How To Install Git on Debian 9

Posted September 7, 2018 23.9k GIT OPEN SOURCE DEBIAN 9

By: Lisa Tagliaferri

Not using **Debian 9**? Choose a different version:

- CentOS 7
- Debian 8
- Ubuntu 18.04

Introduction

Software version control systems enable you to keep track of your software at the source level. With versioning tools, you can track changes, revert to previous stages, and branch to create alternate versions of files and directories.

Git is one of the most popular version control systems currently available. Many projects' files are maintained in a Git repository, and sites like GitHub, GitLab, and Bitbucket help to facilitate software development project sharing and collaboration.

In this tutorial, we'll install and configure Git on a Debian 9 server. We will cover how to install the software in two different ways, each of which have their own benefits depending on your specific needs.

Prerequisites

In order to complete this tutorial, you should have a non-root user with `sudo` privileges on an Debian 9 server. To learn how to achieve this setup, follow our Debian 9 initial server setup guide.

With your server and user set up, you are ready to begin.

SCROLL TO TOP

Installing Git with Default Packages

Debian's default repositories provide you with a fast method to install Git. Note that the version you install via these repositories may be older than the newest version currently available. If you need the latest release, consider moving to the next section of this tutorial to learn how to install and compile Git from source.

First, use the apt package management tools to update your local package index. With the update complete, you can download and install Git:

```
$ sudo apt update
$ sudo apt install git
```

You can confirm that you have installed Git correctly by running the following command:

```
$ git --version
```

```
Output
git version 2.11.0
```

With Git successfully installed, you can now move on to the Setting Up Git section of this tutorial to complete your setup.

Installing Git from Source

A more flexible method of installing Git is to compile the software from source. This takes longer and will not be maintained through your package manager, but it will allow you to download the latest release and will give you some control over the options you include if you wish to customize.

Before you begin, you need to install the software that Git depends on. This is all available in the default repositories, so we can update our local package index and then install the packages.

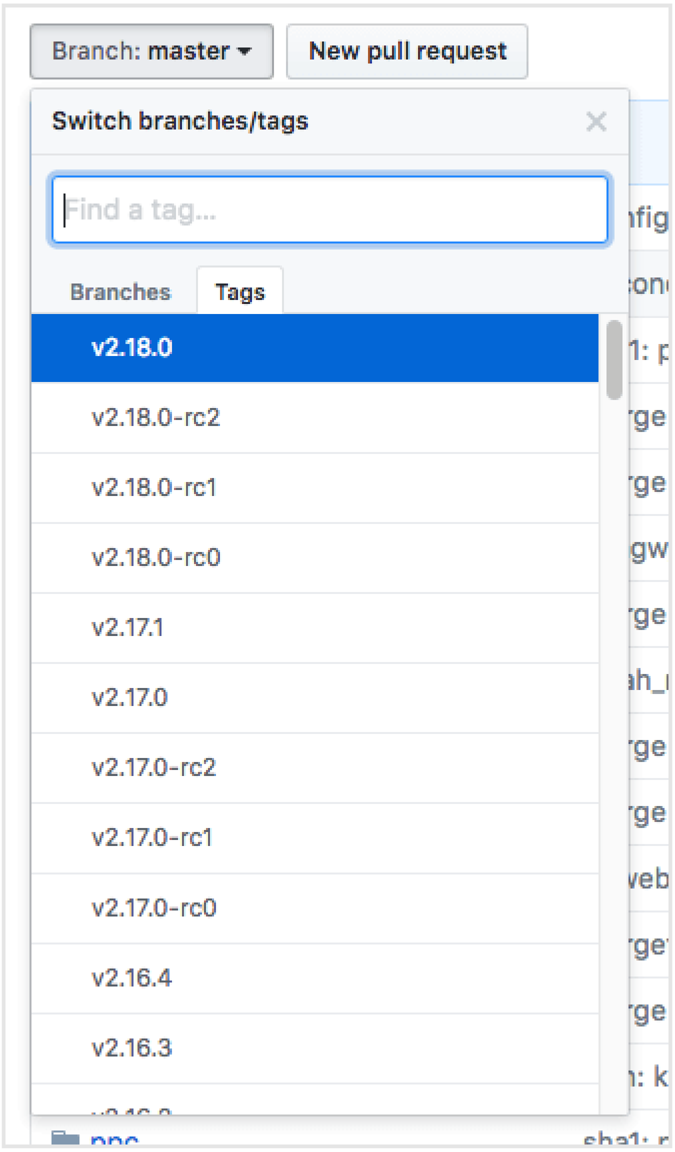
```
$ sudo apt update
$ sudo apt install make libssl-dev libghc-zlib-dev libcurl4-gnutls-dev libexpat1-dev gettext unzip
```

