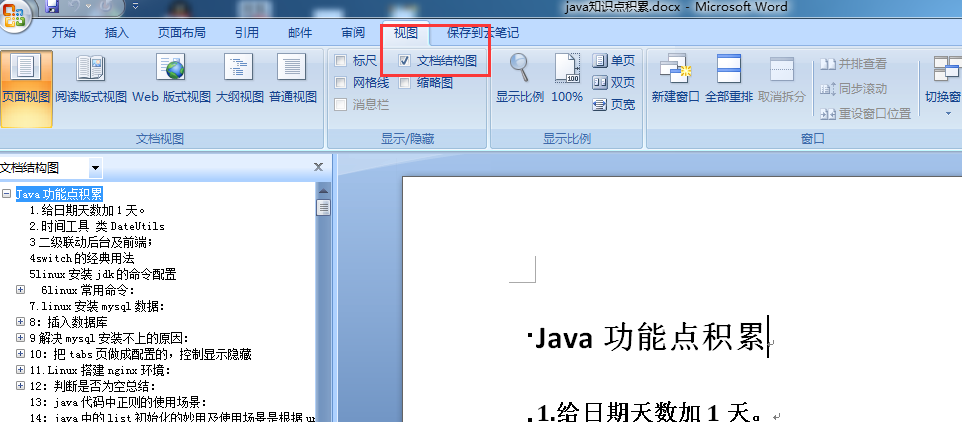
# Java功能点积累



## 1.给日期天数加1天。

private String formatDate(String projectInitTimeEnd){

try {

//截止日期处理

SimpleDateFormat dateFormater = new SimpleDateFormat("yyyy-MM-dd");

Date initEnd;

initEnd = dateFormater.parse(projectInitTimeEnd);

Calendar calendar = new GregorianCalendar();

calendar.setTime(initEnd);

calendar.add(calendar.DATE,1);

projectInitTimeEnd = dateFormater.format(calendar.getTime());

} catch (ParseException e) {

e.printStackTrace();

}

return projectInitTimeEnd; }

## 2.时间工具 类DateUtils

<http://note.youdao.com/noteshare?id=eefca05044eba03b334e46cd58ca2b0f>

## 3二级联动后台及前端；

$('#customerSubCategory').removeClass('fn-hide');//给id移除css类 $('#customerSubCategory').addClass('fn-hide');//给id增加css类

**（1）前端**

**function** twoLevelLink(dom, index, data) {

**var** n = index || 0,

html = '<option value="">请选择</option>';

data = JSON.parse(data);

**if**(n > 0){

n = n - 1;

**var** len = data[n].subConfigs.length;

**if**(len > 0){

**for**(**var** i = 0; i < len; i++){

html += '<option value="' + data[n].subConfigs[i].code + '">' + data[n].subConfigs[i].name + '</option>';

}

}

}

dom.html(html);

}

$('#customerCategory').on('change', **function**(e){

/\*调用行业二级联动\*/

**var** index = $(**this**).find(':selected').index();

console.log("=======", index);

twoLevelLink($('#customerSubCategory'), index, $('#customerSubCategorySelect').val());

});

})

（2）后台：

// 获取应用分类

ListResult<Config> bizCategoryResult = queryCustomerInfoService.findBizCategoryConfig();

**if** (bizCategoryResult != **null** && bizCategoryResult.isSuccess()) {

List<Config> data = bizCategoryResult.getData();

**for** (Config config : data) {

**if**(projectBaseInfo.getCustomerCategory().equals(config.getCode())){

context.put("currentBizSubConfigList", config.getSubConfigs());

}

}

context.put("bizConfigList", data);

context.put("bizSubConfigList", JSON.*toJSON*(data));

}

（3）前端页面：

<td>\*行业</td>

<td class="tdContent">

## <select id="customerCategory" class="length\_2 readonly" disabled name="customerCategory">

## </select>

## <select id="customerSubCategory" class="length\_3 readonly" disabled name="customerSubCategory">

## </select>

<select class="length\_3 readonly" disabled id="customerCategory" name="customerCategory">

<option>请选择</option>

#foreach($stage in $!bizConfigList)

<option value="$stage.code" #if($stage.code == $!projectBaseInfo.customerCategory)selected#end>$stage.name</option>

#end

</select>

<select class="length\_3 readonly" disabled id="customerSubCategory" name="customerSubCategory">

#foreach($stage in $!currentBizSubConfigList)

<option value="$stage.code" #if($stage.code == $!projectBaseInfo.customerSubCategory)selected#end>$stage.name</option>

#end

</select>

<input id="customerSubCategorySelect" type="hidden" value="$bizSubConfigList" />

</td>

## 4switch的经典用法

List<ProjectRelationPerson> cbmList = Lists.*newArrayList*();

StringBuilder cbmEmpIds = **new** StringBuilder();

List<ProjectRelationPerson> caList = Lists.*newArrayList*();

StringBuilder caEmpIds = **new** StringBuilder();

List<ProjectRelationPerson> pmList = Lists.*newArrayList*();

StringBuilder pmEmpIds = **new** StringBuilder();

List<ProjectRelationPerson> daList = Lists.*newArrayList*();

StringBuilder daEmpIds = **new** StringBuilder();

List<ProjectRelationPerson> seList = Lists.*newArrayList*();

StringBuilder seEmpIds = **new** StringBuilder();

**if** (!CollectionUtils.*isEmpty*(projectBaseInfo.getProjectRelationPersons())) {

**for** (ProjectRelationPerson prp : projectBaseInfo.getProjectRelationPersons()) {

**switch** (ProjectRelationPersonTypeEnum.*getEnum*(prp.getJobCategory())) {

**case** ***CBM***:

cbmList.add(prp);

cbmEmpIds.append(prp.getEmployName()).append(",");

**continue**;

**case** ***CA***:

caList.add(prp);

caEmpIds.append(prp.getEmployName()).append(",");

**continue**;

**case** ***PM***:

pmEmpIds.append(prp.getEmployName()).append(",");

pmList.add(prp);

**continue**;

**case** ***DA***:

daEmpIds.append(prp.getEmployName()).append(",");

daList.add(prp);

**continue**;

**case** ***SE***:

seEmpIds.append(prp.getEmployName()).append(",");

seList.add(prp);

**continue**;

**default**:

**break**;

}

}

}

//判断字符串的时候使用StringUtils.isNotBlank();

**if**(StringUtils.*isNotBlank*(cbmEmpIds.toString())){

row2.createCell(9).setCellValue(cbmEmpIds.toString().substring(0, cbmEmpIds.length()-1));//cbm

}

**if**(StringUtils.*isNotBlank*(caEmpIds.toString())){

row2.createCell(10).setCellValue(caEmpIds.toString().substring(0, caEmpIds.length()-1));//CA

}

**if**(StringUtils.*isNotBlank*(pmEmpIds.toString())){

row2.createCell(11).setCellValue(pmEmpIds.toString().substring(0, pmEmpIds.length()-1)); //PM

}

**if**(StringUtils.*isNotBlank*(daEmpIds.toString())){

row2.createCell(12).setCellValue(daEmpIds.toString().substring(0, daEmpIds.length()-1));//DA

}

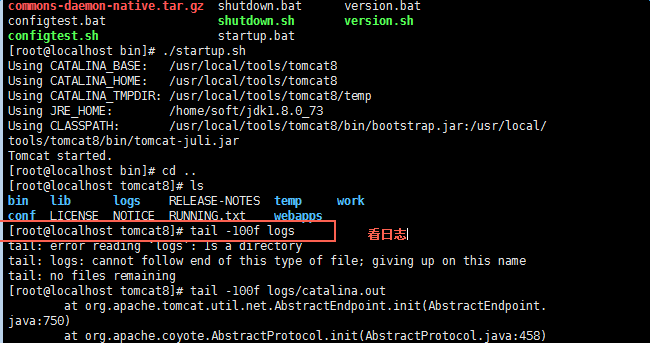
**if**(StringUtils.*isNotBlank*(seEmpIds.toString())){

row2.createCell(33).setCellValue(seEmpIds.toString().substring(0, seEmpIds.length()-1));//服务经理

}

## 5linux安装jdk的命令配置

1. 全局配置  
   vi /etc/profile  
   添加如下内容  
   export JAVA\_HOME=/usr/jdk1.6.0\_45  
   export JAVA\_BIN=/usr/jdk1.6.0\_45/bin  
   export PATH=$PATH:$JAVA\_HOME/bin  
   export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar  
   export JAVA\_HOME JAVA\_BIN PATH CLASSPATH  
   立即生效，执行  
   source /etc/profile  
   2、重启测试  
   java -version  
   屏幕输出:  
   java version "1.6.0\_45"  
   Java(TM) SE Runtime Environment (build 1.6.0\_45-b06)  
   Java HotSpot(TM) 64-Bit Server VM (build 20.45-b01, mixed mode)
2. 看tomcat日志：



查看tomcat进程：ps -ef | grep tomcat

查看端口是否开放



## 6linux常用命令：

#### 6.1.首先需要按ESC键回到命令模式；

vi保存文件有不同的选项，对应于不同的命令，你可以从下面的命令中选择一个需要的输入：

* :wq 保存文件并退出vi
* :w 保存文件但不退出vi
* :w file 将修改另外保存到file中，不退出vi
* :q 不保存文件，退出vi
* :wq! 强制保存并退出

