Secure Code Management Homework

Preda Mihail Irinel, ISM

Setup jenkins

Steps

1. Go to https://www.jenkins.io/download/ and download Download `Jenkins 2.319.1 LTS` for your OS system
2. Open Installer

Fig.1

2. Choose where to install it:

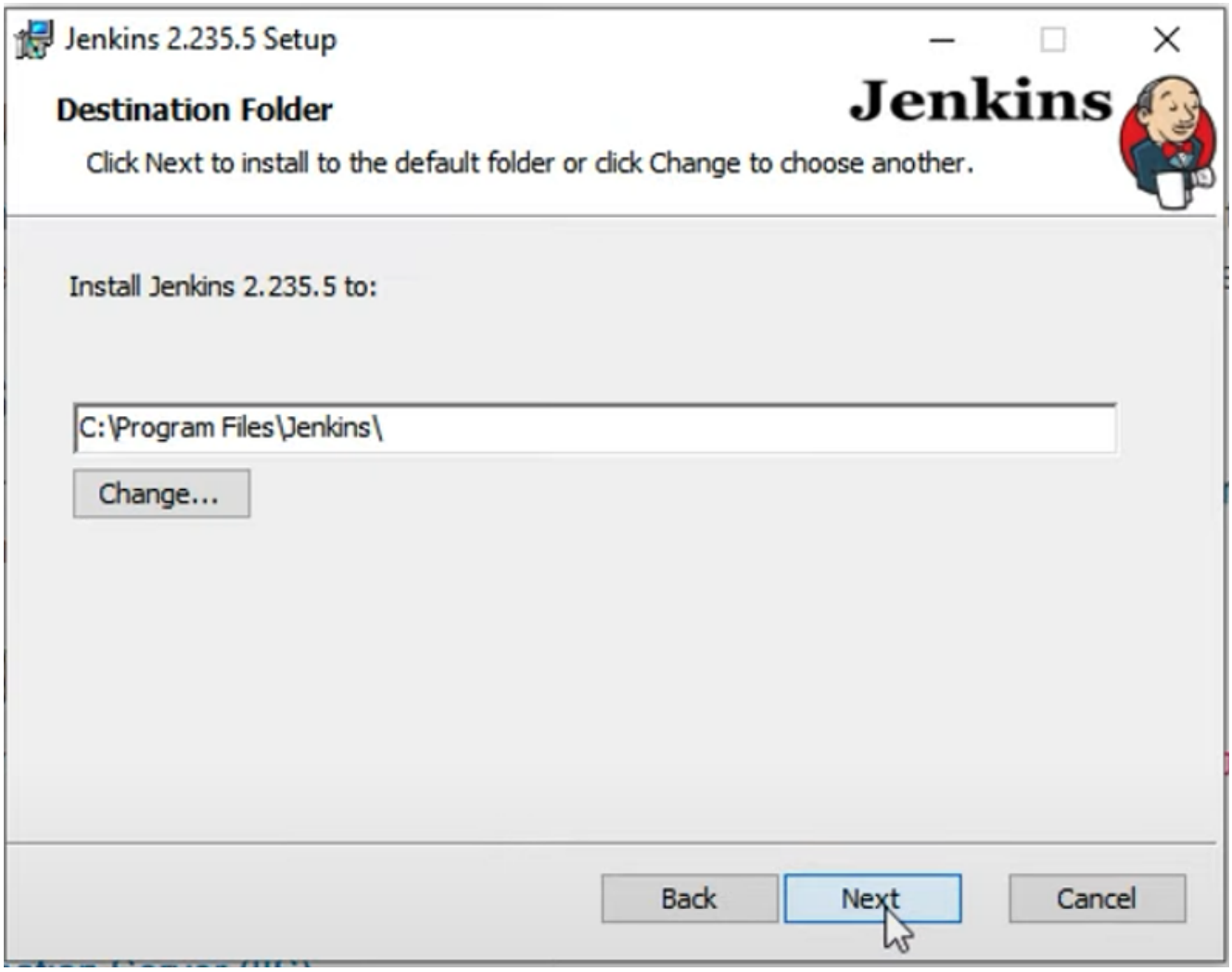

Fig.2

3. Choose Logon type. Select `Run service as LocalSystem`

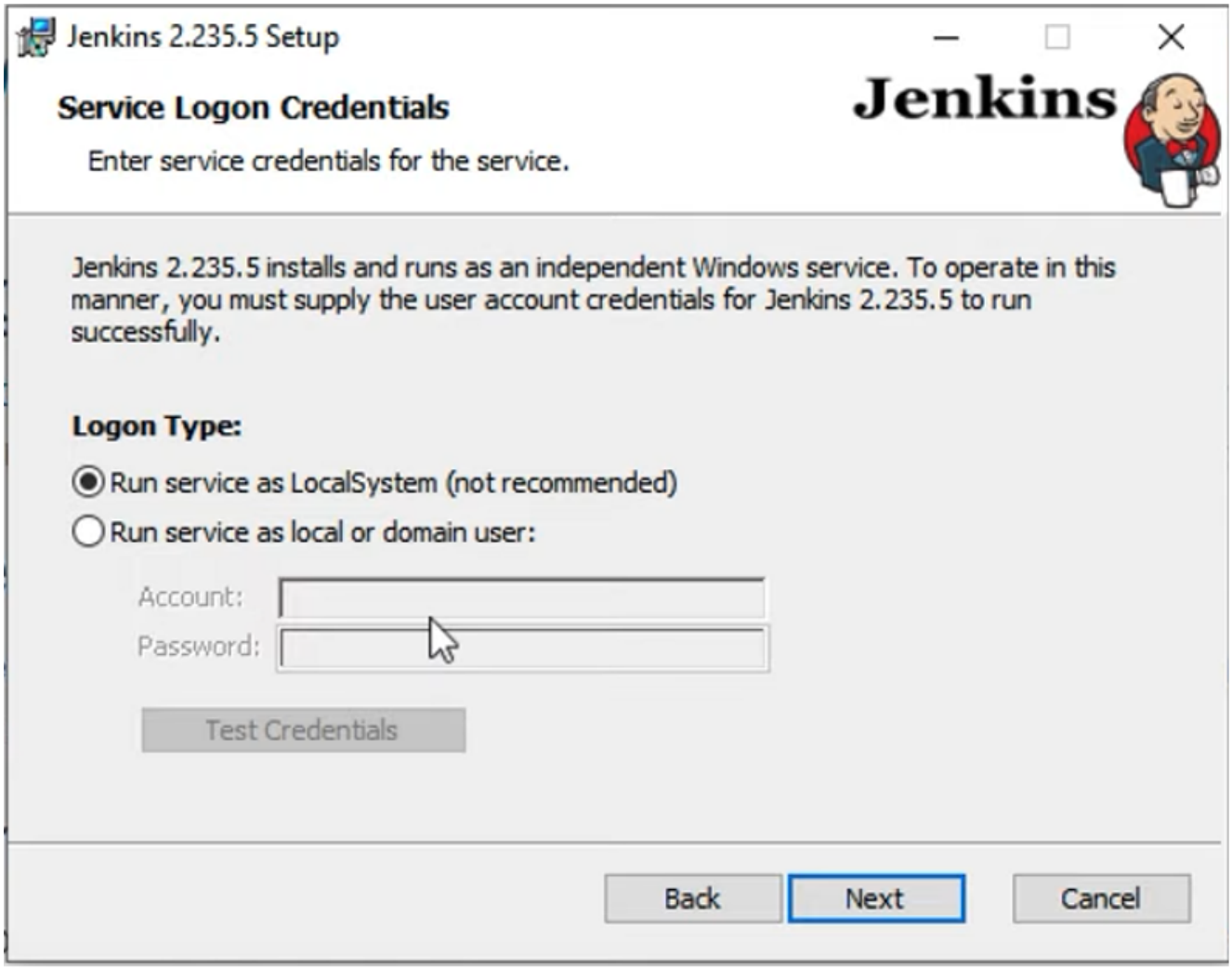

Fig.3

4. Select a port number or leave it as default (8080)

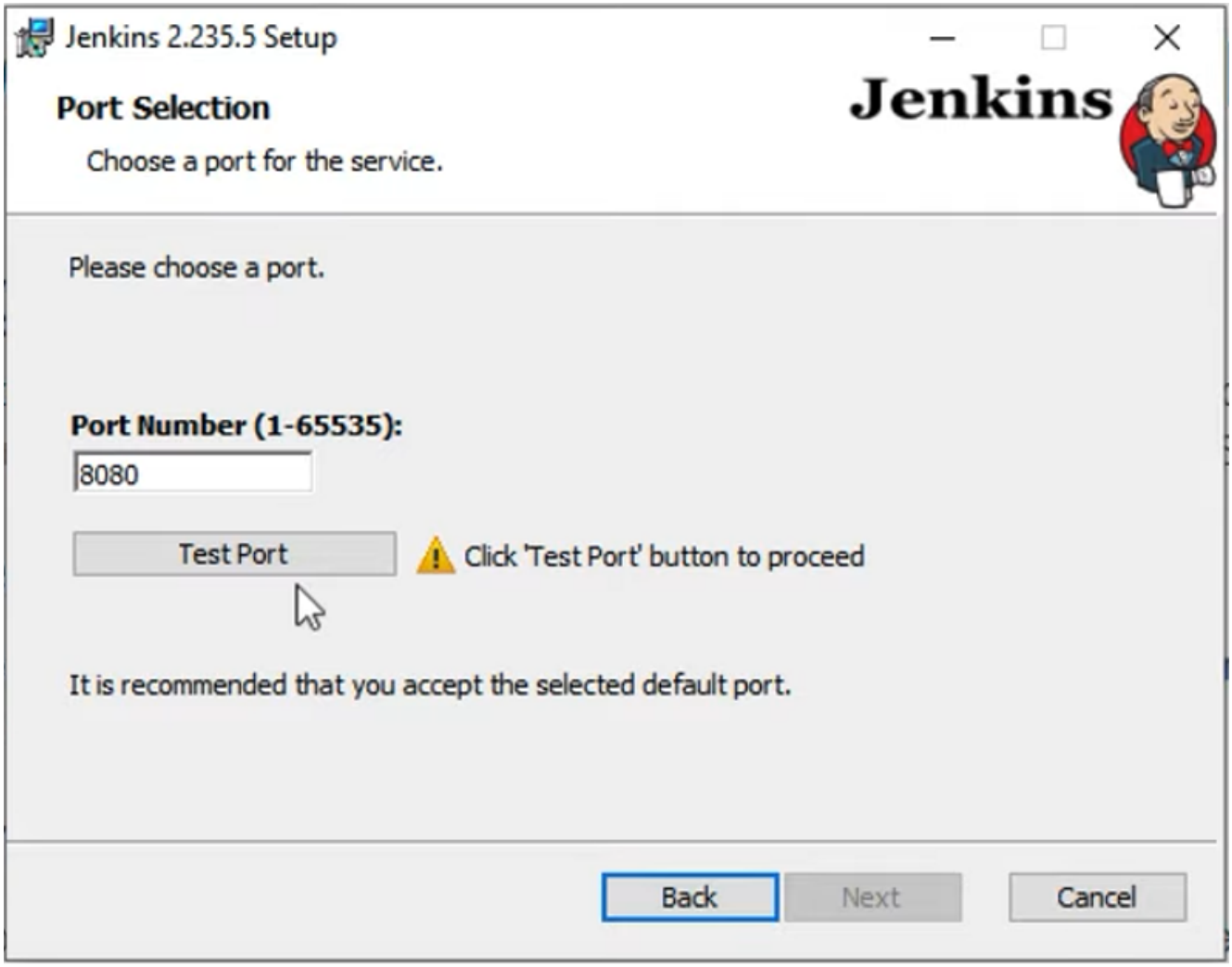

Fig.4

5. Test the port to see if it is available:

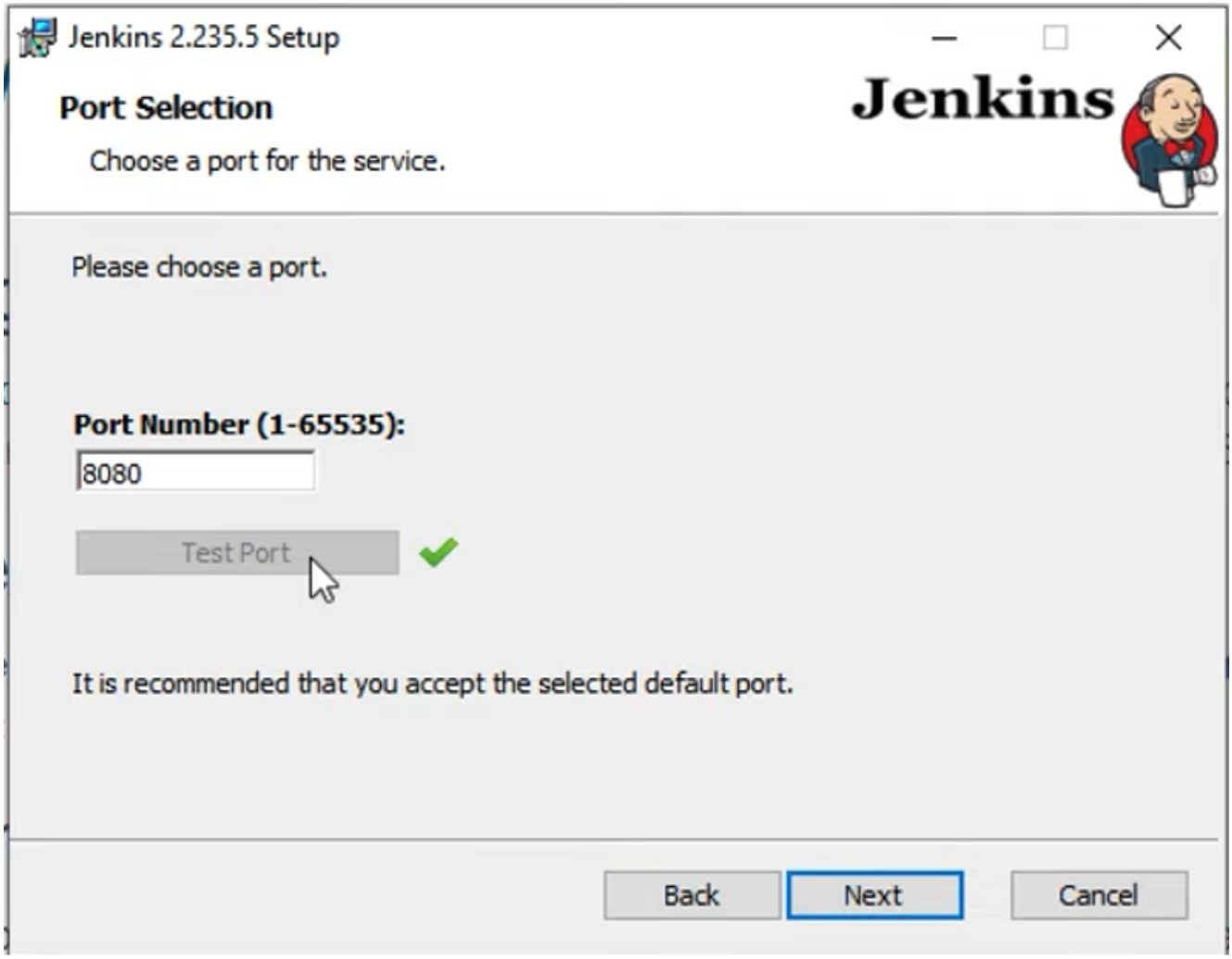

Fig.5

6. Select where your Java JDK is installed:

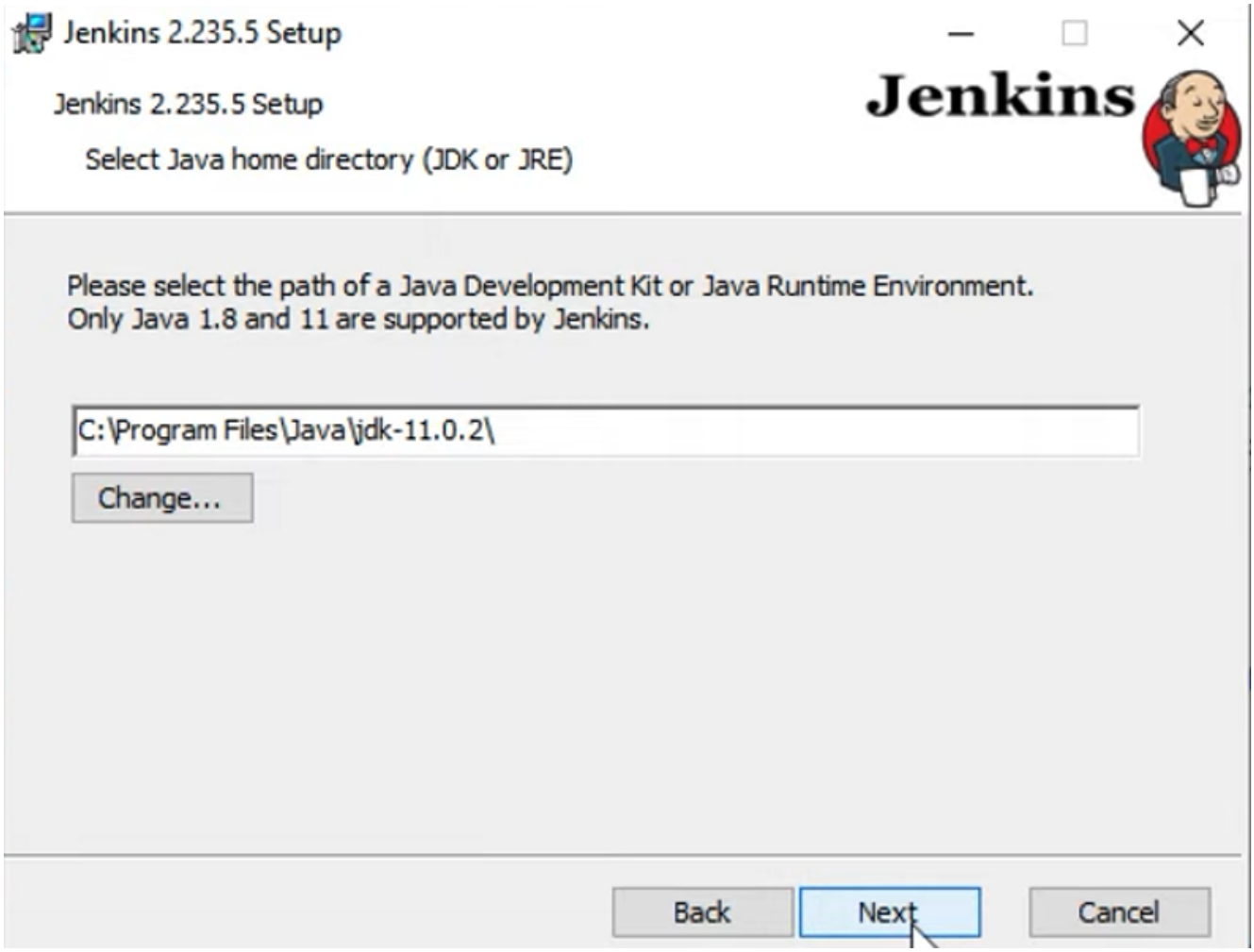

Fig.6

7. Click next and don't change `Firewall Exception`

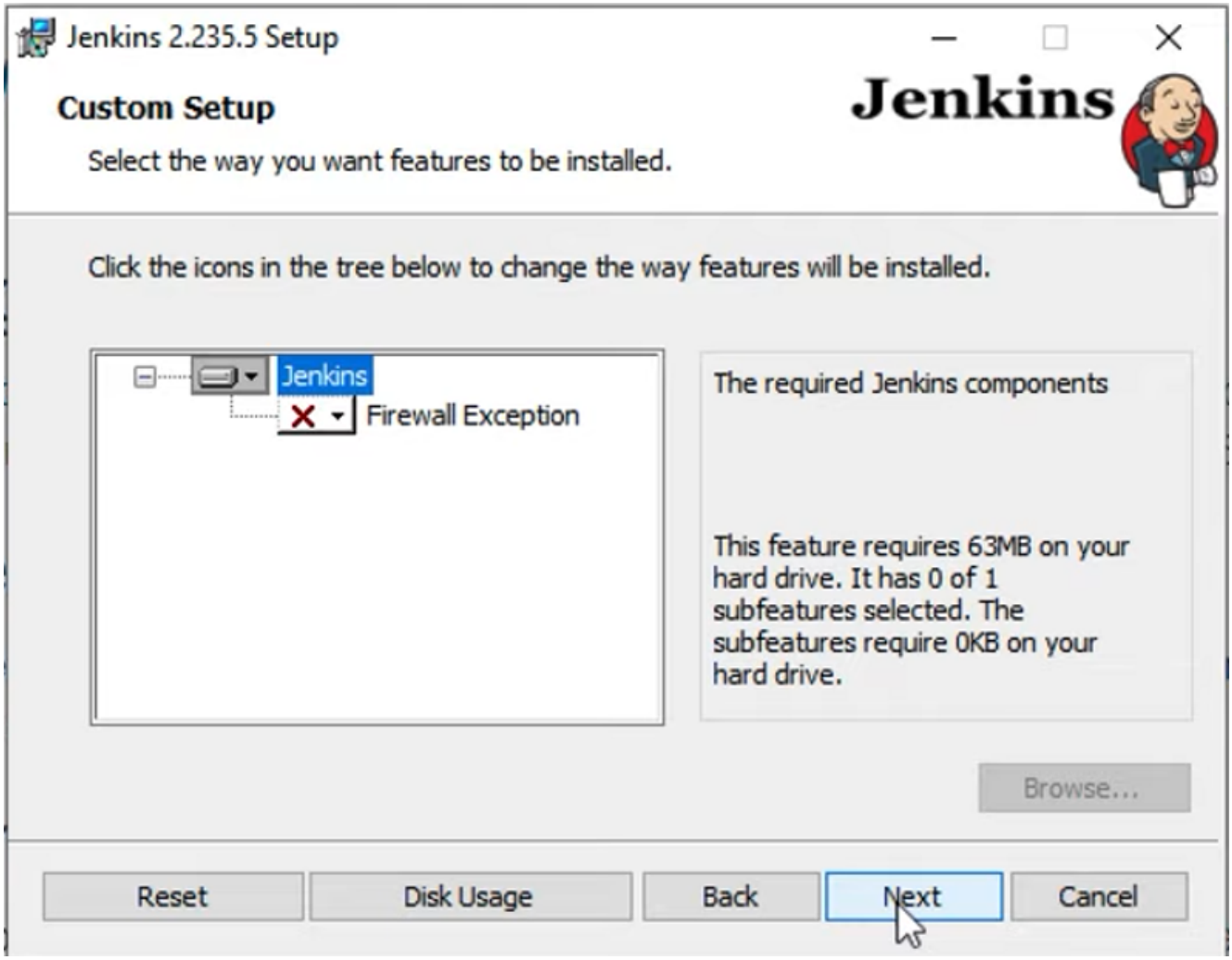

Fig.7

8. Install it

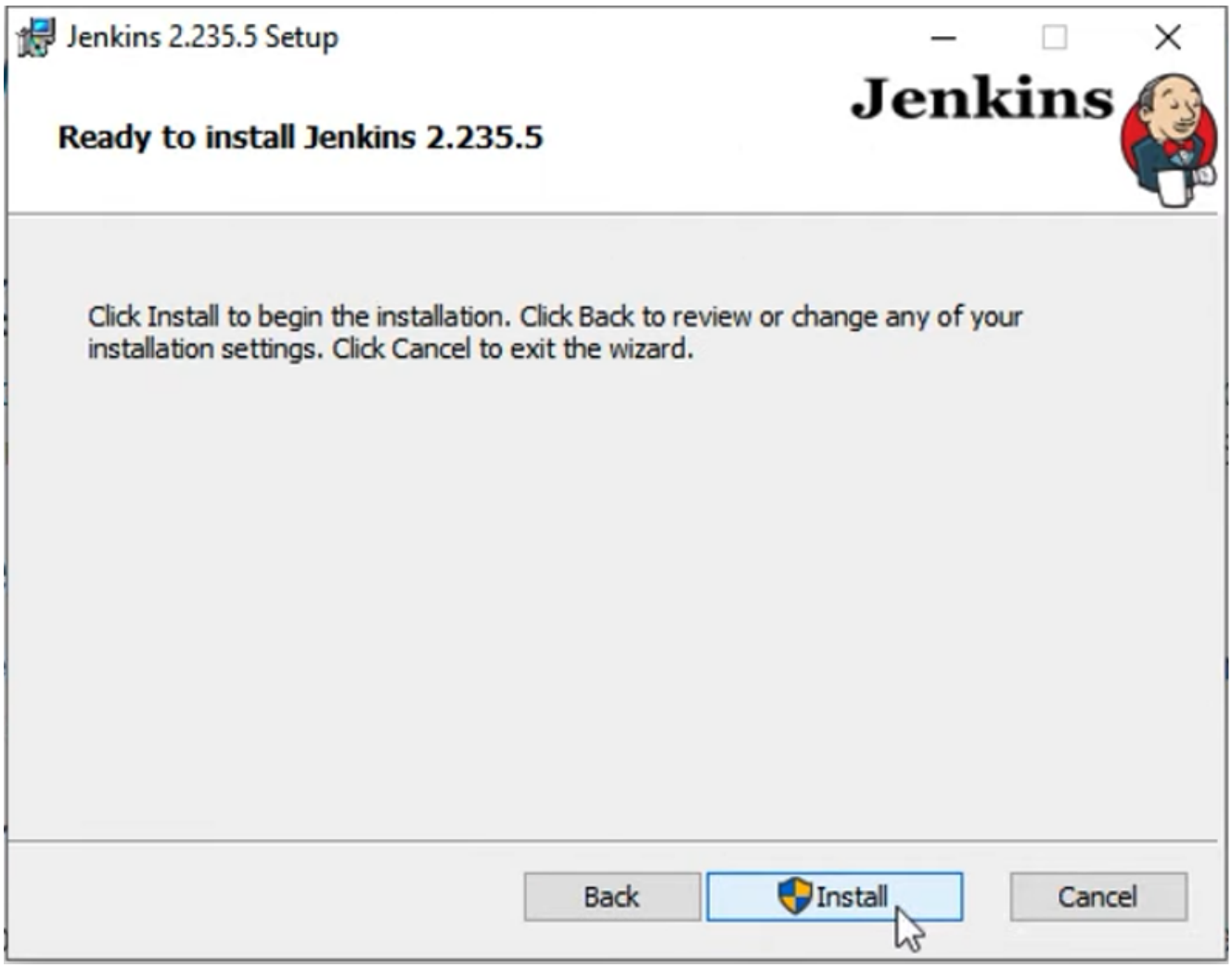

Fig.8

9. After install you need to unlock the Jenkins. Go to `http://localhost:{PORT_NUMBER}/` In my case is http://localhost:8080/

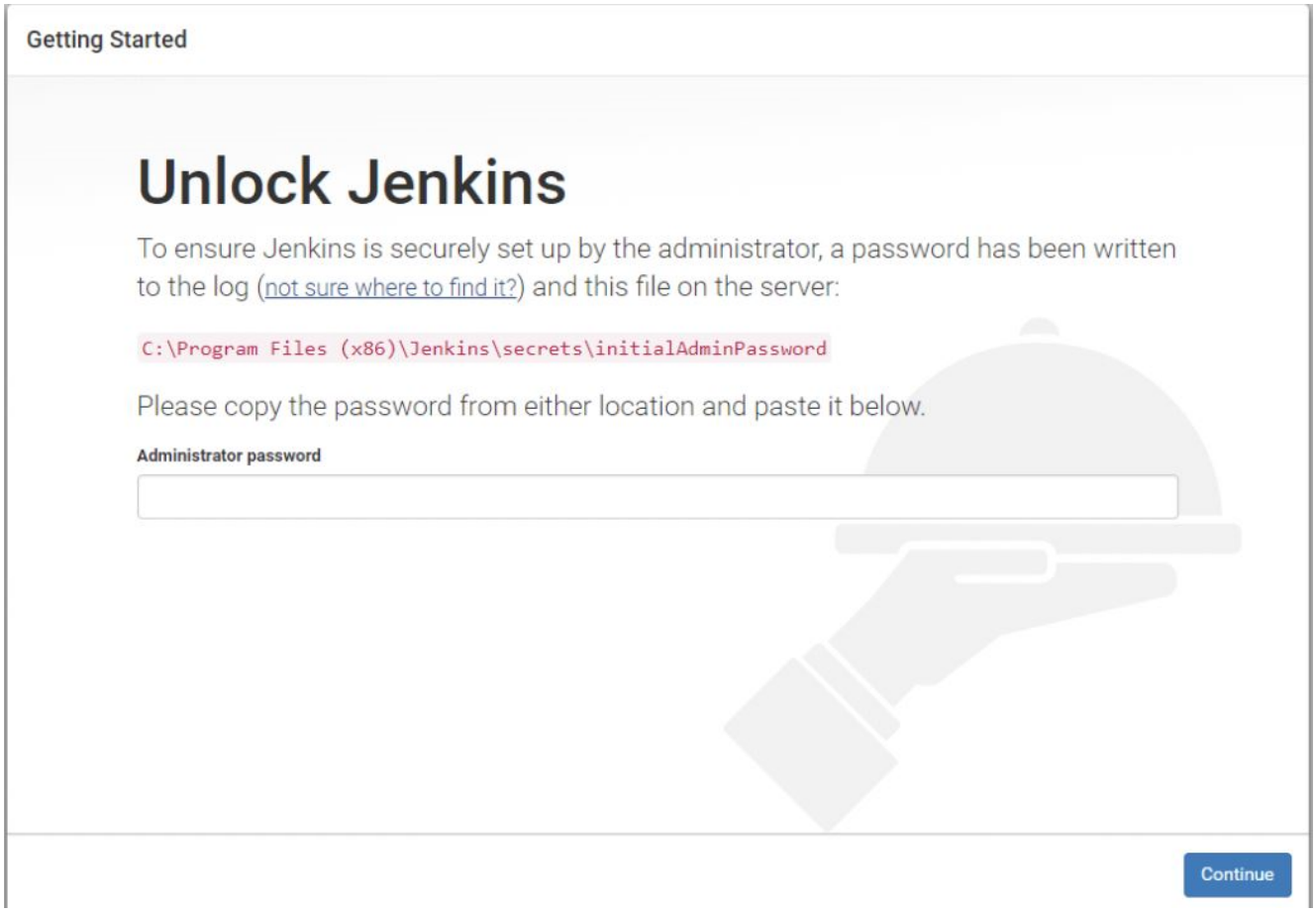

Fig.9

10. You will see a window which will require from you an initial administrator passsword For default installation location of the password is `C:\Program Files (x86)\Jenkins\.jenkins\secrets\initialAdminPassword`,

