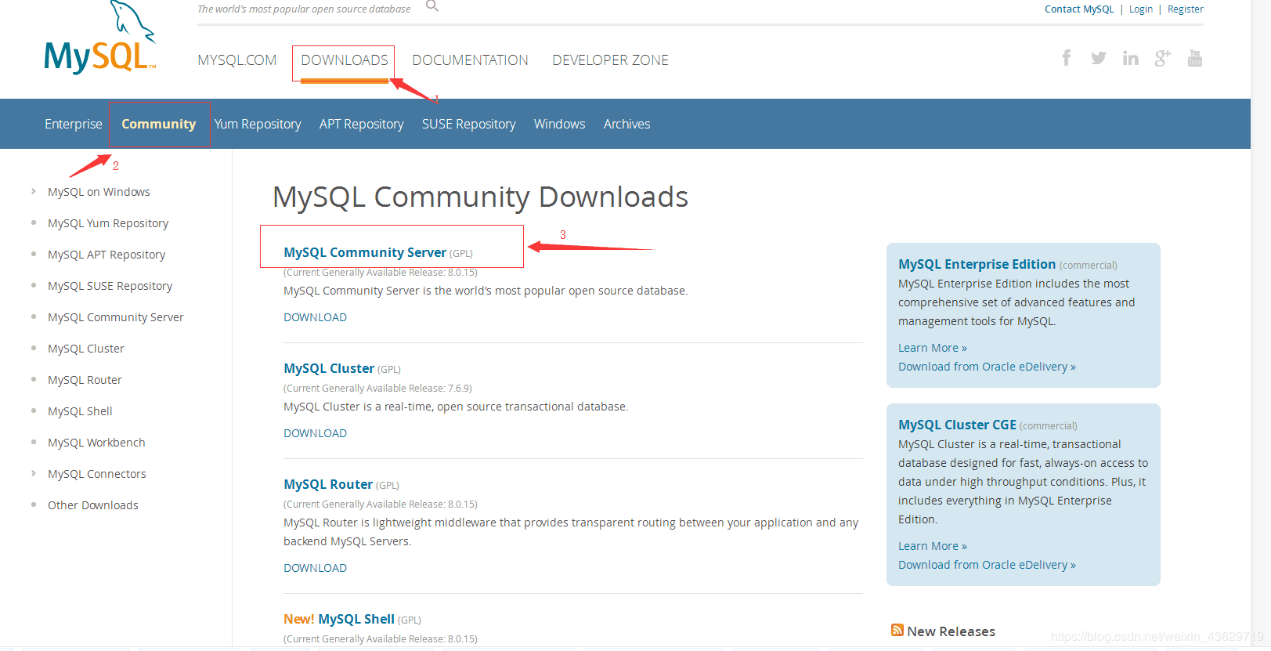
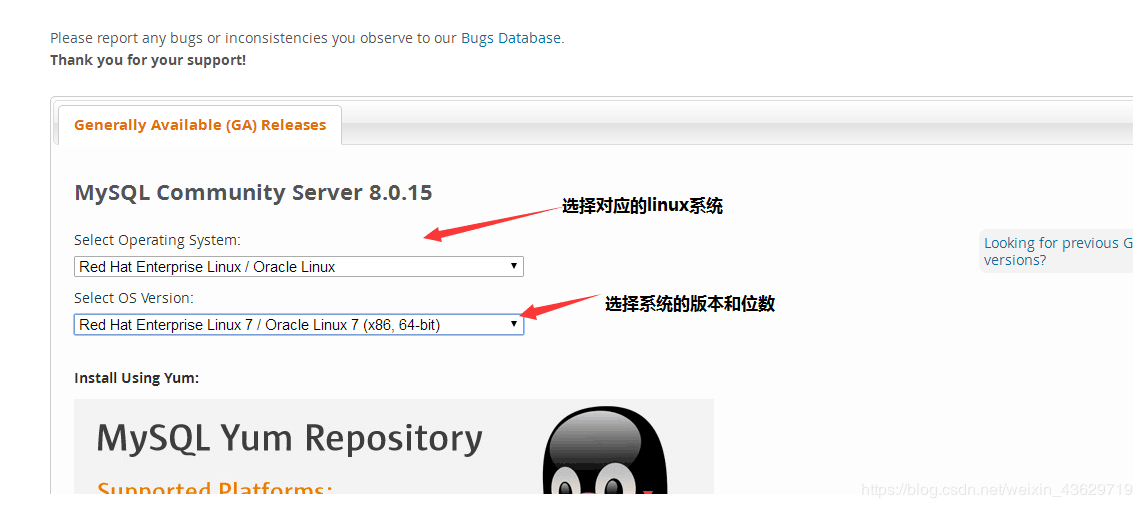
<http://www.360doc.com/content/19/0316/10/51449331_821858850.shtml>

