How to install latest version of git on CentOS 7.x/6.x

stackoverflow.com/questions/21820715/how-to-install-latest-version-of-git-on-centos-7-x-6-x

205

I used the usual:

It did not install the latest version of git on my CentOS 6. How can I update to the latest version of git for CentOS 6? The solution can be applicable to newer versions of CentOS such as CentOS 7.

265

✓

You can use WANDisco's CentOS repository to install Git 2.x: for CentOS 6, for CentOS 7

1. Install WANDisco repo package:

   ```
   yum install http://opensource.wandisco.com/centos/6/git/x86_64/wandisco-git-release-6-1.noarch.rpm
   - or -
   yum install http://opensource.wandisco.com/centos/7/git/x86_64/wandisco-git-release-7-1.noarch.rpm
   - or -
   yum install http://opensource.wandisco.com/centos/7/git/x86_64/wandisco-git-release-7-2.noarch.rpm
   ```

2. Install the latest version of Git 2.x:

3. Verify the version of Git that was installed:

   ```
   git --version
   ```

As of 24th Apr. 2019, the latest available version from WANDisco is **2.18.0**.

128

Having a look at the blog here I found the solution in one of the comments. Make sure you got the `rpmforge` repository added to your CentOS yum and just run the flowing command:

If you already have git installed then use:

Related question(s):

1. Facing issues while upgrading git to latest version on CentOS 6.4

Note update:

Thanks to Anthony Hatzopoulos, for `git v1.8x` you'll need to use git18 as in:

Note update 2:

Also thanks to @Axlrod for the below hint and @Hiphip for the feedback:

Change the `rpmforge.repo` file to have `rpmforge-extras` to enabled, . Otherwise it complained about dependency problems.

Note update 3:

> Installing a specific version of git say 2.x I found this nice and easy-to-follow guide on how to download the GIT source and compile it yourself (and install it). If the accepted answer does not give you the version you want, try the following instructions:

http://tecadmin.net/install-git-2-0-on-centos-rhel-fedora/

(And pasted/reformatted from above source in case it is removed later)

Step 1: Install Required Packages

Firstly we need to make sure that we have installed required packages on your system. Use following command to install required packages before compiling Git source.

Step 2: Uninstall old Git RPM

Now remove any prior installation of Git through RPM file or Yum package manager. If your older version is also compiled through source, then skip this step.

Step 3: Download and Compile Git Source

Download git source code from kernel git or simply use following command to download Git 2.0.4.

```
# cd /usr/src
# wget https://www.kernel.org/pub/software/scm/git/git-2.0.4.tar.gz
# tar xzf git-2.0.4.tar.gz
```

After downloading and extracting Git source code, Use following command to compile source code.

```
# cd git-2.0.4
# make prefix=/usr/local/git all
# make prefix=/usr/local/git install
#
# echo 'export PATH=$PATH:/usr/local/git/bin' >> /etc/bashrc
#  or
# echo 'export PATH=$PATH:/usr/local/git/bin' > /etc/profile.d/git.sh
#
# source /etc/bashrc
```

> HINT 1: Updated method of adding compiled git bin directory to bashrc. Because `echo "export PATH=$PATH:/usr/local/git/bin" >> /etc/bashrc` used "" instead of '', it would expand the current session's value for $PATH instead of keeping it as a variable, and could adversely affect the entire system. At the minimum, it should use '' instead of "" and should really be a separate script in `/etc/profile.d/`
>
> HINT 2 (@DJB): `/usr/local/git/bin` before `$PATH`, since the older version of git was already on $PATH: `export PATH=/usr/local/git/bin:$PATH`

Step 4. Check Git Version

One completion of above steps, you have successfully install Git in your system. Let use following command to check git version

```
# git --version
```

```
git version 2.0.4
```

I also wanted to add that the "Getting Started" guide at the GIT website also includes instructions on how to download and compile it yourself:

http://git-scm.com/book/en/v2/Getting-Started-Installing-Git

81

I found this nice and easy-to-follow guide on how to download the GIT source and compile it yourself (and install it). If the accepted answer does not give you the version you want, try the following instructions:

http://tecadmin.net/install-git-2-0-on-centos-rhel-fedora/

(And pasted/reformatted from above source in case it is removed later)

Step 1: Install Required Packages

Firstly we need to make sure that we have installed required packages on your system. Use following command to install required packages before compiling Git source.

Step 2: Uninstall old Git RPM

Now remove any prior installation of Git through RPM file or Yum package manager. If your older version is also compiled through source, then skip this step.

