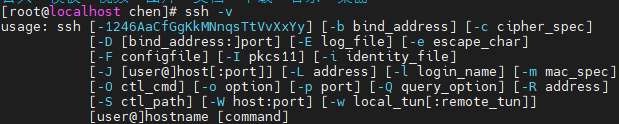
## 开启远程root登陆以及Vscode远程ssh连接root用户

## **1.检查ssh是否安装**

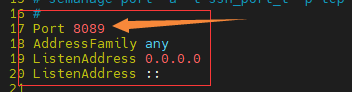
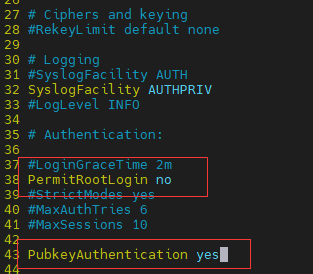
ssh -v



表示已经安装成功

## **进入**[ssh配置](https://so.csdn.net/so/search?q=ssh%E9%85%8D%E7%BD%AE&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/hyf112233/article/details/_blank)**文件sshd\_config**

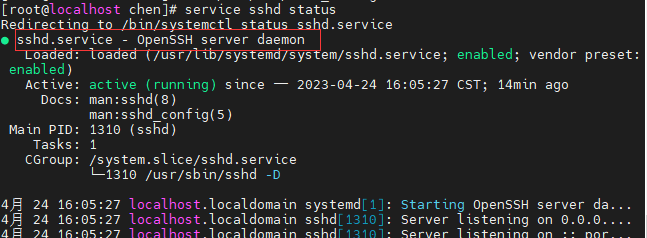
vi /etc/ssh/[sshd\_config](https://so.csdn.net/so/search?q=sshd_config&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/hyf112233/article/details/_blank)

 端口为22：最好改下

yes,运行远程登录

**3. 查看ssh 服务状态**

service sshd status



此时处于开启状态

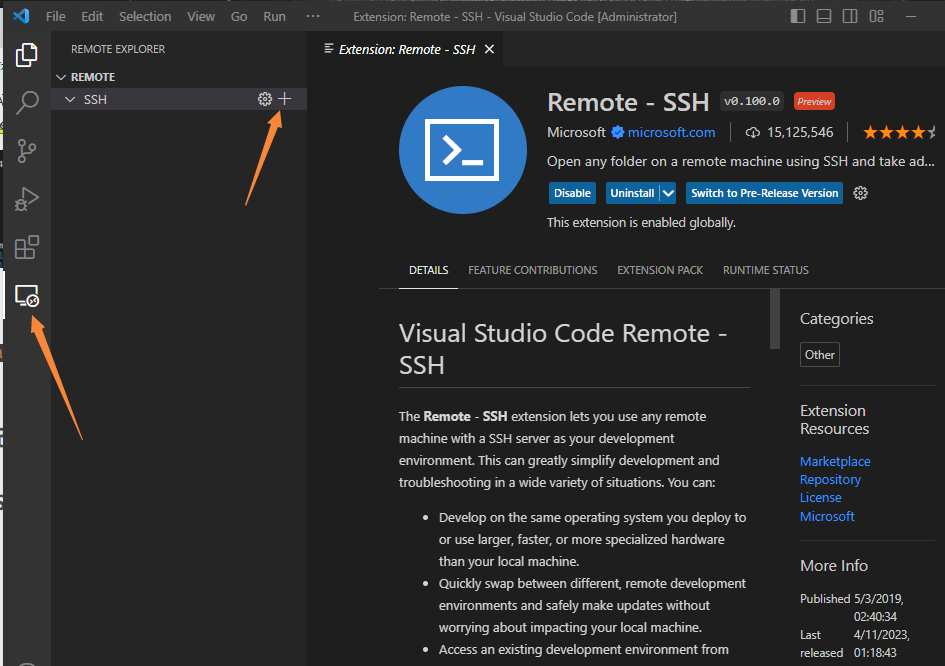
**4.启动ssh服务**

service sshd start  / systemctl start sshd.service

状态为running表示服务已经启动，此时能够进行远程root访问

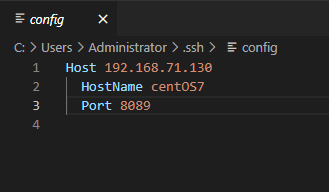
# **Vscode远程ssh连接配置**

## **Vscode安装Remote-SSH插件**

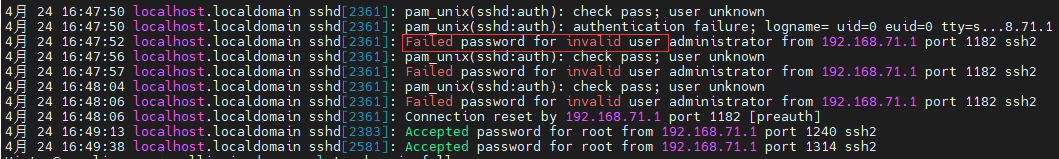


在弹出框中输入centOS的IP

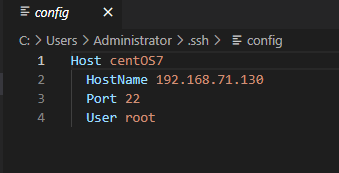
设置配置文件:



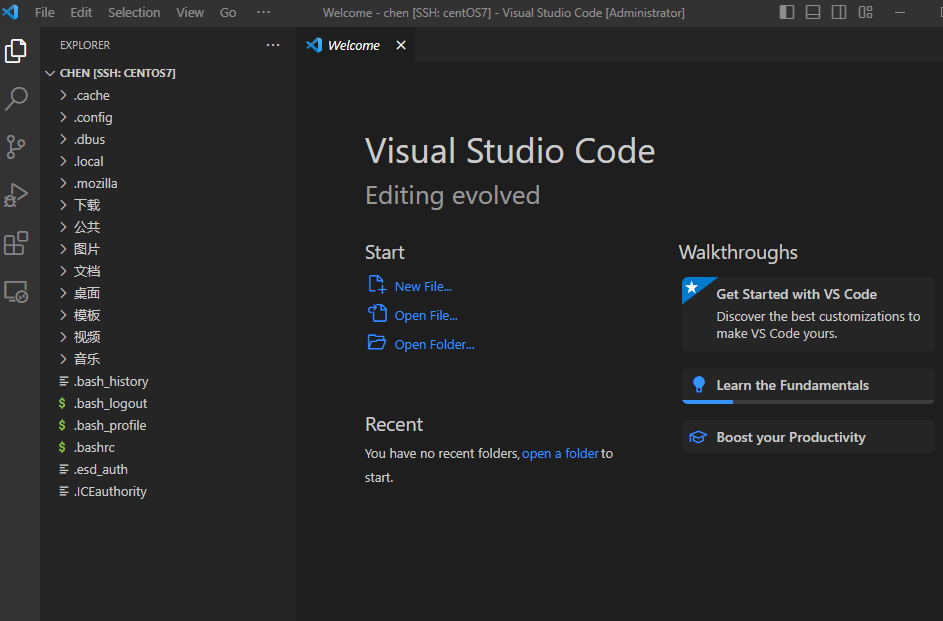
保存退出,如果没有看到连接则重启vscode



没有设置登录用户,应该是以当前用户登录了



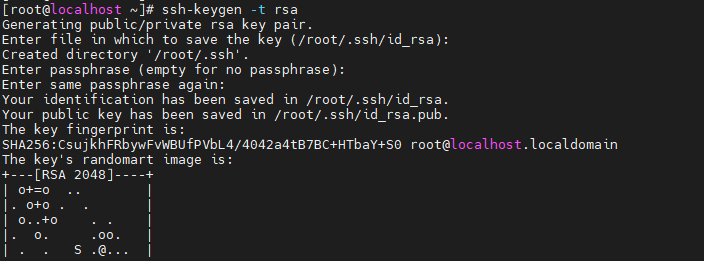
再次登录，输入密码，选择文件夹，输入密码



vscode免密登录centOS

**在centOS生成公钥+私钥**

ssh-keygen -t rsa



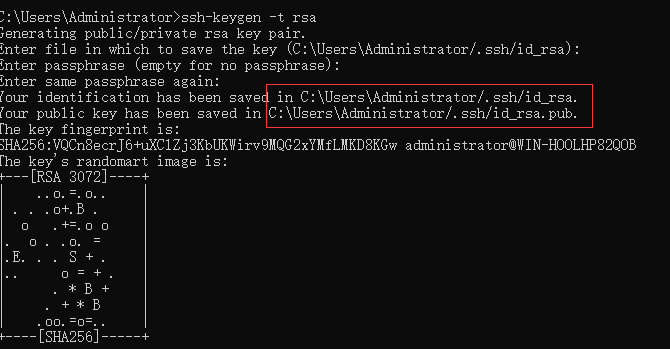
/root/.ssh/id\_rsa.pub.

