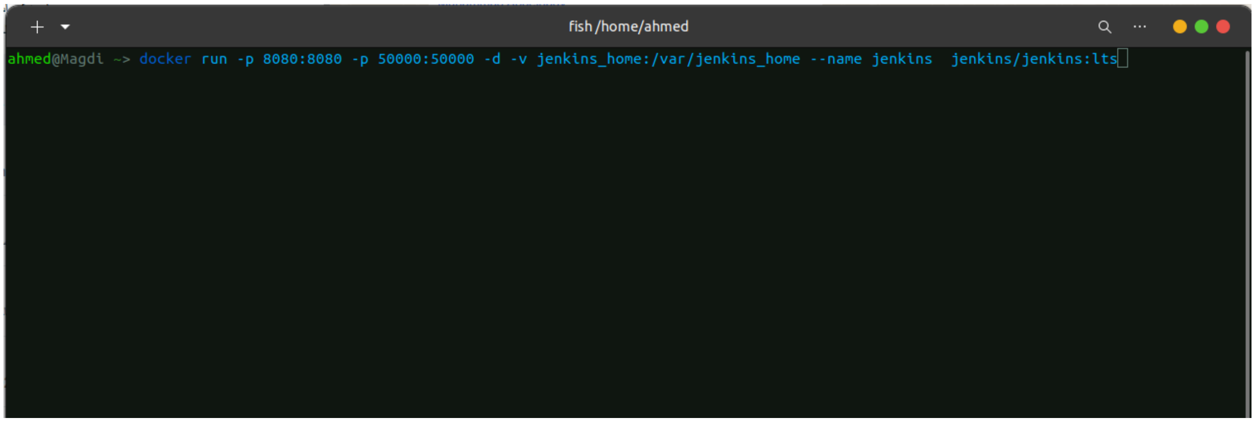
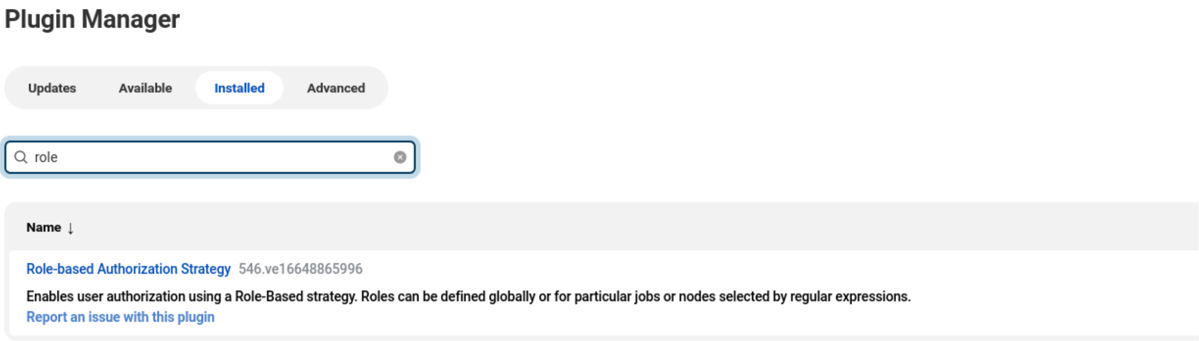
Jenkins

1) **install jenkins with docker image.**



```
ahmed@Magdi ~> docker run -p 8080:8080 -p 50000:50000 -d -v jenkins_home:/var/jenkins_home --name jenkins  jenkins/jenkins:lts
```

