

centos 卸载 docker

首先查看Docker版本  
# yum list installed | grep docker  
docker-ce.x86\_64  18.05.0.ce-3.el7.centos @docker-ce-edge  
   
执行卸载  
# yum -y remove docker-ce.x86\_64  
   
删除存储目录  
   
# rm -rf /etc/docker  
# rm -rf /run/docker  
# rm -rf /var/lib/dockershim  
# rm -rf /var/lib/docker

如果发现删除不掉，需要先 umount，如  
# umount /var/lib/docker/devicemapper

redis的安装与卸载

**安装Redis**

**一、下载Redis**

[redis下载地址](https://links.jianshu.com/go?to=https%3A%2F%2Fredis.io%2Fdownload)  
也可以直接下载到服务器：wget http://download.redis.io/releases/redis-4.0.11.tar.gz

**二、将Redis的tar包上传到服务器**

上传命令：rz 选择你的tar包文件

#可以考虑将Redis放到/usr/local/文件夹下面

cd /usr/local/

#在执行上传命令

rz 选择你的tar包文件

**三、解压、编译**

#解压命令

tar -zxvf redis-4.0.11.tar.gz

#切换目录

cd redis-4.0.11

#编译

make

#再次切换目录

cd src

#再次编译

make install

这是就已经安装好了，但是我们没有指定安装目录，所以启动文件都在/usr/local/bin/文件夹下面。我们可以在cd /usr/local/redis-4.0.11创建一个文件夹(创建文件夹命令:mkdir bin)，存放Redis的可执行文件，将Redis相关的文件全部移到新建的bin文件夹下面cp -r /usr/local/bin/redis-\* ./这样就整理完成了。可以正常启动了

#启动命令：同时为Redis启动指定配置文件

cd /usr/local/redis-4.0.11/bin

./redis-server /usr/local/redis/etc/redis.conf

#查看是否启动成功了，查看进程

ps -ef |grep redis

**解释一下Redis比较重要的三个可执行文件：**

**redis-server**：Redis服务器程序  
**redis-cli**：Redis客户端程序，它是一个命令行操作工具。也可以使用telnet根据其纯文本协议操作。  
**redis-benchmark**：Redis性能测试工具，测试Redis在你的系统及配置下的读写性能

**四、Redis配置**

列举Redis常用的配置项：  
**daemonize** 如果需要将Redis服务以守护进程在后台运行，则把该项的值改为yes  
**pidfile** 配置多个pid的地址，默认在/var/run/redis/pid  
**bind** 绑定ip，设置后只接受来自该ip的请求  
**port** 监听端口，默认是6379  
**timeout** 客户端连接超时的设定，单位是秒  
**loglevel** 分为4级，debug、verbose、notice、warning  
**logfile** 配置log文件地址  
**databases** 设置数据库的个数，默认使用的数据库为0  
**save** 设置redis进行数据库镜像的频率  
**rdbcompression** 在进行镜像备份时，是否进行压缩  
**Dbfilename** 镜像备份文件的文件名  
**Dir** 数据库镜像备份文件的存放路径  
**Slaveof** 设置数据库为其他数据库的从数据库  
**Masterauth** 主数据库连接需要的密码验证  
**Requirepass** 设置登录时，需要使用的密码  
**Maxclients** 设置同时连接的最大客户端数量  
**Maxmemory** 设置redis能够使用的最大内存  
**Appendonly** 开启append only模式  
**Appendfsync** 设置对appendonly.aof文件同步的频率  
**vm-enabled** 是否开启虚拟内存支持  
**vm-swap-file** 设置虚拟内存的交换文件路径  
**vm-max-memory** 设置redis能够使用的最大虚拟内存  
**vm-page-size** 设置虚拟内存的页大小  
**vm-pages** 设置交换文件的总的page数量  
**vm-max-threads** 设置VMIO同时使用的线程数量  
**Glueoutputbuf** 把小的输出缓存存放在一起  
**hash-max-zipmap-entries** 设置hash的临界值  
**Activerehashing** 重新hash

