

Install the Ubuntu on the Virtual Machine

Create Bitcoin coin local copy of source code with:

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

```
Kneron@ubuntu:~/bitcoin-blockchain$ git clone https://github.com/bitcoin/bitcoin.git
Cloning into 'bitcoin'...
remote: Enumerating objects: 198185, done.
remote: Counting objects: 100% (171/171), done.
remote: Compressing objects: 100% (99/99), done.
remote: Total 198185 (delta 109), reused 122 (delta 72), pack-reused 198014
Receiving objects: 100% (198185/198185), 165.89 MiB | 4.59 MiB/s, done.
Resolving deltas: 100% (141508/141508), done.
```

Use `cd` to get in `bitcoin` file

```
Kneron@ubuntu:~/bitcoin-blockchain$ la
bitcoin
Kneron@ubuntu:~/bitcoin-blockchain$ cd bitcoin/
Kneron@ubuntu:~/bitcoin-blockchain/bitcoin$ ls
autogen.sh  configure.ac  depends  Makefile.am  share
build-aux   contrib      doc      README.md    src
build_msvc  CONTRIBUTING.md  INSTALL.md  REVIEWERS    test
ci          COPYING      libbitcoinconsensus.pc.in  SECURITY.md
Kneron@ubuntu:~/bitcoin-blockchain/bitcoin$ ls -lrt
total 184
-rw-r--r-- 1 Kneron Kneron 161 Jun 15 20:04 INSTALL.md
-rw-r--r-- 1 Kneron Kneron 1142 Jun 15 20:04 COPYING
-rw-r--r-- 1 Kneron Kneron 21375 Jun 15 20:04 CONTRIBUTING.md
-rw-r--r-- 1 Kneron Kneron 850 Jun 15 20:04 SECURITY.md
-rw-r--r-- 1 Kneron Kneron 4548 Jun 15 20:04 REVIEWERS
-rw-r--r-- 1 Kneron Kneron 3570 Jun 15 20:04 README.md
-rw-r--r-- 1 Kneron Kneron 13827 Jun 15 20:04 Makefile.am
drwxr-xr-x 3 Kneron Kneron 4096 Jun 15 20:04 build-aux
-rwxr-xr-x 1 Kneron Kneron 546 Jun 15 20:04 autogen.sh
-rw-r--r-- 1 Kneron Kneron 76406 Jun 15 20:04 configure.ac
drwxr-xr-x 5 Kneron Kneron 4096 Jun 15 20:04 ci
drwxr-xr-x 26 Kneron Kneron 4096 Jun 15 20:04 build_msvc
drwxr-xr-x 20 Kneron Kneron 4096 Jun 15 20:04 contrib
drwxr-xr-x 6 Kneron Kneron 4096 Jun 15 20:04 depends
-rw-r--r-- 1 Kneron Kneron 267 Jun 15 20:04 libbitcoinconsensus.pc.in
drwxr-xr-x 4 Kneron Kneron 4096 Jun 15 20:04 doc
drwxr-xr-x 6 Kneron Kneron 4096 Jun 15 20:04 share
drwxr-xr-x 27 Kneron Kneron 4096 Jun 15 20:04 src
drwxr-xr-x 7 Kneron Kneron 4096 Jun 15 20:04 test
Kneron@ubuntu:~/bitcoin-blockchain/bitcoin$
```

Checkout the latest release:

`git checkout v0.19.0`

`git status`

```
Kneron@ubuntu:~/bitcoin-blockchain/bitcoin$ git checkout v0.19.0
Note: checking out 'v0.19.0'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

    git checkout -b <new-branch-name>

HEAD is now at 42414bfd0 build: set CLIENT_VERSION_RC to 0 pre-final
Kneron@ubuntu:~/bitcoin-blockchain/bitcoin$ get status
Command 'get' not found, but there are 18 similar ones.

Kneron@ubuntu:~/bitcoin-blockchain/bitcoin$ git status
HEAD detached at v0.19.0
nothing to commit, working tree clean
```

Install require packages:

`sudo apt install autoconf autogen libtool`

```
Kneron@ubuntu:~/bitcoin-blockchain/bitcoin$ sudo apt install autoconf autogen libtool
Reading package lists... Done
Building dependency tree
Reading state information... Done
autoconf is already the newest version (2.69-11).
libtool is already the newest version (2.4.6-2).
autogen is already the newest version (1:5.18.12-4).
The following packages were automatically installed and are no longer required:
  libhttp-parser2.7.1 nodejs-doc
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 382 not upgraded.
Kneron@ubuntu:~/bitcoin-blockchain/bitcoin$ ./autogen.sh
libtoolize: putting auxiliary files in AC_CONFIG_AUX_DIR, 'build-aux'.
libtoolize: copying file 'build-aux/ltmain.sh'
libtoolize: putting macros in AC_CONFIG_MACRO_DIRS, 'build-aux/m4'.
libtoolize: copying file 'build-aux/m4/libtool.m4'
libtoolize: copying file 'build-aux/m4/ltoptions.m4'
libtoolize: copying file 'build-aux/m4/ltugar.m4'
libtoolize: copying file 'build-aux/m4/ltversion.m4'
libtoolize: copying file 'build-aux/m4/lt~obsolete.m4'
configure.ac:45: installing 'build-aux/compile'
configure.ac:45: installing 'build-aux/config.guess'
configure.ac:45: installing 'build-aux/config.sub'
configure.ac:28: installing 'build-aux/install-sh'
configure.ac:28: installing 'build-aux/missing'
Makefile.am: installing 'build-aux/depcomp'
```

Then run with: `./autogen.sh`

```
LibreOffice Writer installing 'build-aux/compile'
configure.ac:5: installing 'build-aux/config.guess'
configure.ac:5: installing 'build-aux/config.sub'
configure.ac:9: installing 'build-aux/install-sh'
configure.ac:9: installing 'build-aux/missing'
Makefile.am: installing 'build-aux/depcomp'
parallel-tests: installing 'build-aux/test-driver'
libtoolize: putting auxiliary files in AC_CONFIG_AUX_DIR, 'build-aux'.
libtoolize: copying file 'build-aux/ltmain.sh'
libtoolize: putting macros in AC_CONFIG_MACRO_DIRS, 'build-aux/m4'.
libtoolize: copying file 'build-aux/m4/libtool.m4'
libtoolize: copying file 'build-aux/m4/ltoptions.m4'
libtoolize: copying file 'build-aux/m4/ltugar.m4'
libtoolize: copying file 'build-aux/m4/ltversion.m4'
libtoolize: copying file 'build-aux/m4/lt~obsolete.m4'
configure.ac:80: installing 'build-aux/compile'
configure.ac:30: installing 'build-aux/config.guess'
configure.ac:30: installing 'build-aux/config.sub'
configure.ac:40: installing 'build-aux/install-sh'
configure.ac:40: installing 'build-aux/missing'
src/Makefile.am: installing 'build-aux/depcomp'
parallel-tests: installing 'build-aux/test-driver'
Kneron@ubuntu:~/bitcoin-blockchain/bitcoin$
```

Install the miss libraries with:

`sudo apt-get install gcc g++`

`sudo apt-get install build-essential`

`sudo apt-get install pkg-config`

`sudo apt-get install libdb`


```
sudo apt-get install libdb4.8-dev libdb4.8++-dev
sudo apt-get update
sudo apt-get install libboost-all-dev
sudo apt-get install openssl libssl-dev libcurl4-openssl-dev
sudo apt-get install libevent-dev
```

