

≡ Menu

- [Home](#)
- [Free eBook](#)
- [Start Here](#)
- [Contact](#)
- [About](#)

Introduction to GIT on Linux – Install, Create Project, Commit Files

by Ramesh Natarajan on August 17, 2011



GIT is the most versatile distributed version control system.

The way GIT tracks and handles file changes is very efficient and different than how other version control software tracks the changes (including CVS and Subversion).

This article is for those who are new to GIT. This is a jump-start guide that will show you how to install GIT from source, create a new project, commit changes to the GIT repository.

If you are from CVS/SVN background, you are used to the client-server model, where the repository is installed on a server, and you'll use a client to download the project from the repository, make changes, and commit it to the repository in the server.

GIT doesn't use the client-server model. When you download a project from a remote GIT repository, you download everything, including the version history and changes of the individual files, and your local GIT acts as a server, where you can do check-in, check-out, and all other typical version control activities. Later when you are ready, you can merge your changes to the remote GIT repository.

So, the installation and configuration steps are exactly the same whether you using GIT on your local machine, to manage your own project, or you are installing GIT on a server, from where other developers will download the project to their local GIT repositories.

If you are a developer, you might want to install GIT on your local machine for two reasons: 1) You like to manage your own project locally using a version control tool 2) You want to modify a code that is located in a remote central GIT repository.

If you are a sysadmin, you might want to install GIT on a server so that it acts as a central repository to hold all the source code for your company. From here, you can allow developers to download the projects to their local GIT repositories, make changes and they can check-in back to your central repository when they are done.

Irrespective of how you are planning to use GIT, the installation steps, and the basic commands mentioned below are exactly the same.

1. Download and Install GIT

First, download the GIT from [here](http://kernel.org/pub/software/scm/git/git-1.7.6.tar.bz2). Or, download it directly using wget as shown below.

```
cd
wget http://kernel.org/pub/software/scm/git/git-1.7.6.tar.bz2
```

Next, extract the downloaded file.

```
tar xvfj git-1.7.6.tar.bz2
cd git-1.7.6
```

Finally, install GIT as shown below using the default configure option. If you want to customize the installation, do “./configure --help” to view all available configuration options.

```
./configure
make
make install
```

2. Initial Configuration

Git is installed by default under /usr/local/bin. Once you've installed GIT, verify it as shown below.

```
$ whereis git
git: /usr/local/bin/git
```

```
$ git --version
git version 1.7.6
```

```
$ git --help.
```

The first step is to specify your username and email address to your GIT repository using “git config” as shown below.

```
git config --global user.name "GIT Admin"
```

```
git config --global user.email ramesh@thegeekstuff.com
```

Verify the git configuration information as shown below.

```
$ git config --list
user.name=GIT Admin
user.email=ramesh@thegeekstuff.com
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
```

This information is stored in the .gitconfig file under your home directory.

```
$ cat ~/.gitconfig
[user]
    name = GIT Admin
    email = ramesh@thegeekstuff.com
```

3. Create a Project

You can make any of your local directory as a GIT project (i.e repository). For example, if your project is located under /home/ramesh/projects/passworddragon, you can make that as your GIT project. First, cd to that directory, and execute git init as shown below.

```
$ cd /home/ramesh/projects/passworddragon
```

```
$ git init
Initialized empty Git repository in /home/ramesh/projects/passworddragon/.git/
```

This will create a .git directory under your project folder. Following is the content of the .git directory. GIT uses this directory to store information on how it is tracking the changes.

```
$ ls -altr .git
total 40
drwxrwxr-x 4 git git 4096 Aug 13 22:39 refs
drwxrwxr-x 4 git git 4096 Aug 13 22:39 objects
drwxrwxr-x 2 git git 4096 Aug 13 22:39 info
drwxrwxr-x 2 git git 4096 Aug 13 22:39 hooks
-rw-rw-r-- 1 git git  23 Aug 13 22:39 HEAD
-rw-rw-r-- 1 git git  73 Aug 13 22:39 description
-rw-rw-r-- 1 git git  92 Aug 13 22:39 config
drwxrwxr-x 2 git git 4096 Aug 13 22:39 branches
drwxrwxr-x 36 git git 4096 Aug 13 22:39 ..
drwxrwxr-x 7 git git 4096 Aug 13 22:39 .
```

Note: If you are sysadmin, who is trying to create a GIT central repository for your company, from where developers can download the projects, you may want to create a username called ‘git’ and organize all your projects under this account. For example: /home/git/project1, /home/git/project2, etc. Once you have the project organized, cd to the project directory, and do ‘git init’ from there as git user.

