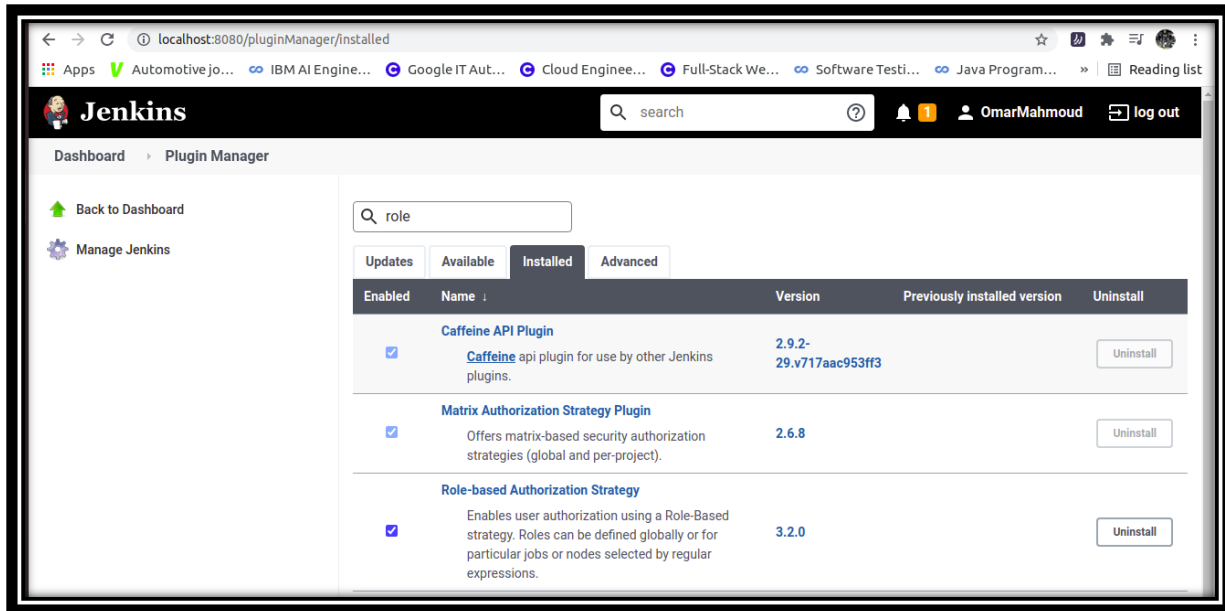


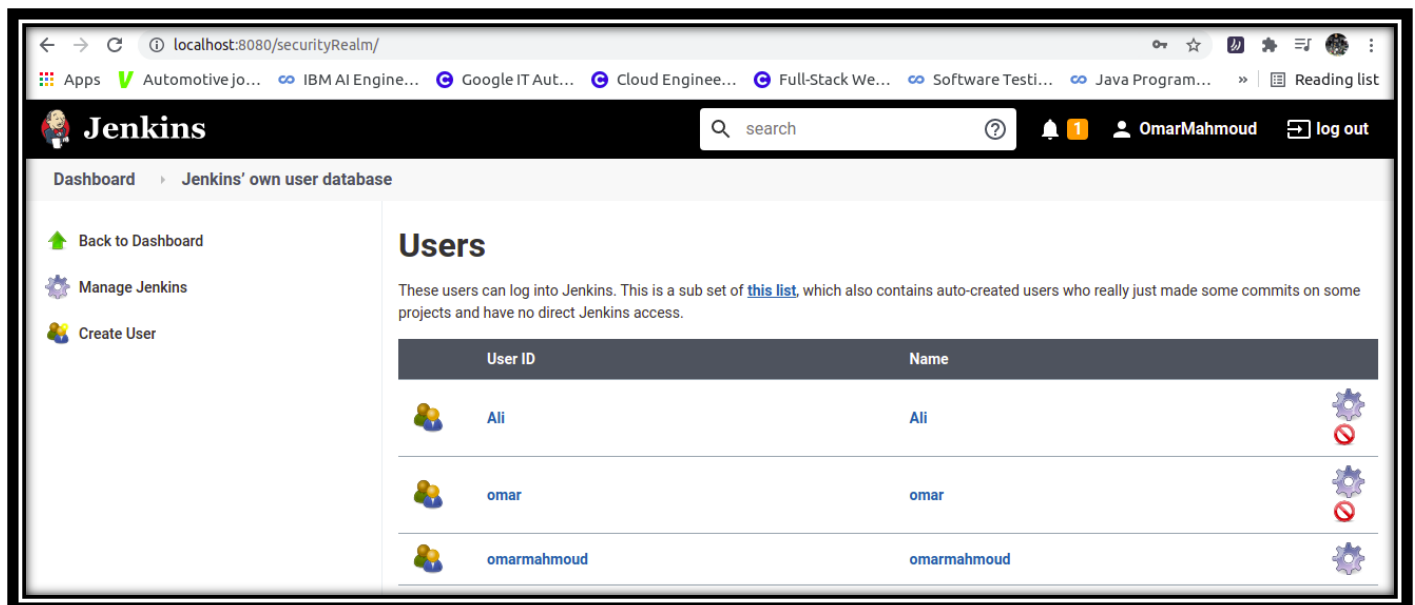
Jenkins

Task day 1

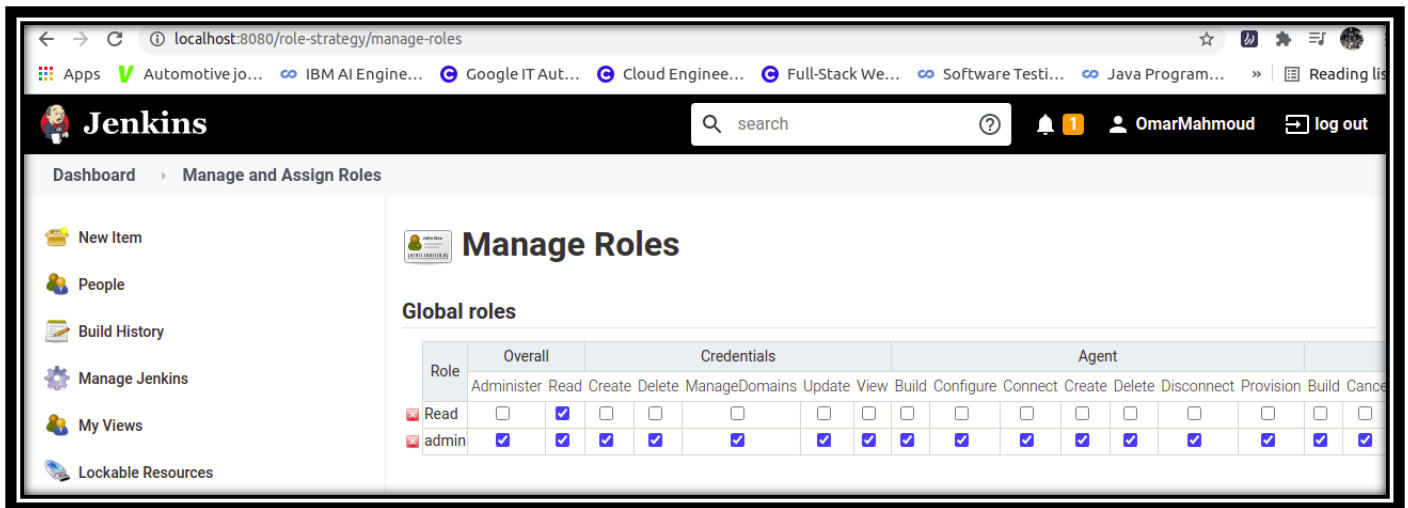
- 1- install jenkins with docker image
- 2- install role based authorization plugin



- 3- create new user:

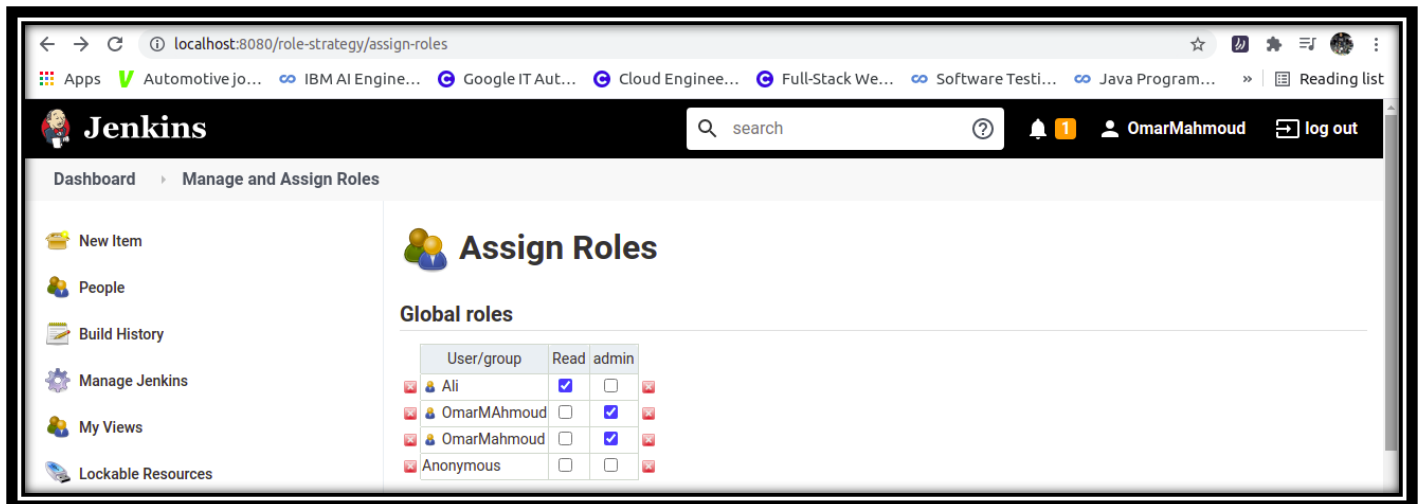


4- create read role and assign it to the new user:



The screenshot shows the Jenkins 'Manage Roles' page. The left sidebar contains links for 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', and 'Lockable Resources'. The main content area is titled 'Manage Roles' and shows a table of 'Global roles'. The table has columns for 'Role', 'Overall', 'Credentials', and 'Agent'. The 'Read' role is highlighted in red, and the 'admin' role is highlighted in red. The 'Read' role has checkboxes for 'Read', 'Create', 'Delete', 'ManageDomains', 'Update', 'View', 'Build', 'Configure', 'Connect', 'Create', 'Delete', 'Disconnect', 'Provision', 'Build', and 'Cancel'. The 'admin' role has checkboxes for 'Read', 'Create', 'Delete', 'ManageDomains', 'Update', 'View', 'Build', 'Configure', 'Connect', 'Create', 'Delete', 'Disconnect', 'Provision', 'Build', and 'Cancel'.