启动后如果使用RedisDesktopManager连接是不成功的，还需要设置密码和设置绑定

#修改配置

cd /usr/local/redis-4.0.11/

vim redis.conf

修改里面的  
Requirepass yourpassword  
bind 0.0.0.0

重新启动Redis即可！！！

**卸载Redis**

停服务、删文件即可！

#查看进程

ps aux |grep redis

#杀掉进程

kill -9 进程号

#查看相关文件

find / -name "redis"

#删除文件

rm -rf 文件

# [在vi或vim上查找字符串](https://www.cnblogs.com/z-x-y/p/9198481.html)

从开头搜索

在命令模式下，输入/你要查找的字符

按下回车，可以看到vim把光标移动到该字符处

再按n（小写）查看下一个匹配

按N(大写）查看上一个匹配（capslock切换大小写，也可以在小写状态下按shift+n）

从结尾处搜索

？要搜索的字符串或字符

搜索后，打开别的文件发现也被高亮了，怎么关闭

命令行模式下，输入：nohlsearch或者：set nohlsearch

可以简写成noh喝set-noh

vi不保存退出的命令

:q!

|  |  |
| --- | --- |
| :wq! | 强制保存文件内容后退出vim编辑器。这个命令的作用是把内存缓冲区中的数据强制写到启动vim时指定的文件中，然后退出vim编辑器。 |
| 命令 | 简单说明 |
| :w | 保存编辑后的文件内容，但不退出vim编辑器。这个命令的作用是把内存缓冲区中的数据写到启动vim时指定的文件中。 |
| :w! | 强制写文件，即强制覆盖原有文件。如果原有文件的访问权限不允许写入文件，例如，原有的文件为只读文件，则可使用这个命令强制写入。但是，这种命令用法仅当用户是文件的属主时才适用，而超级用户则不受此限制。 |
| :wq | 保存文件内容后退出vim编辑器。这个命令的作用是把内存缓冲区中的数据写到启动vim时指定的文件中，然后退出vim编辑器。另外一种替代的方法是用ZZ命令。 |
| :wq! | 强制保存文件内容后退出vim编辑器。这个命令的作用是把内存缓冲区中的数据强制写到启动vim时指定的文件中，然后退出vim编辑器。 |
| ZZ | 使用ZZ命令时，如果文件已经做过编辑处理，则把内存缓冲区中的数据写到启动vim时指定的文件中，然后退出vim编辑器。否则只是退出vim而已。注意，ZZ命令前面无需加冒号“：”，也无需按Enter键。 |
| :q | 在未做任何编辑处理而准备退出vim时，可以使用此命令。如果已做过编辑处理，则vim不允许用户使用“:q”命令退出，同时还会输出下列警告信息：  No write since last change (:quit! overrides) |
| :q! | 强制退出vim编辑器，放弃编辑处理的结果。如果确实不需要保存修改后的文件内容，可输入“:q!”命令，强行退出vim编辑器。 |
| :w filename | 把编辑处理后的结果写到指定的文件中保存 |
| :w! filename | 把编辑处理后的结果强制保存到指定的文件中，如果文件已经存在，则覆盖现有的文件。 |
| :wq! filename | 把编辑处理后的结果强制保存到指定的文件中，如果文件已经存在，则覆盖现有文件，并退出vim编辑器。 |

# 1、安装redis

第一步：下载redis安装包

wget http://download.redis.io/releases/redis-4.0.11.tar.gz

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | [root@localhost local]# wget http://download.redis.io/releases/redis-4.0.11.tar.gz  --2017-12-13 12:35:12--  http://download.redis.io/releases/redis-4.0.11.tar.gz  Resolving download.redis.io (download.redis.io)... 109.74.203.151  Connecting to download.redis.io (download.redis.io)|109.74.203.151|:80... connected.  HTTP request sent, awaiting response... 200 OK  Length: 1723533 (1.6M) [application/x-gzip]  Saving to: ‘redis-4.0.11.tar.gz’    100%[==========================================================================================================>] 1,723,533    608KB/s   in 2.8s    2017-12-13 12:35:15 (608 KB/s) - ‘redis-4.0.11.tar.gz’ saved [1723533/1723533] |

第二步：解压压缩包

tar -zxvf redis-4.0.11.tar.gz

|  |  |
| --- | --- |
| 1 | [root@localhost local]# tar -zxvf redis-4.0.11.tar.gz |

