

ClickHouse 在 CentOS7 下的编译

作者: [flowaters](#)

原文链接: <https://ld246.com/article/1513442438997>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

背景

ClickHouse是目前很热的一款OLAP产品。

为了深入了解ClickHouse，从源码级别来进行从头编译是必须的。

编译ClickHouse

准备编译环境

准备虚拟机

安装虚拟机VirtualBox

安装CentOS7

1. 下载CentOS7镜像
2. 安装镜像，内存8G，硬盘60G.
3. 选择最小安装

编译

下载最新稳定压缩包(失败)

目前是54318, <https://github.com/yandex/ClickHouse/archive/v1.1.54318-stable.zip>

解压安装包

```
sudo yum install unzip -y
unzip ClickHouse-1.1.54318-stable.zip
```

clone稳定包

安装git

```
sudo yum install git -y
```

```
git clone -b 'v1.1.54318-stable' --recursive https://github.com/yandex/ClickHouse.git
```

或者

```
git clone -b stable --recursive https://github.com/yandex/ClickHouse.git
git submodule update --init --recursive
```

编写时的最新稳定版本为

```
git clone -b 'v1.1.54318-stable' --recursive https://github.com/yandex/ClickHouse.git
```

clone主干包

安装git

```
sudo yum install git -y
```

```
git clone --recursive https://github.com/yandex/ClickHouse.git
```

或者

```
git clone https://github.com/yandex/ClickHouse.git  
git submodule update --init --recursive
```

安装gcc7

安装scl源

```
sudo yum install centos-release-scl -y
```

安装 gcc7

```
sudo yum install devtoolset-7-gcc -y  
sudo yum install devtoolset-7-gcc-c++ -y  
source /opt/rh/devtoolset-7/enable
```

确认下版本号为 7.2.1

```
[abeffect@localhost ~]$ gcc --version  
gcc (GCC) 7.2.1 20170829 (Red Hat 7.2.1-1)  
Copyright (C) 2017 Free Software Foundation, Inc.  
This is free software; see the source for copying conditions. There is NO  
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

安装cmake3

安装epel源

```
sudo yum install epel-release -y
```

安装cmake3

```
sudo yum install cmake3 -y
```

安装openssl

```
sudo yum install openssl-devel -y
```

cmake编译

进入目录

```
cd ClickHouse
```

```
mkdir build
cd build/
```

执行cmake

```
cmake3 ..
```

如果出现了

```
CMake Error in contrib/librdkafka/src/CMakeLists.txt:
  Imported target "OpenSSL::SSL" includes non-existent path

    "/usr/local/opt/openssl/include"
```

的异常

再执行一次cmake

```
cmake3 ..
```

就成功了

编译

```
make
```

或者想更快的编译

```
export THREADS=$(grep -c ^processor /proc/cpuinfo)
make -j $THREADS
```

后记

1. 编译时，需要目录剩余空间有20G+
2. 编译出来的二进制文件有500M+，但是参考[4]中的文件仅有65M.

源安装ClickHouse

Altinity给出了[编译好的ClickHouse镜像源](#)，版本也是最新的稳定版，经本人测试，下载安装正常。

EL6

编辑文件 /etc/yum.repos.d/altinity_clickhouse.repo，内容如下

```
[altinity_clickhouse]
name=altinity_clickhouse
baseurl=https://packagecloud.io/altinity/clickhouse/el/6/$basearch
repo_gpgcheck=1
gpgcheck=0
enabled=1
gpgkey=https://packagecloud.io/altinity/clickhouse/gpgkey
sslverify=1
```

```
sslcert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300
```

```
[altinity_clickhouse-source]
name=altinity_clickhouse-source
baseurl=https://packagecloud.io/altinity/clickhouse/el/6/SRPMS
repo_gpgcheck=1
gpgcheck=0
enabled=1
gpgkey=https://packagecloud.io/altinity/clickhouse/gpgkey
sslverify=1
sslcert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300
```

安装

```
sudo yum install clickhouse-server clickhouse-client -y
```

EL7

编辑文件 /etc/yum.repos.d/altinity_clickhouse.repo，内容如下

```
[altinity_clickhouse]
name=altinity_clickhouse
baseurl=https://packagecloud.io/altinity/clickhouse/el/7/$basearch
repo_gpgcheck=1
gpgcheck=0
enabled=1
gpgkey=https://packagecloud.io/altinity/clickhouse/gpgkey
sslverify=1
sslcert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300
```

```
[altinity_clickhouse-source]
name=altinity_clickhouse-source
baseurl=https://packagecloud.io/altinity/clickhouse/el/7/SRPMS
repo_gpgcheck=1
gpgcheck=0
enabled=1
gpgkey=https://packagecloud.io/altinity/clickhouse/gpgkey
sslverify=1
sslcert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300
```

安装

```
sudo yum install clickhouse-server clickhouse-client -y
```

会发现，依赖的库有：

```
unixODBC
clickhouse-server-common
```

后记二

20180723补充, yandex官方clickhouse源http://repo.yandex.ru/clickhouse/rpm/stable/x86_64/

参考

1. [red soft ru build packages](#): 好久不更新了
2. [How to build ClickHouse on Linux](#): 官方文档
3. [Altinity clickhouse-rpm](#): v目录下有完整的编译脚本
4. [Altinity clickhouse-rpm-install](#): 其实不用自己编译, 直接使用这里的源安装即可