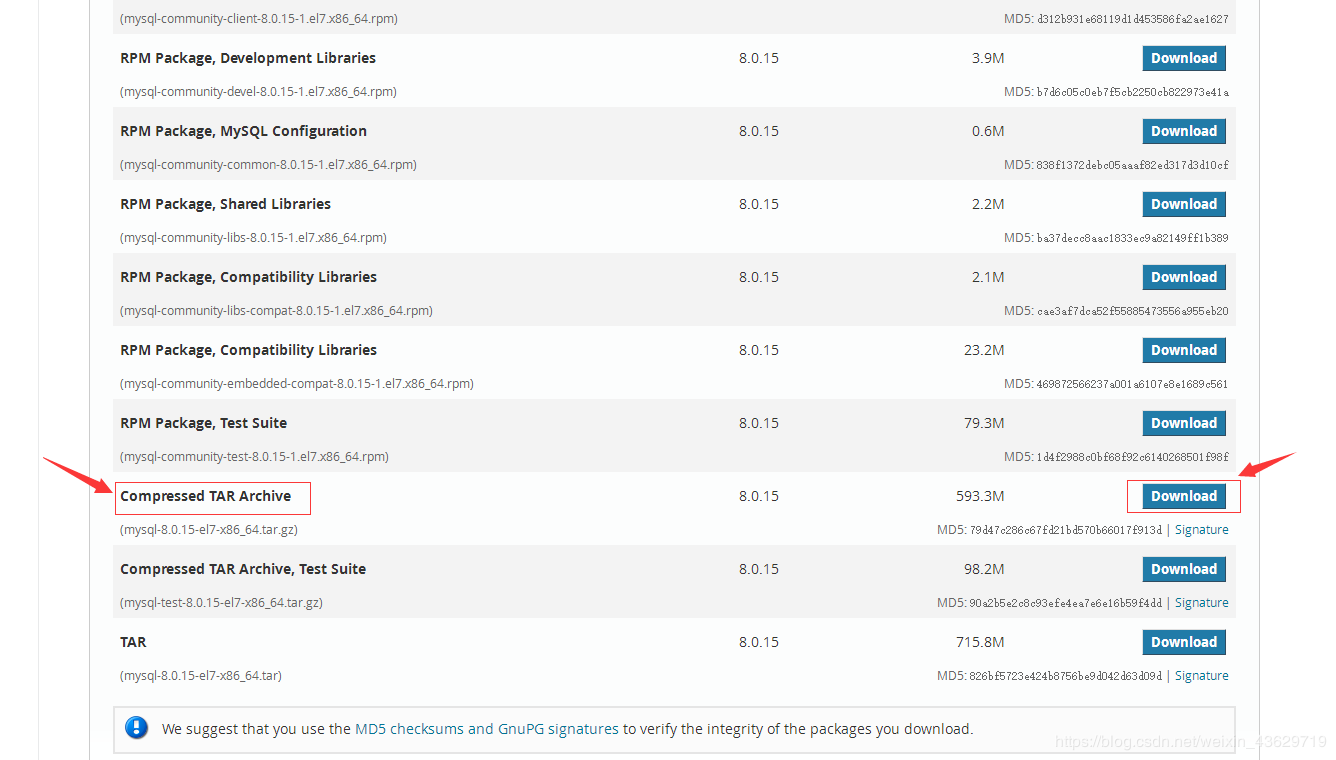
**Centos7下安装mysql8.0.15完整详细教程**

## 1.进官网，选择适合自己版本的linux下的MYSQL版本。



## ****2.按照箭头上的数字，依次进行操作，进入下面这个图景。****





等待下载好，这样第一步就完成了。  
2.将其下载好的mysql-8.0.15-el7-x86\_64.tar.gz 上传至linux系统中进行解压 。  
(1)本人将这个文件上传至/root/ 目录下，首先查看 /root/目录下的文件。以确保系统中有这个文件。

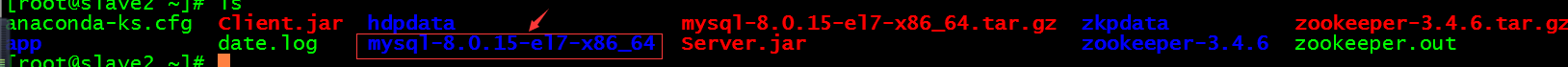




(2)将文件进行解压。

tar -zxvf mysql-8.0.15-el7-x86\_64.tar.gz

然后会在当前 /root/ 目录下会生成 这个目录。



3.对mysql进行安装。  
(1)将mysql-8.0.15-el7-x86\_64 安装到 /usr/local/mysql 下。

