

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:

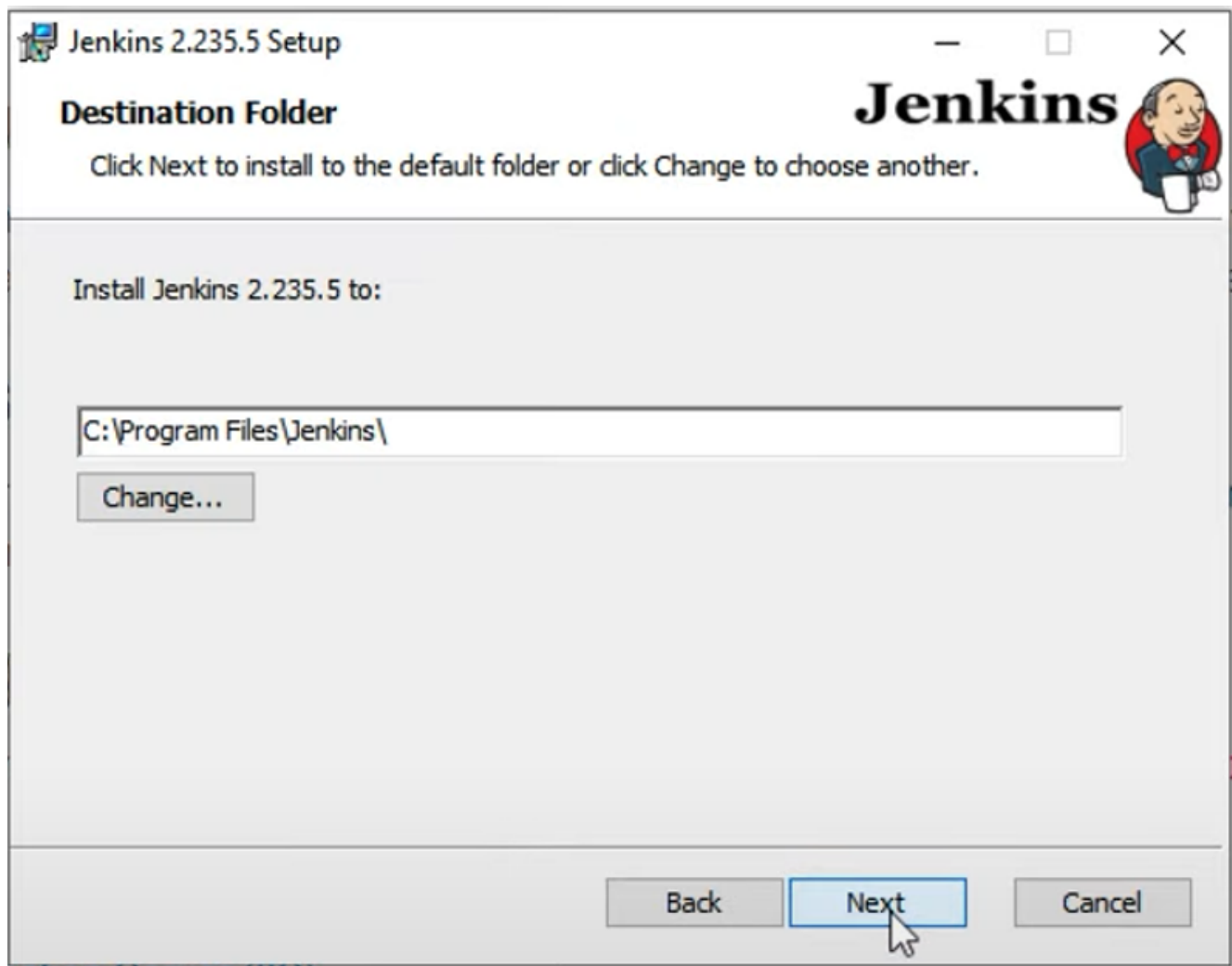
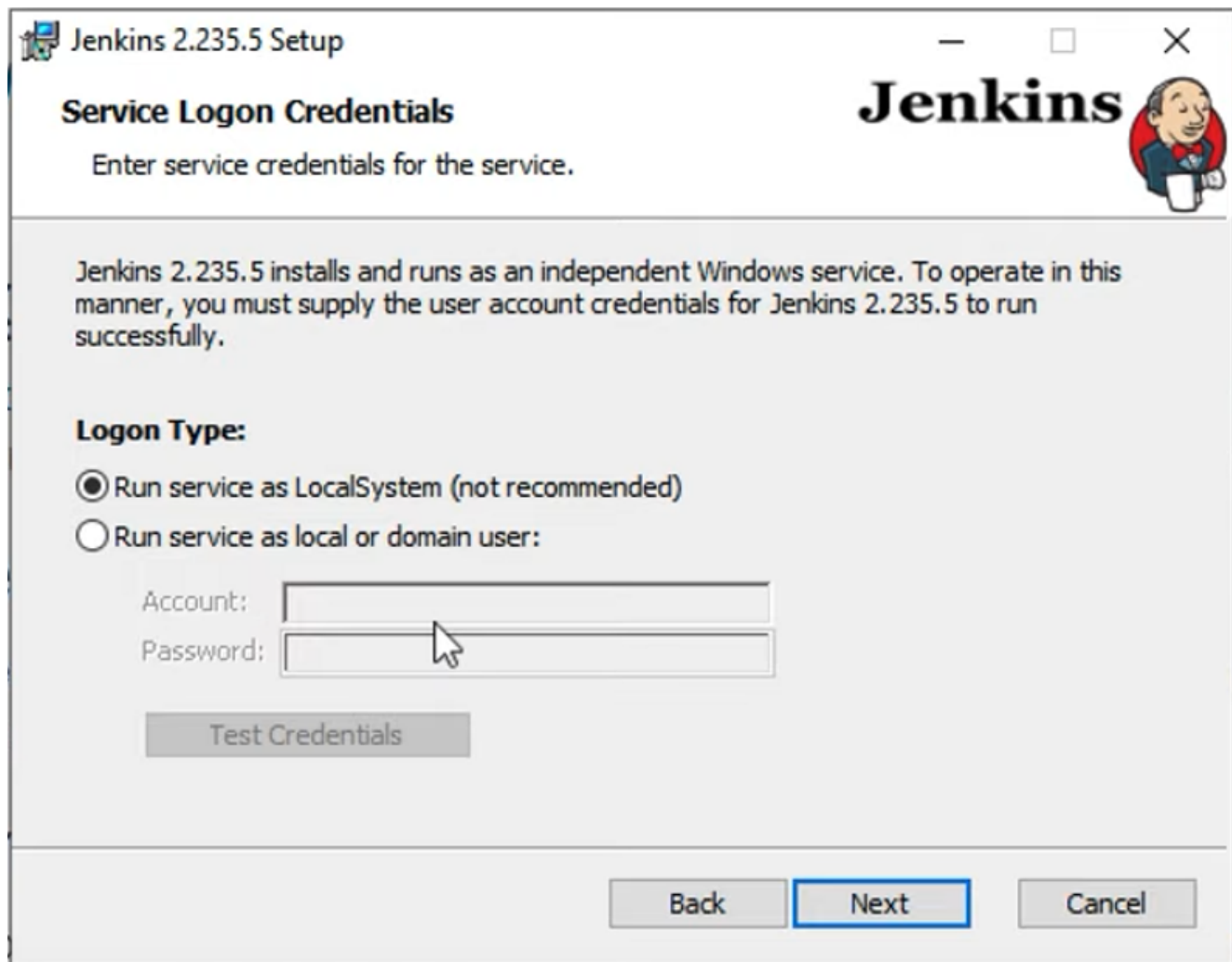
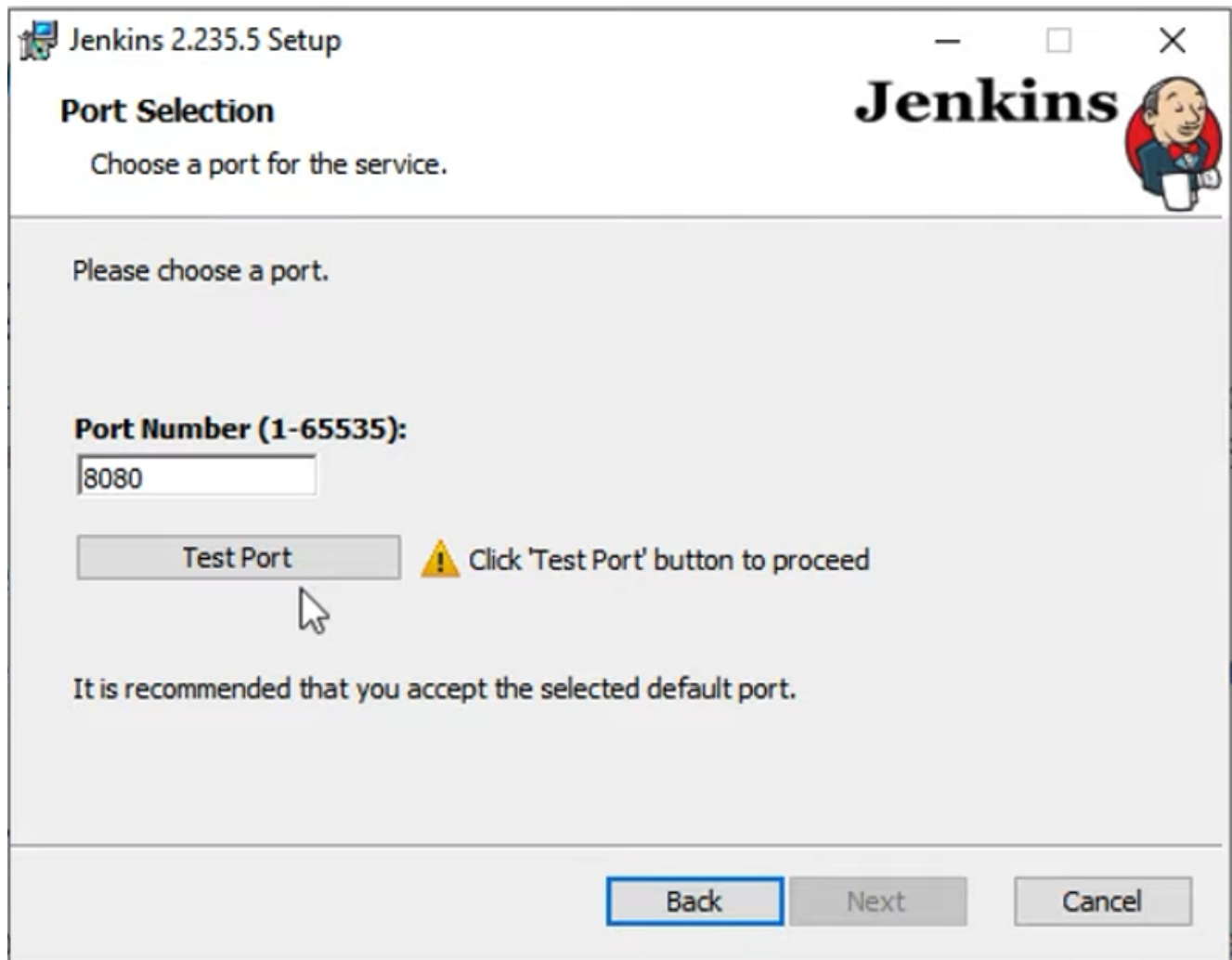


Fig.2

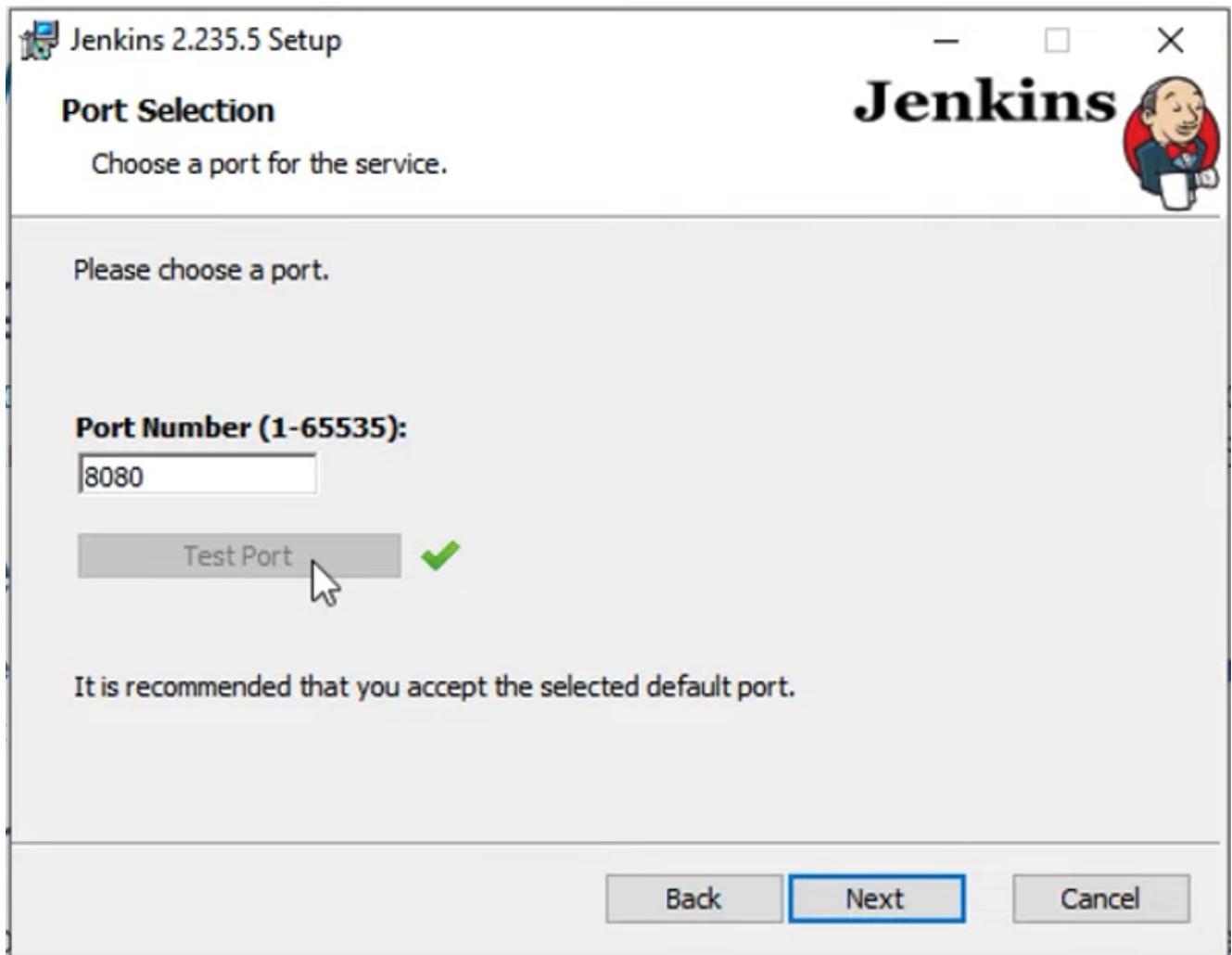
3. Choose Logon type. Select `Run service as LocalSystem`