/root/.ssh/id\_rsa.

创建tuoch /root/.ssh/****authorized\_keys****

### **在windows端**

1. ****win+R打开cmd(命令提示符)****  
   ****2、依次输入如下命令****



****打开.ssh/id\_rsa.pub****，可用记事本打开，然后****复制其中的内容****

****将在windows下复制的.ssh/id\_rsa.pub中的内容粘贴到linux端下的authorized\_keys中****

****ssh-rsa  administrator@WIN-HOOLHP82QOB****

****------------------------------------------------------------------------------------------------------****

****centOS安装C++基础开发环境****

1. ****安装****zlib --> 解压包需要用到

yum install zlib zlib-devel

2.下载 gcc8.3.0的源码, 并解压：

wget <http://ftp.gnu.org/gnu/gcc/gcc-8.3.0/gcc-8.3.0.tar.gz>

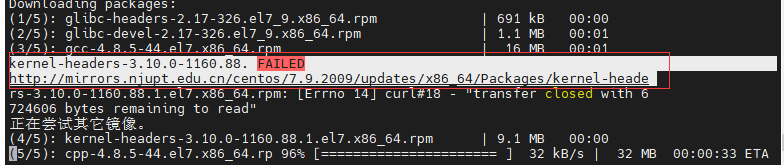
tar -xvf gcc-8.3.0.tar.gz

（或者直接直接使用yum install gcc-g++）

这里直接使用 yum install gcc-g++

gcc -v

g++ -v



我们先换成国内的源，然后再进行安装

wget -O /etc/yum.repos.d/CentOS-Base.repo https://mirrors.aliyun.com/repo/Centos-7.repo

yum makecache

sed -i -e '/mirrors.cloud.aliyuncs.com/d' -e '/mirrors.aliyuncs.com/d' /etc/yum.repos.d/CentOS-Base.repo

yum update -y

g++需要下载包，手动安装

gcc安装的版本是4.8,有点低

安装g++

Yum install gcc-c++

1. 安装cmake

yum install cmake

cmake -version

至此，基础的环境已经安装好了。。。。

如果要安装数据库之类的，之后再说

## 动态库无法加载

/etc/ld.so.conf

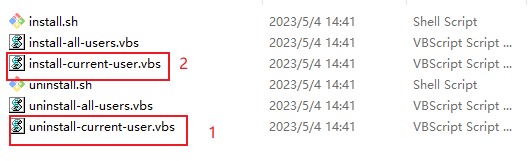
## CLion2022.1.3开发环境的搭建

### CLion2022.1.3安装激活教程

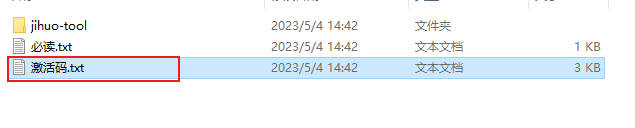
安装: Toolbox

在Toolbox中会检测到安装CLion2022.1.3

激活教程:



然后输入激活码:可以激活到2025年



**汉化过程:**



复制到CLion安装目录的lib文件夹下

### 在windows上运行

CLion2019及以前版本在配置中可能没有工具链选项，不方便使用

安装cmake

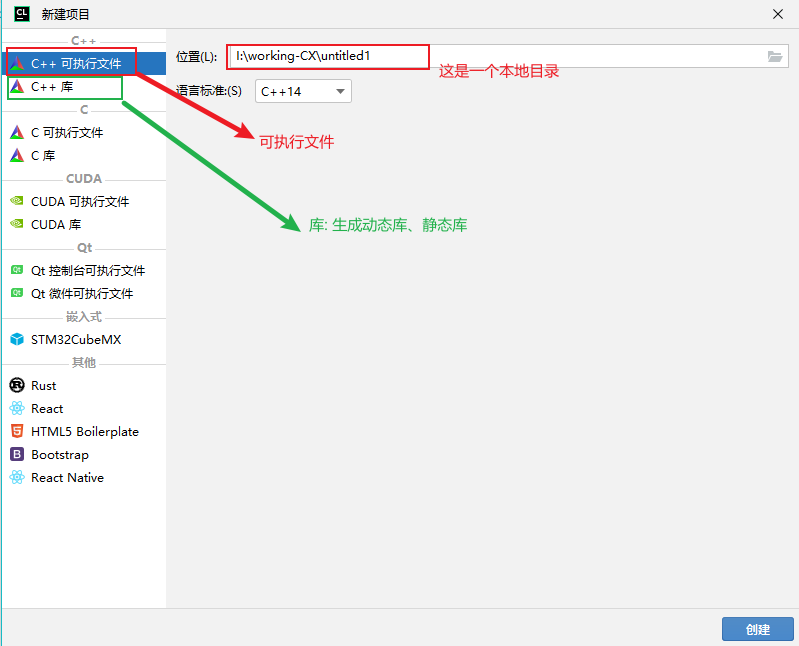
安装mingw64

解压之后将bin目录配置到环境变量 path中,CLion能够自动识别

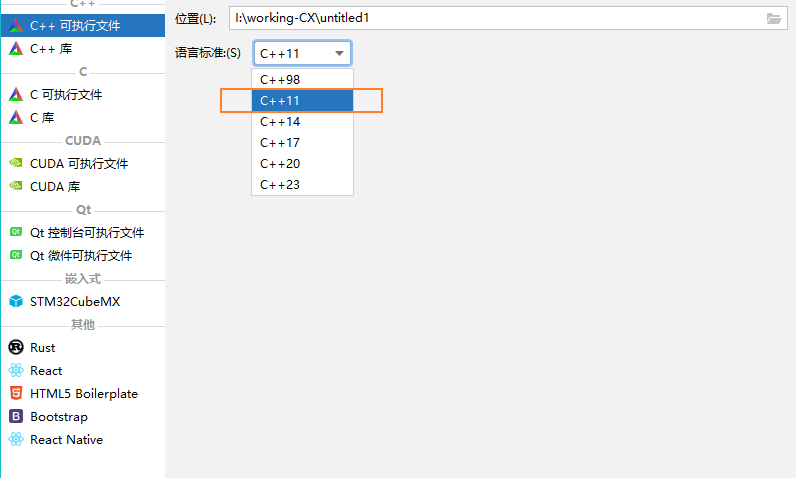
如果不安装mingw64,项目将不会自动配置cmake,很不方便,无法进行调试和运行

示例:

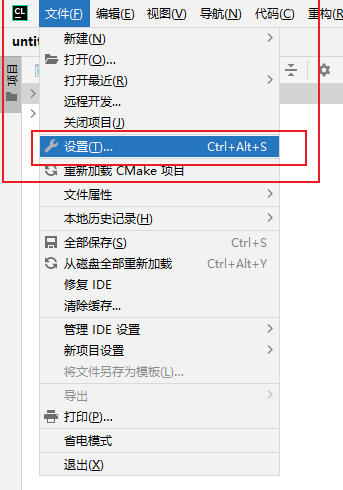
1. 创建项目



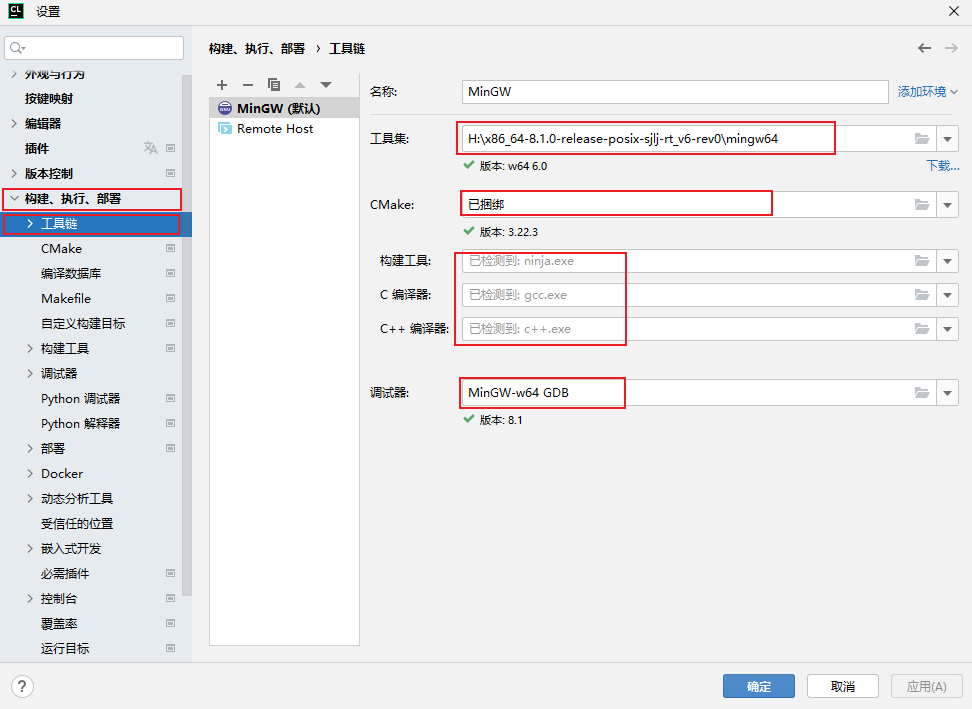
选择C++的版本(如果低版本的g++可能不支持C++14及以上)



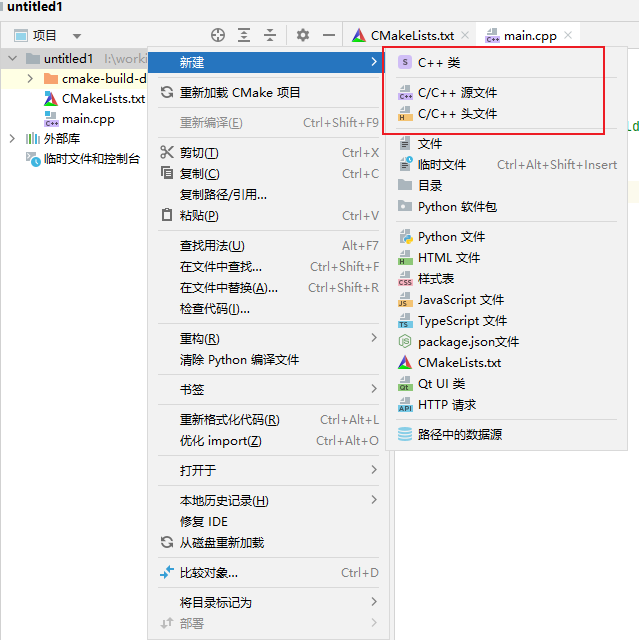
打开设置:



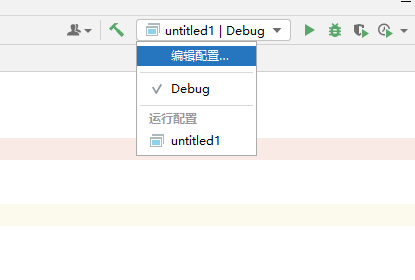
如果配置了系统的环境变量会自动识别,不需要自己配置了



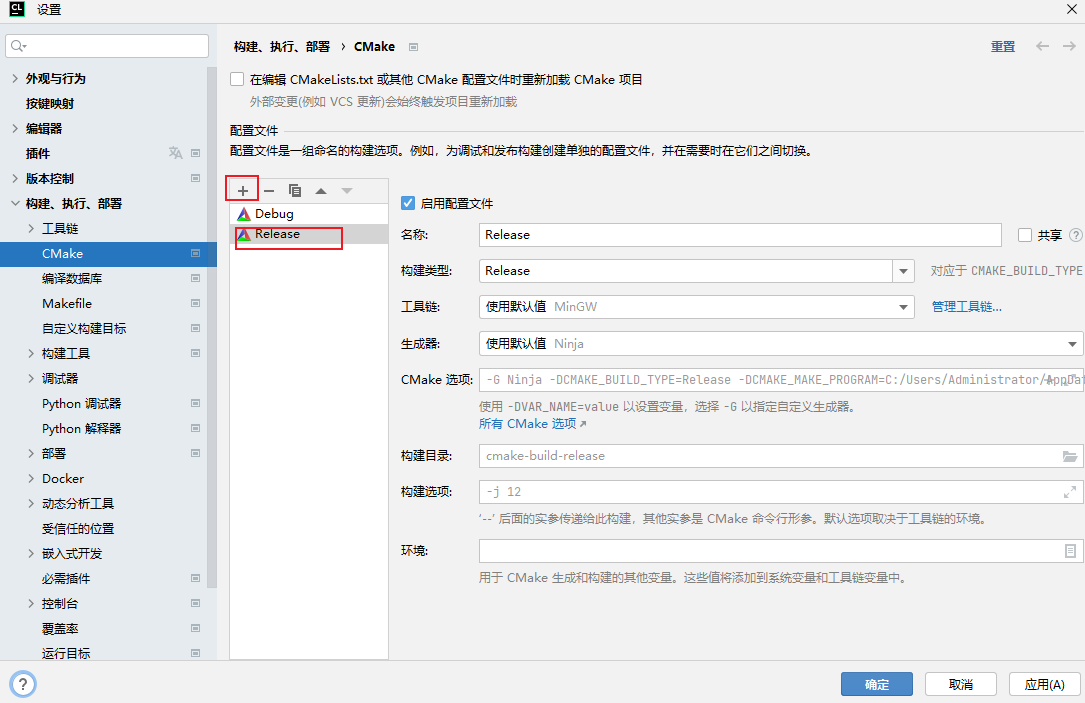
添加和删除cmake可以自动更改CMakeList.txt文件



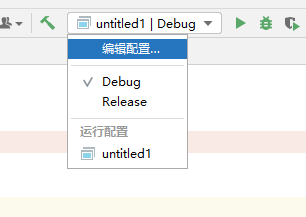
默认情况下,只有Debug版本,需要Release版本,可以在工具链中进行设置



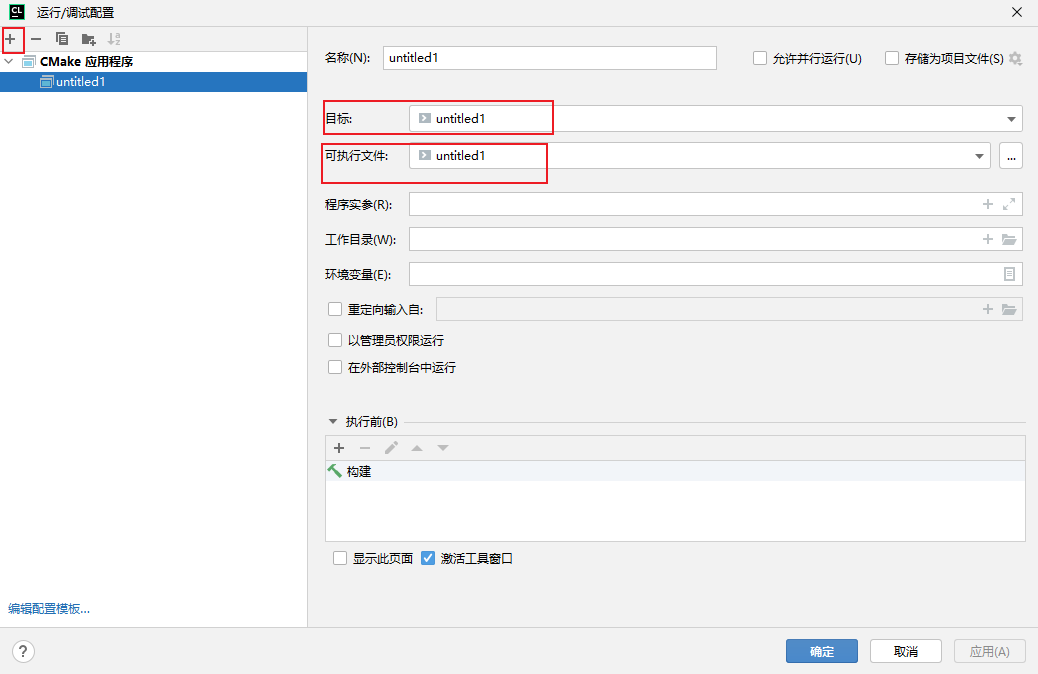
增加Release版本:

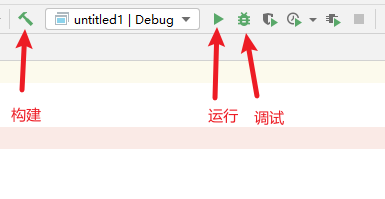


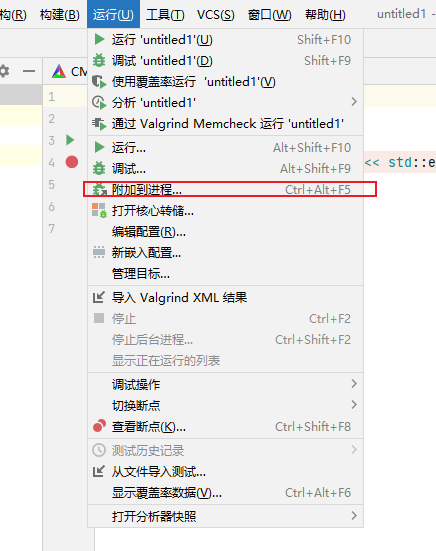
这右上角就会出现两种模式可以选择



可以增加构建目标







### 远程连接centOS上运行(不建议在虚拟机中之间安装CLion,使用起来会很不流畅和很不方便)

1. 在Linux下准备好C++开发环境

上面过程中在 yum 源中安装的g++ 和 cmake 版本太低,重新安装

* 删除cmake2 --> 3.22以上 3.13也可以使用(需要手动做更改)

yum remove cmake

* 首先安装必要的环境

yum install -y gcc g++ gcc-c++ make automake texinfo wget

* 下载Cmake 3.13(其它版本也可以,最后cmake3以上，否则cLion可能不支持cmake3版本一下的)

wget https://cmake.org/files/v3.13/cmake-3.13.4.tar.gz

下载完成后解压，并构建

tar -xf cmake-3.13.4.tar.gz

cd cmake-3.13.4

./configure

make

sudo make install

构建软连接: 在任何目录下都可以运行该命令(为了让CLion能够自动识别)

ln -s /usr/local/bin/cmake /usr/bin/cmake

* **termcap 安装**

不安装termcap将无法安装gdb（*****在安装GDB 前请先安装****termcap****库，否则在安装gdb时会报该库找不到的错误*****）

下载termcap 1.3.1 , 你可以在[这里](https://ftp.gnu.org/gnu/termcap/)找到你需要版本的termcap 下载，这里采用了1.3.1的版本

wget https://ftp.gnu.org/gnu/termcap/termcap-1.3.1.tar.gz

下载完成后解压，并构建

tar -xf termcap-1.3.1.tar.gz

cd termcap-1.3.1

./configure

make

sudo make install

### **gdb 安装(请使用gdb7.8以上版本，CLion才支持)**

卸载预装的gdb

yum remove -y gdb

下载gdb 7.9 , 你可以在[这里](http://mirrors.ustc.edu.cn/gnu/gdb/)找到你需要版本的gdb 下载，这里采用了7.9的版本

