JENKINS ASSIGNMENT

```
master@mastervm:~/Desktop$ cd ..
master@mastervm:~$ git --version
Command 'git' not found, but can be installed with:
sudo apt install git
master@mastervm:~$ sudo apt update
[sudo] password for master:
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:2 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:4 http://in.archive.ubuntu.com/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:6 http://in.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:7 http://in.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
275 packages can be upgraded. Run 'apt list --upgradable' to see them.
master@mastervm:~$ sudo apt install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm17t64
```

```
master@mastervm:~$ git --version
git version 2.43.0
master@mastervm:~$ sudo apt install maven -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm17t64
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libaopalliance-java libapache-pom-java libatinject-jsr330-api-java
  libcdi-api-java libcommons-cli-java libcommons-io-java libcommons-lang3-java
  libcommons-parent-java liberror-prone-java
  libgeronimo-annotation-1.3-spec-java libgeronimo-interceptor-3.0-spec-java
  libguava-java libguice-java libjansi-java libjsr305-java
  libmaven-parent-java libmaven-resolver-java libmaven-shared-utils-java
  libmaven3-core-java libplexus-cipher-java libplexus-classworlds-java
  libplexus-component-annotations-java libplexus-interpolation-java
  libplexus-sec-dispatcher-java libplexus-utils2-java libsisu-inject-java
  libsisu-plexus-java libslf4j-java libwagon-file-java
```

File   Edit   View   History   Bookmarks   Tools   Help

groovy-pipeline Config [J ×   Jenkins Pipeline Optimiz ×   +

192.168.203.128:8080/job/groovy-pipeline/configure

Dashboard > groovy-pipeline > Configuration

Configure

- ⚙ General
- ⏱ Triggers
- 🎋 Pipeline
- 🔧 Advanced

☐ Throttle builds ?

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

☐ Build after other projects are built ?

☐ Build periodically ?

☑ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

☐ Trigger builds remotely (e.g., from scripts) ?

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

[Save]   [Apply]

File   Edit   View   History   Bookmarks   Tools   Help

groovy-pipeline Config [J ×   Jenkins Pipeline Optimiz ×   +

192.168.203.128:8080/job/groovy-pipeline/configure

Dashboard > groovy-pipeline > Configuration

Configure

- ⚙ General
- ⏱ Triggers
- 🎋 Pipeline
- 🔧 Advanced

Definition

Pipeline script   ⌄

Script ?

```
 1  pipeline {
 2      agent any
 3
 4      tools {
 5          maven 'Maven 3.8.4'
 6      }
 7
 8      stages {
 9          stage('Checkout') {
10              steps {
11                  git credentialsId: 'KIREETI1234', url: 'https://github.com/KIREETI1234/helloworld.git'
12              }
13          }
14
15          stage('Build') {
16              steps {
17                  sh 'mvn clean package'
18              }
19
```

try sample Pipeline... ⌄

☑ Use Groovy Sandbox ?

[Save]   [Apply]

groovy-pipeline Config [J ✕   Jenkins Pipeline Optimiz ✕   +

192.168.203.128:8080/job/groovy-pipeline/configure

Dashboard > groovy-pipeline > Configuration

Configure

- ⚙ General
- ⏱ Triggers
- ⊓ Pipeline
- 🔧 Advanced

```
21      stage('Test') {
22          steps {
23              sh 'mvn test'
24          }
25      }
26
27      stage('Deploy') {
28          steps {
29              script {
30                  try {
31                      sh 'kubectl apply -f deployment.yaml'
32                  } catch (Exception e) {
33                      error "Deployment failed: ${e.message}"
34                  }
35              }
36          }
37      }
38  }
```

try sample Pipeline... ▾

☑ Use Groovy Sandbox ?

Pipeline Syntax

Advanced

Advanced ▾

[Save] [Apply]

Git error troubleshooting ✕   second-pipeline [Jenkins] ✕   +

192.168.203.128:8080/job/second-pipeline/

Dashboard > second-pipeline >

- </> Changes
- ▷ Build Now
- ⚙ Configure
- 🗑 Delete Pipeline
- 🔍 Full Stage View
- ☆ Favorite
- ◯ Open Blue Ocean
- ⊗ Stages
- ✎ Rename
- ? Pipeline Syntax

Builds ⋯ ⟲

🔍 Filter /

Today

Clone the git repo

Stage View

	Declarative: Checkout SCM	Clone Repository	Build
Average stage times: (Full run time: ~15s)	2s	1s	637ms
#5 15:10 No Changes	2s	1s	637ms
#4 15:06 No Changes			
#3 15:05 No Changes			
#2 13:05 No Changes			

Changes

Console Output

Edit Build Information

Delete build '#5'

Timings

Git Build Data

Open Blue Ocean

Pipeline Overview

Pipeline Console

Restart from Stage

Replay

Pipeline Steps

Workspaces

← Previous Build

```
Started by user admin
Obtained Jenkinsfile from git https://github.com/KIREETI1234/jenkins-repo.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/second-pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential github-credentials
Cloning the remote Git repository
Cloning repository https://github.com/KIREETI1234/jenkins-repo.git
 > git init /var/lib/jenkins/workspace/second-pipeline # timeout=10
Fetching upstream changes from https://github.com/KIREETI1234/jenkins-repo.git
 > git --version # timeout=10
 > git --version # 'git version 2.43.0'
using GIT_ASKPASS to set credentials
 > git fetch --tags --force --progress -- https://github.com/KIREETI1234/jenkins-repo.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/KIREETI1234/jenkins-repo.git # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
```

```
[Pipeline] { (Clone Repository)
[Pipeline] sh
+ git clone https://github.com/KIREETI1234/jenkins-repo.git
Cloning into 'jenkins-repo'...
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] sh
+ echo build successful
build successful
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```