
List of *Git* Repositories

On

Ubuntu 20.04

CONFIGURATION AND LOCATION DETAILS



[HTTP://WWW.MAADIS.IE/](http://www.maadis.ie/)
DAMAR@MAADIS.IE

Contents

This article is to document all available *Git* repositories in the Ubuntu Server (IP address 192.168.0.136).

1 Introduction

The Ubuntu <i>Git</i> Repository Server	
Server name	server.mijas.ie
IP Address	192.168.0.136
Operating Systems & version	Ubuntu 20.04.2 LTS
RAM Size	8GB
Processors	Intel Core i7-7700 CPU @ 3.60GHz x 8
<i>Git</i> Repository Base Directory	/home/git/projects/git/
<i>Git</i> user	git ¹

In the above:

1. The `/home/git/projects/git/` directory, in the *Ubuntu Git* repository server, is where I will place all *Git* projects. This directory will have further subdirectories—such as SystemAdmin, Programming, Books, etc.—to divide projects into related groups. All these projects are *bare Git* projects meaning that we cannot edit project contents in these projects.
2. The *git* user is the one all remote users will login as to clone, push or pull these projects.

2 The *Git* Projects

The *Git* project URL for cloning, pushing or pulling will be in the form of:

`git@192.168.0.136:projectLocation`

Hence, the absolute URL for *javaBook.git* project should be:

`git@192.168.0.136:/home/git/projects/git/Books/javaBook.git`

The `~/` symbol (user home), at the start of *Project location* in table ??, represents `home/git` directory.

To clone a git project, say `javaBook.git` project, in a **remote** client, say Solus 4.2, run the followings:

```
1 :~$ cd ~/Documents/git/Books
2 :~$ git clone git@192.168.0.136:/home/git/projects/git/Books/javaBook.git javaBook.git
```

Listing 1: Cloning *javaBook.git Git* project in a remote client

The above statement will clone the *javaBook.git* project at `~/Documents/git/Books` directory of the client machine (in which the statements have been executed).

Likewise, to clone the same *Git* project **locally** (inside the Ubuntu *Git* repository server)—cloning

from `/home/git/projects/git/Books/javaBook.git` to `/home/damar/Documents/git/Books/javaBook.git`—run the followings:

```
1 :~!\$! cd !\sim!/Documents/git/Books
2 :~!\$! git clone /home/git/projects/git/Books/javaBook.git javaBook.git
```

The above statement will clone the *javaBook.git* project at `~/Documents/git/Books` directory.

Table 1: Available *Git* Projects (Summary)

Project name	Project location	Project description
javaBook.git	\cong /Books/javaBook.git	Java book Project
latexBook.git	\cong /Books/latexBook.git	L ^A T _E X book
MyNotesCISSP.git	\cong /Books/MyNotesCISSP.git	CISSP notes
.....
rust.git	\cong /rust.git	Rust Programming Notes
.....
GnuPG.git	\cong /agalo/GnuPG.git	GnuPG notes
keys.git	\cong /agalo/keys.git	Keys notes
pass.git	\cong /agalo/pass.git	pass notes
passwdMngmt.git	\cong /agalo/passwdMngmt.git	passwdMngmt notes
.....
git.git	\cong /SystemAdmin/git.git	Git Tutorials and Setup Notes
vim.git	\cong /SystemAdmin/vim.git	Vim notes and tutorials
macbookPro2011.git	\cong /SystemAdmin/macbookPro2011.git	MacBook 2011 Fix
gitProjAdmin.git	\cong /SystemAdmin/gitProjAdmin.git	Temp. Doc. 4 Admin
virtualbox.git	\cong /SystemAdmin/virtualbox.git	Virtualbox notes
linux.git	\cong /SystemAdmin/linux.git	Linux Admin/notes
passwdMngmt.git	\cong /SystemAdmin/passwdMngmt.git	Password Management
docker.git	\cong /SystemAdmin/docker.git	Docker Notes
GNUCash.git	\cong /SystemAdmin/GNUCash.git	GNUCash notes
OpenLDAP.git	\cong /SystemAdmin/OpenLDAP	OpenLDAP notes
tmux.git	\cong /SystemAdmin/tmux.git	Tmux notes
kvm.git	\cong /SystemAdmin/kvm.git	kvm VM notes

In the above, please replace \cong (congruence symbol) with `/home/git/projects/git`. For example, \cong /Books/javaBook.git should end up as `/home/git/projects/git/Books/javaBook.git`.

3 The “*javaBook.git*” project

3.1 Introduction

It is a Java Book project.