第三步：yum安装gcc依赖

yum install gcc

|  |  |
| --- | --- |
| 1 | [root@localhost local]# yum install gcc |
| 1 | 遇到选择,输入y即可<br><br><br> |

第四步：跳转到redis解压目录下

cd redis-4.0.11

|  |  |
| --- | --- |
| 1 | [root@localhost local]# cd redis-4.0.11 |

第五步：编译安装

make MALLOC=libc

|  |  |
| --- | --- |
| 1 | [root@localhost redis-4.0.11]# make MALLOC=libc |

将/usr/local/redis-4.0.11/src目录下的文件加到/usr/local/bin目录

cd src && make install

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | [root@localhost redis-4.0.11]# cd src && make install      CC Makefile.dep    Hint: It's a good idea to run 'make test' ;)        INSTALL install      INSTALL install      INSTALL install      INSTALL install      INSTALL install |

第六步：测试是否安装成功

先切换到redis src目录下

|  |  |
| --- | --- |
| 1 | [root@localhost redis-4.0.11]# cd src |

1、直接启动redis

./redis-server

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | [root@localhost src]# ./redis-server  18685:C 13 Dec 12:56:12.507 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo  18685:C 13 Dec 12:56:12.507 # Redis version=4.0.11, bits=64, commit=00000000, modified=0, pid=18685, just started  18685:C 13 Dec 12:56:12.507 # Warning: no config file specified, using the default config. In order to specify a config file use ./redis-server /path/to/redis.conf                  \_.\_             \_.-``\_\_ ''-.\_        \_.-``    `.  `\_.  ''-.\_           Redis 4.0.11 (00000000/0) 64 bit    .-`` .-```.  ```\/    \_.,\_ ''-.\_   (    '      ,       .-`  | `,    )     Running in standalone mode   |`-.\_`-...-` \_\_...-.``-.\_|'` \_.-'|     Port: 6379   |    `-.\_   `.\_    /     \_.-'    |     PID: 18685    `-.\_    `-.\_  `-./  \_.-'    \_.-'   |`-.\_`-.\_    `-.\_\_.-'    \_.-'\_.-'|   |    `-.\_`-.\_        \_.-'\_.-'    |           http://redis.io    `-.\_    `-.\_`-.\_\_.-'\_.-'    \_.-'   |`-.\_`-.\_    `-.\_\_.-'    \_.-'\_.-'|   |    `-.\_`-.\_        \_.-'\_.-'    |    `-.\_    `-.\_`-.\_\_.-'\_.-'    \_.-'        `-.\_    `-.\_\_.-'    \_.-'            `-.\_        \_.-'                `-.\_\_.-'    18685:M 13 Dec 12:56:12.508 # WARNING: The TCP backlog setting of 511 cannot be enforced because /proc/sys/net/core/somaxconn is set to the lower value of 128.  18685:M 13 Dec 12:56:12.508 # Server initialized  18685:M 13 Dec 12:56:12.508 # WARNING overcommit\_memory is set to 0! Background save may fail under low memory condition. To fix this issue add 'vm.overcommit\_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit\_memory=1' for this to take effect.  18685:M 13 Dec 12:56:12.508 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will create latency and memory usage issues with Redis. To fix this issue run the command 'echo never > /sys/kernel/mm/transparent\_hugepage/enabled' as root, and add it to your /etc/rc.local in order to retain the setting after a reboot. Redis must be restarted after THP is disabled.  18685:M 13 Dec 12:56:12.508 \* Ready to accept connections |

如上图：redis启动成功，但是这种启动方式需要一直打开窗口，不能进行其他操作，不太方便。

按 ctrl + c可以关闭窗口

2、以后台进程方式启动redis

第一步：修改redis.conf文件

[root@localhost src]# cd ..