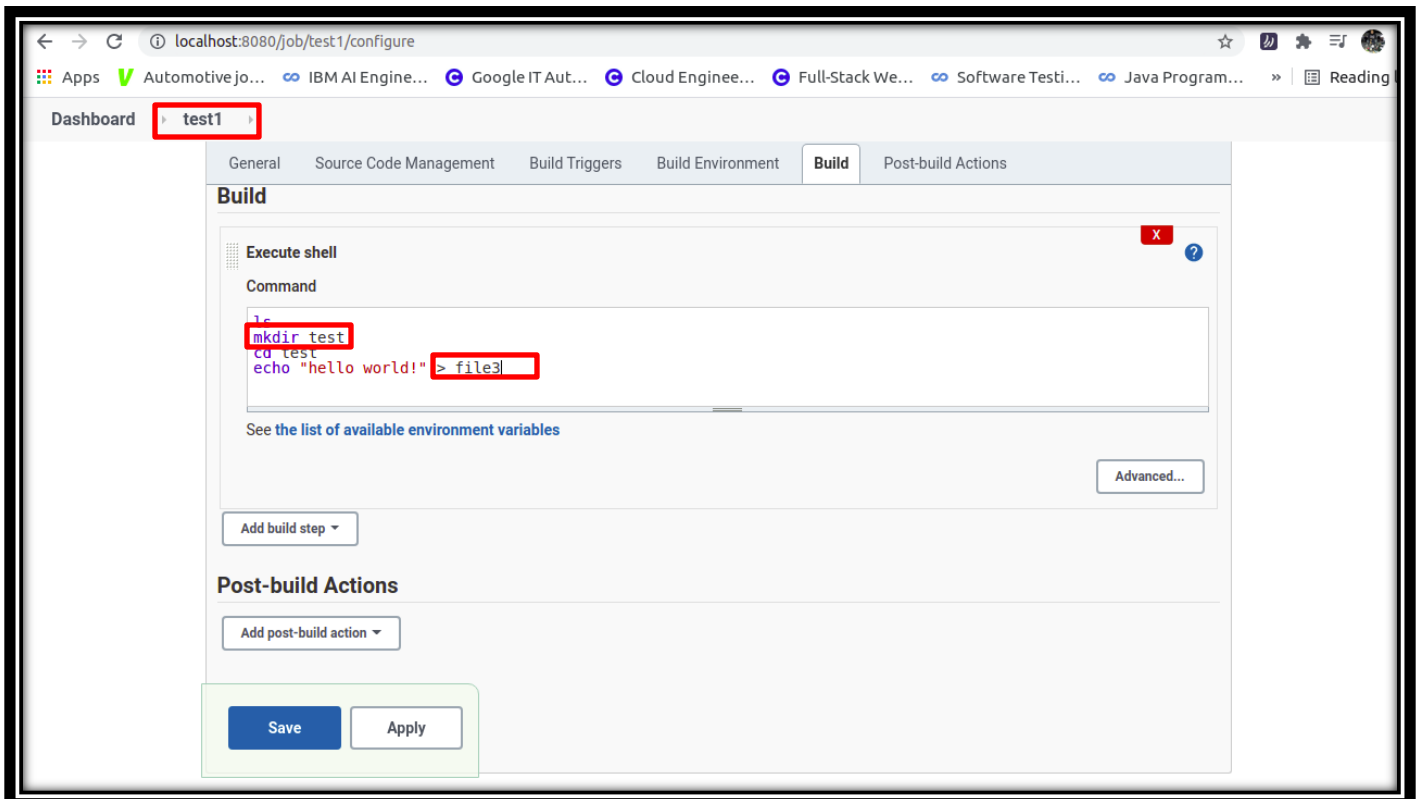
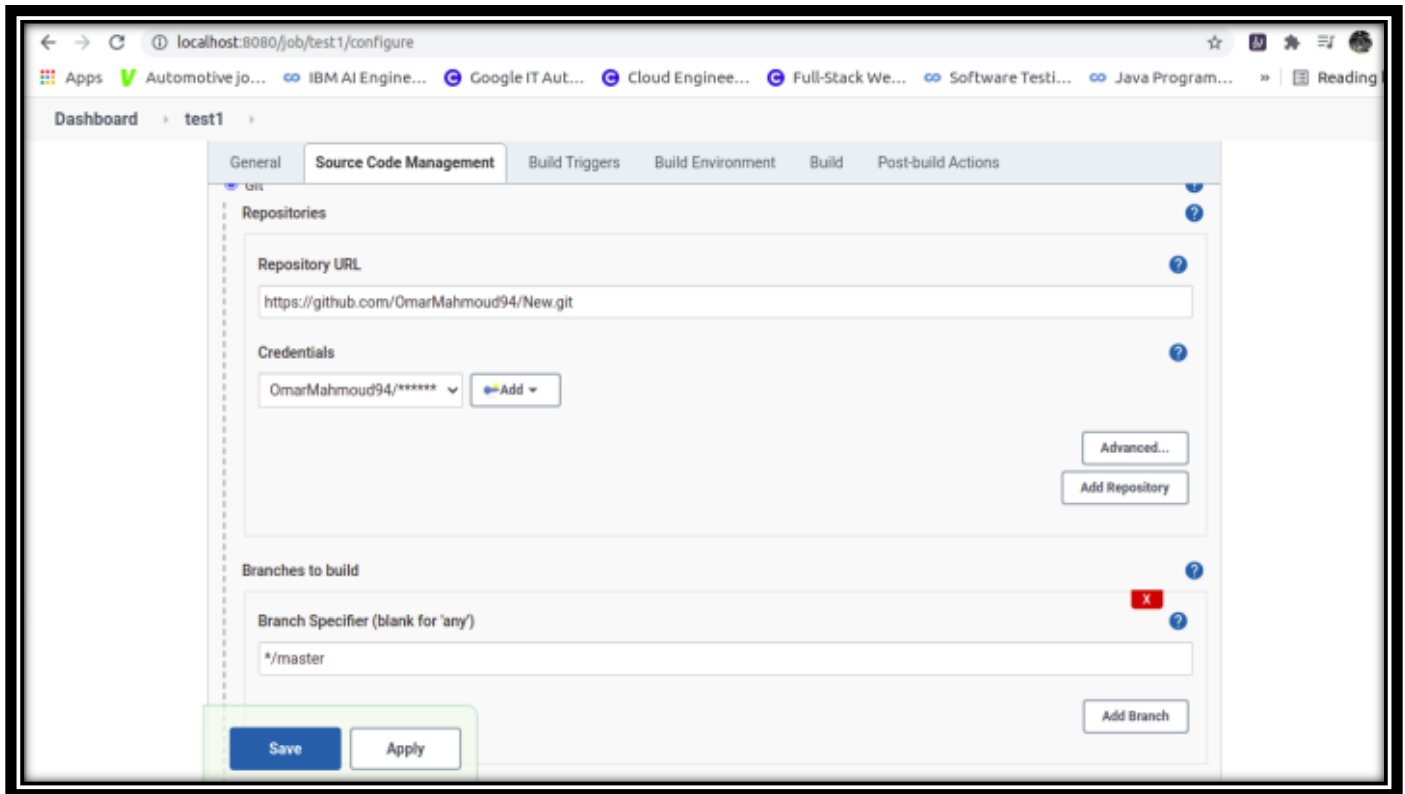
Role	Overall	Credentials	Agent
Read	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>



