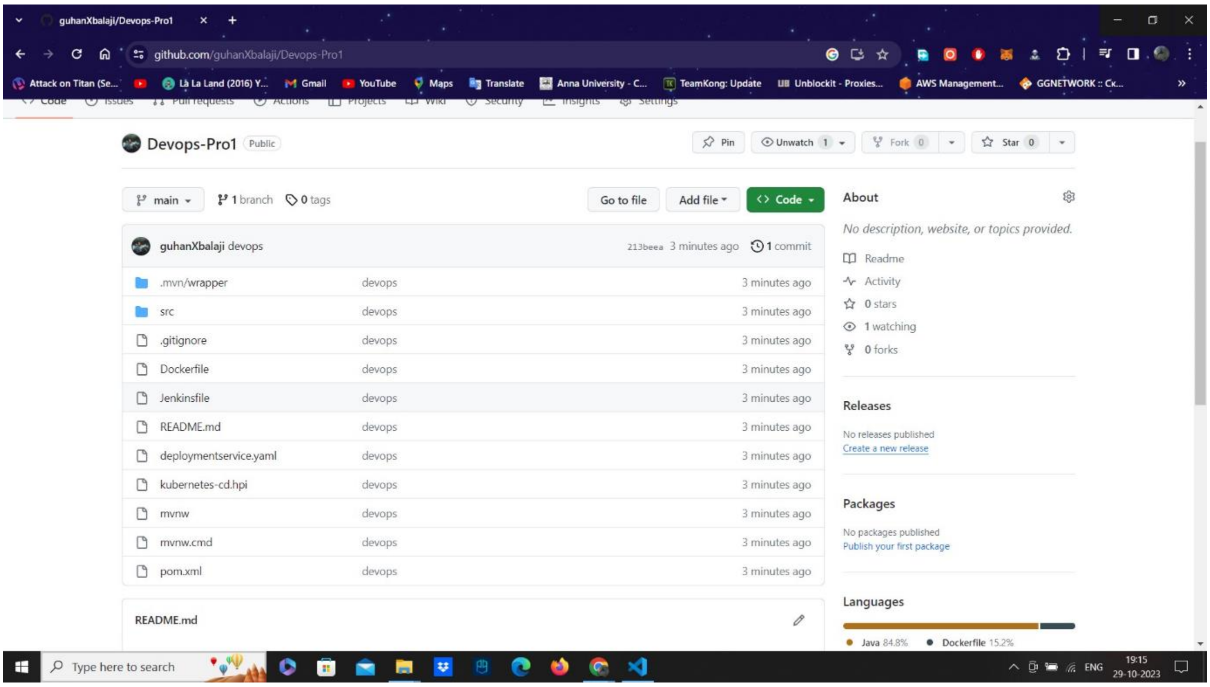
Steps Involved:

Step 1:

- Create a git hub repository for your Application.
- Open the VS code and the open the application folder. Use git init to initialize the git repository.
- Add and commit the files. And add remote origin with url of git repository.
- Then Push to the master using push -u origin master.
- We finally committed the file of our application to our git hub.


File Edit Selection View Go Run Terminal Help
devops-automation-main
EXPLORER
DEVOPS-AUTOMATION-MAIN
.mvn
src
.gitignore
deploymentservice.yaml
Dockerfile
Jenkinsfile
kubernetes-cd.hpi
mvnw
mvnw.cmd
pom.xml
README.md
OUTLINE
TIMELINE
MAVEN
JAVA PROJECTS
Welcome
Start
New File...
Open File...
Open Folder...
Clone Git Repository...
Connect to...
Recent
Task Git C:\Users\guhan\OneDrive\Desktop
Git New C:\Users\guhan\OneDrive\Desktop
Git C:\Users\guhan\OneDrive\Desktop
Recommended
GitHub Copilot
Supercharge your coding experience for as little as $10/month with cutting edge AI code generation.
Walkthroughs
Get Started with VS Code
Discover the best customizations to make VS Code yours.
Learn the Fundamentals
Get Started with Java Development Updated
More...
☑ Show welcome page on startup
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
main
Type here to search
ENG
19:13
29-10-2023

```
git
error: remote origin already exists.
PS C:\Users\guhan\OneDrive\Desktop\Devops project\devops-automation-main> git push -u origin main
Enumerating objects: 29, done.
Counting objects: 100% (29/29), done.
Delta compression using up to 8 threads
Compressing objects: 100% (21/21), done.
Writing objects: 100% (29/29), 15.72 MiB | 3.60 MiB/s, done.
Total 29 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/guhanXbalaji/Devops-Pro1.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\guhan\OneDrive\Desktop\Devops project\devops-automation-main>
```


guhanXbalaji/Devops-Pro1
github.com/guhanXbalaji/Devops-Pro1
Attack on Titan (Se... La La Land (2016) Y... Gmail YouTube Maps Translate Anna University - C... TeamKong: Update Unblockit - Proxies... AWS Management... GGNETWORK :: Cx...
Code Issues Pull requests Actions Projects Wiki Security Insights Settings
Devops-Pro1 Public
Pin Unwatch 1 Fork 0 Star 0
main 1 branch 0 tags
Go to file Add file Code
Type here to search
ENG 19:15 29-10-2023

guhanXbalaji devops 213beea 3 minutes ago 1 commit

.mvn/wrapper	devops	3 minutes ago
src	devops	3 minutes ago
.gitignore	devops	3 minutes ago
Dockerfile	devops	3 minutes ago
Jenkinsfile	devops	3 minutes ago
README.md	devops	3 minutes ago
deploymentservice.yaml	devops	3 minutes ago
kubernetes-cd.hpi	devops	3 minutes ago
mvnw	devops	3 minutes ago
mvnw.cmd	devops	3 minutes ago
pom.xml	devops	3 minutes ago

README.md

About

No description, website, or topics provided.

Readme
Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

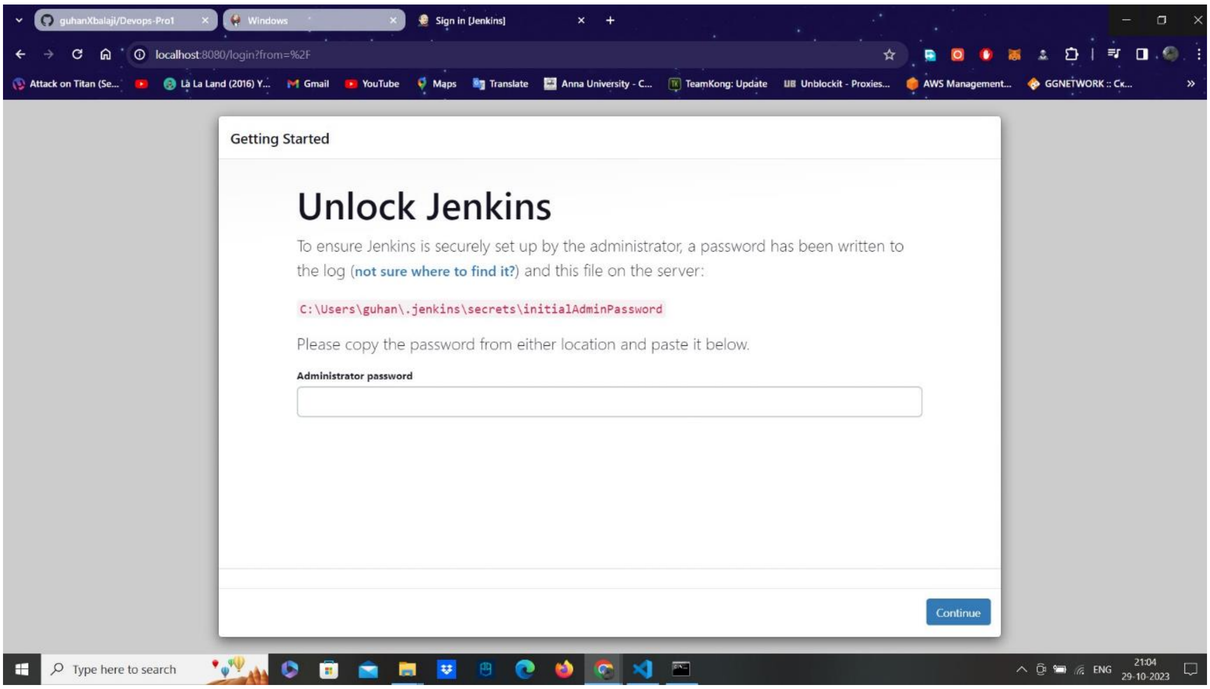
No packages published
Publish your first package

Languages

● Java 84.8% ● Dockerfile 15.2%

Step 2:

- Install openjdk 8 or higher in your pc and download Jenkins war file from the jenkins official website.
- Using command prompt as administrator. Run the jenkins war file.
- Now start Jenkins by opening a web browser and navigating to http://localhost:8080.
- Now login the Jenkins using the secret password, which will be found in jenkins/secrets/initialadminpassword in your PC.


guhanXbalaji/Devops-Pro1
Windows
Sign in [Jenkins]
localhost:8080/login?from=%2F
Attack on Titan (Se... La La Land (2016) Y... Gmail YouTube Maps Translate Anna University - C... TeamKong: Update Unblockit - Proxies... AWS Management... GGNETWORK :: Cx...
Continue
Type here to search
ENG 21:04 29-10-2023