3.2 Setting up *javaBook.git* bare project in Ubuntu *Git* Repository Server

The followings steps were taken to create a bare *Git* project:

```
1  :~$ su git
2  :~$ cd
3  :~$ cd projects/git
4  :~$ mkdir Books
5  :~$ cd Books
6  :~$ mkdir javaBook.git
7  :~$ cd javaBook.git
8  :~$ git init --bare
```

Listing 2: Setting up *javaBook.git* bare project

The full path of *javaBook.git* project, therefore, is `/home/git/projects/git/Books/javaBook.git`.

3.3 Initial *javaBook.git* Project Setup in *Solus 4.2* Virtualbox host

The following steps were taken:

1. The `JavaProgrammingLanguage` directory of `~/Documents/latex/Books/` directory is copied to `~/Documents/git/Books/` directory, and renamed it to `javaBook.git`. Further details are as follows:
 - a) The `JavaProgrammingLanguage`, now *javaBook.git*, directory has all files and directories I so far have managed to complete for this book project.
 - b) All projects from `~/Documents/latex/` directory will be moved to `~/Documents/git/` directory; hence, the latter will be the *Git* project base (the directory where all *Git* projects will reside).
 - c) Since I have worked, and will be working, as user *damar*, the above ‘~/’ symbol (user home directory) represents `/home/damar/` directory.

The `~/Documents/git` directory eventually will be the working directory for all *Git* projects (when working in *Solus 4.2* VirtualBox).

2. The following steps were taken to initialise `~/Documents/git/Books/javaBook.git/` directory as a *Git* project:

```
1  :~$ cd ~/Documents/git/Books/javaBook.git
2  :~$ git init
```

Listing 3: Setting up `~/Documents/git/Books/javaBook.git/` directory as a *Git* project

3. At the root (`~/Documents/git/Books/javaBook.git`) of the *javaBook.git* project, create `.gitignore` file with the following content:

```
1  out/
2  OnPipeLine/
3  *.aux
4  *.fdb_latexmk
5  *.fls
```

```
6 *.out
7 *.pdf
8 *.toc
```

Listing 4: The content of .gitignore file for the *javaBook.git* project

4. Add all content and perform the initial commit:

```
1 :~$ git add .
2 :~$ git commit -m "Initial commit before pushing to Ubuntu Git repository server"
```

Listing 5: Adding all changed files and directories and committing

3.4 Setting up Connection Configuration to *Ubuntu Git* Repository Server

To setup the connection with the Ubuntu *Git* repository server:

```
1 :~$ cd ~ /Documents/git/Books/javaBook.git
2 :~$ git remote add origin git@192.168.0.136:/home/git/projects/git/Books/javaBook.git
```

Listing 6: Git connection configuration to *javaBook.git* project from remote host

The line 2 statement should have inserted the following entry in `.git/config` file:

```
1 [remote "origin"]
2   url = git@192.168.0.136:/home/git/projects/git/Books/javaBook.git
3   fetch = +refs/heads/*:refs/remotes/origin/*
```

Listing 7: The content of `javaBook.git/.git/config` file for the *javaBook.git* project

If pushing/pulling has to be done from the same server (from one directory to a git repository in another directory in the same server):

```
1 :~$ cd ~ /Documents/git/Books/javaBook.git
2 :~$ git remote add origin /home/git/projects/git/Books/javaBook.git
```

Listing 8: Connection setup directory to directory (same server)

The line 2 statement should have inserted the following entry in `.git/config` file:

```
1 [remote "origin"]
2   url = /home/git/projects/git/Books/javaBook.git
3   fetch = +refs/heads/*:refs/remotes/origin/*
```

Listing 9: The content of `javaBook.git/.git/config` file for the *javaBook.git* project

3.5 Pushing project content (or changes) to the *Ubuntu Git* repository

Once the above [remote “origin”] configuration is setup, for the *javaBook.git* project, the initial commit is pushed to the Ubuntu *Git* repository server with the followings:

```
1 :~$ cd ~/Documents/git/Books/javaBook.git
2 :~$ git push origin master
```

Listing 10: Pushing the project to Ubuntu *Git* repository server's *javaBook.git* project

4 The “*git.git*” project

4.1 Introduction

This project includes notes, tutorial, and setup and configuration details of *Git* projects, *Git* Repository server, and *Git* client hosts.

4.2 Setting up *git.git* bare project in Ubuntu *Git* Repository Server

The followings steps were taken to create a bare *Git* project:

```
1 :~$ su git
2 :~$ cd
3 :~$ cd projects/git
4 :~$ mkdir SystemAdmin
5 :~$ cd SystemAdmin
6 :~$ mkdir git.git
7 :~$ cd git.git
8 :~$ git init --bare
```

Listing 11: Setting up *git.git* bare project

The full path of *git.git* project, therefore, is `/home/git/projects/git/SystemAdmin/git.git`.