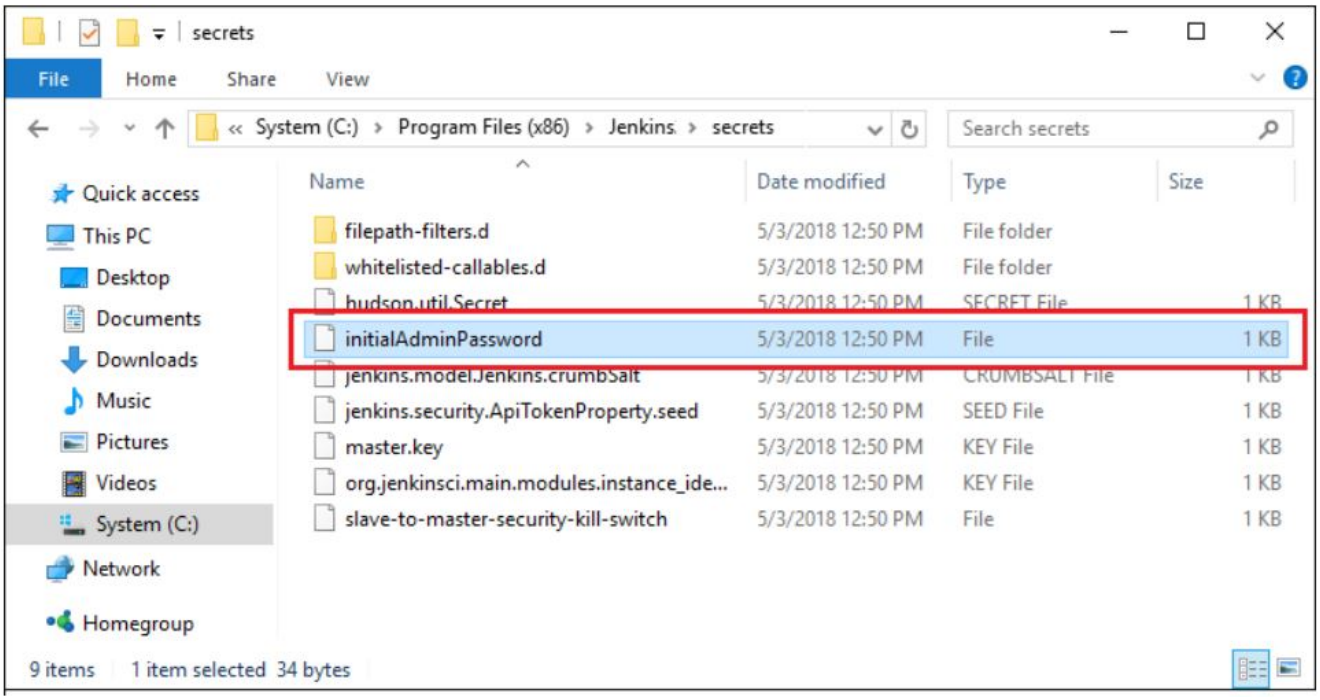

Fig.10

Copy and past it into the window dialog

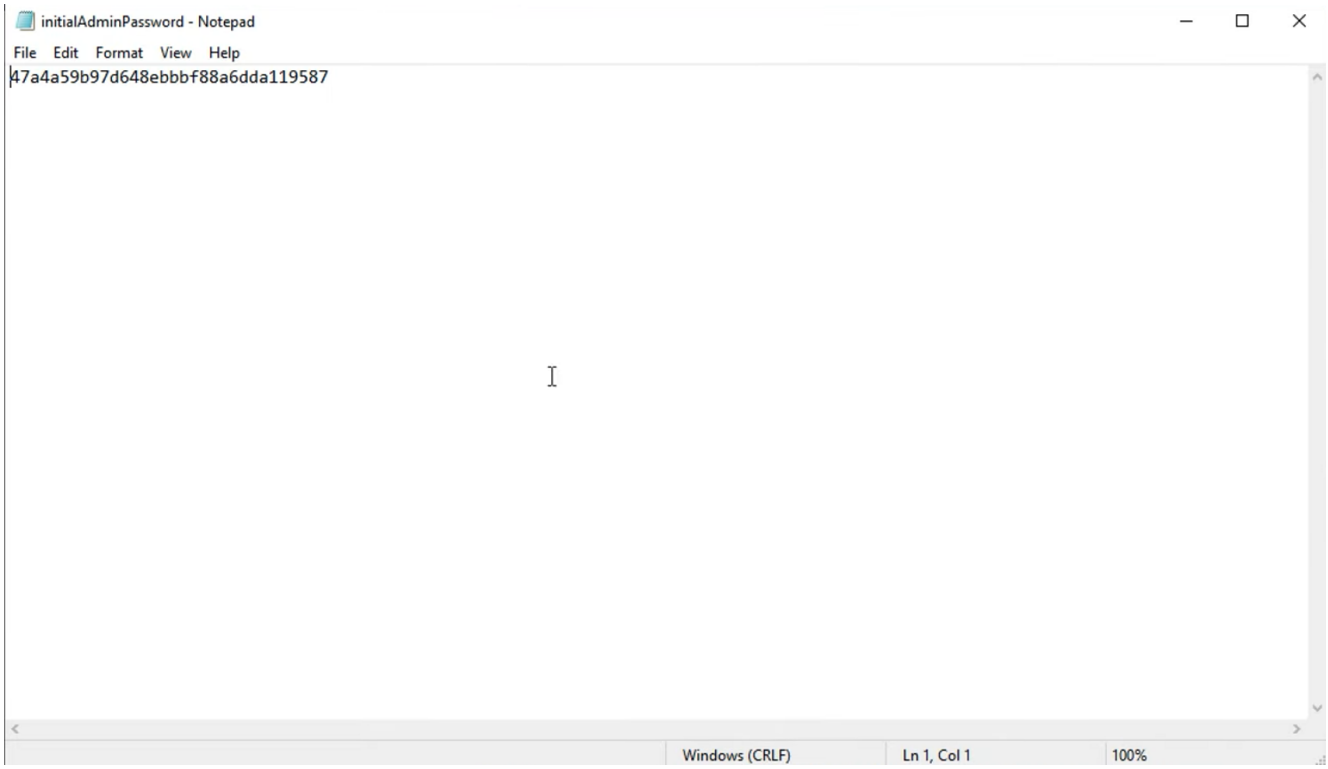

Fig.11

11. After you enter the passsword you will have 2 options : `Install suggested plugins` or `Select plugins to install`

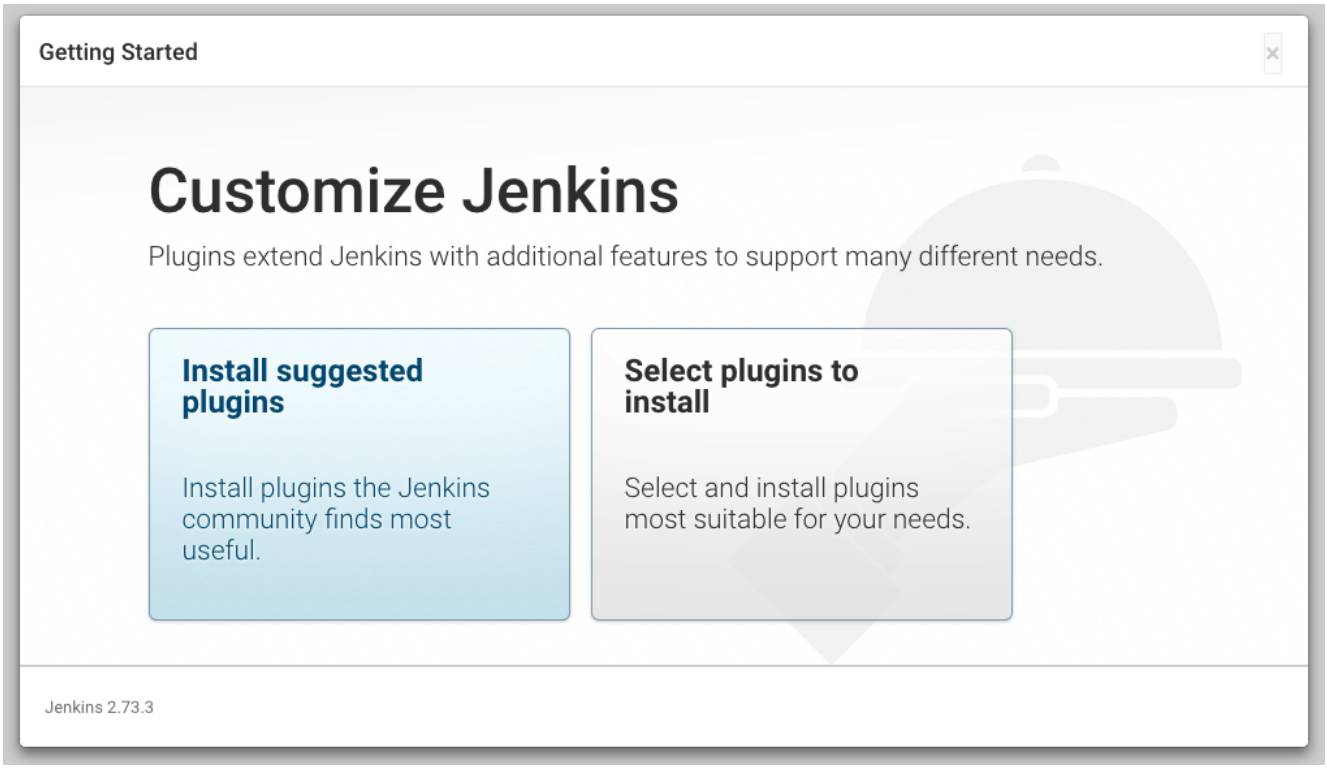

Fig.12

12. After you have choosen, the plugin will start to install and you will see the progress.

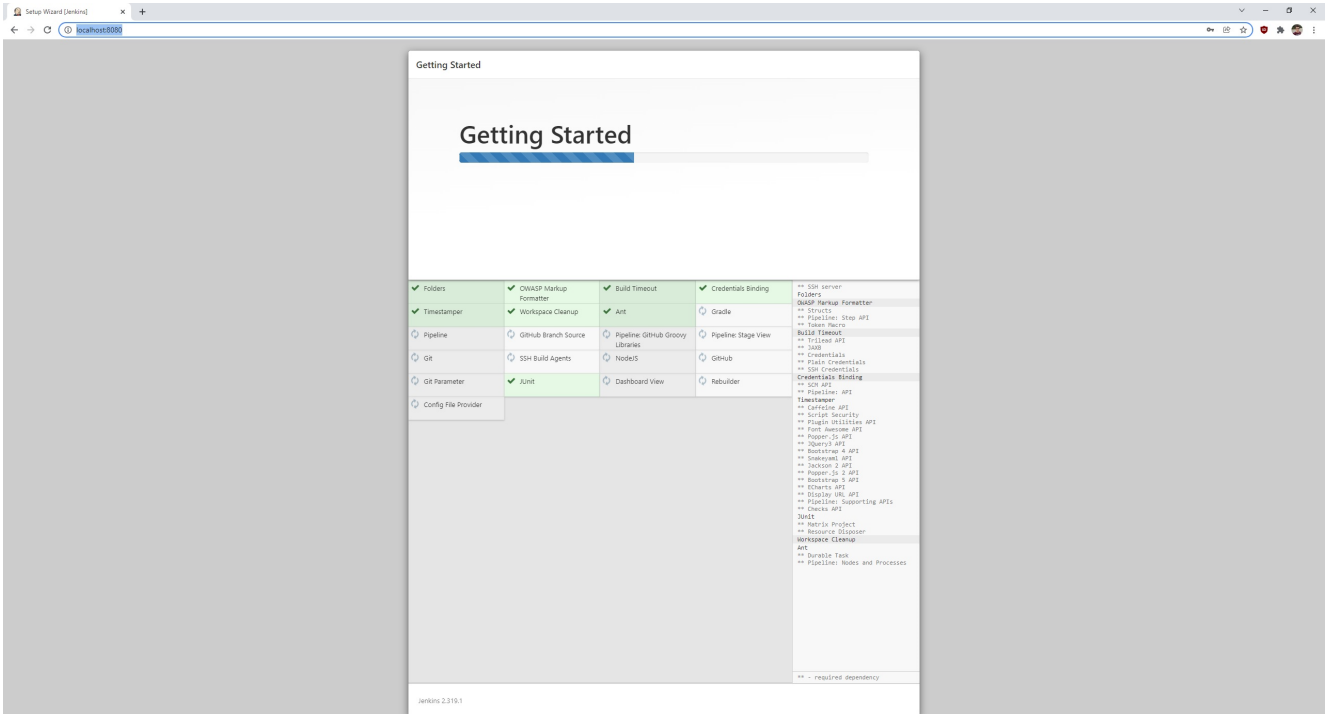

Fig.13

13. After the plugins have been installed you need to create an admin

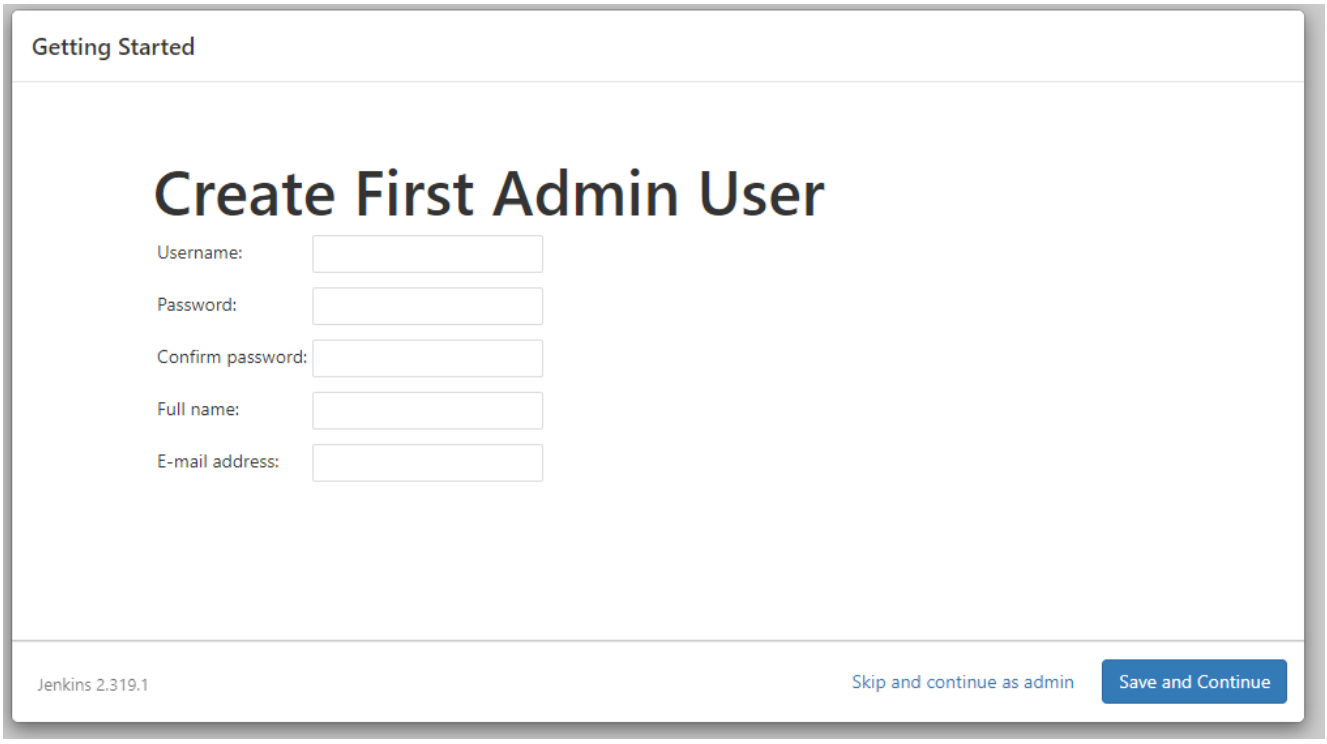

Fig.14

14. Set the URL address and press `Save and Finish`

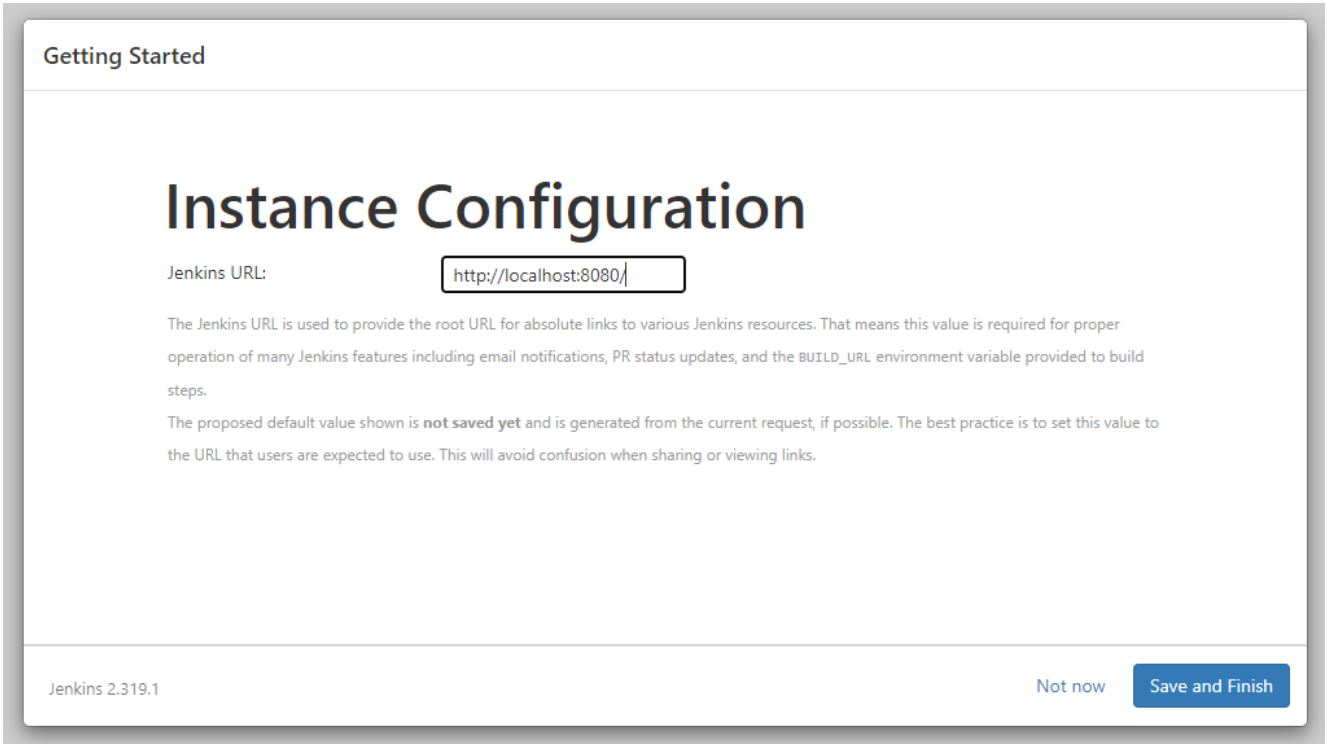

Fig.15

15. Now Jenkins is ready

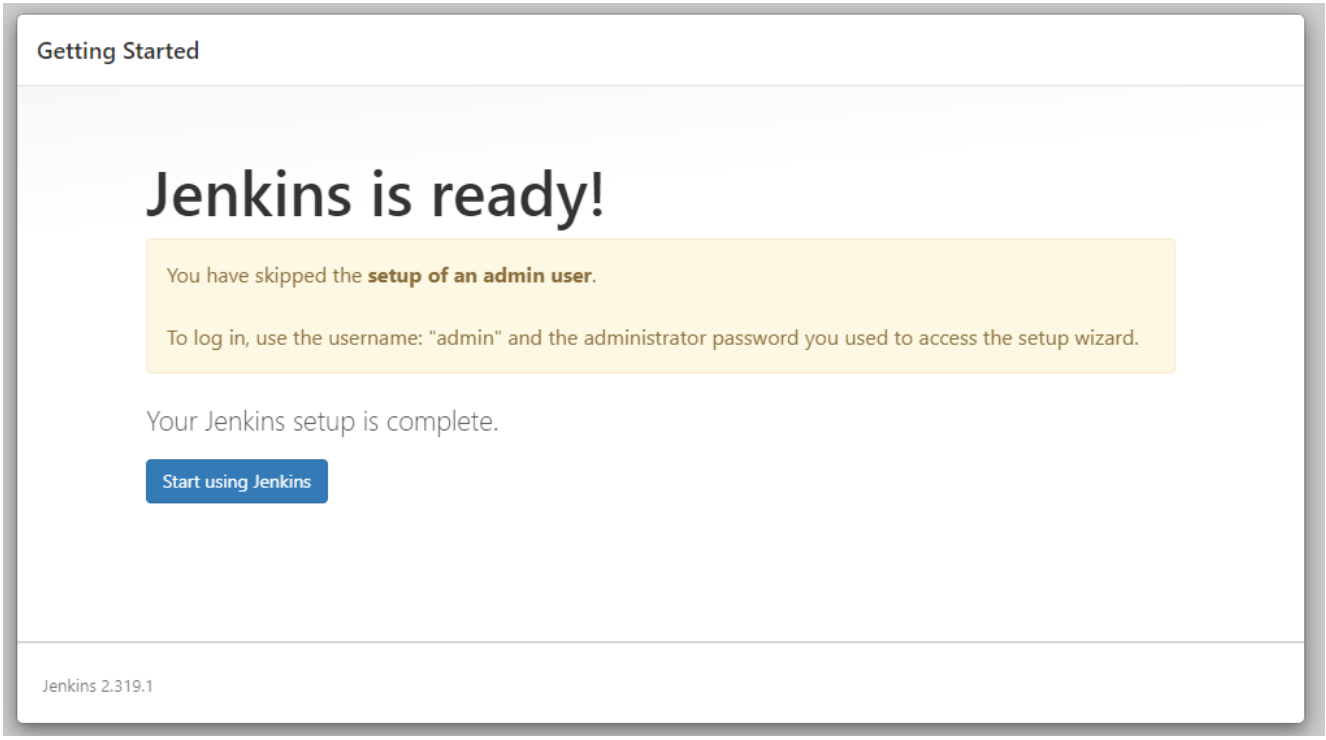

Fig.16

Homework 1

Steps

1. Go to https://github.com/ and login into your account

2. Create a new repository. Click on `new`

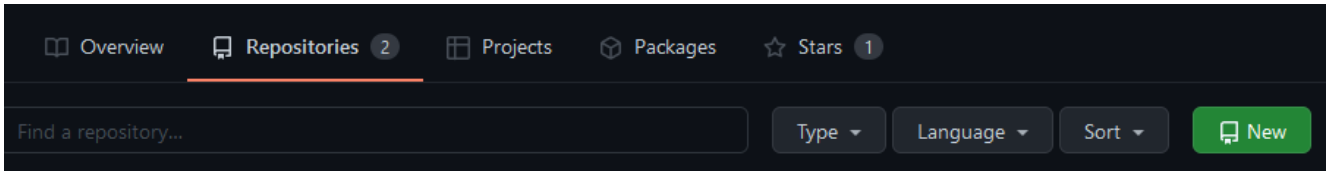

Fig.1