**Fig.3**

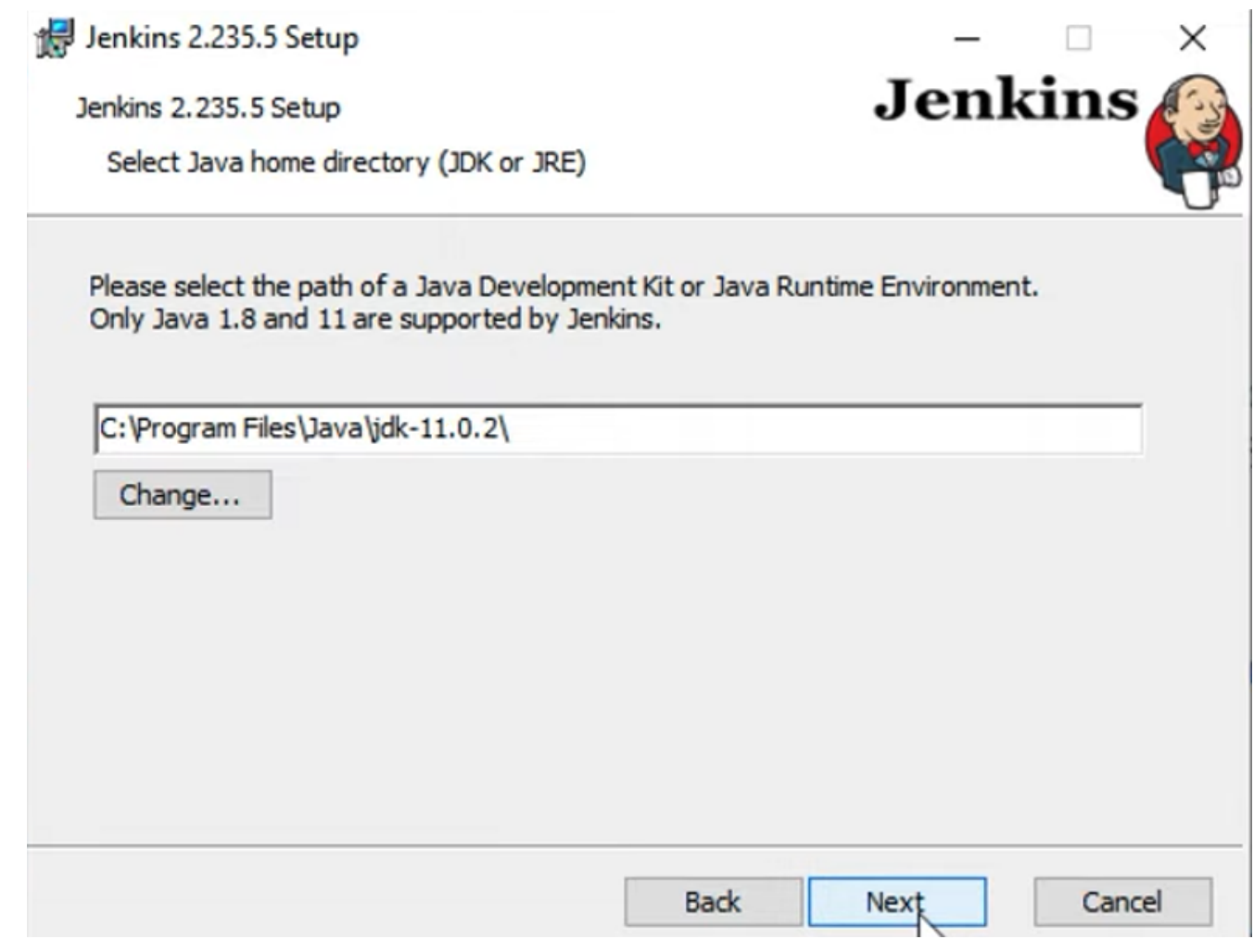
4. Select a port number or leave it as default (8080)

**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**

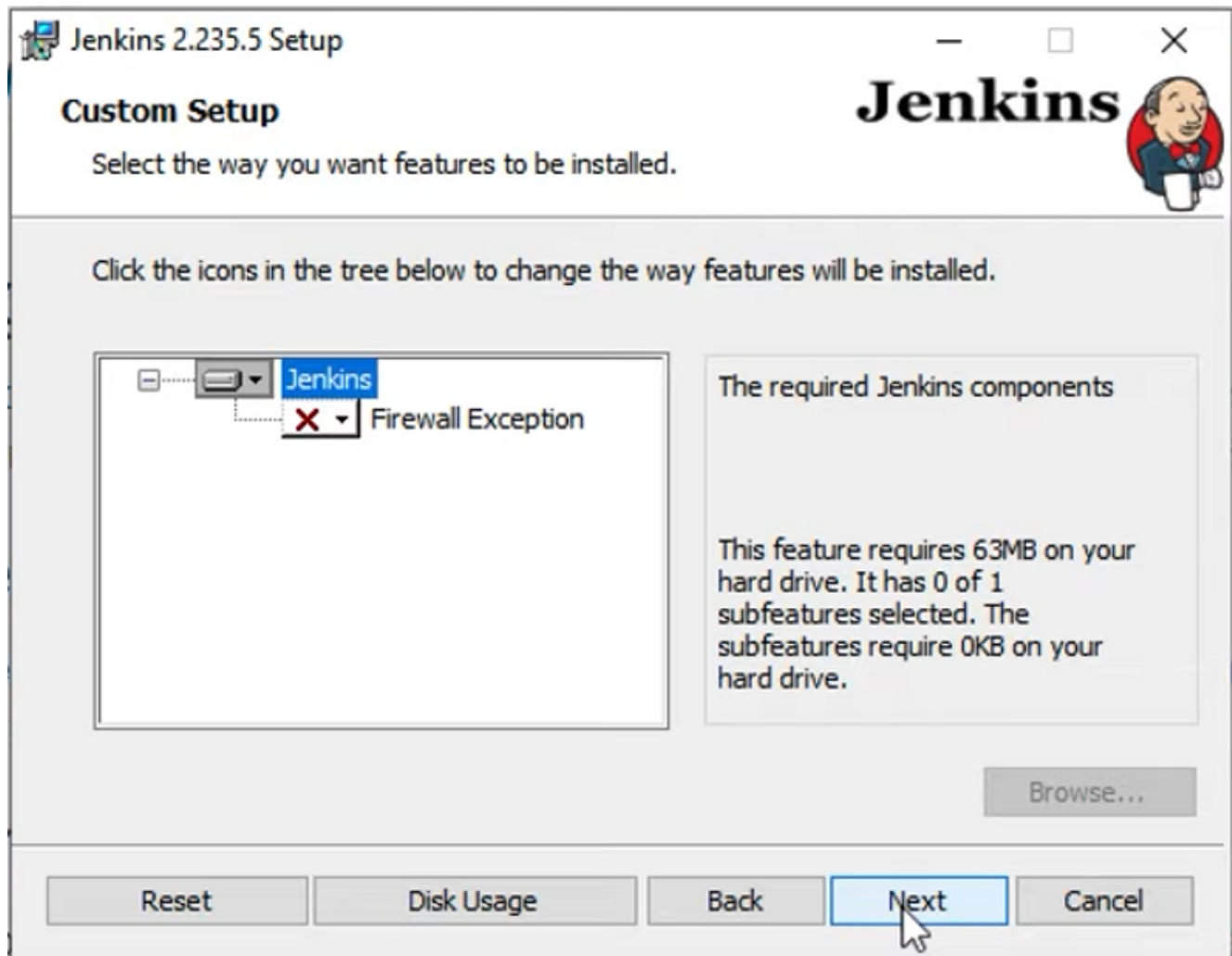
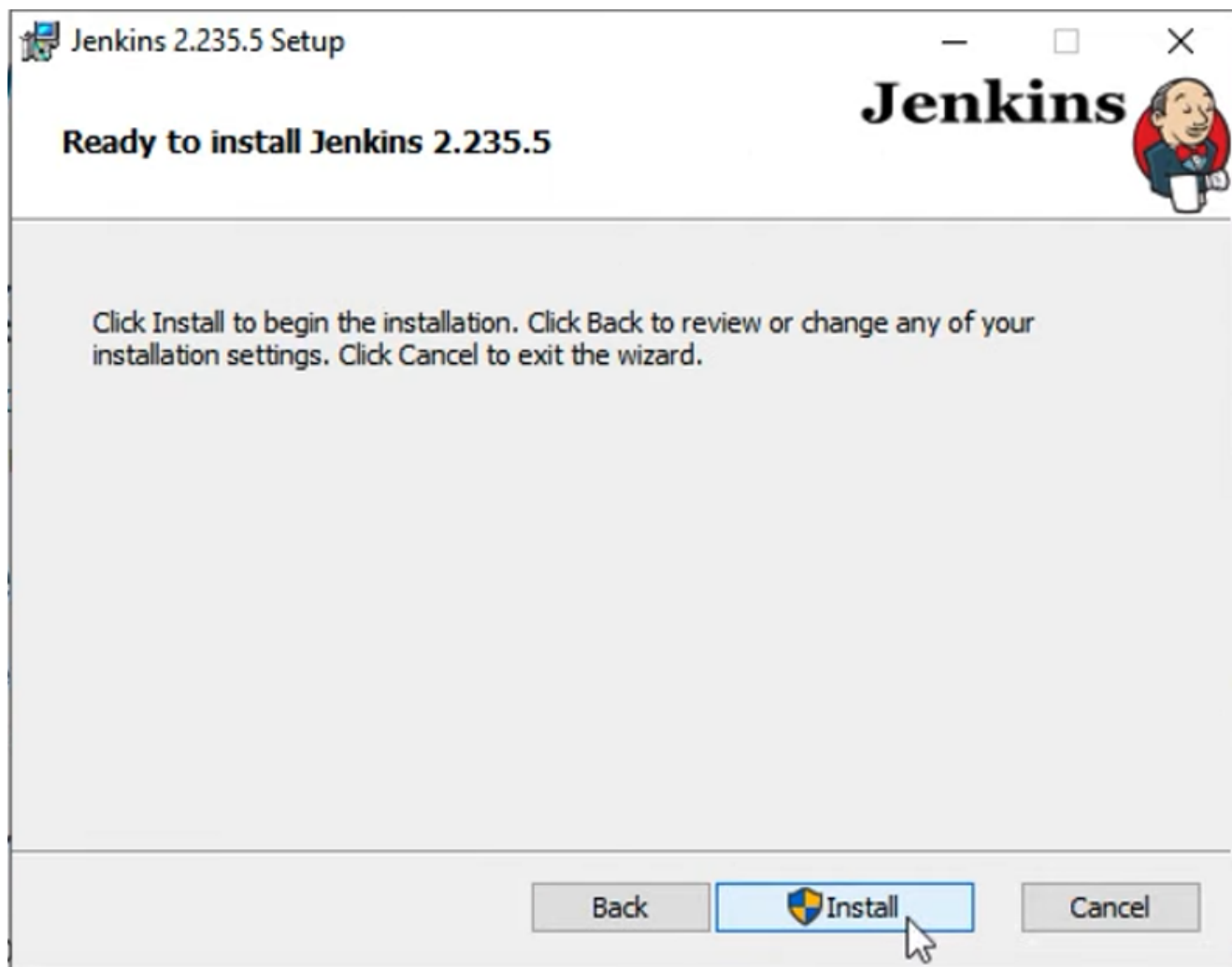