Getting Started

Unlock Jenkins

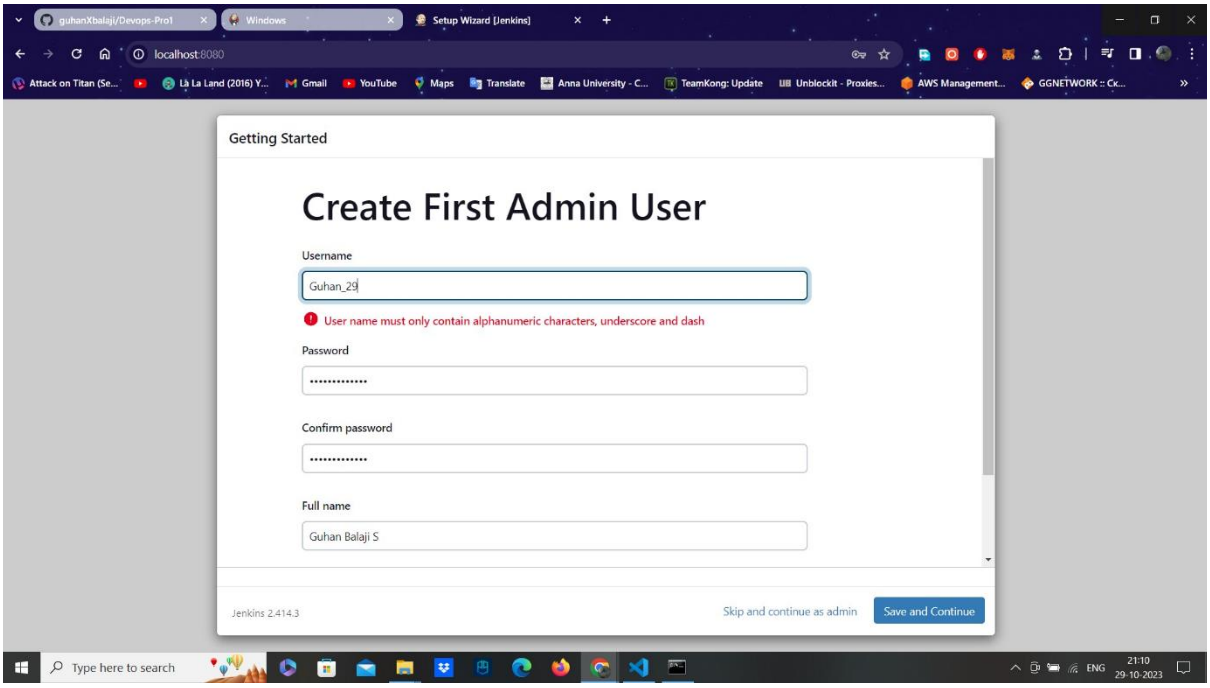
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`C:\Users\guhan\.jenkins\secrets\initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

- Now login in the jenkins. Add username ad create new password. Install the plugins.



Getting Started

Create First Admin User

Username

Guhan_29

User name must only contain alphanumeric characters, underscore and dash

Password

•••••••••••••

Confirm password

•••••••••••••

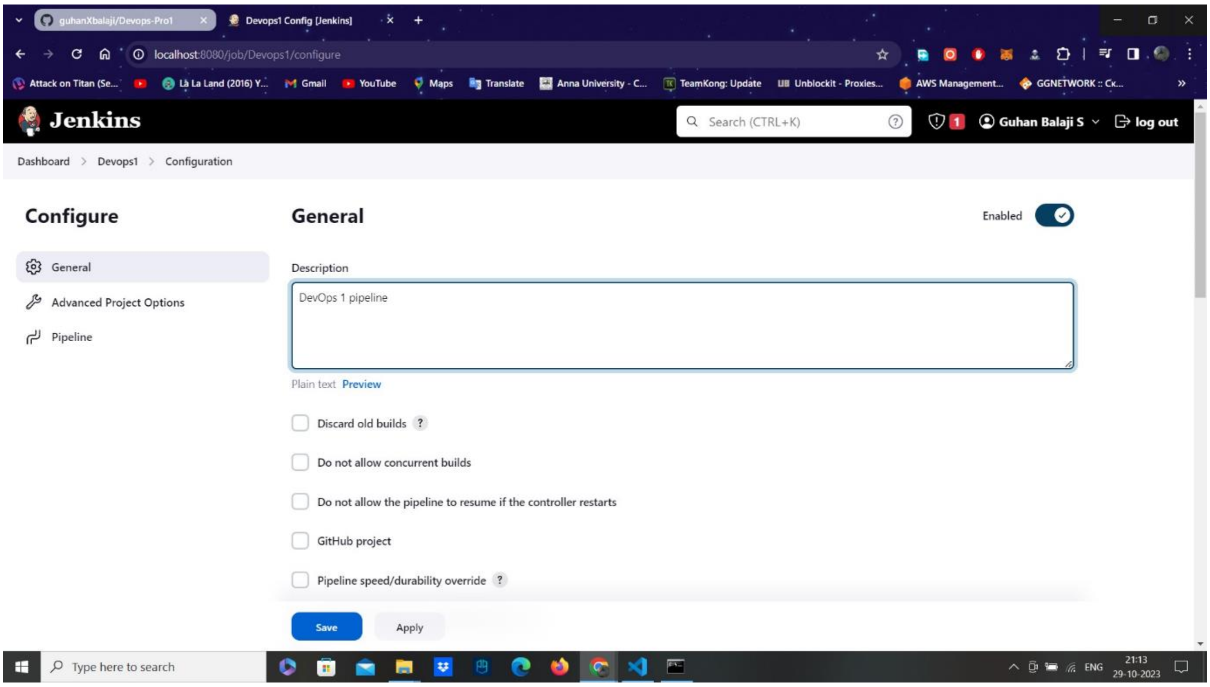
Full name

Guhan Balaji S

Jenkins 2.414.3

Step 3:

- Create a pipe line. Give the name and description.
- Select GitHub project and the URL of our repository. And also select build trigger snap.
- Now we should write the declarative pipeline. With any agent and tools with Maven



localhost:8080/job/Devops1/configure

Dashboard > Devops1 > Configuration

Configure

General

Enabled

Description

DevOps 1 pipeline

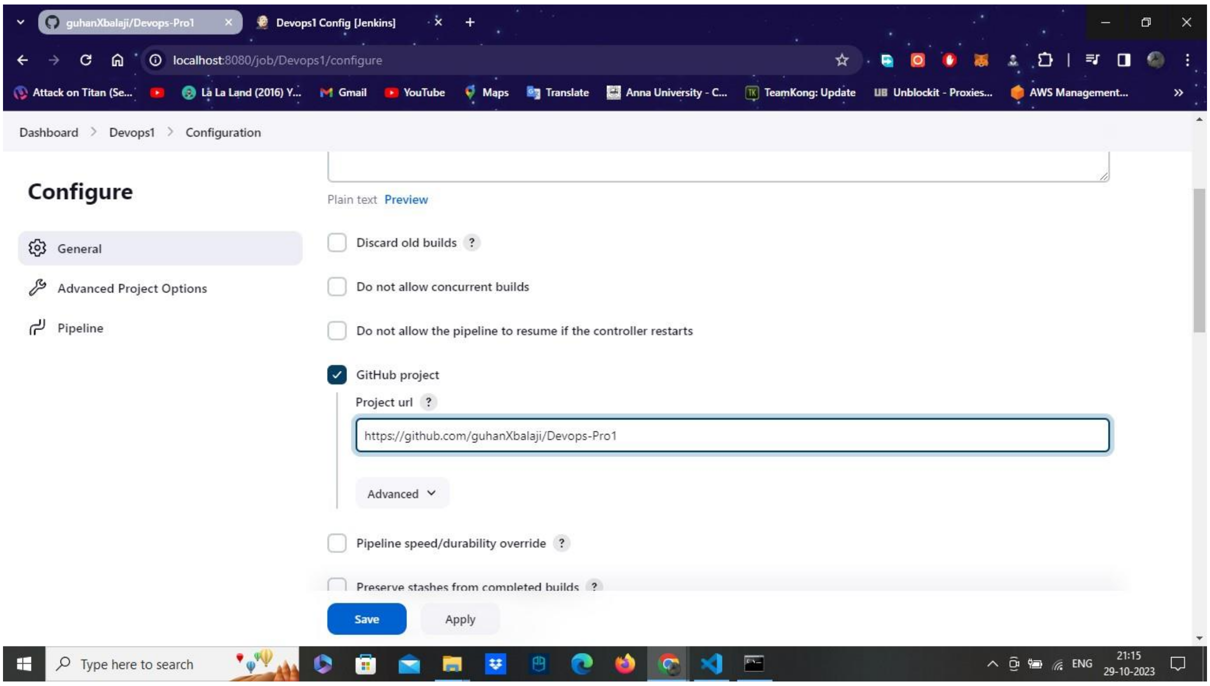
□ Discard old builds

□ Do not allow concurrent builds

□ Do not allow the pipeline to resume if the controller restarts

□ GitHub project

□ Pipeline speed/durability override



Dashboard > Devops1 > Configuration

Configure

Plain text Preview

□ Discard old builds

□ Do not allow concurrent builds

□ Do not allow the pipeline to resume if the controller restarts

☑ GitHub project

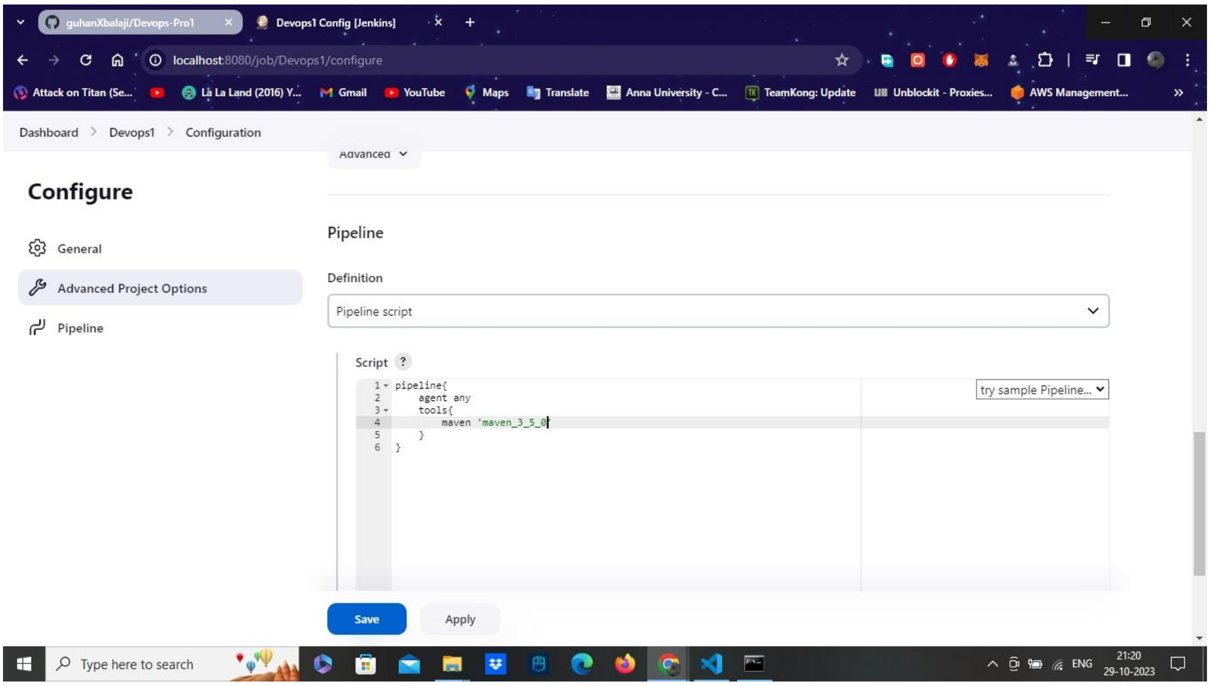
Project url

https://github.com/guhanXbalaji/Devops-Pro1

□ Pipeline speed/durability override

□ Preserve stashes from completed builds

..