2) **install role based authorization plugin.**

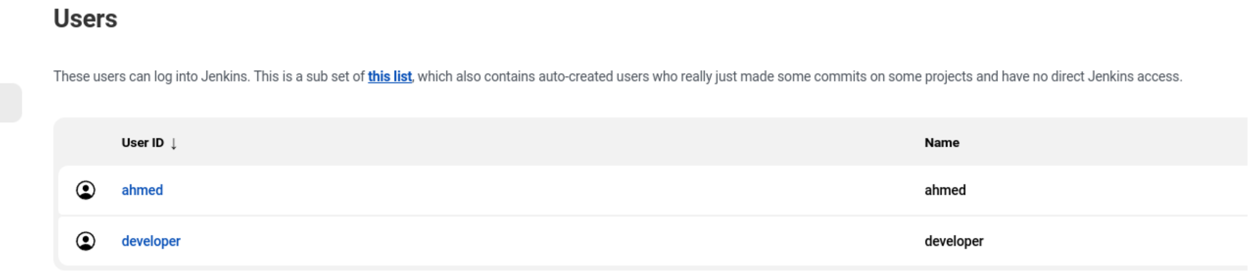


Plugin Manager

role

Name ↓
Role-based Authorization Strategy 546.ve16648865996 Enables user authorization using a Role-Based strategy. Roles can be defined globally or for particular jobs or nodes selected by regular expressions. Report an issue with this plugin

3) **create new user.**

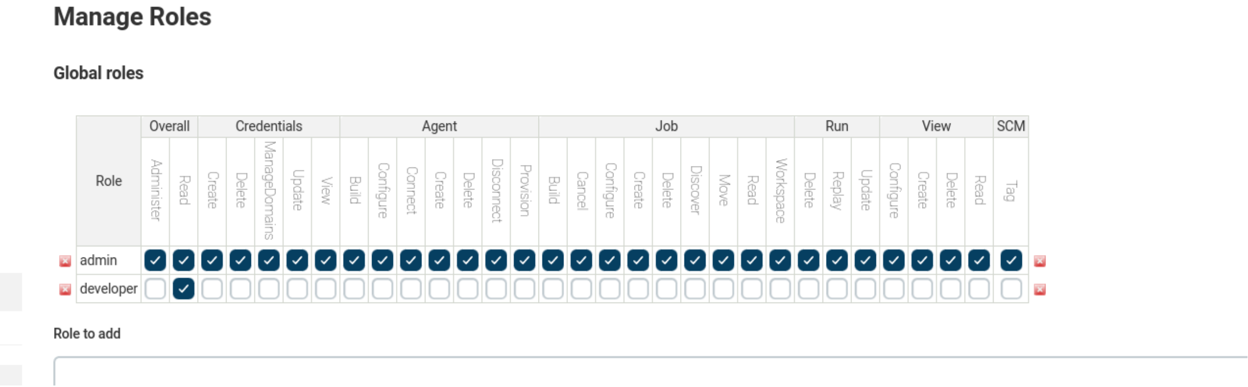


Users

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
	ahmed	ahmed
	developer	developer

4) create read role and assign it to the new user.

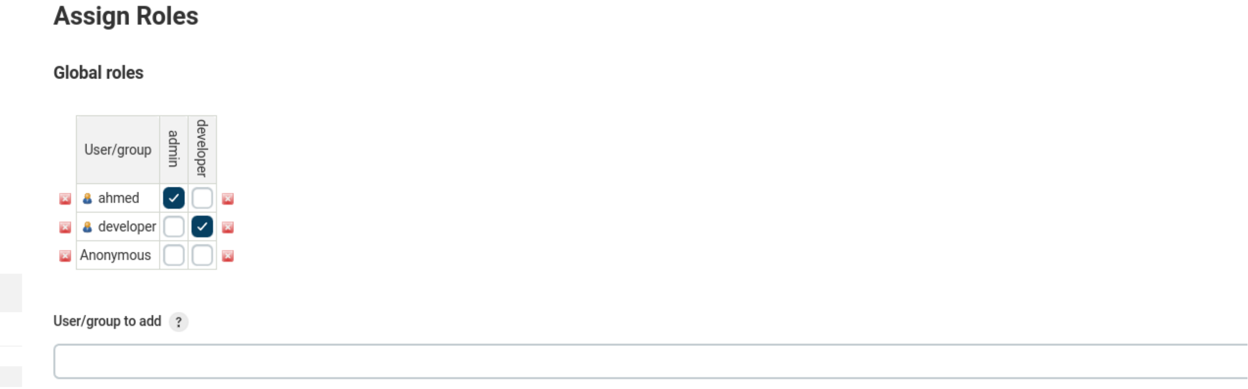


Manage Roles

Global roles

Role	Overall		Credentials					Agent							Job									Run			View				SCM
	Administer	Read	Create	Delete	ManageDomains	Update	View	Build	Configure	Connect	Create	Delete	Disconnect	Provision	Build	Cancel	Configure	Create	Delete	Discover	Move	Read	Workspace	Delete	Replay	Update	Configure	Create	Delete	Read	Tag
admin	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
developer	□	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

