

常用命令：

- 1、关机：shutdown
- 2、安装unzip命令：yum install unzip
- 3、删除目录：rm -rf 目录名
- 4、强制终止程序的执行：Ctrl + C
- 5、安装：rpm
- 6、检查tomcat是否启动：ps -ef |grep tomcat

```
[root@localhost conf]# ps -ef |grep tomcat
root      4701      1  0 14:32 pts/0    00:00:00 /bin/sh /usr/local/tomcat/bin/catalina.sh start
root      4702      1  0 14:32 pts/0    00:00:00 /usr/local/sbin/cronolog /usr/local/tomcat/logs/catalina.%Y-%m-%d.out
root      4703    4701 10 14:32 pts/0    00:02:12 /usr/java/jdk1.8.0_181-amd64/jre/bin/java -Djava.util.logging.config.file=/usr/local/tomcat/conf/logg
ng.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom -serve
r -Xms1024m -Xmx2048m -XX:PermSize=128M -XX:MaxNewSize=512m -XX:MaxPermSize=256m -Djdk.tls.ephemeralDHKeySize=2048 -Djava.protocol.handler.pkgs=org.apac
e.catalina.webresources -Dorg.apache.catalina.security.SecurityListener.Umask=0027 -Dignore.endorsed.dirs= -classpath /usr/local/tomcat/bin/bootstrap.jar
r:/usr/local/tomcat/bin/tomcat-juli.jar -Dcatalina.base=/usr/local/tomcat -Dcatalina.home=/usr/local/tomcat -Djava.io.tmpdir=/usr/local/tomcat/temp org
apache.catalina.startup.Bootstrap start
root      5027    3640  0 14:53 pts/0    00:00:00 grep --color=auto tomcat
```

- 7、查看日志：tail -500f ../logs/catalina.2020-12-07.out
- 8、查看空间磁盘空间：df -h
- 9、使用nmap工具查询端口使用情况 nmap 127.0.0.1
- 11、查看文件属性：ls -al

```
[root@www ~]# ls -al
total 156
drwxr-x---  4 root root  4096 Sep  8 14:06 .
drwxr-xr-x 23 root root  4096 Sep  8 14:21 ..
-rw-----  1 root root  1474 Sep  4 18:27 anaconda-ks.cfg
-rw-r--r--  1 root root   100 Jan  6  2007 .cshrc
drwx-----  3 root root  4096 Sep  5 10:37 .gconf      <=范例说明处
drwx-----  2 root root  4096 Sep  5 14:09 .gconfd
-rw-r--r--  1 root root 42304 Sep  4 18:26 install.log <=范例说明处
-rw-r--r--  1 root root  5661 Sep  4 18:25 install.log.syslog
[  1  ] [  2  ] [  3  ] [  4  ] [  5  ] [  6  ] [  7  ]
[ 权限 ] [文件数] [拥有者] [群组]  [文件大小] [  修改日期  ] [  文件名  ]
```

- 12、修改所属组群：chgrp -R dirname/filename ...

选项与参数：

-R：进行递归(recursive)的持续变更，亦即连同次目录下的所有文件、目录都更新成为这个群组之意。常常用在变更某一目录内所有的文件之情况。

- 13、修改文件拥有者：chown -R 账号名称:组名 文件或目录

选项与参数：

-R：进行递归(recursive)的持续变更，亦即连同次目录下的所有文件都变更

- 14、卸载软件

yum install Percona-XtraDB-Cluster-57*

- 15、永久修改主机名

hostnamectl set-hostname xxx

重启就好了

解压：

- 1、*.tar 用 tar -xvf 解压 -C 加路径名
- 2、*.gz 用 gzip -d或者gunzip 解压 -C 加路径名
- 3、*.tar.gz和*.tgz 用 tar -xzf 解压
- 4、*.bz2 用 bzip2 -d或者用bunzip2 解压
- 5、*.tar.bz2用tar -xjf 解压
- 6、*.Z 用 uncompress 解压

- 7、*.tar.Z 用tar -xZf 解压
- 8、*.rar 用 unrar e解压
- 9、*.zip 用 unzip 解压 -d 加路径名

防火墙:

- 1、关闭系统防火墙: systemctl stop firewalld
 - 2、启动系统防火墙: systemctl start firewalld
 - 3、查看系统防火墙状态: systemctl status firewalld
 - 4、重启系统防火墙: firewall-cmd --reload
 - 5、查看所有开启的访问端口: firewall-cmd --list-ports
 - 6、防火墙开启端口访问: firewall-cmd --zone=public --add-port=80/tcp --permanent
- 命令含义: --zone #作用域 --add-port=80/tcp #添加端口, 格式为: 端口/通讯协议 --permanent

MySQL:

