

Linux 的软件安装

一、JDK 安装

注意：rpm 与软件相关命令相当于 window 下的软件助手管理软件

步骤：

- (1) 查看当前 Linux 系统是否已经安装 java

输入 rpm -qa | grep java

```
java-1.6.0-openjdk-1.6.0.35-1.13.7.1.el6_6.i686  
tzdata-java-2015e-1.el6.noarch  
java-1.7.0-openjdk-1.7.0.79-2.5.5.4.el6.i686
```

- (2) 卸载两个 openJDK

输入 rpm -e --nodeps 要卸载的软件

```
[root@itheima23 ~]# rpm -e --nodeps java-1.7.0-openjdk-1.7.0.79-2.5.5.4.el6.i686  
[root@itheima23 ~]# rpm -e --nodeps java-1.6.0-openjdk-1.6.0.35-1.13.7.1.el6_6.i686
```

- (3) 用 WinSCP 上传 JDK 到 Linux

- (4) 解压 JDK 到 /usr/local 下 tar -xvf jdk-8u144-linux-x64.tar.gz -C /usr/local

- (5) 配置 JDK 环境变量，打开 /etc/profile 配置文件，将下面配置拷贝进去

```
#set java environment  
JAVA_HOME=/usr/local/jdk1.8.0_144  
CLASSPATH=.:$JAVA_HOME/lib/tools.jar  
PATH=$JAVA_HOME/bin:$PATH  
export JAVA_HOME CLASSPATH PATH
```

- (6) 重新加载 /etc/profile 配置文件 source /etc/profile

二、MySQL 安装

Linux 下彻底卸载 MySQL

1、停止 MySQL 服务

systemctl stop mysqld

2、查看当前安装 MySQL 情况

rpm -qa | grep -i mysql

```
[root@master ~]# rpm -qa|grep -i mysql
mysql-community-common-5.7.19-1.el7.x86_64
mysql-community-client-5.7.19-1.el7.x86_64
mysql57-community-release-el7-11.noarch
mysql-community-libs-5.7.19-1.el7.x86_64
mysql-community-server-5.7.19-1.el7.x86_64
```

3、按顺序卸载以下内容

```
rpm -ev mysql-community-server-5.7.19-1.el7.x86_64
rpm -ev mysql-community-client-5.7.19-1.el7.x86_64
rpm -ev mysql-community-libs-5.7.19-1.el7.x86_64
rpm -ev mysql57-community-release-el7-11.noarch
rpm -ev mysql-community-common-5.7.19-1.el7.x86_64
```

4、再次查看

```
rpm -qa|grep -i mysql
```

如果没有，则进行下一步。

5、查找之前老版本 mysql 的目录、并且删除老版本 mysql 的文件和库

```
find / -name mysql
```

```
[root@master ~]# find / -name mysql
/var/lib/mysql
/var/lib/mysql/mysql
/usr/share/mysql
```

6、删除对应的目录

```
rm -rf /var/lib/mysql
rm -rf /var/lib/mysql/mysql
rm -rf /usr/share/mysql
```

注意：卸载后/etc/my.cnf 不会删除，需要进行手工删除

```
rm -rf /etc/my.cnf
```

7、再次查找机器是否安装 mysql

```
rpm -qa|grep -i mysql
```

如果没有，则说明卸载彻底了，即可重新安装 MySQL。

8、安装 MySQL

Linux 下安装 MySQL(参考：<http://www.centoscn.com/mysql/2016/0626/7537.html>)

安装环境：CentOS7 64 位 MINI 版，安装 MySQL5.7

9、配置 YUM 源

在 [MySQL](http://dev.mysql.com/downloads/repo/yum/) 官网中下载 YUM 源 rpm 安装包：<http://dev.mysql.com/downloads/repo/yum/>

Red Hat Enterprise Linux 7 / Oracle Linux 7 (Architecture Independent), RPM Package (mysql57-community-release-el7-8.noarch.rpm)	8.9K	Download
Red Hat Enterprise Linux 6 / Oracle Linux 6 (Architecture Independent), RPM Package (mysql57-community-release-el6-8.noarch.rpm)	8.9K	Download
Red Hat Enterprise Linux 5 / Oracle Linux 5 (Architecture Independent), RPM Package (mysql57-community-release-el5-7.noarch.rpm)	8.2K	Download
Fedora 24 (Architecture Independent), RPM Package (mysql57-community-release-fc24-8.noarch.rpm)	13.0K	Download
Fedora 23 (Architecture Independent), RPM Package (mysql57-community-release-fc23-8.noarch.rpm)	12.9K	Download
Fedora 22 (Architecture Independent), RPM Package (mysql57-community-release-fc22-8.noarch.rpm)	12.9K	Download