Fig.7

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/>

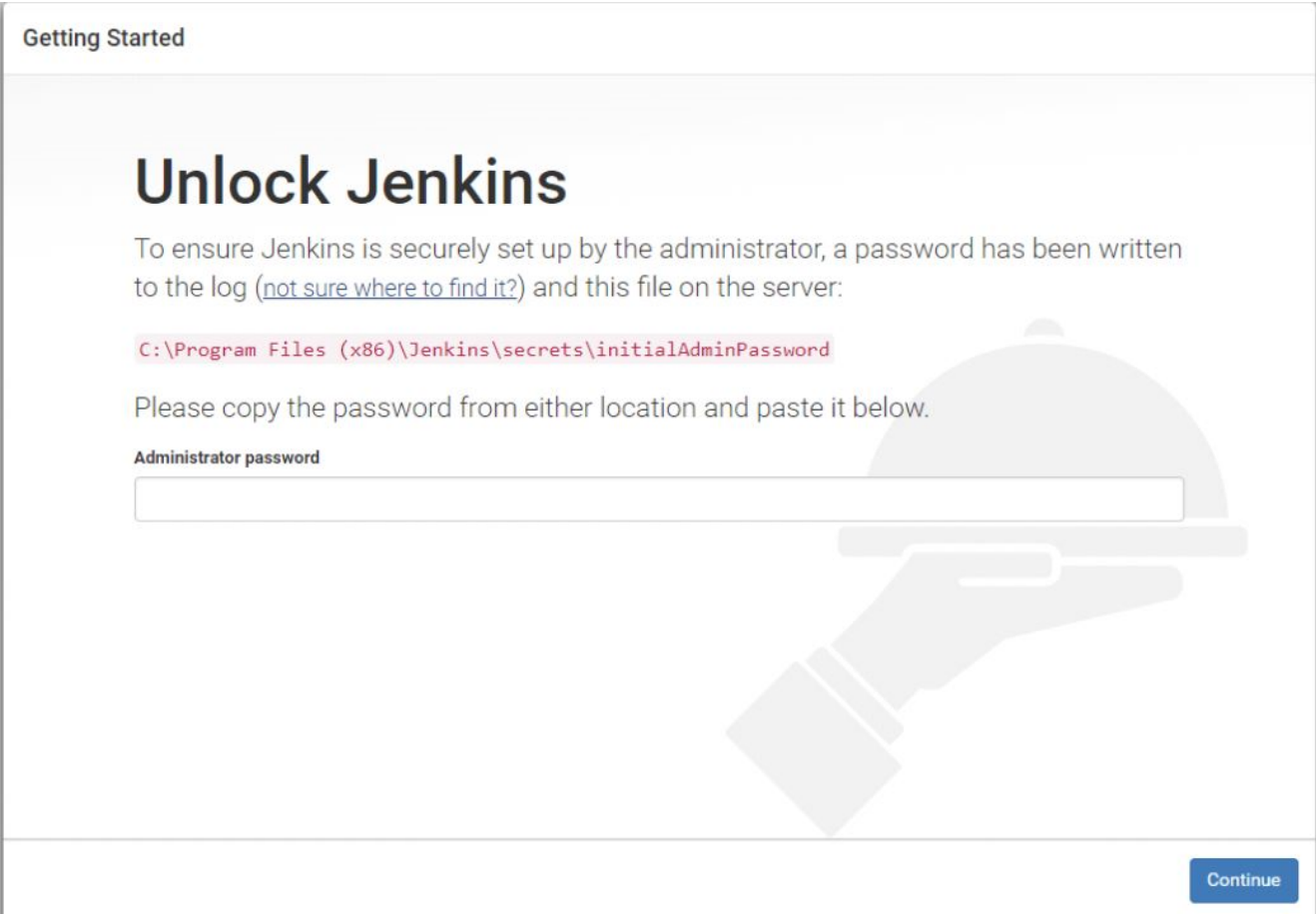


Fig.9

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

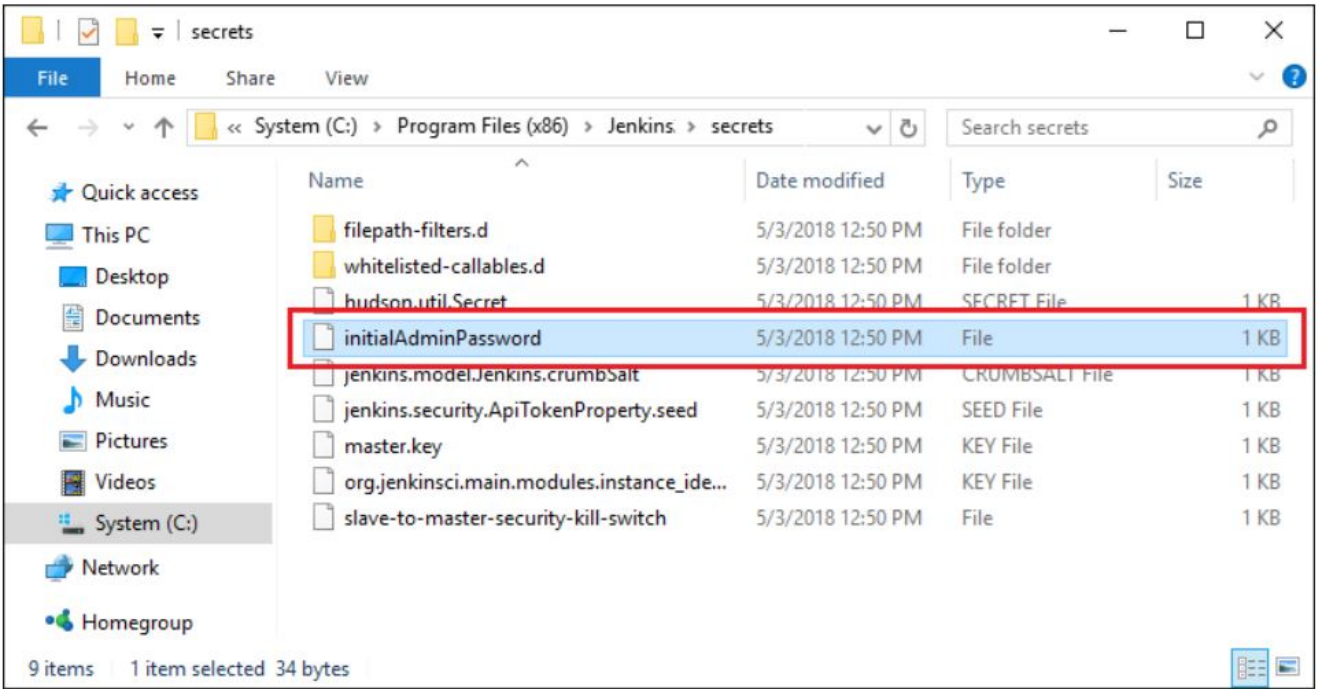


Fig.10

Copy and past it into the window dialog

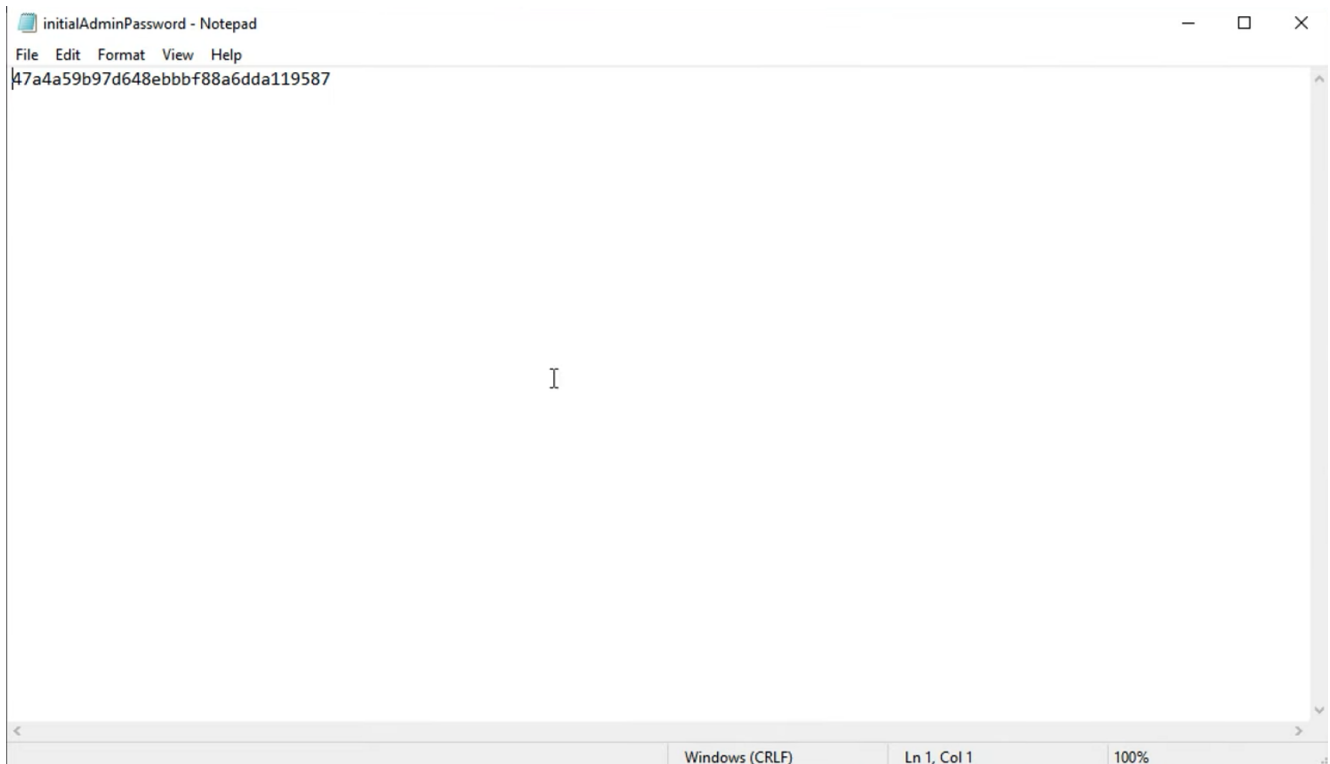


Fig.11

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

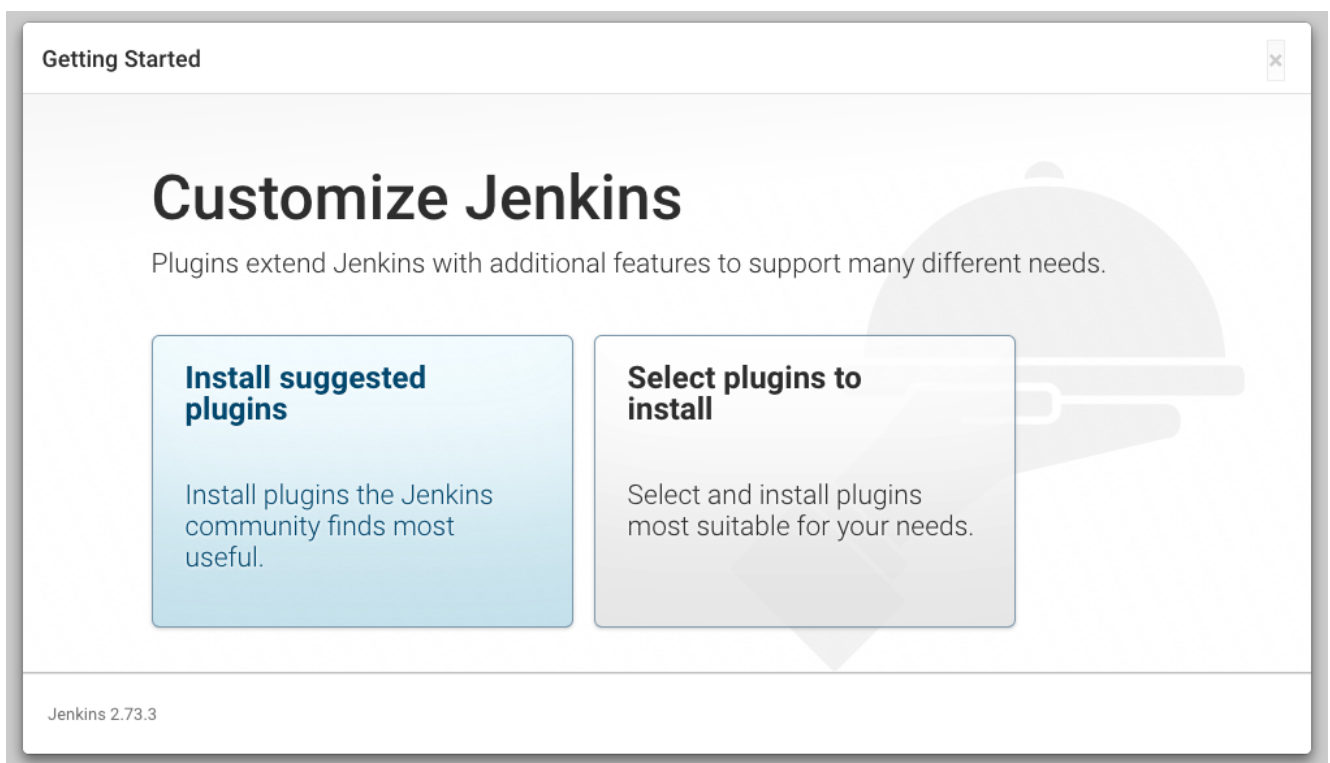


Fig.12

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



Getting Started

Create First Admin User

Username:

Password:

Confirm password:

Full name:

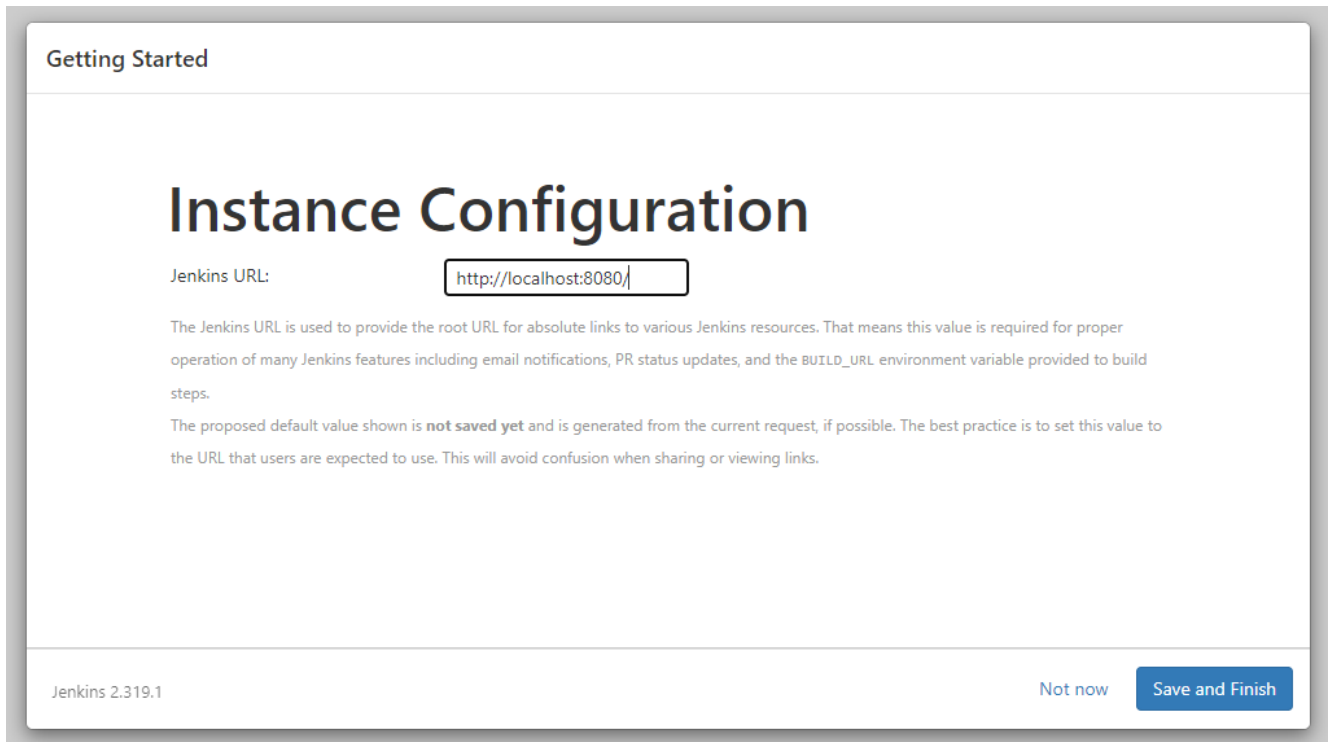
E-mail address:

Jenkins 2.319.1

Skip and continue as admin

Save and Continue

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



The screenshot shows the 'Getting Started' section of the Jenkins Instance Configuration page. The title 'Instance Configuration' is prominently displayed. Below it, the 'Jenkins URL' field is pre-filled with 'http://localhost:8080/'. A detailed explanation of the Jenkins URL's purpose is provided, along with a note about the default value being 'not saved yet'. At the bottom, there are two buttons: 'Not now' and 'Save and Finish'.

Getting Started

Instance Configuration

Jenkins URL:

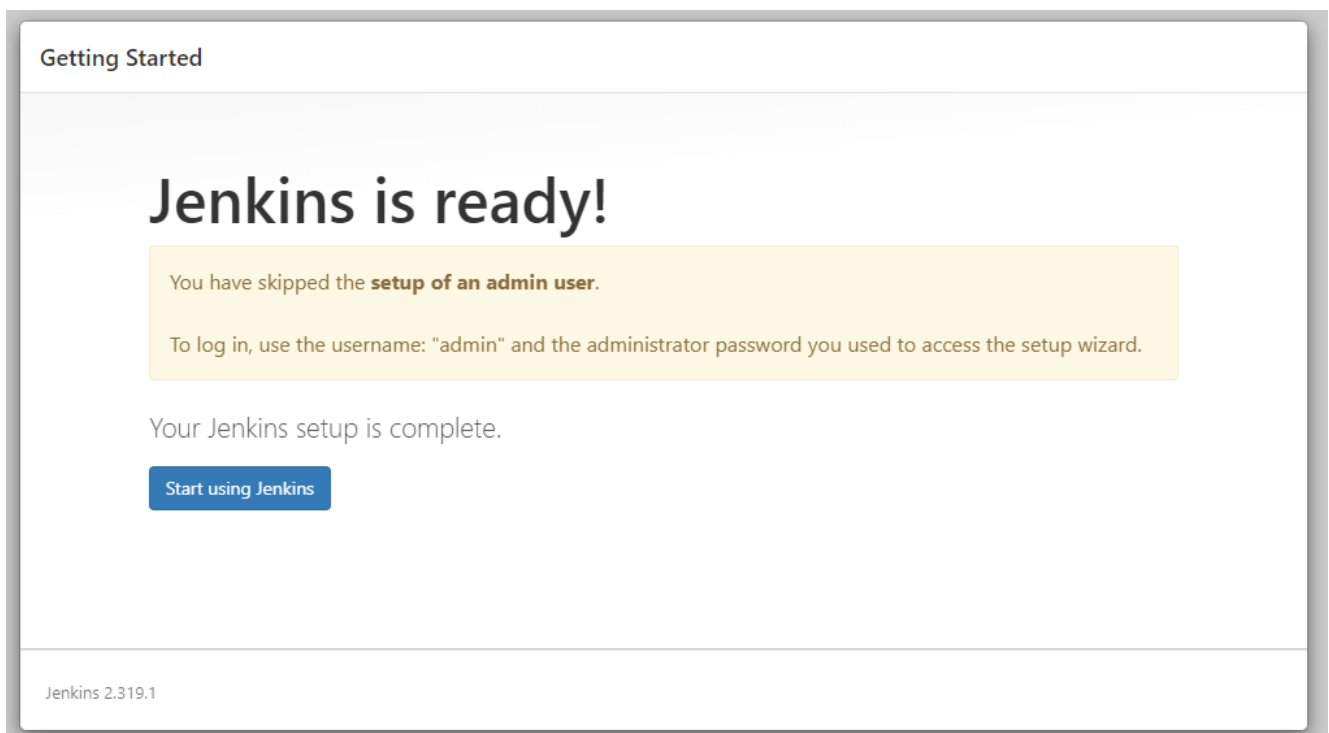
The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.319.1 Not now Save and Finish

Fig.15

15. Now Jenkins is ready



The screenshot shows the 'Getting Started' section of the Jenkins 'ready' page. The title 'Jenkins is ready!' is prominently displayed. Below it, a yellow box contains a message about skipping the admin user setup and instructions on how to log in. A blue button labeled 'Start using Jenkins' is visible. The footer shows 'Jenkins 2.319.1'.

Getting Started

Jenkins is ready!

You have skipped the **setup of an admin user**.

To log in, use the username: "admin" and the administrator password you used to access the setup wizard.

Your Jenkins setup is complete.