- 1、进入: mysql -p (输入密码)
- 2、退出: quit 使用数据库: use 数据库名
- 3、执行sql脚本: source 路径下的sql脚本
- 4、查看安装后的随机密码: cat /var/log/mysql.log | grep password
- 5、停止运行数据库: sudo systemctl stop mysqld
- 6、查看数据库运行状态: service mysqld status

```
[sxzyzb@localhost mysql]$ service mysqld status
Redirecting to /bin/systemctl status mysqld.service
● mysqld.service - MySQL Server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; vendor preset: disabled)
   Active: inactive (dead) since 二 2020-12-15 18:35:19 CST; 3min 10s ago
     Docs: man:mysqld(8)
           http://dev.mysql.com/doc/refman/en/using-systemd.html
   Process: 15145 ExecStart=/usr/sbin/mysqld $MYSQLD_OPTS (code=exited, status=0/SUCCESS)
   Process: 15114 ExecStartPre=/usr/bin/mysqld_pre_systemd (code=exited, status=0/SUCCESS)
  Main PID: 15145 (code=exited, status=0/SUCCESS)
   Status: "Server shutdown complete"

12月 15 18:32:26 localhost.localdomain systemd[1]: Starting MySQL Server...
12月 15 18:32:28 localhost.localdomain systemd[1]: Started MySQL Server.
12月 15 18:35:16 localhost.localdomain systemd[1]: Stopping MySQL Server...
12月 15 18:35:19 localhost.localdomain systemd[1]: Stopped MySQL Server.
[sxzyzb@localhost mysql]$
```

报错及解决方法:

Q: 打开Vi编辑器时提示E325: ATTENTION的解决方法:

A: 删除当前目录下的.swp文件

Q: 在安装MySQL8.0时, 修改临时密码, 因密码过于简单 (如: 123456), 不符合MySQL密码规范, 会触发一个报错信息:

ERROR 1819 (HY000): Your password does not satisfy the current policy requirements.

A: 1、解决办法调整MySQL密码验证规则, 修改 policy 和 length 的值。

2、MySQL 5.7 进行如下设置, 即可解决问题:

mysql> set global validate_password_policy=0;

mysql> set global validate_password_length=1;

3、MySQL 8.0 执行代码:

mysql> set global validate_password_policy=0;

ERROR 1193 (HY000): Unknown system variable 'validate_password_policy'

mysql> set global validate_password_length=1;

ERROR 1193 (HY000): Unknown system variable 'validate_password_length'

问题解决

1、分析： 可以看到，修改 policy 和 length 的值，在MySQL5.7中好使，在MySQL8.0中无效。'validate_password_policy' 变量不存在。

2、解决： 先修改一个满足的密码（如：Root_12root）

3、密码修改后，可用命令查看 validate_password 密码验证插件是否安装。

mysql> SHOW VARIABLES LIKE 'validate_password%';

MySQL 8.0 调整密码验证规则：

mysql> set global validate_password.policy=0;

mysql> set global validate_password.length=1;

重新设置密码： ALTER USER 'root'@'localhost' IDENTIFIED BY '你的密码';

问题总结

1、通过查看 MySQL5.7 和 MySQL8.0 密码验证插件对比，可知两个版本中，变量名不一样。（_password_policy 和 _password.policy） --- 问题原因所在。

Q：安装unzip命令

A：yum install unzip

Q：CentOS7 yum安装软件提示:

another app is currently holding the yum lock;waiting for it to exit

A：rm -f /var/run/yum.pid

Q：日志出现这样的错误

```
12-Dec-2020 14:17:20.136 警告 [localhost-startStop-1] org.apache.catalina.loader.WebappClassLoaderBase.clearReferencesThreads Web应用程序[ROOT]似乎启动了一个名为[Druid-ConnectionPool-Create-568245235]的线程，但未能停止它。这可能会造成内存泄漏。线程的堆栈跟踪：[sun.misc.Unsafe.park(Native Method)
java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(AbstractQueuedSynchronizer.java:2039)
com.alibaba.druid.pool.DruidDataSource$CreateConnectionThread.run(DruidDataSource.java:1837)]
```

A：数据库链接有问题

打开并修改 “/项目文件夹/WEB-INF/classes/jeesite.properties”

Q：yum命令时报错

```
Error downloading packages:
Percona-XtraDB-Cluster-server-57-5.7.31-31.45.3.el7.x86_64: [Errno 256] No more mirrors to try.

[root@192 ~]# yum clean all
已加载插件：fastestmirror, langpacks
正在清理软件源： base extras percona-release-noarch percona-release-x86_64 updates
Cleaning up everything
Cleaning up list of fastest mirrors
```