[root@localhost redis-4.0.11]# ls

00-RELEASENOTES CONTRIBUTING deps Makefile README.md runtest runtest-sentinel src utils

BUGS COPYING INSTALL MANIFESTO redis.conf runtest-cluster sentinel.conf tests

[root@localhost redis-4.0.11]# vim redis.conf

redis.conf    是redis的配置文件，启动需要指定启用该文件的配置方式启动。该文件在    redis-4.0.11  目录下面。（如果你没有vim  就使用 vi  这个编辑命令。vim安装方式  ：  yum install vim）

将

|  |  |
| --- | --- |
| 1 | daemonize no |

修改为

|  |  |
| --- | --- |
| 1 | daemonize yes |

第二步：指定redis.conf文件启动,此时启动要到src目录下去启动，

|  |  |
| --- | --- |
| 1 | ./redis-server ../redis.conf |
| 1  2  3  4 | [root@localhost src]# ./redis-server  ../redis.conf  18713:C 13 Dec 13:07:41.109 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo  18713:C 13 Dec 13:07:41.109 # Redis version=4.0.11, bits=64, commit=00000000, modified=0, pid=18713, just started  18713:C 13 Dec 13:07:41.109 # Configuration loaded |

第三步：关闭redis进程

首先使用ps -aux | grep redis查看redis进程

|  |  |
| --- | --- |
| 1  2  3 | [root@localhost src]# ps -aux | grep redis  root     18714  0.0  0.1 141752  2008 ?        Ssl  13:07   0:00 ./redis-server 127.0.0.1:6379  root     18719  0.0  0.0 112644   968 pts/0    R+   13:09   0:00 grep --color=auto redis |

使用kill命令杀死进程

|  |  |
| --- | --- |
| 1 | [root@localhost src]# kill -9 18714 |

第七步：设置redis开机自启动

1、在/etc目录下新建redis目录

mkdir redis

|  |  |
| --- | --- |
| 1 | [root@localhost etc]# mkdir redis |

2、将/usr/local/redis-4.0.11/redis.conf 文件复制一份到/etc/redis目录下，并命名为6379.conf

|  |  |
| --- | --- |
| 1 | [root@localhost redis]# cp /usr/local/redis-4.0.11/redis.conf /etc/redis/6379.conf |

3、将redis的启动脚本复制一份放到/etc/init.d目录下

|  |  |
| --- | --- |
| 1 | [root@localhost init.d]# cp /usr/local/redis-4.0.11/utils/redis\_init\_script /etc/init.d/redisd |

4、设置redis开机自启动

先切换到/etc/init.d目录下

然后执行自启命令

|  |  |
| --- | --- |
| 1  2 | [root@localhost init.d]# chkconfig redisd on  service redisd does not support chkconfig |

看结果是redisd不支持chkconfig

解决方法：

使用vim编辑redisd文件，在第一行加入如下两行注释，保存退出

|  |  |
| --- | --- |
| 1  2 | # chkconfig:   2345 90 10  # description:  Redis is a persistent key-value database |

注释的意思是，redis服务必须在运行级2，3，4，5下被启动或关闭，启动的优先级是90，关闭的优先级是10。

再次执行开机自启命令，成功

|  |  |
| --- | --- |
| 1 | [root@localhost init.d]# chkconfig redisd on |

现在可以直接已服务的形式启动和关闭redis了

启动：

service redisd start

|  |  |
| --- | --- |
| 1  2  3  4  5 | [root@localhost ~]# service redisd start  Starting Redis server...  2288:C 13 Dec 13:51:38.087 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo  2288:C 13 Dec 13:51:38.087 # Redis version=4.0.11, bits=64, commit=00000000, modified=0, pid=2288, just started  2288:C 13 Dec 13:51:38.087 # Configuration loaded |

关闭：

service redisd stop

|  |  |
| --- | --- |
| 1  2  3 | [root@localhost ~]# service redisd stop  Stopping ...  Redis stopped |

参考资料：

1、http://blog.csdn.net/zc474235918/article/details/50974483

2、http://blog.csdn.net/gxw19874/article/details/51992125

如果出现如下问题：

|  |  |
| --- | --- |
| 1  2 | [root@localhost ~]# service redisd start  /var/run/redis\_6379.pid exists, process is already running or crashed |

 到    /var/run/    目录下删除    redis\_6379.pid   文件即可

**下面选看。但很重要**

以下为设置redis登陆密码，在服务器上装完redis后因为没有密码。容易被攻击。

博主一个国外的服务器因为没有设置密码被连续攻击后，服务器公司直接给封ip一年。歇菜。

另一个阿里云服务器被攻击，直接redis端口对外开放被暂停3天。歇菜。mmp

redis在生产环境中通常都会设置密码以保证一定的安全性，本篇blog就简单记录一下如何在redis中设置客户端登录密码。

# 修改redis.conf，如果设置过自启动则修改 /etc/redis/6379.conf

如果没有设置则在redis的解压目录下修改

[root@wangcheng redis-4.0.11]# ls

00-RELEASENOTES CONTRIBUTING deps Makefile README.md runtest runtest-sentinel src utils