Step 3: Download and Compile Git Source

Download git source code from kernel git or simply use following command to download Git 2.5.3.

```
# cd /usr/src
# wget https://www.kernel.org/pub/software/scm/git/git-2.5.3.tar.gz
# tar xzf git-2.5.3.tar.gz
```

After downloading and extracting Git source code, Use following command to compile source code.

```
# cd git-2.5.3
# make prefix=/usr/local/git all
# make prefix=/usr/local/git install
# echo 'pathmunge /usr/local/git/bin/' > /etc/profile.d/git.sh
# chmod +x /etc/profile.d/git.sh
# source /etc/bashrc
```

Step 4. Check Git Version

On completion of above steps, you have successfully install Git in your system. Use the following command to check the git version

```
# git --version

git version 2.5.3
```

I also wanted to add that the "Getting Started" guide at the GIT website also includes instructions on how to download and compile it yourself:

http://git-scm.com/book/en/v2/Getting-Started-Installing-Git

edited Nov 3 '17 at 16:46
answered Nov 3 '14 at 1:46

Pytry
3,29711 gold badge2121 silver badges3737 bronze badges

80

Rackspace maintains the ius repository, which contains a reasonably up-to-date git, but the stock git has to first be removed.

CentOS 7 instructions:

CentOS 6 instructions:

answered Jun 30 '16 at 21:53

Mark Visser
1,9211919 silver badges1414 bronze badges

9

My personal preference is to build rpm packages for CentOS when installing non-standard software and replacing distributed components. For this I recommend that you use Mock to create a clean build environment.

The procedure is:

1. Obtain the source RPMS or a suitable SPEC file and pristine source tarball. In this case one may find source RPM packages for git2X for CentOS-6 at: `http://dl.iuscommunity.org/pub/ius/archive/CentOS/6/SRPMS/` . Packages for other CentOS releases are also available.

2. Add a rpm build user account (*do not build as root or as a real user - security issues* ***will*** *come back to bite you*).

   ```
   sudo adduser builder --home-dir /home/builder \
   --create-home --user-group --groups mock \
   --shell /bin/bash --comment "rpm package builder"
   ```

3. Next we need a build environment.

   ```
   su -l builder
   rpmdev-setuptree
   ```

   This produces the following directory structure:

   ```
   ~
   └── rpmbuild
       ├── BUILD
       ├── RPMS
       ├── SOURCES
       ├── SPECS
       └── SRPMS
   ```

4. We are using a prepared SRPMS so the SOURCES tarballs can be ignored for this case and we can go direct to SRPMS.

   ```
   wget http://dl.iuscommunity.org/pub/ius/archive/CentOS/6/SRPMS/git2u-2.5.3-1.ius.centos6.src.rpm
   \
   -O ~/rpmbuild/SRPMS/git2u-2.5.3-1.ius.centos6.src.rpm
   ```

5. Configure mock (as root)

   ```
   cd /etc/mock
   rm default.cfg
   ln -s epel-6-x86_64.cfg default.cfg
   vim default.cfg
   ```

   Disable the `beta` repos. Enable the `base` and `update` repos.

6. Initialize the build tree (/var/lib/mock is default)

   ```
   mock --init
   ```

7. If we were building from SOURCES then this is where we would employ the SPEC file and use `mock --buildsrpm . . .` . But in this case we go directly to the binary build step:

   ```
   mock --no-clean --rebuild ~/rpmbuild/SRPMS/git2u-2.5.3-1.ius.centos6.src.rpm
   ```

   This will resolve the build dependencies and download them (about 95 or so packages) into the clean build root. It will then extract the sources and build the binary from the provided SRPM and leave it in `/var/lib/mock/epel-6-x86_64/result` ; or in whatever custom build root location and architecture you provided. It will take a long time. There is a lot to this package; particularly documentation.