Role to add



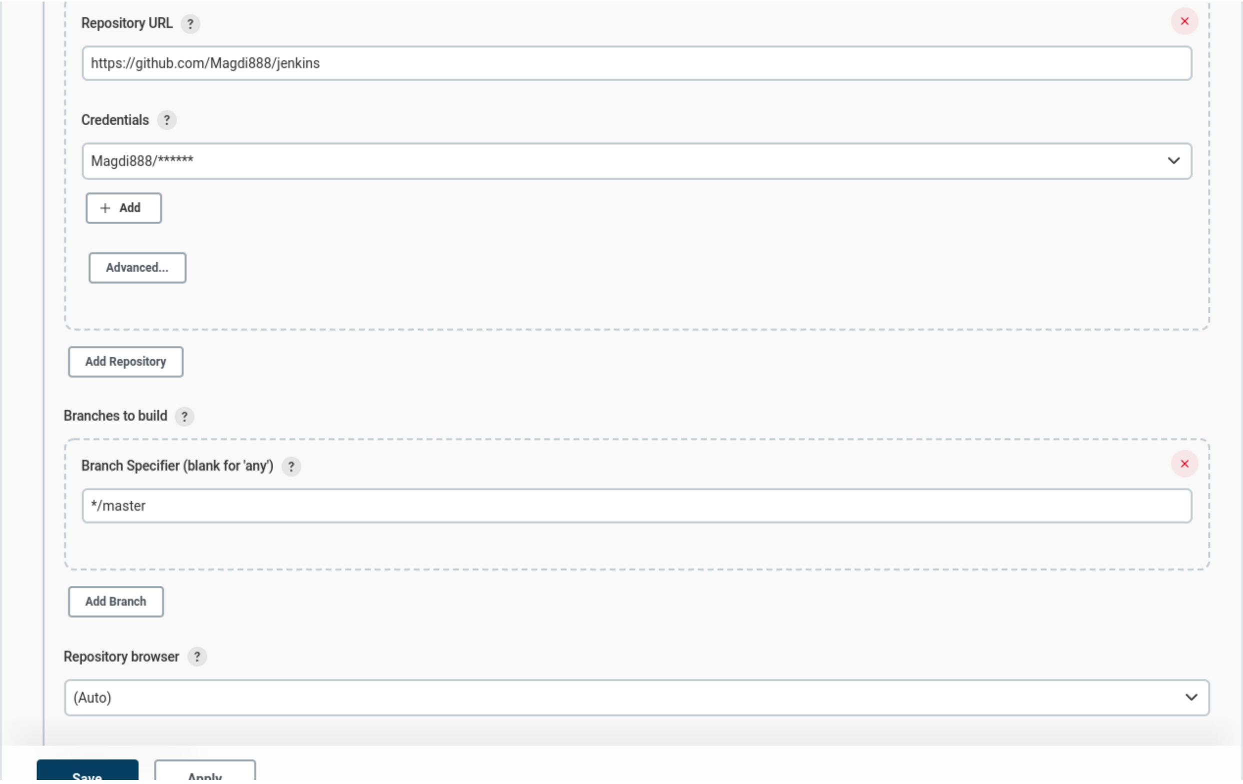
Assign Roles

Global roles

User/group	admin	developer
ahmed	☑	□
developer	□	☑
Anonymous	□	□

User/group to add ?

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


Repository URL ?

https://github.com/Magdi888/jenkins

Credentials ?