最后回车即可

#### 6.2.切换用户：C:\Users\tangjinhui\Desktop\sdf.png

#### 6.3.退出命令行：按快捷键crtl+d

#### 6.4.解压tar包；

tar jxvf mysql jxvf 解压zip和gz2格式的压缩包；

tar zxvf mysql zxvf 解压tar.gz 格式的压缩包；

#### 6.5.创建目录：mkdir 文件夹名 ；

删除目录：rmdir 文件夹名 ；

删除文件 rm 文件名 ；

强制删除文件 rm -f 文件名；

#### 6.6. 查看由那些目录

垂直展示：ll

水平展示：ls

#### 6.7.查看linux的版本：cat /proc/version

[root@localhost yum.repos.d]# cat CentOS7-Base.repo 查看文件内容

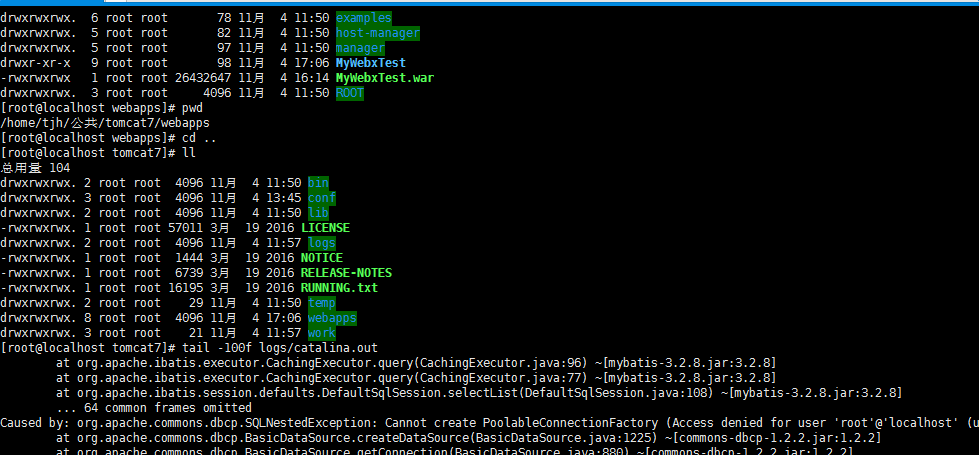
#### 6.8.查看是否安装的：rpm -qa |grep mysql

#### 6.9.查看tomcat日志：

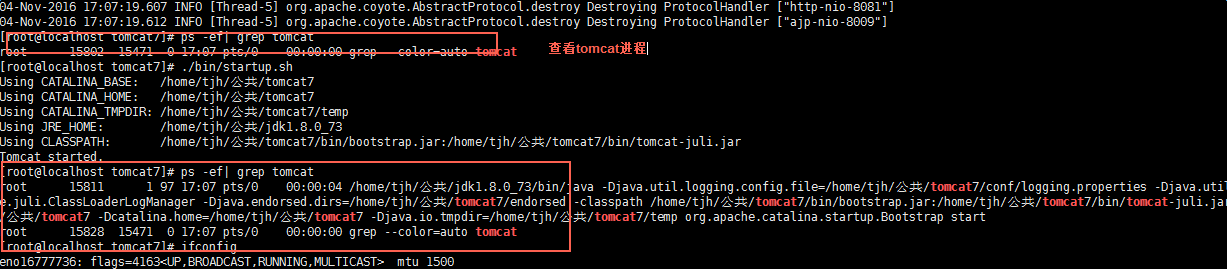
cd /alidata/www/logs/tomcat7/

tail -f catalina.out 查看日志

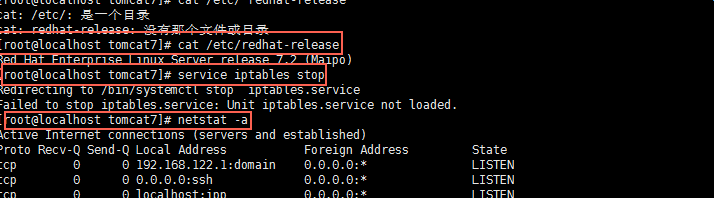
tail -100f catalina.out 查看最近的一百行日志。



#### 6.10查看tomcat进程：



#### 6.11删除war包；

[root@localhost yum.repos.d]# rm -rf redhat.repo删除命令  


#### 6.12.查看开启的ssh服务：

rpm -V openssh-server

查看yum 中的问题

#### 6.13 查询yum源中有没有mysql （i是不区分大小写）

rpm -qa | grep -i mysql

yum源安装命令软件

[root@localhost yum.repos.d]# yum install -y perl-Data-Dumper



#### 6.14[ps -ef |grep 输出的具体含义](http://blog.csdn.net/sinat_24928447/article/details/49869199) [root@localhost ~]# ps -ef | grep ApacheJetspeed root     18887 18828  0 08:09 pts/0    00:00:00 grep ApacheJetspeed 字段含义如下： UID   PID PPID C STIME   TTY   TIME   CMD root 18887 18828   0  08:09     pts/0  00:00:00   grep ApacheJetspeed

ps:将某个进程显示出来

-A 　显示所有程序。

-e 　此参数的效果和指定"A"参数相同。

-f 　显示UID,PPIP,C与STIME栏位。

grep命令是查找

中间的|是管道命令 是指ps命令与grep同时执行

这条命令的意思是显示有关Apachejetspeed有关的进程

UID PID PPID C STIME TTY TIME CMD

各相关信息的意义：

UID 程序被该 UID 所拥有

PID 就是这个程序的 ID

PPID 则是其上级父程序的ID

C CPU 使用的资源百分比

STIME 系统启动时间

TTY 登入者的终端机位置

TIME 使用掉的 CPU 时间。

CMD 所下达的指令为何

对于查询结果，如何判断是运行与否呢？

  这是因为ps -ef是显示所有进程的消息，包括ApacheJetspeed和grep ApacheJetspeed这两个甚至包括ps -ef本身，而grep是查找输出包含想要的字符串的行，也就是说grep ApacheJetspeed是在所有运行的进程中查找输出包含“ApacheJetspeed”字符串的输出行，这里面就包含ApacheJetspeed，和grep ApacheJetspeed 两个进程。

即，如果运行了会显示两条输出一条是ApacheJetspeed的，令一条是grep ApacheJetspeed的。  
如果没运行只会显示grep ApacheJetspeed的。

#### 6.15命令总结：

date

echo $LANG

LANG=en\_US

cal

cal 2016

bc

Tab连续按两次提示

man date

who 用户在线状态

netstat -a 网络连接状态

ps -aux 使用BSD语法查看所有进程

ps -ef 使用标准语法查看所有进程

sync 数据同步写入硬盘

shutdown

reboot

half

poweroff

init 3 纯命令模式

init 5 图形界面模式

chgrp 改变文件所属用户组

chgrp users file.txt

chown 改变文件所有者

chown root:root file.txt

chmod 改变权限

r:4

w:2

x:1

-rwxrwx--- 4+2+1=7

4+2+1=7

0+0+0=0

chmod 770 fileName

uname -a 查看内核版本

ll -h 以kb或者mb或者gb样式呈现文件

cat -n 带行号查看文件

tac 倒叙查看文件

nl 带行号查看文件

head -n 20 file.txt 显示文件前20行

tail 只看结尾几行

tail -100f file.txt 查看文件最后100行并持续加载该文件

more 查看大文件一页一页看

less 前后翻页看

touch创建一个新的空文件

touch -t 160823 file.txt 将文件日期改为2016年08月23日

chattr +i file.txt 设置文件保护属性root也无法删除

chattr -i 去除保护权限

chattr +ai 文件只能增加数据不能删除也不能修改数据

file \*.txt 查看文件的基本属性

which -a ifconfig 查询ifconfig命令在哪儿

whereis nginx 查询nginx文件安装在哪儿，速度比find快因为走了数据库表并未走文件系统

find / -name test.java 在根目录查找文件，走文件系统，速度较慢

find / -type class 根目录查询类型文class的文件

find / -size +1000k 根目录查询文件大于1M的文件

df 调出目前挂载的设备

df -h 以人们较易阅读的格式展示磁盘使用情况

df -h /home 分析home文件夹属性大小

du 评估文件系统使用量，应用于所有文件查看

du -sm /\* 检查根目录个文件夹使用情况

ln -sf file file2 创建file的链接文件file2，似windows得快捷方式

last 查询这个月有哪些用户登陆了主机

fdisk -l 列出所有可用分区

fdisk /dev/sdc 进入c分区界面开始分区

指令有：q w n d p

mkfs -t ext3 /dev/hdc2 把c2磁盘格式化为ext3格式

硬盘大于2TB的分区：

parted

parted /dev/hdc mkpart logical ext3 15GB 17GB #新建一个2G的分区

parted /dev/hdc rm 8 #删除8号分区

命令一旦执行无法挽回，慎用，除非硬盘大于2T否则还是用fdisk吧

fsck -C -f -t ext3 /dev/hdc2 强制检测c2磁盘并修复

badblocks -sv /dev/hdc2 检测c2有没有坏轨

mkdir /media/cdrom

mount -t iso9660 /dev/cdrom /media/cdrom 挂在光驱内容到media的cdrom文件夹内

mkdir /mnt/usb

mount -t vfat -o iocharset=cp950 /dev/hdc2 /mnt/usb 挂在fat格式的U盘到usb目录

mount -l

umount /mnt/usb 卸载挂载点

设置开机挂在：

总共6个参数

vim /etc/fstab

添加到最后一行：

/dev/mapper/rhel-root / xfs defaults 0 0

挂在文件系统到根目录

验证是否正确：mount -a

df -h

如果由于输入有误开机挂载失败：

mount -n -o remount,rw /

镜像ISO挂载：

mount -o loop /home/rhml\_7.2.iso /mnt/redhat7

内存交换空间swap的构建：

fdisk /dev/hdc

n

p

t #修改系统ID

7 #7号分区

82 #改成swap的ID

p

w

最后一定要让内核更新分区表：

partprobe

构建swap格式：

mkswap /dev/hdc7

加载swap分区：

swapon /dev/hdc7

查看目前使用到的swap设备：

swapon -s

tar -cvf file.tar file/ #仅仅打包file文件夹

tar -jcvf file.tar.gz2 file/ #打包并以gzip2格式压缩文件

tar -zvf file.tar #解压普通tar文件

tar -xzvf file.tar.gz #解压gz文件

tar -jzvf file.tar.gz2 #解压gz2文件

vi文本编辑器：

三种模式：一般模式、编辑模式、命令行模式

一般模式命令：

hjkl 上下左右

dd 删除光标所在行

yy 复制光标所在行

p 粘贴内容到下一行

/name 向下查找name在哪儿

?name 向上查找name在哪儿

u 复原前一个造作

Ctrl+r 重做上一个造作

编辑模式：

i 进入插入模式

ESC 退出编辑模式

命令行模式：

:w 保存

:wq 保存并退出

:q 退出不保存

bash shell：

\ 在命令行让enter键失去执行能力

“” 中的变量可以生效

‘’ 中的变量仅仅是字符 $name

`` 也很重要

ulimit 用户文件系统及程序限制

ulimit -a 查看限制

ulimit -f 10240 限制用户仅能创建10M以下的文件

alias 别名设置

alias lm='ls -l | more' 简化较长的命令输入

unalias lm 取消已经设置的别名

history 显示历史命令

!! 执行上一条命令

source 读入环境配置文件

source /etc/profile

命令执行可以用；隔开以便一行执行多个命令

cd /home ; ls -l

还可以这样：cd /home && ls -l 逻辑与关系，前面执行成功了后面再执行

cd/home || cd/root 逻辑或的关系 前面执行成功了后面就不执行了

ls /tmp/abc || mkdir /tmp/abc && touch /tmp/abc/file.txt

ls /tmp/abc && echo "abc exist" || echo "no exist"

