How To Install Git on Ubuntu 18.04

Posted July 6, 2018 71.1k GIT OPEN SOURCE UBUNTU 18.04

6

By: Lisa Tagliaferri

Not using **Ubuntu 18.04**? Choose a different version:

An earlier version of this tutorial was written by Brennen Bearnes.

Introduction

Version control systems are increasingly indispensable in modern software development as versioning allows you to keep track of your software at the source level. You can track changes, revert to previous stages, and branch to create alternate versions of files and directories.

One of the most popular version control systems currently available is Git. 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 guide, we will demonstrate how to install and configure Git on an Ubuntu 18.04 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 Ubuntu 18.04 server. To learn how to achieve this setup, follow our manual initial server setup guide or run our automated script.

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

Installing Git with Default Packages

Ubuntu'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 2.17.1
```

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.

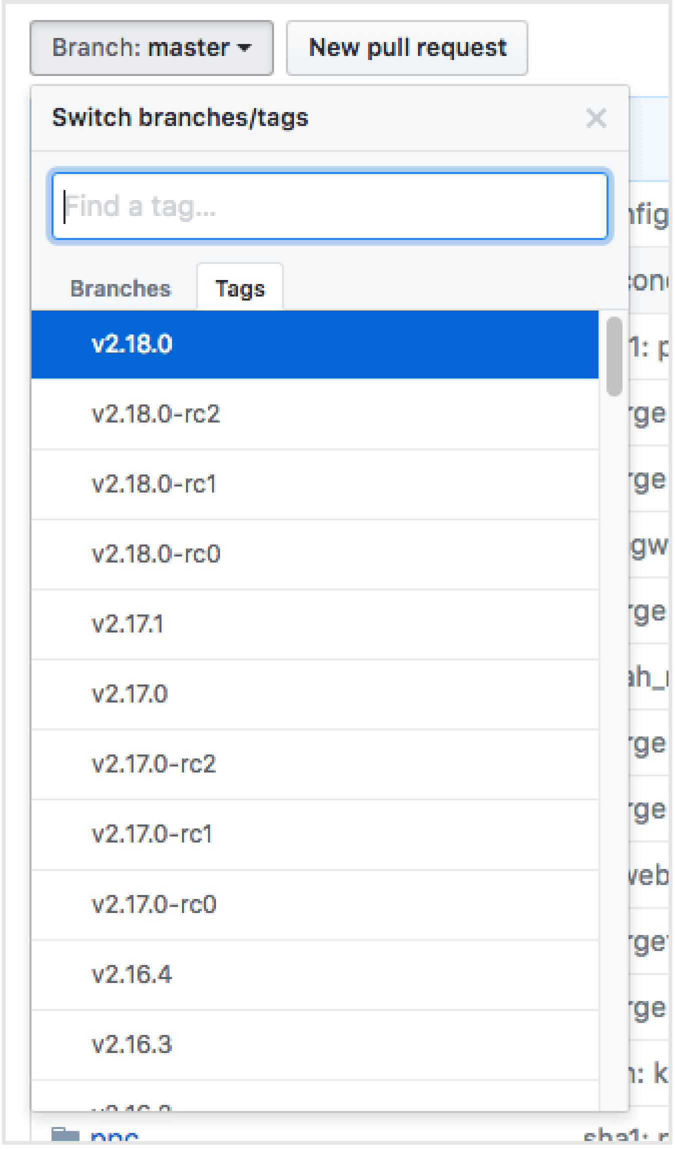
Before you begin, you need to install the software that Git depends on. This is all available in the default repositories, so we can update our local package index and then install the packages.

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

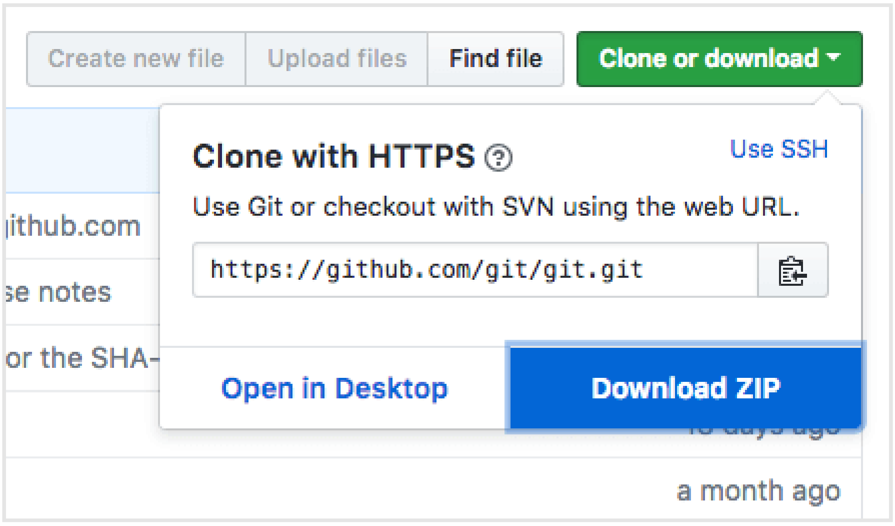
After you have installed the necessary dependencies, you can go ahead and get the version of Git you want by visiting the Git project's mirror on GitHub, available via the following URL:

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

From here, be sure that you are on the `master` branch. Click on the **Tags** link and select your desired Git version. Unless you have a reason for downloading a *release candidate* (marked as **rc**) version, try to avoid 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 Ubuntu 16.04 server, move into the `tmp` directory to download temporary files.