mv mysql-8.0.15-el7-x86\_64 /usr/local/mysql //将文件移动到 /usr/local/ 目录下，并更名为mysql 。

(2)为系统添加mysql 组和用户。

groupadd mysql和useradd -r -g mysql mysql

(3)进入 /usr/local/mysql 目录下，修改相关权限。

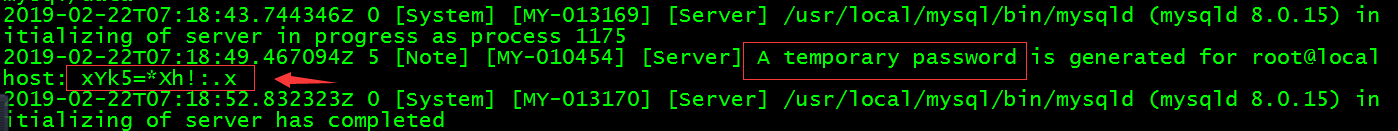
cd /usr/local/mysql //进入/usr/local/mysql 目录

chown -R mysql:mysql ./ //修改当前目录为mysql用户

(4)mysql初始化操作,记录下临时密码,之后第一次登录的时候会用到。

bin/mysqld --initialize --user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data //运行完成会生成一个暂时的密码。

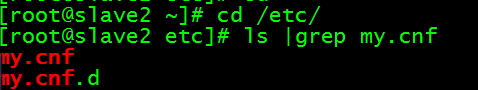
结果如下



(5)查看初始化过程中有没有自动生成my.cnf 文件.