三项式：cmd && cmd || cmd

管道命令：cut,grep,sort,wc,uniq,tee,tr,col,join,paste,expand,split,xargs

| 界定符号

cut,grep 选取命令

last | cut -d ' ' -f 1 显示登陆记录并只选取第一列数据

last | grep 'root' | cut -d ' ' -f 1 只显示root登陆的记录并且只显示第一列

ps -ef | grep 'tomcat' 使用标准语法查看系统中的tomcat进程

grep 'MANPATH' /etc/man\_db.conf 文件中筛选查看有MANPATH的所有行

cat /etc/passwd | sort 查看密码文件并排序

cat /etc/passwd | sort -t ':' -k 3 以：分割并以第三列为准排序

last | cut -d ' ' -f 1 | sort | uniq -c 排序显示去重后的登陆用户名和登陆总数量

wc 查看文本行数字数字符数

last | wc 列出登陆记录有多少行，多少字，多少字符

last | grep [a-zA-Z] | grep -v 'twmp' | wc -l 统计登陆次数

tee 双向重定向

last | tee ~/last.txt | cut -d ' ' -f1 统计登陆过的用户并输出到last.txt文件中并在屏幕中显示

last | tr [a-z] [A-Z] 输出信息中的小写替换成大写

last | tr -d ':' 输出信息中的：删除

col 将tab键替换成对等的空格键

join 两个文件中相同数据行合并在一起

join -t ':' /etc/passwd /etc/shadow 用：分割并合并相同的行数据

paste 将两个文件内容粘贴在一起，默认中间以tab分割

paste /etc/passwd /etc/shadow

查看正在监听的端口：

netstat -tunl

列出已连接的网络状态：

netstat -tun

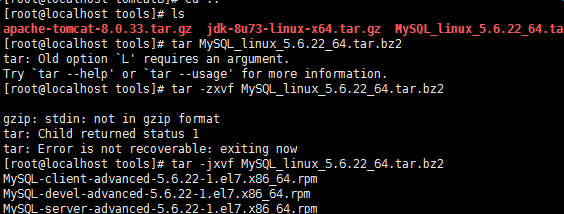
删除已经建立的链接：

netstat -tunp

kill -9 13112

## 7.linux安装mysql数据：

安装mysql数据库：5.6.22为例



卸载MariaDB: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。

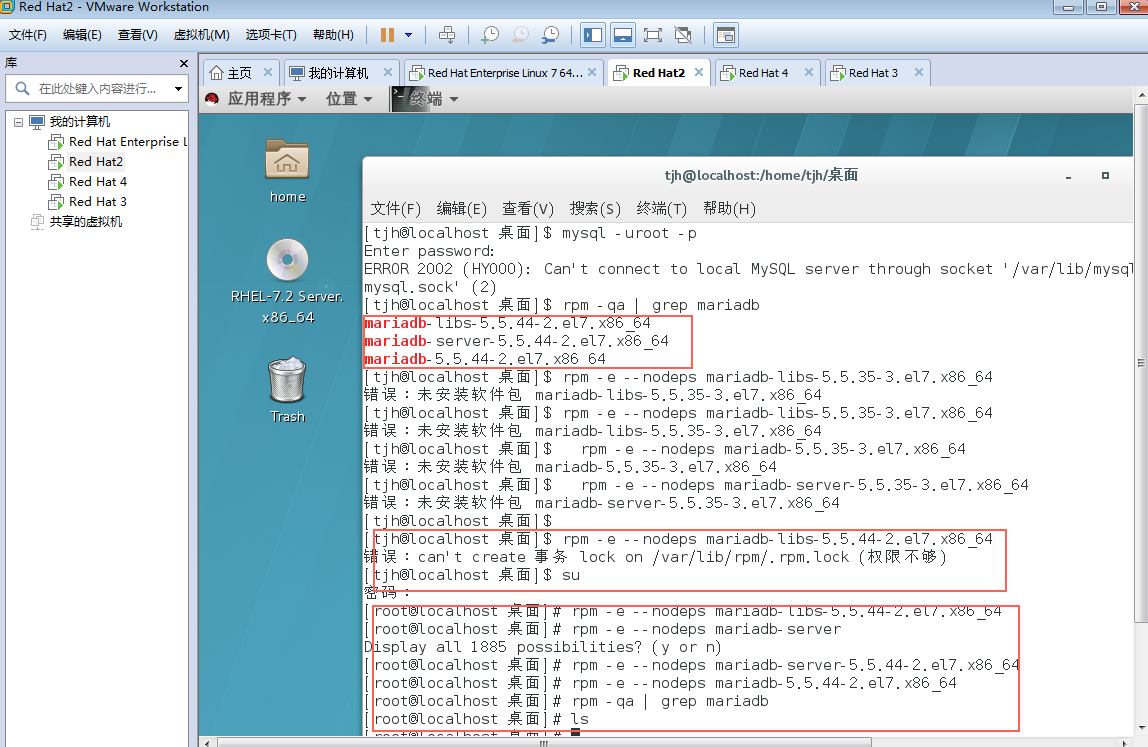
MariaDB的目的是完全兼容MySQL，包括API和命令行，使之能轻松成为MySQL的代替品。

查看当前安装的mariadb包：rpm -qa | grep mariadb

强制卸载：rpm -e --nodeps mariadb-libs-5.5.35-3.el7.x86\_64

rpm -e --nodeps mariadb-5.5.35-3.el7.x86\_64

rpm -e --nodeps mariadb-server-5.5.35-3.el7.x86\_64



安装mysql：

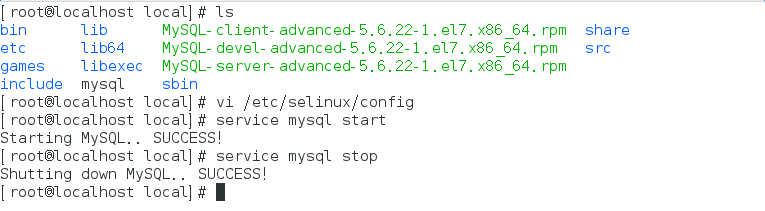
rpm -ivh --nodeps MySQL-client-advanced-5.6.22-1.el7.x86\_64.rpm

rpm -ivh --nodeps MySQL-devel-advanced-5.6.22-1.el7.x86\_64.rpm

rpm -ivh --nodeps MySQL-server-advanced-5.6.22-1.el7.x86\_64.rpm

vi /etc/selinux/config

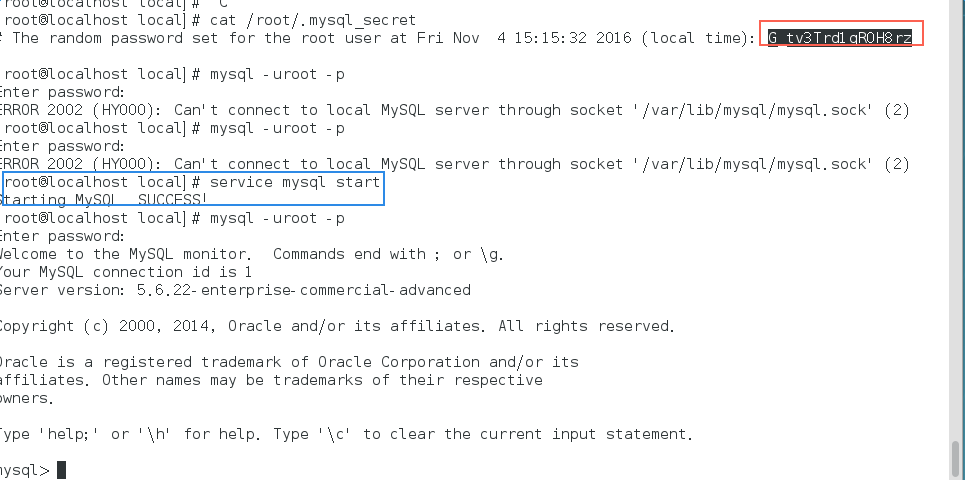
把SELINUX=enforcing改为SELINUX=disabled后存盘退出重启



service mysql stop

service mysql start

安装成功后会有密码提示：可以执行如下命令查看：cat /root/.mysql\_secret



mysql -u root -p

password：\*\*\*

进入后修改密码：

SET PASSWORD = PASSWORD('root');

也可以重置密码：

killall -TERM mysqld

mysqld\_safe –skip-grant-tables &

mysql -u root

use mysql;

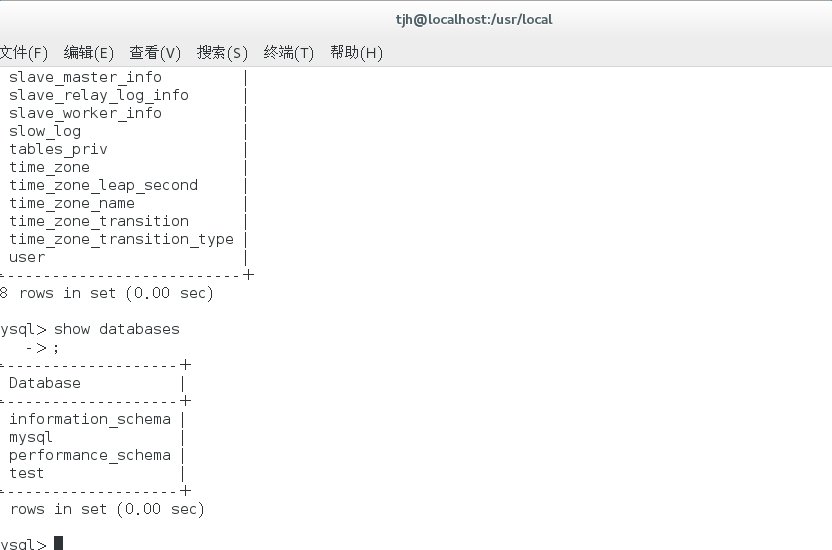
update user set password=password(“新密码”) where user=”root”;

flush privileges;