After you have installed the necessary dependencies, you can go ahead and get the version of Git you want by visiting the Git project's mirror on GitHub, available via the following URL:

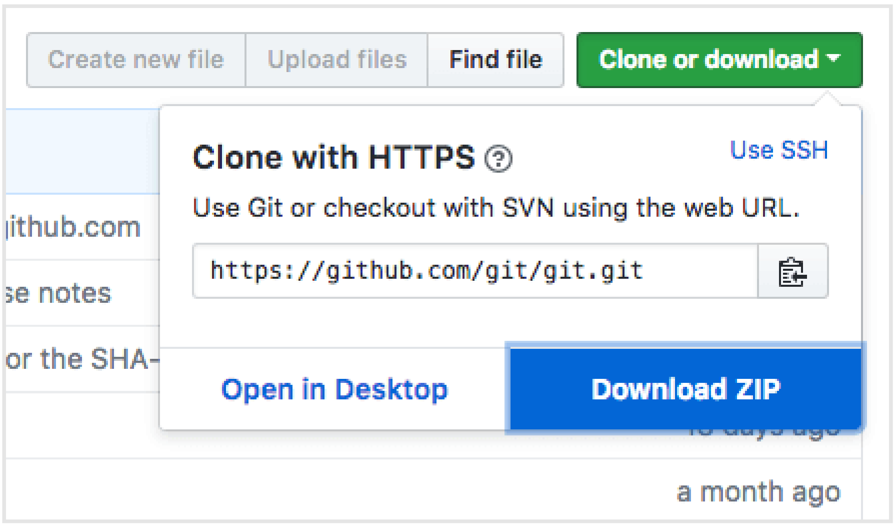
```
https://github.com/git/git
```

From here, be sure that you are on the `master` branch. Click on the **Tags** link and select
version. Unless you have a reason for downloading a *release candidate* (marked as **rc**) v

these as they may be unstable.



Next, on the right side of the page, click on the **Clone or download** button, then right-click on **Download ZIP** and copy the link address that ends in `.zip`.



Back on your Debian 9 server, move into the `tmp` directory to download temporary files.

```
$ cd /tmp
```

SCROLL TO TOP

From there, you can use the `wget` command to install the copied zip file link. We'll specify a new name for the file: `git.zip`.

```
$ wget https://github.com/git/git/archive/v2.18.0.zip -O git.zip
```

Unzip the file that you downloaded and move into the resulting directory by typing:

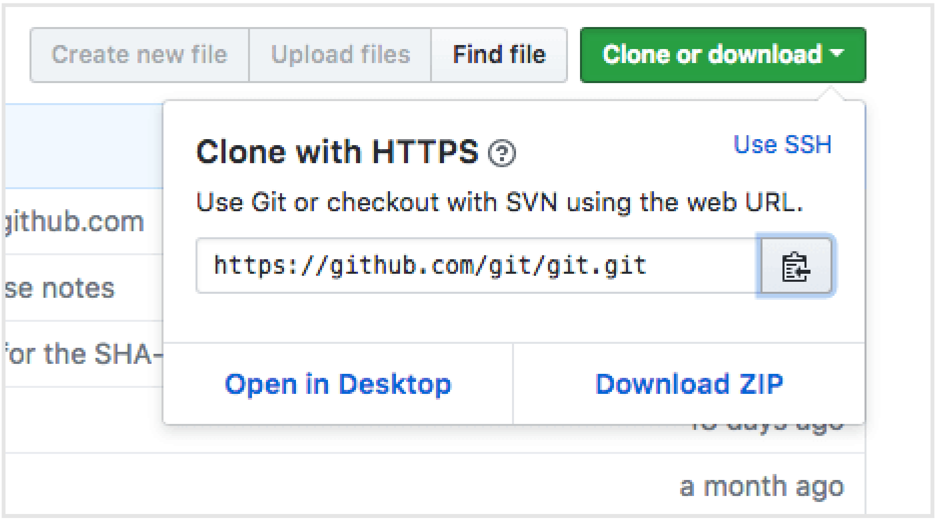
```
$ unzip git.zip
$ cd git-*
```

Now, you can make the package and install it by typing these two commands:

```
$ make prefix=/usr/local all
$ sudo make prefix=/usr/local install
```

To ensure that the install was successful, you can type `git --version` and you should receive relevant output that specifies the current installed version of Git.

Now that you have Git installed, if you want to upgrade to a later version, you can clone the repository, and then build and install. To find the URL to use for the clone operation, navigate to the branch or tag that you want on the project's GitHub page and then copy the clone URL on the right side:



Clone with HTTPS

Use Git or checkout with SVN using the web URL.

https://github.com/git/git.git

At the time of writing, the relevant URL is:

```
https://github.com/git/git.git
```

Change to your home directory, and use `git clone` on the URL you just copied:

```
$ cd ~
$ git clone https://github.com/git/git.git
```

SCROLL TO TOP

This will create a new directory within your current directory where you can rebuild the package and reinstall the newer version, just like you did above. This will overwrite your older version with the new version:

```
$ cd git
$ make prefix=/usr/local all
$ sudo make prefix=/usr/local install
```

With this complete, you can be sure that your version of Git is up to date.

Setting Up Git

Now that you have Git installed, you should configure it so that the generated commit messages will contain your correct information.

This can be achieved by using the `git config` command. Specifically, we need to provide our name and email address because Git embeds this information into each commit we do. We can go ahead and add this information by typing:

```
$ git config --global user.name "Sammy"
$ git config --global user.email "sammy@domain.com"
```

We can see all of the configuration items that have been set by typing:

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

```
Output
user.name=Sammy
user.email=sammy@domain.com
...
```

The information you enter is stored in your Git configuration file, which you can optionally edit by hand with a text editor like this:

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

~/.gitconfig contents

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

There are many other options that you can set, but these are the two essential ones needed. If you skip this step, you'll likely see warnings when you commit to Git. This makes more work for you because you will then have to revise the commits you have done with the corrected information.

Conclusion

You should now have Git installed and ready to use on your system.

To learn more about how to use Git, check out these articles and series:

- How To Use Git Effectively
- How To Use Git Branches
- An Introduction to Open Source series

By: Lisa Tagliaferri

Upvote (1) Subscribe Share

Related Tutorials

How To Use Git: A Reference Guide

How To Install and Configure GitLab on Debian 9

How to Use Node.js and Github Webhooks to Keep Remote Projects in Sync

How To Install Git on Ubuntu 18.04 [Quickstart]

How To Install and Configure GitLab on Ubuntu 18.04

SCROLL TO TOP

2 Comments

Leave a comment...

Log In to Comment

zozo36769 *October 27, 2018*

0 i have a little problem with installing it

zozo36769 *October 27, 2018*

0 https://subwaysurfers.vip/ , https://psiphon.vip/ , https://hillclimbracing.vip/

SCROLL TO TOP