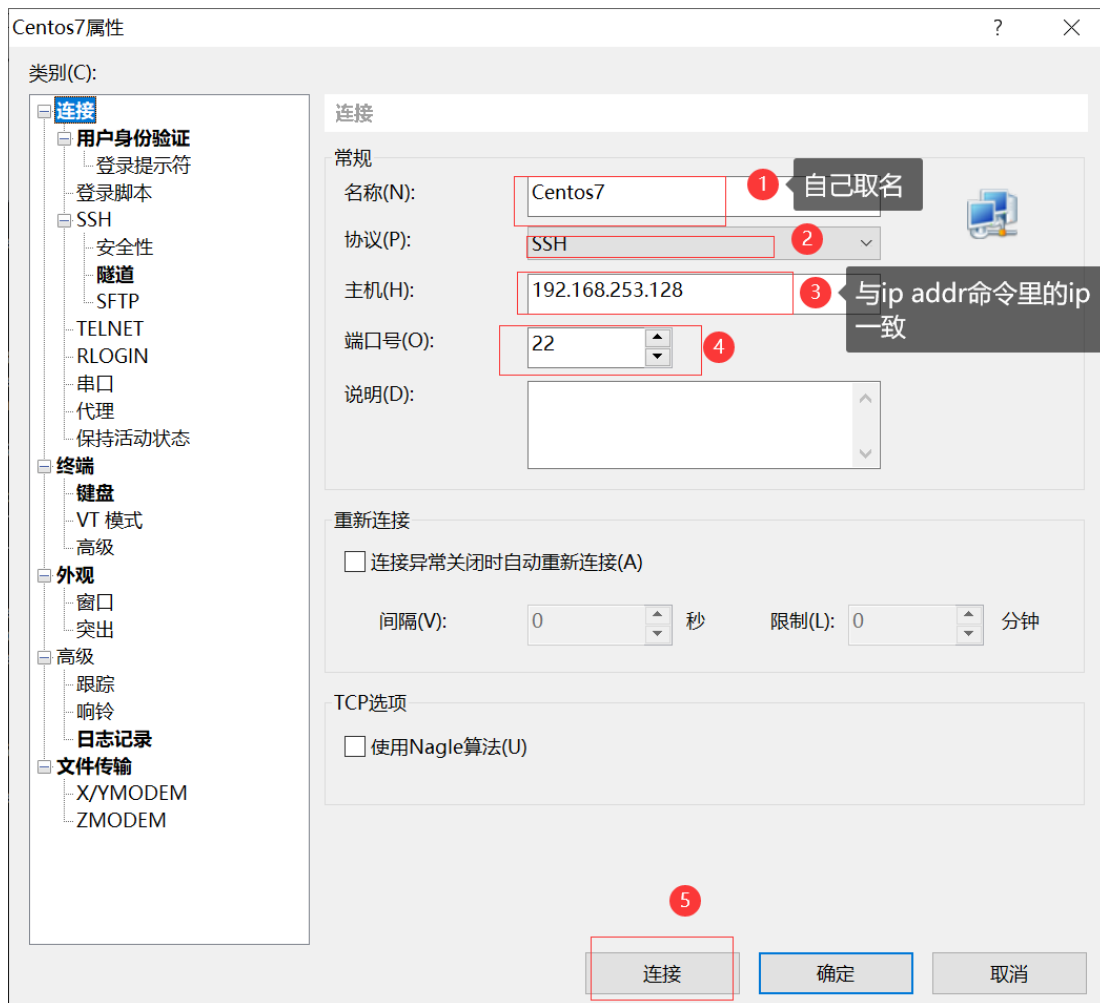
A：执行命令yum clean all

清理客户端yum源配置的缓存

1.安装Xshell与Xftp连接CentOS-7

Xshell

新建会话



Xftp

Centos7属性

常规 选项

站点

名称(N): Centos7 1 自己取名

主机(H): 192.168.253.128 2 与ip addr命令的ip一致

协议(R): SFTP 3 设置(S)...

端口号(O): 22 4

代理服务器(X): <无> 浏览(W)...

说明(D):

登录

☐ 匿名登录(A)

☐ 使用身份验证代理(G)

方法(M): Password 5 设置(S)...

用户名(U): root 6 CentOS-7的用户名

密码(P): 7 CentOS-7的密码

用户密钥(K): <无> 浏览(B)...

密码(E):

8

连接 确定 取消

第四步端口为22;

2.安装JDK

检查系统是否自带

命令: java -version

```
[root@localhost ~]# java -version
openjdk version "1.8.0_102"
OpenJDK Runtime Environment (build 1.8.0_102-b14)
OpenJDK 64-Bit Server VM (build 25.102-b14, mixed mode)
```

卸载

查询所安装的JDK套件

命令: rpm -qa | grep java

```
[root@localhost share]# rpm -qa | grep java
python-javapackages-3.4.1-11.el7.noarch
tzdata-java-2016g-2.el7.noarch
java-1.8.0-openjdk-headless-1.8.0.102-4.b14.el7.x86_64
java-1.8.0-openjdk-1.8.0.102-4.b14.el7.x86_64
javapackages-tools-3.4.1-11.el7.noarch
```

下面这几个可以删除

```
java-1.7.0-openjdk-1.7.0.111-2.6.7.8.el7.x86_64
java-1.8.0-openjdk-1.8.0.102-4.b14.el7.x86_64
java-1.8.0-openjdk-headless-1.8.0.102-4.b14.el7.x86_64
java-1.7.0-openjdk-headless-1.7.0.111-2.6.7.8.el7.x86_64
```

noarch文件可以不用删除

```
python-javapackages-3.4.1-11.el7.noarch
tzdata-java-2016g-2.el7.noarch
javapackages-tools-3.4.1-11.el7.noarch
```

