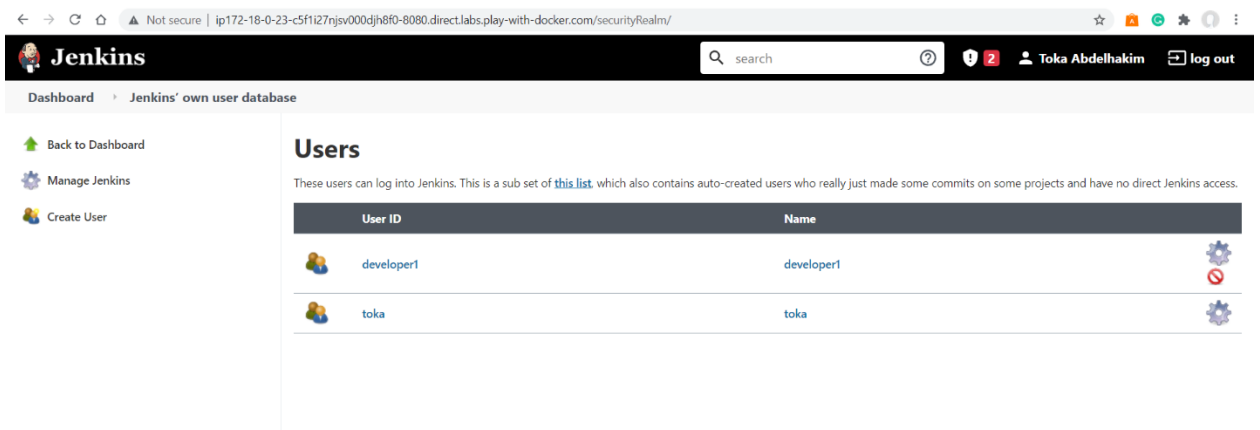


- 1-install jenkins with docker image
- 2- install role based authorization plugin
- 3- create new user
- 4- create read role and assign it to the new user

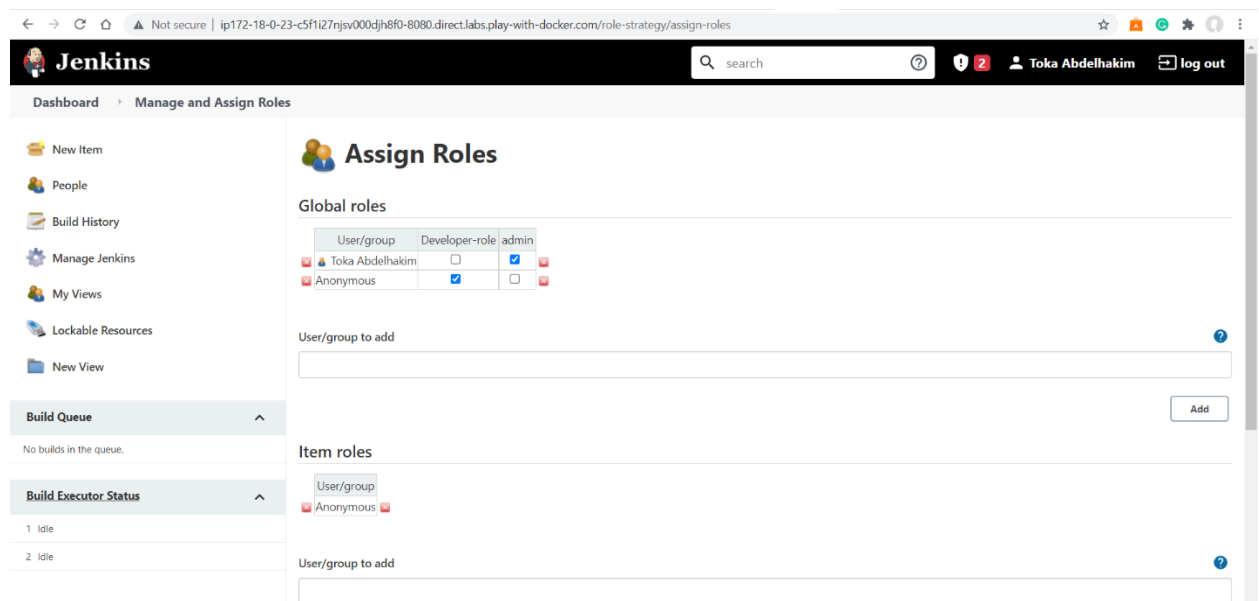
## New user:



The screenshot shows the Jenkins 'Users' page. The breadcrumb trail is 'Dashboard > Jenkins' own user database'. The left sidebar contains links: 'Back to Dashboard', 'Manage Jenkins', and 'Create User'. The main content area is titled 'Users' and includes a description: 'These users can log into Jenkins. This is a sub set of [this list](#), which also contains auto-created users who really just made some commits on some projects and have no direct Jenkins access.' Below this is a table with two columns: 'User ID' and 'Name'. The table lists two users: 'developer1' and 'toka'. Each user entry has a gear icon for configuration and a red 'X' icon for deletion.