The screenshot shows the Jenkins 'Assign Roles' page. The left sidebar contains links for 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', and 'Lockable Resources'. The main content area is titled 'Assign Roles' and shows a table of 'Global roles'. The table has columns for 'User/group', 'Read', and 'admin'. The 'Ali' user is highlighted in red, and the 'OmarMAhmad' user is highlighted in red. The 'Anonymous' user is highlighted in red.

User/group	Read	admin
Ali	<input checked="" type="checkbox"/>	<input type="checkbox"/>
OmarMAhmad	<input type="checkbox"/>	<input checked="" type="checkbox"/>
OmarMAhmad	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Anonymous	<input type="checkbox"/>	<input type="checkbox"/>

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



localhost:8080/job/test1/ws/

Jenkins

test1

Workspace of test1 on master

test

1	Oct 3, 2021 9:24:25 PM	0 B	view
file1	Oct 3, 2021 9:24:25 PM	7 B	view
jenkinsfile	Oct 3, 2021 9:24:25 PM	273 B	view

(all files in zip)

localhost:8080/job/test1/ws/test/

Jenkins

test1

Workspace of test1 on master

test /

file3

file3	Oct 3, 2021 9:24:29 PM	13 B	view
-------	------------------------	------	------

(all files in zip)

es Text Editor

Sun 23:25

file3

hello world!