Start using Jenkins

Jenkins 2.319.1

Fig.16

Homework 1

Steps

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

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

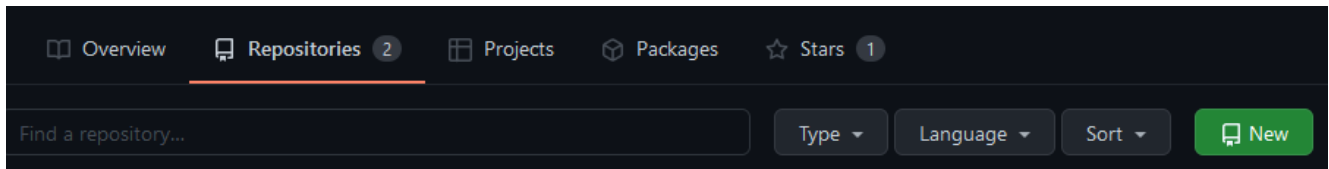


Fig.1

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

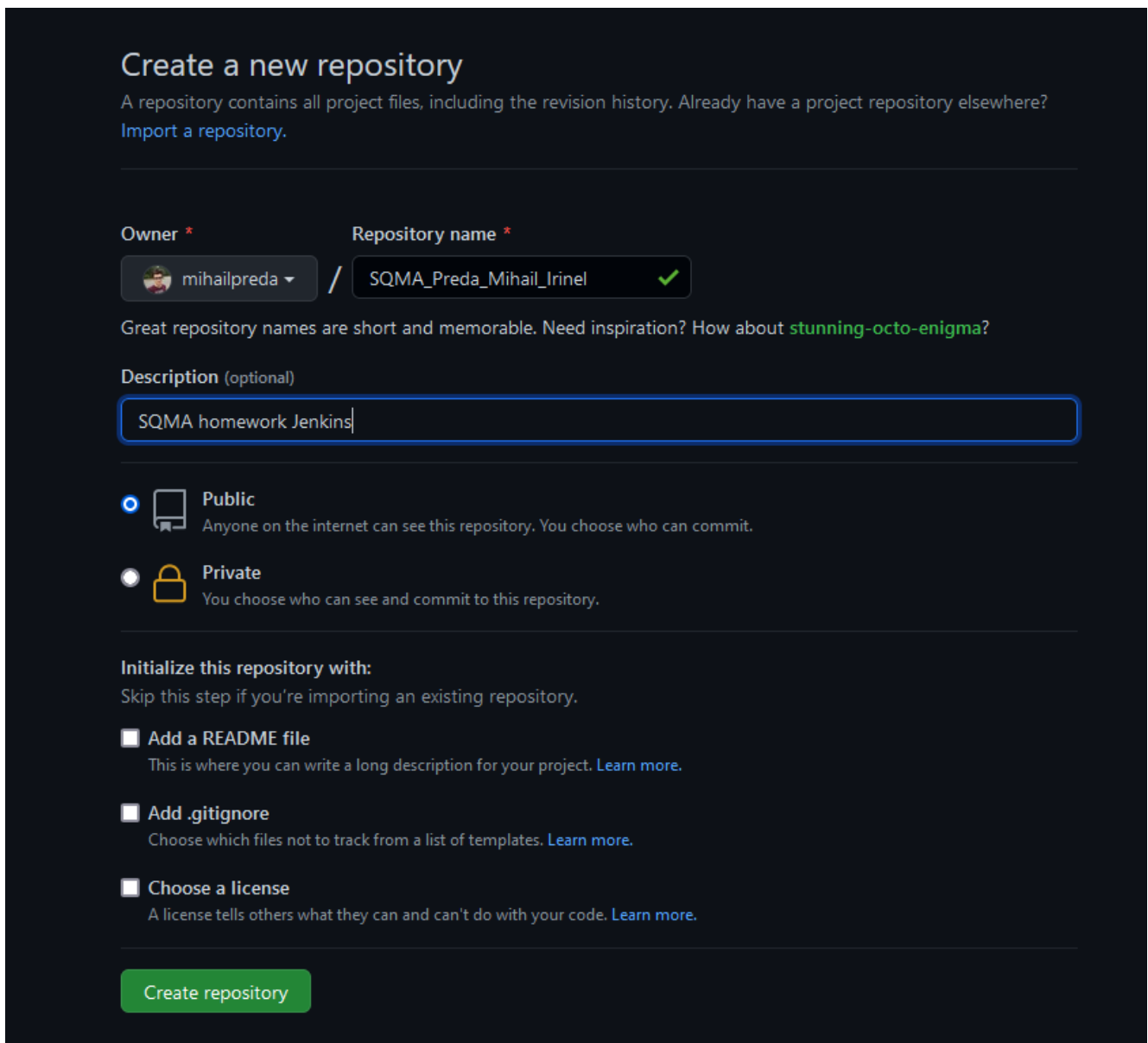


Fig.2

4. Set the origin to your repo

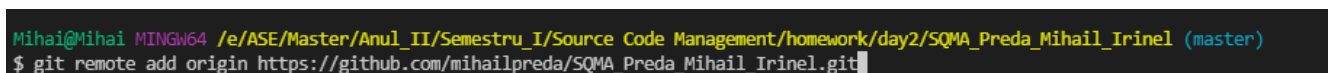


Fig.3

5. Add a new Job :

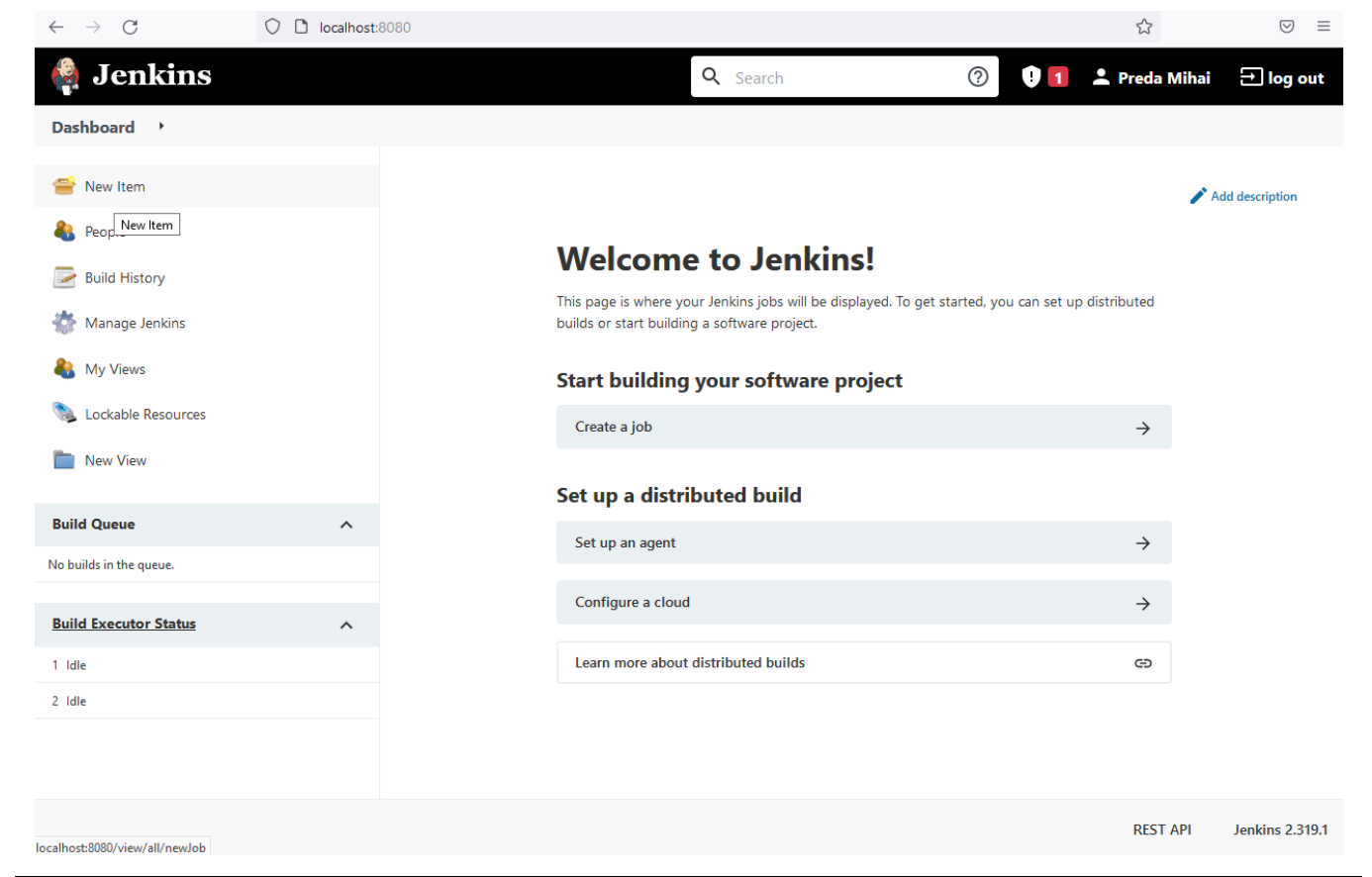


Fig.4

6. Enter Job Name







← → ↻ localhost:8080/view/all/newJob ☆

Dashboard > All >

Enter an item name

SQMA_Preda_Mihail_IrineI_Jenkins_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 than software build.
-  **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.
-  **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, so you can have multiple things of the same name as long as 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.

OK

Fig.5

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

The screenshot shows the Jenkins configuration interface for a job named 'SQMA_Preda_Mihail_Irinel_Jenkins_1'. The browser address bar indicates the URL is 'localhost:8080/job/SQMA_Preda_Mihail_Irinel_Jenkins_1/configure'. The page has a breadcrumb trail: 'Dashboard > SQMA_Preda_Mihail_Irinel_Jenkins_1 >'. Below this is a tabbed interface with tabs for 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build', and 'Post-build Actions'. The 'General' tab is active.

General Tab:

- Description:** A text area containing 'Jenkins Job 1 SQMA homework'.
- Options:** A list of checkboxes for job settings:
 - ☐ Discard old builds
 - ☐ GitHub project
 - Rebuild options:**
 - ☐ Rebuild Without Asking For Parameters
 - ☐ Disable Rebuilding for this job
 - ☐ This build requires lockable resources
 - ☐ This project is parameterized
 - ☐ Throttle builds
 - ☐ Disable this project
 - ☐ Execute concurrent builds if necessary

Source Code Management Tab:

- Repository:** A radio button selection with 'None' and 'Git' (selected).
- Repositories:** A section for adding repositories.
 - Repository URL:** A text field containing 'https://github.com/mihailpreda/SQMA_Preda_Mihail_Irinel.git'.
 - Credentials:** A dropdown menu showing '- none -' and an 'Add' button.
- Buttons:** 'Advanced...' and 'Add Repository' buttons.

Footer:

- Branches to build:** A section with a 'Save' button and an 'Apply' button.
- Errors:** A red 'X' icon and a question mark icon are visible in the bottom right corner.

Fig.6

8. If everything is okay you will see the workspace that the Jenkins successfully downloaded the files from Github.

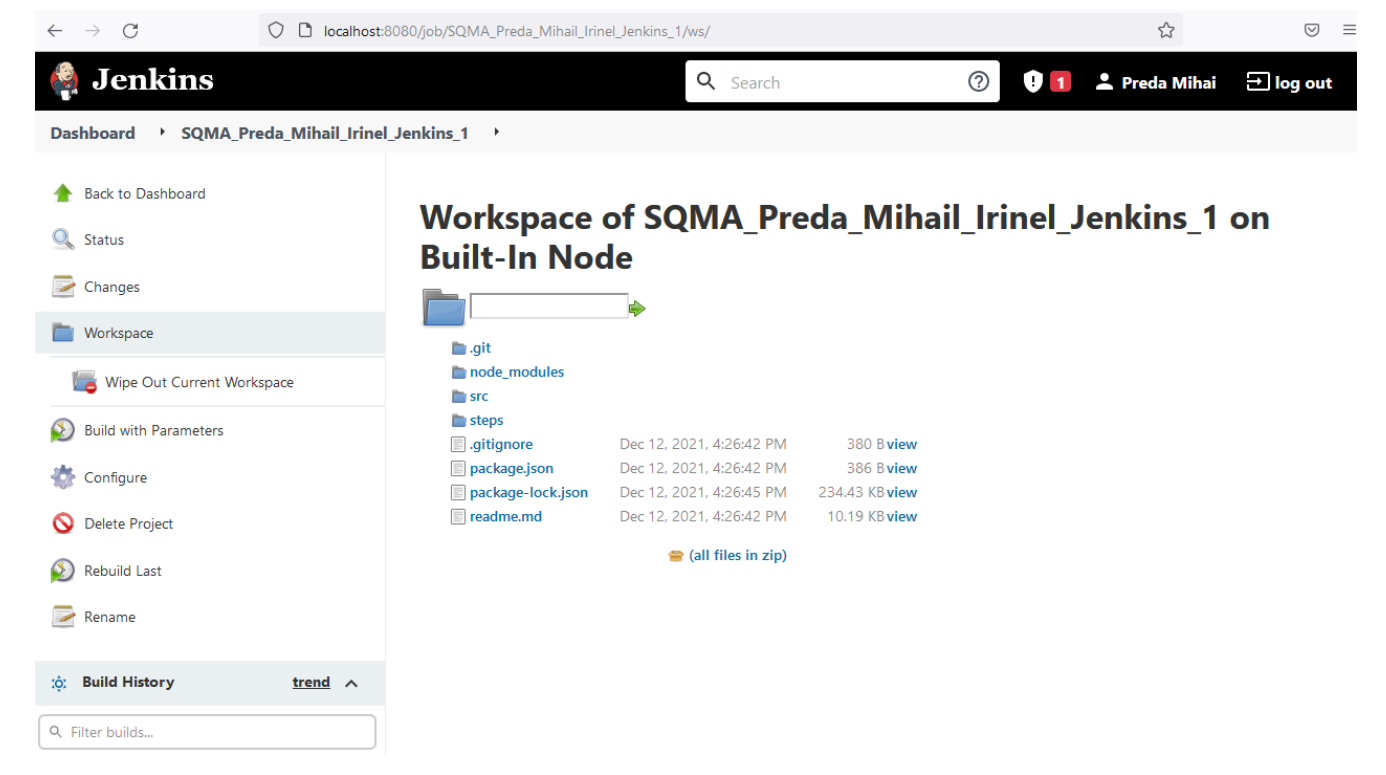


Fig.7

9. Then go to **Configure**

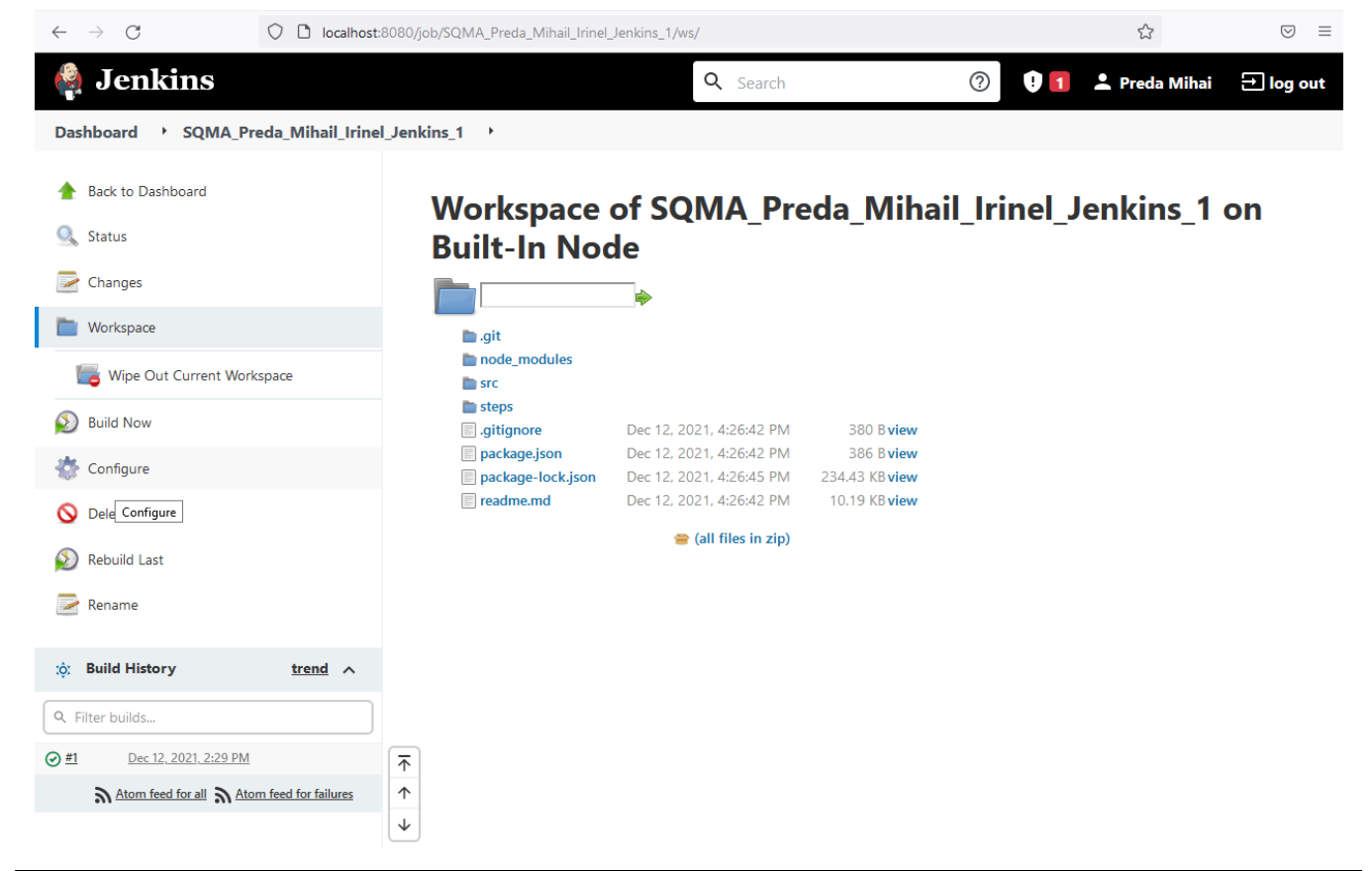


Fig.8

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`

localhost:8080/job/SQMA_Preda_Mihail_Irinel_Jenkins_1/configure

Dashboard ▸ SQMA_Preda_Mihail_Irinel_Jenkins_1 ▸

General Source Code Management Build Triggers Build Environment Build Post-build Actions