wget http://mirrors.ustc.edu.cn/gnu/gdb/gdb-7.9.tar.xz

tar -xf gdb-7.9.tar.xz

cd gdb-7.9

./configure

make

sudo make install

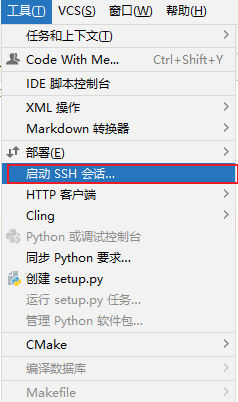
为了让CLion能够自动识别，构建软连接

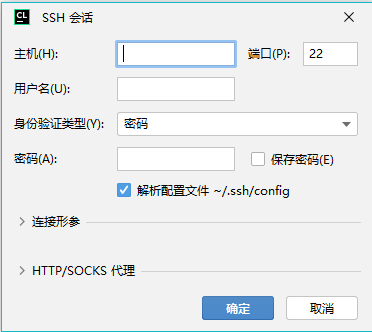
ln -s /usr/local/bin/gdb /usr/bin/gdb

**CentOS里的基础环境已经构建完成:**

**还需要通过CLion去配置远程连接Linux服务器**

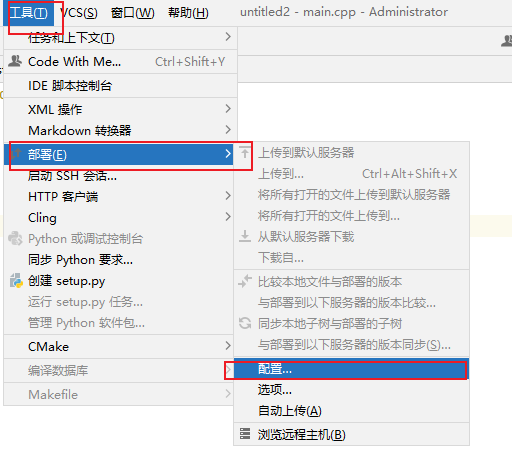
可以先配置下SSH会话,方便在CLion使用Linux的终端

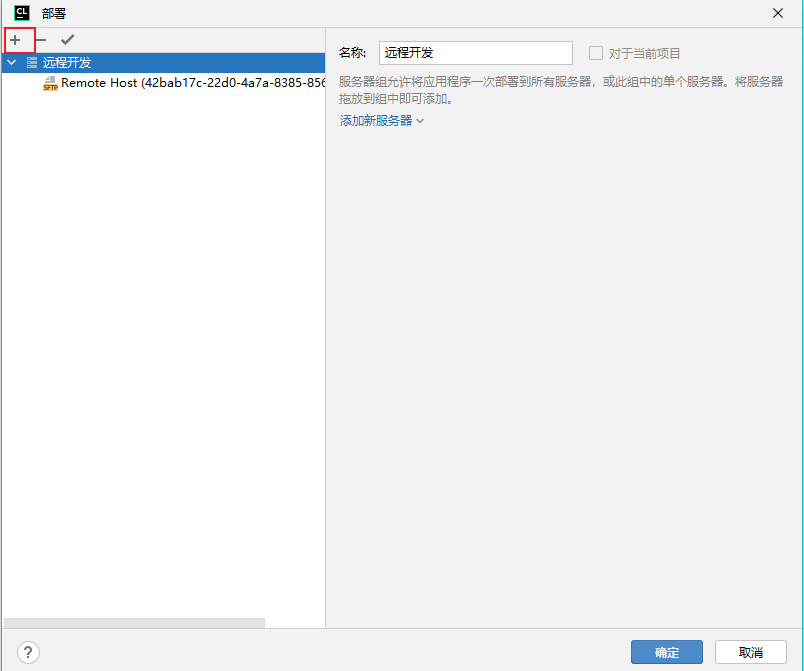




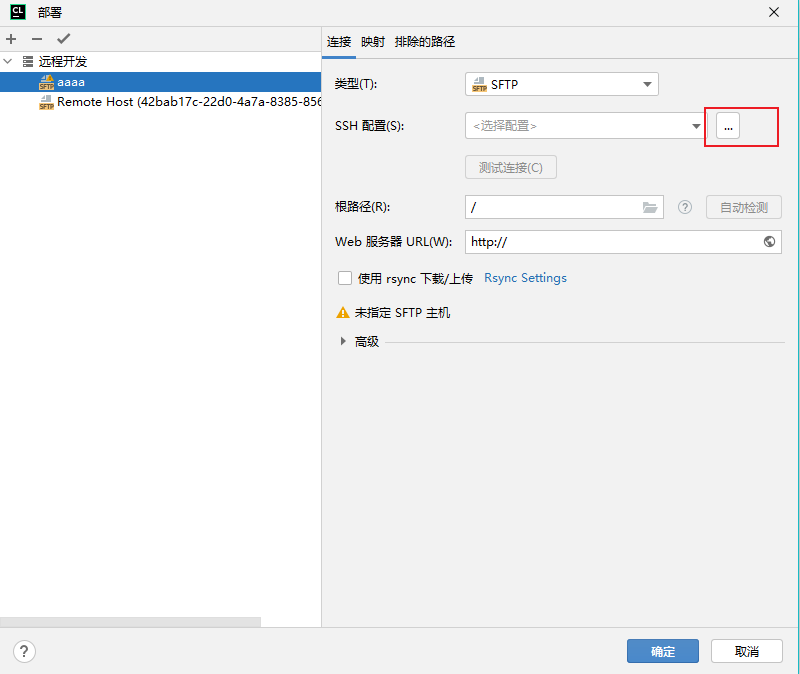
填完-->确定

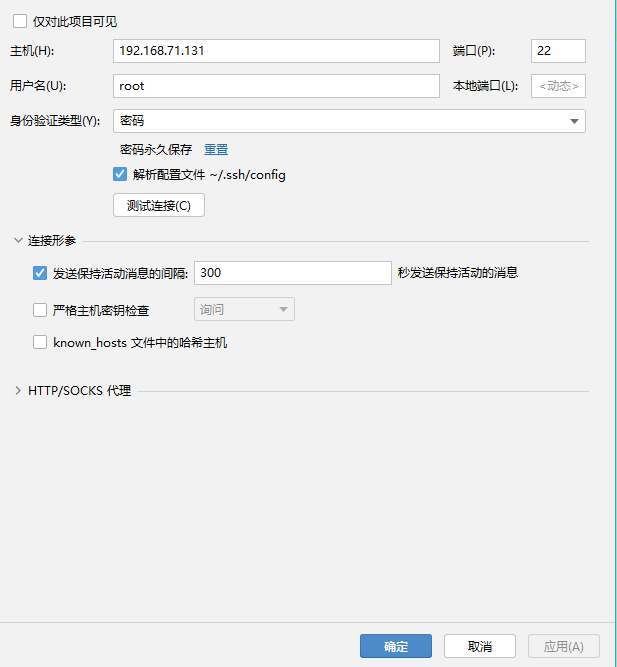
通过远程连接部署Linux服务器上开发



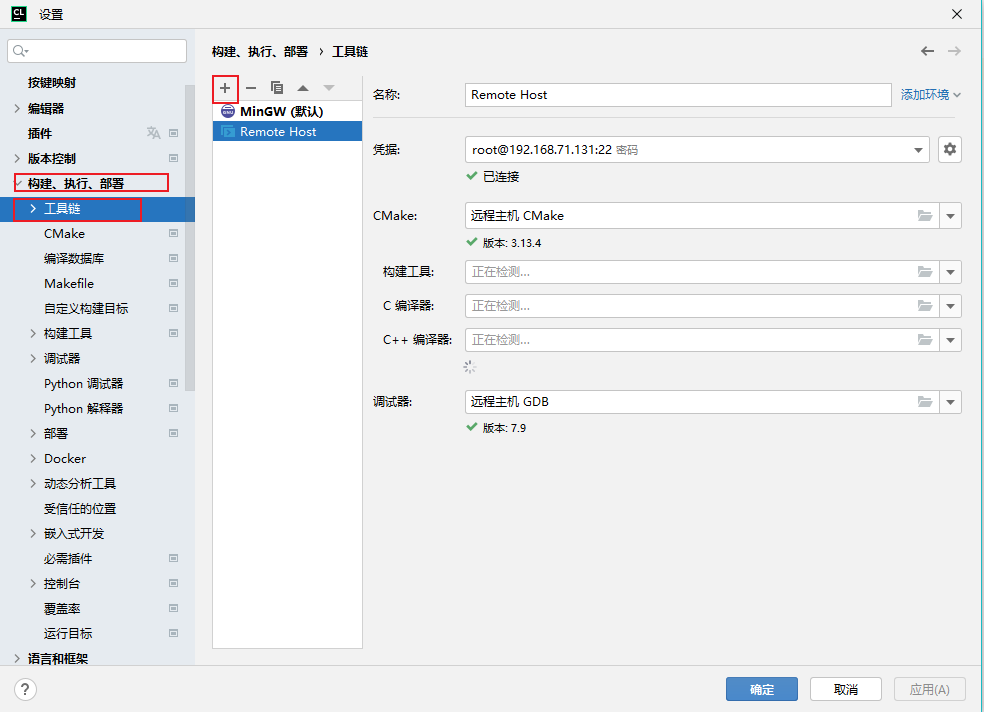








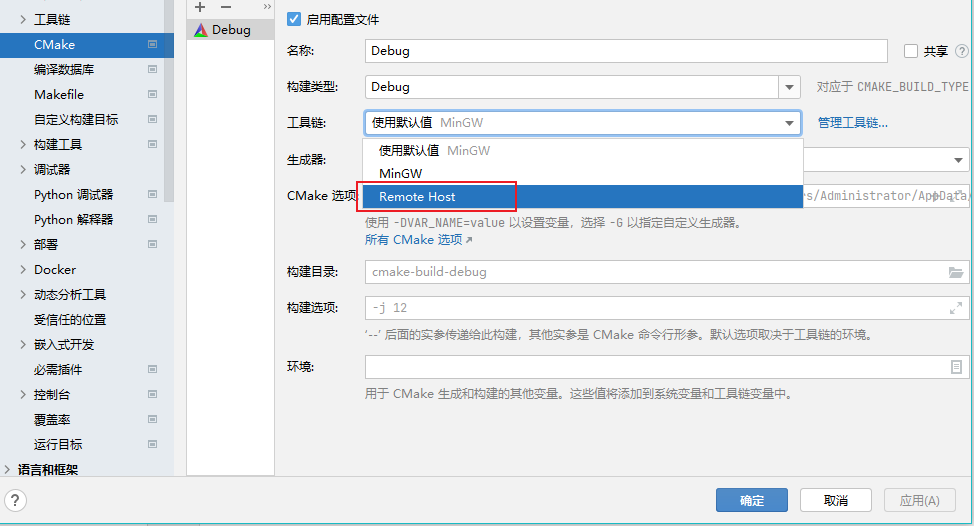
再打开文件-->设置-->编译/构建/部署

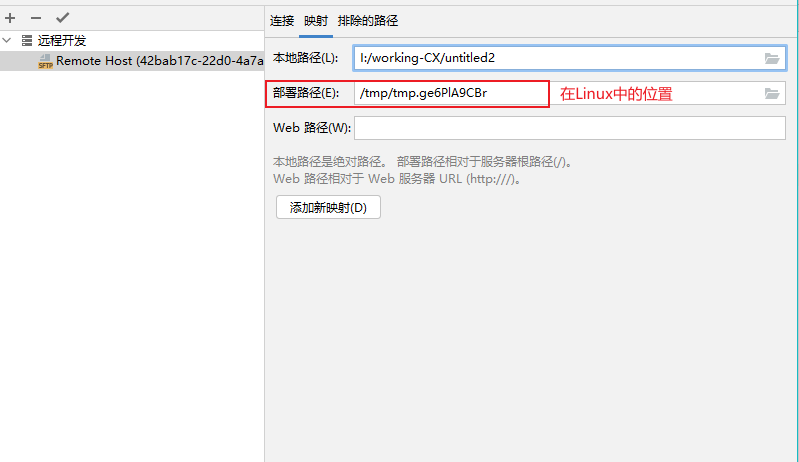






Cmake-->工具链-->选择远程主机的工具链,负责无法使用Linux的库(MinGW是本机的工具链)