10、安装 yum 源

```
yum localinstall mysql57-community-release-el7-8.noarch.rpm
```

11、检查 MySQL 是否安装成功

```
yum repolist enabled | grep "mysql.*-community.*"
```

```
[root@master opt]# yum repolist enabled | grep "mysql.*-community.*"
!mysql-connectors-community/x86_64      MySQL Connectors Community
36
!mysql-tools-community/x86_64          MySQL Tools Community
47
!mysql57-community/x86_64             MySQL 5.7 Community Server
207
```

看到上图所示表示安装成功

12、安装

```
yum install mysql-community-server
```

```
已安装:
mysql-community-server.x86_64 0:5.7.19-1.el7

作为依赖被安装:
mysql-community-client.x86_64 0:5.7.19-1.el7
mysql-community-common.x86_64 0:5.7.19-1.el7
mysql-community-libs.x86_64 0:5.7.19-1.el7

完毕！
```

13、启动 MySQL 服务

```
systemctl start mysqld
```

查看 MySQL 的启动状态

```
systemctl status mysqld
```

```
[root@master opt]# systemctl status mysqld
● mysqld.service - MySQL Server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; vendor preset: disabled)
   Active: active (running) since 四 2017-07-27 09:14:29 CST; 11min ago
     Docs: man:mysqld(8)
           http://dev.mysql.com/doc/refman/en/using-systemd.html
   Process: 3708 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid $MYSQLD_OPTS (code=exited, status=0/SUCCESS)
   Process: 3635 ExecStartPre=/usr/bin/mysqld_pre_systemd (code=exited, status=0/SUCCESS)
   Main PID: 3711 (mysqld)
     CGroup: /system.slice/mysqld.service
             └─3711 /usr/sbin/mysqld --daemonize --pid-file=/var/run/...

7月 27 09:14:22 master systemd[1]: Starting MySQL Server...
7月 27 09:14:29 master systemd[1]: Started MySQL Server.
```

14、修改 root 默认密码

mysql 安装完成之后，在/var/log/mysqld.log 文件中给 root 生成了一个默认密码。通过下面的方式找到 root 默认密码，然后登录 mysql 进行修改：

```
grep 'temporary password' /var/log/mysqld.log
```

```
2017-07-27T01:14:24.555799Z 1 [Note] A temporary password is generated for root@localhost: thpNYe=ln8FI
```

```
shell> mysql -uroot -p
```

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'MySQL57pw!';
```

注意：mysql5.7默认安装了密码安全检查插件（validate_password），默认密码检查策略要求密码必须包含：大小写字母、数字和特殊符号，并且长度不能少于8位。否则会提示ERROR 1819 (HY000): Your password does not satisfy the current policy requirements错误，如下图所示：

```
mysql> set password for 'yangxin'@'%'=password('123456abc!');
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
mysql>
```

通过 mysql 环境变量可以查看密码策略的相关信息：

```
mysql> show variables like '%password%';
```

```
mysql> show variables like '%password%'
-> ;
+-----+-----+
| Variable_name | Value |
+-----+-----+
| default_password_lifetime | 0 |
| disconnect_on_expired_password | ON |
| log_built_in_as_identified_by_password | OFF |
| mysql_native_password_proxy_users | OFF |
| old_passwords | 0 |
| report_password |  |
| sha256_password_proxy_users | OFF |
| validate_password_check_user_name | OFF |
| validate_password_dictionary_file |  |
| validate_password_length | 8 |
| validate_password_mixed_case_count | 1 |
| validate_password_number_count | 1 |
| validate_password_policy | MEDIUM |
| validate_password_special_char_count | 1 |
+-----+-----+
14 rows in set (0.00 sec)
```

validate_password_policy：密码策略，默认为MEDIUM策略 validate_password_dictionary_file：密码策略文件，策略为STRONG才需要 validate_password_length：密码最少长度
validate_password_mixed_case_count：大小写字符长度，至少1个 validate_password_number_count：数字至少1个 validate_password_special_char_count：特殊字符至少1个 上述参数是默认策略MEDIUM的密码检查规则。

15、修改密码策略

在/etc/my.cnf 文件添加 validate_password_policy 配置，指定密码策略

修改密码策略

在/etc/my.cnf文件添加validate_password_policy配置，指定密码策略