cd /etc/ //进入/etc/目录下

ls | grep my.cnf //查看是否有my.cnf 文件。有则会产生结果。

(6) 初始化生成了my.cnf 文件,跳至第八步，修改my.cnf 文件的信息。  


(7) 如果初始化过程中没有生成my.cnf , 新建一个my-defalut.cnf文件,将其复制到/etc/my.cnf

touch my-defalut.cnf //新建一个文件

chmod 755 my-defalut.cnf // 赋予权限

cp my-defalut.cnf /etc/my.cnf // 将文件复制到/etc/ 目录下，并更名为my.cnf 文件名。

(8)my.cnf 的信息如下

**[mysqld]**

**basedir=/usr/local/mysql**

**datadir=/usr/local/mysql/data**

**port=3306**

**socket=/tmp/mysql.sock**

**pid-file=/usr/local/mysql/$hostname.pid //注意了,小细节，这里的 $hostname 是linux的主机名。一般每个人主机名都是不一样的。**

**sql\_mode=NO\_ENGINE\_SUBSTITUTION,STRICT\_TRANS\_TABLES**

**(9)为mysql配置环境变量。**

**vim /etc/profile // 打开profile文件。**

**export MYSQL\_HOME**

**MYSQL\_HOME=/usr/local/mysql**

**export PATH=$PATH:$MYSQL\_HOME/lib:$MYSQL\_HOME/bin**

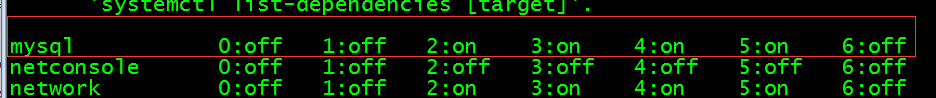
(10) 设置为开机自启动项。

依次执行一下代码。

**cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysql**

**chmod +x /etc/init.d/mysql //添加可执行权限。**

**chkconfig --add mysql // 注册启动服务**

输入chkconfig --list //查看是否添加成功。  


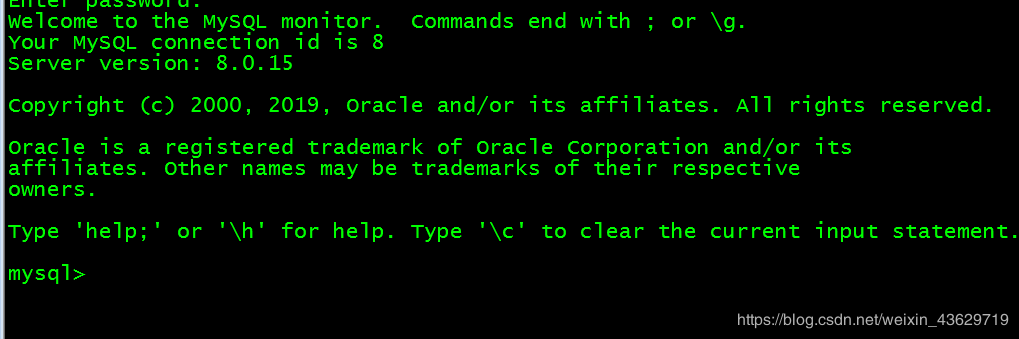
表示成功。  
(11)开启服务器，登录进入mysql。

service mysql start //开启服务器。

mysql -uroot -p //登录进入mysql，然后提示输入密码。

å¨è¿éæå¥å¾çæè¿°

输入初始化过程中生成的临时密码，然后回车就行。进入一下页面。

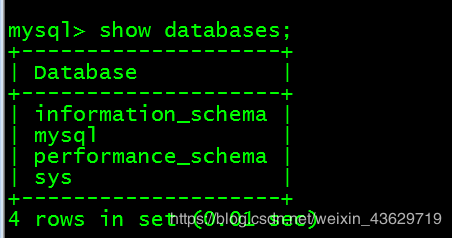


到此，说明已经安装成功了。  
(12)进入mysql后，修改密码。不然你什么也做不了。

alter user 'root'@'localhost' identified by 'your\_password';

然后 exit; 退出，重新进入，进行测试。

show databases;



到此，关于安装mysql 8.0.15版本的全部过程就已经结束了!