BUGS COPYING INSTALL MANIFESTO redis.conf runtest-cluster sentinel.conf tests

RT，打开redis.conf文件，搜索requirepass关键字，如下图：

关注标记的那一行，#requirepass foobared。设置密码的方法就是去掉注释的#，把foobared替换成自己的密码即可，例如将密码设置为123456： 

修改完成后重启redis，再次通过redis客户端redis-cli登录并操作可以发现会报一个身份认证错误： 

这就说明我们已经成功的设置了密码，所以通过客户端连接的话必须加上密码参数才能正常连接： 

如上图所示，加了-a参数之后即可正常连接并操作redis。

# jedis设置密码

当我们用Java客户端连接redis时会遇到同样的问题，下面看一段简单的jedis连接redis的测试代码：

所以在创建了Jedis的实例后再加上一行jedis.auth("123456"); 即可，最后看一下运行结果： 

# spring-data-redis设置密码

通常情况下在实际的java项目中我们会选择Spring提供的spring-data-redis来操作redis，spring的封装可以给我们提供很多便捷之处。那么spring-data-redis又是如何设置密码的呢？

首先定义一个redis.properties配置文件，定义一组redis属性供spring加载使用，其中就包含密码（redis.password）

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | # Redis settings  redis.host=192.168.145.10  redis.port=6379  redis.password=123456  redis.timeout=100000  redis.maxTotal=300  redis.maxIdle=100  redis.maxWaitMillis=1000  redis.testOnBorrow=true或者如下： |

spring.redis.database=0  
spring.redis.host=192.168.145.10  
  
spring.redis.port=6379

spring.redis.password=123456

|  |  |
| --- | --- |
| 1 | <br> |

下面可以忽略不理会。如果上面没有设置成功，可采用下面方法

一.**如何初始化redis的密码？**

总共2个步骤：

a.在配置文件中有个参数： requirepass  这个就是配置redis访问密码的参数。

比如 requirepass test123

b.配置文件中参数生效需要重启重启redis 。

二**.不重启redis如何配置密码?**

a. 在配置文件中配置requirepass的密码（当redis重启时密码依然有效）。

# requirepass foobared  
 如  修改成 :

requirepass  test123

b. 进入redis重定义参数

查看当前的密码：

[root@slaver251 redis-2.4.16]# service redisd start  
redis 127.0.0.1:6379>  
redis 127.0.0.1:6379> config get requirepass  
1) "requirepass"  
2) (nil)

显示密码是空的，

然后设置密码：

redis 127.0.0.1:6379> config set requirepass test123  
OK

再次查询密码：

redis 127.0.0.1:6379> config get requirepass  
(error) ERR operation not permitted

此时报错了！

现在只需要密码认证就可以了。

redis 127.0.0.1:6379> auth test123  
OK

再次查询密码：

redis 127.0.0.1:6379> config get requirepass  
1) "requirepass"  
2) "test123"

密码已经得到修改。

当到了可以重启redis的时候 由于配置参数已经修改 所以密码会自动生效。

要是配置参数没添加密码 那么redis重启 密码将相当于没有设置。

三.**如何登录有密码的redis？**

a.在登录的时候 密码就输入

[root@slaver251 redis-2.4.16]#  service redisd start  
redis 127.0.0.1:6379>  
redis 127.0.0.1:6379> config get requirepass  
1) "requirepass"  
2) "test123"

b.先登录再验证：

[root@slaver251 redis-2.4.16]#  service redisd start  
redis 127.0.0.1:6379>  
redis 127.0.0.1:6379> auth test123  
OK  
redis 127.0.0.1:6379> config get requirepass  
1) "requirepass"  
2) "test123"  
redis 127.0.0.1:6379>

