

Experiment No : 7

Aim : Build and install software from source code, familiarity with make and cmake utilities expected.

Step 1: Get The Server Ready

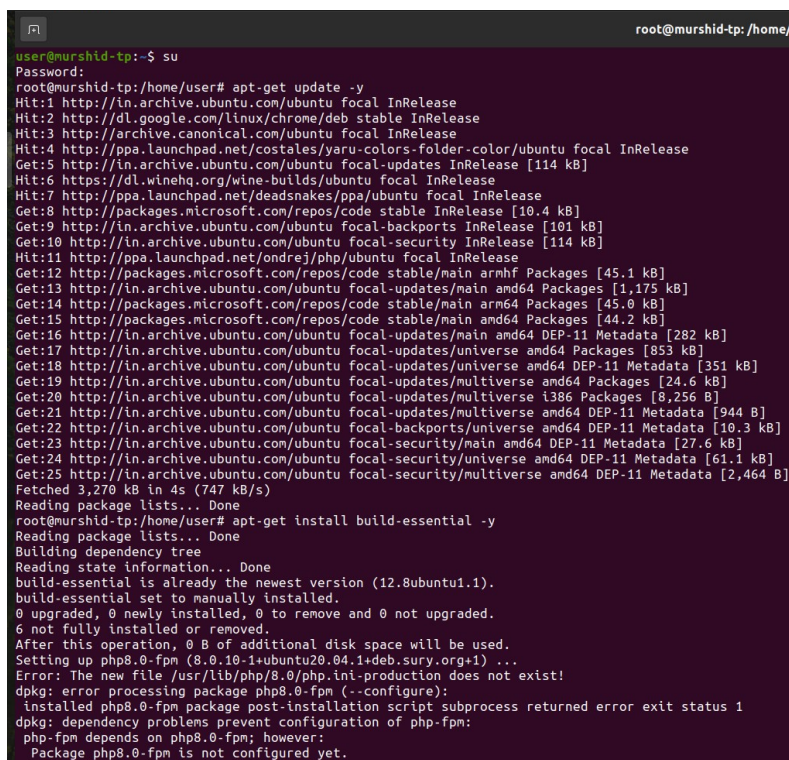
First change the user to super user by **su** command

As a best practice, make sure your packages are up to date:

apt-get update -y

Next, you'll need to make sure you have a compiler available. Run this command to install build-essential:

apt-get install build-essential -y



```
root@murshid-tp: /home/user
user@murshid-tp:~$ su
Password:
root@murshid-tp:/home/user# apt-get update -y
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://dl.google.com/linux/chrome/deb stable InRelease
Hit:3 http://archive.canonical.com/ubuntu focal InRelease
Hit:4 http://ppa.launchpad.net/costales/yaru-colors-folder-color/ubuntu focal InRelease
Get:5 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Hit:6 https://dl.winehq.org/wine-builds/ubuntu focal InRelease
Hit:7 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu focal InRelease
Get:8 http://packages.microsoft.com/repos/code stable InRelease [10.4 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:11 http://ppa.launchpad.net/ondrej/php/ubuntu focal InRelease
Get:12 http://packages.microsoft.com/repos/code stable/main armhf Packages [45.1 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1,175 kB]
Get:14 http://packages.microsoft.com/repos/code stable/main arm64 Packages [45.0 kB]
Get:15 http://packages.microsoft.com/repos/code stable/main amd64 Packages [44.2 kB]
Get:16 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [282 kB]
Get:17 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [853 kB]
Get:18 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [351 kB]
Get:19 http://in.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [24.6 kB]
Get:20 http://in.archive.ubuntu.com/ubuntu focal-updates/multiverse i386 Packages [8,256 B]
Get:21 http://in.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [944 B]
Get:22 http://in.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [10.3 kB]
Get:23 http://in.archive.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [27.6 kB]
Get:24 http://in.archive.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [61.1 kB]
Get:25 http://in.archive.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [2,464 B]
Fetched 3,270 kB in 4s (747 kB/s)
Reading package lists... Done
root@murshid-tp:/home/user# apt-get install build-essential -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.8ubuntu1.1).
build-essential set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
6 not fully installed or removed.
After this operation, 0 B of additional disk space will be used.
Setting up php8.0-fpm (8.0.10-1+ubuntu20.04.1+deb.sury.org+1) ...
Error: The new file /usr/lib/php/8.0/php.ini-production does not exist!
dpkg: error processing package php8.0-fpm (--configure):
 installed php8.0-fpm package post-installation script subprocess returned error exit status 1
dpkg: dependency problems prevent configuration of php-fpm:
 php-fpm depends on php8.0-fpm; however:
  Package php8.0-fpm is not configured yet.
```