my.conf配置

**[client]**

**port=3306**

**socket=/usr/local/mysql/mysql.sock**

**[mysqld]**

**#是否启用bin log**

**skip-log-bin**

**#Mysql服务的唯一编号 每个mysql服务Id需唯一**

**#主从复制必须，并且各服务器具有唯一性**

**server-id = 1**

**#默认启动用户，一般不需要修改，可能出现启动不成功**

**user = mysql**

**#端口号**

**port = 3306**

**#套接字文件 （套接字方式登陆比TCP/IP方式连接快）**

**socket=/usr/local/mysql/mysql.sock**

**#数据库默认字符集,主流字符集支持一些特殊表情符号（特殊表情符占用4个字节）**

**#character-set-server = utf8mb4**

**#数据库字符集对应一些排序等规则，注意要和character-set-server对应**

**#collation-server = utf8mb4\_general\_ci**

**#数据库安装目录--指定此参数可解决相对路径造成的问题**

**basedir = /usr/local/mysql**

**#数据库目录，数据库目录切换时需要用到**

**datadir = /usr/local/mysql/data**

**#外部锁定(非多服务器可不设置该选项，默认skip-external-locking)**

**#external-locking = FALSE**

**#主要用于MyISAM存储引擎,如果多台服务器连接一个数据库则建议注释下面内容**

**#skip-external-locking**

**#跳过主机名解析，直接IP访问，可提升访问速度**

**skip-name-resolve**

**# 跳过密码登录**

**#skip-grant-tables**

**#默认为1,表示启用警告信息记录日志,不需要置0即可,大于1时表示将错误或者失败连接记录日志**

**#log-warnings**

**#连接数**

**#全局只读变量,文件描述符限制 注：上限其实为OS文件描述符上限，小于OS上限时生效 可用lsof查看限制并修改相应配置**

**#open\_files\_limit = 10240**

**#短时间内的多少个请求可以被存在堆栈中建议512倍数**

**back\_log = 600**

**#是否对sql语句大小写敏感，1表示不敏感 8.0 --initialize lllower-case-table-names=1**

**#MySQL默认的最大连接数为100，MySQL服务器允许的最大连接数16384 如果系统在短时间内有很多连接，则需要增大该参数的值**

**max\_connections = 3000**

**#设置每个主机的连接请求异常中断的最大次数，当超过该次数，MYSQL服务器将禁止host的连接请求**

**max\_connect\_errors = 6000**

**#thread\_concurrency应设为总CPU核数的2倍**

**#thread\_concurrency = 2**

**#线程缓存变量，减少连接的创建，重新利用保存在缓存中线程的数量**

**#thread\_cache\_size = 300**

**#每个线程的堆栈大小，默认值足够大，可满足普通操作。可设置范围为128K至4GB，默认为192KB**

**#thread\_stack = 192K**

**#缓存**

**#查询缓存，设置不宜过大，主要是因为缓存越大，设置失效时间越长。如果高并发写，可以禁用该功能**

**#query\_cache\_size = 64M**

**#指定单个查询能够使用的缓冲区大小，默认为1Mquery\_cache\_limit = 2M**

**#大数据查询4k,小数据查询2k,目的是提高缓存命中率 值大对大数据查询有好处，你的查询都是小数据查询，就容易造成内存碎片和浪费 query\_cache\_min\_res\_unit = 4k**

**#限制接收数据包的大小,单条数据超过该值时插入或更新失败,如果有BLOB对象建议修改成1G**

**max\_allowed\_packet = 800M**

**#每个连接独享，用于优化不能通过sql或者索引优化的group和order等，设置的值应适中 太大将导致在连接数增高时，内存不足**

**sort\_buffer\_size = 2M**

**#用于表间关联缓存的大小,每个连接独享 500个连接，500\*2 = 1G,将消耗4G内存**

**join\_buffer\_size = 2M**

**#查询生成的临时表大小超过该值时会在硬盘生成MyISAM表，如果存在大量group by 等语句，可调整大小 默认32M**

**tmp\_table\_size = 128M**

**#进行排序查询时，MySql会首先扫描一遍该缓冲，以避免磁盘搜索**

**read\_buffer\_size=1M**

**read\_rnd\_buffer\_size=2M**

**innodb\_file\_per\_table=1**

**default-storage-engine=INNODB**

**[mysqld\_safe]**

**#mysql进程文件，可指定自己的进程文件**

**pid-file = /usr/local/mysql/mysql.pid**

**#数据库错误日志文件**

**#log-error = var/local/mysql/data/log/mysqld.log**

<https://blog.csdn.net/qq_34769161/article/details/81811102>