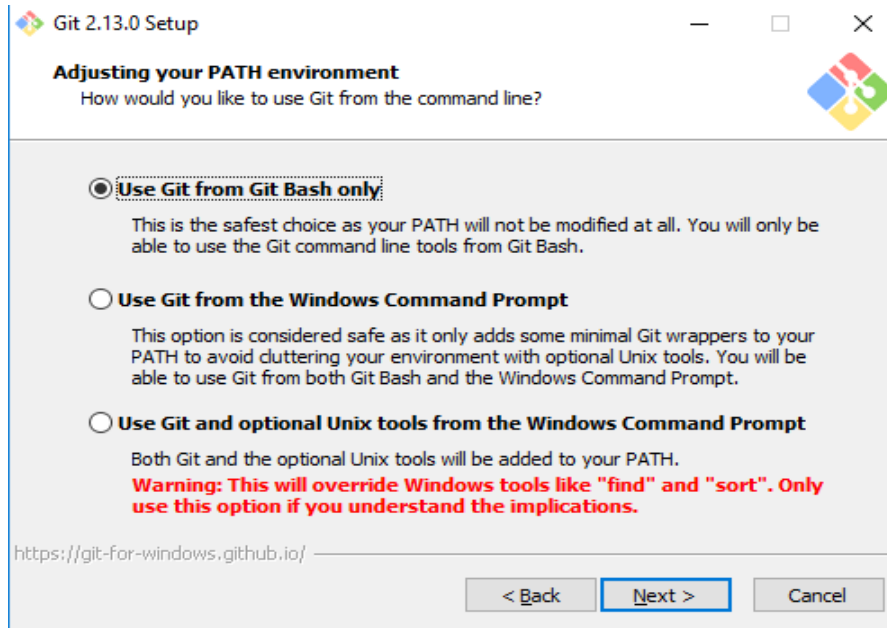
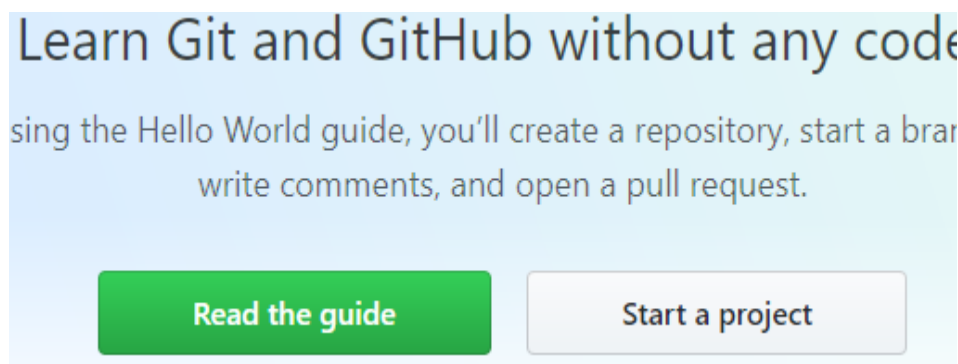


STEPS TO INSTALL AND CONFIGURE GIT

- Install Git.exe by click on Next → Next and select below radio button



- And finally click on Install
- Open <https://github.com/> link from browser
- Create an account with your details and verify E-Mail address
- Click on Start Project



- Give a Repository name as your wish and click on create repository

Owner: Dileep1811 / Repository name: Repository_1 ✓

Great repository names are short and memorable. Need inspiration? How about [miniature-octo-potato](#).

Description (optional)

☒ Public
Anyone can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

☐ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None ⓘ

Create repository

➤ Then it shows as below. Copy that URL

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/Dileep1811/Repository_1.git

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

- Goto desktop, Create a folder as GitPractice
- Open that folder and right click select **Git bash here** then it opens Git command Prompt
- From there we have to create a directory by the following commands

- **mkdir project1**
- **cd project1**

```
Dileep Chakravarthi@DESKTOP-4PE1TGE MINGW64 /f/Devops/Repository_git
$ mkdir project1

Dileep Chakravarthi@DESKTOP-4PE1TGE MINGW64 /f/Devops/Repository_git
$ cd project1
```