Dashboard > Devops1 > Configuration

Configure

Pipeline

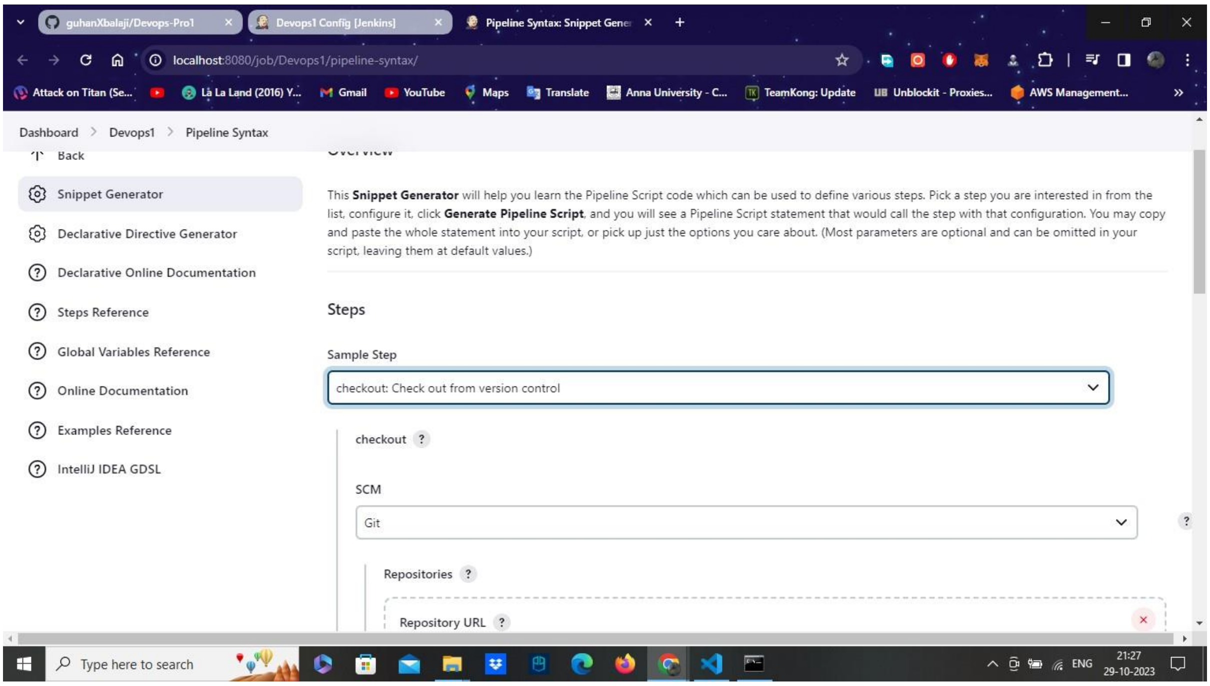
Definition

Pipeline script

Script

```
pipeline{
    agent any
    tools{
        maven 'maven_3_5_0'
    }
}
```

- Now add the stages in your script . Where add the stage as build maven.
- Use the pipeline syntax below and generate the script for git checkout .
- Select Checkout in sample step, git from scm and add your repository URL.
- Change the branch to master or main. And finally generate the script, paste in to the main pipeline script.
- And add the code bat ‘mvn clean install’ .
- And select build now and build it.


Dashboard > Devops1 > Pipeline Syntax
Back
Snippet Generator
Declarative Directive Generator
Declarative Online Documentation
Steps Reference
Global Variables Reference
Online Documentation
Examples Reference
IntelliJ IDEA GDSL

localhost:8080/job/Devops1/pipeline-syntax/

This **Snippet Generator** will help you learn the Pipeline Script code which can be used to define various steps. Pick a step you are interested in from the list, configure it, click **Generate Pipeline Script**, and you will see a Pipeline Script statement that would call the step with that configuration. You may copy and paste the whole statement into your script, or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving them at default values.)

Steps

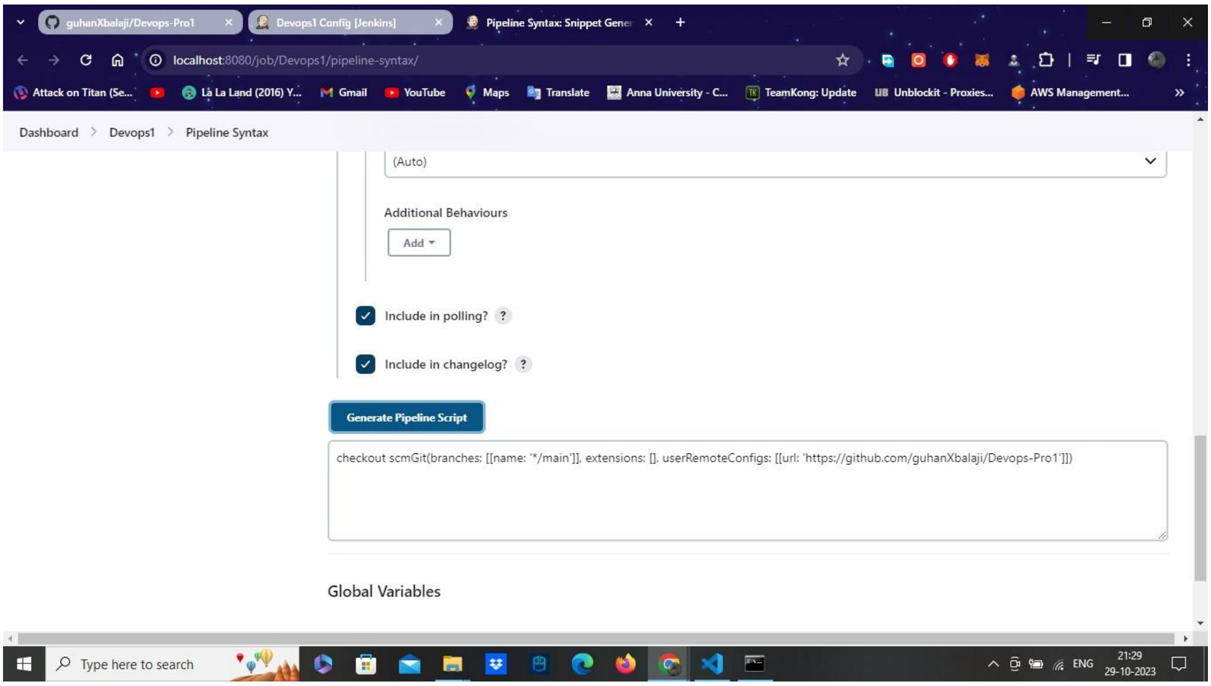
Sample Step: checkout: Check out from version control

checkout

SCM: Git

Repositories

Repository URL



localhost:8080/job/Devops1/pipeline-syntax/

(Auto)

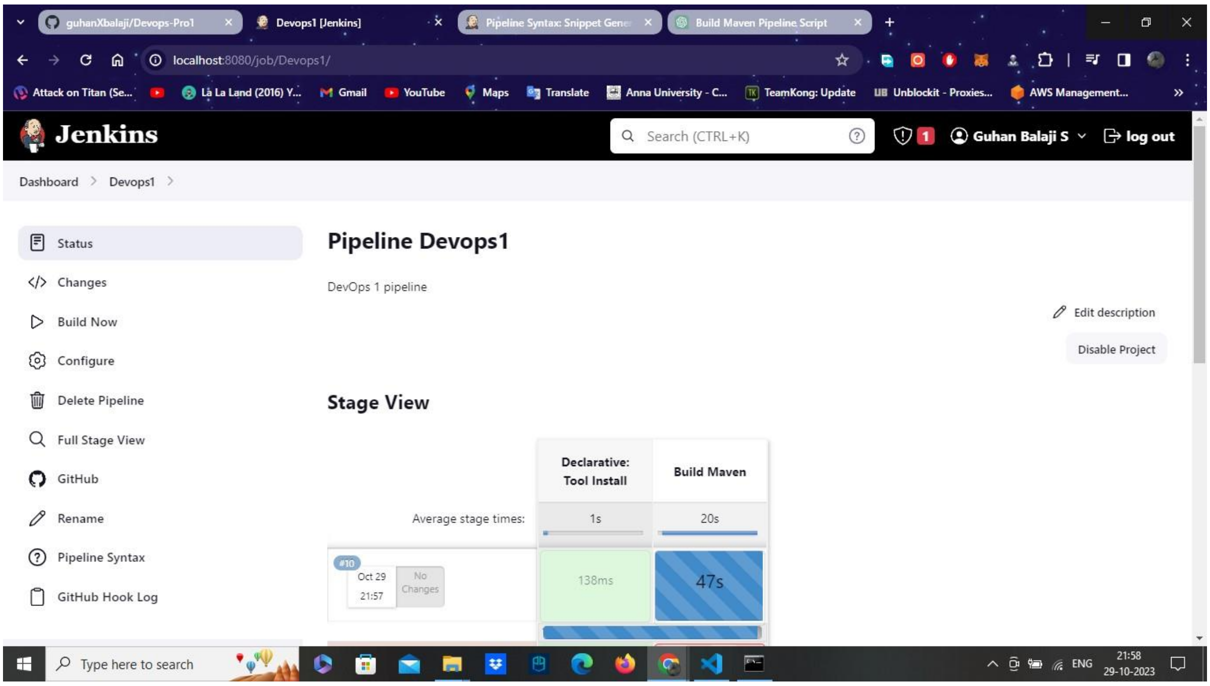
Additional Behaviours

☑ Include in polling?