远程访问mysql配置：

use mysql; //必须执行才能使用下面的命令

## 8：插入数据库



SELECT Host,User FROM user;

UPDATE user SET Host = '%' WHERE User= 'root' LIMIT 1; //“%”表示所有字段

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%'IDENTIFIED BY 'root' WITH GRANT OPTION; //允许所有通过密码访问的用户

插入localhost用户：

INSERT INTO `user` (`Host`, `User`, `Password`, `Select\_priv`, `Insert\_priv`, `Update\_priv`, `Delete\_priv`, `Create\_priv`, `Drop\_priv`, `Reload\_priv`, `Shutdown\_priv`, `Process\_priv`, `File\_priv`, `Grant\_priv`, `References\_priv`, `Index\_priv`, `Alter\_priv`, `Show\_db\_priv`, `Super\_priv`, `Create\_tmp\_table\_priv`, `Lock\_tables\_priv`, `Execute\_priv`, `Repl\_slave\_priv`, `Repl\_client\_priv`, `Create\_view\_priv`, `Show\_view\_priv`, `Create\_routine\_priv`, `Alter\_routine\_priv`, `Create\_user\_priv`, `Event\_priv`, `Trigger\_priv`, `Create\_tablespace\_priv`, `ssl\_type`, `ssl\_cipher`, `x509\_issuer`, `x509\_subject`, `max\_questions`, `max\_updates`, `max\_connections`, `max\_user\_connections`, `plugin`, `authentication\_string`, `password\_expired`) VALUES ('localhost', 'root', '\*98FDF2A96E5C765844B2280D814606072AA2C7F1', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', '', '', '', '', '0', '0', '0', '0', '', '', 'Y');

UPDATE user SET password='\*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B';

UPDATE user SET password\_expired='N'; //设置密码不过期

flush privileges;

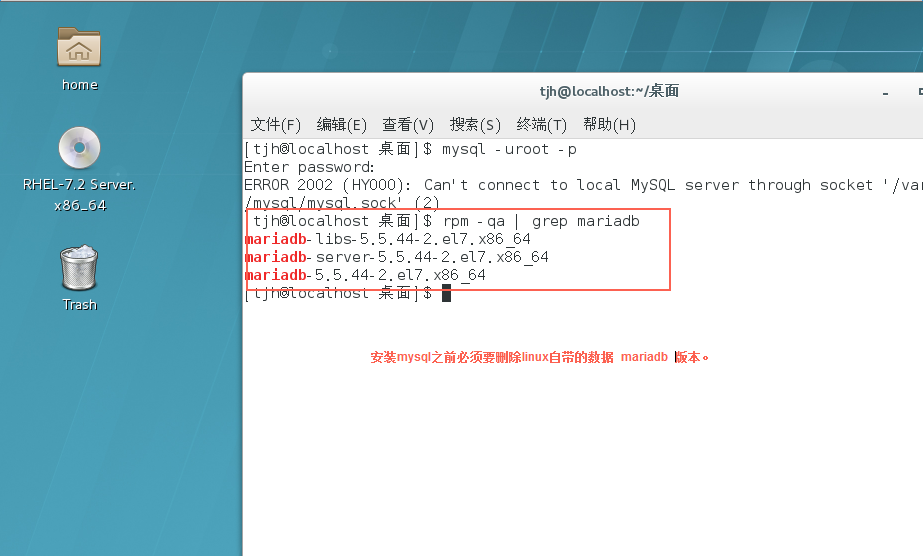
或者在远程主机上设置user表也行;

### 8.2：mysql输入库中导入相关表：

create database my\_table;

charset utf8;

source /opt/project/my\_table.sql



## 9解决mysql安装不上的原因：

\*\*\*安装失败：缺少必要的组件

yum install -y perl

yum install -y perl-Data-Dumper

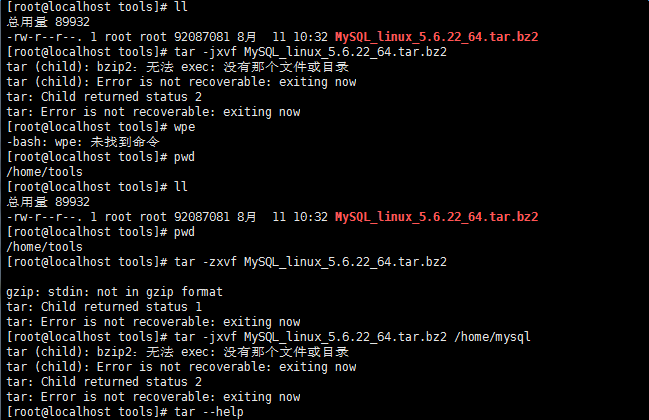
然后再初始化数据库：sudo mysql\_install\_db --user=mysql --basedir=/usr/ --ldata=/var/lib/mysql/

或者重新安装也行：rpm -ivh --nodeps MySQL-server-advanced-5.6.22-1.el7.x86\_64.rpm

启动数据库：service mysql start

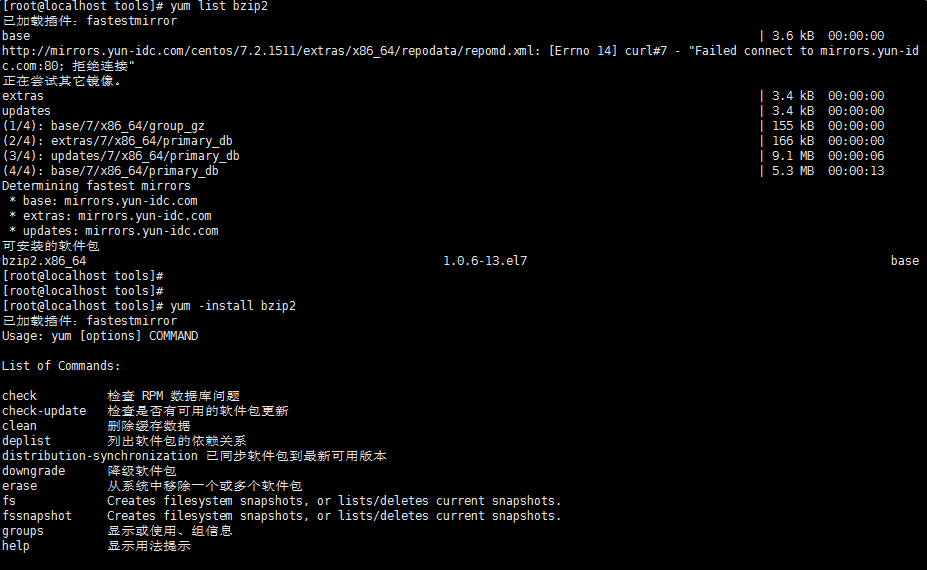
最后设置一下root密码：/usr//bin/mysqladmin -u root password 'root'

### 9.1错误问题描述：



### 9.2解决办法：

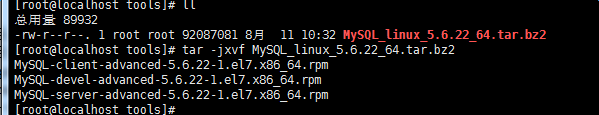
错误的原因是：centos不没有安装解压缩的biz2的应用文件



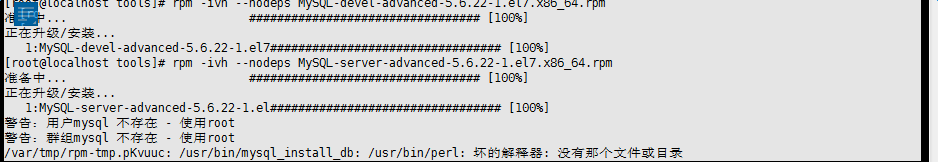
安装成功



解压成功



安装过程中出现的错误





1、安装查看有没有安装过：

 yum list installed [**MySQL**](http://lib.csdn.net/base/mysql)\*

 rpm -qa | grep mysql\*

查看有没有安装包：

 yum list mysql\*

移除已经安装的 mysql

yum remove mysql mysql-server mysql-libs compat-mysql51  
rm -rf /var/lib/mysql  
rm /etc/my.cnf  
查看是否还有mysql软件：  
rpm -qa|grep mysql  
有的话继续删除

安装mysql客户端：

 yum install mysql

安装mysql 服务器端：

 yum install mysql-server

yum install mysql-devel

2、启动&&停止 [**数据库**](http://lib.csdn.net/base/mysql)字符集设置

 mysql配置文件/etc/my.cnf中加入character-set-server=utf8

启动mysql服务： service mysqld start或者/etc/init.d/mysqld start

开机启动：

添加开机启动：chkconfig --add mysqld;

开机启动：chkconfig mysqld on;

查看开机启动设置是否成功chkconfig --list | grep mysql\* mysqld 0:关闭 1:关闭 2:启用 3:启用 4:启用 5:启用 6:关闭停止：

service mysqld stop

3、登录 创建root管理员：

 mysqladmin -u root password 123456

 登录： mysql -u root -p输入密码即可。忘记密码：

service mysqld stop;

mysqld\_safe --user=root --skip-grant-tables;

这一步骤执行的时候不会出现新的命令行，你需要重新打开一个窗口执行下面的命令

mysql -u root;

use mysql ;

update user set password=password("123456") where user="root";

flush privileges;