```
1 # 选择0 (LOW), 1 (MEDIUM), 2 (STRONG) 其中一种, 选择2需要提供密码字典文件
2 validate_password_policy=0
```

如果不需要密码策略，添加my.cnf文件中添加如下配置禁用即可：

```
1 validate_password = off
```

重新启动mysql服务使配置生效：

```
1 systemctl restart mysqld
```

重新启动 mysql 服务使配置生效：

systemctl restart mysqld

登录 mysql，更改密码为 root

ALTER USER 'root'@'localhost' IDENTIFIED BY 'root';

16、设置开机启动

shell> systemctl enable mysqld

shell> systemctl daemon-reload

1) 开启 mysql 的远程登录

默认情况下 mysql 为安全起见，不支持远程登录 mysql，所以需要设置开启远程登录 mysql 的权限

登录 mysql 后输入如下命令：

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

flush privileges;

2) 开放 Linux 的对外访问的端口 3306

iptables -I INPUT -p tcp --dport 3306 -j ACCEPT

service iptables save

```
[root@centos ~]# iptables -I INPUT -p tcp --dport 3306 -j ACCEPT
[root@centos ~]# service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ 确定 ]
[root@centos ~]#
```

如果上面这条命令执行失败，那么需要先安装 iptables 服务，再执行上面的命令

yum install iptables-services

三、Tomcat 的安装

步骤：

1) 上传 Tomcat 到 linux 上

2) 解压 Tomcat 到/usr/local 下

```
tar -xvf apache-tomcat-8.0.52.tar.gz -C /usr/local
```

```
cd /usr/local
```

```
mv apache-tomcat-8.0.52 tomcat8.0
```

3) 开放 Linux 的对外访问的端口 8080

```
iptables -I INPUT -p tcp --dport 8080 -j ACCEPT
```

```
service iptables save
```

```
[root@centos ~]# service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ 确定 ]
[root@centos ~]#
```

如果上面这条命令执行失败，那么需要先安装 iptables 服务，再执行上面的命令
yum install iptables-services

4) 启动关闭 Tomcat

进入 tomcat 的 bin 下启动：./startup.sh

进入 tomcat 的 bin 下关闭：./shutdown.sh

四、Redis 安装

1、Redis 的安装

1) 安装 redis 编译的 c 环境

```
yum install gcc-c++
```

2) 将 redis-3.0.0.tar.gz 用 WinSCP 上传到 Linux 系统中

3) 解压到/usr/local 下 tar -xvf redis-3.0.0.tar.gz -C /usr/local

4) 进入 redis-3.0.0 目录 使用 make 命令编译 redis

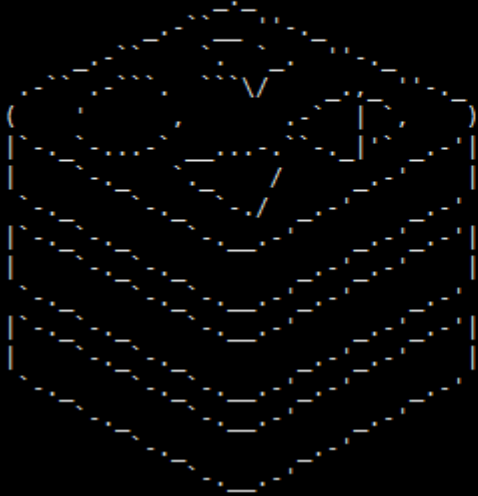
```
[root@centos ~]# cd /usr/local/redis-3.0.0
[root@centos redis-3.0.0]# make
```

5) 在 redis-3.0.0 目录中 使用 make PREFIX=/usr/local/redis install 命令安装 redis 到/usr/local/redis 中

```
[root@centos redis-3.0.0]# make PREFIX=/usr/local/redis install
```

6) 在/redis/bin 目录下使用 ./redis-server 启动 redis (前端启动模式)