```
$ cd /tmp
```

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:

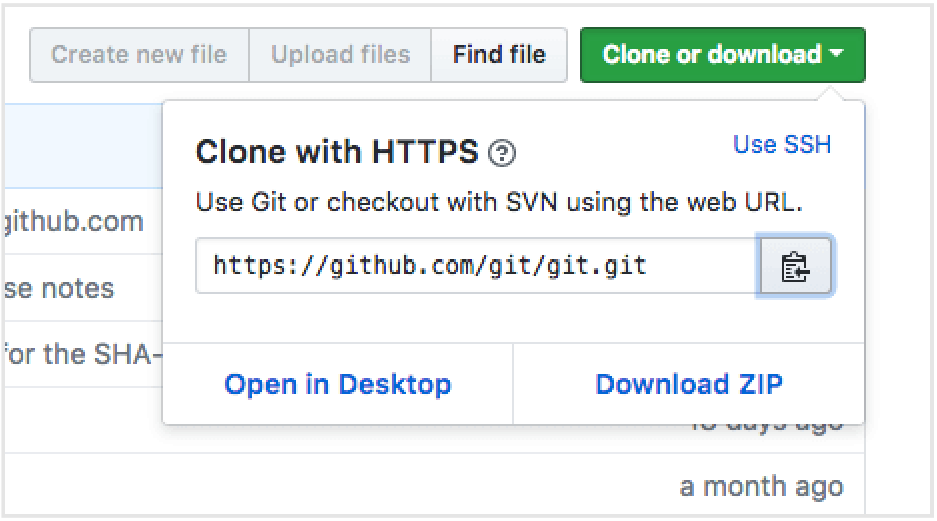
```
$ unzip git.zip
$ cd git-*
```

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:



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
```

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 "Your Name"
$ git config --global user.email "youremail@domain.com"
```

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

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

```
Output
user.name=Your Name
user.email=youremail@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 = Your Name
  email = youremail@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

By: Lisa Tagliaferri

Upvote (6) Subscribe Share

We just made it easier for you to deploy faster.

TRY FREE

Related Tutorials

How To Use Git: A Reference Guide

How To Install Git on Debian 9

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]

8 Comments

Leave a comment...

Log In to Comment

varunchopra *July 7, 2018*

0 I think the title should specify "and configure" as well since we're configuring our email address as well. Installation is just `apt install git` but the articles tells a lot more.

ltagliaferri MOD *July 9, 2018*

0 Thank you for your comment. The tutorial does scratch the surface on configuring Git. Those who are interested in learning more can take a look at the official `git config` documentation.

johnaaronrose *August 16, 2018*

0 git fails to install on 18.04 using sudo apt install git:

```
john@JohnPC:~$ sudo apt install git
[sudo] password for john:
Reading package lists... Done
Building dependency tree

Reading state information... Done
The following additional packages will be installed:
git-man liberror-perl
Suggested packages:
git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed
git git-man liberror-perl
0 to upgrade, 3 to newly install, 0 to remove and 3 not to upgrade.
4 not fully installed or removed.
Need to get 4,731 kB of archives.
After this operation, 33.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://gb.archive.ubuntu.com/ubuntu bionic/main amd64 liberror-perl all 0.17025-1 [22.8 kB]
Get:2 http://gb.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git-man all 1:2.17.1-1ubuntu0.1 [803 kB]
Get:3 http://gb.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git amd64 1:2.17.1-1ubuntu0.1 [3,905 kB]
Fetched 4,731 kB in 1s (6,041 kB/s)
Selecting previously unselected package liberror-perl.
```

```
(Reading database ... 221319 files and directories currently installed.)
Preparing to unpack .../liberror-perl0.17025-1all.deb ...
Unpacking liberror-perl (0.17025-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man1%3a2.17.1-1ubuntu0.1all.deb ...
Unpacking git-man (1:2.17.1-1ubuntu0.1) ...
Selecting previously unselected package git.
Preparing to unpack .../git1%3a2.17.1-1ubuntu0.1amd64.deb ...
Unpacking git (1:2.17.1-1ubuntu0.1) ...
Setting up git-man (1:2.17.1-1ubuntu0.1) ...
Setting up liberror-perl (0.17025-1) ...
Setting up winbind (2:4.7.6+dfsg~ubuntu-0ubuntu2.2) ...
Job for winbind.service failed because the control process exited with error code.
See "systemctl status winbind.service" and "journalctl -xe" for details.
invoke-rc.d: initscript winbind, action "restart" failed.
● winbind.service - Samba Winbind Daemon
Loaded: loaded (/lib/systemd/system/winbind.service; enabled; vendor preset: enabled)
Active: failed (Result: exit-code) since Thu 2018-08-16 09:01:22 BST; 20ms ago
Docs: man:winbindd(8)
man:samba(7)
man:smb.conf(5)
Process: 8571 ExecStart=/usr/sbin/winbindd --foreground --no-process-group $WINBINDOPTIONS
(code=exited, status=1/FAILURE)
Main PID: 8571 (code=exited, status=1/FAILURE)
```