4、远程访问 开放防火墙的端口号mysql

增加权限：mysql库中的user表新增一条记录host为“%”，user为“root”。

5、[**Linux**](http://lib.csdn.net/base/linux) MySQL的几个重要目录

 数据库目录 /var/lib/mysql/

配置文件 /usr/share /mysql（mysql.server命令及配置文件）

相关命令 /usr/bin（mysqladmin mysqldump等命令）

启动脚本 /etc/rc.d/init.d/（启动脚本文件mysql的目录）

6、删除 mysql 数据库

如果使用的是 yum 安装的 mysql，需要删除的话，就使用如下命令：

yum -y remove mysql\*

然后将 /var/lib/mysql文件夹下的所有文件都删除干净

最后再重新执行上面的安装步骤。

7.授权用户从远程登录

注意：下面的2步骤都需要执行，步骤1.只是修改了用户的密码，但是并没有对用户进行授权，步骤2的作用就是，授权，让用户可以拥有指定的权限（连接数据库，查询数据库...）；

1。 改表法。可能是你的帐号不允许从远程登陆，只能在localhost。这个时候只要在localhost的那台电脑，登入mysql后，更改 "mysql" 数据库里的 "user" 表里的 "host" 项，从"localhost"改称"%"

mysql -u root -pvmware;

mysql>use mysql;

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

mysql>select host, user from user;

2. 授权法。例如，你想myuser使用mypassword从任何主机连接到mysql服务器的话。

GRANT ALL PRIVILEGES ON \*.\* TO 'myuser'@'%' IDENTIFIED BY 'mypassword' WITH GRANT OPTION;

如果你想允许用户myuser从ip为192.168.1.3的主机连接到mysql服务器，并使用mypassword作为密码

GRANT ALL PRIVILEGES ON \*.\* TO 'myuser'@'192.168.1.3' IDENTIFIED BY 'mypassword' WITH GRANT OPTION;

【下面这一句一定要执行，否则还是无法登陆】

mysql>flush privileges ;

如果用户无法从本地登陆，这个时候就执行如下

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'localhost.localdomain' IDENTIFIED BY '123456' WITH GRANT OPTION;

别忘了加上如下的语句

flush privileges ;

1、安装查看有没有安装过：

 yum list installed [**MySQL**](http://lib.csdn.net/base/mysql)\*

 rpm -qa | grep mysql\*

查看有没有安装包：

 yum list mysql\*

移除已经安装的 mysql

yum remove mysql mysql-server mysql-libs compat-mysql51  
rm -rf /var/lib/mysql  
rm /etc/my.cnf  
查看是否还有mysql软件：  
rpm -qa|grep mysql  
有的话继续删除

安装mysql客户端：

 yum install mysql

安装mysql 服务器端：

 yum install mysql-server

yum install mysql-devel

2、启动&&停止 [**数据库**](http://lib.csdn.net/base/mysql)字符集设置

 mysql配置文件/etc/my.cnf中加入character-set-server=utf8

启动mysql服务： service mysqld start或者/etc/init.d/mysqld start

开机启动：

添加开机启动：chkconfig --add mysqld;

开机启动：chkconfig mysqld on;

查看开机启动设置是否成功chkconfig --list | grep mysql\* mysqld 0:关闭 1:关闭 2:启用 3:启用 4:启用 5:启用 6:关闭停止：

service mysqld stop

3、登录 创建root管理员：

 mysqladmin -u root password 123456

 登录： mysql -u root -p输入密码即可。忘记密码：

service mysqld stop;

mysqld\_safe --user=root --skip-grant-tables;

这一步骤执行的时候不会出现新的命令行，你需要重新打开一个窗口执行下面的命令

mysql -u root;

use mysql ;

update user set password=password("123456") where user="root";

flush privileges;

4、远程访问 开放防火墙的端口号mysql

增加权限：mysql库中的user表新增一条记录host为“%”，user为“root”。

5、[**Linux**](http://lib.csdn.net/base/linux) MySQL的几个重要目录

 数据库目录 /var/lib/mysql/

配置文件 /usr/share /mysql（mysql.server命令及配置文件）

相关命令 /usr/bin（mysqladmin mysqldump等命令）

启动脚本 /etc/rc.d/init.d/（启动脚本文件mysql的目录）

6、删除 mysql 数据库

如果使用的是 yum 安装的 mysql，需要删除的话，就使用如下命令：

yum -y remove mysql\*

然后将 /var/lib/mysql文件夹下的所有文件都删除干净

最后再重新执行上面的安装步骤。

7.授权用户从远程登录

注意：下面的2步骤都需要执行，步骤1.只是修改了用户的密码，但是并没有对用户进行授权，步骤2的作用就是，授权，让用户可以拥有指定的权限（连接数据库，查询数据库...）；

1。 改表法。可能是你的帐号不允许从远程登陆，只能在localhost。这个时候只要在localhost的那台电脑，登入mysql后，更改 "mysql" 数据库里的 "user" 表里的 "host" 项，从"localhost"改称"%"

mysql -u root -pvmware;

mysql>use mysql;

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

mysql>select host, user from user;

2. 授权法。例如，你想myuser使用mypassword从任何主机连接到mysql服务器的话。

GRANT ALL PRIVILEGES ON \*.\* TO 'myuser'@'%' IDENTIFIED BY 'mypassword' WITH GRANT OPTION;

如果你想允许用户myuser从ip为192.168.1.3的主机连接到mysql服务器，并使用mypassword作为密码

GRANT ALL PRIVILEGES ON \*.\* TO 'myuser'@'192.168.1.3' IDENTIFIED BY 'mypassword' WITH GRANT OPTION;

【下面这一句一定要执行，否则还是无法登陆】

mysql>flush privileges ;

如果用户无法从本地登陆，这个时候就执行如下

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'localhost.localdomain' IDENTIFIED BY '123456' WITH GRANT OPTION;

别忘了加上如下的语句

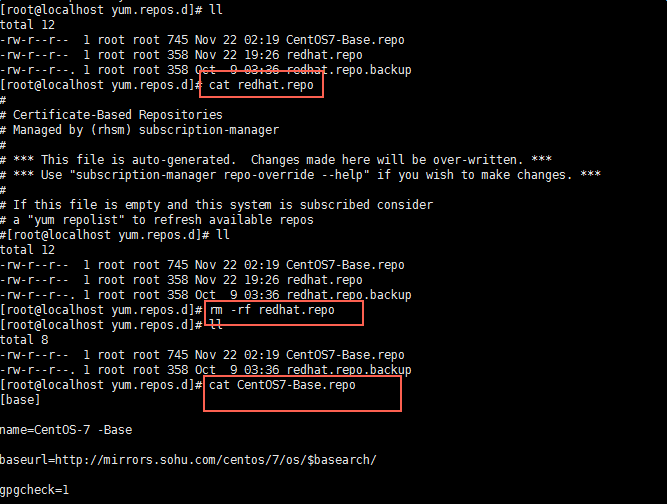
flush privileges ;

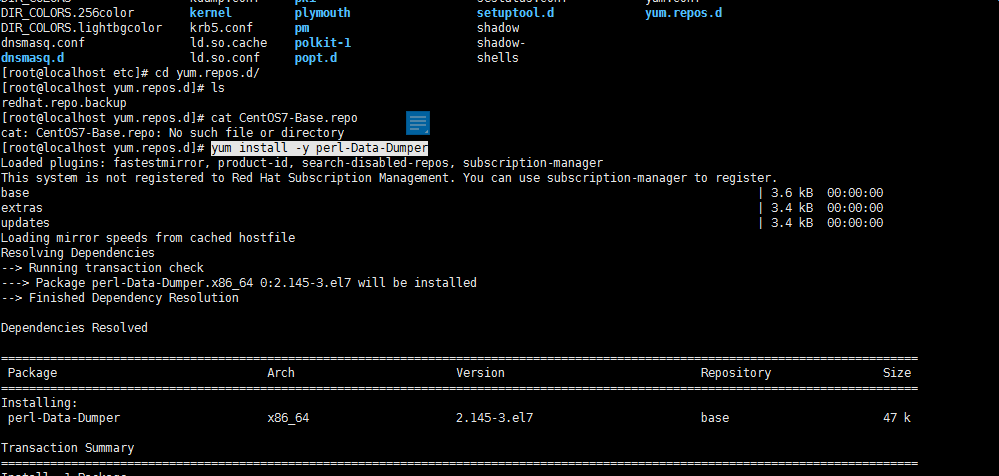
### 9.3解决数据库装不上的问题总结：

[root@localhost yum.repos.d]# yum makecache ##初始化缓存

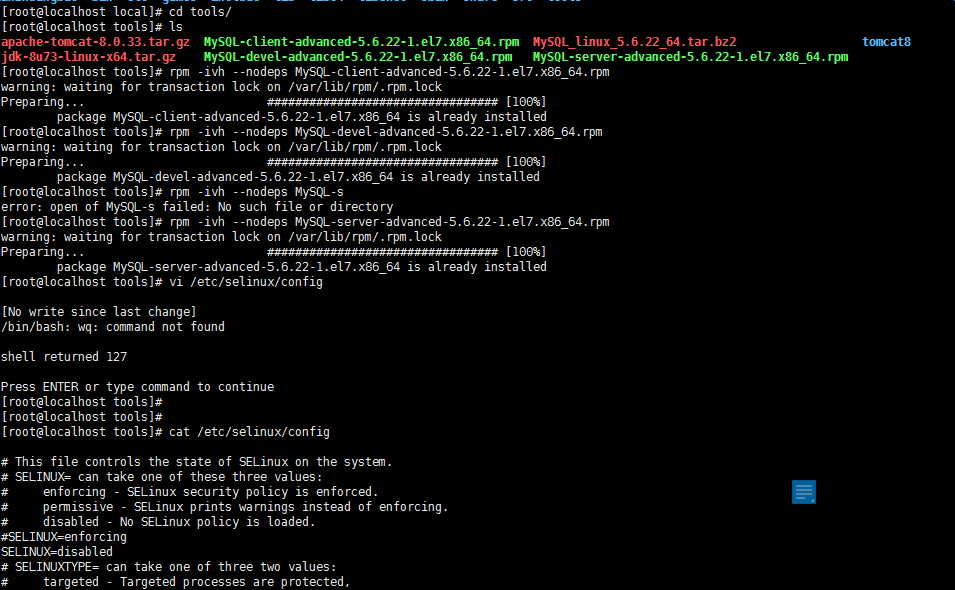
[root@localhost yum.repos.d]# yum clean all 清楚缓存