Magdi888/******

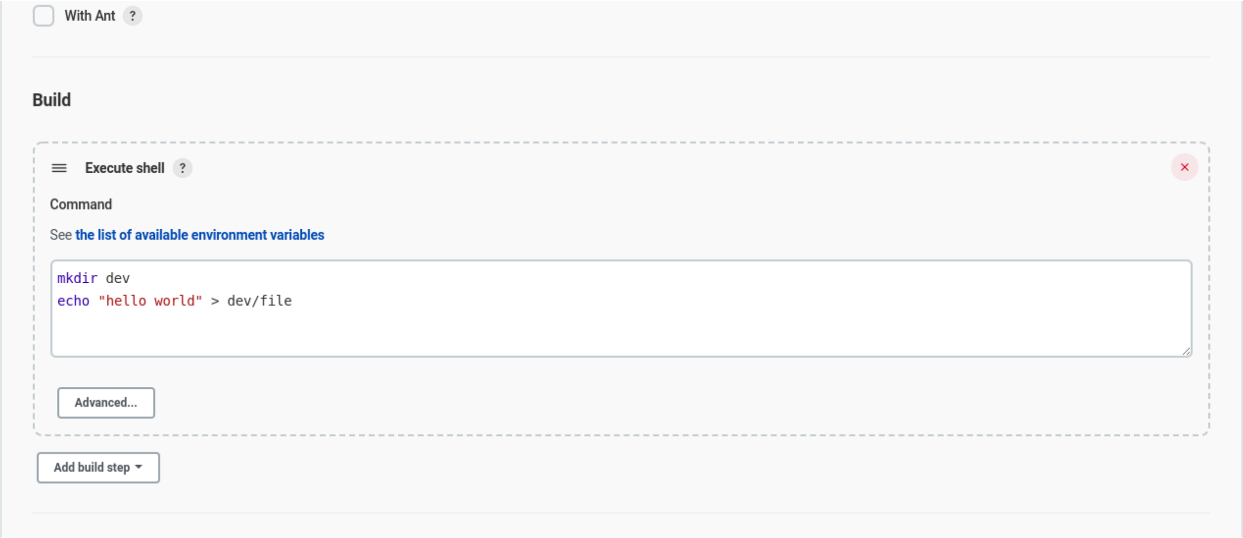
Branches to build ?

Branch Specifier (blank for 'any') ?

*/master

Repository browser ?

(Auto)


□ With Ant ?

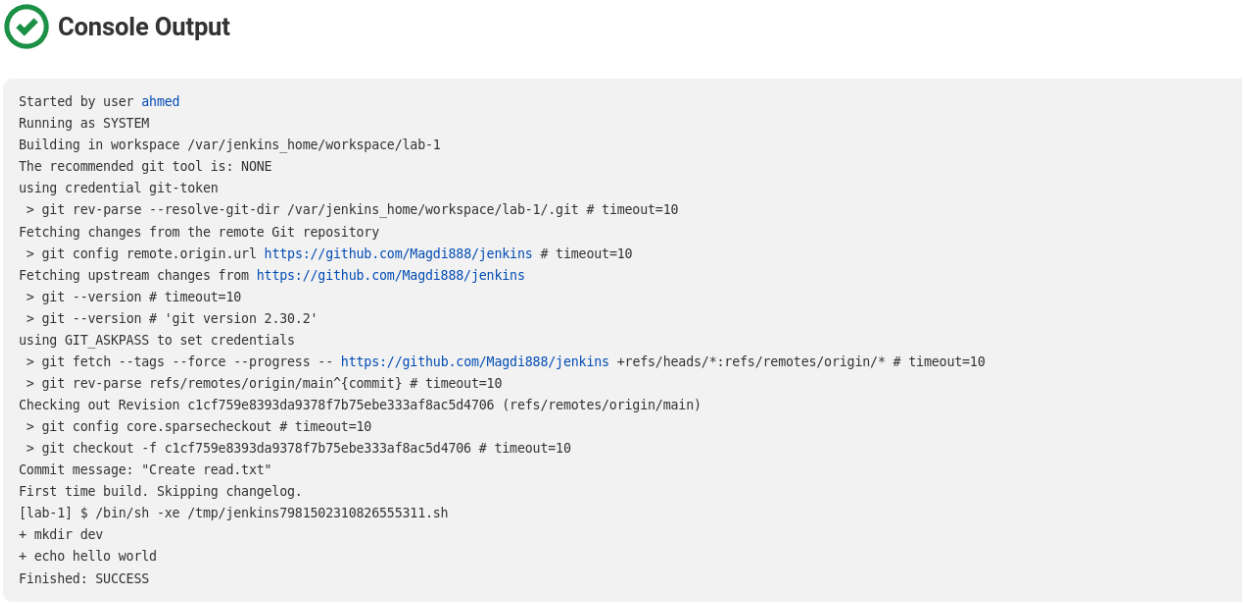
Build

Execute shell ?