3. Choose a name and leave it as public. Press `Create Repository`

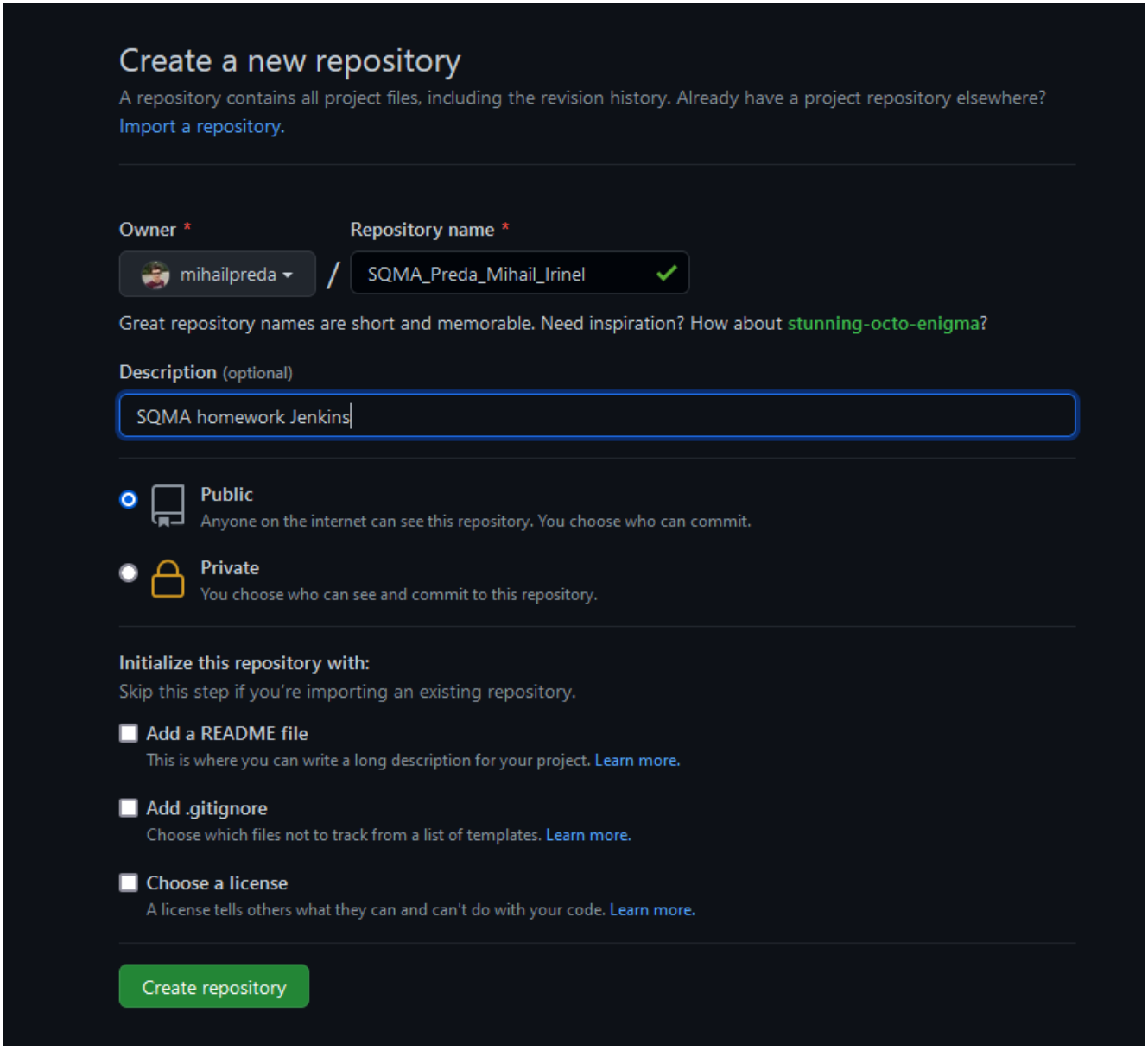

Fig.2

4. Set the origin to your repo

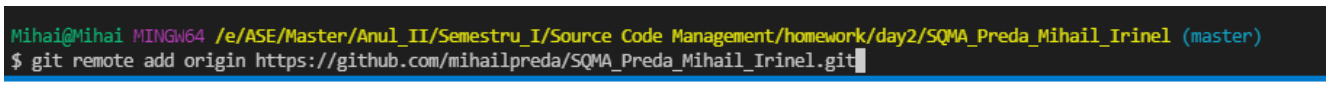

Fig.3

5. Add a new Job :

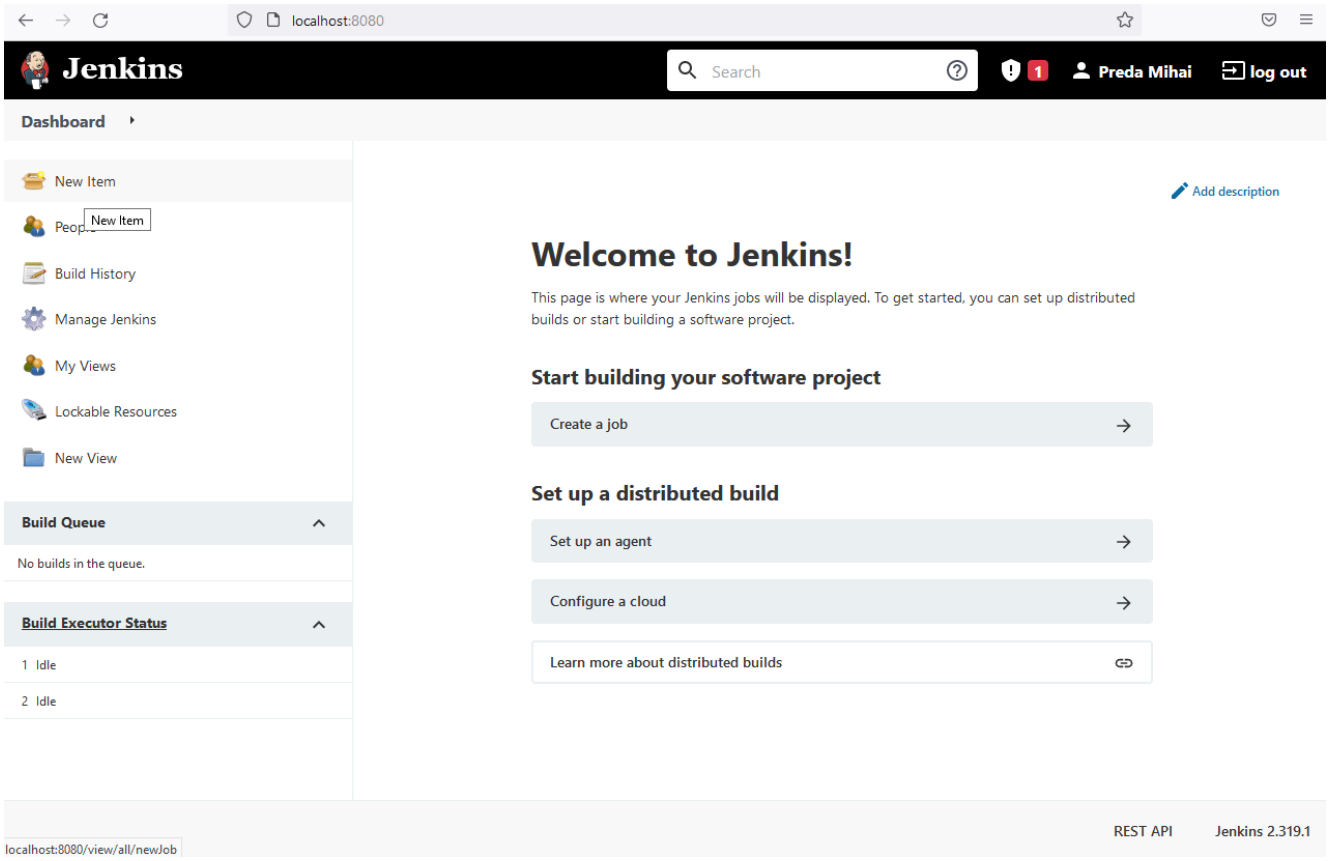

Fig.4

6. Enter Job Name

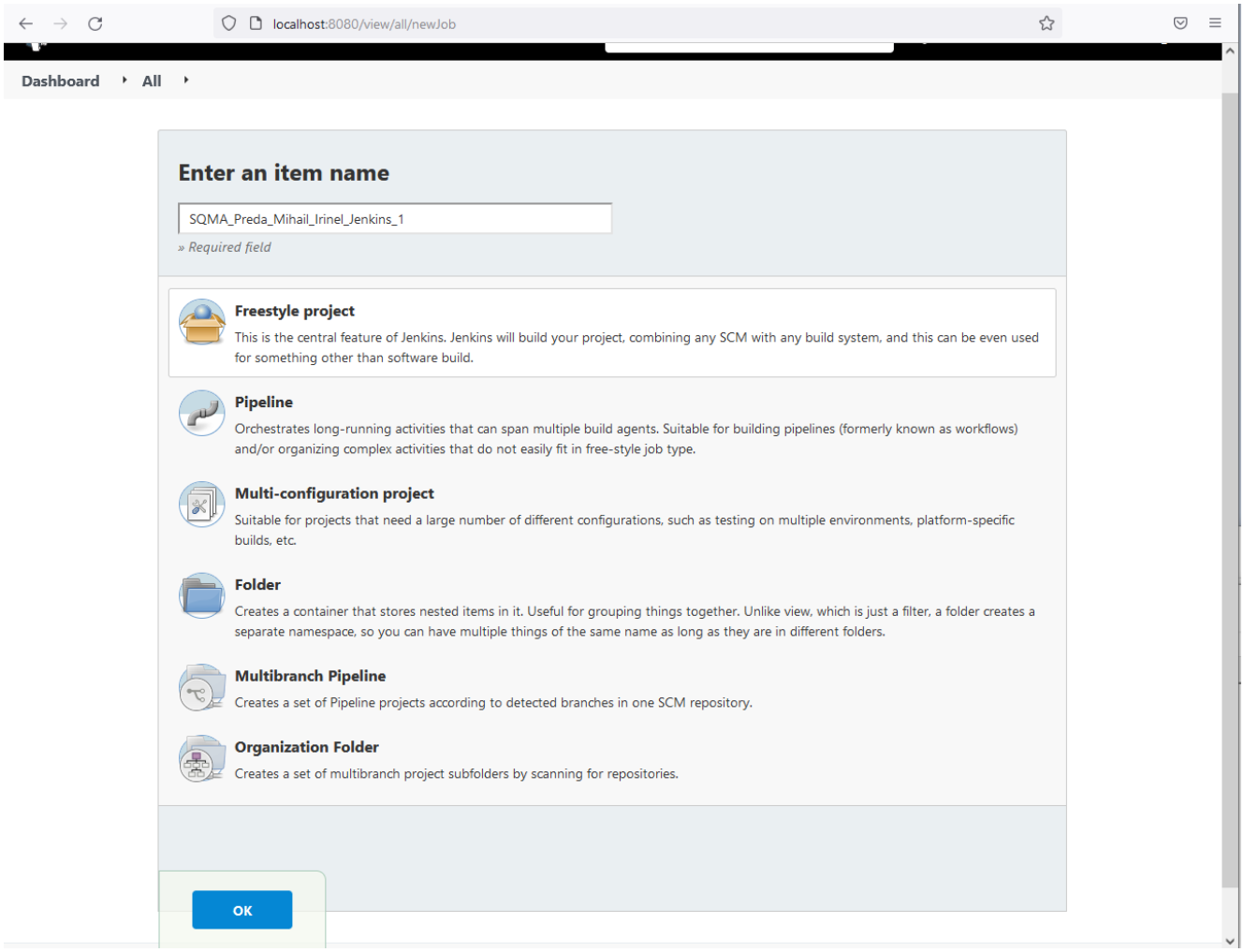

Fig.5

7. Enter a description. Go to `Source Code Management`, check `Git`, put the link to the repository and click `Save`

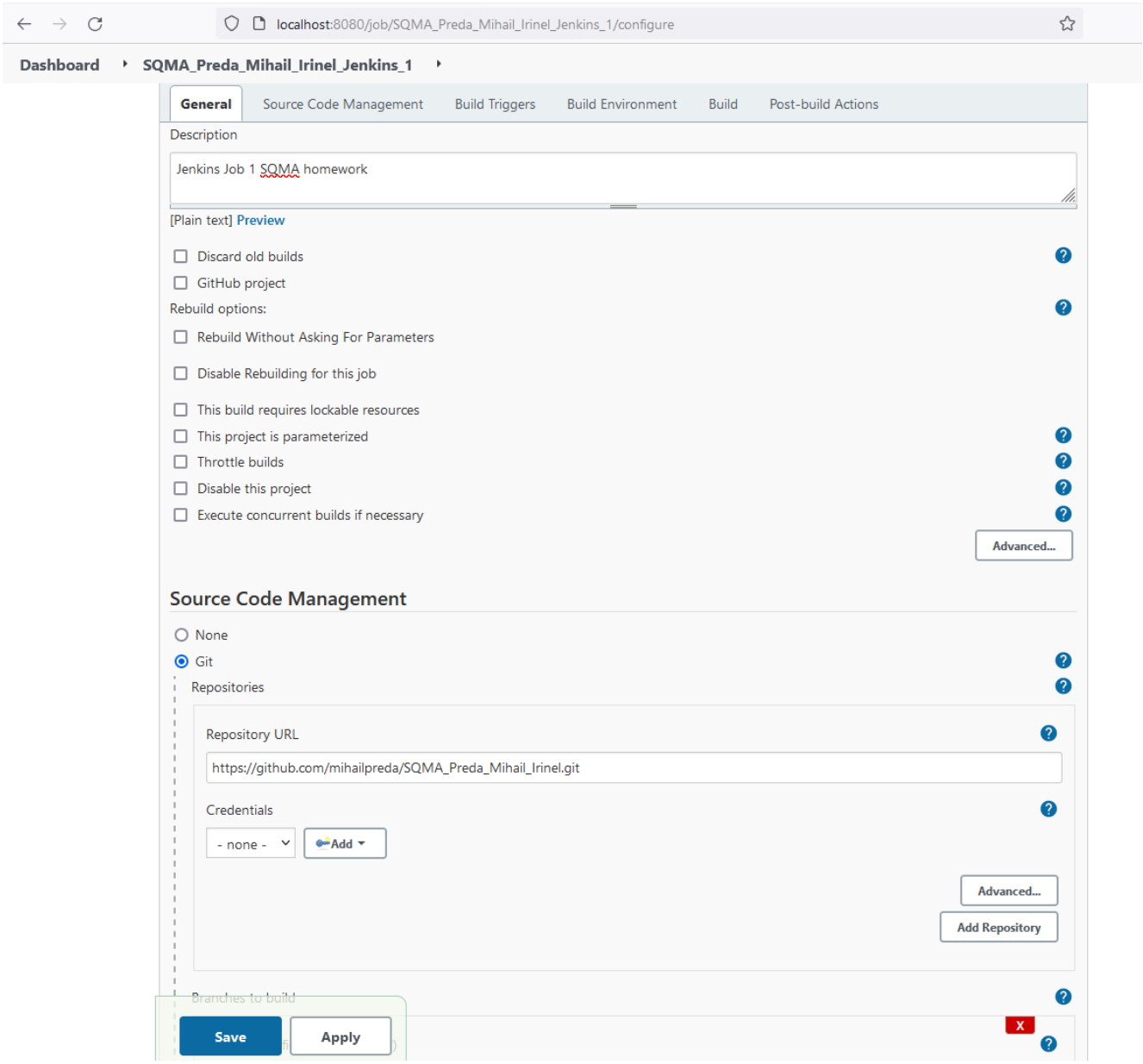

Fig.6

8. If everyting is okey you will seein the workspace that the Jenkins successfully downloaded the files from Github.

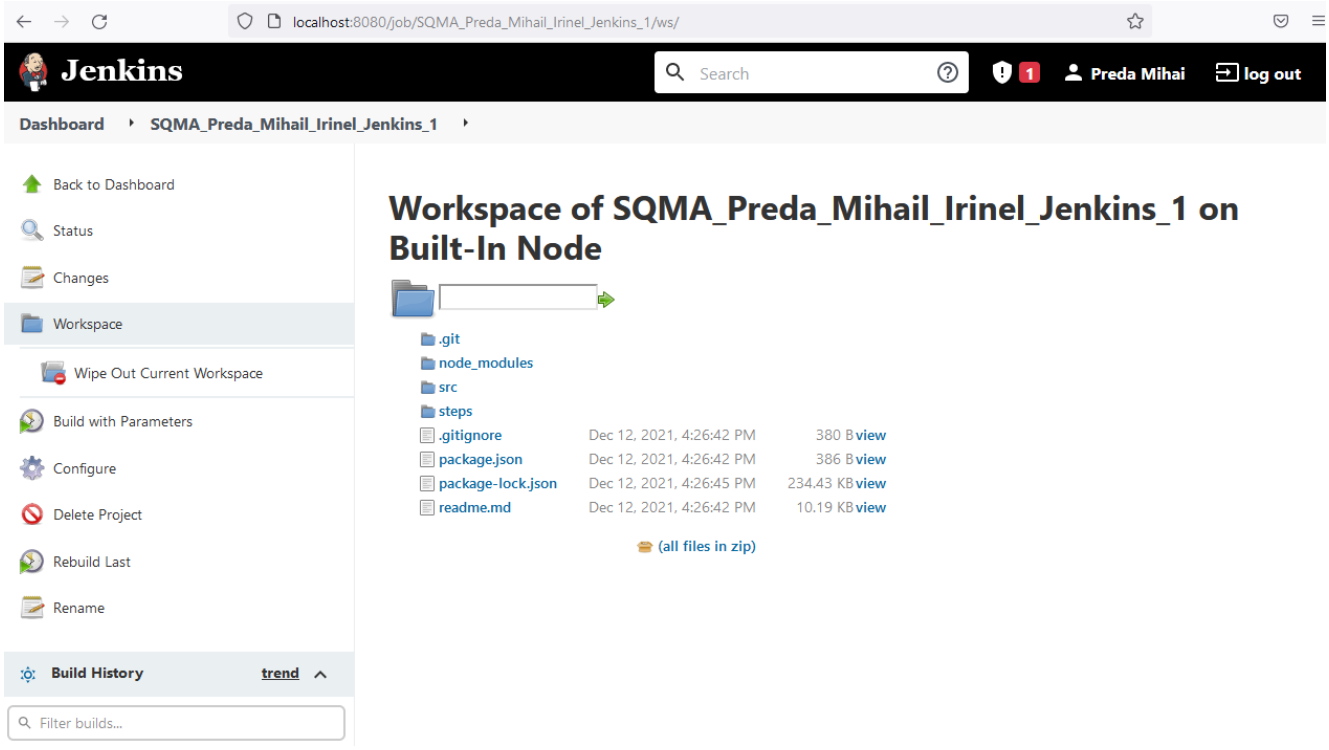

Fig.7

9. Then go to `Configure`

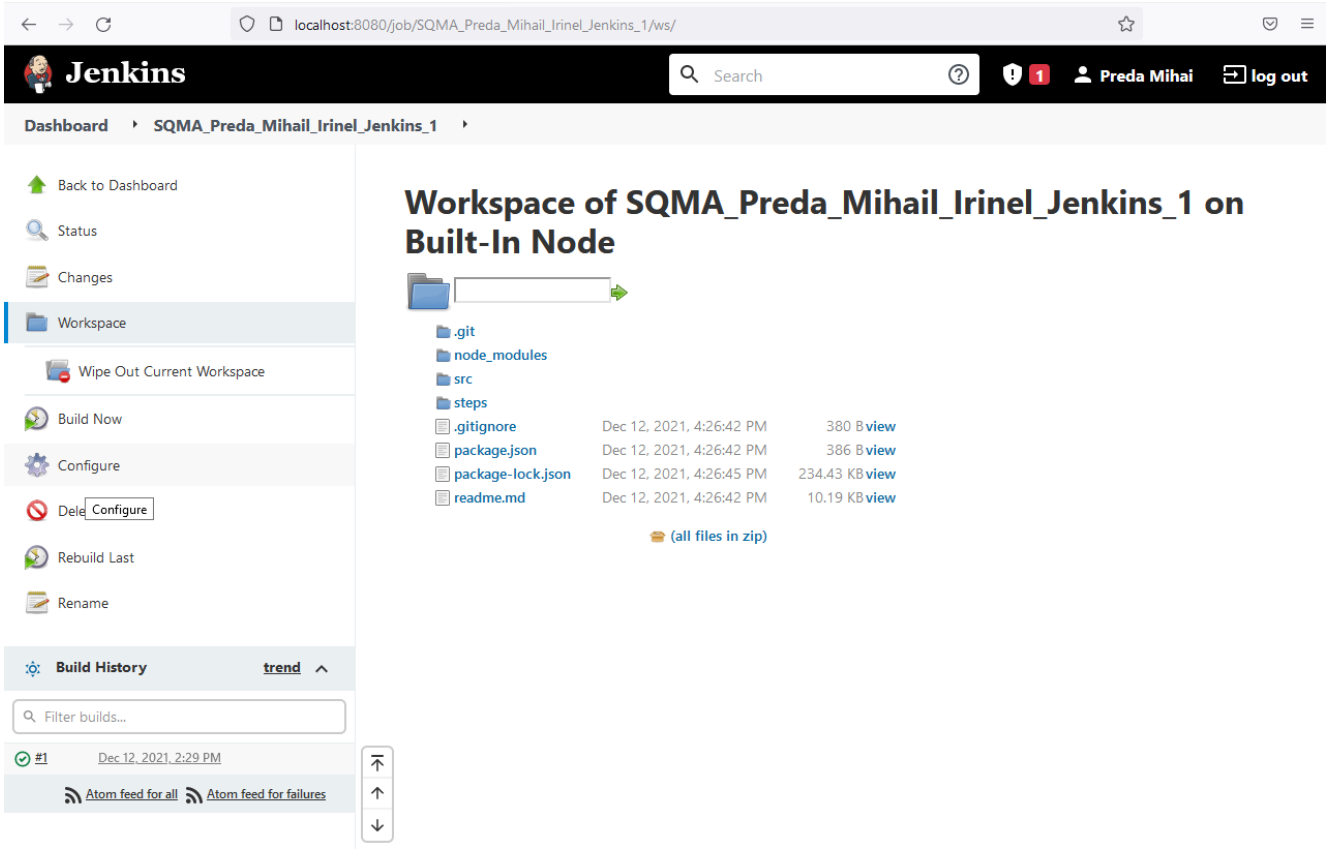

Fig.8

10. Go to `General` and tick `This project is parameterized`. After that you put the parameters that you have. In my case I have 2 test suits, each one is run by specifying `npm run testA` or `npm run testB`