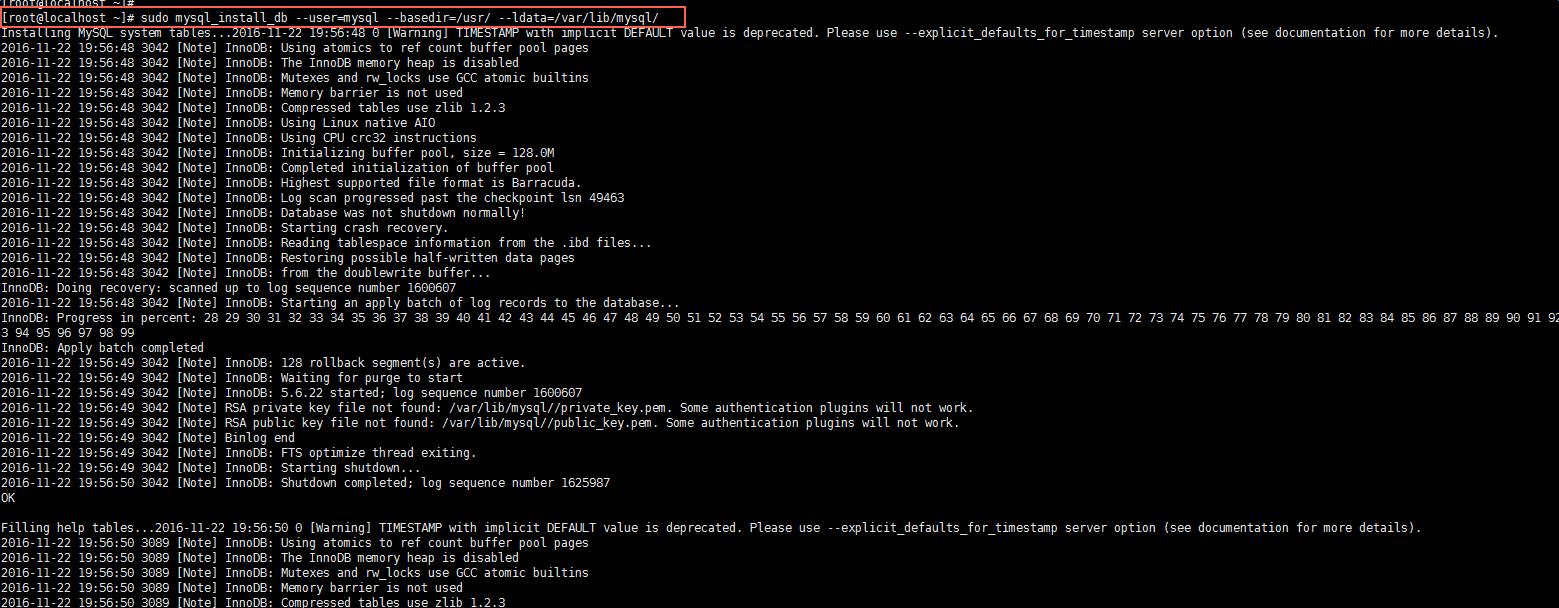




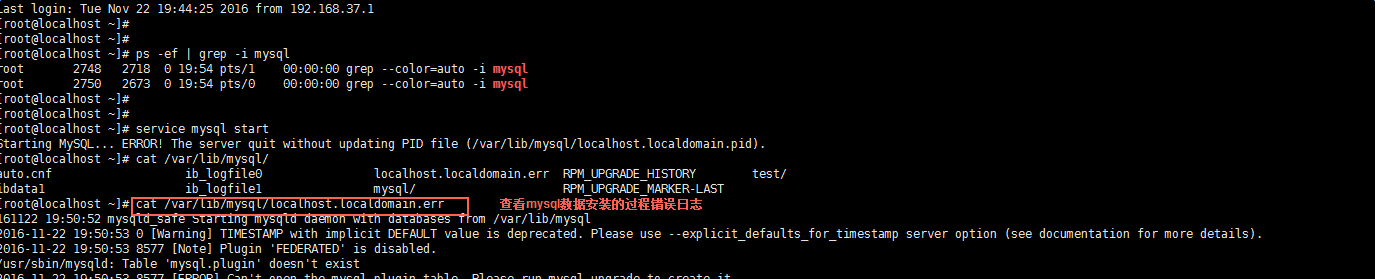


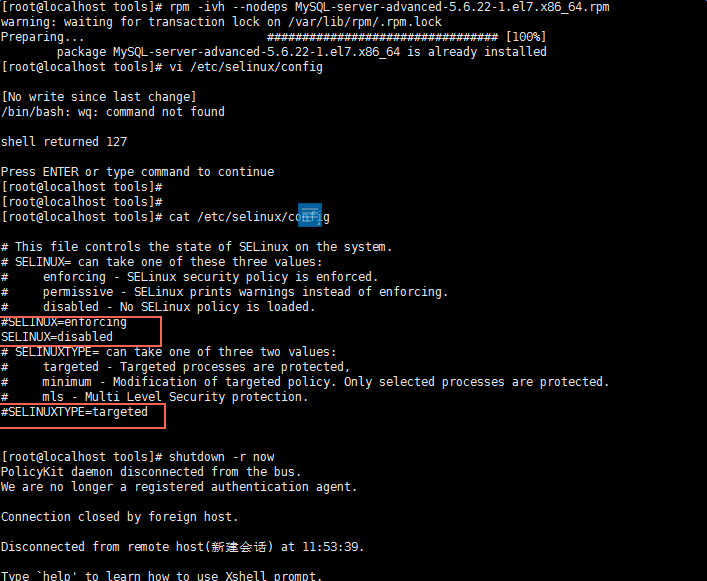
安装完成后没有pid文件问题解决：



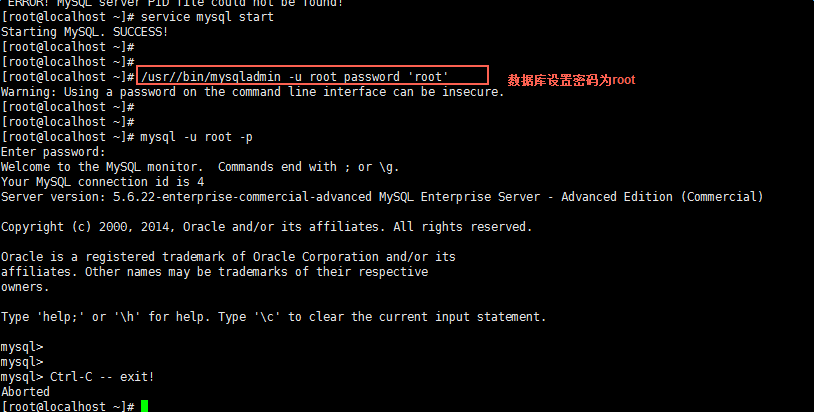
[root@localhost ~]#sudo mysql\_install\_db --user=mysql --basedir=/usr/ --ldata=/var/lib/mysql/

[root@localhost ~]# cat /var/lib/mysql/localhost.localdomain.err





启动数据库，数据库设置密码



## 10：把tabs页做成配置的，控制显示隐藏

### 10.1代码后代码：

/\*\*

\* 根据项目类型控制tab页展示

\*/

**private** **void** controlTabPageDisplay(Integer projectCate,Context context){

ProjectFormIdRelationInfo findByProjectTypeCode = projectFormIdRelationService.findByProjectTypeCode(Long.*valueOf*(projectCate));

**if**(**null** != findByProjectTypeCode){

String tab = findByProjectTypeCode.getTabsId();

List<String> subString = BpmBaseUtils.*getSubString*(tab);

List<ProjectTabEnum> projectTab = Lists.*newArrayList*();

**for** (ProjectTabEnum projectTabEnum : ProjectTabEnum.*values*()) {

**if** (subString != **null**) {

**if** (subString.contains(String.*valueOf*(projectTabEnum.getCode()))) {

projectTabEnum.setChecked(**true**);

} **else** {

projectTabEnum.setChecked(**false**);

}

} **else** {

projectTabEnum.setChecked(**false**);

}

projectTab.add(projectTabEnum);

}

context.put("tabList", projectTab);

}**else**{

context.put("tabs","");

}

}

### 10.2 vm页面：

#set($cate = $param.dataCate)

<div class="nav" style="margin-top: 40px;">

<ul class="cc" id="J-cate-ul">

#foreach($tabs in $!tabList)

<li #cur($cate "current")>

#if($!tabs.code==1 && $!tabs.Checked)

<a href="javascript:void(0)" data-cate="current">详细信息</a>

#end

</li>

<li #cur($cate "changelist") >

#if($!tabs.code==2 && $!tabs.Checked)

<a href="javascript:void(0)" data-cate="changelist">变更记录</a>

#end

</li>

<li #cur($cate "contactnote")>

#if($!tabs.code==3 && $!tabs.Checked)

<a href="javascript:void(0)" data-cate="contactnote">联系小记</a>

#end

</li>

##if($!projectBaseInfo.projectCate == 1)

<li #cur($cate "planconfigure")>

#if($!tabs.code==4 && $!tabs.Checked)

<a href="javascript:void(0)" data-cate="planconfigure">商务过程</a>

#end

</li>

##end

<li #cur($cate "projectfile")>

#if($!tabs.code==5 && $!tabs.Checked)

<a href="javascript:void(0)" data-cate="projectfile">项目文档</a>

#end

</li>

#end

</ul>

</div>#if($cate == "current")