User ID	Name
developer1	developer1
toka	toka

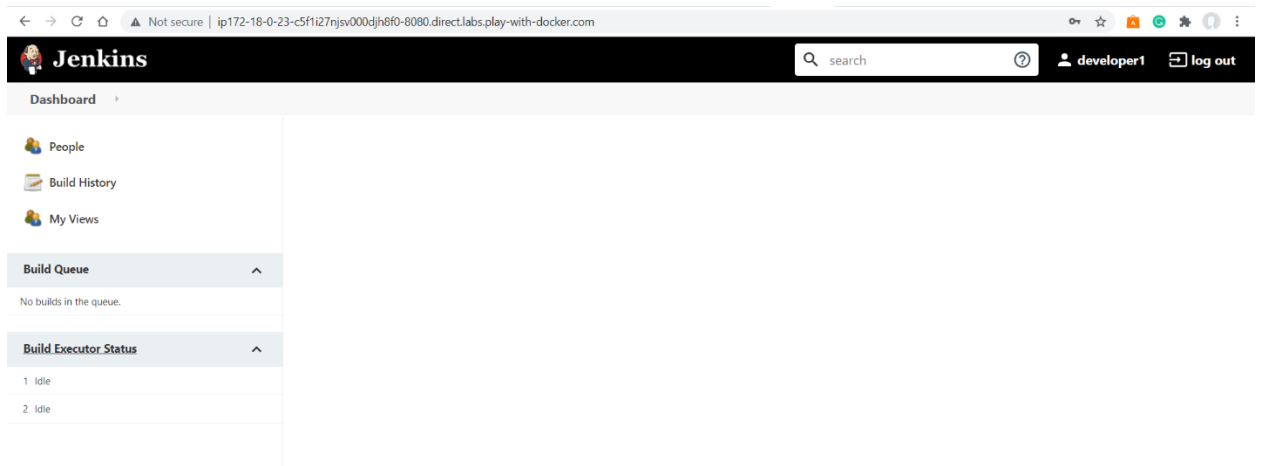
## Assign Role to User:



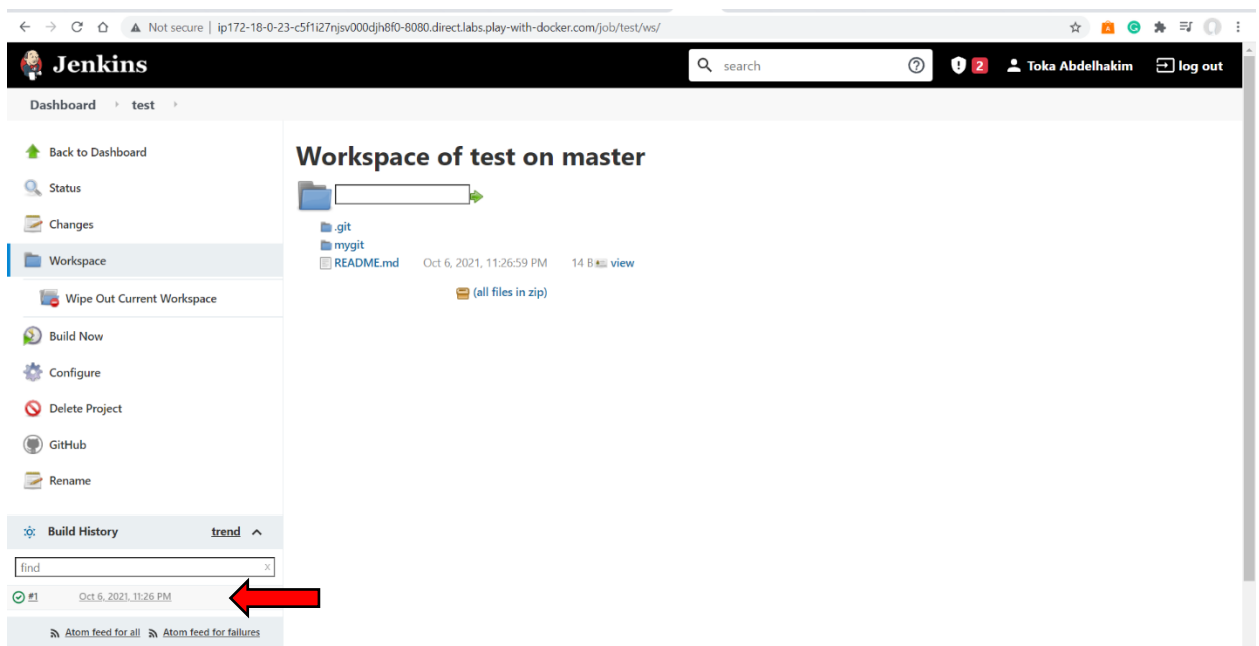
The screenshot shows the Jenkins 'Assign Roles' page. The breadcrumb trail is 'Dashboard > Manage and Assign Roles'. The left sidebar contains links: 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', 'Lockable Resources', and 'New View'. The main content area is titled 'Assign Roles' and is divided into two sections: 'Global roles' and 'Item roles'. The 'Global roles' section contains a table with columns 'User/group', 'Developer-role', and 'admin'. The table lists two users: 'Toka Abdelhakim' and 'Anonymous'. The 'Anonymous' user has the 'Developer-role' checked. Below the table is a text input field labeled 'User/group to add' and an 'Add' button. The 'Item roles' section contains a table with columns 'User/group' and 'Anonymous'. The 'Anonymous' user has the 'Anonymous' role checked. Below the table is a text input field labeled 'User/group to add' and an 'Add' button.