删除命令

```
rpm -e --nodeps java-1.8.0-openjdk-1.8.0.102-4.b14.el7.x86_64
rpm -e --nodeps java-1.8.0-openjdk-headless
```

将JDK的rpm包上传至系统，执行rpm命令

命令: rpm -ivh jdk-8u181-linux-x64.rpm

JDK环境配置

命令: vi /etc/profile

文件末尾加上

```
JAVA_HOME=/usr/java/jdk1.8.0_181-amd64
```

```
JRE_HOME=/usr/java/jdk1.8.0_181-amd64/jre
```

```
PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin
```

```
CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:$JRE_HOME/lib
```

```
export JAVA_HOME JRE_HOME PATH CLASSPATH
```

(注意JAVA_HOME、JRE_HOME为安装目录)

验证

运行命令java -version

```
[root@192 java]# java -version
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
[root@192 java]#
```

3.安装tomcat

将tomcat两个压缩包上传到系统，解压

📁 Tomcat日志分割

📁 Tomcat中间件

运行cronolog-1.6.2/下的configure

./configure

出现

```
configure: error: no acceptable cc found in $PATH
```

再make

```
make: *** 没有指明目标并且找不到 makefile。 停止。
```

解决方法：

安装gcc命令

yum install gcc

再次./configure

```
[root@localhost cronolog-1.6.2]# ./configure
loading cache ./config.cache
checking for a BSD compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking whether make sets ${MAKE}... yes
checking for working aclocal... missing
checking for working autoconf... missing
checking for working automake... missing
checking for working autoheader... missing
checking for working makeinfo... missing
checking for gcc... gcc
checking whether the C compiler (gcc ) works... yes
checking whether the C compiler (gcc ) is a cross-compiler... no
checking whether we are using GNU C... yes
checking whether gcc accepts -g... yes
checking for a BSD compatible install... /usr/bin/install -c
checking whether ln -s works... yes
checking for ranlib... ranlib
checking for perl... /usr/bin/perl
checking how to run the C preprocessor... gcc -E
checking for ANSI C header files... yes
checking whether stat file-mode macros are broken... no
checking whether time.h and sys/time.h may both be included... yes
checking whether struct tm is in sys/time.h or time.h... time.h
checking for tm_zone in struct tm... yes
checking for fcntl.h... yes
checking for limits.h... yes
checking for unistd.h... yes
checking for working const... yes
checking for size_t... yes
checking whether struct tm is in sys/time.h or time.h... (cached) time
checking for strftime... yes
checking for vprintf... yes
checking for mkdir... yes
```