4. Add and Commit files to the Project

Once you’ve initialized the project using “git init”, add the files located under this project directory, using “git add”.

If there are different types of files under your project directory, and you want GIT to manage only certain types of files, add only those to the GIT as shown below. This example adds only the *.java and *.c files.

```
git add *.java
git add *.c
```

Typically you would like to add all the files under the project directory to the GIT project. Just do “git add .”, which will add all the files in the current directory and all the sub-directories to the GIT project.

```
git add .
```

Once you’ve added the files to the repository, you should commit those files, as shown below.

```
$ git commit -m 'Initial upload of the project'
create mode 100755 PasswordDragon.java
create mode 100755 pwm/ui/DataManager.java
create mode 100755 pwm/ui/PasswordFrame.java
create mode 100755 pwm/tools/StrongEncryption.java
create mode 100755 pwm/tools/PasswordStrength.java
..
```

If you didn’t specify your username and email address using “git config” as explained above, you’ll get the following error message.

```
$ git commit -m 'Initial upload of the project'
```

```
*** Please tell me who you are.
```

Run

```
git config --global user.email "you@example.com"  
git config --global user.name "Your Name"
```

to set your account's default identity.

Omit --global to set the identity only in this repository.

```
fatal: empty ident  not allowed
```

5. Make Changes and Commit the File

You've installed GIT, created a project repository, committed all the files to the GIT project.

Now it is time to start making some changes to a file and commit it to the repository.

```
vi PasswordDragon.java
```

Once you've modified a file locally, you can view the changes. i.e The difference between your local copy, and the copy already committed in the GIT project using "git diff" as shown below.

```
$ git diff  
diff --git a/PasswordDragon.java b/PasswordDragon.java  
index 6166ed1..fd82d32 100644  
--- a/PasswordDragon.java  
+++ b/PasswordDragon.java  
@@ -2,7 +2,7 @@  
- public counter=10  
+ public counter=55
```

Once you've made modifications to it, reviewed the changes, and happy with it, you can commit the file to GIT repository. This is a two step process. First, you should add the file to the staging area, and commit to the GIT project as shown below.

```
git add PasswordDragon.java
```

When you perform commit, it will open your default editor, where you can enter the comment. Once you save your comment and exit the editor, it will commit the file to the GIT project and display the following message.

```
$ git commit  
[master 80f10a9] Added password strength meter functionality  
1 files changed, 56 insertions(+), 7 deletions(-)
```

Note: You can also do “git commit -a”, which will do the add and commit at the same time.

6. View Status and Commit Logs

From your local repository, when you perform “git status”, it will display the current status. When the local copy is not changed (or when all the files are committed), you’ll see the following message.

```
$ git status
# On branch master
nothing to commit (working directory clean)
```

If you’ve made changes to a file, and not committed yet, you’ll see the following message.

```
$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add ..." to update what will be committed)
#   (use "git checkout -- ..." to discard changes in working directory)
#
#       modified:   PasswordDragon.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

You can also view the history of a file as shown below.

```
$ git log PasswordDragon.java
commit c919ced7f42f4bc06d563c1a1eaa107f2b2420d5
Author: GIT Admin
Date:   Sat Aug 13 22:54:57 2011 -0700
```

Added password strength meter functionality