☐ Discard old builds

☐ GitHub project

Rebuild options:

☐ Rebuild Without Asking For Parameters

☐ Disable Rebuilding for this job

☐ This build requires lockable resources

☒ This project is parameterized

Choice Parameter

Name

Choices

Description

[Plain text] [Preview](#)

[Add Parameter](#)

☐ Throttle builds

☐ Disable this project

☐ Execute concurrent builds if necessary

[Advanced...](#)

[Save](#) [Apply](#)

Fig.9

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

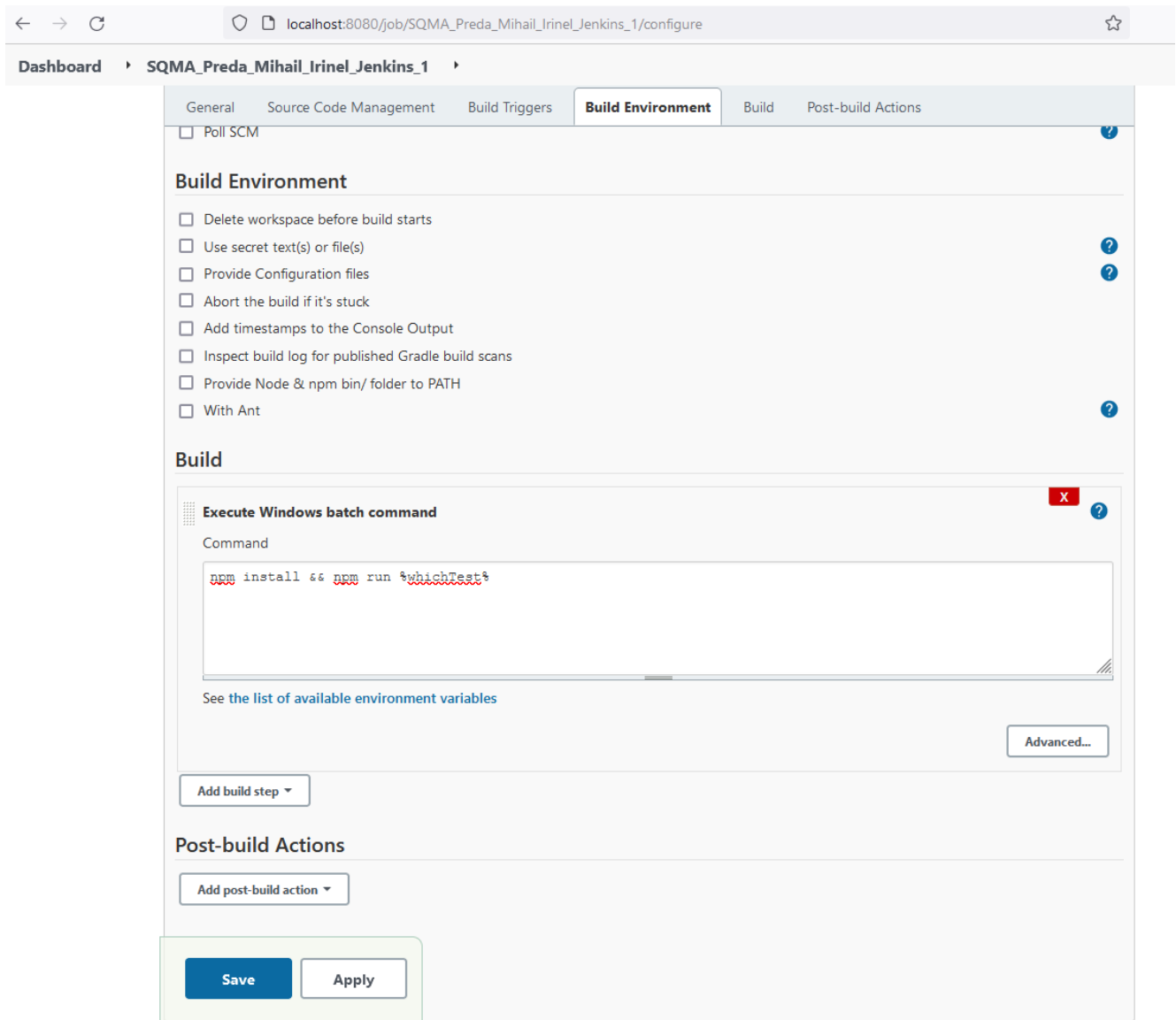


Fig.10

12. Click on **Build with parameters**

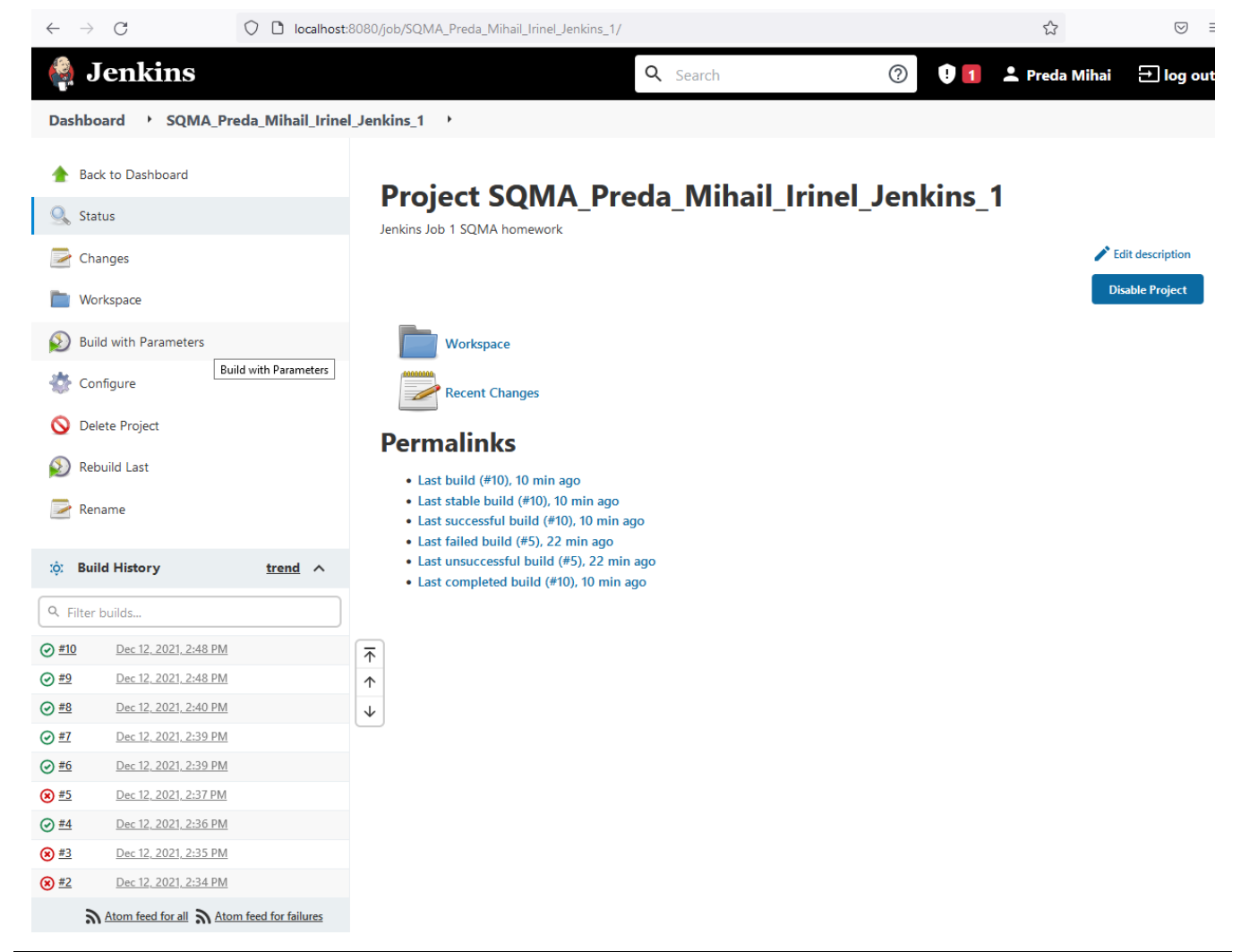


Fig.11

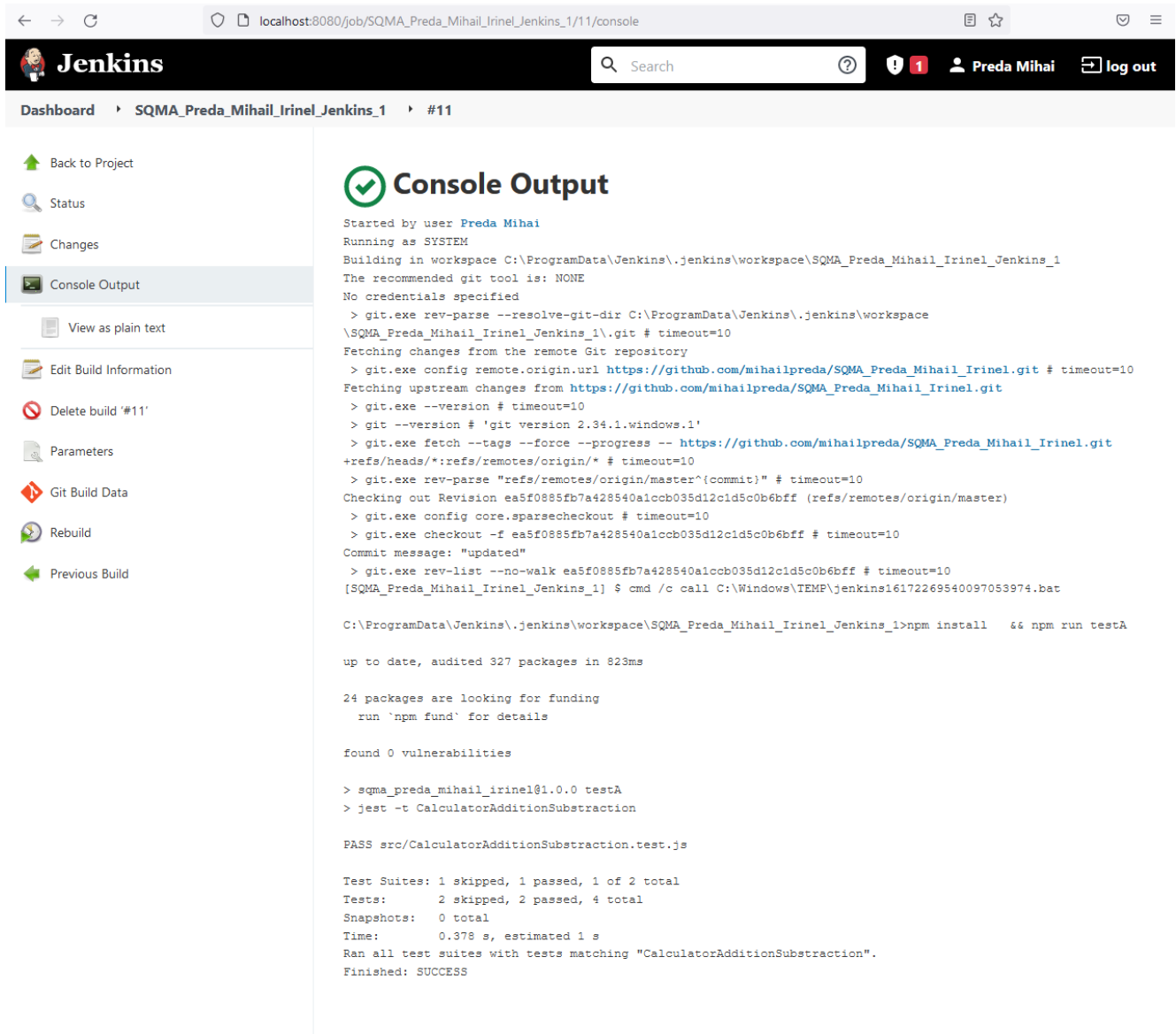
13. Choose `testA`, then press `Build`.

The screenshot shows the Jenkins web interface at `localhost:8080/job/SQMA_Preda_Mihail_Irinel_Jenkins_1/build?delay=0sec`. The top navigation bar includes the Jenkins logo, a search bar, a notification icon with a red '1', the user name 'Preda Mihai', and a 'log out' button. The breadcrumb trail is 'Dashboard > SQMA_Preda_Mihail_Irinel_Jenkins_1 >'. On the left sidebar, there are links for 'Back to Dashboard', 'Status', 'Changes', 'Workspace', 'Build with Parameters', 'Configure', 'Delete Project', 'Rebuild Last', and 'Rename'. The main content area is titled 'Project SQMA_Preda_Mihail_Irinel_Jenkins_1' and states 'This build requires parameters:'. Below this, there is a parameter 'whichTest' with a dropdown menu showing 'testA' and a blue 'Build' button. At the bottom left, the 'Build History' section is visible, showing a list of builds with their status (success or failure) and timestamps. The latest build (#10) is successful and completed at 2:48 PM on Dec 12, 2021.

Build Number	Status	Timestamp
#10	Success	Dec 12, 2021, 2:48 PM
#9	Success	Dec 12, 2021, 2:48 PM
#8	Success	Dec 12, 2021, 2:40 PM
#7	Success	Dec 12, 2021, 2:39 PM
#6	Success	Dec 12, 2021, 2:39 PM
#5	Failure	Dec 12, 2021, 2:37 PM
#4	Success	Dec 12, 2021, 2:36 PM
#3	Failure	Dec 12, 2021, 2:35 PM
#2	Failure	Dec 12, 2021, 2:34 PM

Fig.12

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



The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and the user 'Preda Mihai' with a 'log out' button. The breadcrumb trail is 'Dashboard > SQMA_Preda_Mihail_Irinel_Jenkins_1 > #11'. The left sidebar contains links to 'Back to Project', 'Status', 'Changes', 'Console Output' (which is selected), 'View as plain text', 'Edit Build Information', 'Delete build #11', 'Parameters', 'Git Build Data', 'Rebuild', and 'Previous Build'.