- Then we have to checkout the Repository with the below command

- **git clone** https://github.com/Dileep1811/Repository_1.git
(this is repository URL, copied from the GIT Hub)

```
Dileep Chakravarthi@DESKTOP-4PE1TGE MINGW64 /f/Devops/Repository_git/project1
$ git clone https://github.com/Dileep1811/Repository_1.git
Cloning into 'Repository_1'...
warning: You appear to have cloned an empty repository.
```

- **cd Repository_1** (This is your repository name)
- We have to Copy **Src** folder and **build.xml** file manually into this repository from our local machine
- Open Git command prompt and check the status
 - **git status**

```
Dileep Chakravarthi@DESKTOP-4PE1TGE MINGW64 /f/Devops/Repository_git/project1/Repository_1 (master)
$ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        build.xml
        src/

nothing added to commit but untracked files present (use "git add" to track)
```

- **git add Src/ build.xml**

```
Dileep Chakravarthi@DESKTOP-4PE1TGE MINGW64 /f/Devops/Repository_git/project1/Repository_1 (master)
$ git add src/ build.xml
```

- **git commit -m "message"**

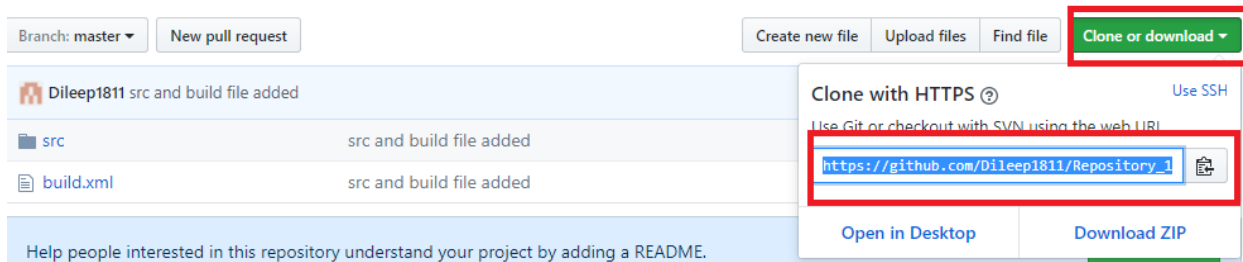
```
Dileep Chakravarthi@DESKTOP-4PE1TGE MINGW64 /f/Devops/Repository_git/project1/Repository_1 (master)
$ git commit -m"src and build file added"
[master (root-commit) 37dab3a] src and build file added
2 files changed, 86 insertions(+)
create mode 100644 build.xml
create mode 100644 src/EnhancedFor.java
```

- Push the code to server
 - **git push origin master**

```
Dileep Chakravarthi@DESKTOP-4PE1TGE MINGW64 /f/Devops/Repository_git/project1/Repository_1 (master)
$ git push origin master
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 1.06 KiB | 0 bytes/s, done.
Total 5 (delta 0), reused 0 (delta 0)
To https://github.com/Dileep1811/Repository_1.git
 * [new branch]      master -> master
```

- Then Refresh the browser, now our src and build.xml files reflects into the server

- Click on **clone or Download** and copy the **URL**



- Open **Jenkins from command prompt** (Jenkins from windows)

- Run Jenkins.war from the path where the war file is located on cmd
 - **Java -jar Jenkins.war**

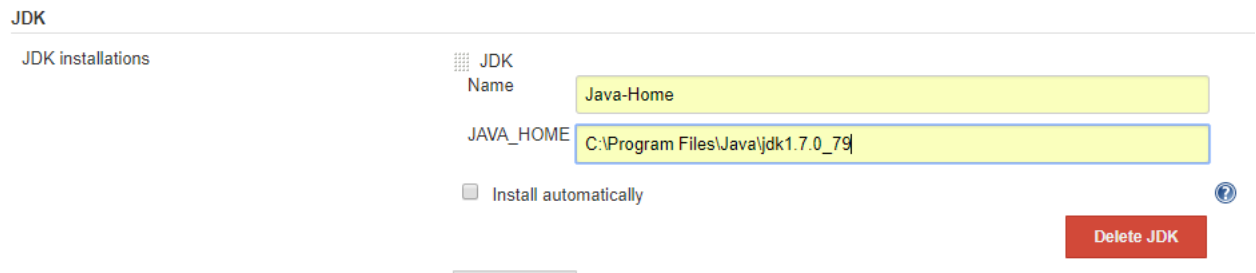
- Open **localhost:8080** from browser

- Install GIT Plugin

→Click on Manage Jenkins→Manage Plugins→Available then Search from GIT Plugin and install