运行成功

执行make

```
[root@localhost cronolog-1.6.2]# make
Making all in lib
make[1]: 进入目录"/usr/local/cronolog-1.6.2/lib"
gcc -DPACKAGE=\"cronolog\" -DVERSION=\"1.6.2\" -DSTDC_HEADERS=1 -DTIME_WITH_SYS_TIME=1 -DHAVE_TM_ZONE=1 -DHAVE_FCNTL_H=1 -DHAVE_LIMITS_H=1 -DHAVE_UNISTD_H=1 -DHAVE_STRFTIME=1 -DHAVE_VPRINTF=1 -DHAVE_MKDIR=1 -DHAVE_MKTIME=1 -DHAVE_PUTENV=1 -DHAVE_STRPTIME=1 -DHAVE_LOCALTIME_R=1 -I. -I. -g -O2 -c topt.c
gcc -DPACKAGE=\"cronolog\" -DVERSION=\"1.6.2\" -DSTDC_HEADERS=1 -DTIME_WITH_SYS_TIME=1 -DHAVE_TM_ZONE=1 -DHAVE_FCNTL_H=1 -DHAVE_LIMITS_H=1 -DHAVE_UNISTD_H=1 -DHAVE_STRFTIME=1 -DHAVE_VPRINTF=1 -DHAVE_MKDIR=1 -DHAVE_MKTIME=1 -DHAVE_PUTENV=1 -DHAVE_STRPTIME=1 -DHAVE_LOCALTIME_R=1 -I. -I. -g -O2 -c topt1.c
rm -f libutil.a
ar cru libutil.a getopt.o getopt1.o
ranlib libutil.a
make[1]: 离开目录"/usr/local/cronolog-1.6.2/lib"
Making all in src
make[1]: 进入目录"/usr/local/cronolog-1.6.2/src"
gcc -DPACKAGE=\"cronolog\" -DVERSION=\"1.6.2\" -DSTDC_HEADERS=1 -DTIME_WITH_SYS_TIME=1 -DHAVE_TM_ZONE=1 -DHAVE_FCNTL_H=1 -DHAVE_LIMITS_H=1 -DHAVE_UNISTD_H=1 -DHAVE_STRFTIME=1 -DHAVE_VPRINTF=1 -DHAVE_MKDIR=1 -DHAVE_MKTIME=1 -DHAVE_PUTENV=1 -DHAVE_STRPTIME=1 -DHAVE_LOCALTIME_R=1 -I. -I. -I../lib -g -O2 -c cronotest.c
gcc -DPACKAGE=\"cronolog\" -DVERSION=\"1.6.2\" -DSTDC_HEADERS=1 -DTIME_WITH_SYS_TIME=1 -DHAVE_TM_ZONE=1 -DHAVE_FCNTL_H=1 -DHAVE_LIMITS_H=1 -DHAVE_UNISTD_H=1 -DHAVE_STRFTIME=1 -DHAVE_VPRINTF=1 -DHAVE_MKDIR=1 -DHAVE_MKTIME=1 -DHAVE_PUTENV=1 -DHAVE_STRPTIME=1 -DHAVE_LOCALTIME_R=1 -I. -I. -I../lib -g -O2 -c cronoutils.c
cronoutils.c: 在函数'parse_time'中:
cronoutils.c:673:22: 警告: 将一个整数转换为大小不同的指针 [-Wint-to-pointer-cast]
    if (check_end((const char *)strptime(time_str, "date_formats, %tm)))
                        ^
gcc -g -O2 -o cronotest cronotest.o cronoutils.o ../lib/libutil.a
gcc -DPACKAGE=\"cronolog\" -DVERSION=\"1.6.2\" -DSTDC_HEADERS=1 -DTIME_WITH_SYS_TIME=1 -DHAVE_TM_ZONE=1 -DHAVE_FCNTL_H=1 -DHAVE_LIMITS_H=1 -DHAVE_UNISTD_H=1 -DHAVE_STRFTIME=1 -DHAVE_VPRINTF=1 -DHAVE_MKDIR=1 -DHAVE_MKTIME=1 -DHAVE_PUTENV=1 -DHAVE_STRPTIME=1 -DHAVE_LOCALTIME_R=1 -I. -I. -I../lib -g -O2 -c cronolog.c
gcc -g -O2 -o cronolog cronolog.o cronoutils.o ../lib/libutil.a
make[1]: 离开目录"/usr/local/cronolog-1.6.2/src"
Making all in doc
make[1]: 进入目录"/usr/local/cronolog-1.6.2/doc"
```

成功

再执行make install


```
[root@localhost cronolog-1.6.2]# make install
Making install in lib
make[1]: 进入目录"/usr/local/cronolog-1.6.2/lib"
make[2]: 进入目录"/usr/local/cronolog-1.6.2/lib"
make[2]: 对"install-exec-am"无需做任何事。
make[2]: 对"install-data-am"无需做任何事。
make[2]: 离开目录"/usr/local/cronolog-1.6.2/lib"
make[1]: 离开目录"/usr/local/cronolog-1.6.2/lib"
Making install in src
make[1]: 进入目录"/usr/local/cronolog-1.6.2/src"
make[2]: 进入目录"/usr/local/cronolog-1.6.2/src"
/bin/sh ../mkinstalldirs /usr/local/sbin
/usr/bin/install -c cronolog /usr/local/sbin/cronolog
/bin/sh ../mkinstalldirs /usr/local/sbin
/usr/bin/install -c cronosplit /usr/local/sbin/cronosplit
make[2]: 对"install-data-am"无需做任何事。
make[2]: 离开目录"/usr/local/cronolog-1.6.2/src"
make[1]: 离开目录"/usr/local/cronolog-1.6.2/src"
Making install in doc
make[1]: 进入目录"/usr/local/cronolog-1.6.2/doc"
make[2]: 进入目录"/usr/local/cronolog-1.6.2/doc"
make[2]: 对"install-exec-am"无需做任何事。
/bin/sh ../mkinstalldirs /usr/local/info
mkdir /usr/local/info
/usr/bin/install -c -m 644 ./cronolog.info /usr/local/info/cronolog.info
install-info --info-dir=/usr/local/info /usr/local/info/cronolog.info
make install-man1
make[3]: 进入目录"/usr/local/cronolog-1.6.2/doc"
/bin/sh ../mkinstalldirs /usr/local/man/man1
mkdir /usr/local/man
```

4. 安装数据库

先卸载系统自带的mariadb-lib

命令: rpm -qa|grep mariadb

```
[root@localhost /]# rpm -qa|grep mariadb
mariadb-libs-5.5.52-1.el7.x86_64
```

命令: sudo rpm -e mariadb-libs-5.5.52-1.el7.x86_64 --nodeps

```
[root@localhost /]# sudo rpm -e mariadb-libs-5.5.52-1.el7.x86_64
--nodeps
错误: 未安装软件包 mariadb-libs-5.5.52-1.el7.x86_64
```