```
[root@centos bin]# ./redis-server
5605:C 21 Jun 10:34:42.730 # Warning: no config file specified, using the default config
. In order to specify a config file use ./redis-server /path/to/redis.conf
5605:M 21 Jun 10:34:42.734 * Increased maximum number of open files to 10032 (it was ori
ginally set to 1024).
```



```
Redis 3.0.0 (00000000/0) 64 bit

Running in standalone mode
Port: 6379
PID: 5605

http://redis.io
```

7) 拷贝 redis-3.0.0 中的 redis.conf 到安装目录 redis 中

cp redis.conf /usr/local/redis/bin

```
[root@centos redis-3.0.0]# cp redis.conf /usr/local/redis/bin
[root@centos redis-3.0.0]# cd ../
[root@centos local]# cd redis
[root@centos redis]# ls
bin
[root@centos redis]# cd bin
[root@centos bin]# ls
dump.rdb      redis-check-aof  redis-cli      redis-sentinel
redis-benchmark  redis-check-dump  redis.conf     redis-server
[root@centos bin]#
```

8) 修改 redis.conf 配置文件, daemonize yes 以后端模式启动

```
[root@centos bin]# vi redis.conf
[root@centos bin]#
```

9) 在/usr/local/redis/bin 目录下后端模式启动 redis

./redis-server redis.conf

```
[root@centos bin]# ./redis-server redis.conf
[root@centos bin]#
```

10) 启动客户端

./redis-cli

```
[root@centos bin]# ./redis-cli
127.0.0.1:6379>
```

11) 退出客户端

exit

```
127.0.0.1:6379> exit
[root@centos bin]#
```

12) Redis 默认端口 6379, 通过当前服务进行查看


```
ps -ef | grep redis
```

```
[root@centos bin]# ps -ef | grep redis
root      5735      1  0 11:02 ?        00:00:01 ./redis-server *:6379
root      5766    2289  0 11:18 pts/0    00:00:00 grep --color=auto redis
```

- 13) 强制结束程序。强行终止 Redis 进程可能会导致 Redis 持久化数据丢失
kill -9 5735

```
[root@centos bin]# ps -ef | grep redis
root      5735      1  0 11:02 ?        00:00:01 ./redis-server *:6379
root      5766    2289  0 11:18 pts/0    00:00:00 grep --color=auto redis
[root@centos bin]# kill -9 5166
-bash: kill: (5166) - 没有那个进程
[root@centos bin]# kill -9 5735
[root@centos bin]#
```

- 14) 正确停止 redis 的方式应该是向 Redis 发送 shutdown 命令, 方法为 : (关闭默认端口)
./redis-cli shutdown

```
[root@centos bin]# ./redis-cli shutdown
[root@centos bin]# ./redis-cli
Could not connect to Redis at 127.0.0.1:6379: Connection refused
not connected>
```

- 15) 如需远程连接 redis, 需配置 redis 端口 6379 在 linux 防火墙中开放
iptables -I INPUT -p tcp --dport 6379 -j ACCEPT
service iptables save

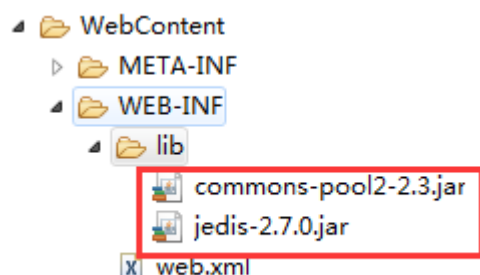
```
[root@centos ~]# iptables -I INPUT -p tcp --dport 6379 -j ACCEPT
[root@centos ~]# service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ 确定 ]
[root@centos ~]#
```

如果上面这条命令执行失败, 那么需要先安装 iptables 服务, 再执行上面的命令
yum install iptables-services



2、使用 Java 去操作 redis

1) 在 Eclipse 中创建一个 2.5 版本 web 项目 demo，将下面的 jar 包拷贝到 lib 下



2) 配置 redis 远程连接端口 (参考上面)

3) 利用 Junit 测试 Java 代码

```
public class JedisTest {

    //通过java程序访问redis数据库

    @Test
    //获得单一的jedis对象操作数据库
    public void test1(){

        //1、获得连接对象
        Jedis jedis = new Jedis("192.168.133.128", 6379);

        //2、获得数据
        String username = jedis.get("username");
        System.out.println(username);

        //3、存储
        jedis.set("addr", "北京");
        System.out.println(jedis.get("addr"));

    }
}
```