部署完成

可以愉快的开发了

### 可以选择安装一些好用的插件:



Linux安装源码安装libevent

先安装:yum install -y [openssl](https://so.csdn.net/so/search?q=openssl&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/m0_71741835/article/details/_blank) openssl-devel --> 如果需要支持openSSL

/usr/lib/libevent

解压安装: tar -xzvf libevent-xxxxxx

./configure -prefix=XXXX 指定安装目录

Make

Make install

/home/chen/pack/libevent-2.1.12-stable/.libs

中文乱码:



Linux中安装telnet-server

使用方法:<https://www.cnblogs.com/nmap/p/6148306.html>

*yum* -y install telnet-server.x86\_64

*yum* -y install telnet.x86\_64

*yum* -y install xinetd.x86\_64

上面3个都安装成功后开始配置并启动Telnet，xinetd和telnet必须设置开机启动，否则无法启动Telnet服务

*system*ctl enable xinetd.service

*system*ctl enable telnet.socket

接下来启动服务

*system*ctl start telnet.socket

*system*ctl start xinetd

可以 正常使用telnet去测试ip端口了

telnet 127.0.0.1 8080

## Linux安装mysql5.7(很容易出错，建议先快照)

检查是否安装过mysql

rpm -qa|grep mysql

删除以前安装过的mysql

rpm -e --nodeps mysql-xxxx

****如果有 mysql 的话，把 mysql 的文件全部删除，和卸载 mysql****

find / -name mysql

rm XXXXX

下载安装包

wget <https://dev.mysql.com/get/Downloads/MySQL-5.7/mysql-5.7.24-linux-glibc2.12-x86_64.tar.gz>

很慢的话就使用本机下载好安装包，然后拷贝到虚拟机中，解压安装

<https://downloads.mysql.com/archives/community/>

选择对应的版本

## **卸载 Centos7 自带的 mariadb**

一般用的 centos 的系统都会自带一个系统数据库，那就是 mariadb，因为没有激活，只有一个安装好的 mariadb-libs-5.5.60-1.el7\_5.x86\_64，可以把 mariadb-libs-5.5.60-1.el7\_5.x86\_64 卸载。

### **查找系统自带的 mariadb**

rpm -qa | grep mariadb



没有就不用删除

### **卸载系统自带的 mariadb**

#卸载mariadbrpm -e --nodeps mariadb-libs-5.5.60-1.el7-5.x86\_64#卸载完查看还有没有mariadbrpm -qa | grep mariadb

## **安装 mysql5.7**

tar xf mysql-5.7.38-linux-glibc2.12-x86\_64.tar

#解压mysql-5.7.38-linux-glibc2.12-x86\_64.tar.gz

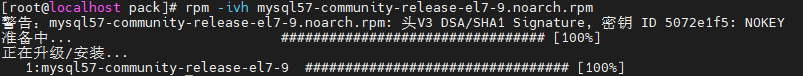
tar xf mysql-5.7.38-linux-glibc2.12-x86\_64.tar.gz

方式2

通过rpm文件安装

wget https://dev.mysql.com/get/mysql57-community-release-el7-9.noarch.rpm

rpm -ivh mysql57-community-release-el7-9.noarch.rpm



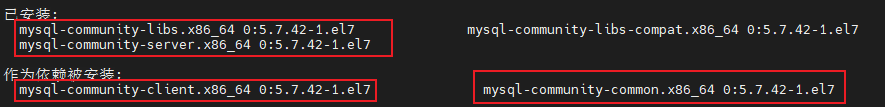
执行完成后会在/etc/yum.repos.d/目录下生成两个repo文件mysql-community.repo mysql-community-source.repo



安装

yum install mysql-server --nogpgcheck

检查下server和client是否被安装

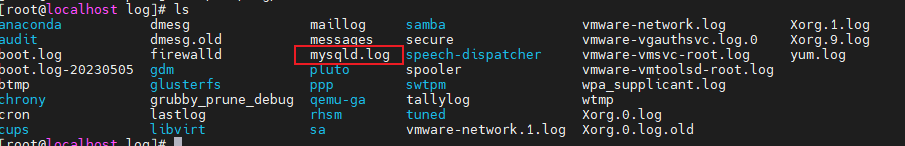


查看安装版本：mysqladmin --version



获取安装时的临时密码（在第一次登录时就是用这个密码）:（通过日志查看）

cd /var/log/



日志目前是空的

### **MySQL服务初始化**

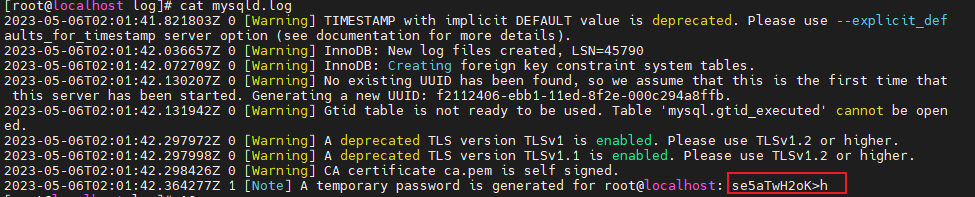
执行下面的命令初始化：

mysqld --initialize --user=mysql

查看日志

cd /var/log/

cat mysqld.log

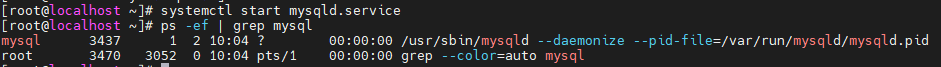


可以看到初始密码

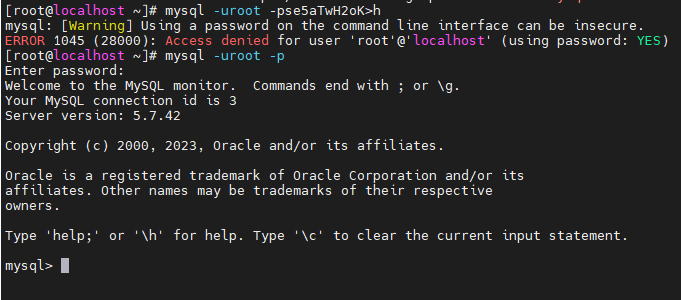
### **修改初始化密码**

要修改MySQL，首先必须启动MySQL服务，使用初始化密码登录后，才能修改密码：

启动MySQL服务：systemctl start mysqld.service



关闭MySQL服务：systemctl stop mysqld.service



注意:输入密码时需要手动输入,好像复制粘贴没有用

修改密码：ALTER USER 'root'@'localhost' IDENTIFIED BY 'xxx';

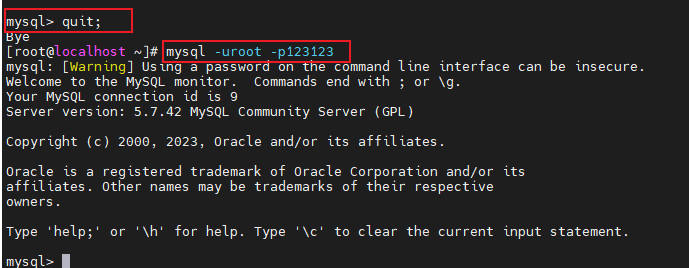
XXX：需要设置的密码 如:123123 ==> root用户设置密码



本质:修改数据库中用户表中的root用户的密码

验证密码是否修改成功:

Quit ==> 退出



设置mysql服务开机自动启动:

查看mysql是否自启动（****默认自启动****）

systemctl list-unit-files|grep mysqld.service



如不是enabled可以运行如下命令设置自启动

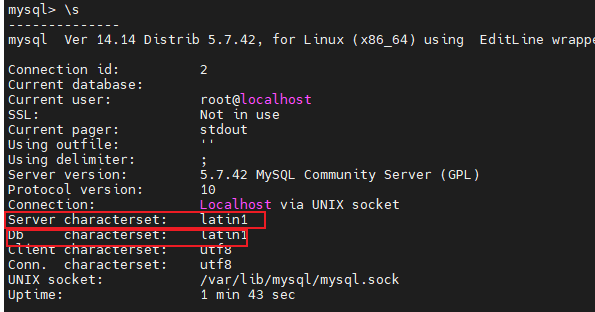
systemctl enable mysqld.service

重启电脑



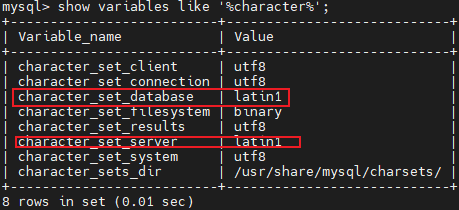
设置字符集(防止中文乱码)

查看字符集:登录进去,然后 \s



或者

show variables like '%character%';

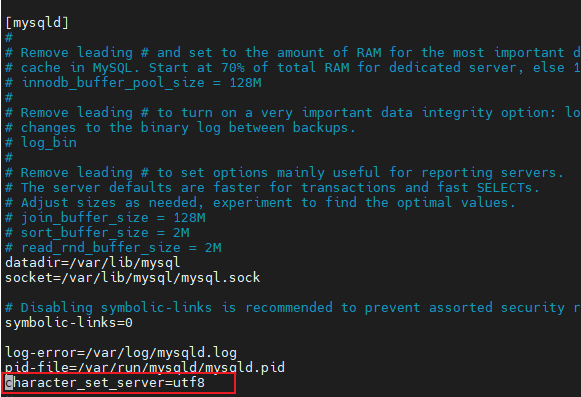


修改服务端的字符集:

需要修改配置文件:

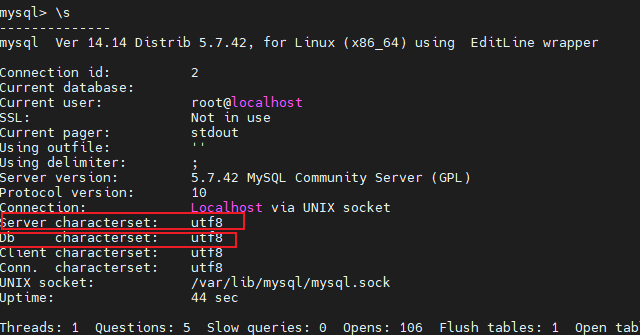
修改编码，需要修改mysql配置文件：vi /etc/my.cnf

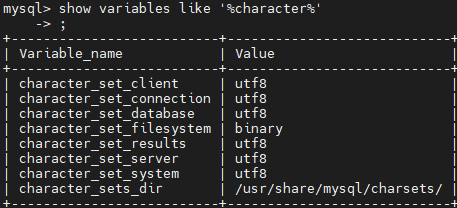
在mysqld下添加一行编码配置：character\_set\_server=utf8



保存并退出，重启mysql服务：systemctl restart mysqld

查看字符集是否改变





===> 我们没有创建数据库,不用修改

建立修改下表和数据库的编码(不修改,在后面可能出现一些问题,查询时找不到字段)

修改数据库的字符集 XXX创建的数据库

alter database XXX character set 'utf8';

修改数据表的字符集：XXX表

alter table XXX convert to character set 'utf8';

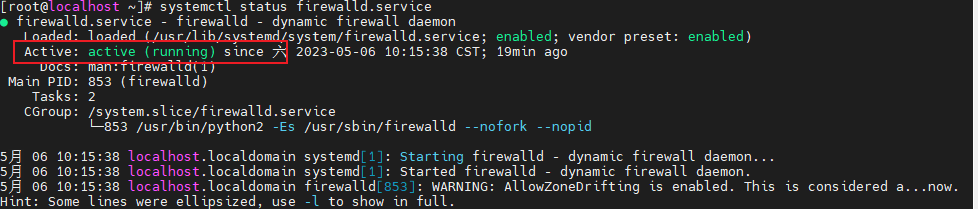
### 设置远程访问mysql（默认情况下，mysql不允许远程连接。只允许localhost连接）

方便使用本机之间访问centOS中的mysql

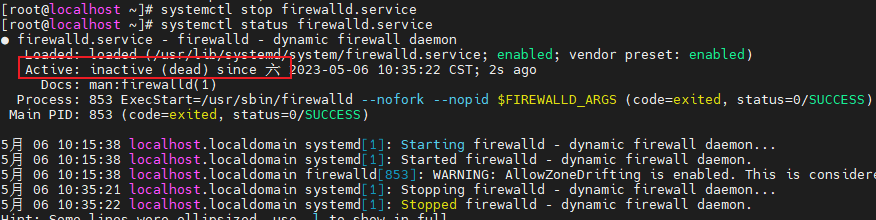
1. 关闭centOS的防火墙 ===> 不关闭,远程连接不了（连接被拒绝）

查看防火墙是否运行:

systemctl status firewalld.service



systemctl stop firewalld.service



这样关闭，重启之后防火墙会自动打开

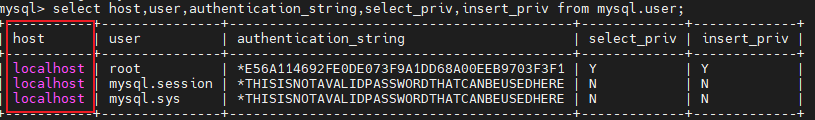
永久关闭防火墙:

systemctl disable firewalld.service



给root用户远程登录权限

查看root用户登录的host --> 默认只允许本机登录



#授予通过网络方式登录的的root用户，对所有库所有表的全部权限，密码设为123123.

grant all privileges on \*.\* to root@'%' identified by '123123';

\*.\* ==> XXX数据库.XXX表

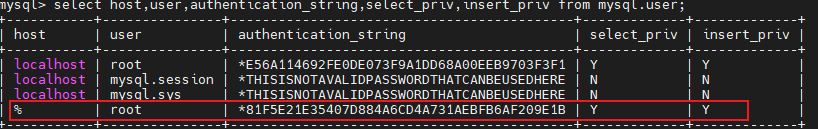


刷新，生效

flush privileges;

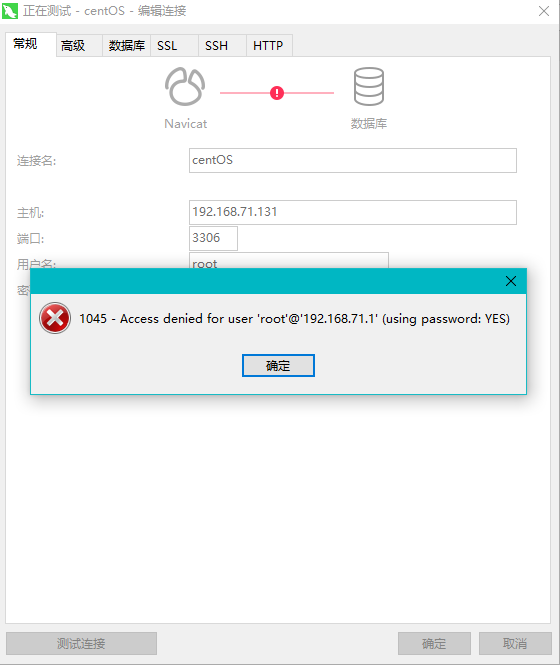


检验是否成功:

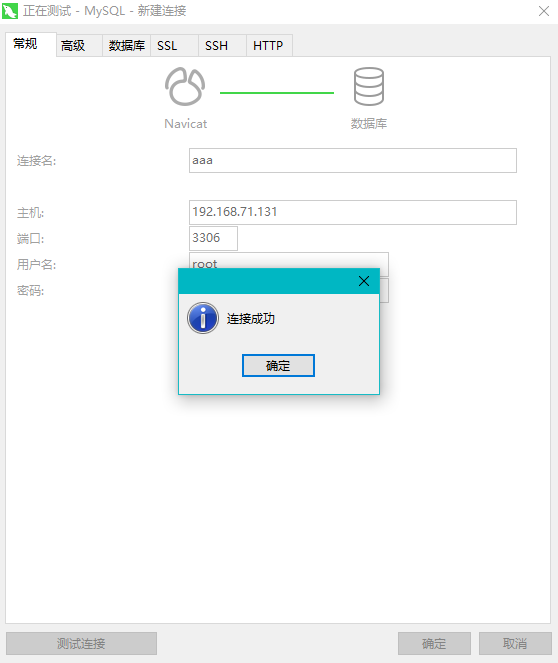


如果存在其它用户，操作相同

使用netvicat测试



重写设置密码，然后刷新(可能设置远程登录的密码设置错误)

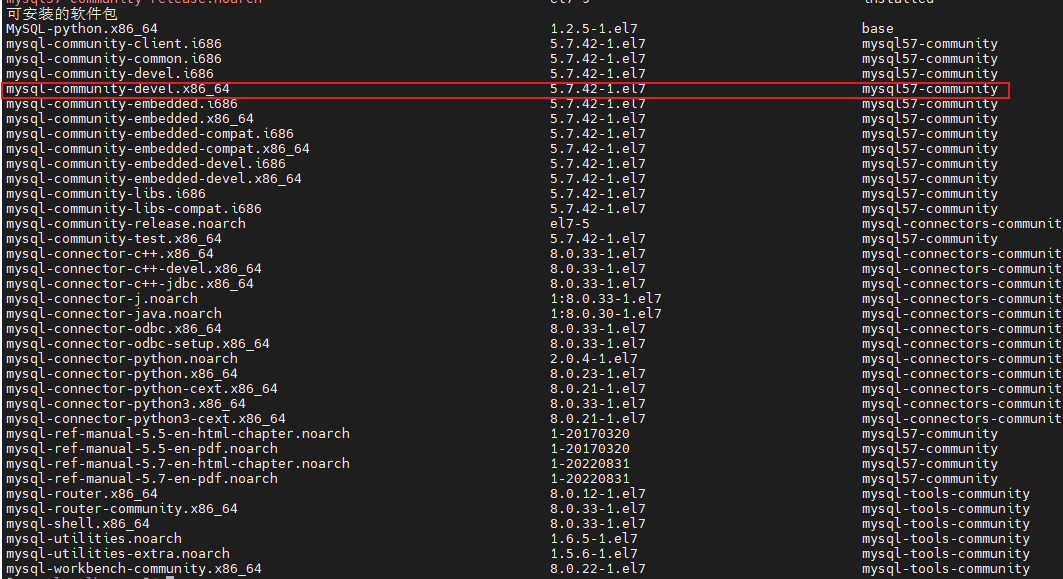


数据库就设置完成了，但是还不能之间被C++程序调用,需要安装开发包

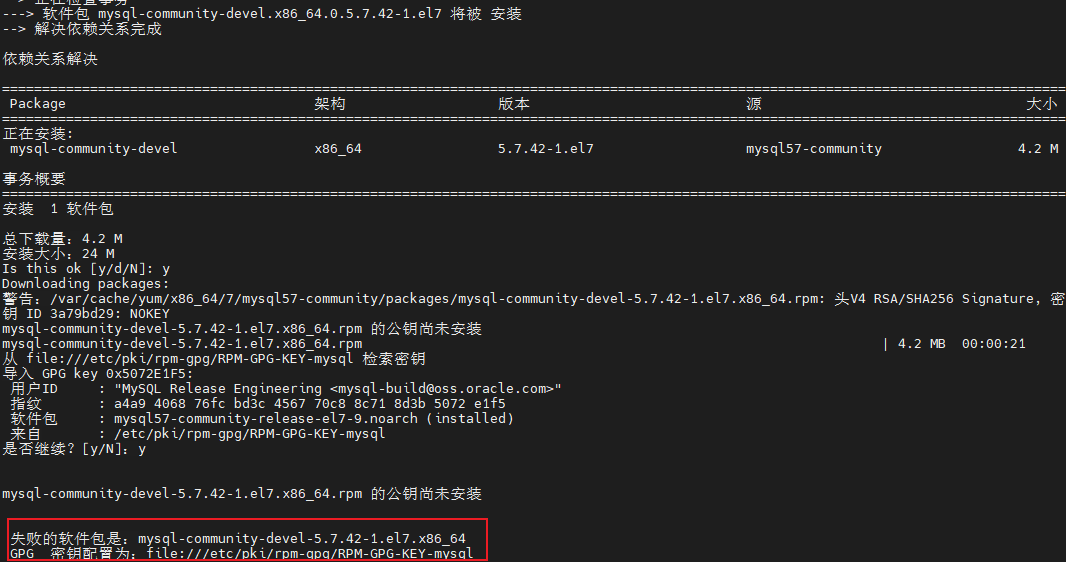
mysql-devel 开发用到的库以及包含文件

查看源里面有没有开发包:(选择对应版本)

yum list mysql\*



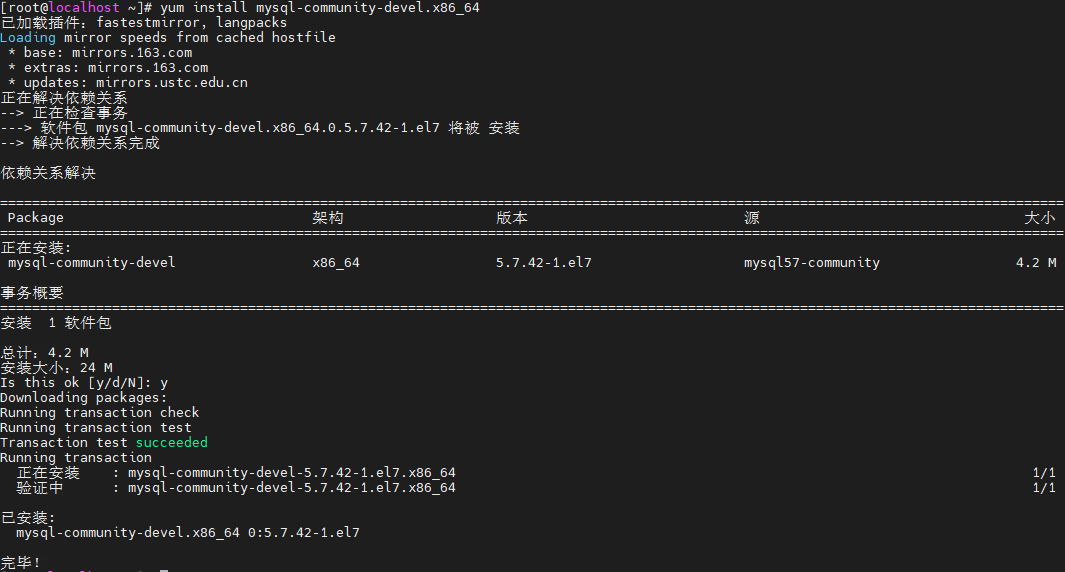
yum install mysql-community-devel.x86\_64



安装密钥

rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2022

yum install mysql-community-devel.x86\_64



安装的目录:

默认情况下会安装在: /usr/XXX ==> 需要找一下(include/mysql lib/mysql lib64/mysql)

使用:

<https://blog.csdn.net/weixin_51609435/article/details/127642258>

编译:

g++ demo.cpp -o demo -L/usr/lib64/mysql -lmysqlclient -I./include

或者把库拷到项目的目录中使用