四. **master 有密码,slave 如何配置？**

当**master** 有密码的时候 配置slave 的时候 相应的密码参数也得相应的配置好。不然slave 是无法进行正常复制的。

相应的参数是：

#masterauth

比如:

masterauth  mstpassword

centos 卸载mysql

采用yum安装mysql后，如果想要完全卸载mysql，可以采用如下方式：  
1、查看mysql安装了哪些东西

rpm -qa |grep -i mysql

image.png

2、开始卸载

yum remove mysql-community-common-5.7.20-1.el7.x86\_64

yum remove mysql-community-client-5.7.20-1.el7.x86\_64

yum remove mysql57-community-release-el7-11.noarch

yum remove mysql-community-libs-5.7.20-1.el7.x86\_64

yum removemysql-community-server-5.7.20-1.el7.x86\_64

3、查看是否卸载完成

image.png

4、查找mysql相关目录

find / -name mysql

image.png

5、删除相关目录

rm -rf

image.png

6、删除/etc/my.cnf

rm -rf /etc/my.cnf

7、删除/var/log/mysqld.log（如果不删除这个文件，会导致新安装的mysql无法生存新密码，导致无法登陆）

rm -rf /var/log/mysqld.log

卸载Centos7自带的Maridb数据库

查看是否存在Maridb数据库，然后删除。

[root@192 ~]# rpm -qa | grep mariadb

mariadb-libs-5.5.60-1.el7\_5.x86\_64

[root@192 ~]# yum -y remove mari\*

[root@192 ~]# rm -rf /var/lib/mysql/\*

# . Centos7.5 安装 mysql 5.7 ( 卸载自带 MariaDB)（实测）

[镜中隐](https://me.csdn.net/qq_36401512) 2018-12-04 21:30:32  1614  收藏 1

分类专栏： [python](https://blog.csdn.net/qq_36401512/category_8388103.html)

ps原作者的话介绍下背景：CentOS 7 版本将MySQL数据库软件从默认的程序列表中移除，用MariaDB代替了，MariaDB数据库管理系统是MySQL的一个分支，主要由开源社区在维护，采用GPL授权许可。开发这个分支的原因之一是：甲骨文公司收购了MySQL后，有将MySQL闭源的潜在风险，因此社区采用分支的方式来避开这个风险。MariaDB的目的是完全兼容MySQL，包括API和命令行，使之能轻松成为MySQL的代替品。

# 1.卸载自带 MariaDB（安装centos7时，我时选择了桌面和开发工具）

yum list installed | grep mariadb    #检查mariadb是否已安装

[root@DESKTOP-2T8A4KQ ntsyai]# yum list installed | grep mariadb  
mariadb-libs.x86\_64                     1:5.5.56-2.el7                 @anaconda

yum -y remove mariadb\*    #全部卸载

安装大小：17 M  
Downloading packages:  
Running transaction check  
Running transaction test  
Transaction test succeeded  
Running transaction  
  正在删除    : 2:postfix-2.10.1-6.el7.x86\_64                                                                                     1/2  
  正在删除    : 1:mariadb-libs-5.5.56-2.el7.x86\_64                                                                                2/2  
  验证中      : 1:mariadb-libs-5.5.56-2.el7.x86\_64                                                                                1/2  
  验证中      : 2:postfix-2.10.1-6.el7.x86\_64                                                                                     2/2

删除:  
  mariadb-libs.x86\_64 1:5.5.56-2.el7

作为依赖被删除:  
  postfix.x86\_64 2:2.10.1-6.el7

完毕！

# 2.下载并安装mysql的YUM源（篇幅太长不显示结果）

下载mysql的YUM源：wget -P /home/ntsyai http://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm

由于我们是下载到/home/ntsyai目录下，所以先切换到该目录下：cd /home/ntsyai

安装mysql的YUM源：rpm -ivh mysql57-community-release-el7-11.noarch.rpm

检查mysql的YUM源是否安装成功：yum repolist enabled | grep "mysql.\*-community.\*"

[root@DESKTOP-2T8A4KQ ntsyai]# yum repolist enabled | grep "mysql.\*-community.\*"  
mysql-connectors-community/x86\_64 MySQL Connectors Community                  74  
mysql-tools-community/x86\_64      MySQL Tools Community                       74  
mysql57-community/x86\_64          MySQL 5.7 Community Server                 307