Step 2: Download Dependencies

When installing a package from source code, you'll need to manage the installation of the package dependencies. We'll use **apt-get** to install git's dependencies:

apt install build-essential dh-autoreconf libcurl4-gnutls-dev libexpat1-dev gettext libz-dev libssl-dev -y

```
root@murshid-tp:/home/user# apt install build-essential dh-autoreconf libcurl4-gnutls-dev libexpat1-dev gettext libz-dev libssl-dev -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'zlib1g-dev' instead of 'libz-dev'
dh-autoreconf is already the newest version (19).
dh-autoreconf set to manually installed.
gettext is already the newest version (0.19.8.1-10build1).
gettext set to manually installed.
libexpat1-dev is already the newest version (2.2.9-1build1).
libexpat1-dev set to manually installed.
build-essential is already the newest version (12.8ubuntu1.1).
libssl-dev is already the newest version (1.1.1f-1ubuntu2.8).
libssl-dev set to manually installed.
zlib1g-dev is already the newest version (1:1.2.11.dfsg-2ubuntu1.2).
zlib1g-dev set to manually installed.
Suggested packages:
  libcurl4-doc libgnutls28-dev libidn11-dev libkrb5-dev libldap2-dev librtmp-dev libssh2-1-dev
The following NEW packages will be installed:
  libcurl4-gnutls-dev
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
6 not fully installed or removed.
Need to get 319 kB of archives.
After this operation, 1,526 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 libcurl4-gnutls-dev amd64 7.68.0-1ubuntu2.6 [319 kB]
Fetched 319 kB in 2s (129 kB/s)
Selecting previously unselected package libcurl4-gnutls-dev:amd64.
(Reading database ... 215333 files and directories currently installed.)
Preparing to unpack .../libcurl4-gnutls-dev_7.68.0-1ubuntu2.6_amd64.deb ...
Unpacking libcurl4-gnutls-dev:amd64 (7.68.0-1ubuntu2.6) ...
Setting up libcurl4-gnutls-dev:amd64 (7.68.0-1ubuntu2.6) ...
Setting up php8.0-fpm (8.0.10-1+ubuntu20.04.1+deb.sury.org+1) ...
Error: The new file /usr/lib/php/8.0/php.ini-production does not exist!
dpkg: error processing package php8.0-fpm (--configure):
 installed php8.0-fpm package post-installation script subprocess returned error exit status 1
dpkg: dependency problems prevent configuration of php-fpm:
 php-fpm depends on php8.0-fpm; however:
  Package php8.0-fpm is not configured yet.

dpkg: error processing package php-fpm (--configure):
 dependency problems - leaving unconfigured
Setting up libapache2-mod-php8.0 (8.0.10-1+ubuntu20.04.1+deb.sury.org+1) ...
No apport report written because the error message indicates its a followup error from a previous failure.
```

Step 3: Download The Source Package

Once the package dependencies are in place, it's time to download the package with wget:

We download git v2.23.0 by using the command

wget https://github.com/git/git/archive/v2.23.0.tar.gz

```
root@murshid-tp:/home/user# wget https://github.com/git/git/archive/v2.23.0.tar.gz
--2021-09-04 11:35:35-- https://github.com/git/git/archive/v2.23.0.tar.gz
Resolving github.com (github.com)... 13.234.176.102
Connecting to github.com (github.com)|13.234.176.102|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/git/git/tar.gz/v2.23.0 [following]
--2021-09-04 11:35:36-- https://codeload.github.com/git/git/tar.gz/v2.23.0
Resolving codeload.github.com (codeload.github.com)... 13.127.152.42
Connecting to codeload.github.com (codeload.github.com)|13.127.152.42|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/x-gzip]
Saving to: 'v2.23.0.tar.gz'

v2.23.0.tar.gz
[
<=>

2021-09-04 11:35:43 (1.44 MB/s) - 'v2.23.0.tar.gz' saved [8647535]

root@murshid-tp:/home/user#
```