## 11.Linux搭建nginx环境：

### 11.1安装必备工具：

yum -y install gcc gcc-c++ autoconf automake

yum -y install zlib zlib-devel openssl openssl-devel pcre-devel

pcre: 用来作地址重写的功能。

zlib：nginx 的gzip模块,传输数据打包，省流量（但消耗资源）。

openssl：提供ssl加密协议。

新建一个系统级用户组和匿名用户，以及下面编译时使用：

sudo groupadd -r nginx

sudo useradd -s /sbin/nologin -g nginx -r nginx-user

Nginx编译安装：

tar -zxvf nginx-1.9.2.tar.gz

cd nginx-1.9.2/

./configure \

运行后生成：

Configuration summary

+ using system PCRE library

+ OpenSSL library is not used

+ md5: using system crypto library

+ sha1: using system crypto library

+ using system zlib library

nginx path prefix: "/usr/local/nginx"

nginx binary file: "/usr/local/nginx/sbin/nginx"

nginx configuration prefix: "/usr/local/nginx/conf"

nginx configuration file: "/usr/local/nginx/conf/nginx.conf"

nginx pid file: "/usr/local/nginx/logs/nginx.pid"

nginx error log file: "/usr/local/nginx/logs/error.log"

nginx http access log file: "/usr/local/nginx/logs/access.log"

nginx http client request body temporary files: "client\_body\_temp"

nginx http proxy temporary files: "proxy\_temp"

nginx http fastcgi temporary files: "fastcgi\_temp"

nginx http uwsgi temporary files: "uwsgi\_temp"

nginx http scgi temporary files: "scgi\_temp"

make && make install

安装目录查看：whereis nginx

启动：

nginx -c /usr/local/nginx/conf/nginx.conf

如果提示缺什么包，直接用yum安装

平滑重启nginx：

nginx -s reload

### 11.2启动nginx：

[root@localhost local]# /usr/local/nginx/sbin/nginx -s reload

[root@localhost nginx-1.8.1]# cd objs/ nginx启动程序所在的目录

[root@localhost objs]# ll

total 3344

-rw-r--r-- 1 root root 13667 Nov 23 19:52 autoconf.err

-rw-r--r-- 1 root root 36750 Nov 23 19:52 Makefile

-rwxr-xr-x 1 root root 3314327 Nov 23 19:55 nginx

-rw-r--r-- 1 root root 5253 Nov 23 19:55 nginx.8

-rw-r--r-- 1 root root 6147 Nov 23 19:52 ngx\_auto\_config.h

-rw-r--r-- 1 root root 657 Nov 23 19:52 ngx\_auto\_headers.h

-rw-r--r-- 1 root root 3812 Nov 23 19:52 ngx\_modules.c

-rw-r--r-- 1 root root 26648 Nov 23 19:55 ngx\_modules.o

drwxr-xr-x 8 root root 71 Nov 23 19:52 src

[root@localhost objs]# ./nginx 启动nginx

### 11.3查看nginx 的运行状态及查看端口是否开放：

ps –ef |grep nginx

例子：

[root@localhost nginx-1.8.1]# ps -ef| grep nginx

root 8144 1 0 20:06 ? 00:00:00 nginx: master process ./nginx

nobody 8145 8144 0 20:06 ? 00:00:00 nginx: worker process

root 8303 2747 0 22:37 pts/0 00:00:00 grep --color=auto nginx

查看是否开启80端口：

[root@localhost nginx-1.8.1]# firewall-cmd --query-port=80/tcp

no

[root@localhost nginx-1.8.1]# firewall-cmd --add-port=80/tcp

success

11.4 访问查看是否成功启动nginx：

#### 1在本地host 配置ip：



#### 2在浏览器上访问nginx出现如下图为成功：

### 11.5搭建nginx集群：

nginx –v 查看版本

nginx –V 查看nginx配置

vim是vi的增强版，完全兼容vi。vim优势主要体现在以下几个方面：

1、多级撤消

我们知道在vi里，按u只能撤消上次命令，而在vim里可以无限制的撤消。

2、易用性

vi只能运行于unix类中，而vim不仅可以运行于unix,windows ,mac等多操作平台。

3、语法加亮

vim可以用不同的颜色来加亮你的代码。

4、可视化操作

就是说vim不仅可以在终端运行，也可以运行于x window、 mac os、 windows。

5、对vi的完全兼容

某些情况下，你可以把vim当成vi来使用。

#### 1.参考文档：

#user nobody;

worker\_processes 1;

#error\_log logs/error.log;

#error\_log logs/error.log notice;

#error\_log logs/error.log info;

#pid logs/nginx.pid;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

upstream localhost{

server 192.168.32.101:8080 weight=3 max\_fails=3 fail\_timeout=20s;

server 192.168.32.102:8080 weight=1 max\_fails=3 fail\_timeout=20s;

server 192.168.32.103:8080 weight=4 max\_fails=3 fail\_timeout=20s;

}

server {

listen 80;

server\_name localhost www.zhaoyangyang.win;

location / {

proxy\_pass http://localhost;

proxy\_next\_upstream http\_500 http\_502 http\_503 error timeout invalid\_header;

#include /usr/local/nginx/conf/proxy.conf;

}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

}

}

#### 2.实战演练记录：

##### 1查看一下：nginx所在的位置

[root@localhost nginx-1.8.1]# whereis nginx

nginx: /usr/local/nginx

[root@localhost nginx-1.8.1]# /usr/local/nginx/objs/nginx -t

-bash: /usr/local/nginx/objs/nginx: No such file or directory

##### 2检出一下nginx是否正常

[root@localhost nginx-1.8.1]# /usr/local/nginx/sbin/nginx -t

nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok

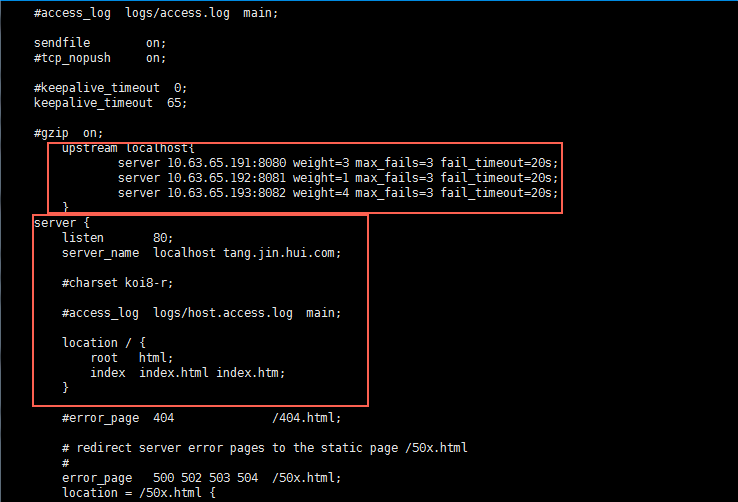
nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful

##### 3平滑重启nginx

[root@localhost nginx-1.8.1]# /usr/local/nginx/sbin/nginx -s reload

[root@localhost nginx-1.8.1]#

4.nginx的配置：



## 12：判断是否为空总结：

### 12.1判断map类型是否为null开发中常用：

例子：

**public** **void** setAclTokens(Map<String, String> aclTokens) {

**this**.aclTokens = aclTokens;

}

/\*\*

\* 依据权限项获取acltoken

\*

\* **@param** key

\* **@return**

\*/

**public** String getAclTokensByKey(String key) {

**if** (aclTokens != **null** && aclTokens.containsKey(key)) {

**return** aclTokens.get(key);

}

**return** **null**;

}

### 12.2判断String类型是否为null开发中常用：

## 13：java代码中正则的使用场景：

**private** **static** **final** Pattern ***KVP\_PATTERN*** = Pattern.*compile*("([\_.a-zA-Z0-9][-\_.a-zA-Z0-9]\*)[=](.\*)"); //key value pair pattern.

**private** **static** Map<String, String> parseKeyValuePair(String str, String itemSeparator)

{

String[] tmp = str.split(itemSeparator);

Map<String, String> map = **new** HashMap<String, String>(tmp.length);

**for**(**int** i=0;i<tmp.length;i++)

{

Matcher matcher = ***KVP\_PATTERN***.matcher(tmp[i]);

**if**( matcher.matches() == **false** )

**continue**;

map.put(matcher.group(1), matcher.group(2));

}

**return** map;

}

## 14：java中的list初始化的妙用及使用场景是根据url进行验权：

**private** List<String> cidsPermission = **new** ArrayList<String>() {

**private** **static** **final** **long** ***serialVersionUID*** = 4095141418283523030L;

{

add("/operationLog/operationLog");

add("/customer/customerInfo");

add("/customer/customerContactInfo");

add("/opportunityInfo/opportunityInfo");

add("/financeInfo/productLineRevenueInfo");

add("/contactnotes/contactNotes");

add("/customer/modifyCustomerInfo");

add("/customer/modifyContactInfo");

add("/customer/editCustomerServiceInfo");

add("/customer/modifyAccountInfo");

add("/opportunityInfo/createOpportunityInfo");

add("/customer/modifyPeopleCustomer");

add("/customer/customerPortrait");

}

};

**if** (cidsPermission.contains(url)) {

param.setCid(cid);

PlainResult<Boolean> result = queryPermissionsService.checkURLPermissionByACL(param);

**if** (result != **null** && result.getData() != **null**) {

**if** (result.getData() == **false**) {

rundata.setRedirectTarget("/error/noAccess");

}

}

}

## 15对字符串的处理方式：

String url=“add/use/UpdateAge.json”

url.endsWith(".json")) {

**boolean** flag=**true**;

**if**(url.contains("//")){

flag=**false**;

}

**if**(url.contains("./")){

flag=**false**;

}

**if**(url.contains("\\\\")){

flag=**false**;

}

**if**(url.contains(".\\")){

flag=**false**;

}

**if**(url.contains("..")){

flag=**false**;

}

**if**(url.contains("\\/")){

flag=**false**;

}

**if**(url.contains("/\\")){

flag=**false**;

}

**if**(url.matches(".\*\\s.\*")){

flag=**false**;

}

**if**(!flag){

// 无权限

BaseResult baseResult = **new** BaseResult();

baseResult.setCode(CommonResultCode.ERROR\_CALL\_ADMIN.code);

baseResult.setMessage(CommonResultCode.ERROR\_CALL\_ADMIN.message);

baseResult.setSuccess(**true**);

response.setContentType("application/json");

PrintWriter out = response.getWriter();

String jsonResult = JSON.toJSONString(baseResult);

String anquan = SecurityUtil.escapeJson(jsonResult);

out.print(anquan);

**return**;

/\*\*

#### \* 根据逗号进行切分

\* **@param** str

\* **@return**

\*/

**public** **static** List<String> getSubString(String str) {

**if** (str != **null**) {

String[] uidArr = str.split(",");