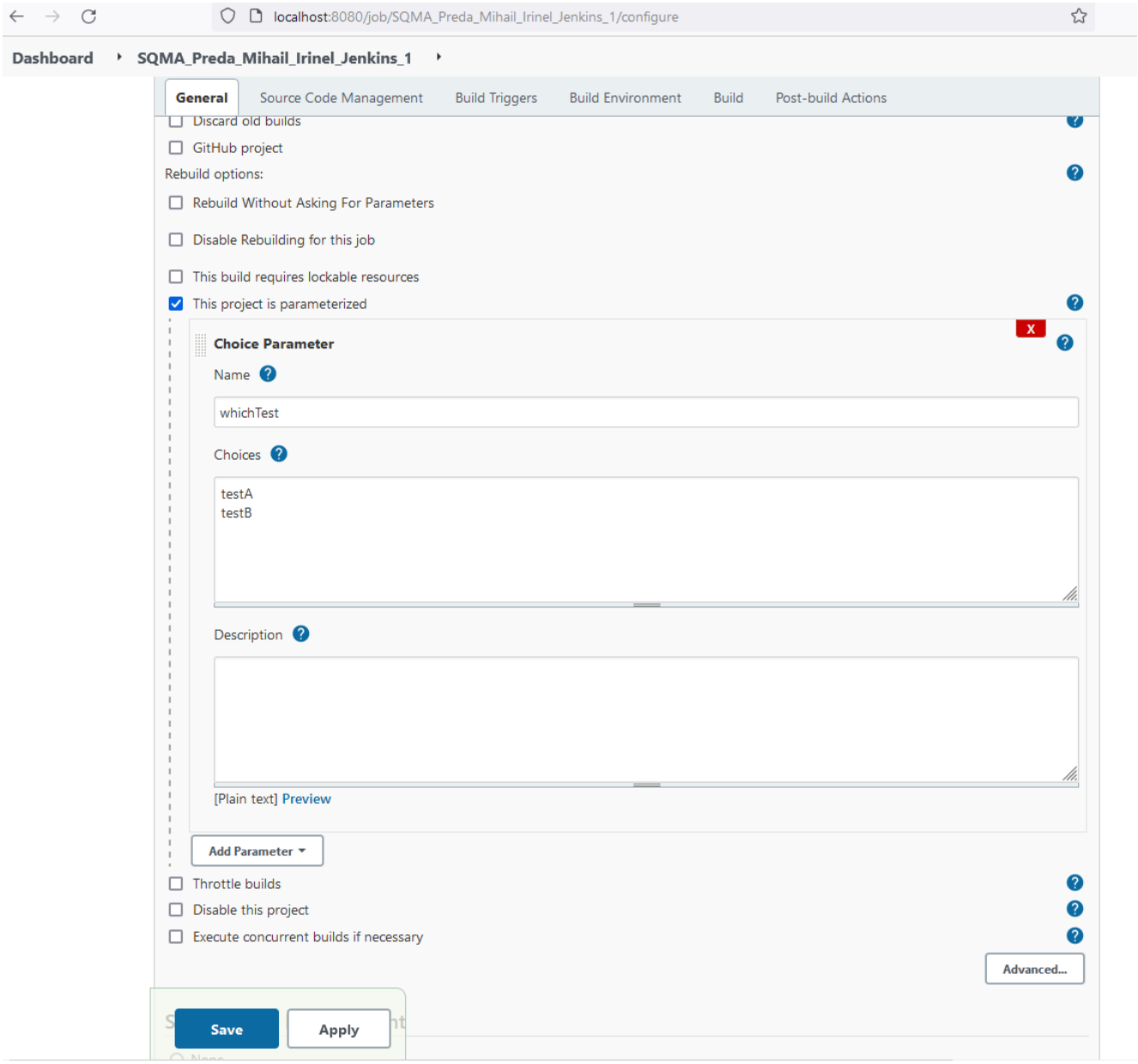

Fig.9

11. Then go to `Build`, add a `Windows batch command` and add the following line `npm install && npm run %whichTest%`

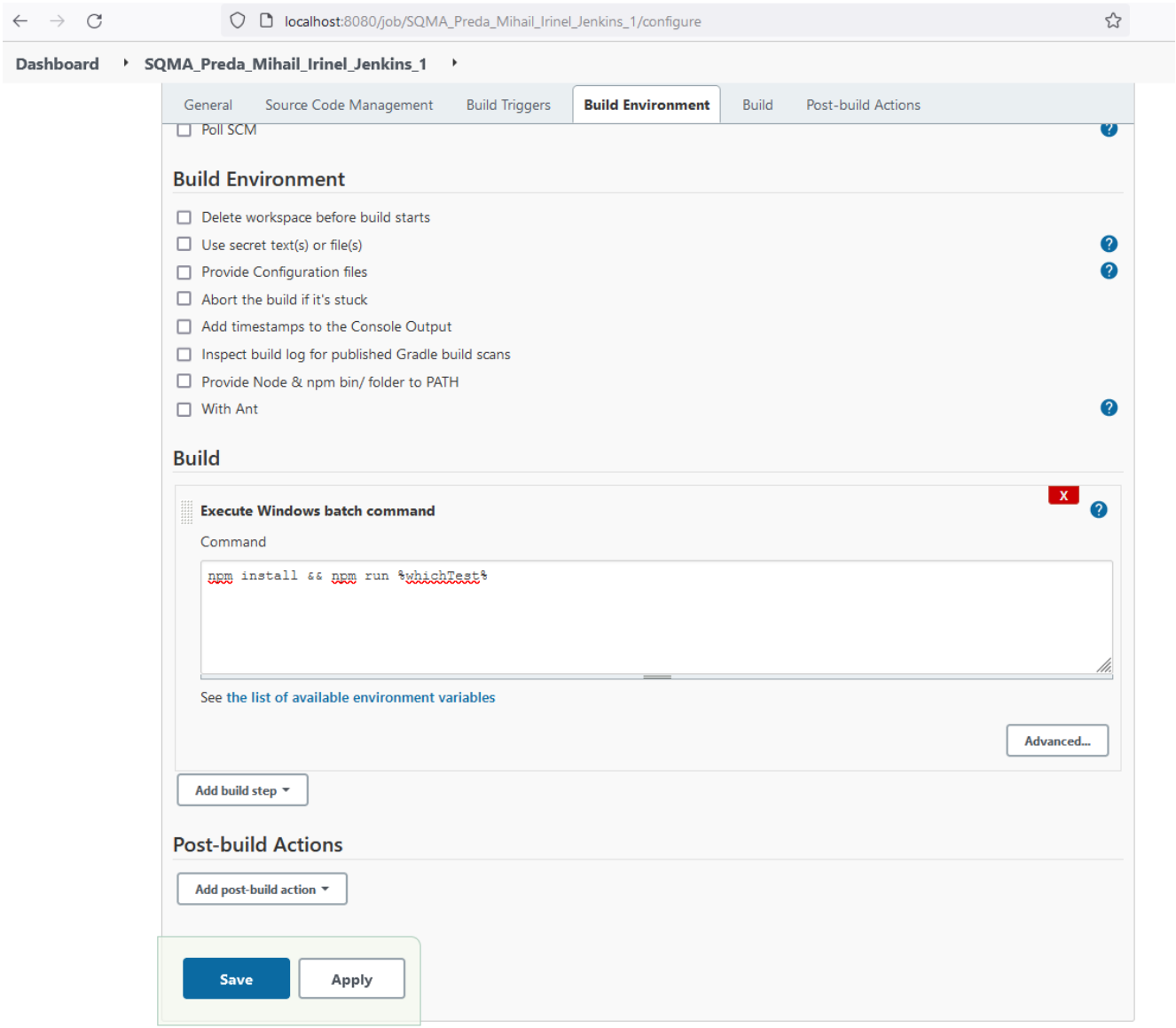

Fig.10

12. Click on `Build with parameters`

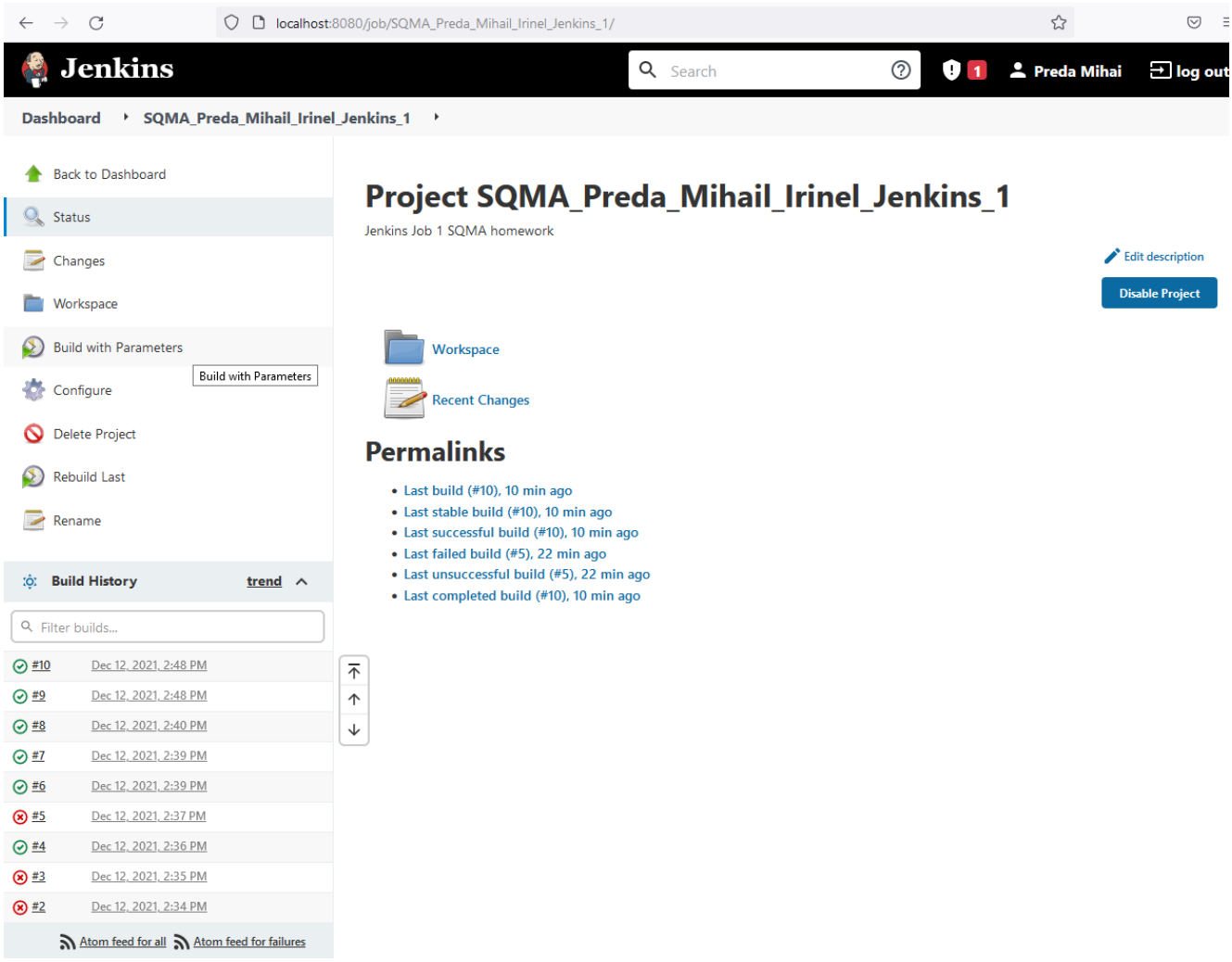

Fig.11

13. Choose `testA`, then press `Build`.

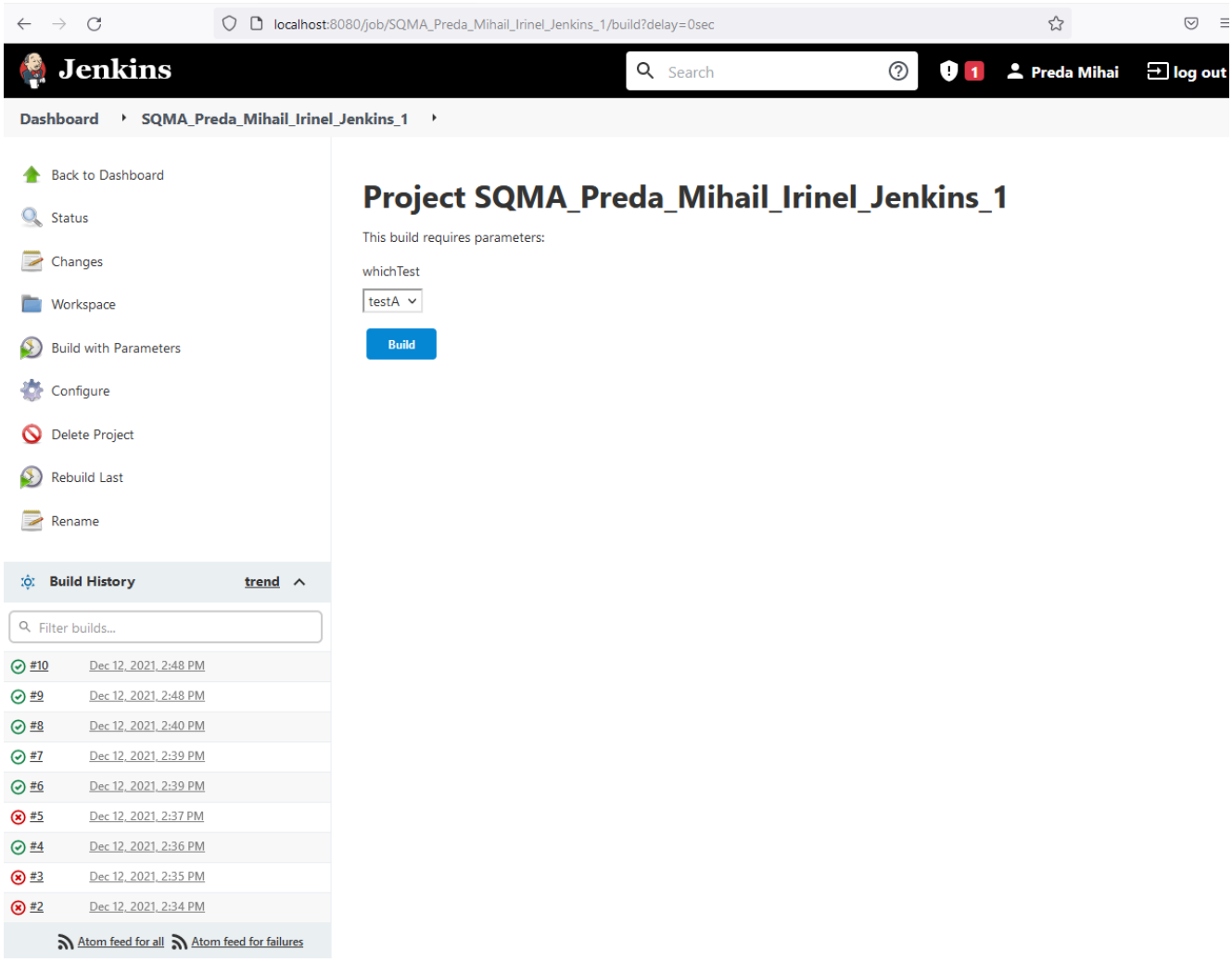

Fig.12

14. Click on the latest done build, go to `Console Output` and see the results :

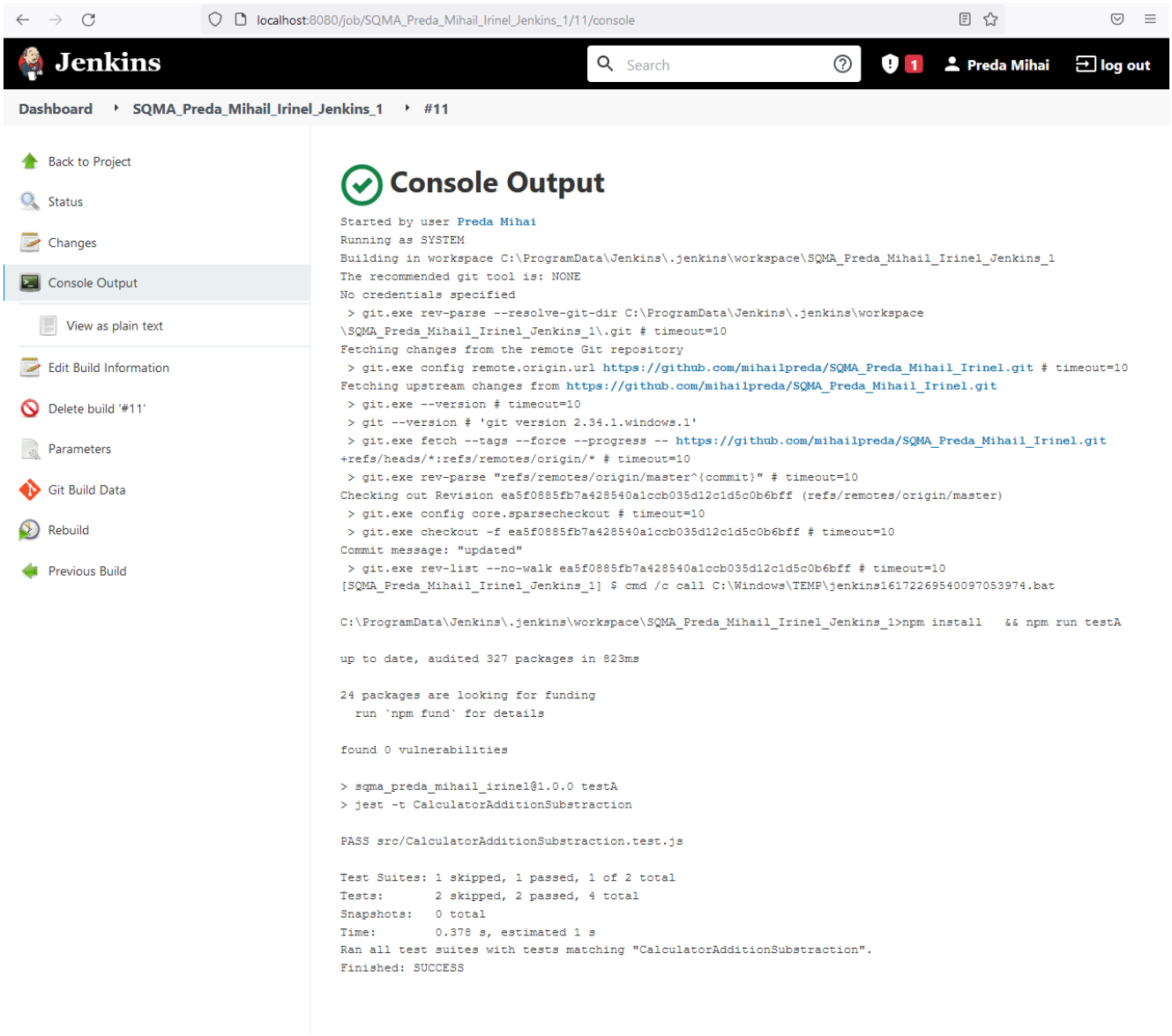

Fig.13

15. Click on `Build with parameters`

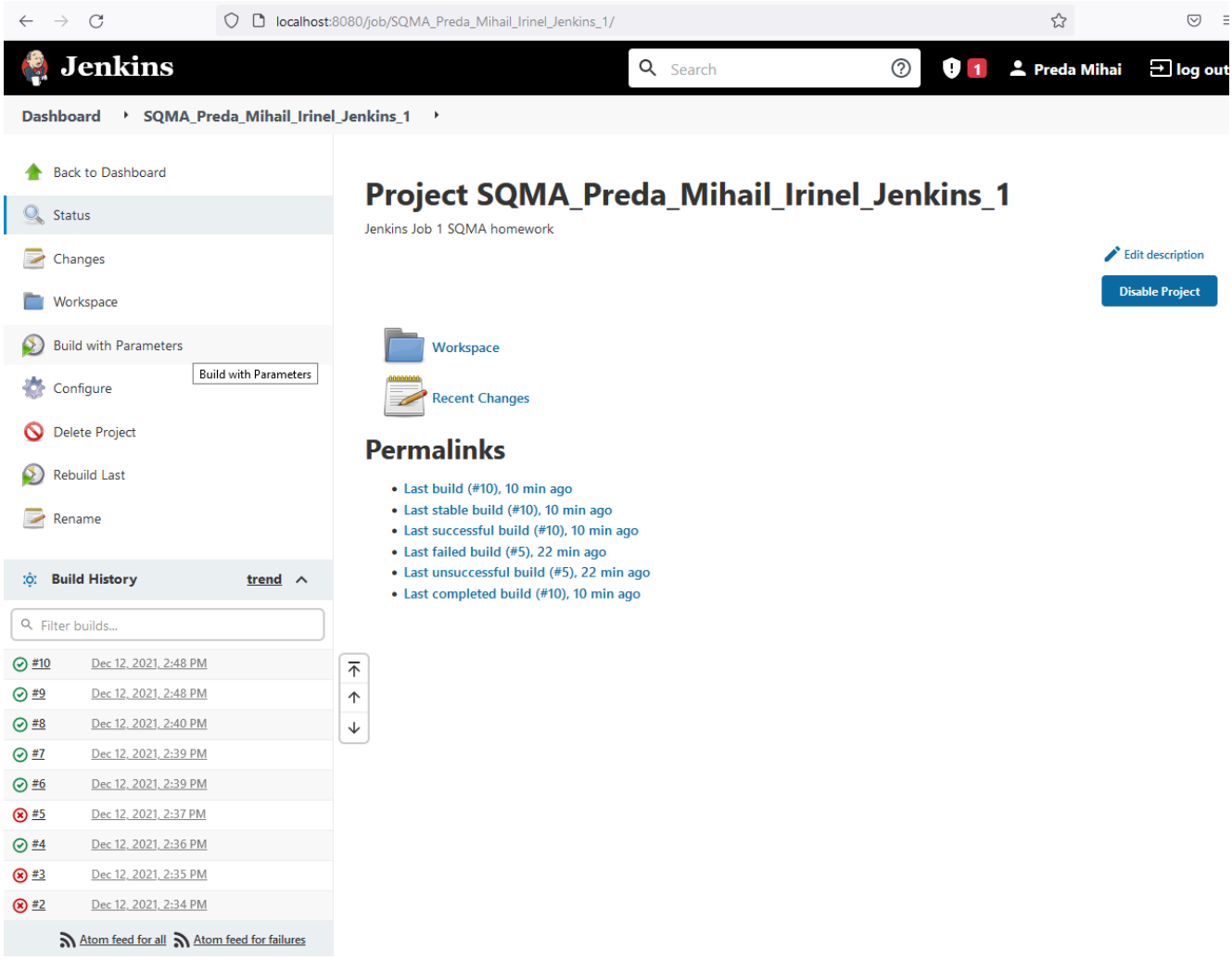

Fig.14

16. Choose `testB`, then press `Build`.

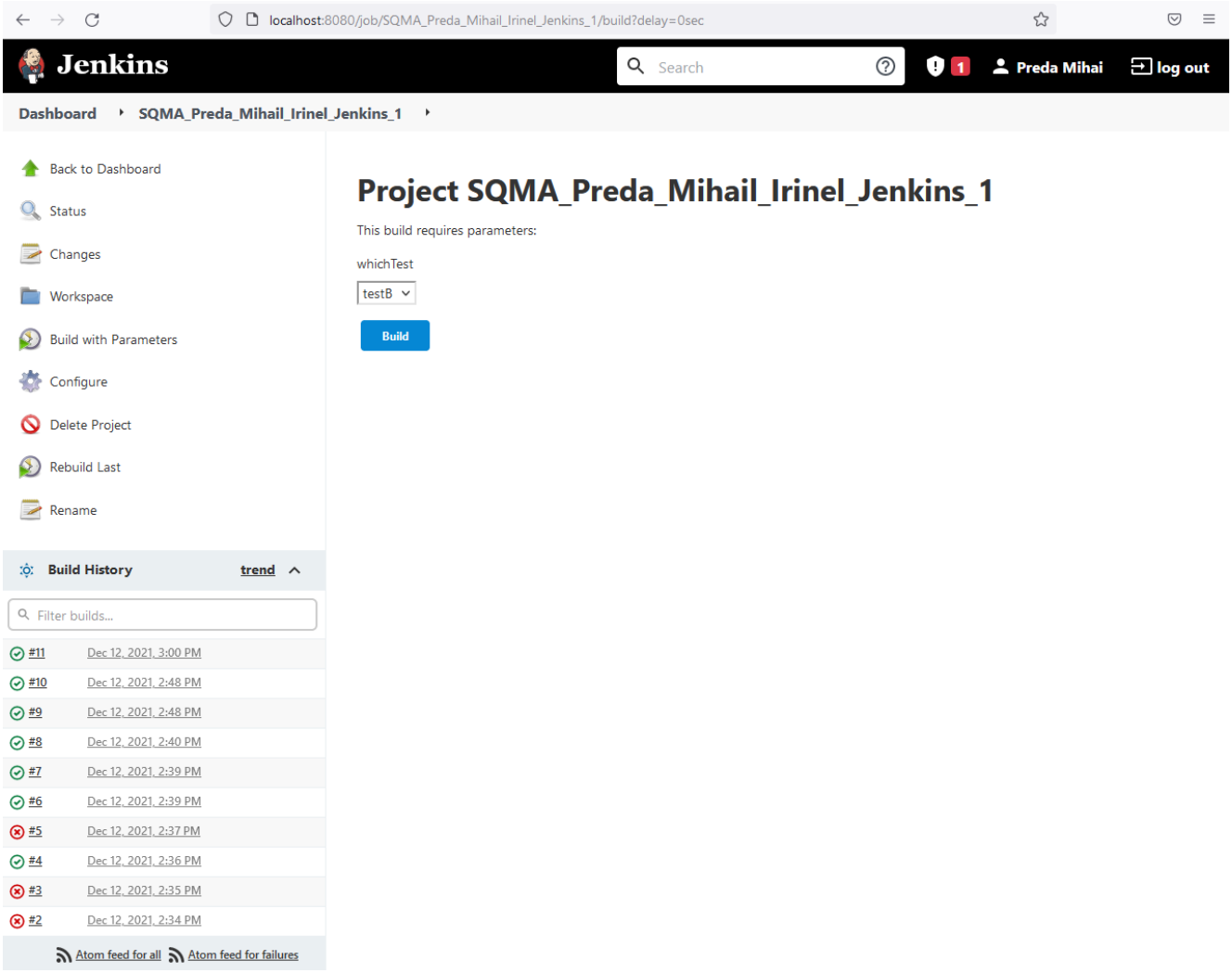

Fig.15

17. Click on the latest done build, go to `Console Output` and see the results :

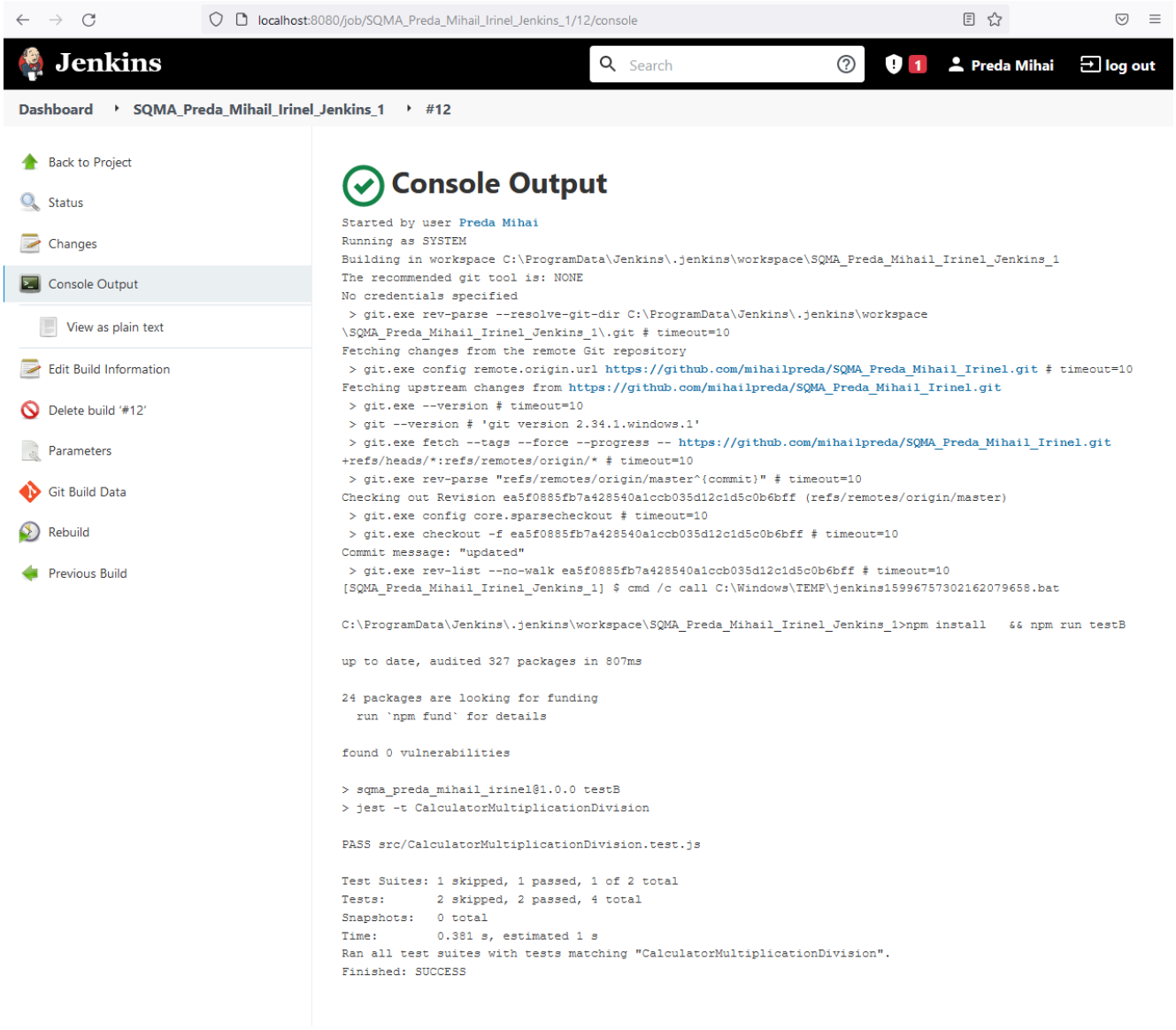

Fig.16