```
commit c141b7bdbff429de35e36bafb2e43edc655e9957
Author: GIT Admin
Date:   Sat Aug 13 20:08:02 2011 -0700
```

Initial upload of the project

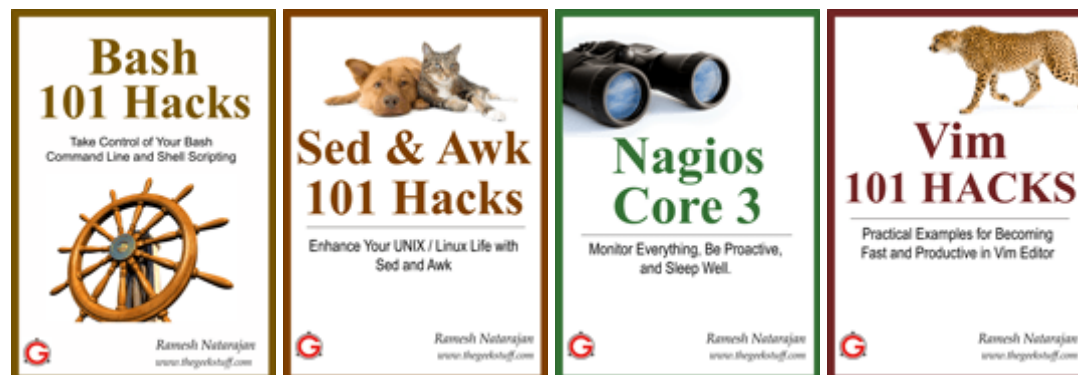
  Tweet  Like 37 > [Add your comment](#)

If you enjoyed this article, you might also like..

1. [50 Linux Sysadmin Tutorials](#)

• [Awk Introduction – 7 Awk Print Examples](#)

2. [50 Most Frequently Used Linux Commands \(With Examples\)](#)
 3. [Top 25 Best Linux Performance Monitoring and Debugging Tools](#)
 4. [Mommy, I found it! – 15 Practical Linux Find Command Examples](#)
 5. [Linux 101 Hacks 2nd Edition eBook](#) **Free**
- [Advanced Sed Substitution Examples](#)
 - [8 Essential Vim Editor Navigation Fundamentals](#)
 - [25 Most Frequently Used Linux IPTables Rules Examples](#)
 - [Turbocharge PuTTY with 12 Powerful Add-Ons](#)



{ 24 comments... [add one](#) }

- manimaran August 17, 2011, 3:30 am

nice article

[Link](#)

- BalaC August 17, 2011, 7:49 am

Nice start. I am finding difficult to get around the branching concept, if possible continue your explanation on those as well.

[Link](#)

- vaibhav August 17, 2011, 8:09 am

In step5 ,
PasswordDragon.java was modified and while committing
git add index.html
git commit
was done

So is index.html to be used everytime? or after add there should be name of the file in which changes are made.

[Link](#)

- jalal hajigholamali August 17, 2011, 8:36 am

Nice article,... thanks

[Link](#)

- [Andrew](#) August 17, 2011, 10:58 am

this was awesome! will there be a part 2? I want to know how to get the files from my local to github along with push and pull. Thanks!

[Link](#)

- [Ashok](#) August 17, 2011, 11:20 am

I believe this article will be a good starting point for me to explore about Version control system.

Thanks,
Ashok

[Link](#)

- Abhijeet Kasurde August 17, 2011, 11:50 pm

Very nice article,Thanks.

[Link](#)

- kabamaru August 18, 2011, 1:28 am

Thanks, great info. A part two would be great as others already mentioned ☺

[Link](#)

- sjs August 20, 2011, 8:38 am

Thank you for the nice intro to git; I have a few questions: You state that git “is the most versatile distributed version control system”, why? what makes it more versatile than cvs/svn?

You mention the ‘merge’ to a server repo, what does this do, and why would anyone do it? how is this different from the cvs/svn “commit to the server” methodology?

Other ideas to cover in the next installment on git: tagging and branching, merging code with other team members, and handling conflicts in merging.

[Link](#)

- Alex January 13, 2012, 5:44 am

Thanks a lot! I'm inspired to start learning git

[Link](#)

- DEEPAK August 26, 2012, 8:14 am

Thanks for the information,
but i have one question,

* if i have many servers & one central git server, than how to configure the git server to auto deploy the files from the other servers ==> git server ==> developers local machine.

[Link](#)

- Geert September 16, 2012, 2:57 am

Hi,

The URL given for the wget command (section “1. Download and Install GIT”) is not working any more. I had to use the following command:

wget <http://git-core.googlecode.com/files/git-1.7.12.tar.gz>

HTH,

Best rgds,
–Geert

[Link](#)

- Rohit October 29, 2012, 9:19 am

A very very simple step, even an unknown person will use, will be able to install GIT.
Hats offff 2 u. Can you please share the prerequisite required for installation for GIT

[Link](#)

- teji April 7, 2013, 11:14 pm

can u please add something related to syncing between server and client side.

[Link](#)

- Slava April 11, 2013, 10:45 am

I am sorry... how can you say “git doesn't use server-client model”? it is of course client-server application... even more... git uses mid-server, which resides between end client and git server...

wget <http://kernel.org/pub/software/scm/git/git-1.7.6.tar.bz2>

wget: error while loading shared libraries: /usr/lib64/libc.so.6: ELF file OS ABI invalid

[Link](#)

- Vidhya June 5, 2013, 3:55 am

Good Post. Indeed it was helpful!

[Link](#)

- Marc October 31, 2013, 5:00 pm

Thanks a lot for the Tutorial. That helps me a lot in my studies.

[Link](#)

- vicks March 18, 2014, 6:52 pm

Wonderful article. However, I would like to know how to get the git repository url in webstorm. For e.g. git clone user@server:/var/www/myproject. Something similar.

Thanks.

[Link](#)

- Terrence Sun July 8, 2014, 10:35 pm

Thanks. It's a good start point for playing with git.

[Link](#)

- nagendra March 5, 2015, 12:54 am

pls explain push pull also

[Link](#)

- Jeff September 16, 2015, 3:04 pm

Very clear, thanks!

[Link](#)

- Bimal January 22, 2016, 12:30 am

Unable to compile git on Linux due to dependency on zlib.h. Any help to solve this issue?