---

ltagliaferri MOD *August 16, 2018*

0 Hi there, are you ensuring that you are updating your package list prior to installing Git with `sudo apt update`?

I just ran through the first two commands again and installation ran with the following complete output:

```
Output
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  git-man
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb
  git-cvs git-mediawiki git-svn
The following packages will be upgraded:
  git git-man

2 upgraded, 0 newly installed, 0 to remove and 130 not upgraded.
Need to get 0 B/4708 kB of archives.
After this operation, 50.2 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
(Reading database ... 59932 files and directories currently installed.)
```

```
Preparing to unpack .../git_1%3a2.17.1-1ubuntu0.1_amd64.deb ...
Unpacking git (1:2.17.1-1ubuntu0.1) over (1:2.17.0-1ubuntu1) ...
Preparing to unpack .../git-man_1%3a2.17.1-1ubuntu0.1_all.deb ...
Unpacking git-man (1:2.17.1-1ubuntu0.1) over (1:2.17.0-1ubuntu1) ...
Setting up git-man (1:2.17.1-1ubuntu0.1) ...
Processing triggers for man-db (2.8.3-2) ...
Setting up git (1:2.17.1-1ubuntu0.1) ...
```

```
$ git --version
```

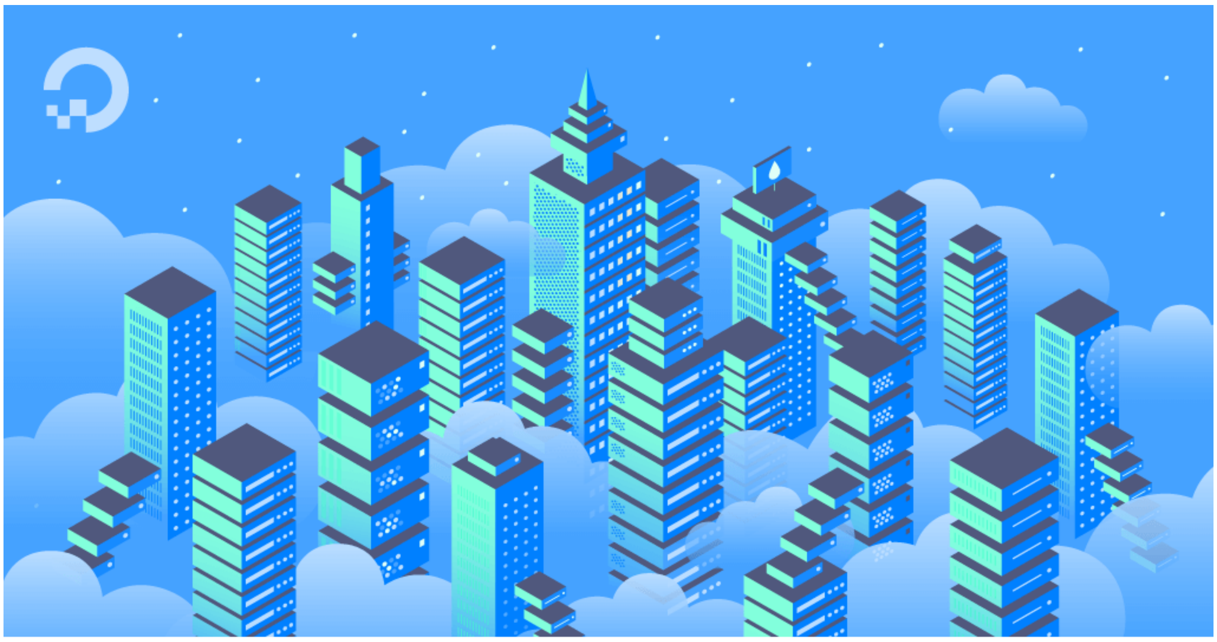
```
Output
git version 2.17.1
```

johnaaronrose *August 16, 2018*

0 Yes. I did "sudo apt update" followed by "sudo apt upgrade" before trying "sudo apt install git". BTW I notice that you are using Lubuntu. IMO that can be significantly different from regular Ubuntu.