Command

See the list of available environment variables

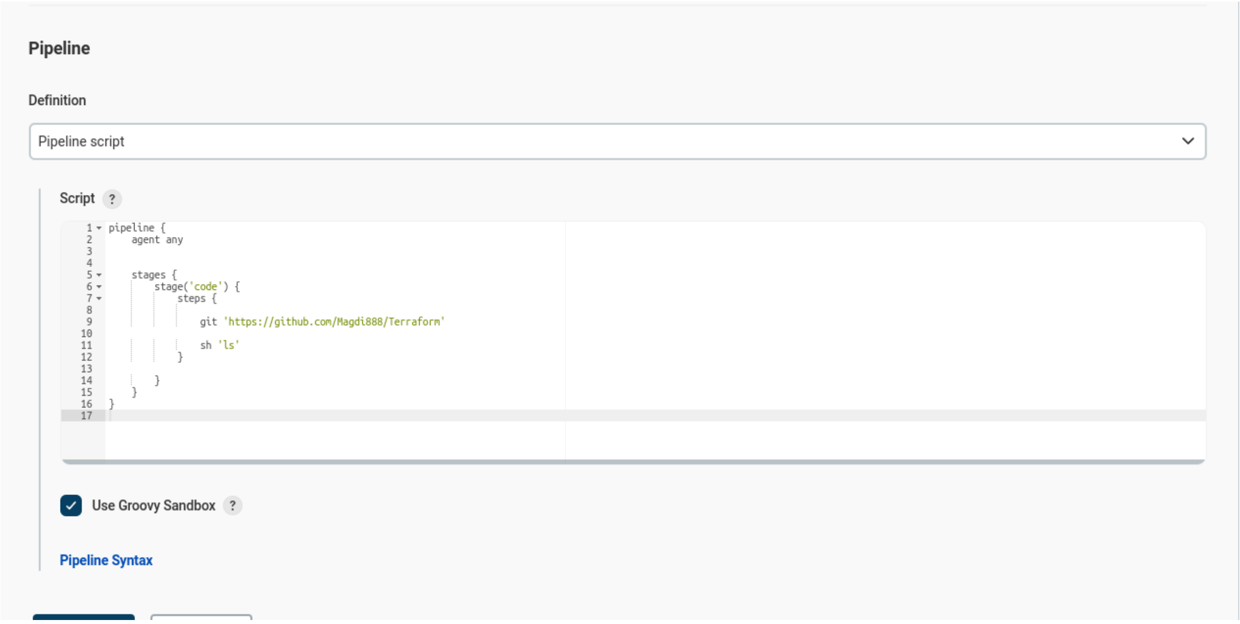
```
mkdir dev
echo "hello world" > dev/file
```



Console Output

```
Started by user ahmed
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/lab-1
The recommended git tool is: NONE
using credential git-token
 > git rev-parse --resolve-git-dir /var/jenkins_home/workspace/lab-1/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/Magdi888/jenkins # timeout=10
Fetching upstream changes from https://github.com/Magdi888/jenkins
 > 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/Magdi888/jenkins +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision c1cf759e8393da9378f7b75ebe333af8ac5d4706 (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f c1cf759e8393da9378f7b75ebe333af8ac5d4706 # timeout=10
Commit message: "Create read.txt"
First time build. Skipping changelog.
[lab-1] $ /bin/sh -xe /tmp/jenkins7981502310826555311.sh
+ mkdir dev
+ echo hello world
Finished: SUCCESS
```