上传mysql-8.0.11-1.el7.x86_64.rpm-bundle.tar到服务器，并解压tar包

安装mysql8.0.21版本

其中最简单的安装mysql-server服务，只需要安装如下4个软件包即可，我们可以使用rpm -ivh命令进行安装

```
sudo rpm -ivh mysql-community-common-8.0.21-1.el7.x86_64.rpm
```

```
sudo rpm -ivh mysql-community-libs-8.0.21-1.el7.x86_64.rpm
```

```
sudo rpm -ivh mysql-community-client-8.0.21-1.el7.x86_64.rpm
```

```
sudo rpm -ivh mysql-community-server-8.0.21-1.el7.x86_64.rpm
```

如果检测到依赖失败

```
sudo rpm -ivh mysql-community-libs-8.0.21-1.el7.x86_64.rpm --force --nodeps
```

sudo rpm -ivh mysql-community-server-8.0.21-1.el7.x86_64.rpm --force --nodeps

```
1 sudo rpm -ivh mysql-community-common-8.0.11-1.el7.x86_64.rpm
2 sudo rpm -ivh mysql-community-libs-8.0.11-1.el7.x86_64.rpm -- (依赖于common)
3 sudo rpm -ivh mysql-community-client-8.0.11-1.el7.x86_64.rpm -- (依赖于libs)
4 sudo rpm -ivh mysql-community-server-8.0.11-1.el7.x86_64.rpm -- (依赖于client、common)
```

复制

注意：这一步可能遇到报错

```
1 [bigdata@eric011 mysql]$ sudo rpm -ivh mysql-community-server-8.0.11-1.el7.x86_64.rpm
2 警告: mysql-community-server-8.0.11-1.el7.x86_64.rpm: 头V3 DSA/SHA1 Signature, 密钥 ID 5072e1f5: NOKEY
3 错误: 依赖检测失败:
4     libaio.so.1()(64bit) 被 mysql-community-server-8.0.11-1.el7.x86_64 需要
5     libaio.so.1(LIBAIO_0.1)(64bit) 被 mysql-community-server-8.0.11-1.el7.x86_64 需要
6     libaio.so.1(LIBAIO_0.4)(64bit) 被 mysql-community-server-8.0.11-1.el7.x86_64 需要
```

解决方法: yum install libaio

解决方法: yum install libaio

(批量安装gcc包命令: rpm -Uvh --force --nodeps *rpm或rpm -Uvh *.rpm --nodeps --force)

初始化mysql数据库

命令: sudo mysqld --initialize //该命令会在/var/log/mysqld.log生成随机密码

使用 (cat /var/log/mysqld.log查看随机密码)

(#启动成功后可以查看初始化密码随机生成的cat /var/log/mysqld.log | grep password)

修改mysql数据库目录的所属用户及其所属组, 然后启动mysql数据库

命令: sudo chown mysql:mysql /var/lib/mysql -R

命令: systemctl start mysqld.service

根据之前查看的随机密码, 更改root用户的密码, 新版的mysql必须先修改root用户的密码, 否则登录后是不能执行任何命令的

命令: mysql_secure_installation //根据提示修改root用户密码

安装完成, 接下来既可以正常操作mysql

进入: mysql -u root -p (输入密码)

```
[root@localhost ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.21 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

选择表: use mysql;

```
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
```

显示表: show databases;

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.01 sec)
```

退出数据库: quit

```
mysql> quit
Bye
[root@localhost ~]#
```

再次进入数据库, 在 mysql 数据库的 user 表中查看当前 root 用户的相关信息

命令: select host, user, authentication_string, plugin from user;

```
[root@localhost ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.21 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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
mysql> select host, user, authentication_string, plugin from user;
+-----+-----+-----+-----+
| host | user | authentication_string | plugin |
+-----+-----+-----+-----+
| localhost | mysql.infoschema | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | caching_sha2_password |
| localhost | mysql.session | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | caching_sha2_password |
| localhost | mysql.sys | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | caching_sha2_password |
| localhost | root | $A$005$d02!0%$ED=\0m1znY0F0aKocQjRoWwQa54uGHV97Y3ZU9v0fv2L/42H58 | caching_sha2_password |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

授权 root 用户的所有权限并设置远程访问

update user set host='%' where user='root';

% 表示通配所有 host，可以访问远程。

刷新权限

所有操作后，应执行

flush privileges;

再次查看在 mysql 数据库的 user 表中查看当前 root 用户的相关信息

命令：select host, user, authentication_string, plugin from user;