ltagliaferri MOD *August 16, 2018*

0 This is all on a fresh DigitalOcean Ubuntu 18.04 Droplet that was set up following the initial server setup guide.



Initial Server Setup with Ubuntu 18.04

by Justin Ellingwood

When you first create a new Ubuntu 18.04 server, there are a few configuration steps that you should take early on as part of the basic setup. This will increase the security and usability of your server and will give you a solid foundation for subsequent...

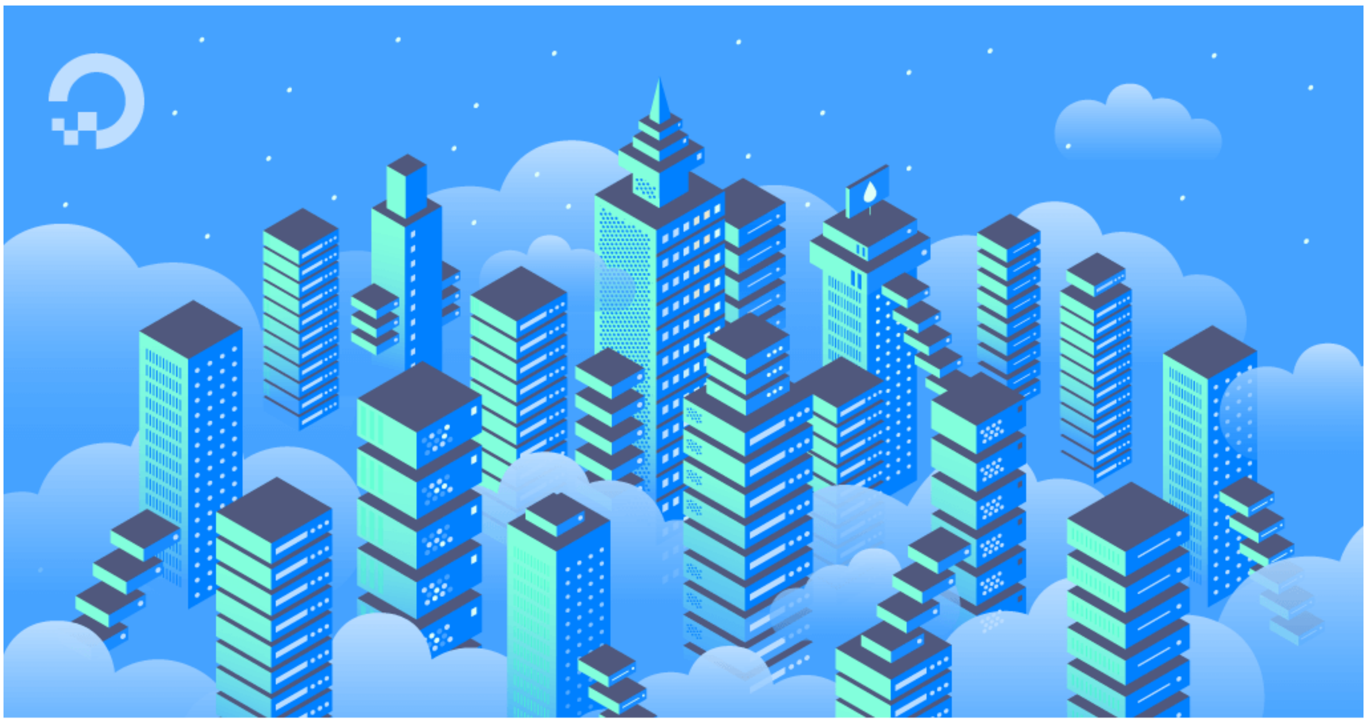
johnaaronrose *August 17, 2018*

0 @Justin
Your posting has been cut off in mid-sentence. What are the steps? BTW I'm using Ubuntu 18.04 Desktop (not Server).

---

ltagliaferri MOD *August 31, 2018*

0 @johnaaronrose — you can look at the official Git documentation for extra guidance. You can click the link for Justin's tutorial to see the full post, but that is a server setup installation not a desktop. This tutorial also assumes the use of a server. Often, desktops will function the same but there may be differences.



Initial Server Setup with Ubuntu 18.04

by Justin Ellingwood

When you first create a new Ubuntu 18.04 server, there are a few configuration steps that you should take early on as part of the basic setup. This will increase the security and usability of your server and will give you a solid foundation for subsequent...