☑ Include in changelog?

```
checkout scmGit(branches: [[name: '*/main']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/guhanXbalaji/Devops-Pro1']])
```

Global Variables

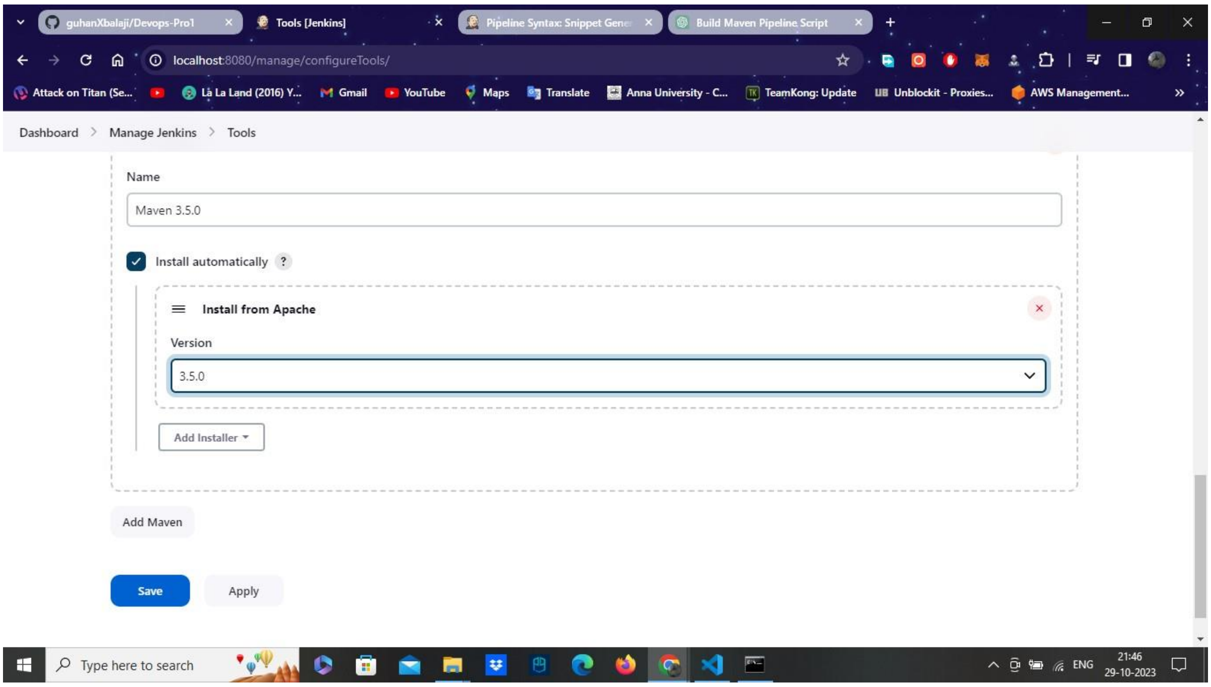


Pipeline Devops1

DevOps 1 pipeline

Stage View

	Declarative: Tool Install	Build Maven
Average stage times:	1s	20s
#10 Oct 29 21:57 No Changes	138ms	47s

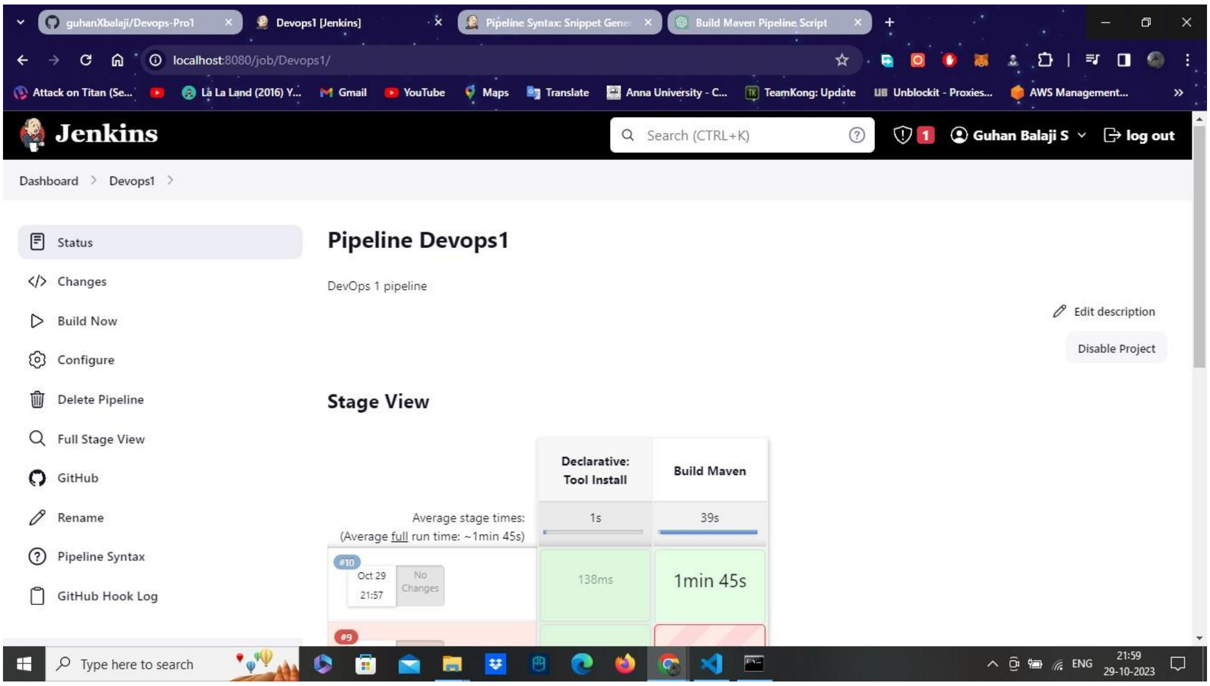


Name: Maven 3.5.0

☑ Install automatically

Install from Apache

Version: 3.5.0



Pipeline Devops1

DevOps 1 pipeline

Stage View

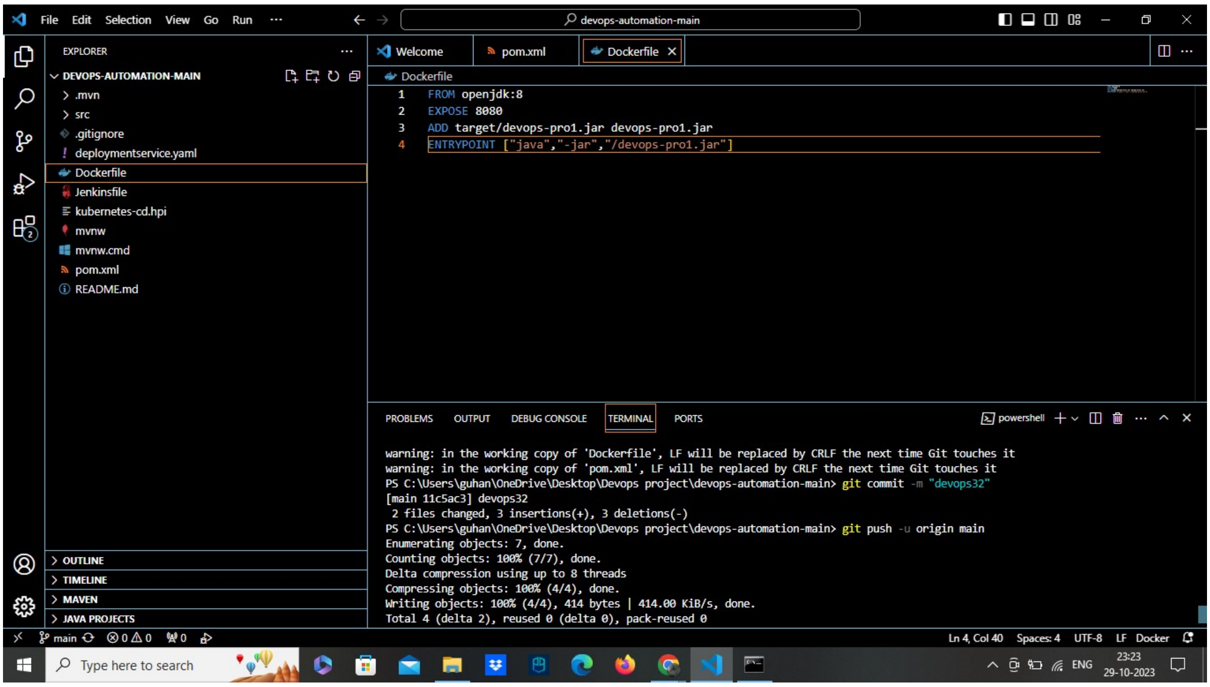
	Declarative: Tool Install	Build Maven
Average stage times: (Average full run time: ~1min 45s)	1s	39s
#10 Oct 29 21:57 No Changes	138ms	1min 45s

—

Step 4:

- Add the jar name in your both pom.xml and docker file from your application folder, which is presented in your GitHub Repository.
- And commit the message and push to your master repository again.

.