Homework 2

Steps

1. From `Dashboard`, click on `New Item`

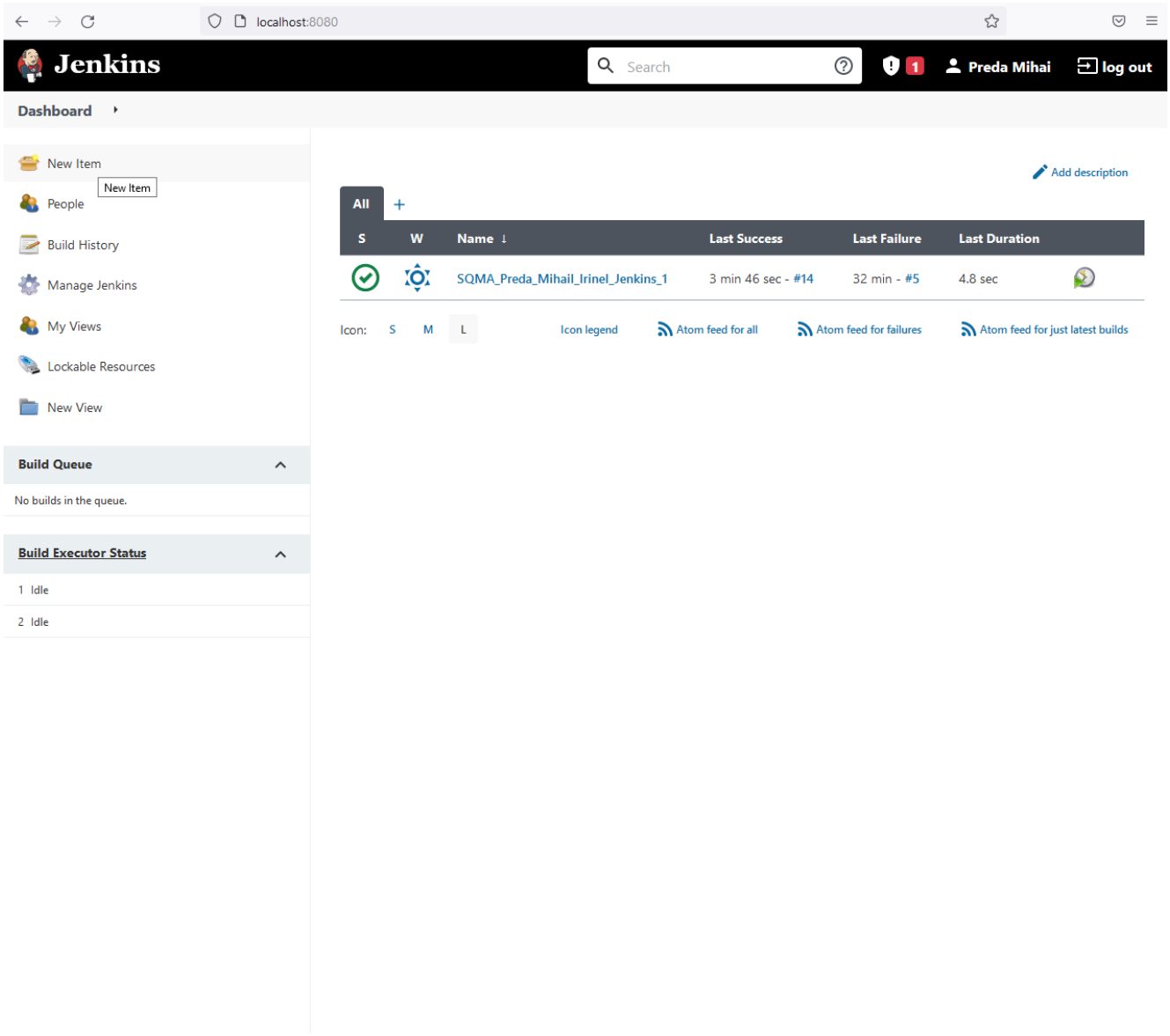

Fig.1

2. Enter a name for the item, select `Pipeline` and press `OK`

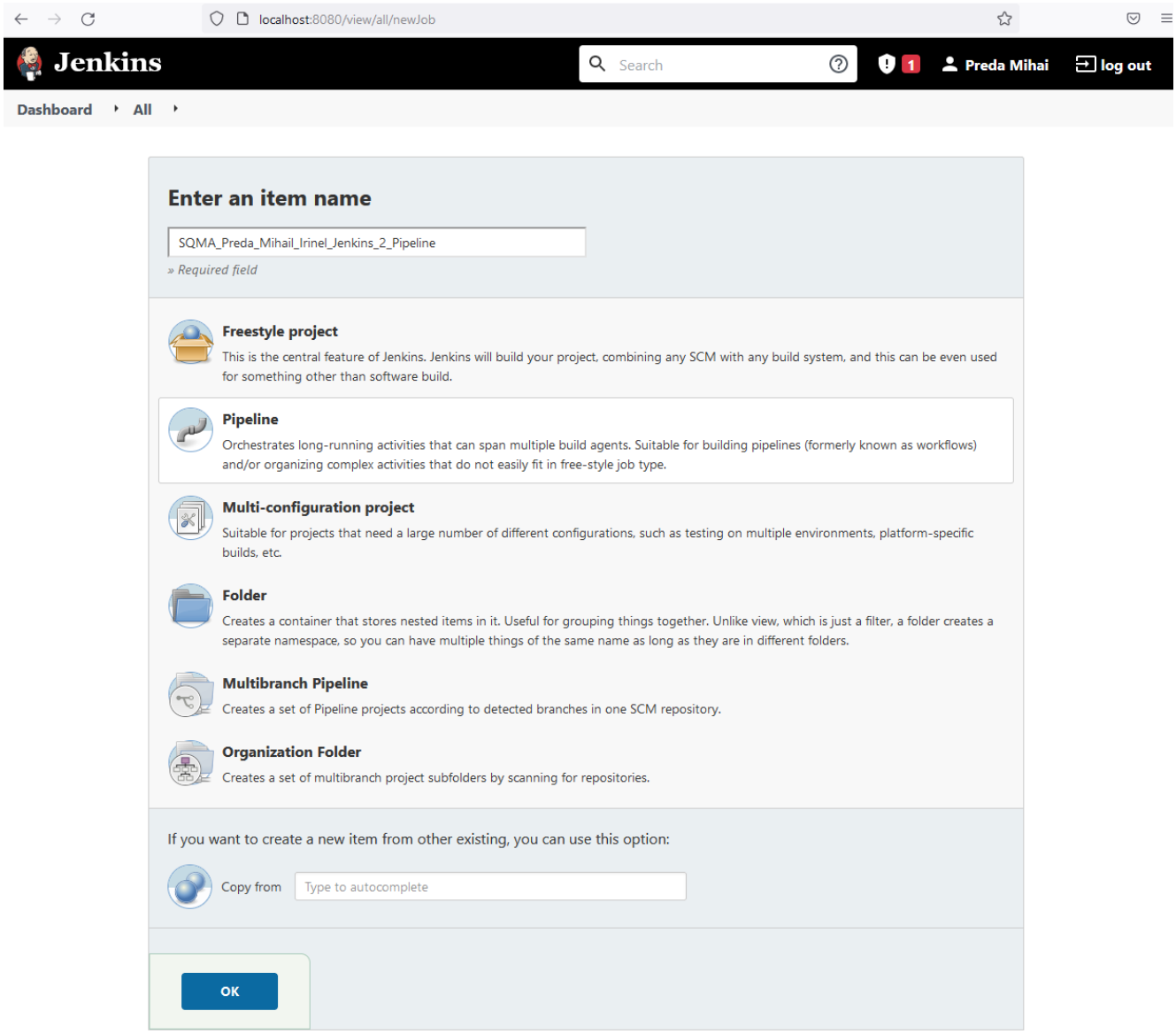

Fig.2

3. Add a description

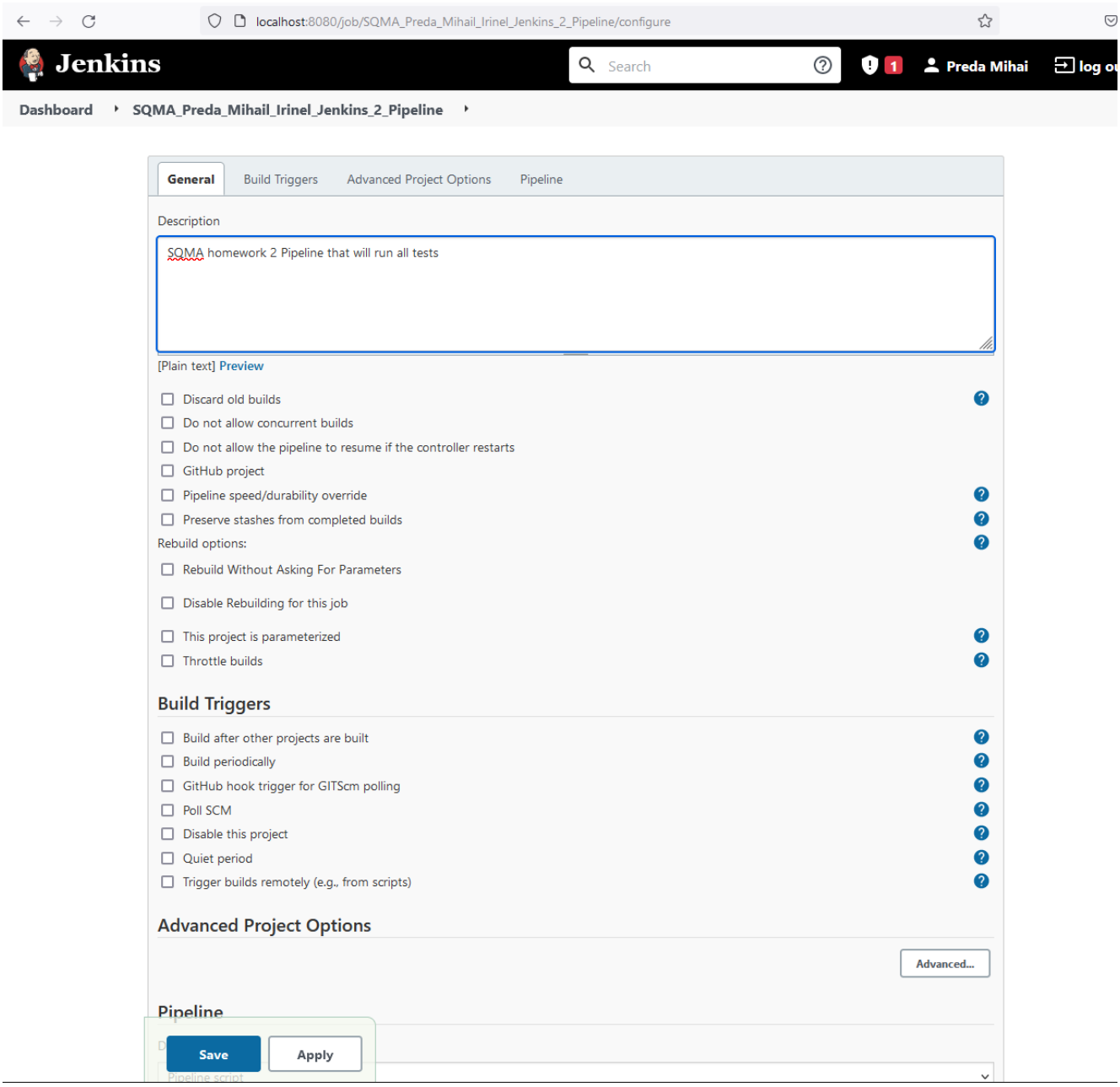

Fig.3

4. Scroll down to `Pipeline` and Add a `Pipeline script`. To do this click on `Pipeline Syntax`

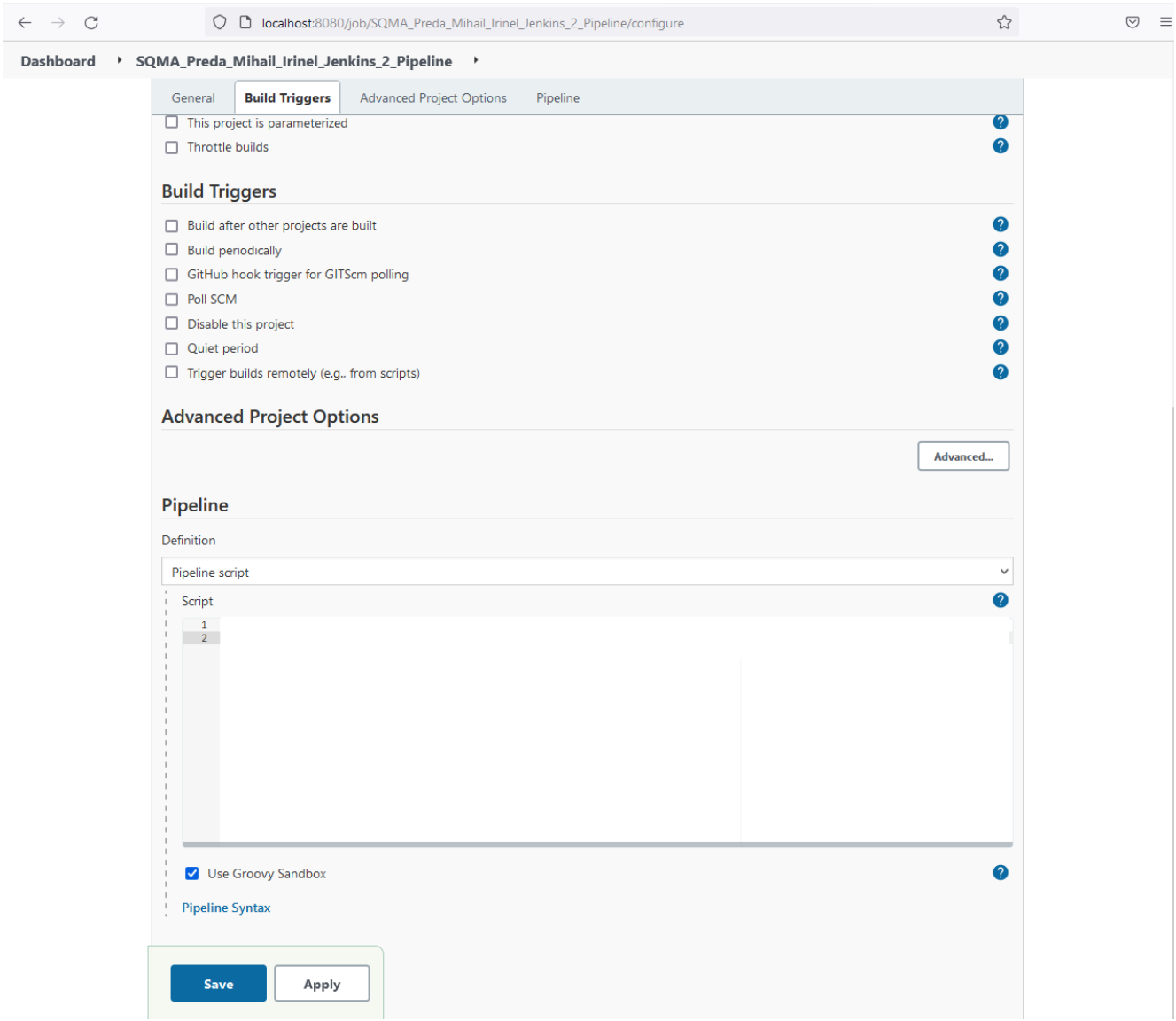

Fig.4

5. Now, select `Snippet Generator` from the left handside, then in the main page at `Sample step` select `build: Build a job`. After that, write the project to be build. Select with what parameter do you want the job and then click on `Generate Pipeline Script`

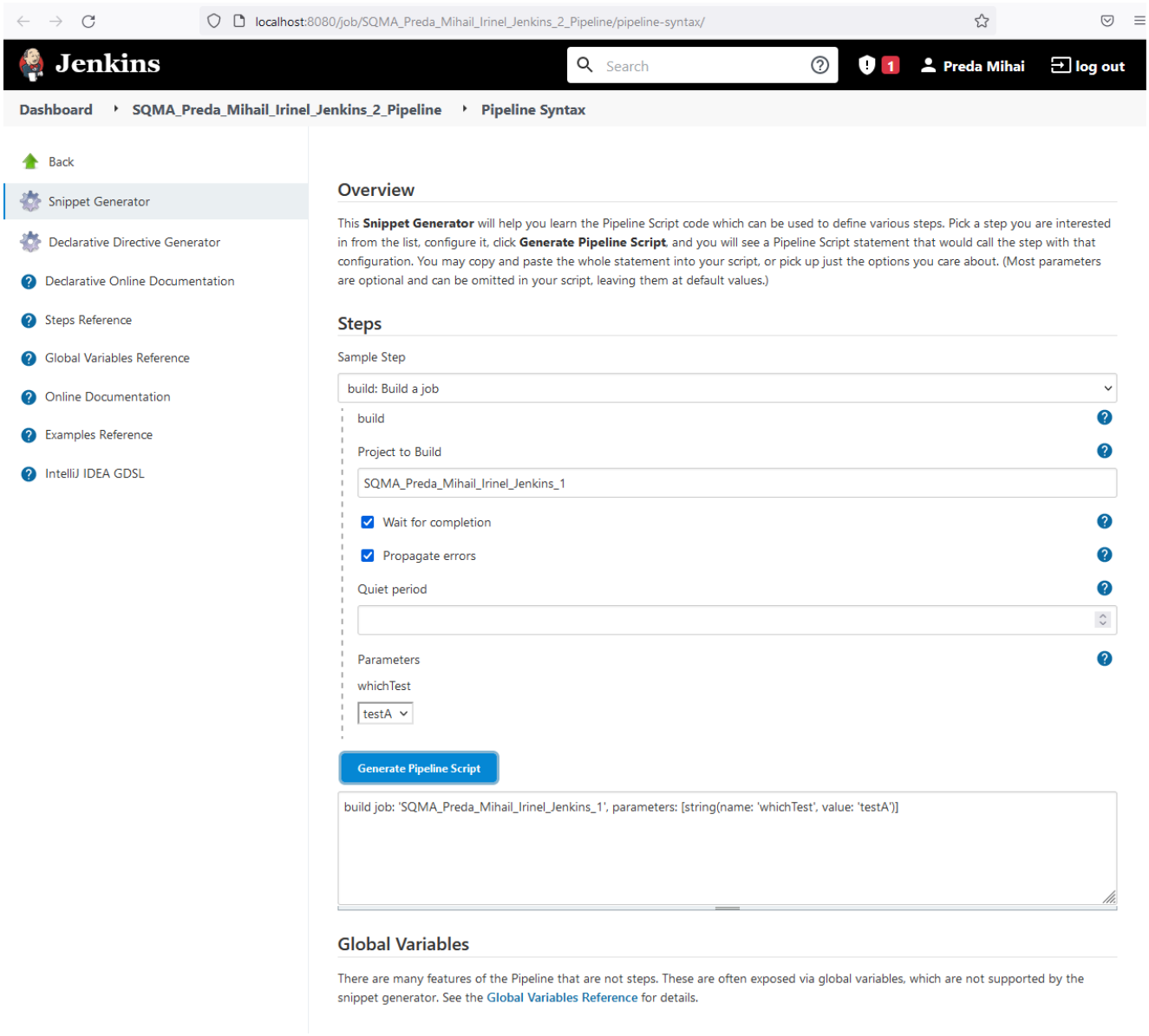
localhost:8080/job/SQMA_Preda_Mihail_Irinel_Jenkins_2_Pipeline/pipeline-syntax/
Jenkins
Search
Preda Mihai
log out
Dashboard
SQMA_Preda_Mihail_Irinel_Jenkins_2_Pipeline
Pipeline Syntax
Back
Snippet Generator
Declarative Directive Generator
Declarative Online Documentation
Steps Reference
Global Variables Reference
Online Documentation
Examples Reference
IntelliJ IDEA GDSL
Overview
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
build: Build a job
build
Project to Build
SQMA_Preda_Mihail_Irinel_Jenkins_1
Wait for completion
Propagate errors
Quiet period
Parameters
whichTest
testA
Generate Pipeline Script
build job: 'SQMA_Preda_Mihail_Irinel_Jenkins_1', parameters: [string(name: 'whichTest', value: 'testA')]
Global Variables
There are many features of the Pipeline that are not steps. These are often exposed via global variables, which are not supported by the snippet generator. See the Global Variables Reference for details.