The main content area is titled 'Console Output' with a green checkmark icon. It displays the following text:

```

Started by user Preda Mihai
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\SQMA_Preda_Mihail_Irinel_Jenkins_1
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\SQMA_Preda_Mihail_Irinel_Jenkins_1\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/mihailpreda/SQMA_Preda_Mihail_Irinel.git # timeout=10
Fetching upstream changes from https://github.com/mihailpreda/SQMA_Preda_Mihail_Irinel.git
> git.exe --version # timeout=10
> git --version # 'git version 2.34.1.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/mihailpreda/SQMA_Preda_Mihail_Irinel.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision ea5f0885fb7a428540a1ccb035d12c1d5c0b6b6ff (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f ea5f0885fb7a428540a1ccb035d12c1d5c0b6b6ff # timeout=10
Commit message: "updated"
> git.exe rev-list --no-walk ea5f0885fb7a428540a1ccb035d12c1d5c0b6b6ff # timeout=10
[SQMA_Preda_Mihail_Irinel_Jenkins_1] $ cmd /c call C:\Windows\TEMP\jenkins16172269540097053974.bat

C:\ProgramData\Jenkins\.jenkins\workspace\SQMA_Preda_Mihail_Irinel_Jenkins_1>npm install  && npm run testA

up to date, audited 327 packages in 823ms

24 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities

> sqma_preda_mihail_irinel@1.0.0 testA
> jest -t CalculatorAdditionSubstraction

PASS src/CalculatorAdditionSubstraction.test.js

Test Suites: 1 skipped, 1 passed, 1 of 2 total
Tests:      2 skipped, 2 passed, 4 total
Snapshots:  0 total
Time:       0.378 s, estimated 1 s
Ran all test suites with tests matching "CalculatorAdditionSubstraction".
Finished: SUCCESS

```

Fig.13

15. Click on **Build with parameters**

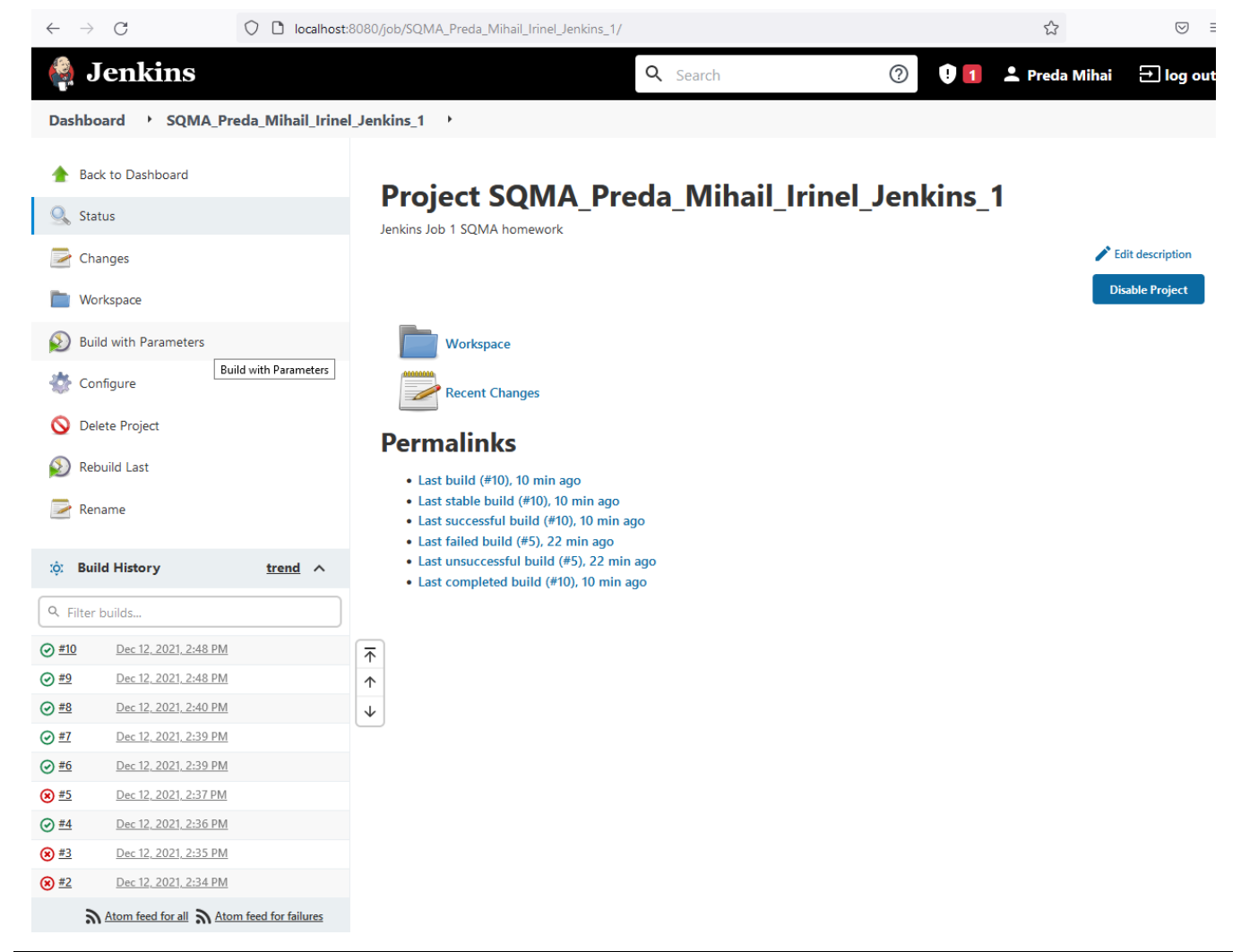


Fig.14

16. Choose testB, then press Build.

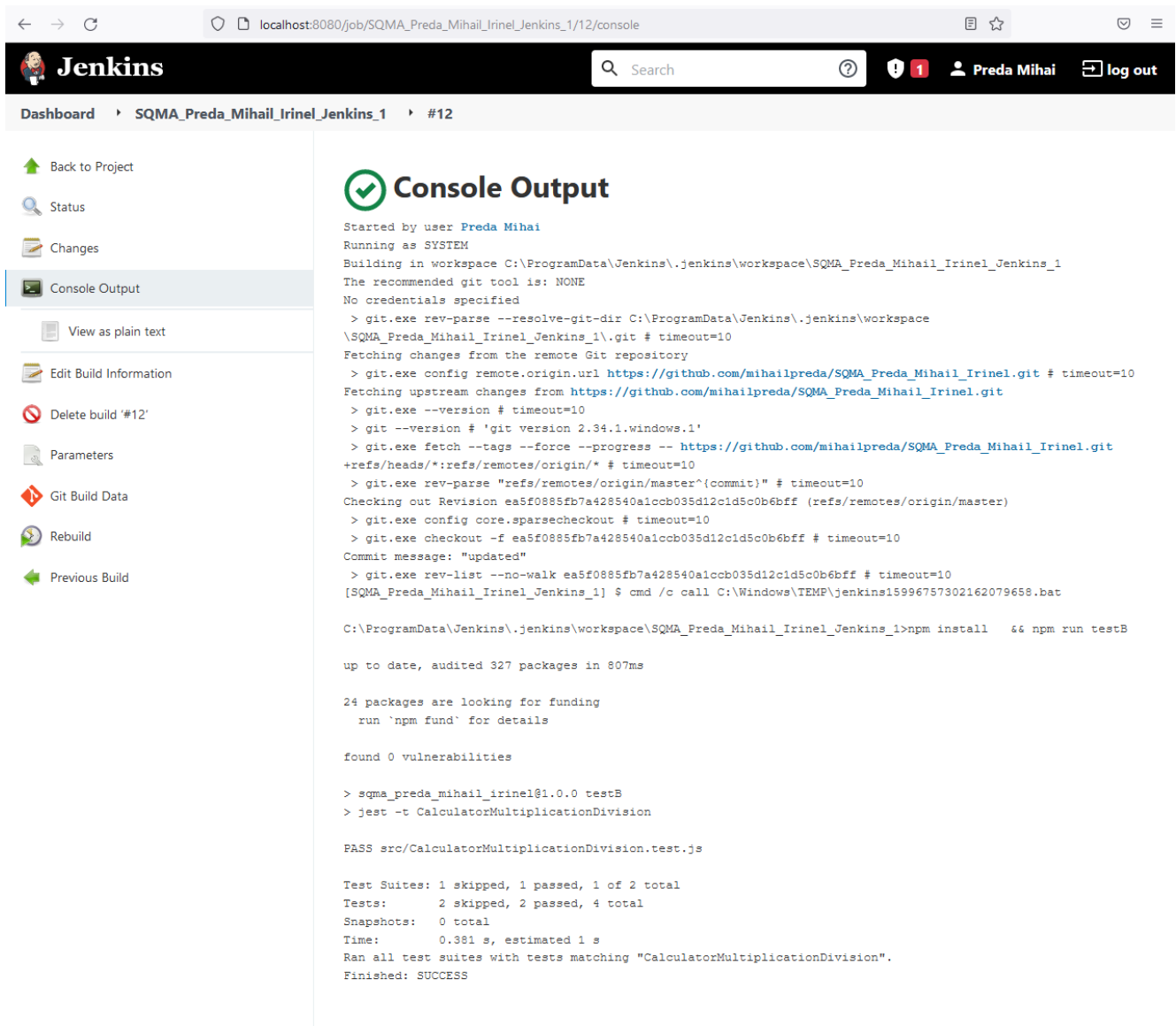
The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, a notification icon with a red '1', the user name 'Preda Mihai', and a 'log out' button. The breadcrumb trail shows 'Dashboard' > 'SQMA_Preda_Mihail_Irinel_Jenkins_1'. The left sidebar contains links: 'Back to Dashboard', 'Status', 'Changes', 'Workspace', 'Build with Parameters', 'Configure', 'Delete Project', 'Rebuild Last', and 'Rename'. The main content area is titled 'Project SQMA_Preda_Mihail_Irinel_Jenkins_1' and states 'This build requires parameters:'. Below this, a parameter 'whichTest' is shown with a dropdown menu set to 'testB' and a 'Build' button. The 'Build History' section is expanded, showing a list of builds with their status (success or failure), build number, and timestamp.

Build Number	Status	Timestamp
#11	Success	Dec 12, 2021, 3:00 PM
#10	Success	Dec 12, 2021, 2:48 PM
#9	Success	Dec 12, 2021, 2:48 PM
#8	Success	Dec 12, 2021, 2:40 PM
#7	Success	Dec 12, 2021, 2:39 PM
#6	Success	Dec 12, 2021, 2:39 PM
#5	Failure	Dec 12, 2021, 2:37 PM
#4	Success	Dec 12, 2021, 2:36 PM
#3	Failure	Dec 12, 2021, 2:35 PM
#2	Failure	Dec 12, 2021, 2:34 PM

At the bottom of the build history, there are links for 'Atom feed for all' and 'Atom feed for failures'.

Fig.15

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



Console Output

```

Started by user Preda Mihai
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\workspace\SQMA_Preda_Mihail_Irinel_Jenkins_1
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\workspace\SQMA_Preda_Mihail_Irinel_Jenkins_1\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe rev-parse --remote.origin.url https://github.com/mihailpreda/SQMA_Preda_Mihail_Irinel.git # timeout=10
Fetching upstream changes from https://github.com/mihailpreda/SQMA_Preda_Mihail_Irinel.git
> git.exe --version # timeout=10
> git --version # 'git version 2.34.1.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/mihailpreda/SQMA_Preda_Mihail_Irinel.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision ea5f0885fb7a428540a1ccb035d12c1d5c0b6b6ff (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f ea5f0885fb7a428540a1ccb035d12c1d5c0b6b6ff # timeout=10
Commit message: "updated"
> git.exe rev-list --no-walk ea5f0885fb7a428540a1ccb035d12c1d5c0b6b6ff # timeout=10
[SQMA_Preda_Mihail_Irinel_Jenkins_1] $ cmd /c call C:\Windows\TEMP\jenkins15996757302162079658.bat

C:\ProgramData\Jenkins\workspace\SQMA_Preda_Mihail_Irinel_Jenkins_1>npm install 66 npm run testB

up to date, audited 327 packages in 807ms

24 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities

> sqma_preda_mihail_irinel@1.0.0 testB
> jest -t CalculatorMultiplicationDivision

PASS src/CalculatorMultiplicationDivision.test.js

Test Suites: 1 skipped, 1 passed, 1 of 2 total
Tests:      2 skipped, 2 passed, 4 total
Snapshots:  0 total
Time:       0.381 s, estimated 1 s
Ran all test suites with tests matching "CalculatorMultiplicationDivision".
Finished: SUCCESS

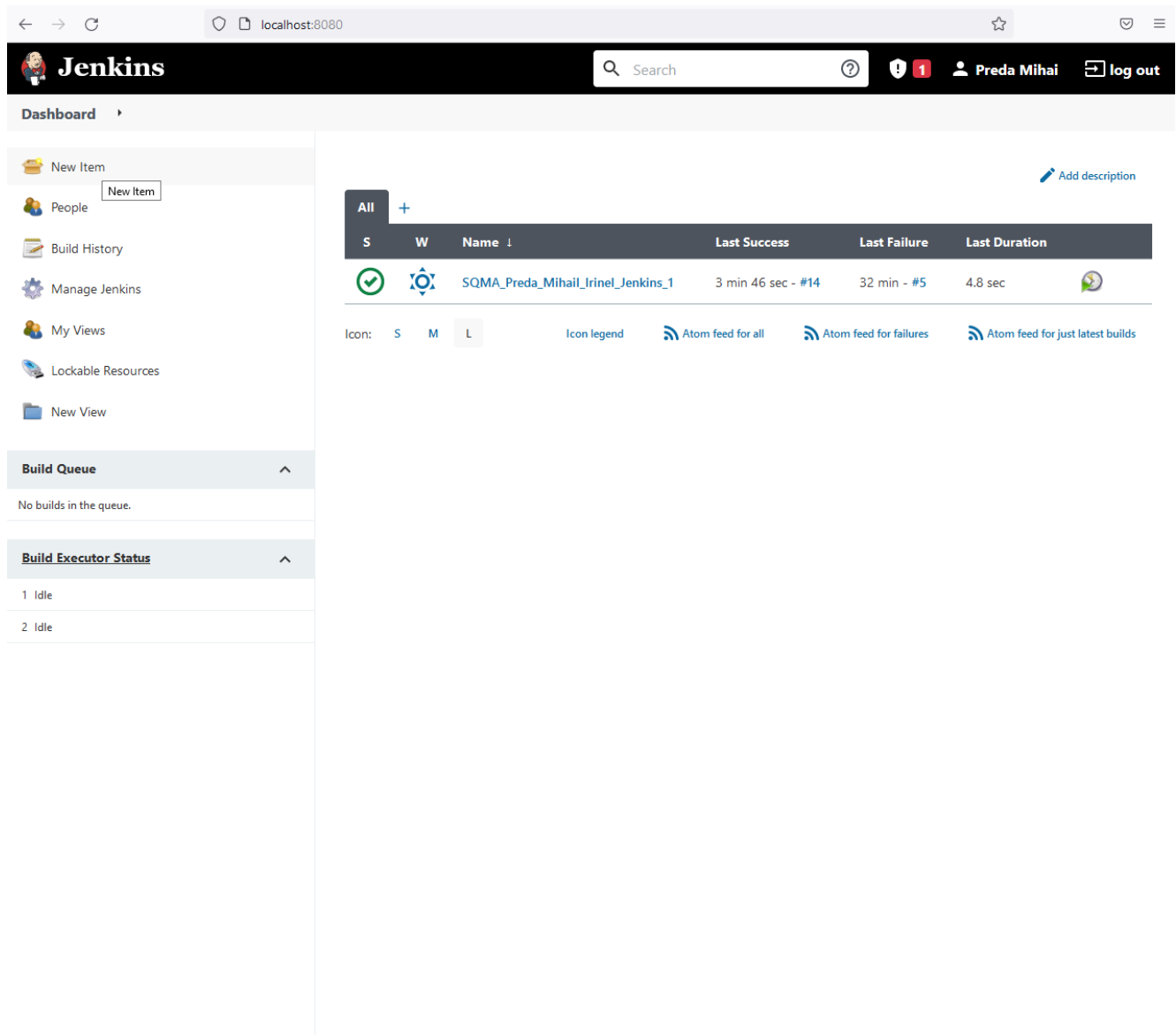
```

Fig.16

Homework 2

Steps

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



The screenshot displays the Jenkins web interface at localhost:8080. The top navigation bar includes the Jenkins logo, a search bar, a notification bell with one alert, the user name 'Preda Mihai', and a 'log out' button. The left sidebar contains a 'Dashboard' menu and a list of links: 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', 'Lockable Resources', and 'New View'. The main content area shows the 'All' view of builds. A table lists one build: 'SQMA_Preda_Mihail_Irinel_Jenkins_1', which is successful, took 3 min 46 sec, and is build #14. Below the table are filters for 'Icon' (S, M, L) and links for 'Icon legend', 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'. On the left, the 'Build Queue' section is empty, and the 'Build Executor Status' section shows two idle executors.