```
linuxcli: bimaljha/git/git-master> make
CC credential-store.o
In file included from credential-store.c:1:
cache.h:39:18: warning: zlib.h: No such file or directory
In file included from credential-store.c:1:
cache.h:41: error: expected specifier-qualifier-list before `z_stream'
make: *** [credential-store.o] Error 1
linuxcli: bimaljha/git/git-master> zypper se zlib-devel
Loading repository data...
Reading installed packages...
No packages found.
linuxcli: bimaljha/git/git-master> zypper install zlib-devel
Loading repository data...
Reading installed packages...
Package 'zlib-devel' not found.
Resolving package dependencies...

Nothing to do.
linuxcli: bimaljha/git/git-master> zypper install zlib
Loading repository data...
Reading installed packages...
'zlib' is already installed.
No update candidate for 'zlib-1.2.7-0.10.128.x86_64'. The highest available version is already installed.
Resolving package dependencies...

Nothing to do.
linuxcli: bimaljha/git/git-master> zypper install git
Loading repository data...
Reading installed packages...
'git' not found in package names. Trying capabilities.
No provider of 'git' found.
Resolving package dependencies...

Nothing to do.
linuxcli: bimaljha/git/git-master>
```

Thanks.

[Link](#)

- shashi January 17, 2017, 8:19 am

IF I MADE ANY CHANGES IN A FILEAFTER MODIFICATION IT WONT COMMIT R8.....THEN HOW WE R GOING TO COMMIT

SUPOSE I HAVE ONE REPO i.e test1 if i made any modification i that after modification hoe to commit it

[Link](#)

- suresh kumar uppada June 26, 2017, 9:19 am

nice

[Link](#)

Leave a Comment

Name

Email

Website

Comment

☐ Notify me of followup comments via e-mail

Next post: [Quick Update on the Upcoming eBook](#)

Previous post: [15 Linux Bash History Expansion Examples You Should Know](#)

[RSS](#) | [Email](#) | [Twitter](#) | [Facebook](#) | [Google+](#)

Google Custom Search



EBOOKS

- **Free** [Linux 101 Hacks 2nd Edition eBook](#) - Practical Examples to Build a Strong Foundation in Linux
- [Bash 101 Hacks eBook](#) - Take Control of Your Bash Command Line and Shell Scripting
- [Sed and Awk 101 Hacks eBook](#) - Enhance Your UNIX / Linux Life with Sed and Awk
- [Vim 101 Hacks eBook](#) - Practical Examples for Becoming Fast and Productive in Vim Editor
- [Nagios Core 3 eBook](#) - Monitor Everything, Be Proactive, and Sleep Well



POPULAR POSTS

- [15 Essential Accessories for Your Nikon or Canon DSLR Camera](#)

- [12 Amazing and Essential Linux Books To Enrich Your Brain and Library](#)
- [50 UNIX / Linux Sysadmin Tutorials](#)
- [50 Most Frequently Used UNIX / Linux Commands \(With Examples\)](#)
- [How To Be Productive and Get Things Done Using GTD](#)
- [30 Things To Do When you are Bored and have a Computer](#)
- [Linux Directory Structure \(File System Structure\) Explained with Examples](#)