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

Dashboard > **declarative**

General Build Triggers Advanced Project Options **Pipeline**

Pipeline

Definition

Pipeline script from SCM

SCM

Git

Repositories

Repository URL

https://github.com/OmarMahmoud94/New.git

Credentials

OmarMahmoud94/***** Add

Advanced... Add Repository

New / jenkinsfile in master Cancel changes

<> Edit file Preview changes Spaces 4 No wrap

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('list_files') {
6       steps {
7         sh 'ls'
8       }
9     }
10  }
11 }
12
```

Dashboard > **declarative** > #8

Pipeline Steps Workspaces Previous Build

```
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 1c5145d993ebd9a19131b336482dfa58efdc961b (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 1c5145d993ebd9a19131b336482dfa58efdc961b # timeout=10
Commit message: "Update jenkinsfile"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { list_files
[Pipeline] sh
+ ls
1
file1
jenkinsfile
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

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

```
8 lines (7 sloc) | 195 Bytes
1 node {
2
3   stage('list_files') {
4     git_branch: 'check', credentialsId: 'tokenforpipeline', url: 'https://github.com/OmarMahmoud94/New.git'
5     sh 'ls'
6     sh 'cat file1'
7   }
8 }
```

Dashboard **descriptive** #7

Pipeline Steps
Workspaces
Previous Build

```
> git --version # 'git version 2.20.1'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/OmarMahmoud94/New.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/check^{commit} # timeout=10
Checking out Revision 012fa6d152e5a92651c6e5d765c56a99f9ca7cfa (refs/remotes/origin/check)
> git config core.sparsecheckout # timeout=10
> git checkout -f 012fa6d152e5a92651c6e5d765c56a99f9ca7cfa # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git checkout -b check 012fa6d152e5a92651c6e5d765c56a99f9ca7cfa # timeout=10
Commit message: "Update Jenkinsfile"
First time build. Skipping changelog.
[Pipeline] sh
+ ls
1
file1
Jenkinsfile
[Pipeline] sh
+ cat file1
hello
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

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

Jenkins

Dashboard > multi

Up
Status
Configure
Scan Multibranch Pipeline Now
Scan Multibranch Pipeline Log
Multibranch Pipeline Events
Delete Multibranch Pipeline

multi

Disable Multibranch Pipeline

Branches (2)

Name	Last Success	Last Failure	Last Duration
check	26 sec - #1	N/A	5.6 sec
master	56 sec - #1	N/A	8.1 sec

Icon: S M L

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

localhost:8080/job/multi/job/check/1/console

Apps Automotive jo... IBM AI Engine... Google IT Aut... Cloud Engine... Full-Stack We... Software Testi... Java Program... Reading list

Dashboard > multi > check > #1

```
> git --version # timeout=10
> git --version # 'git version 2.20.1'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/OmarMahmoud94/New.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/OmarMahmoud94/New.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/check^{commit} # timeout=10
Checking out Revision 012fa6d152e5a92651c6e5d765c56a99f9ca7cfa (refs/remotes/origin/check)
> git config core.sparsecheckout # timeout=10
> git checkout -f 012fa6d152e5a92651c6e5d765c56a99f9ca7cfa # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git checkout -b check 012fa6d152e5a92651c6e5d765c56a99f9ca7cfa # timeout=10
Commit message: "Update Jenkinsfile"
First time build. Skipping changelog.
[Pipeline] sh
+ ls
1
file1
Jenkinsfile
[Pipeline] sh
+ cat file1
hello
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

localhost:8080/job/multi/job/master/1/console

Apps Automotive jo... IBM AI Engine... Google IT Aut... Cloud Engine... Full-Stack We... Software Testi... Java Program... Reading list

Dashboard > multi > master > #1

```
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
Checking out Revision 149ca7c24568e0a5e4cc1551ddc905a33b0a00b (master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 149ca7c24568e0a5e4cc1551ddc905a33b0a00b # timeout=10
Commit message: "Rename jenkinsfile to Jenkinsfile"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (list_files)
[Pipeline] sh
+ ls
1
file1
Jenkinsfile
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```