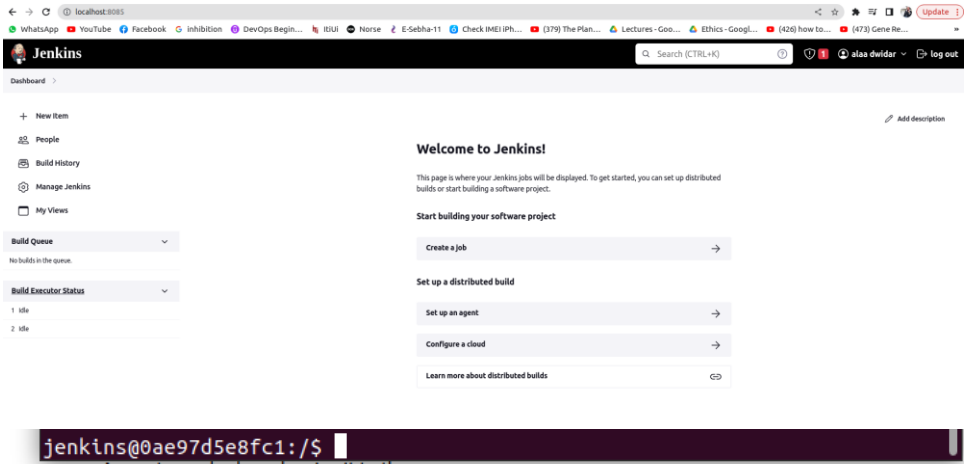
1- install jenkins with docker image

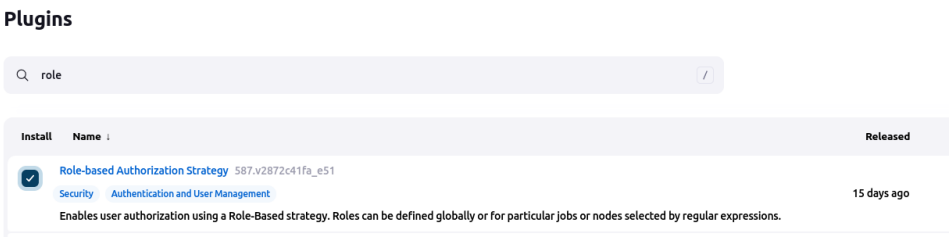
docker run -p 8085:8080 -d jenkins/jenkins:lts