- [Linux Crontab: 15 Awesome Cron Job Examples](#)
- [Get a Grip on the Grep! – 15 Practical Grep Command Examples](#)
- [Unix LS Command: 15 Practical Examples](#)
- [15 Examples To Master Linux Command Line History](#)
- [Top 10 Open Source Bug Tracking System](#)
- [Vi and Vim Macro Tutorial: How To Record and Play](#)
- [Mommy, I found it! -- 15 Practical Linux Find Command Examples](#)
- [15 Awesome Gmail Tips and Tricks](#)
- [15 Awesome Google Search Tips and Tricks](#)
- [RAID 0, RAID 1, RAID 5, RAID 10 Explained with Diagrams](#)
- [Can You Top This? 15 Practical Linux Top Command Examples](#)
- [Top 5 Best System Monitoring Tools](#)
- [Top 5 Best Linux OS Distributions](#)
- [How To Monitor Remote Linux Host using Nagios 3.0](#)
- [Awk Introduction Tutorial – 7 Awk Print Examples](#)
- [How to Backup Linux? 15 rsync Command Examples](#)
- [The Ultimate Wget Download Guide With 15 Awesome Examples](#)
- [Top 5 Best Linux Text Editors](#)
- [Packet Analyzer: 15 TCPDUMP Command Examples](#)
- [The Ultimate Bash Array Tutorial with 15 Examples](#)
- [3 Steps to Perform SSH Login Without Password Using ssh-keygen & ssh-copy-id](#)
- [Unix Sed Tutorial: Advanced Sed Substitution Examples](#)
- [UNIX / Linux: 10 Netstat Command Examples](#)
- [The Ultimate Guide for Creating Strong Passwords](#)
- [6 Steps to Secure Your Home Wireless Network](#)
- [Turbocharge PuTTY with 12 Powerful Add-Ons](#)

CATEGORIES

- [Linux Tutorials](#)
- [Vim Editor](#)
- [Sed Scripting](#)
- [Awk Scripting](#)
- [Bash Shell Scripting](#)
- [Nagios Monitoring](#)

- [OpenSSH](#)
- [IPTables Firewall](#)
- [Apache Web Server](#)
- [MySQL Database](#)
- [Perl Programming](#)
- [Google Tutorials](#)
- [Ubuntu Tutorials](#)
- [PostgreSQL DB](#)
- [Hello World Examples](#)
- [C Programming](#)
- [C++ Programming](#)
- [DELL Server Tutorials](#)
- [Oracle Database](#)
- [VMware Tutorials](#)



Ramesh Natarajan

 Follow

About The Geek Stuff



My name is **Ramesh Natarajan**. I will be posting instruction guides, how-to, troubleshooting tips and tricks on Linux, database, hardware, security and web. My focus is to write articles that will either teach you or help you resolve a problem. Read more about [Ramesh Natarajan](#) and the blog.

Contact Us

Email Me : Use this [Contact Form](#) to get in touch me with your comments, questions or suggestions about this site. You can also simply drop me a line to say hello!.

[Follow us on Google+](#)

[Follow us on Twitter](#)

[Become a fan on Facebook](#)

Support Us

Support this blog by purchasing one of my ebooks.

[Bash 101 Hacks eBook](#)

[Sed and Awk 101 Hacks eBook](#)

[Vim 101 Hacks eBook](#)

[Nagios Core 3 eBook](#)

Copyright © 2008–2017 Ramesh Natarajan. All rights reserved | [Terms of Service](#)