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	⚙	SQMA_Preda_Mihail_Irinel_Jenkins_1	3 min 46 sec - #14	32 min - #5	4.8 sec

Icon: S M L Icon legend Atom feed for all Atom feed for failures Atom feed for just latest builds

Fig.1







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

← → ↻ localhost:8080/view/all/newJob ☆ 🔔 1 Preda Mihai 🚪 log out

Dashboard > All >

Enter an item name

» 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 than software build.
-  **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.
-  **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, so you can have multiple things of the same name as long as 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:


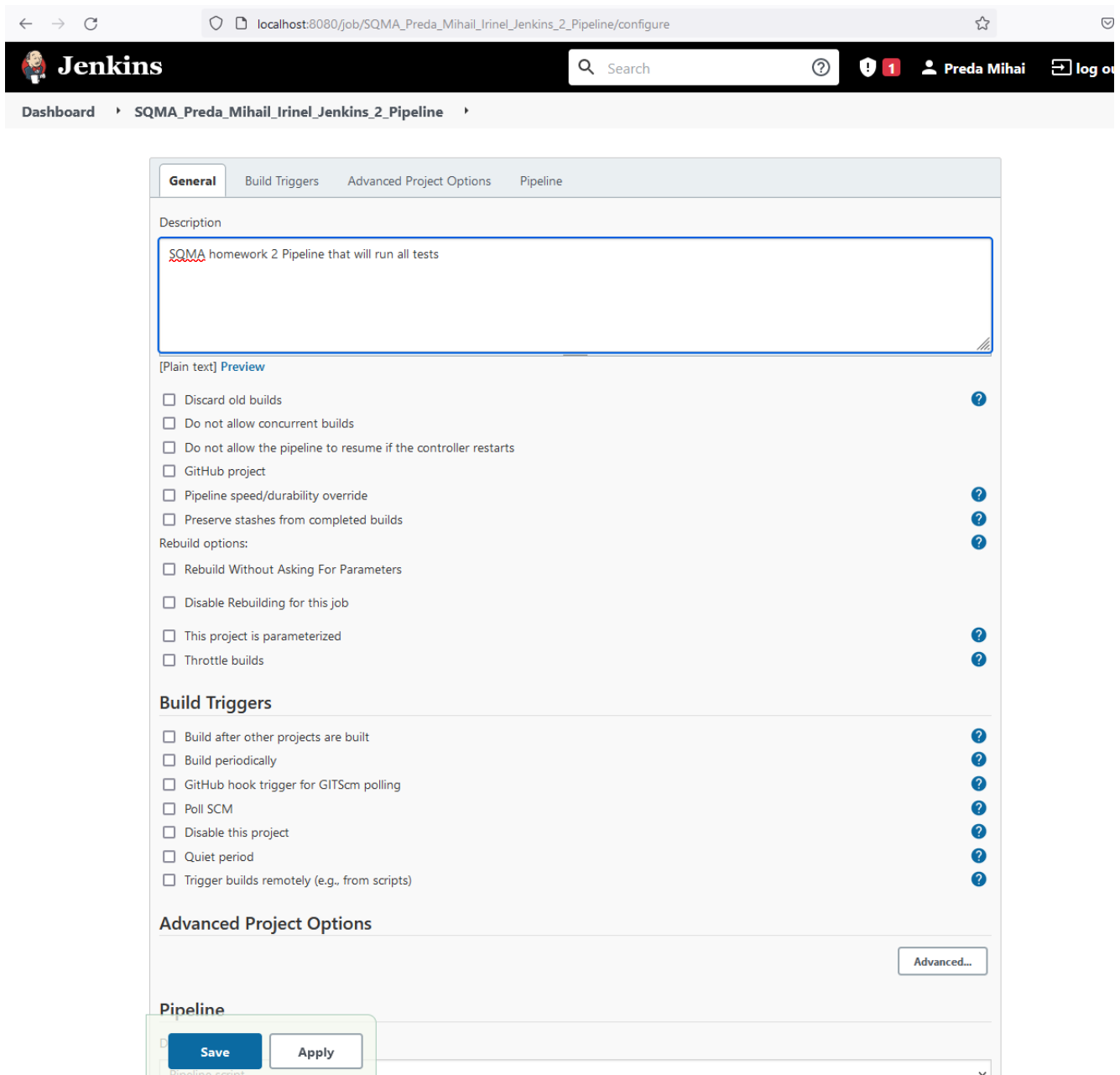
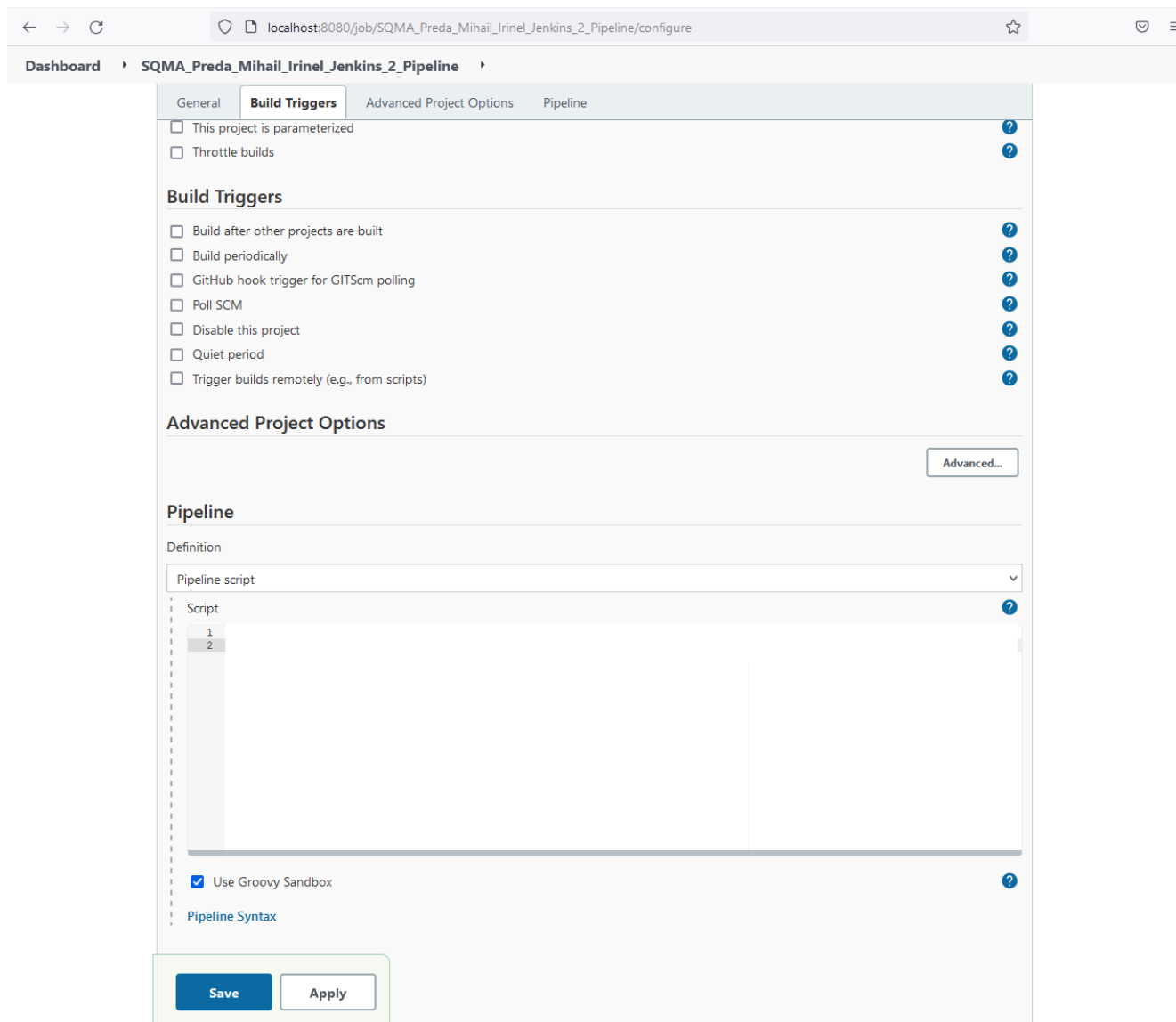
 Copy from