6) **create declarative in jenkins GUI pipeline for your own repo to do "ls".**



Pipeline

Definition

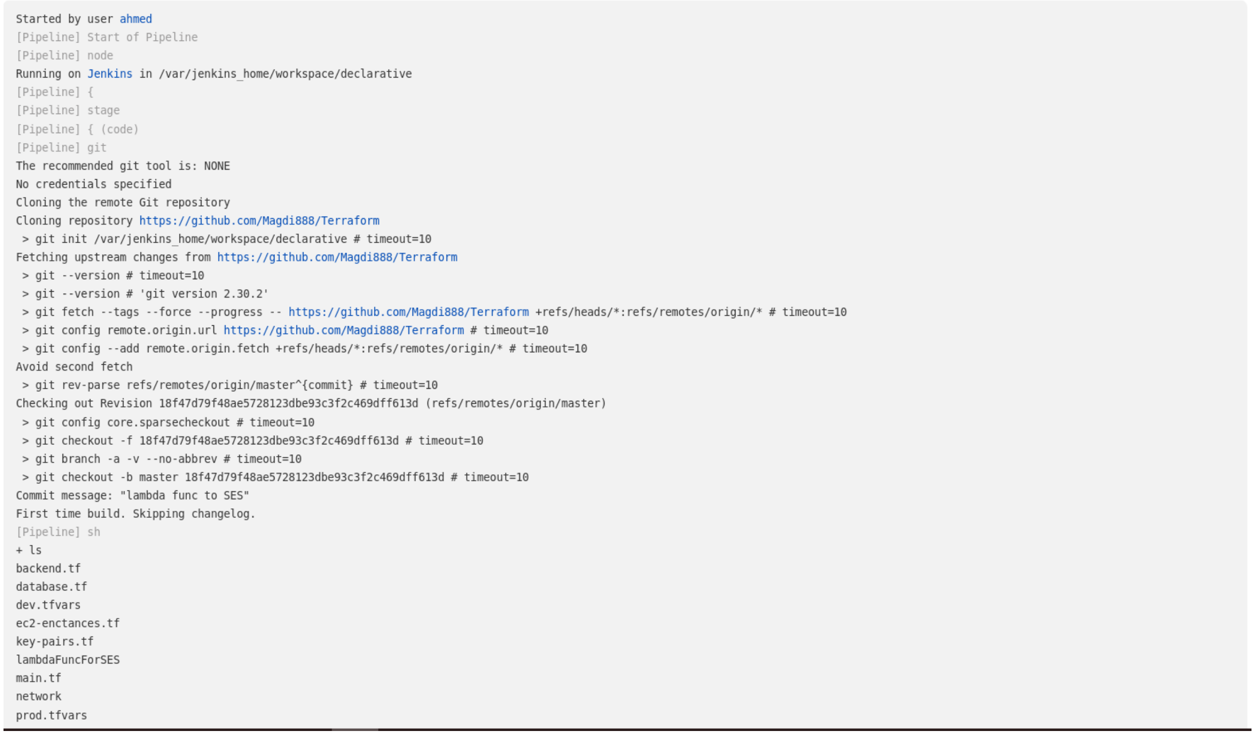
Pipeline script

Script

```
pipeline {
    agent any


    stages {
        stage('code') {
            steps {

                git 'https://github.com/Magdi888/Terraform'

                sh 'ls'
            }

        }
    }
}