4.3 Initial *git.git* Project Setup in *Ubuntu* Server

This project has been prepared in *Ubuntu* Server (same server as the *Git* repository server) at `/home/damar/Documents/git/SystemAdmin/git.git` directory.

Run the followings to initialise the project:

```
1 :~$ cd ~/Documents/git/SystemAdmin/git.git
2 :~$ git init
```

Listing 12: Setting up *git.git* project in *Ubuntu* server

At the root (`/home/damar/Documents/git/SystemAdmin/git.git`) of the *git.git* project, create `.gitignore` file with the following content:

```
1 out/
2 OnPipeLine/
3 *.aux
4 *.fdb_latexmk
5 *.fls
6 *.out
```

```
7 *.pdf
8 *.toc
9 *.sw[lmnop]
```

Listing 13: The content of `.gitignore` file for the `git.git` project

The files and directories listed in `.gitignore` file are not tracked by `Git`.

Add all content and perform the initial commit:

```
1 :~$ cd ~/Documents/git/SystemAdmin/git.git
2 :~$ git add .
3 :~$ git commit -m "Initial commit before pushing to Ubuntu Git repository server"
```

Listing 14: Committing the initial `git.git` project content

4.4 Setting up Connection Configuration to *Ubuntu Git* Repository Server

Since the `Git` project at `/home/damar/Documents/git/SystemAdmin/git.git` directory and the bare `Git` project at `/home/git/projects/git/SystemAdmin/git.git` directory are in the same server (Ubuntu `Git` Repository Server), all necessary `Git` connections have to be made locally.

In other words, the connections have to be between two directories (`/home/damar/Documents/git/SystemAdmin/git.git` and `/home/git/projects/git/SystemAdmin/git.git` directories).

To configure the appropriate connection, run the followings:

```
1 :~$ cd /home/damar/Documents/git/SystemAdmin/git.git
2 :~$ git remote add origin /home/git/projects/git/SystemAdmin/git.git
```

Listing 15: Committing the initial `git.git` project content

The line 2 statement should have inserted the following entry in `git.git/.git/config` file:

```
1 [remote "origin"]
2   url = /home/git/projects/git/SystemAdmin/git.git
3   fetch = +refs/heads/*:refs/remotes/origin/*
```

Listing 16: The content of `git.git/.git/config` file for the `git.git` project

Before we start pushing the project to the repository (at `/home/git/projects/git/SystemAdmin/git.git` directory), the user `damar` (that pushes the project) has to have a full permission on the project directory (this project directory is owned by user `git` not `damar`). To give the require permission, run the followings:

```
1 :~$ cd /home/git/
2 :~$ sudo setfacl -R -m u:damar:rwX projects/
```

Listing 17: Setting Access Control Lists (ACLs) on `/home/git/projects` directory

In the above listing, the statement on line 2 gives a full permission (Read, Write and Execute) to user `damar` on `home/git/projects`, recursively.

The *setfacl* command and newly added subdirectories

Since I have to rerun the statements of listing ?? (above) each time I add a subdirectory in `projects`, possibly, I may have to set it up in `/etc/fstab` file!

4.5 Pushing project content (or changes) to the Ubuntu *Git* repository

Once the processes in listings ?? and ?? are completed, the committed project content (or changes) can be pushed, as *damar*, to the *Git* repository with the following statements:

```
1 :~$ cd ~/Documents/git/SystemAdmin/git.git
2 :~$ git push origin master
```

Listing 18: Pushing the project to Ubuntu *Git* repository server's *git.git* project

5 Some Errors

On pushing *linux.git* project from *Arch Linux* to the main repository (Ubuntu 192.168.0.136), the following error message appeared:

```
! [remote rejected] master -> master (unpacker error)
error: failed to push some refs to
'192.168.0.136:/home/git/projects/git/SystemAdmin/linux.git'
```

Listing 19: Error message when pushing to *linux.git*

On necessary investigation, it was found that the *linux.git* directory was owned by *damar:damar*. The arrangement, however, is as follows:

1. When a project is pushed from a remote host, it pushes as user *git* (and the project repository in ubuntu is owned by *git:git*).
2. When a project is pushed to the repository locally, it is pushed as user *damar*.

The permission to push remotely to *linux.git* project worked fine after running:

```
:~$ cd /home/git/projects/git/SystemAdmin
:~$ sudo chown -R git:git linux.git
```

Listing 20: Error message when pushing to *linux.git*

If the write permission to push repository locally (as user *damar*) is denied, the following should be run:

```
1 :~$ cd /home/git/
2 :~$ sudo setfacl -R -m u:damar:rwx projects/
```

Listing 21: Giving Read, Write and Execute permissions to user damar on projects/ directory, recursively

6 Summary