8. If all goes well then you should end up with a suit of RPM packages suitable for installation in place of the distro version. This is what I ended up with:

```
ll /var/lib/mock/epel-6-x86_64/result
total 34996
-rw-rw-r--. 1 byrnejb mock   448455 Oct 30 10:09 build.log
-rw-rw-r--. 1 byrnejb mock    52464 Oct 30 10:09 emacs-git2u-2.5.3-1.ius.el6.noarch.rpm
-rw-rw-r--. 1 byrnejb mock    47228 Oct 30 10:09 emacs-git2u-el-2.5.3-1.ius.el6.noarch.rpm
-rw-rw-r--. 1 byrnejb mock  8474478 Oct 30 09:57 git2u-2.5.3-1.ius.el6.src.rpm
-rw-rw-r--. 1 byrnejb mock  8877584 Oct 30 10:09 git2u-2.5.3-1.ius.el6.x86_64.rpm
-rw-rw-r--. 1 byrnejb mock    27284 Oct 30 10:09 git2u-all-2.5.3-1.ius.el6.noarch.rpm
-rw-rw-r--. 1 byrnejb mock    27800 Oct 30 10:09 git2u-bzr-2.5.3-1.ius.el6.noarch.rpm
-rw-rw-r--. 1 byrnejb mock   112564 Oct 30 10:09 git2u-cvs-2.5.3-1.ius.el6.noarch.rpm
-rw-rw-r--. 1 byrnejb mock   436176 Oct 30 10:09 git2u-daemon-2.5.3-1.ius.el6.x86_64.rpm
-rw-rw-r--. 1 byrnejb mock 15858600 Oct 30 10:09 git2u-debuginfo-2.5.3-1.ius.el6.x86_64.rpm
-rw-rw-r--. 1 byrnejb mock    60556 Oct 30 10:09 git2u-email-2.5.3-1.ius.el6.noarch.rpm
-rw-rw-r--. 1 byrnejb mock   274888 Oct 30 10:09 git2u-gui-2.5.3-1.ius.el6.noarch.rpm
-rw-rw-r--. 1 byrnejb mock    79176 Oct 30 10:09 git2u-p4-2.5.3-1.ius.el6.noarch.rpm
-rw-rw-r--. 1 byrnejb mock   483132 Oct 30 10:09 git2u-svn-2.5.3-1.ius.el6.x86_64.rpm
-rw-rw-r--. 1 byrnejb mock   173732 Oct 30 10:09 gitk2u-2.5.3-1.ius.el6.noarch.rpm
-rw-rw-r--. 1 byrnejb mock   115692 Oct 30 10:09 gitweb2u-2.5.3-1.ius.el6.noarch.rpm
-rw-rw-r--. 1 byrnejb mock    57196 Oct 30 10:09 perl-Git2u-2.5.3-1.ius.el6.noarch.rpm
-rw-rw-r--. 1 byrnejb mock    89900 Oct 30 10:09 perl-Git2u-SVN-2.5.3-1.ius.el6.noarch.rpm
-rw-rw-r--. 1 byrnejb mock   101026 Oct 30 10:09 root.log
-rw-rw-r--. 1 byrnejb mock      980 Oct 30 10:09 state.log
```

9. You will require `git2u-2.5.3-1.ius.el6.x86_64.rpm` at a minimum and such additional support packages as it requires (`perl-Git2u-2.5.3-1.ius.el6.noarch.rpm`) or you desire.

   This build has a cyclic dependency: `git2u-2.5.3-1.ius.el6.x86_64.rpm` depends upon `perl-Git2u-2.5.3-1.ius.el6.noarch.rpm` and `perl-Git2u-2.5.3-1.ius.el6.noarch.rpm` depends upon `git2u-2.5.3-1.ius.el6.x86_64.rpm` . A straight install with `rpm` will thus fail.

   There are two ways of dealing with it:

   - Install both at the same time via yum:

     ```
     yum localinstall \
       git2u-2.5.3-1.ius.el6.x86_64.rpm \
       perl-Git2u-2.5.3-1.ius.el6.noarch.rpm`
     ```

   - I am including my `LocalFile.repo` file below as it contains instructions on how to do this and provides the necessary repo file at the same time.

```
cat /etc/yum.repos.d/LocalFile.repo
# LocalFile.repo
#
#  This repo is used with a local filesystem repo.
#
# To use this repo place the rpm package in /root/RPMS/yum.repo/Packages.
# Then run: createrepo --database --update /root/RPMS/yum.repo.
#
# To use:
#  yum --enablerepo=localfile [command]
#
# or to use only ONLY this repo, do this:
#
#  yum --disablerepo=\* --enablerepo=localfile [command]

[localfile]
baseurl=file:///root/RPMS/yum.repo
name=CentOS-$releasever - Local Filesystem repo

# Before persistently enabling this repo see the priority note below.
enabled=0
gpgcheck=0

# When this repo is enabled all packages in repos with priority>5
# will not be updated even when they have a more recent version.
# Be careful with this.
priority=5
```

You also may be required to manually pre-install additional dependency packages such as `perl-TermReadKey` available from the usual repositories.

