

# Linux Requirements

## Guide

This article is part of the [Installation Guide](#). You can read it alone or click on the previous link to easily move between the steps.

[Step 2: Core Installation >>](#)

Processor with SSE2 support  
Boost 1.58  
MySQL 5.6.0 up to 5.7.0  
OpenSSL 1.0.x  
CMake 3.2.0  
Clang 3.5 (heavy recommended, especially on master branch) or GCC 6.3.0  
zlib 1.2.7

### Note:

While compiling you may get one error like: "c++: internal compiler error: Killed (program cc1plus)" the reasons of this can be:

Low ram/swap amount: increase ram/swap to a minimum of 2GB of ram and 2GB of swap or decrease the amount of make -j to 1 (more concurrent compile threads = more memory usage). (you can get this using VPS servers)

SELinux/grsecurity/Hardened kernel: [Kernels that use ASLR as a security measure tend to mess up GCC's precompiled header implementation](#). Try using an unhardened kernel (without ASLR), or compiling using clang, or gcc without pch. (you can get this issue when using OVH hosting).

### Hint

(Master only) If you plan to run compile and run TrinityCore on a Linux machine and the World of Warcraft client on a separate Windows PC, you will also need to compile it on the Window PC, so that you have a Windows "connection\_patcher.exe" binary, which needs to be run on the machine where the client will run, you also can download pre-compiled patcher from <https://ci.appveyor.com/project/DDuarte/trinitycore/branch/master> (you can also get tools, but they are compiled on debug mode and they are really slow to extract maps)

**Debian based distributions (heavy recommended debian stable, it's the distribution we use to set minimum requirements)**

**Recommendation:** Use apt-get with stable source list instead of install packages. We recommend the latest stable version of your distribution. Avoid LTS versions since we usually update requirements to the latest stable Debian version. Mixing stable with experimental packages may break your linux OS.

Debian 9.x (you will need to use su to install the packages)

```
apt-get install git clang cmake make gcc g++ libmariadbclient-dev  
libssl1.0-dev libbz2-dev libreadline-dev libncurses-dev libboost-all-dev  
mariadb-server p7zip  
update-alternatives --install /usr/bin/cc cc /usr/bin/clang 100  
update-alternatives --install /usr/bin/c++ c++ /usr/bin/clang 100
```

Ubuntu 17.10 (you will need to use sudo to install the packages).

```
apt-get install git clang cmake make gcc g++ libmysqlclient-dev
libssl-dev libbz2-dev libreadline-dev libncurses-dev libboost-all-dev
mysql-server p7zip
update-alternatives --install /usr/bin/cc cc /usr/bin/clang 100
update-alternatives --install /usr/bin/c++ c++ /usr/bin/clang 100
```

**Not supported:** Debian 8 or lower. Ubuntu 16.04 or lower.

AVOID UBUNTU LTS versions.

### Fedora based distributions

Tested on Fedora Server 27/28 (Fedora Workstation 27/28 should also work).

```
dnf install
https://dev.mysql.com/get/mysql57-community-release-fc27-1.noarch.rpm
dnf install git clang cmake make gcc gcc-c++ community-mysql-devel
compat-openssl10-devel bzip2-devel readline-devel ncurses-devel
boost-devel community-mysql-server p7zip
rm -f /usr/bin/c++
update-alternatives --install /usr/bin/cc cc /usr/bin/clang 100
update-alternatives --install /usr/bin/c++ c++ /usr/bin/clang 100
```

### ▼ Red Hat based distributions (NOT RECOMMENDED)

#### Red Hat based distributions

```
yum install epel-release
yum install git cmake3 make clang mariadb-devel openssl-devel
bzip2-devel readline-devel ncurses-devel
ln -s /usr/bin/cmake3 /usr/bin/cmake

yum install libquadmath-devel python-devel
curl -L
https://dl.bintray.com/boostorg/release/1.64.0/source/boost_1_64_0.tar
.gz -o boost_1_64_0.tar.gz
tar -zxvf boost_1_64_0.tar.gz
cd boost_1_64_0
./bootstrap.sh
./b2 install

yum install mariadb-server p7zip
```

**Note:** You will only have to compile the boost library one time, unless you update your kernel or update certain security packages. You will also need to update boost-devel. The developer libraries will conflict with the compiled version of boost on Red Hat distros. Also be sure to install boost-devel **after** compiling latest version of boost.

**Note:** Some distribution versions might not match our requirements for CMake. If you can't install the EPEL repository on your build server, use the following instructions to install CMake manually.

```
curl https://cmake.org/files/v3.6/cmake-3.6.1.tar.gz -o
cmake-3.6.1.tar.gz
tar -zxvf cmake-3.6.1.tar.gz
cd cmake-3.6.1
./bootstrap
make
make install
```

## Help

If you still have any problem, check:

- [Updating or starting with TrinityCore issues, Trouble with your TrinityCore Install / Readme 1st / FAQs](#)
- [Ask help on the Forum](#)
- If you still have problems, you can try to ask help on [IRC](#), but remember it's not real time 24/7 support, most of people there lives on GMT and they can be sleeping or working.

### Guide

This article is part of the [Installation Guide](#). You can read it alone or click on the previous link to easily move between the steps.

[Step 2: Core Installation >>](#)