```

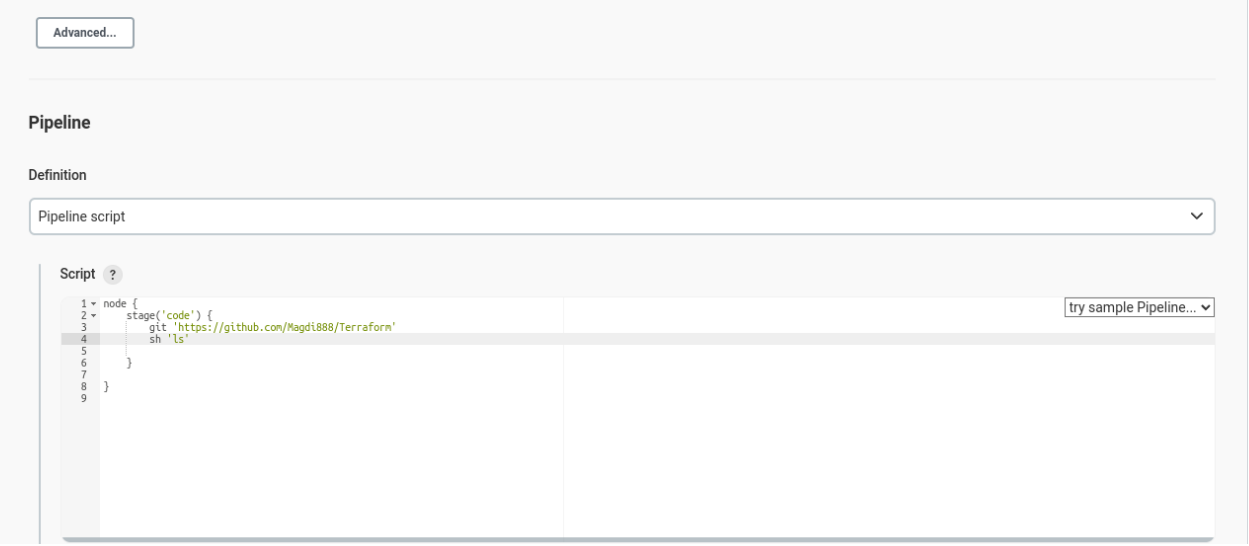
☑ Use Groovy Sandbox

Pipeline Syntax



```
Started by user ahmed
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/declarative
[Pipeline] {
[Pipeline] stage
[Pipeline] { (code)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Magdi888/Terraform
 > git init /var/jenkins_home/workspace/declarative # timeout=10
Fetching upstream changes from https://github.com/Magdi888/Terraform
 > git --version # timeout=10
 > git --version # 'git version 2.30.2'
 > git fetch --tags --force --progress -- https://github.com/Magdi888/Terraform +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/Magdi888/Terraform # 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 18f47d79f48ae5728123dbe93c3f2c469dff613d (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 18f47d79f48ae5728123dbe93c3f2c469dff613d # timeout=10
 > git branch -a -v --no-abbrev # timeout=10
 > git checkout -b master 18f47d79f48ae5728123dbe93c3f2c469dff613d # timeout=10
Commit message: "lambda func to SES"
First time build. Skipping changelog.
[Pipeline] sh
+ ls
backend.tf
database.tf
dev.tfvars
ec2-enctances.tf
key-pairs.tf
lambdaFuncForSES
main.tf
network
prod.tfvars
```