User/group	Developer-role	admin
Toka Abdelhakim	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Anonymous	<input checked="" type="checkbox"/>	<input type="checkbox"/>

New user logged in:



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



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

```
Cloning the remote Git repository
Cloning repository https://github.com/TokaAbdelhakim/jenkins-private
> git init /var/jenkins_home/workspace/Dec # timeout=10
Fetching upstream changes from https://github.com/TokaAbdelhakim/jenkins-private
> git --version # timeout=10
> git --version # 'git version 2.30.2'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/TokaAbdelhakim/jenkins-private +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/TokaAbdelhakim/jenkins-private # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/master^(commit) # timeout=10
Checking out Revision 1a430f61f116fed5f9767bbf8be2c303a5df24e8 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 1a430f61f116fed5f9767bbf8be2c303a5df24e8 # timeout=10
Commit message: "Declarative"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (List)
[Pipeline] sh
+ ls
Jenkinsfile
Jenkinsfile.txt
README.md
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Code:

```
11 lines (10 sloc) | 138 Bytes
1 pipeline {
2   agent any
3
4   stages {
5     stage('List') {
6       steps {
7         sh 'ls'
8       }
9     }
10  }
11 }
```

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

The screenshot shows the Jenkins web interface. On the left sidebar, the 'Scripted' pipeline is selected, and the 'Console Output' tab is active. A red arrow points to the 'Scripted' link in the sidebar. The main area displays the console output for build #3. The output shows the pipeline starting, fetching changes from a GitHub repository, and running a 'node' stage. Within the 'node' stage, a 'stage' named 'Build' is defined, which contains a 'git' step to checkout the repository and a 'sh' step to run 'ls'. A second red arrow points to the output of the 'ls' command, which lists the files in the workspace: 'Jenkinsfile', 'Jenkinsfile.txt', and 'README.md'.

```
Started by user Toka Abdelhakim
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/Scripted
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] git
The recommended git tool is: NONE
using credential GitHub
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/Scripted/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/TokaAbdelhakim/jenkins-private # timeout=10
Fetching upstream changes from https://github.com/TokaAbdelhakim/jenkins-private
> git --version # timeout=10
> git --version # 'git version 2.30.2'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/TokaAbdelhakim/jenkins-private +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 1a430f61f116fed5f9767bbf8be2c303a5df24e8 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 1a430f61f116fed5f9767bbf8be2c303a5df24e8 # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git checkout -b master 1a430f61f116fed5f9767bbf8be2c303a5df24e8 # timeout=10
Commit message: "Declarative"
First time build. Skipping changelog.
[Pipeline] sh
+ ls
Jenkinsfile
Jenkinsfile.txt
README.md
[Pipeline] }
[Pipeline] // stage
```

Code:

The screenshot shows the Jenkins Pipeline Definition editor. The 'Pipeline' tab is selected, and the 'Definition' section is visible. The 'Pipeline script' dropdown is set to 'Scripted'. The script content is as follows:

```
1 node {
2
3   stage('Build') {
4     git credentialsId: 'GitHub', url: 'https://github.com/TokaAbdelhakim/jenkins-private'
5     // Run the maven build
6     sh 'ls'
7   }
8
9
10
```

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

The screenshot shows the Jenkins web interface for a multi-branch pipeline named 'multi-branch'. The left sidebar contains navigation links: Up, Status, Configure, Scan Repository Now, Scan Repository Log, Multibranch Pipeline Events, Delete Multibranch Pipeline, People, Build History, Project Relationship, Check File Fingerprint, GitHub, Rename, and Pipeline Syntax. The main panel displays the 'multi-branch' configuration with tabs for 'Branches (2)' and 'Pull Requests (0)'. A table lists the branches:

S	W	Name	Last Success	Last Failure	Last Duration
...	...	documents	N/A	N/A	N/A
...	...	master	N/A	N/A	N/A

Below the table, there is a 'Icon: S M L' section and a 'Legend' section with options for 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'. A 'Disable Multibranch Pipeline' button is in the top right. An 'Activate Windows' watermark is visible at the bottom right.

The screenshot shows the Jenkins console output for the 'documents' branch. The breadcrumb navigation at the top is 'Dashboard > multi-branch > documents > #3', with a red arrow pointing to the '#3' link. The console output shows the following commands and results:

```
> git --version # 'git version 2.30.2'
using GIT_ASKPASS to set credentials
> git fetch --no-tags --force --progress -- https://github.com/TokaAbdelhakim/jenkins-private.git +refs/heads/documents:refs/remotes/origin/documents #
timeout=10
Checking out Revision e75158235dd95e39e7df07ee0d0e6adc0e851bac (documents)
> git config core.sparsecheckout # timeout=10
> git checkout -f e75158235dd95e39e7df07ee0d0e6adc0e851bac # timeout=10
Commit message: "lab1 questions"
> git rev-list --no-walk e75158235dd95e39e7df07ee0d0e6adc0e851bac # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (List)
[Pipeline] sh
+ ls
Jenkinsfile
Jenkinsfile.txt
README.md
lab1.docx
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline

Github has been notified of this commit's build result

Finished: SUCCESS
```

An 'Activate Windows' watermark is visible at the bottom right.

← → ↻ ⌂ ⚠ Not secure | ip172-18-0-62-c5fglovnjsv000e1rs9g-8080.direct.labs.play-with-docker.com/job/multi-branch/job/master/1/console ☆ 📄 🌐 ⚙️ 👤

Dashboard > multi-branch > master > #1

Pipeline Steps

Workspaces

```
using GIT_ASKPASS to set credentials
> git fetch --no-tags --force --progress -- https://github.com/TokaAbdelhakim/jenkins-private.git +refs/heads/master:refs/remotes/origin/master # timeout=10
> git config remote.origin.url https://github.com/TokaAbdelhakim/jenkins-private.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/master:refs/remotes/origin/master # timeout=10
Avoid second fetch
Checking out Revision 1a430f61f116fed5f9767bbf8be2c303a5df24e8 (master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 1a430f61f116fed5f9767bbf8be2c303a5df24e8 # timeout=10
Commit message: "Declerative"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (List)
[Pipeline] sh
+ ls
Jenkinsfile
Jenkinsfile.txt
README.md
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline

GitHub has been notified of this commit's build result

Finished: SUCCESS
```

Activate Windows  
Go to Settings to activate Windows.

← → ↻ ⌂ ⚠ Not secure | ip172-18-0-62-c5fglovnjsv000e1rs9g-8080.direct.labs.play-with-docker.com/job/multi-branch/

**Jenkins** 🔍 search ? 🔔 1 👤 Toka Abdelhakim 🚪 log out

Dashboard > multi-branch >

Up

Status

Configure

Scan Repository Now

Scan Repository Log

Multibranch Pipeline Events

Delete Multibranch Pipeline

People

Build History

Project Relationship

Check File Fingerprint

GitHub

Rename

Pipeline Syntax

## multi-branch

Disable Multibranch Pipeline

Branches (2) Pull Requests (0)

S	W	Name	Last Success	Last Failure	Last Duration	
✓	🔗	documents	1 min 32 sec - #3	N/A	3.3 sec	🔗
✓	🔗	master	29 sec - #1	N/A	2.7 sec	🔗

Icon: S M L

Legend 🔔 Atom feed for all 🔔 Atom feed for failures 🔔 Atom feed for just latest builds

Activate Windows  
Go to Settings to activate Windows.