


 Subscribe Share Contents ▼

How To Install Git on Debian 8

Posted June 29, 2015

 177.8k

GIT

DEBIAN



By: Sourav Kundu

Not using **Debian 8**? Choose a different version:

Introduction

Git is a **version control** system distributed under the terms of the GNU General Public License v.2 since its release in 2005.

Git is software used primarily for version control which allows for *non-linear* development of projects, even ones with large amounts of data. Every working directory in Git is a full-fledged repository with **complete history and tracking** independent of network access or a central server.

The advantages of using Git stem from the way the program stores data. Unlike other version control systems, it is best to think of Git's storage process as a set of snapshots of a mini filesystem, primarily on your local disk. Git maximizes efficiency and allows for powerful tools to be built on top of it.

In this tutorial we'll install and configure Git on your Debian 8 Linux server.

Prerequisites

You'll need the following items for this tutorial:

- A Droplet running Debian 8
- A sudo user

What the Red Means

The majority of code in this tutorial can be copy and pasted as-is! The lines that you will need to customize will be **red** in this tutorial.

Step 1 — Installing Git with APT

Before you install Git, make sure that your package lists are updated by executing the following command:

```
$ sudo apt-get update
```

Install Git with `apt-get` in one command:

```
$ sudo apt-get install git-core
```

This is the only command you'll need to install Git. The next part is configuring Git.

Using `apt-get` is the easiest and probably one of the most reliable ways to install Git, because APT takes care of all the software dependencies your system might have.

Now, let us take a look at how to configure Git.

Step 2 — Configuring Git

Git implements version control using two primary settings:

- A user name
- The user's email

This information will be embedded in every commit you make with Git so it can track who is making which commits.

We need to add these two settings in our Git configuration file. This can be done with the help of the `git config` utility. Here's how:

Set your Git user name:

```
$ git config --global user.name "Sammy Shark"
```

Set your Git email:

```
$ git config --global user.email sammy@example.com
```

View all Git settings:

You can view these newly-configured settings (and all the previously existing ones, if any) using the `--list` parameter in the `git config` utility.

```
$ git config --list
```

You should see your user settings:

Output

```
user.name=Sammy Shark
user.email=sammy@example.com
```

.gitconfig

If you want to get your hands dirty with the Git configuration file, simply fire up `nano` (or your favorite text editor) and edit to your heart's content:

```
$ nano ~/.gitconfig
```

Here you can manually update your Git settings:

```
~/.gitconfig

[user]
  name = Sammy Shark
  email = sammy@example.com
```

This is the basic configuration you need to get up and running with Git.

Adding your username and email is not *mandatory*, but it is recommended. Otherwise, you'll get a message like this when you use Git:

Output when Git user name and email are not set

```
[master 0d9d21d] initial project version
Committer: root
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:
```

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

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

Congratulations on your very own Git installation.

Conclusion

Here are a few tutorials which you can use to help you take full advantage of Git:

- [How To Use Git Effectively](#)
- [How To Use Git Branches](#)

Happy branching!

By: Sourav Kundu

♥ Upvote (7)

📄 Subscribe

🔗 Share



Editor:
Sharon Campbell



We just made it easier for you to deploy faster.

[TRY FREE](#)

Related Tutorials

How To Use the GitLab User Interface To Manage Projects

How To Set Up a Private Git Server on a VPS

How To Use Git Branches

How To Use Git Effectively

How To Use Git: A Reference Guide

1 Comment

Leave a comment...

Log In to Comment

^ [michaeljroberts](#) February 12, 2018

0 Really nice introduction to this topic. What would be really nice is if we could have an overview of linking various repositories to our server, i.e., setting up keys etc



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.



Copyright © 2019 DigitalOcean™ Inc.

[Community](#) [Tutorials](#) [Questions](#) [Projects](#) [Tags](#) [Newsletter](#) [RSS](#)

[Distros & One-Click Apps](#) [Terms, Privacy, & Copyright](#) [Security](#) [Report a Bug](#) [Write for DOnations](#) [Shop](#)