Fig.5

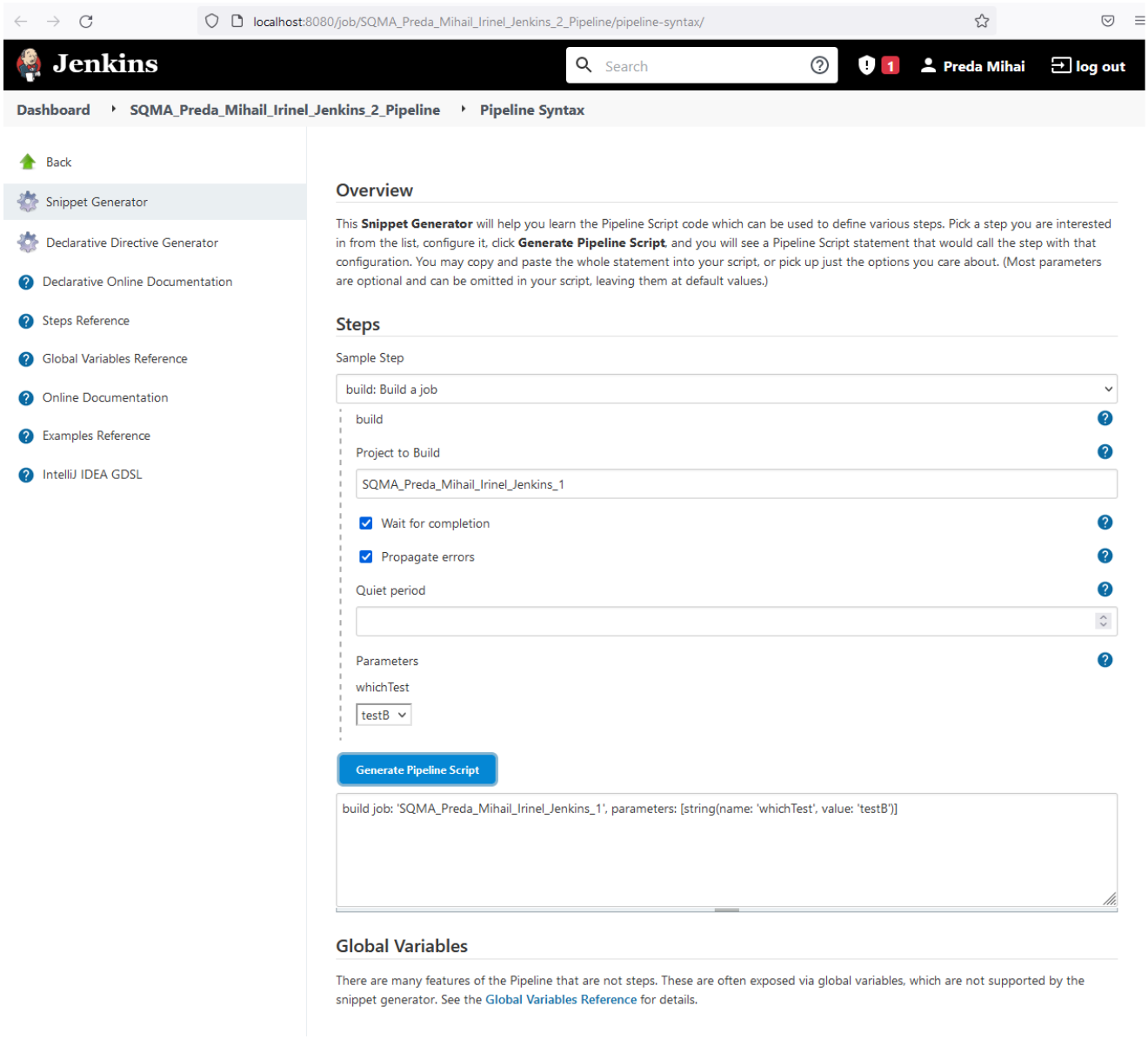

Fig.6

6. Copy the generated codes into the `Pipeline Script` and press `Save`.

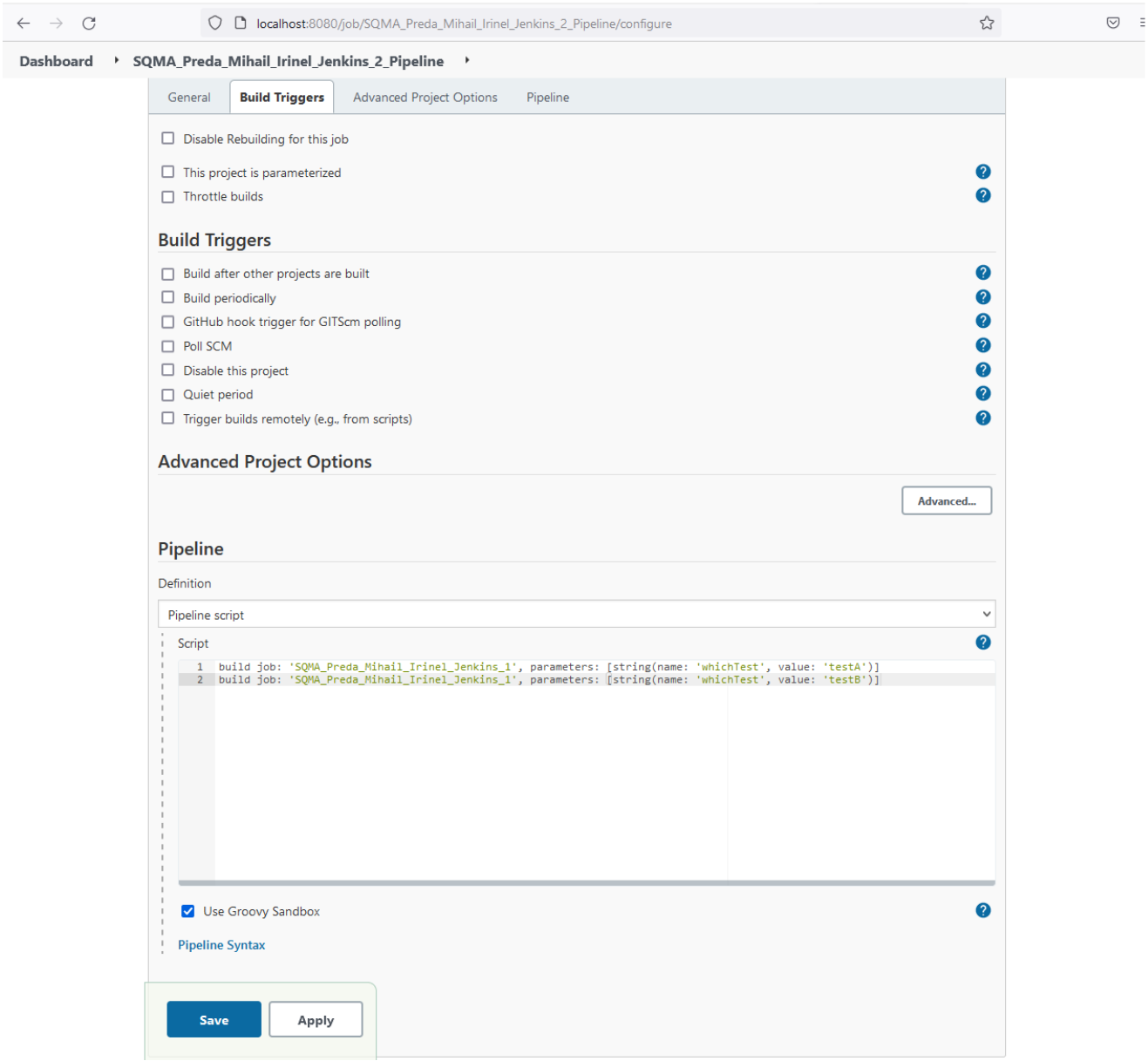

Fig.7

7. From left handside menu click on `Build Now`

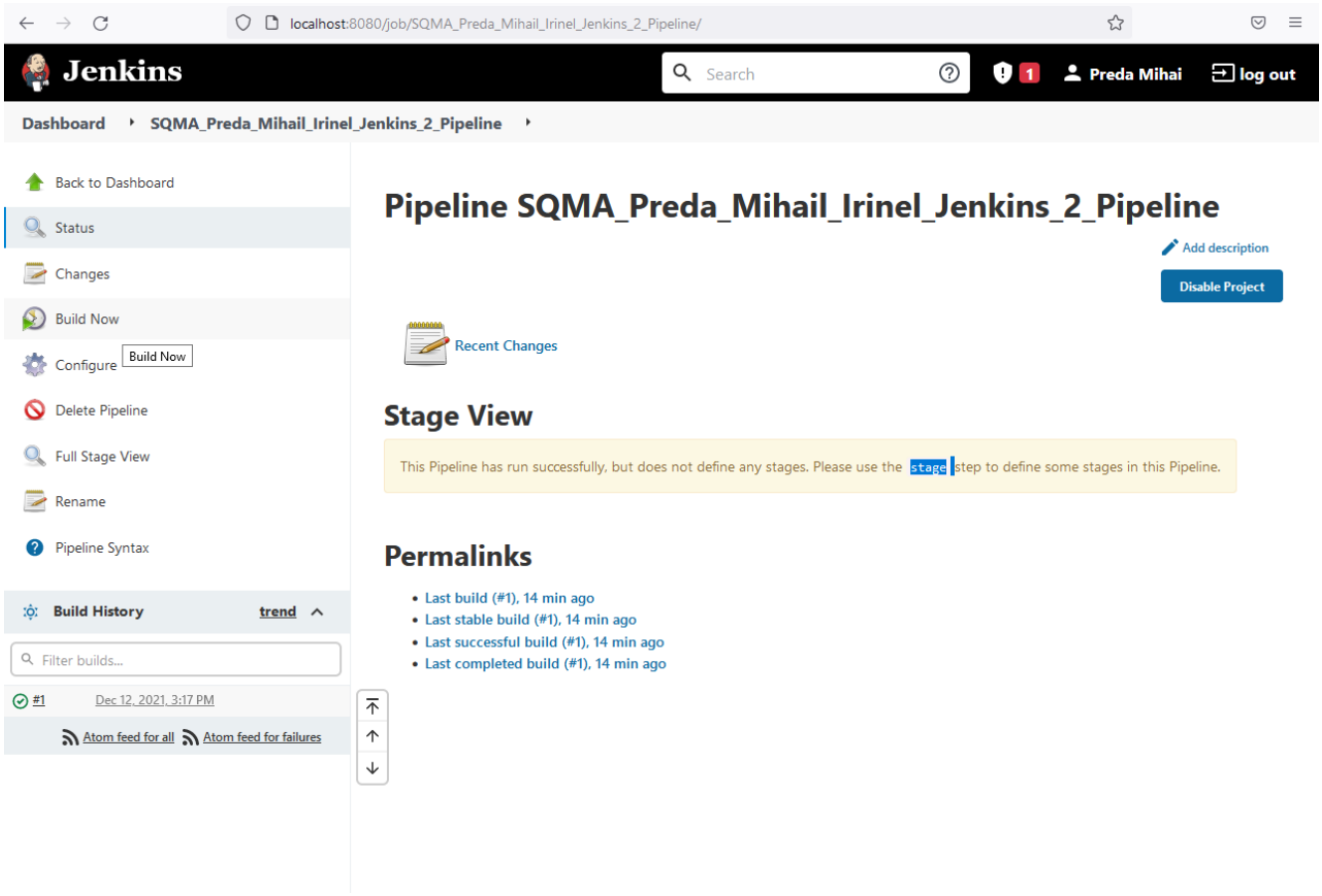

Fig.8

8. To see Pipeline result click on latest done build

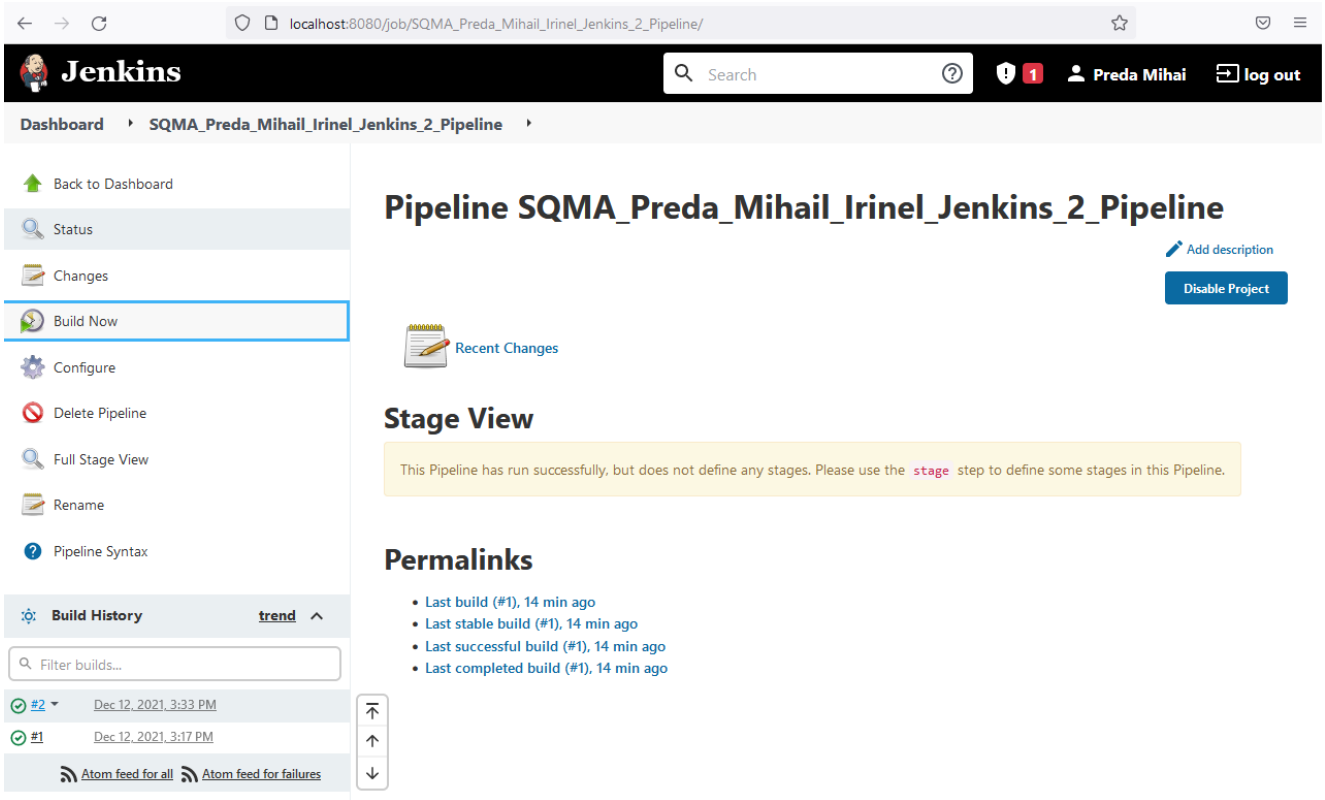

Fig.9

9. Then click on `Console Output` to see the build

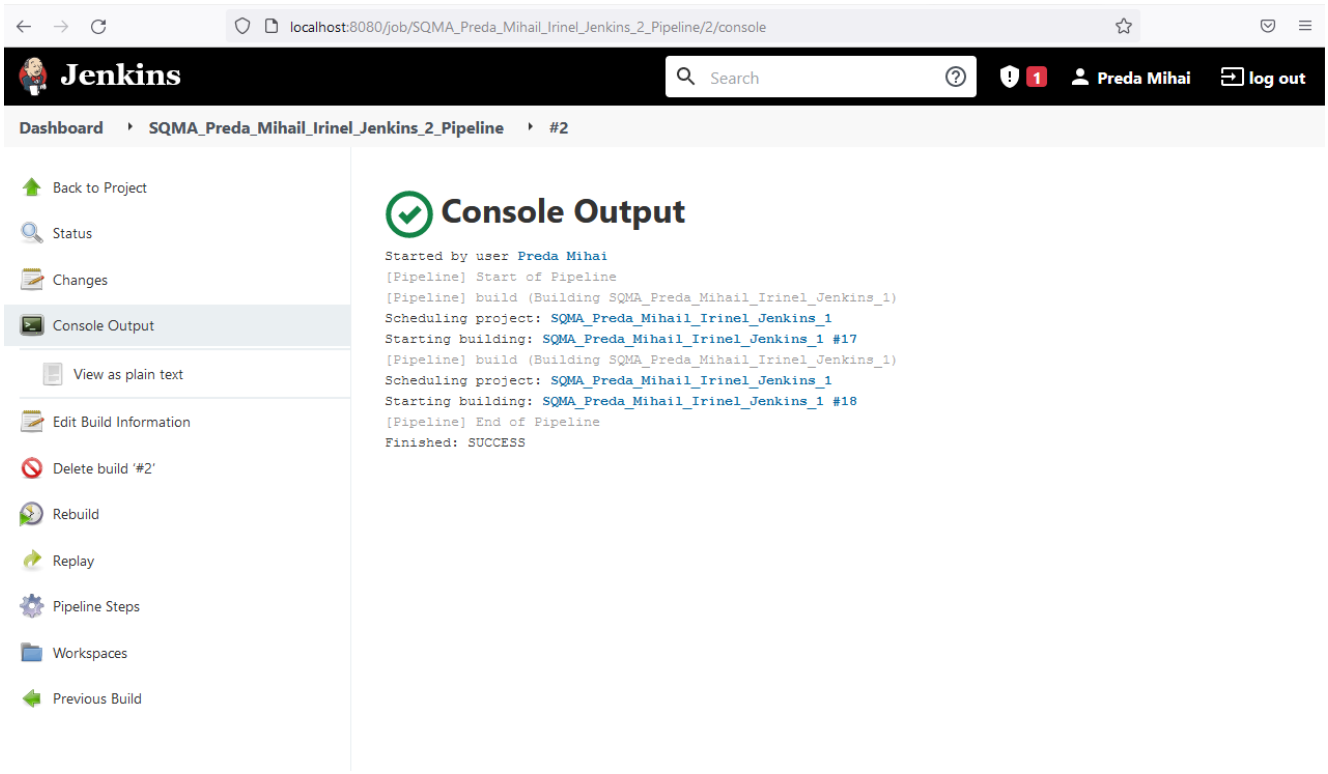

Fig.10

Project setup

```
npm install
```

Project run

```
npm start
```

Project test

```
npm test
```

Project individual test suites

```
npm run testA
npm run testB
```