出现如结果就表明安装成功了

选择要启用的mysql版本

查看mysql版本,执行：yum repolist all | grep mysql

可以通过类似下面的语句来启动或禁用某些版本

yum-config-manager --enable mysql57-community

yum-config-manager --disable mysql56-community

或者通过修改vim /etc/yum.repos.d/mysql-community.repo文件，改变默认安装的mysql版本。比如要安装5.6版本，将5.7源的enabled=1改成enabled=0，然后再将5.6源的enabled=0改成enabled=1即可。

注意： 任何时候，只能启用一个版本。

查看当前的启用的 MySQL 版本：yum repolist enabled | grep mysql

# 3.安装mysql

yum install mysql-community-server

安装过程中一直输入"y"就可以了，当出现下面的结果时，就代表mysql数据库安装成功了

已安装:  
  mysql-community-server.x86\_64 0:5.7.24-1.el7

作为依赖被安装:  
  mysql-community-client.x86\_64 0:5.7.24-1.el7                      mysql-community-common.x86\_64 0:5.7.24-1.el7                       
  mysql-community-libs.x86\_64 0:5.7.24-1.el7

完毕！

# 4.测试

启动mysql服务：systemctl start mysqld

登录进Mysql：mysql -uroot或mysql

如果出现错误：ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)

则说明mysql安装完后给root用户生成了一个默认密码，所以你需要使用密码登录。

查看默认密码：grep 'temporary password' /var/log/mysqld.log

[root@DESKTOP-2T8A4KQ ntsyai]# grep 'temporary password' /var/log/mysqld.log  
2018-12-04T12:29:10.163080Z 1 [Note] A temporary password is generated for root@localhost: gvpiqdLbG0+/

mysql -uroot -p或mysql -p

回车 然后输入默认密码即可登录myql

退出登入exit；即可。

可能用到的命令：

systemctl start mysqld    #启动mysqld

systemctl stop mysqld    #停止mysqld

systemctl restart mysqld    #重启mysqld

systemctl enable mysqld   #设置开机启动

systemctl status mysqld    #查看 MySQL Server 状态

5、mysql相关配置

设置密码

grep 'temporary password' /var/log/mysqld.log 显示密码

mysql -uroot -p

接着输入密码

mysql> set global validate\_password\_special\_char\_count=0;

mysql> set global validate\_password\_length=6

mysql> set global validate\_password\_mixed\_case\_count=0;

mysql> set global validate\_password\_number\_count=0;

mysql> set global validate\_password\_policy=0;

mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'tomtaw'

CentOS 7中默认是firewalld防火墙，如果使用iptables需要先关闭firewalld防火墙（1.关闭防火墙，2.取消开机启动）。

#关闭firewalld

systemctl stop firewalld

#取消开机启动

systemctl disable firewalld

#查看状态

firewall-cmd --state

# 1|0安装启用iptables

#先检查是否安装了iptables

service iptables status

#安装iptables

yum install -y iptables

#安装iptables-services

yum -y install iptables-services

#注册iptables服务，相当于以前的chkconfig iptables on

systemctl enable iptables.service

#开启服务

systemctl start iptables.service

#查看状态

systemctl status iptables.service

# 2|0设置iptables规则

查看iptables现有规则：iptables -L -n  
添加开放80端口：iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT（好像这种方式重启以后就失效了）  
编辑配置文件方式添加端口：  
文件位置：/etc/sysconfig/iptables  
执行vi /etc/sysconfig/iptables  
添加一条规则开放80端口：-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT  
重启生效：service restart iptables

# 3|0关闭iptables

systemctl stop iptables

# 4|0其他规则：

#查看iptables现有规则

iptables -L -n

#先允许所有,不然有可能会杯具

iptables -P INPUT ACCEPT

#清空所有默认规则

iptables -F

#清空所有自定义规则

iptables -X

#所有计数器归0

iptables -Z

#允许来自于lo接口的数据包(本地访问)

iptables -A INPUT -i lo -j ACCEPT

#开放22端口

iptables -A INPUT -p tcp --dport 22 -j ACCEPT

#开放21端口(FTP)

iptables -A INPUT -p tcp --dport 21 -j ACCEPT

#开放80端口(HTTP)