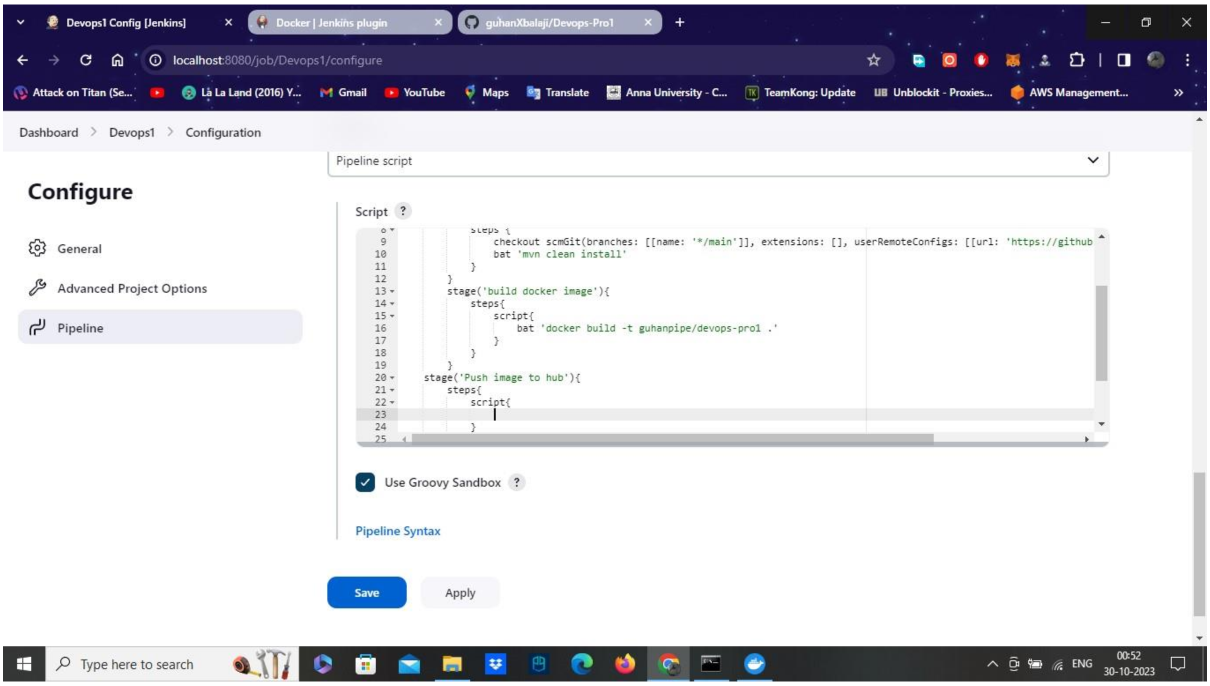
Dockerfile

```
FROM openjdk:8
EXPOSE 8080
ADD target/devops-pro1.jar devops-pro1.jar
ENTRYPOINT ["java","-jar","/devops-pro1.jar"]
```

```
warning: in the working copy of 'Dockerfile', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'pom.xml', LF will be replaced by CRLF the next time Git touches it
PS C:\Users\guhan\OneDrive\Desktop\Devops project\devops-automation-main> git commit -m "devops32"
[main 11c5ac3] devops32
 2 files changed, 3 insertions(+), 3 deletions(-)
PS C:\Users\guhan\OneDrive\Desktop\Devops project\devops-automation-main> git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 414 bytes | 414.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
```

Step 5:

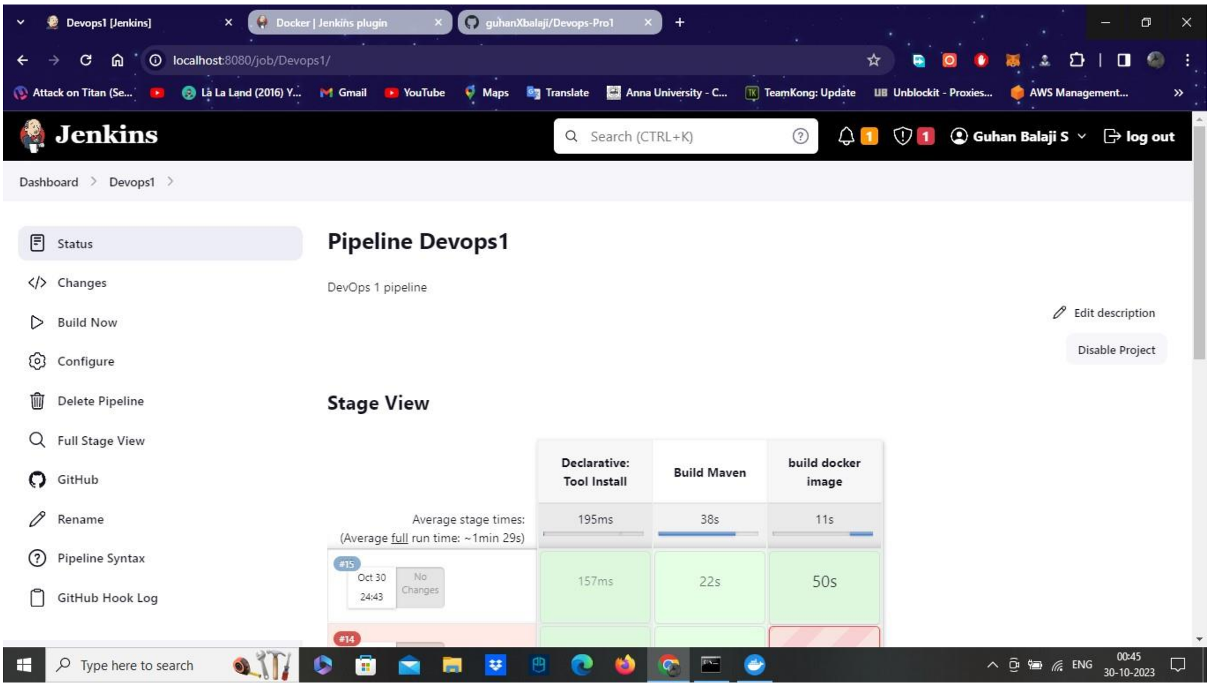
- Open the Docker desktop app and start the docker engine in your Pc.
- Now go back to configure in jenkins.
- Add a stage Build docker image. Add steps and script inside the stage.
- Add bat ‘ docker build -t guhanxdocker/imagename .’ which will build the docker image.
- Apply it and save it. And select build now to build again with this stages now.
- Note: Check the docker got added in your jenkins. if not install docker plugin from manage plugin and add docker in system configuration with the version you want.



```
            steps {
                checkout scmGit(branches: [[name: '*/main']], extensions: [], userRemoteConfigs: [[url: 'https://github
                bat 'mvn clean install'
            }
        }
        stage('build docker image'){
            steps{
                script{
                    bat 'docker build -t guhanpipe/devops-pro1 .'
                }
            }
        }
    stage('Push image to hub'){
        steps{
            script{
                
            }
```

☑ Use Groovy Sandbox

Pipeline Syntax

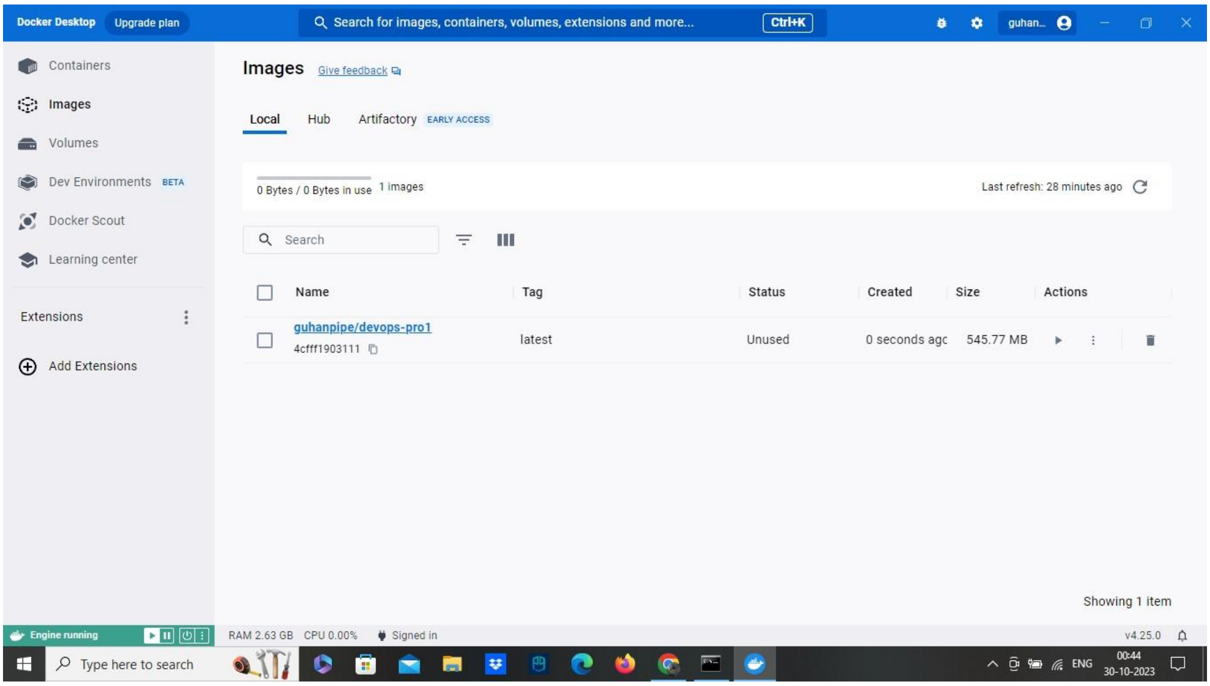


Pipeline Devops1

DevOps 1 pipeline

Stage View

	Declarative: Tool Install	Build Maven	build docker image
Average stage times: (Average full run time: ~1min 29s)	195ms	38s	11s
#15 Oct 30 24:43 No Changes	157ms	22s	50s
#14			



0 Bytes / 0 Bytes in use 1 images

Last refresh: 28 minutes ago

□	Name	Tag	Status	Created	Size	Actions
□	guhanpipe/devops-pro1 4cfff1903111	latest	Unused	0 seconds agc	545.77 MB	▶ ⋮ 🗑