docker ps

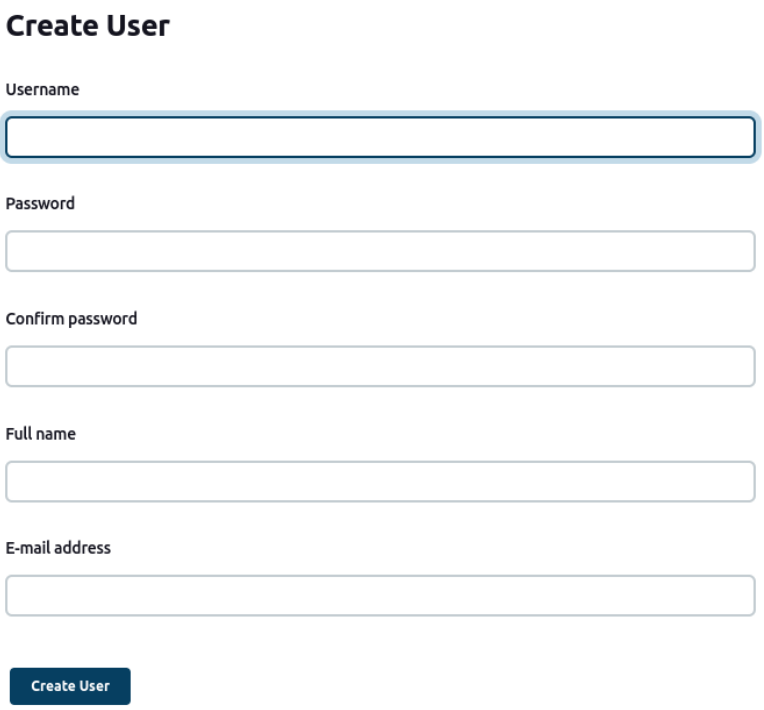
docker exec -it 0ae97d5e8fc1 bash



2- install role based authorization plugin

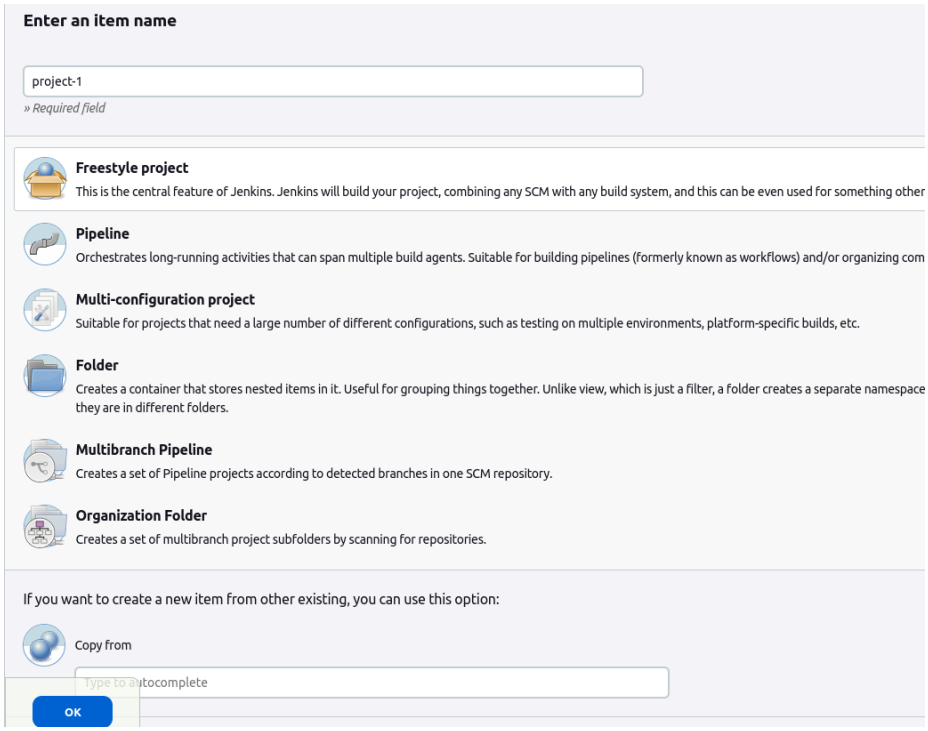


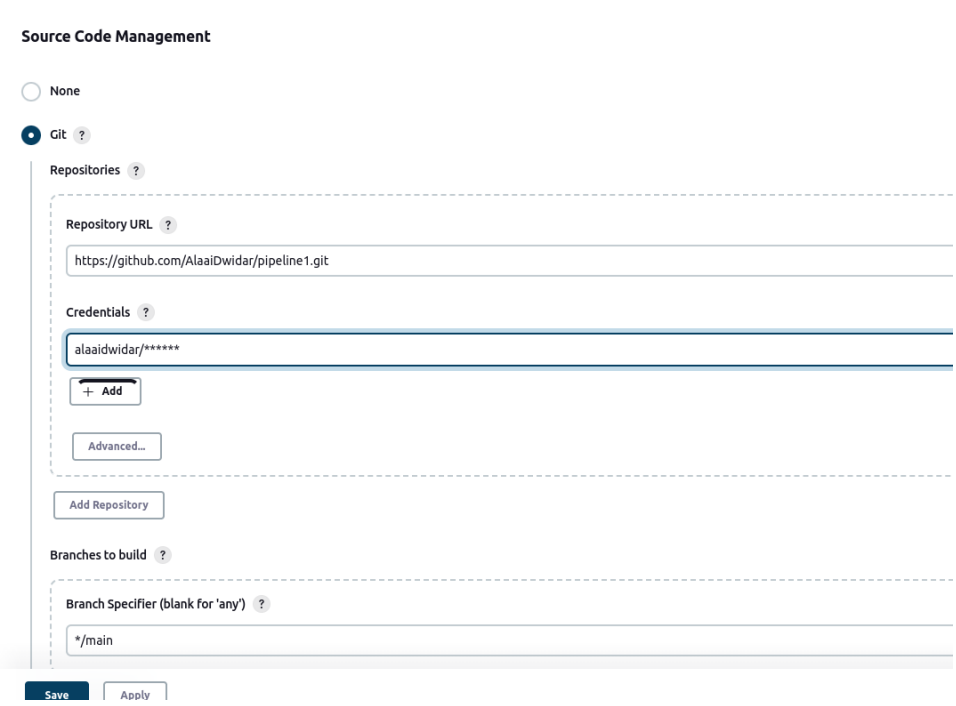
3- create new user

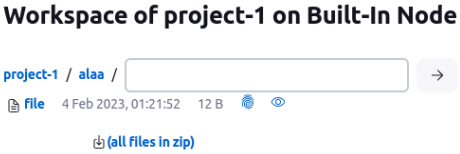


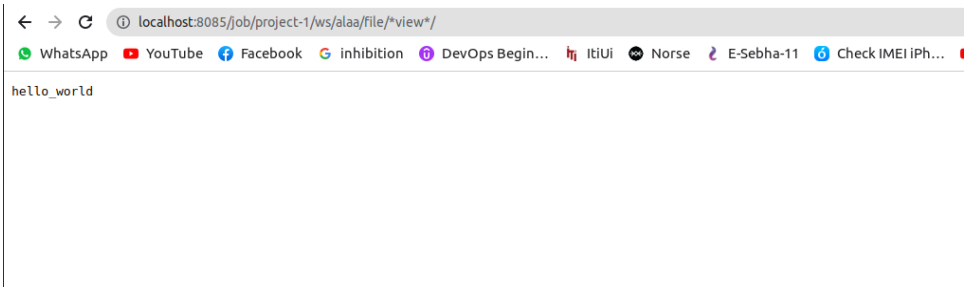
4- create read role and assign it to the new user

5- create free style pipeline and link it to private git repo(inside it create directory and create file with "hello world")


Enter an item name
project-1
» Required field
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing com
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace
they are in different folders.
Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.
Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.
If you want to create a new item from other existing, you can use this option:
Copy from
OK

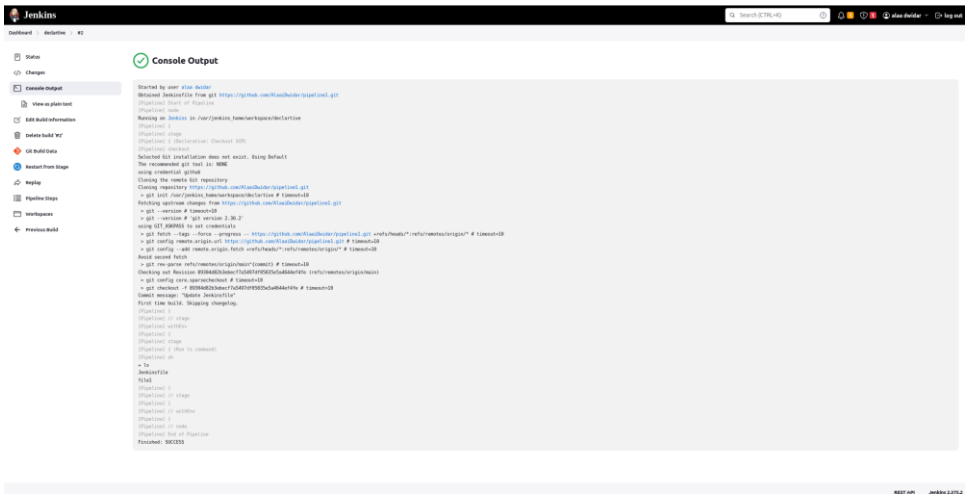

Source Code Management
None
Git
Repositories
Repository URL
https://github.com/AlaaiDwidar/pipeline1.git
Credentials
alaaidwidar/******
Add
Advanced...
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/main
Save
Apply

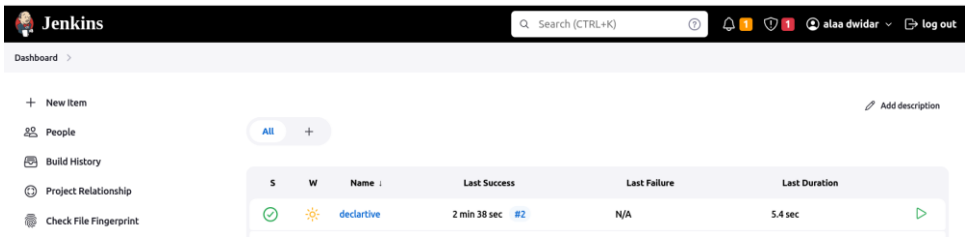

Workspace of project-1 on Built-In Node
project-1 / alaa /
file 4 Feb 2023, 01:21:52 12 B
(all files in zip)