iptables -A INPUT -p tcp --dport 80 -j ACCEPT

#开放443端口(HTTPS)

iptables -A INPUT -p tcp --dport 443 -j ACCEPT

#允许ping

iptables -A INPUT -p icmp --icmp-type 8 -j ACCEPT

#允许接受本机请求之后的返回数据 RELATED,是为FTP设置的

iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

#其他入站一律丢弃

iptables -P INPUT DROP

#所有出站一律绿灯

iptables -P OUTPUT ACCEPT

#所有转发一律丢弃

iptables -P FORWARD DROP

**开启 MySQL 远程访问权限：** 在linux系统上登陆mysql服务。

-- root 是用户名

[root@localhost ~]# mysql -u root -p

Enter password: -- 输入密码

**创建远程连接 MySQL 的用户：**

-- 创建用户、密码及权限范围 第一个 roo t为用户名 @后为适用的主机，‘%’表示所有电脑都可以访问连接，第二个 root 为密码

mysql> GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'192.168.0.2' IDENTIFIED BY 'root' WITH GRANT OPTION;

Query OK, 0 rows affected (1.57 sec)

-- 立即生效

mysql> flush privileges;

Query OK, 0 rows affected (0.00 sec)

**查看数据库用户：**

-- 使用 mysql 库

mysql> use mysql;

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

-- 查看用户

mysql> SELECT DISTINCT CONCAT('User: [', user, '''@''', host, '];') AS USER\_HOST FROM user;

+---------------------------------------+

| USER\_HOST |

+---------------------------------------+

| User: [root'@'127.0.0.1]; |

| User: [root'@'192.168.0.2]; |

| User: [root'@'::1]; |

| User: [root'@'localhost]; |

| User: [root'@'localhost.localdomain]; |

+---------------------------------------+

5 rows in set (0.00 sec)

创建成功。

**查看端口：**

mysql> show global variables like 'port';

+---------------+-------+

| Variable\_name | Value |

+---------------+-------+

| port | 3306 |

+---------------+-------+

1 row in set (0.01 sec)

**开启3306端口：**

[root@localhost ~]# vim /etc/sysconfig/iptables

# Firewall configuration written by system-config-firewall

# Manual customization of this file is not recommended.

\*filter

:INPUT ACCEPT [0:0]

:FORWARD ACCEPT [0:0]

:OUTPUT ACCEPT [0:0]

-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

-A INPUT -p icmp -j ACCEPT

-A INPUT -i lo -j ACCEPT

-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT

-A INPUT -m state --state NEW -m tcp -p tcp --dport 3306 -j ACCEPT

-A INPUT -j REJECT --reject-with icmp-host-prohibited

-A FORWARD -j REJECT --reject-with icmp-host-prohibited

COMMIT

"/etc/sysconfig/iptables" 14L, 543C

**注意：**一定要加在后两行的前面。

**重启防火墙：**

[root@localhost ~]# service iptables restart

iptables：将链设置为政策 ACCEPT：filter [确定]

iptables：清除防火墙规则： [确定]

iptables：正在卸载模块： [确定]

iptables：应用防火墙规则： [确定]

**查看服务器ip地址：**

[root@localhost ~]# ifconfig

eth0 Link encap:Ethernet HWaddr 00:0C:29:16:F4:E6

inet addr:192.168.0.123 Bcast:192.168.0.255 Mask:255.255.255.0

inet6 addr: hjjj:jjji::iii:oooo:oooo:ioio/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:2931 errors:0 dropped:0 overruns:0 frame:0

TX packets:1631 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:322681 (315.1 KiB) TX bytes:266043 (259.8 KiB)

lo Link encap:Local Loopback

inet addr:127.0.0.1 Mask:255.0.0.0

inet6 addr: ::1/128 Scope:Host

UP LOOPBACK RUNNING MTU:16436 Metric:1

RX packets:0 errors:0 dropped:0 overruns:0 frame:0

TX packets:0 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:0

RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)

**连接测试：**

my.cnf是mysql启动时加载的配置文件，一般会放在mysql的安装目录中，用户也可以放在其他目录加载。

安装mysql后，系统中会有多个my.cnf文件，有些是用于测试的。

使用locate my.cnf命令可以列出所有的my.cnf文件

**命令**

locate my.cnf