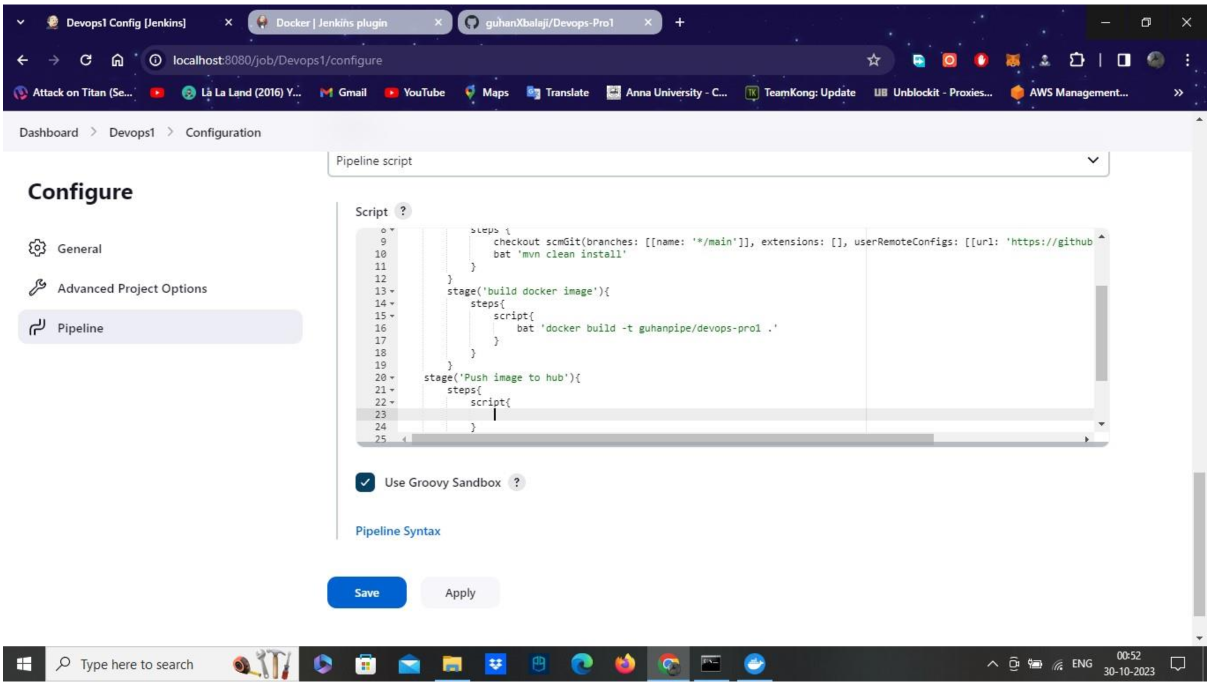
Showing 1 item

RAM 2.63 GB CPU 0.00% Signed in v4.25.0

00:44 30-10-2023

Step 6:

- Now we have to push the docker image we built to our docker hub.
- Add the stage push image to hub. Add a step with script. To add a script, use pipeline syntax.
- Select with credentials in sample step, add secret text. Add variable as dockerhubpwd and add credentials in secret text.
- Add Credentials, choose kind as secret text. Enter your docker hub password in secret. And enter any ID.
- Save it and choose the credentials we created now.
- Generate the script, copy it and paste in our declarative pipeline.
- And add a command below the generated script with bat ‘ docker login -u guhanxdocker -p ${}’.
- So, Now we can able to login in our docker hub with script we given above.
- Now to push to our docker hub.
- Add a command bat ‘ docker push our image name’.
- Apply and save it.
- And Finally select build now to start the build.



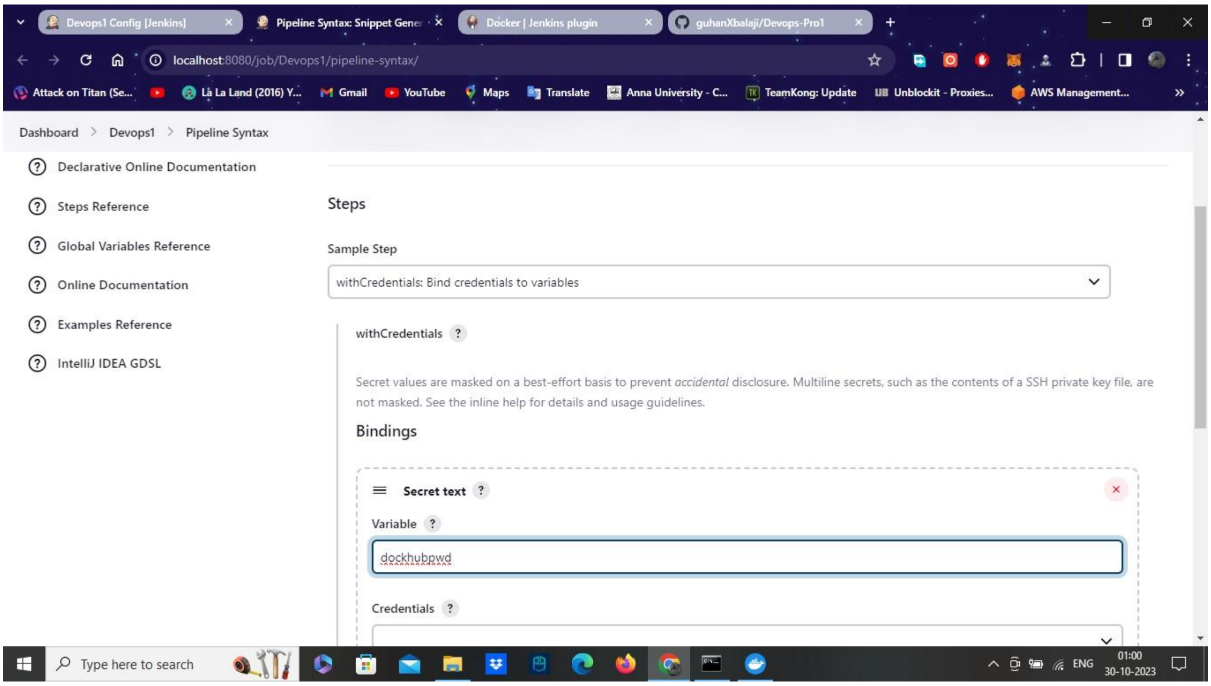
Pipeline script

Script

```
                steps {
                    checkout scmGit(branches: [[name: '*/main']], extensions: [], userRemoteConfigs: [[url: 'https://github
                    bat 'mvn clean install'
                }
            }
            stage('build docker image'){
                steps{
                    script{
                        bat 'docker build -t guhanpipe/devops-pro1 .'
                    }
                }
            }
        stage('Push image to hub'){
            steps{
                script{
                    
                }
```

☑ Use Groovy Sandbox

Pipeline Syntax



Steps

Sample Step

withCredentials: Bind credentials to variables

withCredentials

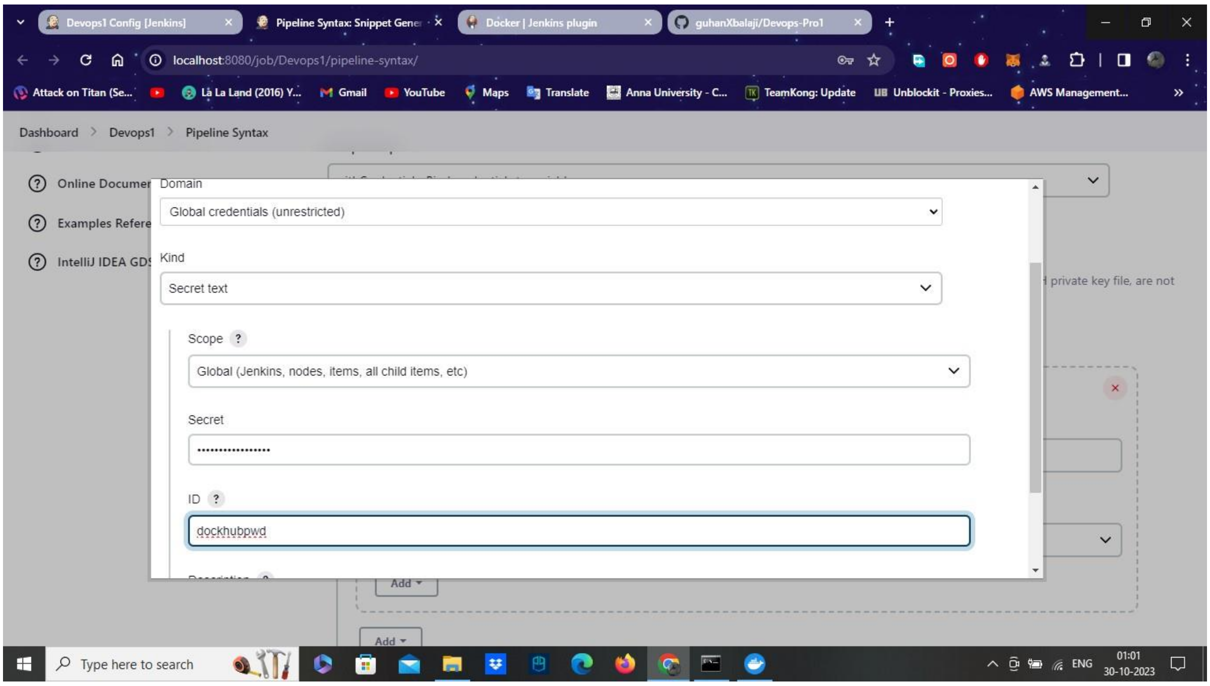
Secret values are masked on a best-effort basis to prevent *accidental* disclosure. Multiline secrets, such as the contents of a SSH private key file, are not masked. See the inline help for details and usage guidelines.