Next, we need to extract the archive and cd (change directories) into the new git directory:

```
tar -xvzf v2.23.0.tar.gz
```

```
cd git-2.23.0/
```

```
root@murshid-tp:/home/user# tar -xvzf v2.23.0.tar.gz
git-2.23.0/
git-2.23.0/.clang-format
git-2.23.0/.editorconfig
git-2.23.0/.gitattributes
git-2.23.0/.github/
git-2.23.0/.github/CONTRIBUTING.md
git-2.23.0/.github/PULL_REQUEST_TEMPLATE.md
git-2.23.0/.gitignore
git-2.23.0/.gitmodules
git-2.23.0/.mailmap
git-2.23.0/.travis.yml
git-2.23.0/.tsan-suppressions
git-2.23.0/COPYING
git-2.23.0/Documentation/
git-2.23.0/Documentation/.gitattributes
git-2.23.0/Documentation/.gitignore
git-2.23.0/Documentation/CodingGuidelines
git-2.23.0/Documentation/Makefile
git-2.23.0/Documentation/MyFirstContribution.txt
git-2.23.0/Documentation/RelNotes/
git-2.23.0/Documentation/RelNotes/1.5.0.1.txt
git-2.23.0/Documentation/RelNotes/1.5.0.2.txt
git-2.23.0/Documentation/RelNotes/1.5.0.3.txt
git-2.23.0/Documentation/RelNotes/1.5.0.4.txt
git-2.23.0/Documentation/RelNotes/1.5.0.5.txt
git-2.23.0/Documentation/RelNotes/1.5.0.6.txt
git-2.23.0/Documentation/RelNotes/1.5.0.7.txt
git-2.23.0/Documentation/RelNotes/1.5.0.txt
git-2.23.0/Documentation/RelNotes/1.5.1.1.txt
git-2.23.0/Documentation/RelNotes/1.5.1.2.txt
git-2.23.0/Documentation/RelNotes/1.5.1.3.txt
git-2.23.0/Documentation/RelNotes/1.5.1.4.txt
git-2.23.0/Documentation/RelNotes/1.5.1.5.txt
git-2.23.0/Documentation/RelNotes/1.5.1.6.txt
git-2.23.0/Documentation/RelNotes/1.5.1.txt
git-2.23.0/Documentation/RelNotes/1.5.2.1.txt
git-2.23.0/Documentation/RelNotes/1.5.2.2.txt
git-2.23.0/Documentation/RelNotes/1.5.2.3.txt
git-2.23.0/Documentation/RelNotes/1.5.2.4.txt
git-2.23.0/Documentation/RelNotes/1.5.2.5.txt
git-2.23.0/Documentation/RelNotes/1.5.2.txt
git-2.23.0/Documentation/RelNotes/1.5.3.1.txt
git-2.23.0/Documentation/RelNotes/1.5.3.2.txt
git-2.23.0/Documentation/RelNotes/1.5.3.3.txt
git-2.23.0/Documentation/RelNotes/1.5.3.4.txt
git-2.23.0/Documentation/RelNotes/1.5.3.5.txt
git-2.23.0/Documentation/RelNotes/1.5.3.6.txt
git-2.23.0/Documentation/RelNotes/1.5.3.7.txt
git-2.23.0/Documentation/RelNotes/1.5.3.8.txt
```

```
git-2.23.0/xdiff-interface.c
git-2.23.0/xdiff-interface.h
git-2.23.0/xdiff/
git-2.23.0/xdiff/xdiff.h
git-2.23.0/xdiff/xdiffi.c
git-2.23.0/xdiff/xdiffi.h
git-2.23.0/xdiff/xemit.c
git-2.23.0/xdiff/xemit.h
git-2.23.0/xdiff/xhistogram.c
git-2.23.0/xdiff/xinclude.h
git-2.23.0/xdiff/xmacros.h
git-2.23.0/xdiff/xmerge.c
git-2.23.0/xdiff/xpatience.c
git-2.23.0/xdiff/xprepare.c
git-2.23.0/xdiff/xprepare.h
git-2.23.0/xdiff/xtypes.h
git-2.23.0/xdiff/xutils.c
git-2.23.0/xdiff/xutils.h
git-2.23.0/zlib.c
root@murshid-tp:/home/user# cd git-2.23.0/
root@murshid-tp:/home/user/git-2.23.0# make configure
GIT_VERSION = 2.23.0
    GEN configure
root@murshid-tp:/home/user/git-2.23.0#
```