To configure run with: `./configure`

```
Kneron@ubuntu:~/bitcoin-blockchain/bitcoin$ ./configure
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking whether make supports nested variables... (cached) yes
checking for g++... g++
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 g++ accepts -g... yes
checking for style of include used by make... GNU
checking dependency style of g++... gcc3
checking whether g++ supports C++11 features with -std=c++11... yes
checking whether std::atomic can be used without link library... yes
checking whether we are using the GNU Objective C++ compiler... no
checking whether g++ -std=c++11 accepts -g... no
checking dependency style of g++ -std=c++11... gcc3
checking how to print strings... printf
checking for gcc... gcc
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 whether gcc understands -c and -o together... yes
checking dependency style of gcc... gcc3
checking for a sed that does not truncate output... /bin/sed
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for fgrep... /bin/grep -F
checking for ld used by gcc... /usr/bin/ld
checking if the linker (/usr/bin/ld) is GNU ld... yes
checking for BSD- or MS-compatible name lister (nm)... /usr/bin/nm -B
checking the name lister (/usr/bin/nm -B) interface... BSD nm
checking whether ln -s works... yes
checking the maximum length of command line arguments... 1572864
checking how to convert x86_64-pc-linux-gnu file names to x86_64-pc-linux-gnu format... func_conv
```

After running completely, run with: **make**

```
Kneron@ubuntu:~/bitcoin-blockchain/bitcoin$ make
Making all in src
make[1]: Entering directory '/home/Kneron/bitcoin-blockchain/bitcoin/src'
make[2]: Entering directory '/home/Kneron/bitcoin-blockchain/bitcoin/src'
CXX      support/libbitcoinconsensus_la-cleanse.lo
CXX      crypto/libbitcoinconsensus_la-aes.lo
CXX      crypto/libbitcoinconsensus_la-chacha_poly_aead.lo
CXX      crypto/libbitcoinconsensus_la-chacha20.lo
CXX      crypto/libbitcoinconsensus_la-hkdf_sha256_32.lo
Terminal crypto/libbitcoinconsensus_la-hmac_sha256.lo
CXX      crypto/libbitcoinconsensus_la-hmac_sha512.lo
CXX      crypto/libbitcoinconsensus_la-poly1305.lo
CXX      crypto/libbitcoinconsensus_la-ripemd160.lo
CXX      crypto/libbitcoinconsensus_la-sha1.lo
CXX      crypto/libbitcoinconsensus_la-sha256.lo
CXX      crypto/libbitcoinconsensus_la-sha512.lo
CXX      crypto/libbitcoinconsensus_la-siphash.lo
CXX      crypto/libbitcoinconsensus_la-sha256_sse4.lo
CXX      libbitcoinconsensus_la-arith_uint256.lo
CXX      consensus/libbitcoinconsensus_la-merkle.lo
CXX      consensus/libbitcoinconsensus_la-tx_check.lo
CXX      libbitcoinconsensus_la-hash.lo
CXX      primitives/libbitcoinconsensus_la-block.lo
CXX      primitives/libbitcoinconsensus_la-transaction.lo
CXX      libbitcoinconsensus_la-pubkey.lo
CXX      script/libbitcoinconsensus_la-bitcoinconsensus.lo
CXX      script/libbitcoinconsensus_la-interpreter.lo
CXX      script/libbitcoinconsensus_la-script.lo
CXX      script/libbitcoinconsensus_la-script_error.lo
CXX      libbitcoinconsensus_la-uint256.lo
CXX      util/libbitcoinconsensus_la-strencodings.lo
make[3]: Entering directory '/home/Kneron/bitcoin-blockchain/bitcoin/src/secp256k1'
gcc -I. -g -O2 -Wall -Wextra -Wno-unused-function -c src/gen_context.c -o gen_context.o
gcc -g -O2 -Wall -Wextra -Wno-unused-function gen_context.o -o gen_context
./gen_context
CC      src/libsecp256k1_la-secp256k1.lo
CCLD    libsecp256k1.la
/usr/bin/ar: `u' modifier ignored since `D' is the default (see `U')
make[3]: Leaving directory '/home/Kneron/bitcoin-blockchain/bitcoin/src/secp256k1'
CXXLD    libbitcoinconsensus.la
/usr/bin/ar: `u' modifier ignored since `D' is the default (see `U')
CXX      bitcoind-bitcoind.o
CXX      libbitcoin_server_a-addrdb.o
CXX      libbitcoin_server_a-addrman.o
CXX      libbitcoin_server_a-banman.o
CXX      libbitcoin_server_a-blockencodings.o
CXX      libbitcoin_server_a-blockfilter.o
CXX      libbitcoin_server_a-chain.o
```