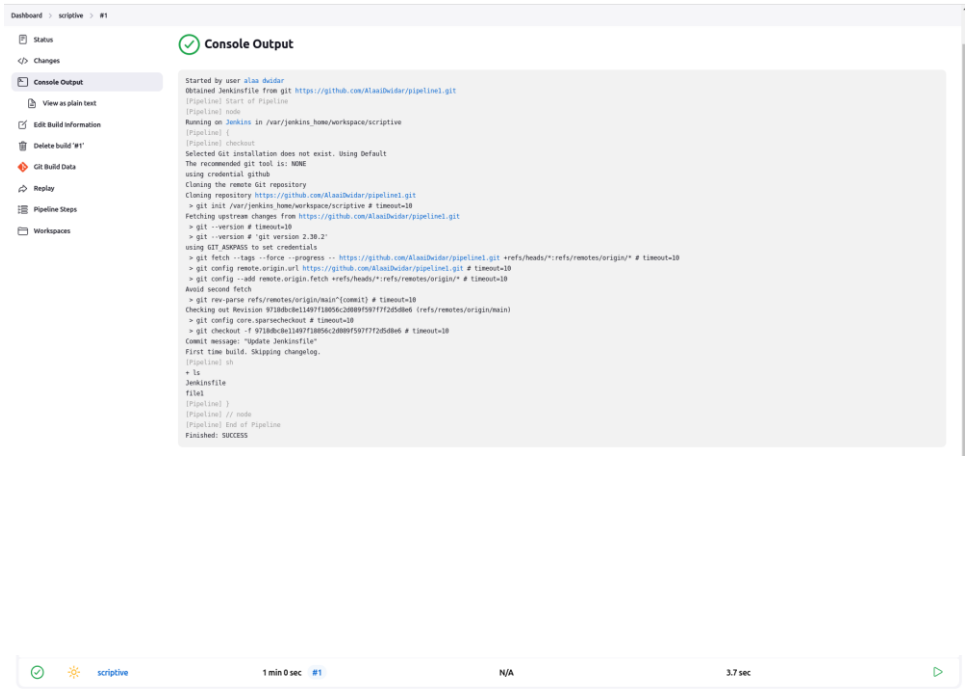
Part 2

1- create declarative in jenkins GUI pipeline for your own repo to do "ls"





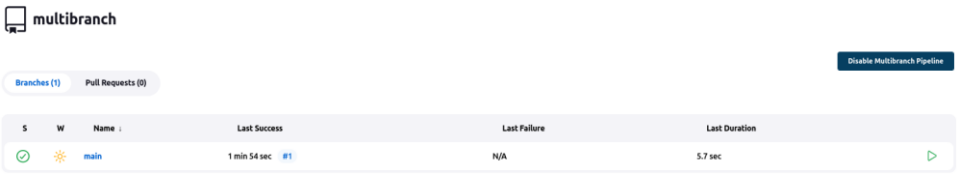
2- create scripted in jenkins GUI pipeline for your own repo to do "ls"



3- create the same with jenkinsfile in your branches as multibranch pipeline

Scan Repository Log

```
Started
[Sat Feb 04 02:37:28 UTC 2023] Starting branch indexing...
02:37:29 Connecting to https://api.github.com using alaaidwidar/******
Examining AlaaiDwidar/pipeline1

  Checking branches...

  Getting remote branches...

    Checking branch main

  Getting remote pull requests...
      'Jenkinsfile' found
    Met criteria
Scheduled build for branch: main

  1 branches were processed

  Checking pull-requests...

  0 pull requests were processed

Finished examining AlaaiDwidar/pipeline1

[Sat Feb 04 02:37:31 UTC 2023] Finished branch indexing. Indexing took 2.6 sec
Finished: SUCCESS
```



Console Output

```
Branch indexing
02:37:37 Connecting to https://api.github.com using alaaidwidar/******
Obtained Jenkinsfile from 9718dbc8e11497f18056c2d089f597f7f2d5d8e6
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/multibranch_main
[Pipeline] {
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential github
Cloning the remote Git repository
Cloning with configured refspecs honoured and without tags
Cloning repository https://github.com/AlaaiDwidar/pipeline1.git
 > git init /var/jenkins_home/workspace/multibranch_main # timeout=10
Fetching upstream changes from https://github.com/AlaaiDwidar/pipeline1.git
 > git --version # timeout=10
 > git --version # 'git version 2.30.2'
using GIT_ASKPASS to set credentials
 > git fetch --no-tags --force --progress -- https://github.com/AlaaiDwidar/pipeline1.git +refs/heads/main:refs/remotes/origin/main # timeout=10
 > git config remote.origin.url https://github.com/AlaaiDwidar/pipeline1.git # timeout=10
 > git config --add remote.origin.fetch +refs/heads/main:refs/remotes/origin/main # timeout=10
Avoid second fetch
Checking out Revision 9718dbc8e11497f18056c2d089f597f7f2d5d8e6 (main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 9718dbc8e11497f18056c2d089f597f7f2d5d8e6 # timeout=10
Commit message: "Update Jenkinsfile"
First time build. Skipping changelog.
[Pipeline] sh
+ ls
Jenkinsfile
file1
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline

GitHub has been notified of this commit's build result

Finished: SUCCESS
```