```
mysql> select host, user, authentication_string, plugin from user;
```

host	user	authentication_string	plugin
%	root	\$A\$005\$oD2!0%\$ED=\0mlznyoF0aKoc0jRoWwQa54uGHV97Y3ZU9v0fV2L/42H58	caching_sha2_password
localhost	mysql.infoschema	\$A\$005\$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED	caching_sha2_password
localhost	mysql.session	\$A\$005\$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED	caching_sha2_password
localhost	mysql.sys	\$A\$005\$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED	caching_sha2_password

4 rows in set (0.00 sec)

利用Navicat远程访问数据库

安装Navicat并破解

CenOS-7 - 编辑连接

常规 高级 数据库 SSL SSH HTTP

连接名: CenOS-7 1 自己取名

主机名或 IP 地址: 192.168.253.1 2

端口: 3306 3

用户名: root 4

密码: 5

☒ 保存密码

连接测试 6 确定 取消

输入访问的 host 和密码，报 2059 错误，这是因为 MySQL8.0 版本 和 5.0 的加密规则不一样，而现在的可视化工具只支持旧的加密方式。

此问题有两种方法，一种是更新 Navicat 驱动来解决此问题，另一种是将 MySQL 用户登录的加密规则修改为 mysql_native_password，第一种方法我试过了没有起作用，我这里采用第二种方法。

修改加密规则

```
ALTER USER 'root'@'%' IDENTIFIED BY 'cloudinte' PASSWORD EXPIRE NEVER;
```

cloudinte 为你当前密码。

报错1819

```
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
```

解决办法：使用SHOW VARIABLES LIKE 'validate_password%';查看当前密码安全策略要求。

运行set global validate_password.policy=0;命令，设置密码强度检查等级，其中对应关系为：0/LOW、1/MEDIUM、2/STRONG。

更新 root 用户密码

```
ALTER USER 'root'@'%' IDENTIFIED WITH mysql_native_password BY 'cloudinte';
```

cloudinte 为你新设置的密码。

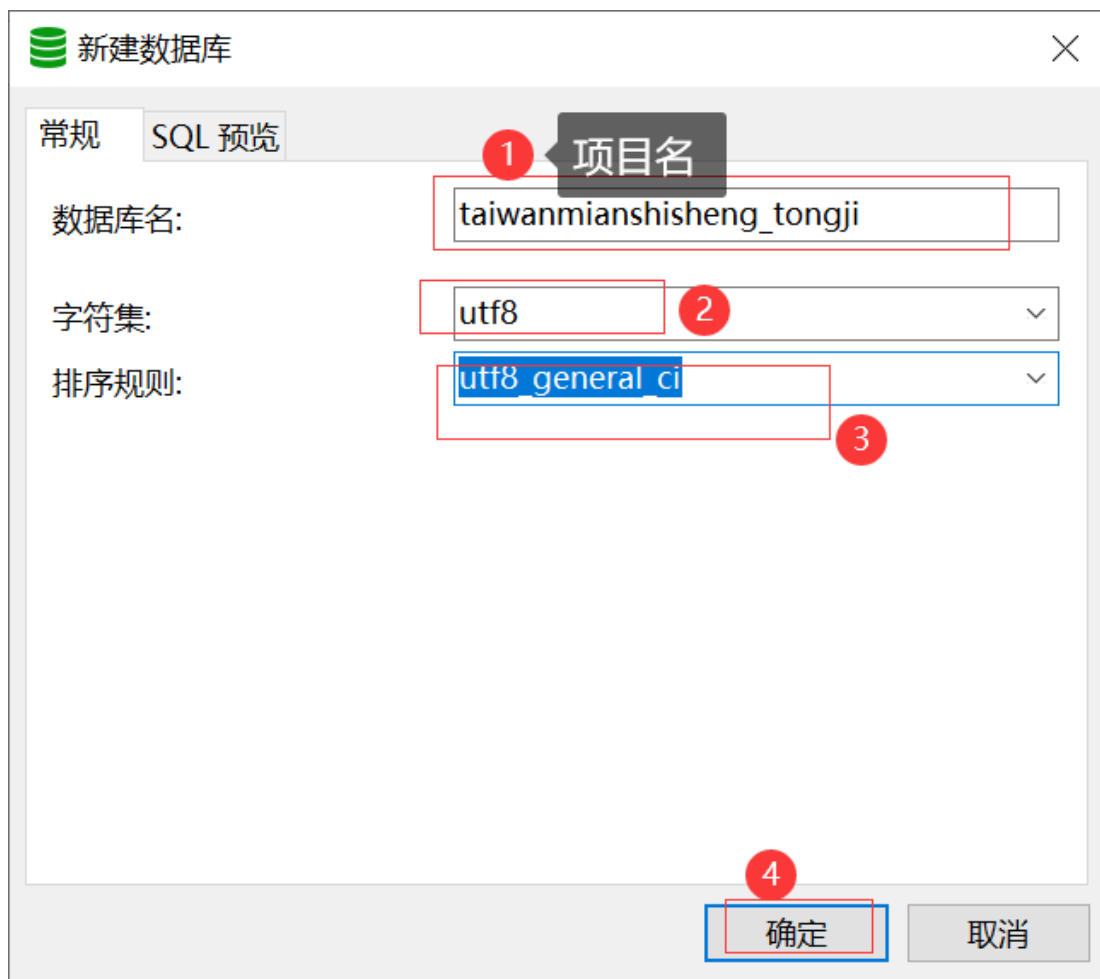
刷新权限