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**

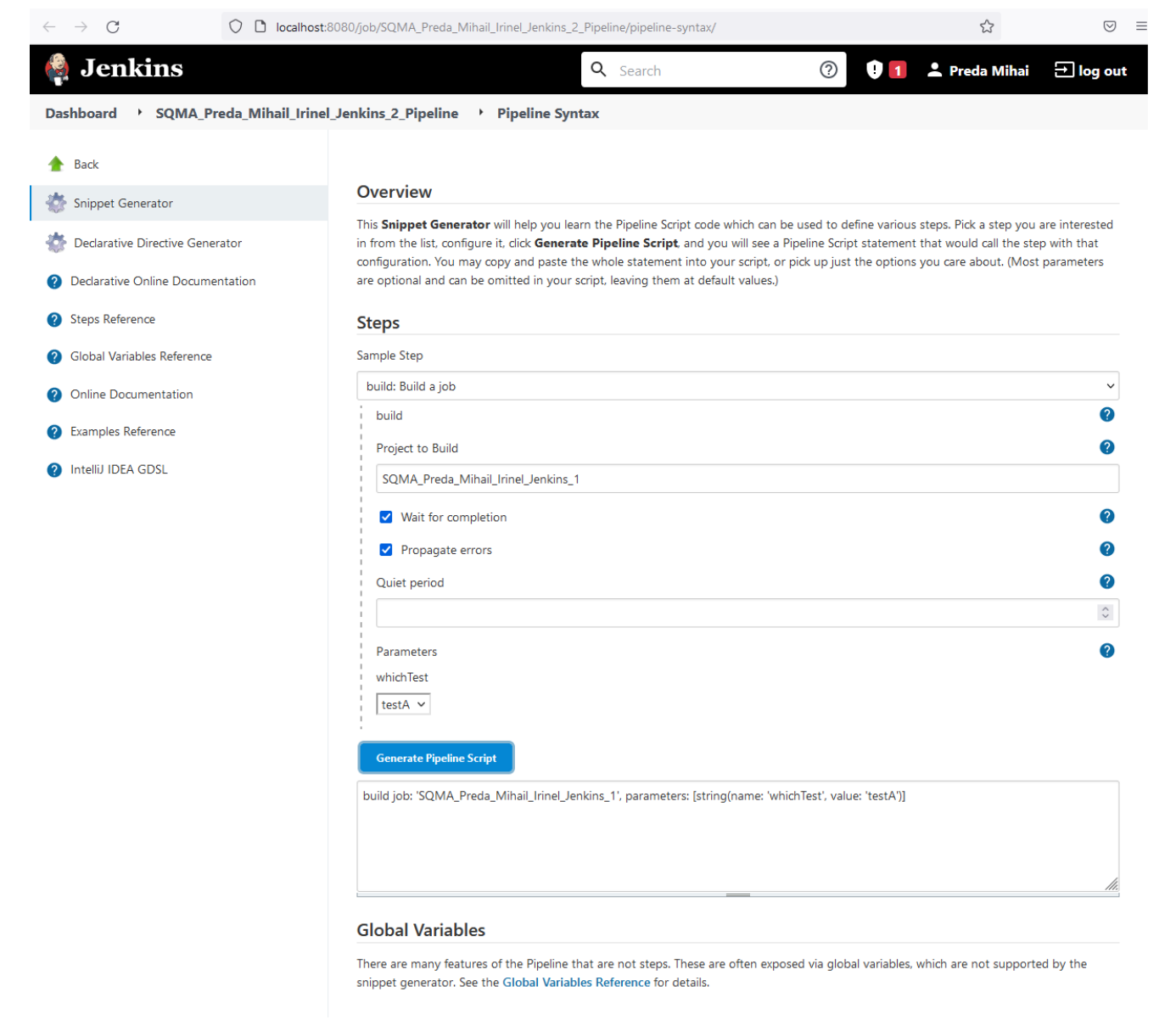


Fig.5

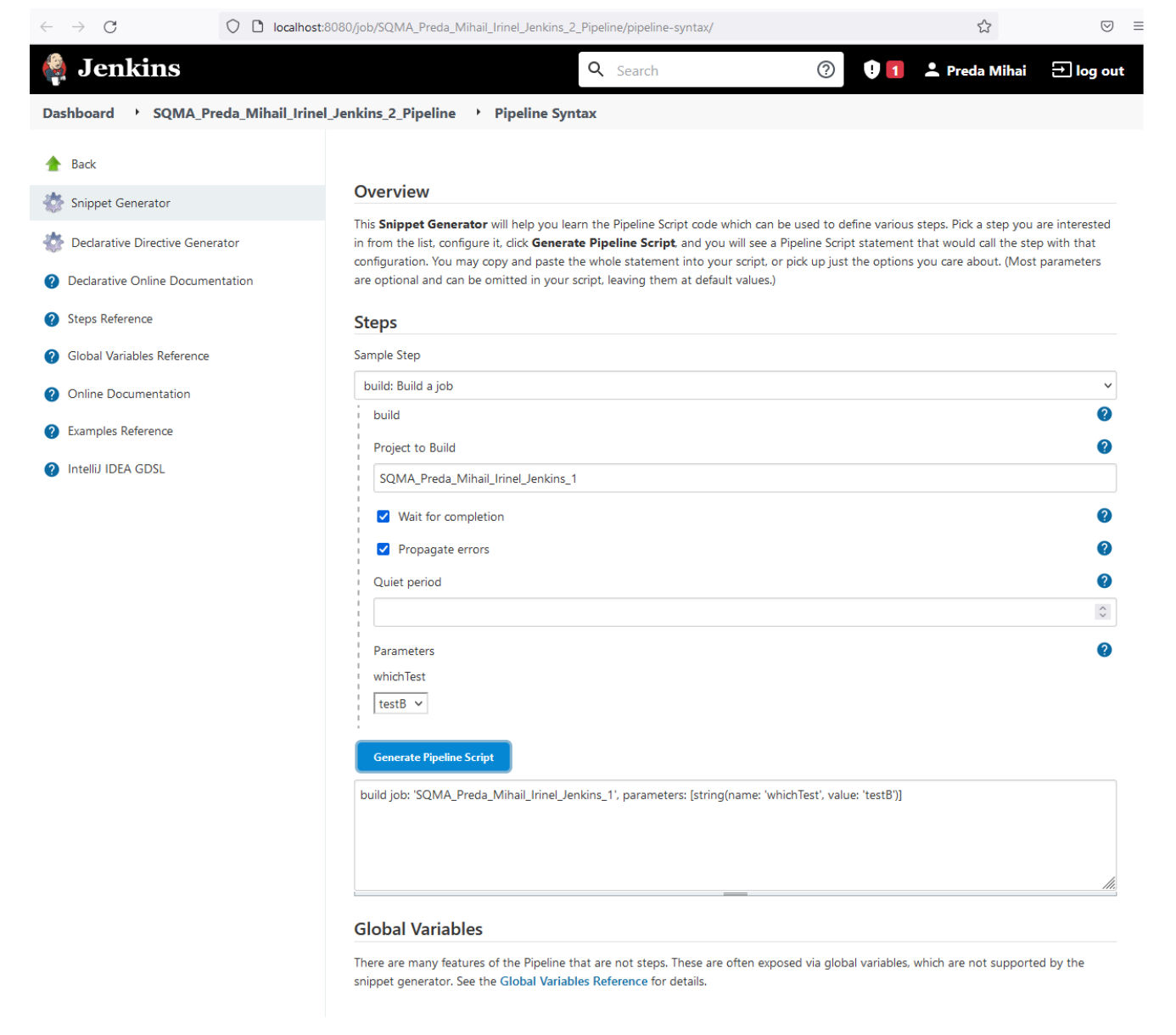


Fig.6

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

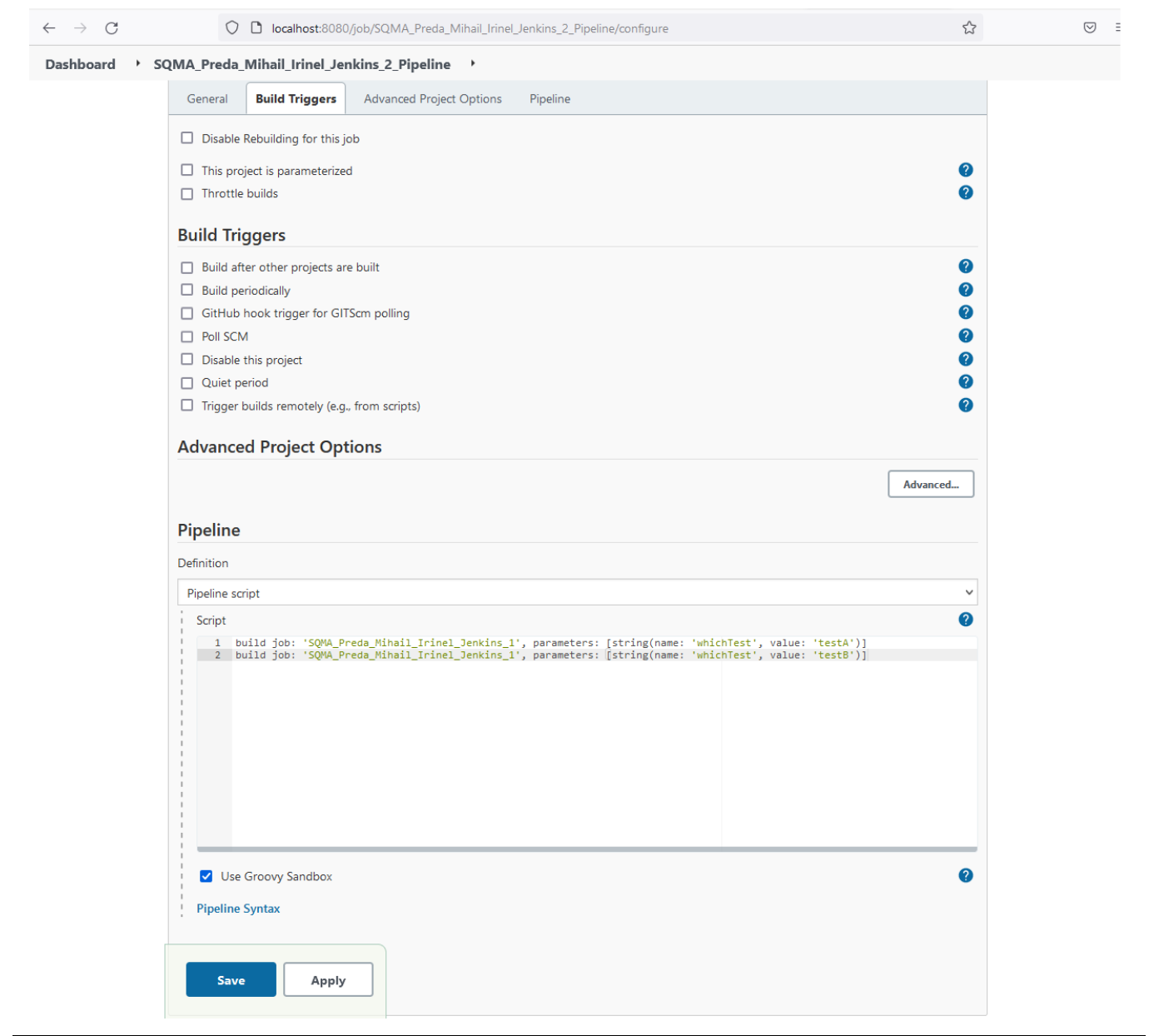


Fig.7

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

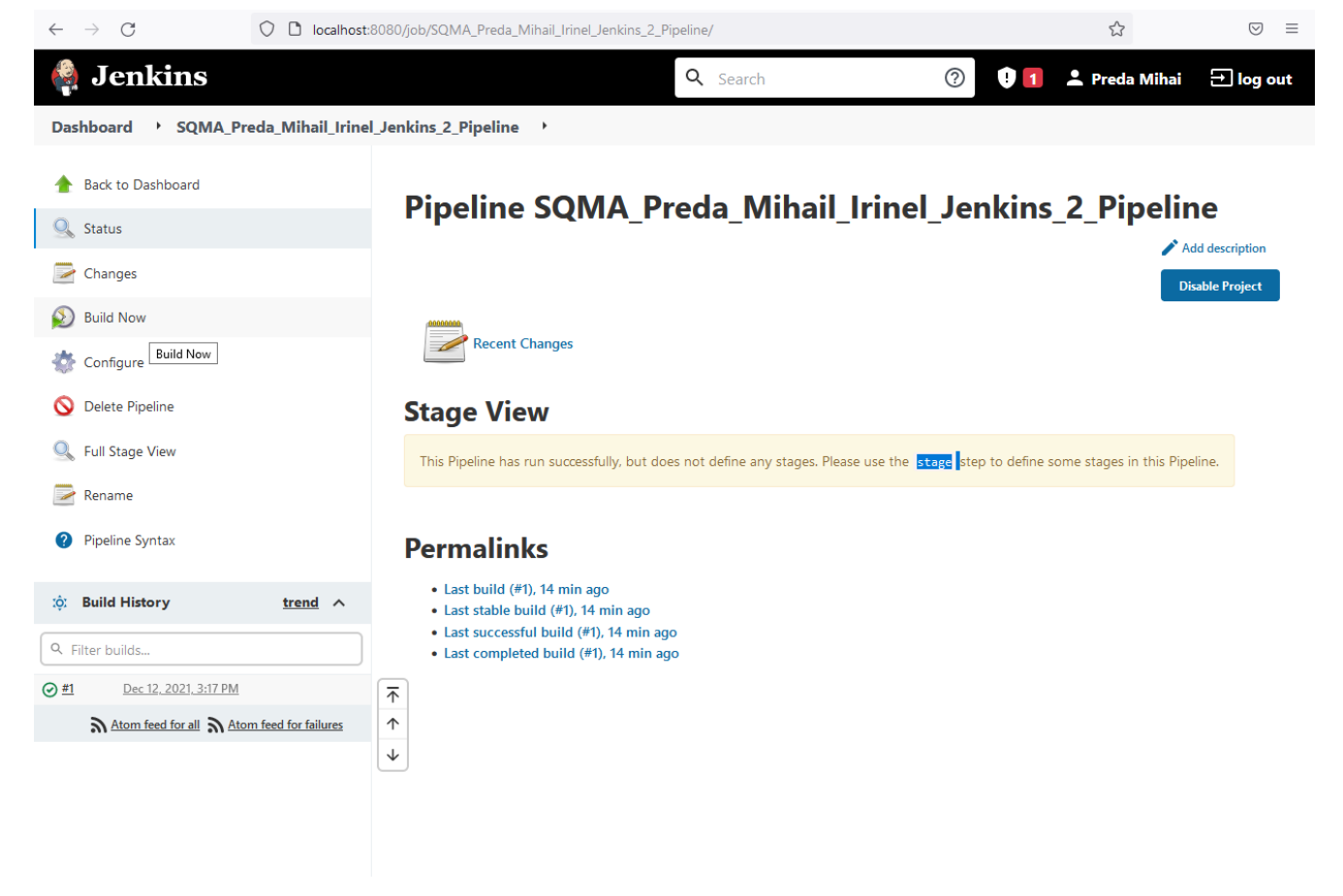


Fig.8

8. To see Pipeline result click on latest done build

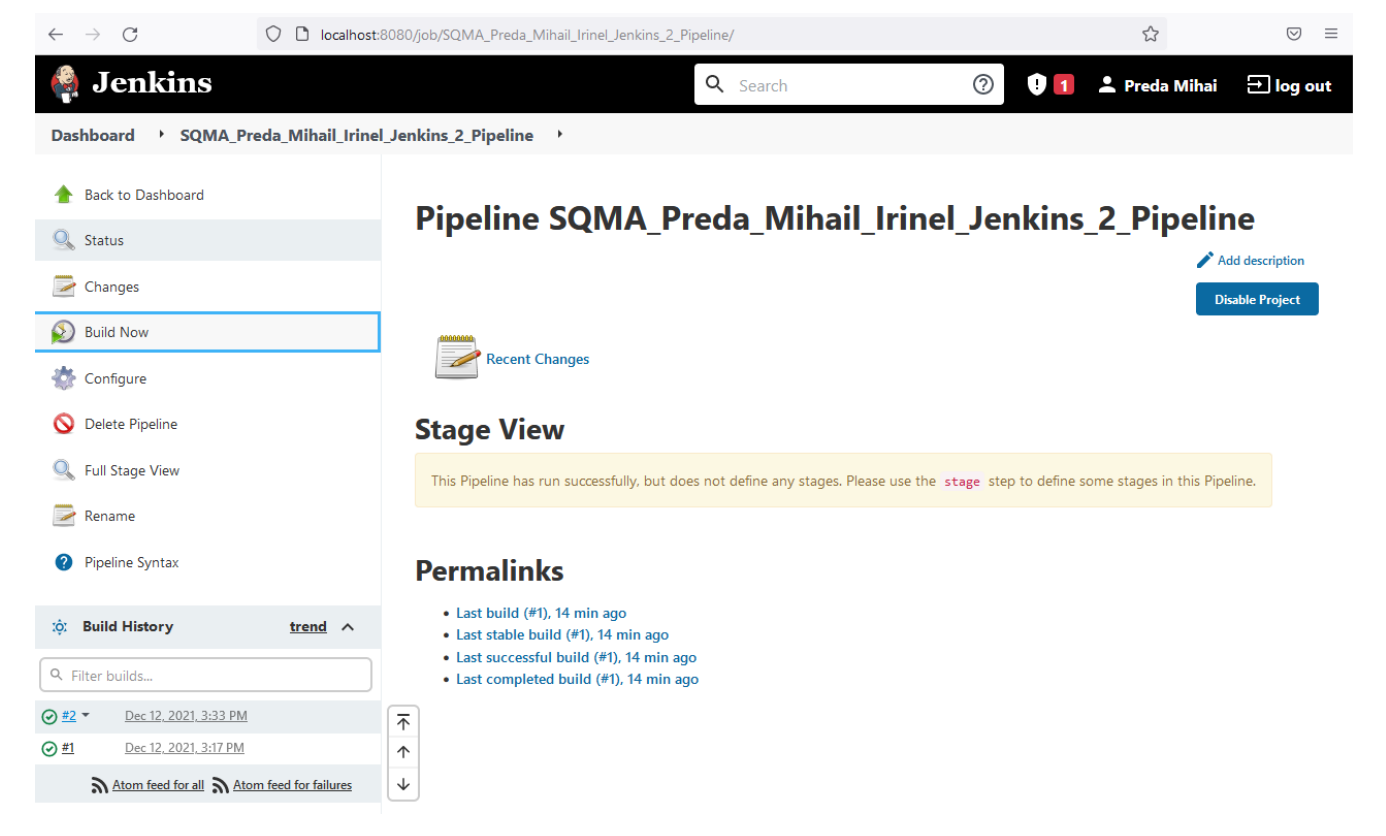


Fig.9

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

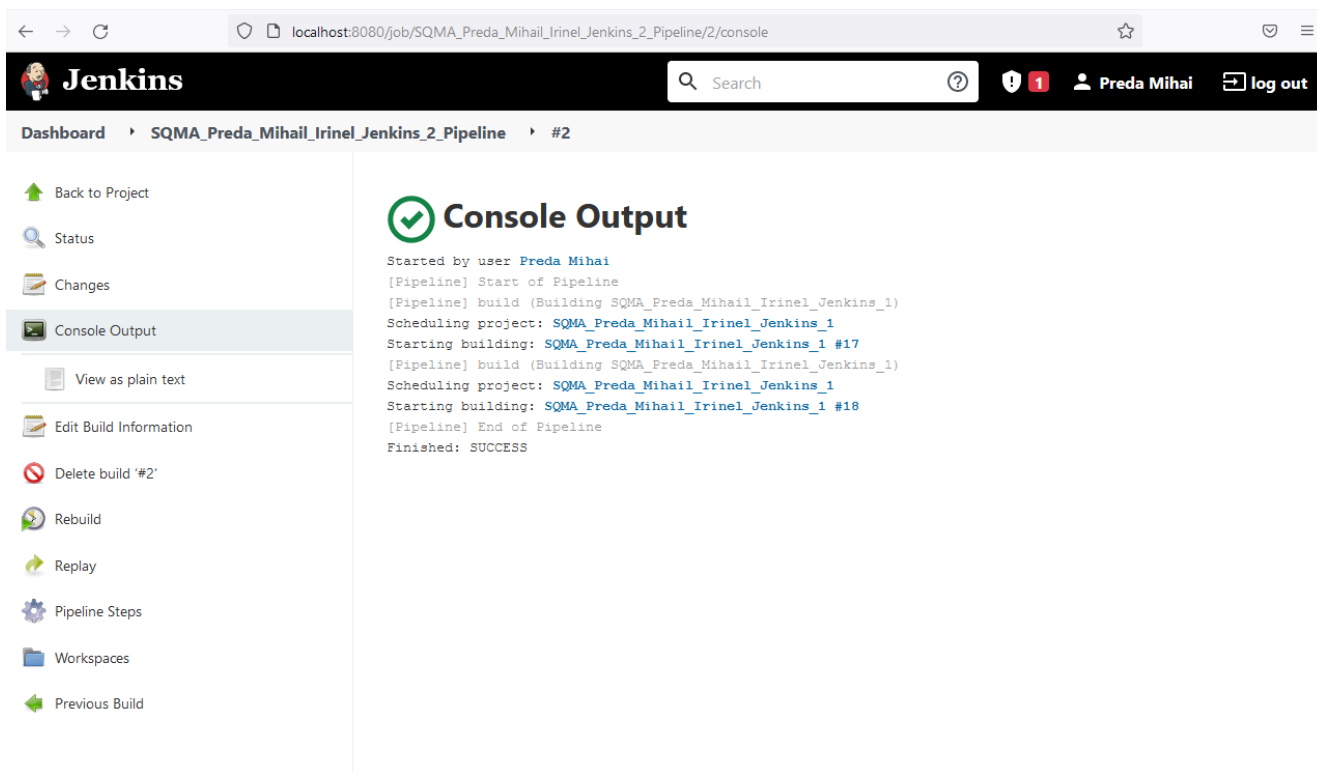


Fig.10

Project setup

```
npm install
```

Project run

```
npm start
```

Project test

```
npm test
```

Project individual test suites

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