answered Oct 30 '15 at 15:24

James B. Byrne
70177 silver badges2020 bronze badges

5

To build and install modern Git on CentOS 6:

```
yum install -y curl-devel expat-devel gettext-devel openssl-devel zlib-devel gcc perl-ExtUtils-MakeMaker
export GIT_VERSION=2.6.4
mkdir /root/git
cd /root/git
wget "https://www.kernel.org/pub/software/scm/git/git-${GIT_VERSION}.tar.gz"
tar xvzf "git-${GIT_VERSION}.tar.gz"
cd git-${GIT_VERSION}
make prefix=/usr/local all
make prefix=/usr/local install
yum remove -y git
git --version # should be GIT_VERSION
```

answered Dec 16 '15 at 18:17

Joe Goggins

1,1511111 silver badges1010 bronze badges

4

as git says:

> RHEL and derivatives typically ship older versions of git. You can download a tarball and build from source, or use a 3rd-party repository such as the IUS Community Project to obtain a more recent version of git.

there is good tutorial here. in my case (Centos7 server) after install had to logout and login again.

answered Oct 21 '18 at 8:16

Alireza Saremi

8322 silver badges66 bronze badges

3

Adding a roundabout answer here. I was struggling to install git on an RHEL 6.0 Amazon instance, and what ended up saving me was ... conda, as in Ana*conda* Python.

I installed conda on the command line from the archives (code modeled after this):

```
wget http://repo.continuum.io/miniconda/Miniconda2-4.2.12-Linux-x86_64.sh -O ~/miniconda.sh
bash ~/miniconda.sh -b -p $HOME/miniconda
export PATH="$HOME/miniconda/bin:$PATH"
```

and then ran

```
conda install git
```

and a relatively recent version git was installed. Today is 12/26/2016 and the version is 2.9.3.

answered Dec 26 '16 at 22:39

Ben Ogorek
23722 silver badges1414 bronze badges

3

Build latest version of `git` on Centos 6/7

Preparing system to building rpms

1. Install epel:

   For EL6, use:

   ```
   sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-6.noarch.rpm
   ```

   For EL7, use:

   ```
   sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
   ```

2. Add yourself into group mock (you might need to re-login to server after this change):

   ```
   sudo usermod -a -G mock $USER
   ```

Download `git`

1. Download `git` sources:

   ```
   fedpkg clone -a git && cd git
   fedpkg sources
   ```

2. Verify sources:

   ```
   sha512sum -c sources
   ```

Build rpm

1. Create srmp. Use `el6` for RHEL6, `el7` for RHEL7.

   ```
   fedpkg --dist el7 srpm
   ```

2. Build package in mock:

   ```
   mock -r epel-7-x86_64 git-2.16.0-1.el7.src.rpm
   ```

3. Install latest version of `git` rpm from `/var/lib/mock/epel-7-x86_64/result/` . Note, you might need to uninstall existing version of the git from your system first.

This instruction is based on the mailing list post by Todd Zullinger.

edited Feb 9 '18 at 14:38
answered Jan 22 '18 at 2:59

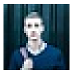
sashk
1,6772020 silver badges3131 bronze badges

1

Here's my method to install git on centos 6.

```
sudo yum groupinstall "Development Tools"
sudo yum install zlib-devel perl-ExtUtils-MakeMaker asciidoc xmlto openssl-devel curl-devel
sudo yum install wget
cd ~
wget -O git.zip https://github.com/git/git/archive/v2.7.2.zip
unzip git.zip
cd git-2.7.2
make configure
./configure --prefix=/usr/local
make all doc
sudo make install install-doc install-html
```

answered Aug 8 '16 at 19:13



Dylan Pearson
11611 silver badge1010 bronze badges

1

This may be irrelevant. It is for people don't want build the latest git on the host meanwhile they still can get the latest git.

I think most people don't like building the latest git on CentOS because the dependencies will contaminate the host and you have to run lots of commands. Therefore, I have an idea which is building git inside the Docker container and then install the executable via the docker volume mount. After that, you can delete the image and container.

Yes, the downside is you have to install docker. But the least dependencies are introduced to the host and you don't have to install other yum repo.

Here is my repository. https://github.com/wood1986/docker-library/tree/master/git

answered Jun 3 '17 at 23:08

WOW
64877 silver badges2121 bronze badges

-3

It can be very confusing, and dangerous, to replace the upstream base repositories with add-on repositories without considerable testing and thought. RPMforge, in particularly, is effectively moribund and is not getting updates.

I personally publish tools for building git 2.4, wrapped as an alternatively named "git24" package, at https://github.com/nkadel/git24-srpm/. Feel free to access and fork those if you want packages distinguished from the standard system packages, much as "samba" and "samba4" packages are differentiated.

protected by Community♦ Oct 2 '18 at 21:15

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 reputation on this site (the association bonus does not count).

Would you like to answer one of these unanswered questions instead?

Not the answer you're looking for? Browse other questions tagged linux git installation centos or ask your own question.