7) **create scripted in jenkins GUI pipeline for your own repo to do "ls".**



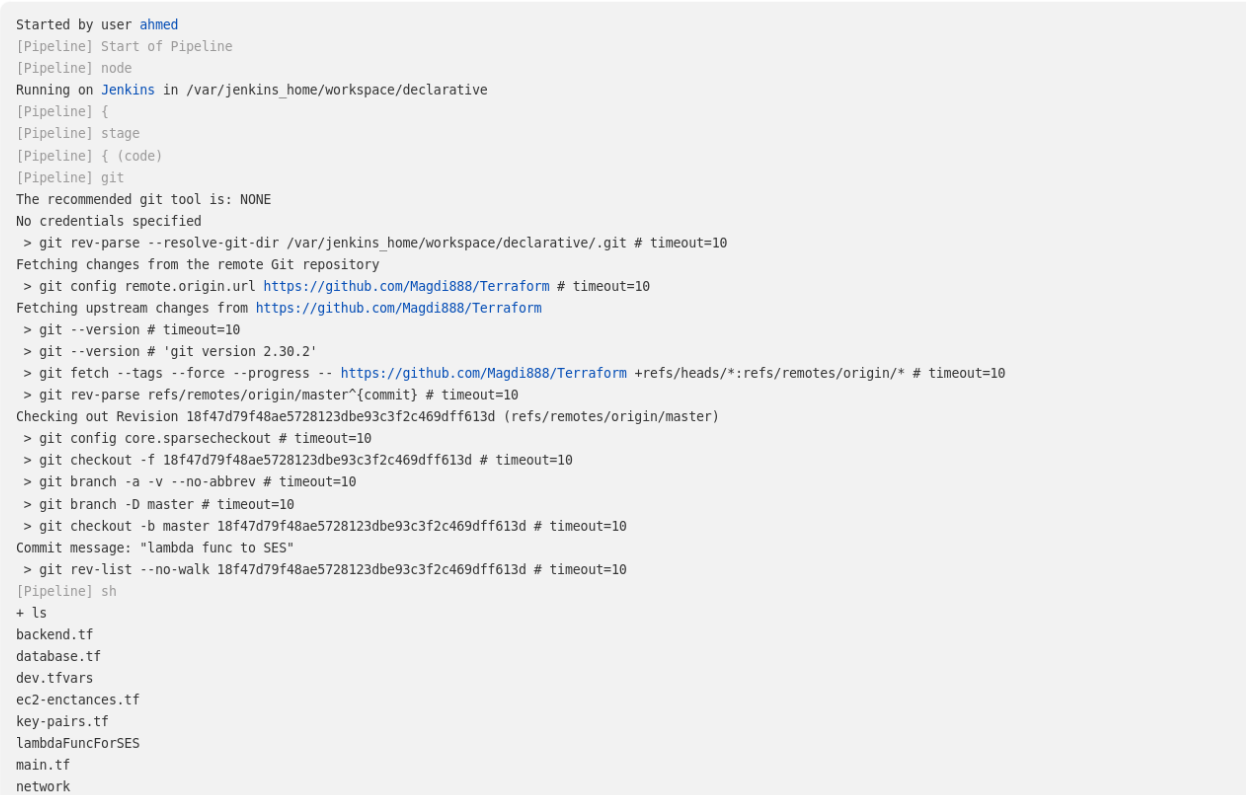
Pipeline

Definition

Pipeline script

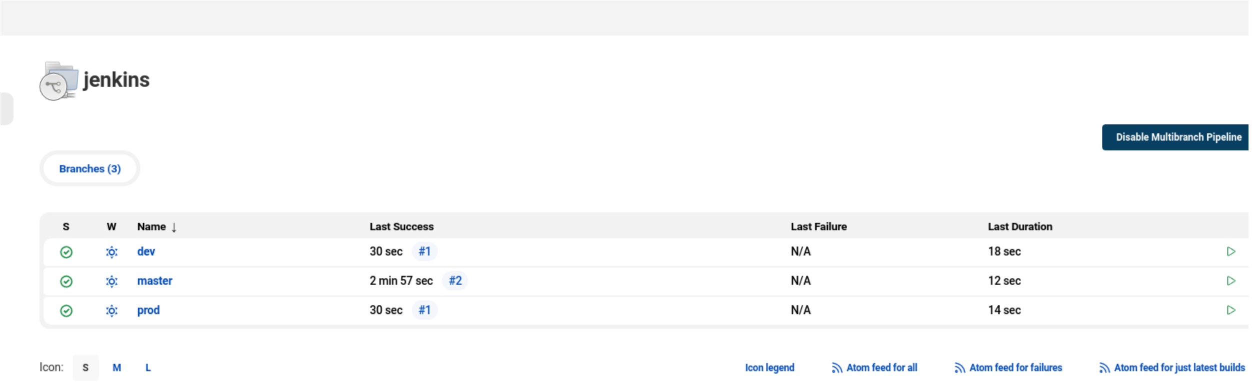
Script

```
node {
    stage('code') {
        git 'https://github.com/Magdi888/Terraform'
        sh 'ls'

    }

}
```



```
Started by user ahmed
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/declarative
[Pipeline] {
[Pipeline] stage
[Pipeline] { (code)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/jenkins_home/workspace/declarative/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/Magdi888/Terraform # timeout=10
Fetching upstream changes from https://github.com/Magdi888/Terraform
 > git --version # timeout=10
 > git --version # 'git version 2.30.2'
 > git fetch --tags --force --progress -- https://github.com/Magdi888/Terraform +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 18f47d79f48ae5728123dbe93c3f2c469dff613d (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 18f47d79f48ae5728123dbe93c3f2c469dff613d # timeout=10
 > git branch -a -v --no-abbrev # timeout=10
 > git branch -D master # timeout=10
 > git checkout -b master 18f47d79f48ae5728123dbe93c3f2c469dff613d # timeout=10
Commit message: "lambda func to SES"
 > git rev-list --no-walk 18f47d79f48ae5728123dbe93c3f2c469dff613d # timeout=10
[Pipeline] sh
+ ls
backend.tf
database.tf
dev.tfvars
ec2-enctances.tf
key-pairs.tf
lambdaFuncForSES
main.tf
network
```

8)Multibranch



S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☼	dev	30 sec #1	N/A	18 sec	▷
✓	☼	master	2 min 57 sec #2	N/A	12 sec	▷
✓	☼	prod	30 sec #1	N/A	14 sec	▷