Install the bitcoind with: **sudo make install**

```
Kneron@ubuntu:~/bitcoin-blockchain/bitcoin$ sudo make install
Making install in src
make[1]: Entering directory '/home/Kneron/bitcoin-blockchain/bitcoin/src'
make[2]: Entering directory '/home/Kneron/bitcoin-blockchain/bitcoin/src'
make[3]: Entering directory '/home/Kneron/bitcoin-blockchain/bitcoin'
make[3]: Leaving directory '/home/Kneron/bitcoin-blockchain/bitcoin'
make[3]: Entering directory '/home/Kneron/bitcoin-blockchain/bitcoin/src'
make[4]: Entering directory '/home/Kneron/bitcoin-blockchain/bitcoin'
make[4]: Leaving directory '/home/Kneron/bitcoin-blockchain/bitcoin'
/bin/mkdir -p '/usr/local/lib'
/bin/bash ../libtool --preserve-dup-deps --mode=install /usr/bin/install -c libbitcoinconsensus.so.0.0.0 '/usr/local/lib'
libtool: install: /usr/bin/install -c .libs/libbitcoinconsensus.so.0.0.0 /usr/local/lib/libbitcoinconsensus.so.0.0.0
libtool: install: (cd /usr/local/lib && { ln -s -f libbitcoinconsensus.so.0.0.0 libbitcoinconsensus.so.0 || { rm -f libbitcoinconsensus.so.0 && ln -s libbitcoinconsensus.so.0.0.0 libbitcoinconsensus.so.0; }; })
libtool: install: (cd /usr/local/lib && { ln -s -f libbitcoinconsensus.so.0.0.0 libbitcoinconsensus.so || { rm -f libbitcoinconsensus.so && ln -s libbitcoinconsensus.so.0.0.0 libbitcoinconsensus.so; }; })
libtool: install: /usr/bin/install -c .libs/libbitcoinconsensus.lai /usr/local/lib/libbitcoinconsensus.la
libtool: install: /usr/bin/install -c .libs/libbitcoinconsensus.a /usr/local/lib/libbitcoinconsensus.a
libtool: install: chmod 644 /usr/local/lib/libbitcoinconsensus.a
libtool: install: /usr/bin/ranlib /usr/local/lib/libbitcoinconsensus.a
libtool: finish: PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin" ldconfig -n /usr/local/lib
-----
Libraries have been installed in:
  /usr/local/lib

If you ever happen to want to link against installed libraries
in a given directory, LIBDIR, you must either use libtool, and
specify the full pathname of the library, or use the '-LLIBDIR'
flag during linking and do at least one of the following:
- add LIBDIR to the 'LD_LIBRARY_PATH' environment variable
  during execution
- add LIBDIR to the 'LD_RUN_PATH' environment variable
  during linking
- use the '-D_LIBTOOL_IS_A_FOOL__' linker flag
- have your system administrator add LIBDIR to '/etc/ld.so.conf'

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
-----
/bin/mkdir -p '/usr/local/bin'
```

in order to check a certain block information, we first run with:

bitcoind --daemon

```
Kneron@ubuntu:~/bitcoin-blockchain/bitcoin$ bitcoind -daemon
Bitcoin Core starting
Kneron@ubuntu:~/bitcoin-blockchain/bitcoin$
```



```
bitcoin-cli getblockchaininfo
```

```
{
  "version": 190000,
  "protocolversion": 70015,
  "blocks": 0,
  "timeoffset": 0,
  "connections": 10,
  "proxy": "",
  "difficulty": 1,
  "chain": "main",
  "walletversion": 169900,
  "balance": 0.00000000,
  "keypoololdest": 1623819082,
  "keypoolsize": 1000,
  "paytxfee": 0.00000000,
  "relayfee": 0.00001000,
  "warnings": ""
}
```

[illegible]