List<String> stringList = Lists.*newArrayList*();

**for** (**int** i = 0; i < uidArr.length; i++) {

**if** (StringUtil.*isNotEmpty*(uidArr[i]))

stringList.add(uidArr[i]);

}

**return** stringList;

}

**return** **null**;

}

## 16huishoulaji.cn域名：

万网域名解析设置方法

更新时间：2016-07-12 13:22:55

域名注册后，客户只拥有了这个域名的使用权，但无法通过域名直接访问您的网站，或作为电子邮箱后缀进行邮件收发。“域名解析”是使用域名访问您的网站或邮箱的必备环节。

建议选择万网云解析服务，确保域名在解析过程中拥有稳定、快速、安全的服务保障。

一、准备域名、网站（或邮箱）

您需要拥有一个域名以及从服务器提供商处获取主机 IP 地址（或电子邮箱 MX 记录），以便完成后续解析设置。

二、设置域名解析

登陆阿里云/万网【管理控制台】，进入域名解析列表，把域名指向网站主机 IP 地址（或电子邮箱 MX 记录），即可快速完成域名解析设置。具体操作流程如下：

1. 进入域名解析设置页

1) 登陆阿里云/万网【管理控制台】，在顶部主导航位置点击【产品与服务】--【云解析】，进入“域名解析列表”；选择需添加解析的域名，点击右侧操作的【解析】入口，即可进入到域名解析设置页；

【提醒】非万网域名用户，进入“域名解析列表”后，点击列表右上方【添加域名】，在弹出框中输入您需要设置解析的域名，点击【添加】即可；添加成功后点击域名右侧【解析】入口，即可进入域名解析设置页。

2、 新增解析

以将域名指向您的网站为例，对于新手用户，我们支持一键解析，您只需要填写网站的 IP 地址，即可完成解析。

【提醒】客户可以【进入高级设置】，点击【添加解析】，选择记录类型为 A 记录；主机记录为空，或填写 www；记录值填写您之前获取的主机 IP 地址；解析线路，TTL 默认即可。点击保存，即可完成域名解析设置。

点击查看 域名 CNAME 解析设置方法

点击查看 域名URL转发解析设置方法

3、DNS 检查

域名解析设置完成后，需确定使用万网云解析 DNS 方能解析生效。

万网域名用户在首次解析时已使用万网 DNS，无需再做修改。

如果使用的是万网 DNS，无需修改，直接进入“4、解析生效时间”，等待生效

如果使用的非万网的 DNS，如何修改成万网 DNS 请 点此查看

4、解析生效时间

新增解析完成设置即时生效。

修改解析则需要 10 分钟~2 小时，最终生效取决于各地运营商的缓存刷新时间，请耐心等待。

解析多久才能生效请 点此查看

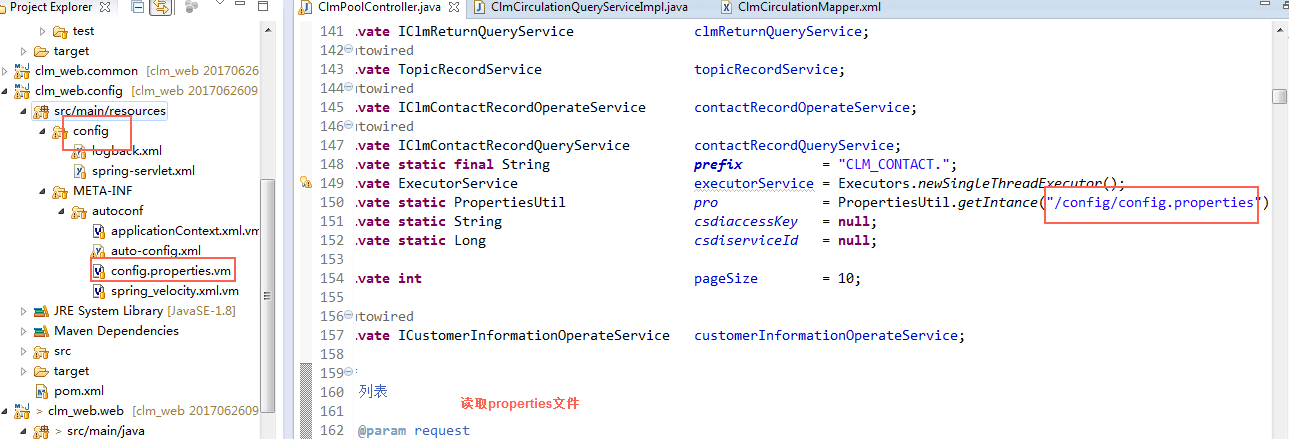
注：如果您的域名指向的 IP 为万网虚拟主机，请在管理控制台绑定域名 。

如果问题还未能解决，请联系售后技术支持。

## 17linux使用yum源安装yum install lrzsz

使用rz ：导入工程或项目（代替链接工具）

## 18实例化properties文件



/\*

\* Copyright (c) 2012-2032 Air China. All Rights Reserved.

\*/

package com.wuage.clm.util;

import java.io.FileInputStream;

import java.io.IOException;

import java.io.InputStream;

import java.io.UnsupportedEncodingException;

import java.util.ArrayList;

import java.util.Enumeration;

import java.util.List;

import java.util.Properties;

import org.apache.commons.lang.StringUtils;

/\*\*

\* @Description:

\* @project login.common

\* @class PropertiesUtil.java

\* @author moyouqing

\* @version 1.0

\* @date 2016年3月9日 下午4:00:35

\*/

public class PropertiesUtil {

private static PropertiesUtil instance = null;

private Properties properties = null;

/\*\*

\* 功能: 获取配置实例

\*

\* @return PropertyReader

\*/

public static PropertiesUtil getIntance() {

if (instance == null) instance = new PropertiesUtil("/conf.properties");

return instance;

}

/\*\*

\* 功能: 根据指定路径获取配置实例

\*

\* @return PropertyReader

\*/

public static PropertiesUtil getIntance(String path) {

instance = new PropertiesUtil(path);

return instance;

}

/\*\*

\* 构造方法

\*/

private PropertiesUtil(String path){

properties = this.createProperties(path);

if (properties == null) properties = new Properties();

}

/\*\*

\* 功能: 得到一个整数属性,默认为0

\*

\* @param key 键值

\* @return Integer

\*/

public Integer getInt(String key) {

String prop = this.getProperty(key);

if ("".equals(prop)) {

return Integer.valueOf(0);

}

return Integer.valueOf(prop);

}

/\*\*

\* 功能: 得到一个长整数属性,默认为0

\*

\* @param key 键值

\* @return Long

\*/

public Long getLong(String key) {

String prop = this.getProperty(key);

if ("".equals(prop)) {

return Long.valueOf(0);

}

return Long.valueOf(prop);

}

/\*\*

\* 功能: 得到一个属性值

\*

\* @param key 键值

\* @return String

\*/

public String getProperty(String key) {

if ("".equals(StringUtils.isBlank(key))) {

return "";

}

return properties.getProperty(key);

}

/\*\*

\* 功能: 把读取的文字进行中文转化GBK格式

\*

\* @param key 键值

\* @return String

\*/

public String getConvertGBK(String key) {

String rs = "";

String prop = this.getProperty(key);

if (prop.equals("")) return rs;

else {

try {

rs = new String(prop.getBytes("ISO8859\_1"), "GBK");

} catch (UnsupportedEncodingException e) {

e.printStackTrace();

}

}

return rs;

}

/\*\*

\* 功能: 把读取的文字进行中文转化UTF-8格式

\*

\* @param key 键值

\* @return String

\*/

public String getPropertyByUTF8(String key) {

String rs = "";

String prop = this.getProperty(key);

if (prop.equals("")) return rs;

else {

try {

rs = new String(prop.getBytes("ISO8859\_1"), "UTF-8");

} catch (UnsupportedEncodingException e) {

e.printStackTrace();

}

}

return rs;

}

/\*\*

\* 功能: 把读取的文字进行中文转化UTF-8格式

\*

\* @param key 键值

\* @return String

\*/

public List<Short> getListShort(String key) {

String prop = this.getProperty(key);

List<Short> ret = new ArrayList<Short>();

if (!"".equals(prop)) {

String[] str = prop.split(",");

for (String s : str) {

ret.add(Short.parseShort(s));

}

}

return ret;

}

/\*\*

\* 功能: 得到所有的属性值

\*

\* @return Enumeration

\*/

@SuppressWarnings("unchecked")

public Enumeration getValues() {

try {

return properties.elements();

} catch (Exception ex) {

return null;

}

}

/\*\*

\* 功能: 得到所有键值

\*

\* @return Enumeration

\*/

@SuppressWarnings("unchecked")

public Enumeration getKeys() {

try {

return properties.propertyNames();

} catch (Exception ex) {

return null;

}

}

/\*\*

\* 功能: 根据给定的路径读取属性文件

\*

\* @param path 指定路径

\* @return properties

\*/

private Properties createProperties(String path) {

Properties p = null;

Object obj = null;

try {

obj = getClass().getResourceAsStream(path);

if (obj == null) obj = new FileInputStream(path);

} catch (IOException e) {

System.err.println("System can not find the property!");

}

if (obj != null) {

try {

p = new Properties();

p.load((InputStream) obj);

((InputStream) obj).close();

} catch (IOException e) {

p = null;

}

}

return p;

}

/\*\*

\* @Description:功能: 把读取的文字进行中文转化UTF-8格式

\* @author chenqhui

\* @version 1.0

\* @date 2016年3月21日 下午4:44:47

\* @param key

\* @return

\*/

public String getPropertyByUTF8(String key, String defaultValue) {

String rs = defaultValue;

String prop = this.getProperty(key);

if (StringUtils.isBlank(prop)) {

return rs;

} else {

try {

rs = new String(prop.getBytes("ISO8859\_1"), "UTF-8");

} catch (UnsupportedEncodingException e) {

e.printStackTrace();

}

}

return rs;

}

}

## java8新特性

### 1.lamda表达式：遍历

list.forEach(modele -> {

net.speedup.workflow.backend.statical.domain.Model model = **new** net.speedup.workflow.backend.statical.domain.Model(modele);

beans.add(model);

});