Bindings

Secret text

Variable: dockhubpwd

Credentials


localhost:8080/job/Devops1/pipeline-syntax/
Dashboard > Devops1 > Pipeline Syntax
Online Documer
Examples Refere
IntelliJ IDEA GDS
Add

Domain: Global credentials (unrestricted)

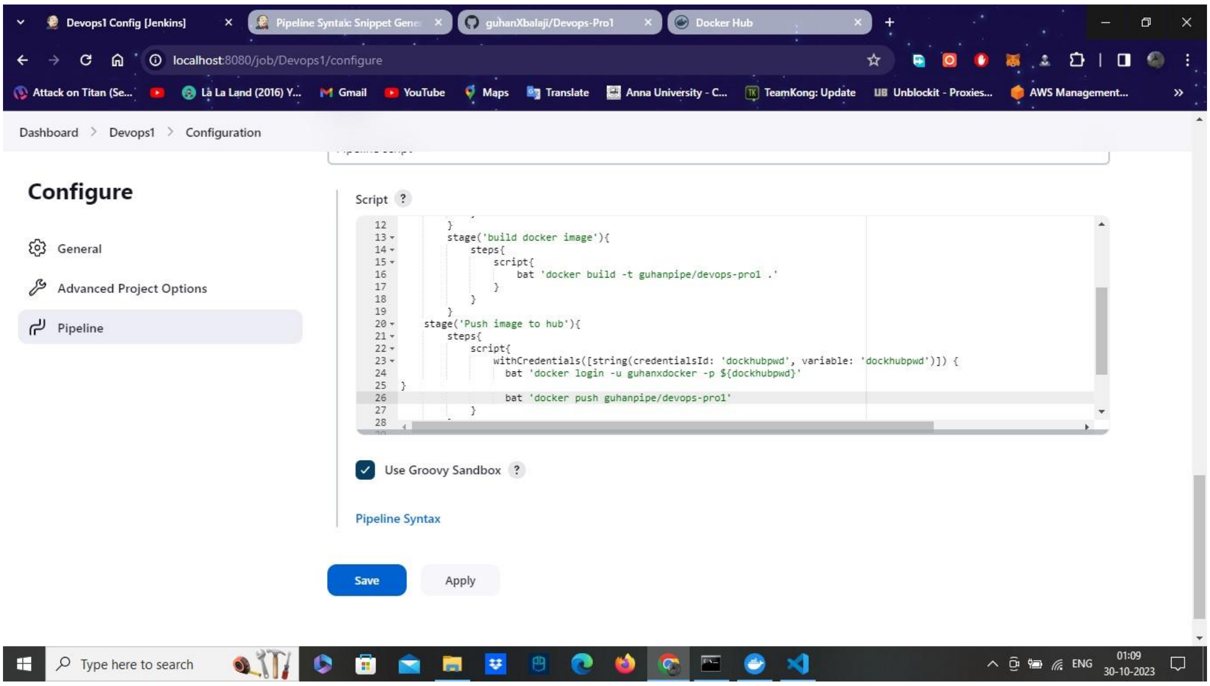
Kind: Secret text

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

Secret: ••••••••••••••••••

ID: dockhubpwd

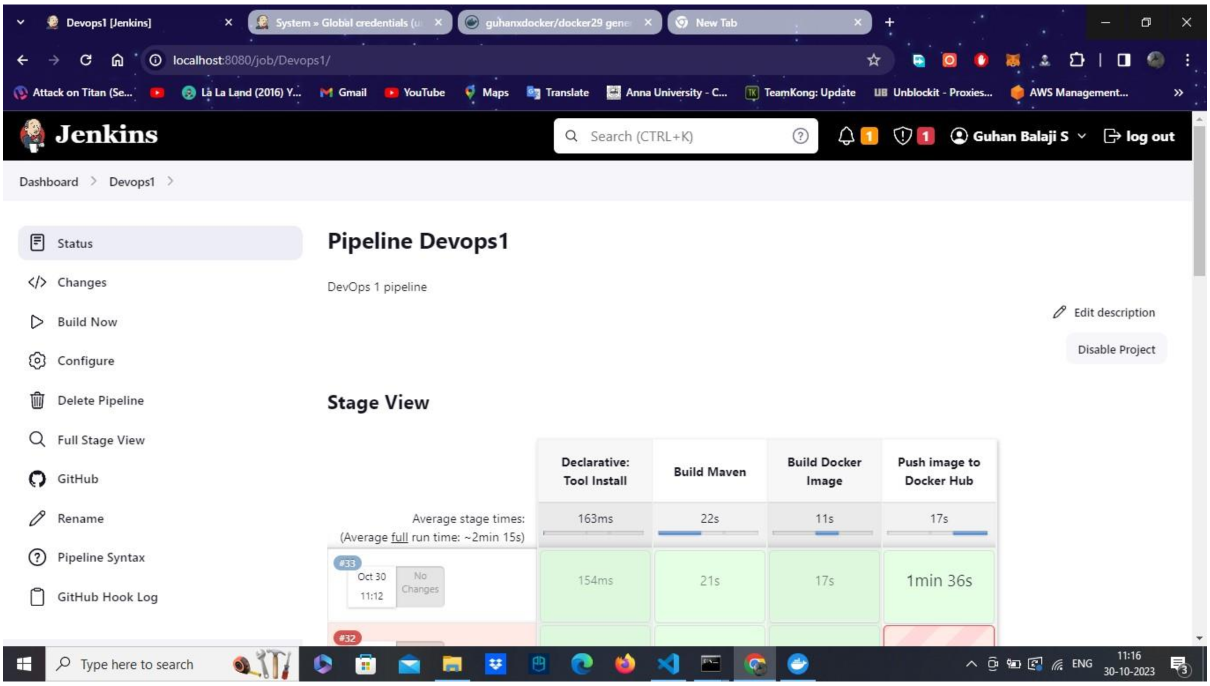
--



Script

```
        }
        stage('build docker image'){
            steps{
                script{
                    bat 'docker build -t guhanpipe/devops-pro1 .'
                }
            }
        }
    stage('Push image to hub'){
        steps{
            script{
                withCredentials([string(credentialsId: 'dockhubpwd', variable: 'dockhubpwd')]) {
                  bat 'docker login -u guhanxdocker -p ${dockhubpwd}'
}
                  bat 'docker push guhanpipe/devops-pro1'
            }
```

☑ Use Groovy Sandbox



Pipeline Devops1

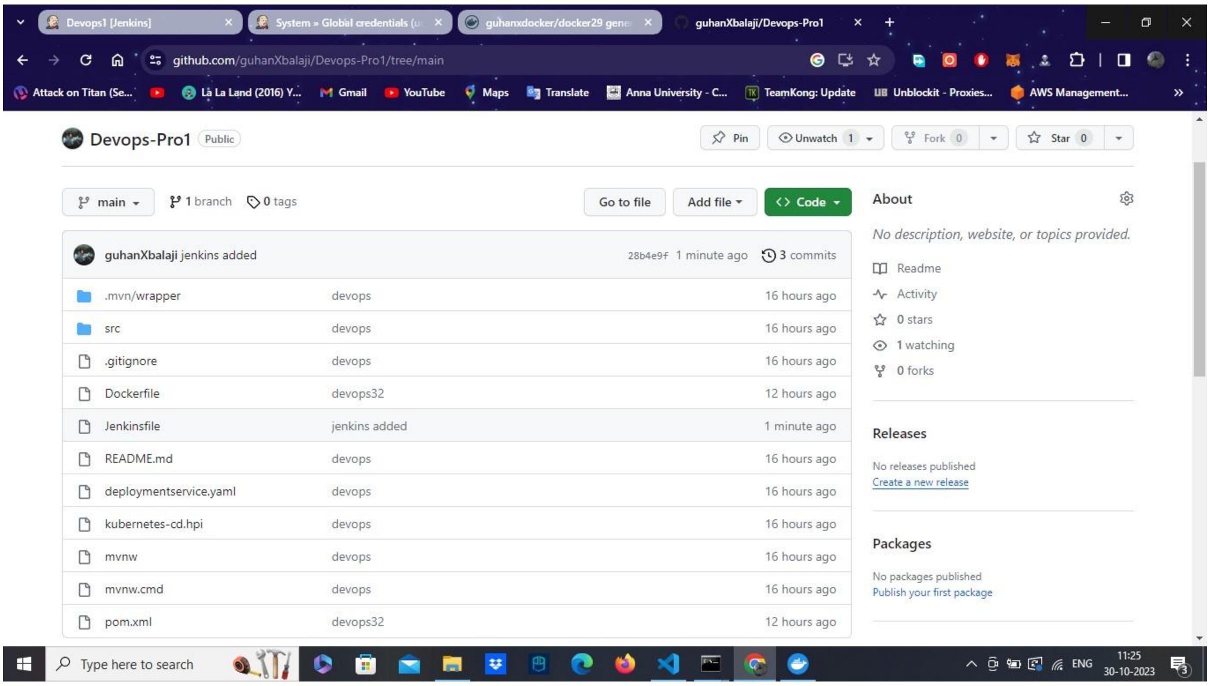
DevOps 1 pipeline

Stage View

	Declarative: Tool Install	Build Maven	Build Docker Image	Push image to Docker Hub
Average stage times: (Average full run time: ~2min 15s)	163ms	22s	11s	17s
#33 Oct 30 11:12 No Changes	154ms	21s	17s	1min 36s
#32				

Step 7:

- We successfully pushed our image to our docker hub
- And the declarative pipeline script also runs without error.
- Now we can try it with using jenkins file.
- Create a jenkins file in our GitHub repository. Copy the pipeline script of our project we created.
- Now paste in to the jenkins file we created and save it.
- Add a New pipeline, give a name.
- Enter the project repo URL from git hub in git hub project.
- Now we select the pipeline script from scm instead of pipeline script.
- Select scm as git.
- Give the repository URL. Select branch as master or main and select path of the jenkins file.
- Save it and Select Build now, where it will read the Jenkinsfile and start to build.



github.com/guhanXbalaji/Devops-Pro1/tree/main

Devops-Pro1 Public

main · 1 branch · 0 tags

guhanXbalaji jenkins added · 28b4e9f 1 minute ago · 3 commits

.mvn/wrapper	devops	16 hours ago
src	devops	16 hours ago
.gitignore	devops	16 hours ago
Dockerfile	devops32	12 hours ago
Jenkinsfile	jenkins added	1 minute ago
README.md	devops	16 hours ago
deploymentservice.yaml	devops	16 hours ago
kubernetes-cd.hpi	devops	16 hours ago
mvnw	devops	16 hours ago
mvnw.cmd	devops	16 hours ago
pom.xml	devops32	12 hours ago