```
FLUSH PRIVILEGES;
```

OK，设置完成，再次使用 Navicat 连接数据库，oh yeah~

完美解决！

新建项目名数据库



运行项目文件夹下的.sql文件

将项目打包上传服务器并解压

使用新建的项目数据库执行项目文件夹下的.sql文件

```
[root@localhost local]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 38
Server version: 8.0.21 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use taiwanmianshisheng_tongji
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
mysql> source /usr/local/taiwanmianshisheng_tongji/taiwanmianshisheng_tongji.sql
```

5.tomcat与mysql连接

连接配置：

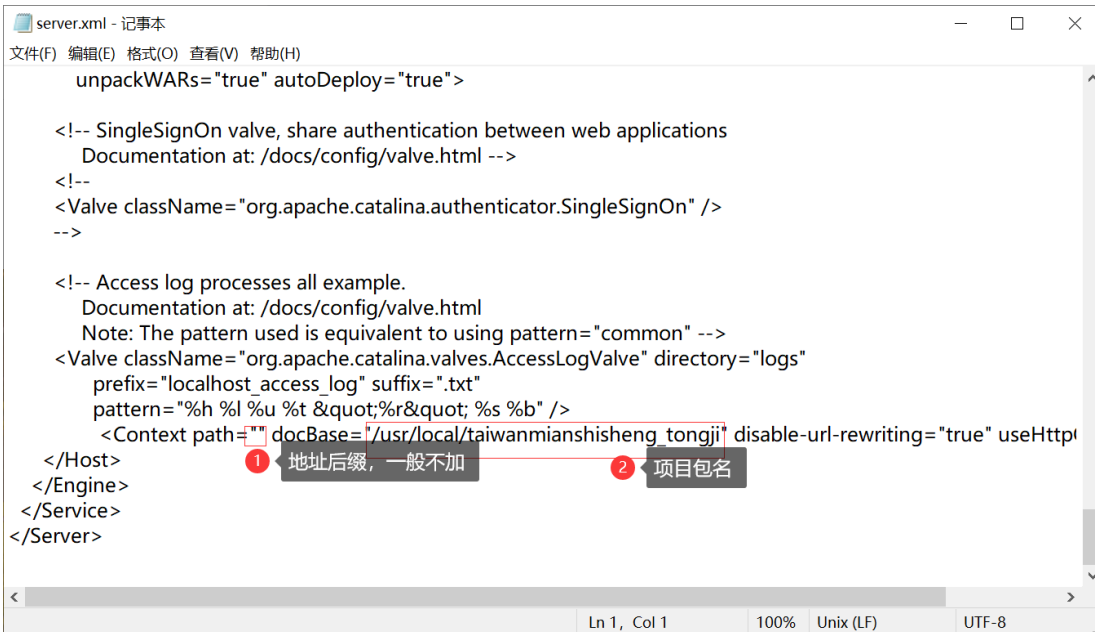
打开并修改 “/项目文件夹/WEB-INF/classes/jeesite.properties”

```
#mysql database setting
jdbc.type=mysql
#jdbc.driver=com.mysql.jdbc.Driver
jdbc.driver=com.mysql.cj.jdbc.Driver

jdbc.url=jdbc:mysql://127.0.0.1:3306/taiwanmianshisheng_tongji?useUnicode=true&characterEncoding=utf-8&allowMultiQueries=true
jdbc.username=root
jdbc.password=000000
```

数据库地址 数据库路径
数据库用户名 数据库密码

打开并修改 "/tomcat/conf/server.xml



进入/tomcat/bin运行./startup.sh

```
[root@localhost bin]# ./startup.sh
Using CATALINA_BASE:   /usr/local/tomcat
Using CATALINA_HOME:   /usr/local/tomcat
Using CATALINA_TMPDIR: /usr/local/tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:        /usr/local/tomcat/bin/bootstrap.jar:/usr/local/tomcat/bin/tomcat-juli.jar
Tomcat started.
```

检查tomcat是否启动: ps -ef |grep tomcat

```
[root@localhost conf]# ps -ef |grep tomcat
root      4701      1  0 14:32 pts/0    00:00:00 /bin/sh /usr/local/tomcat/bin/catalina.sh start
root      4702      1  0 14:32 pts/0    00:00:00 /usr/local/sbin/cronolog /usr/local/tomcat/logs/catalina.%Y-%m-%d.out
root      4703    4701 10 14:32 pts/0    00:02:12 /usr/java/jdk1.8.0_181-amd64/jre/bin/java -Djava.util.logging.config.file=/usr/local/tomcat/conf/logg
ng.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom -serve
r:/usr/local/tomcat/bin/tomcat-juli.jar -Dcatalina.base=/usr/local/tomcat -Dcatalina.home=/usr/local/tomcat -Djava.io.tmpdir=/usr/local/tomcat/temp org
apache.catalina.startup.Bootstrap start
root      5027    3640  0 14:53 pts/0    00:00:00 grep --color=auto tomcat
```

查看日志

tail -500f ../logs/catalina.2020-12-09.out

在浏览器输入应用服务器IP地址即可查看部署的网站