Step 4: Install Git

Now that we have our package extracted and ready to go, we need to configure it: **make configure**

Next, let's verify that all of the dependencies necessary to build the package are available by running this command:

./configure --prefix=/usr

```
root@murshid-tp:/home/user/git-2.23.0# make configure
GIT_VERSION = 2.23.0
GEN configure
root@murshid-tp:/home/user/git-2.23.0# ./configure --prefix=/usr
configure: Setting lib to 'lib' (the default)
configure: Will try -pthread then -lpthread to enable POSIX Threads.
configure: CHECKS for site configuration
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking for size_t... yes
checking for working alloca.h... yes
checking for alloca... yes
configure: CHECKS for programs
checking whether we are using the GNU C compiler... (cached) yes
checking whether gcc accepts -g... (cached) yes
checking for gcc option to accept ISO C89... (cached) none needed
checking for inline... inline
checking if linker supports -R... no
checking if linker supports -Wl,-rpath,... yes
checking for gar... no
checking for ar... ar
checking for gtar... no
checking for tar... tar
```

After that, we'll build the source code:

make all

```
root@murshid-tp:/home/user/git-2.23.0# make all
* new build flags
CC fuzz-commit-graph.o
CC fuzz-pack-headers.o
CC fuzz-pack-idx.o
CC credential-store.o
* new link flags
CC common-main.o
CC abspath.o
CC advice.o
CC alias.o
CC alloc.o
CC apply.o
CC archive.o
CC archive-tar.o
CC archive-zip.o
CC argv-array.o
* new prefix flags
CC attr.o
CC base85.o
CC bisect.o
CC blame.o
CC blob.o
CC branch.o
CC bulk-checkin.o
CC bundle.o
CC cache-tree.o
CC chdir-notify.o
CC checkout.o
CC color.o
CC column.o
CC combine-diff.o
CC commit.o
CC commit-graph.o
CC commit-reach.o
CC compat/obstack.o
CC compat/terminal.o
CC config.o
CC connect.o
CC connected.o
CC convert.o
CC copy.o
```

Now that the binaries are all built, its time to install git:

make install

```
root@murshid-tp:/home/user/git-2.23.0# make install
SUBDIR git-gui
SUBDIR gitk-git
SUBDIR templates
install -d -m 755 '/usr/bin'
install -d -m 755 '/usr/libexec/git-core'
install git-credential-store git-daemon git-fast-import git-http-backend git-inap-send git-sh-i18n--envsubst git-shell git-remote-testsvn git-http-fetch git-http-push git-credential-store git-credential-cache--daemon git-remote-http git-remote-https git-remote-ftp git-remote-ftps git-bisect git-difftool--helper git-filter-branch git-merge-octopus git-merge-one-file git-merge-resolve git-mergetool git-quiltimport git-legacy-stash git-request-pull git-submodule git-web--browse git-add--interactive git-archimport git-cvsexportcommit git-cvsmport git-cvsserver git-send-email git-svn git-p4 git-instaweb '/usr/libexec/git-core'
install -m 644 git-mergetool--lib git-parse-remote git-rebase--preserve-merges git-sh-setup git-sh-i18n '/usr/libexec/git-core'
install git git-upload-pack git-receive-pack git-upload-archive git-shell git-cvsserver '/usr/bin'
make -C templates DESTDIR='' install
make[1]: Entering directory '/home/user/git-2.23.0/templates'
install -d -m 755 '/usr/share/git-core/templates'
(cd bit && tar cf - .) | \
(cd '/usr/share/git-core/templates' && umask 022 && tar xof -)
make[1]: Leaving directory '/home/user/git-2.23.0/templates'
install -d -m 755 '/usr/libexec/git-core/mergetools'
install -m 644 mergetools/* '/usr/libexec/git-core/mergetools'
install -d -m 755 '/usr/share/locale'
(cd po/build/locale && tar cf - .) | \
(cd '/usr/share/locale' && umask 022 && tar xof -)
install -d -m 755 '/usr/share/perl5'
(cd perl/build/lib && tar cf - .) | \
(cd '/usr/share/perl5' && umask 022 && tar xof -)
make -C gitweb install
make[1]: Entering directory '/home/user/git-2.23.0/gitweb'
make[2]: Entering directory '/home/user/git-2.23.0'
make[2]: 'GIT-VERSION-FILE' is up to date.
make[2]: Leaving directory '/home/user/git-2.23.0'
GEN gitweb.cgi
GEN static/gitweb.js
install -d -m 755 '/usr/share/gitweb'
install -m 755 gitweb.cgi '/usr/share/gitweb'
install -d -m 755 '/usr/share/gitweb/static'
install -m 644 static/gitweb.js static/gitweb.css static/git-logo.png static/git-favicon.png '/usr/share/gitweb/static'
make[1]: Leaving directory '/home/user/git-2.23.0/gitweb'
make -C gitk-git install
make[1]: Entering directory '/home/user/git-2.23.0/gitk-git'
install -d -m 755 '/usr/bin'
install -m 755 gitk-wish '/usr/bin'/gitk
install -d -m 755 '/usr/share/gitk/lib/msgsrc'
install -m 644 po/pt_br.msg '/usr/share/gitk/lib/msgsrc' && install -m 644 po/de.msg '/usr/share/gitk/lib/msgsrc' && install -m 644 po/ja.msg '/usr/share/gitk/lib/msgsrc' && install -m 644 po/ca.msg '/usr/share/gitk/lib/msgsrc' && install -m 644 po/sv.msg '/usr/share/gitk/lib/msgsrc' && install -m 644 po/it.msg '/usr/share/gitk/lib/msgsrc' && install -m 644 po/ru.msg '/usr/share/gitk/lib/msgsrc' && install -m 644 po/pt_pt.msg '/usr/share/gitk/lib/msgsrc' && install -m 644 po/hu.msg '/usr/share/gitk/lib/msgsrc' && install -m 644 po/vi.msg '/usr/share/gitk/lib/msgsrc' && install -m 644 po/bg.msg '/usr/share/gitk/lib/msgsrc' && install -m 644 po/es.msg '/usr/share/gitk/lib/msgsrc' && install -m 644 po/fr.msg '/usr/share/gitk/lib/msgsrc' && true
make[1]: Leaving directory '/home/user/git-2.23.0/gitk-git'
make -C git-gui gitexecdir='/usr/libexec/git-core' install
```

That's it! The last thing to do is to verify that git is working:

git --version

```
remote_curl_aliases="git-remote-https git-remote-ftp git-remote-ftps" && \
for p in $remote_curl_aliases; do \
    rm -f "$execdir/$p" && \
    test -n "" && \
    ln -s "git-remote-http" "$execdir/$p" || \
    { test -z "" && \
    ln "$execdir/git-remote-http" "$execdir/$p" 2>/dev/null || \
    ln -s "git-remote-http" "$execdir/$p" 2>/dev/null || \
    cp "$execdir/git-remote-http" "$execdir/$p" || exit; } \
done && \
./check_bindir "z$bindir" "z$execdir" "$bindir/git-add"
root@murshid-tp:/home/user/git-2.23.0# git --version
git version 2.23.0
root@murshid-tp:/home/user/git-2.23.0#
```