

Jenkins Lab 1

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




Role-based Authorization Strategy
Enables user authorization using a Role-Based strategy. Roles can be defined globally or for particular jobs or nodes selected by regular expressions.
3.2.0

Uninstall

- 3- create new user






Dashboard

Jenkins' own user database

 Create User

Users

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.

User ID	Name	
 admin_aya	admin_aya	
 aya	aya	 

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

Dashboard > Manage and Assign Roles

Manage Roles

Global roles

Role	Overall	Credentials						Agent			
	Administer	Read	Create	Delete	ManageDomains	Update	View	Build	Configure	Connect	Create D
admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
developer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Dashboard > Manage and Assign Roles

Assign Roles

Global roles

User/group	admin	developer
aya kamal	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ayakamal	<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")

Dashboard > aya kamal > My Views > All > my_test

General Source Code Management Build Triggers Build Environment Build

Post-build Actions

☐ None

☒ Git

Repositories

Repository URL

Credentials

Dashboard ▶ aya kamal ▶ My Views ▶ All ▶ my_test ▶

General Source Code Management Build Triggers Build Environment **Build**

Post-build Actions

Build

Execute shell

Command

```
ls
mkdir sprints
cd sprints
touch test
echo "hello World" >> test
```

See [the list of available environment variables](#)

Advanced...

Save

Apply

Dashboard ▶ aya kamal ▶ My Views ▶ All ▶ my_test ▶ #5

Console Output

```
04:02:00 Started by user aya kamal
04:02:00 Running as SYSTEM
04:02:00 Building in workspace /var/jenkins_home/workspace/my_test
04:02:00 [WS-CLEANUP] Deleting project workspace...
04:02:00 [WS-CLEANUP] Deferred wipeout is used...
04:02:00 [WS-CLEANUP] Done
04:02:00 The recommended git tool is: NONE
04:02:00 using credential mygithub
04:02:00 Cloning the remote Git repository
04:02:00 Cloning repository https://github.com/ayakamal/Jenkins_Demo.git
04:02:00 > git init /var/jenkins_home/workspace/my_test # timeout=10
04:02:00 Fetching upstream changes from https://github.com/ayakamal/Jenkins_Demo.git
04:02:00 > git --version # timeout=10
04:02:00 > git --version # 'git version 2.30.2'
04:02:00 using GIT_ASKPASS to set credentials
04:02:00 > git fetch --tags --force --progress -- https://github.com/ayakamal/
/Jenkins_Demo.git +refs/heads/*:refs/remotes/origin/* # timeout=10
04:02:01 > git config remote.origin.url https://github.com/ayakamal/Jenkins_Demo.git
# timeout=10
```

Dashboard ▶ aya kamal ▶ My Views ▶ All ▶ my_test ▶ #5

```
04:02:00 > git fetch --tags --force --progress -- https://github.com/ayakamal/Jenkins_Demo.git +refs/heads/*:refs/remotes/origin/* # timeout=10
04:02:01 > git config remote.origin.url https://github.com/ayakamal/Jenkins_Demo.git # timeout=10
04:02:01 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
04:02:01 Avoid second fetch
04:02:01 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
04:02:01 Checking out Revision 9c7e392694282fe69e1afa389bb59fafe534771d (refs/remotes/origin/master)
04:02:01 > git config core.sparsecheckout # timeout=10
04:02:01 > git checkout -f 9c7e392694282fe69e1afa389bb59fafe534771d # timeout=10
04:02:05 Commit message: "Initial commit"
04:02:05 First time build. Skipping changelog.
04:02:05 [my_test] $ /bin/sh -xe /tmp/jenkins15894399312291541654.sh
04:02:05 + ls
04:02:05 README.md
04:02:05 + mkdir sprints
04:02:05 + cd sprints
04:02:05 + touch test
04:02:05 + echo hello World
04:02:05 Finished: SUCCESS
```

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

Dashboard ▶ aya kamal ▶ My Views ▶ All ▶ first_declarative_pipeline ▶

General Build Triggers Advanced Project Options Pipeline

Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/ayakamal/Jenkins.git

Credentials ?

- none - Add

Save Apply Advanced...

Dashboard > aya kamal > My Views > All > first_declarative_pipeline >

General Build Triggers Advanced Project Options **Pipeline**

*/master

Add Branch

Repository browser ?

(Auto) v

Additional Behaviours

Add v

Script Path ?

Jenkinsfile

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

Dashboard > aya kamal > My Views > All > first_pipeline >

General Build Triggers Advanced Project Options **Pipeline**

Definition

Pipeline script v

Script ?


```

1 node {
2   def mvnHome
3   stage('Preparation') { // for display purposes
4     // Get some code from a GitHub repository
5     git 'https://github.com/ayakamal/Jenkins.git'
6     // Get the Maven tool.
7     // ** NOTE: This 'M3' Maven tool must be configured
8     // **      in the global configuration.
9     mvnHome = tool 'test'
10  }
11  stage('Build') {
12    // Run the maven build
13    withEnv(["MVN_HOME=$mvnHome"]) {
14      if (isUnix()) {
15        sh """ ls
16          mkdir scripted_dir
17          touch scripted_file
18          """
19      } else {

```

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

[Dashboard](#) > [first_multibranch_pipeline](#) > [Scan Multibranch Pipeline](#)




Scan Multibranch Pipeline Log

Started

[Sun Oct 10 08:16:52 UTC 2021] Starting branch indexing...

```
> git --version # timeout=10
> git --version # 'git version 2.30.2'
> git ls-remote --symref -- https://github.com/ayakamal/Jenkins.git # timeout=10
> git rev-parse --resolve-git-dir /var/jenkins_home/caches/git-dca832d4c971e5cf551dfa88f98600/.git # timeout=10
Setting origin to https://github.com/ayakamal/Jenkins.git
> git config remote.origin.url https://github.com/ayakamal/Jenkins.git # timeout=10
Fetching & pruning origin...
Listing remote references...
> git config --get remote.origin.url # timeout=10
> git --version # timeout=10
> git --version # 'git version 2.30.2'
> git ls-remote -h -- https://github.com/ayakamal/Jenkins.git # timeout=10
Fetching upstream changes from origin
> git config --get remote.origin.url # timeout=10
> git fetch --tags --force --progress --prune -- origin +refs/heads/*:refs/remotes/origin/* # timeout=10
Checking branches
```




[Dashboard](#) > [first_multibranch_pipeline](#)



first_multibranch_pipeline


[Disable Multibranch Pipeline](#)


Branches (1)


S	W	Name ↓	Last Success	Last Failure	Last Duration
		master	N/A	N/A	N/A 

Icon:
S M L

Legend

 Atom feed for all

 Atom feed for failures

 Atom feed for just latest builds