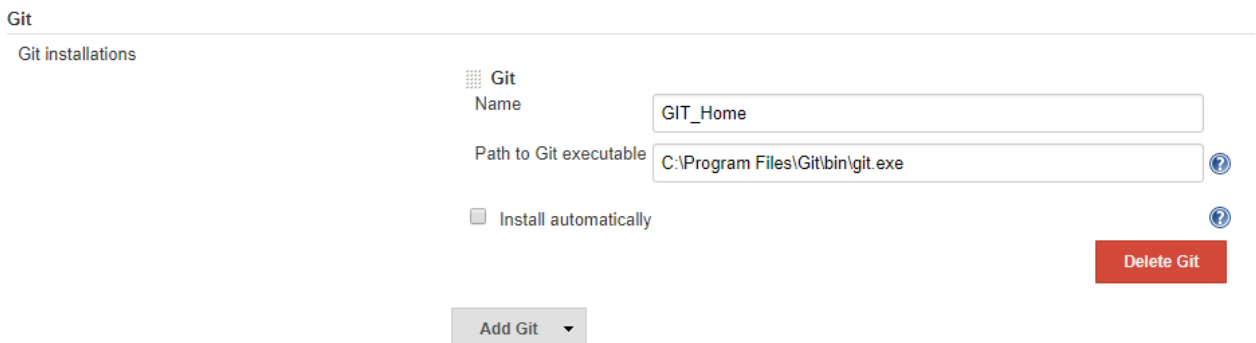
- Configure java path to jenkins

→Click on Manage Jenkins→Configure System→Add JDK (copy java path without bin from C: drive Program files)



- Configure GIT path

→Click on ADD GIT (Copy GIT path with bin from C: drive program files)



- Configure ANT Path to Jenkins
 - ➔ Click on ADD ANT (Copy Ant installation path without bin from ur local machine) and save

Ant

Ant installations

Ant Name: ANT_Home

ANT_HOME: F:\SOFTWARE\DEVOPS\ANT Installation\apache-ant-1.9.7-bin\apache-ant-1.9

☐ Install automatically

Delete Ant

Add Ant

List of Ant installations on this system

- Create a new job
 - ➔ Click on new item and then ok

New Item

People

Build History

Manage Jenkins

Jenkins 100%

Item name: Git_Sample_job

☒ **Freestyle project**
This is the central feature of Jenkins other than software build.

☐ **Maven project**
Build a maven project. Jenkins takes

- Then select Git radio button and pass the repository URL copied from GITHUB

Source Code Management

☐ None

☐ CVS

☐ CVS Projectset

☒ **Git**

Repositories

Repository URL: https://github.com/Dileep1811/Repository_1.git

Credentials: - none - **Add**

- Click on Add build step➔Invoke Ant

Build

Add build step

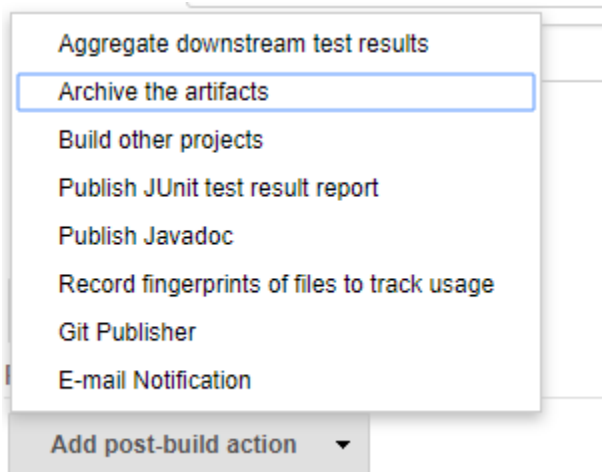
- Execute Windows batch command
- Execute shell
- Invoke Ant**
- Invoke top-level Maven targets

Invoke Ant

Ant Version

Targets

- Click in Add post build action → Archive the Artifacts



Post-build Actions

Archive the artifacts

Files to archive

- Finally save and Build the job