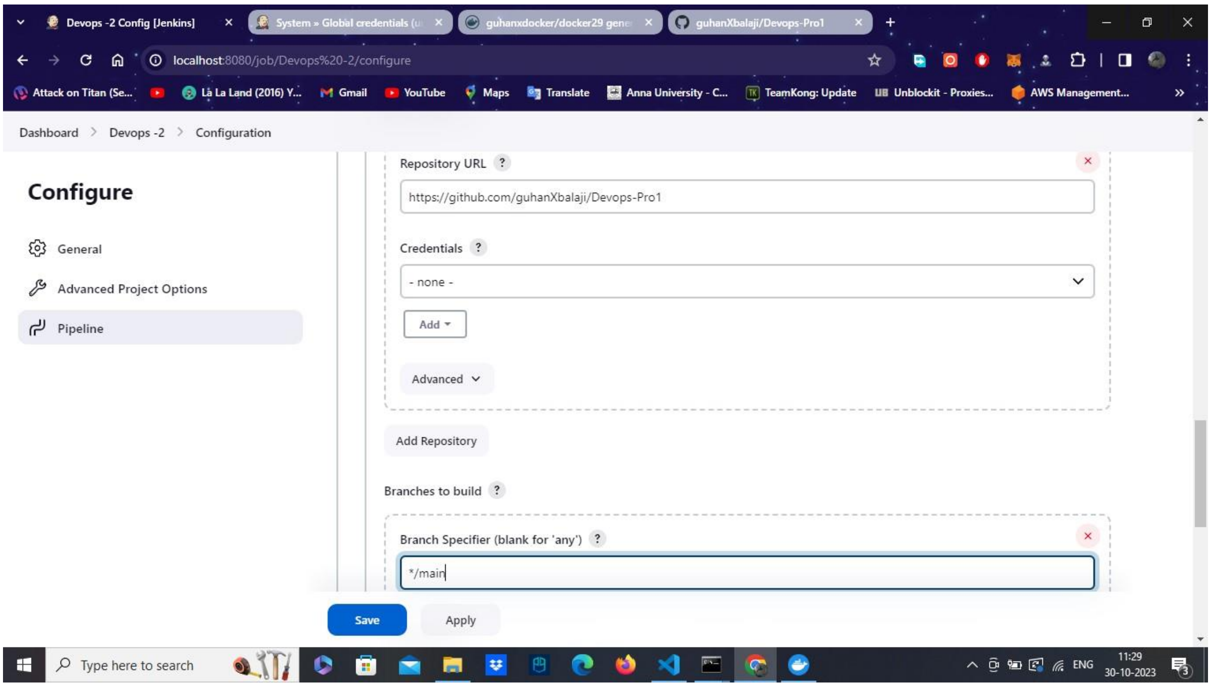
About

No description, website, or topics provided.

Readme · Activity · 0 stars · 1 watching · 0 forks

Releases: No releases published · Create a new release

Packages: No packages published · Publish your first package



localhost:8080/job/Devops%20-2/configure

Dashboard > Devops -2 > Configuration

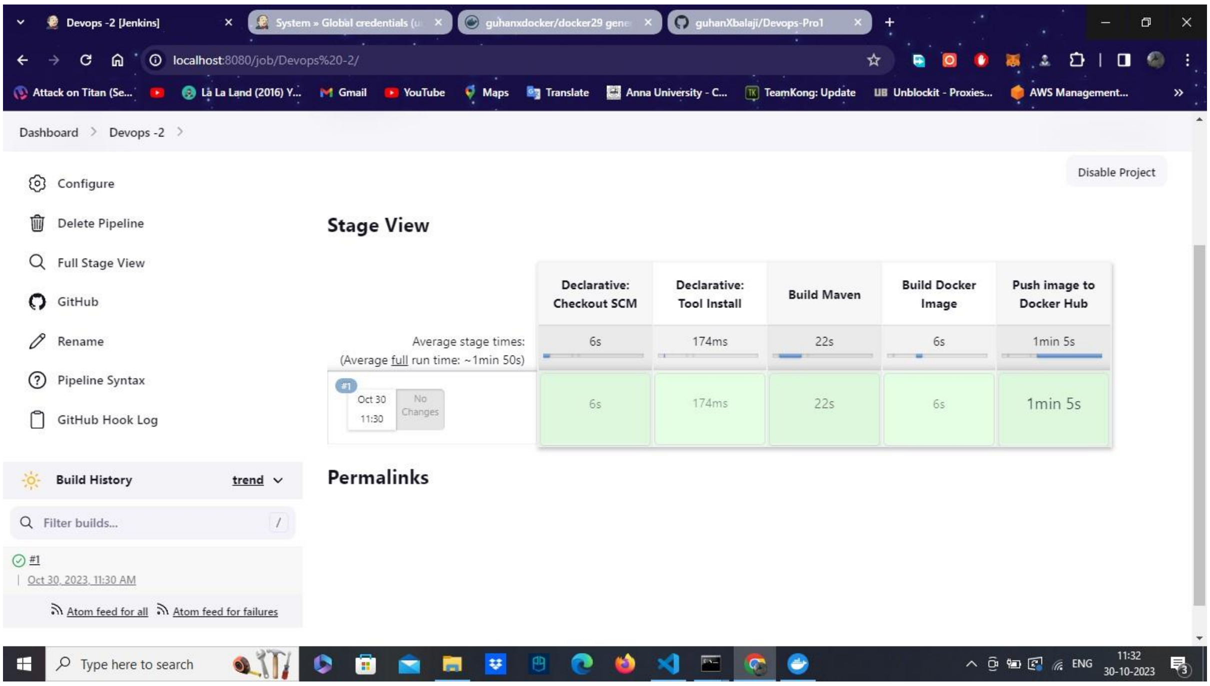
Configure

Repository URL: https://github.com/guhanXbalaji/Devops-Pro1

Credentials: - none -

Branches to build

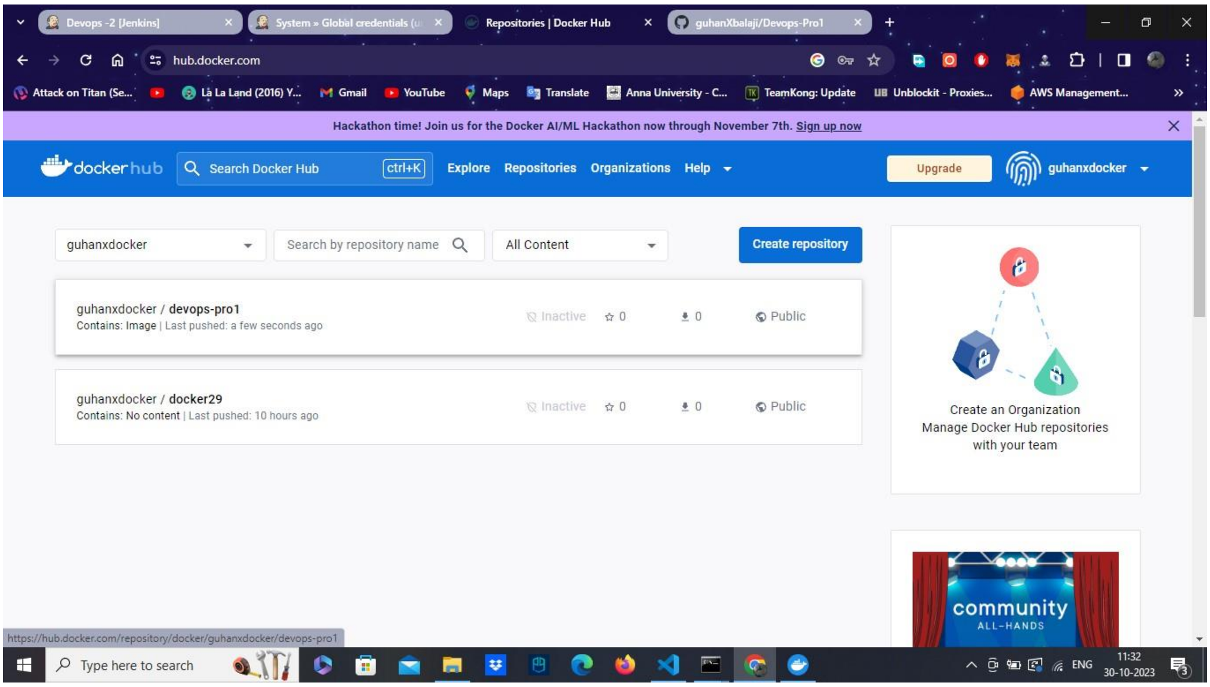
Branch Specifier (blank for 'any'): */main



Stage View

	Declarative: Checkout SCM	Declarative: Tool Install	Build Maven	Build Docker Image	Push image to Docker Hub
Average stage times: (Average full run time: ~1min 50s)	6s	174ms	22s	6s	1min 5s
#1 Oct 30 11:30 No Changes	6s	174ms	22s	6s	1min 5s

#1
Oct 30, 2023, 11:30 AM


Repositories | Docker Hub
hub.docker.com
Hackathon time! Join us for the Docker AI/ML Hackathon now through November 7th. Sign up now
docker hub
Search Docker Hub
Explore
Repositories
Organizations
Help
Upgrade
guhanxdocker
Search by repository name
All Content
Create repository
Create an Organization
Manage Docker Hub repositories with your team
community ALL-HANDS

Repository	Status	Stars	Pulls	Visibility
guhanxdocker / devops-pro1 Contains: Image | Last pushed: a few seconds ago	Inactive	0	0	Public
guhanxdocker / docker29 Contains: No content | Last pushed: 10 hours ago	Inactive	0	0	Public

Now finally the Docker image pushed to the hub with the help of jenkins file also, This how we will build and push docker image using jenkins pipeline. And this how DevOps end to end integration works.