# 京淘项目Day07

## 项目优化

### 业务需求

说明:有时文件上传地址/url地址可能会根据环境不同发生变化.如果路径发生修改则必然修改java代码! 如何优化

解决思路:

将重要的配置提取到配置文件中.

### 编辑Properties文件

#配置本地磁盘路径 不要由多余的空格

image.localDirPath=D:/1-jt/image/

#配置虚拟路径url路径

image.urlPath=http://image.jt.com/

### 编辑Service业务类

@Service

@PropertySource("classpath:/properties/image.properties")

//@ConfigurationProperties(prefix = "image") 依赖get/set方法

**public** **class** FileServiceImpl **implements** FileService {

//定义本地磁盘路径

@Value("${image.localDirPath}")

**private** String localDirPath;

//定义虚拟路径名称

@Value("${image.urlPath}")

**private** String urlPath;

}

## 负载均衡

### 准备3台tomcat服务器

#### 添加打包插件

在jt-manage中添加插件

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

#### 原始手段

1. 将项目打成jar/war
2. 将war发布到tomcat服务器中的webapps,并且更名为ROOT.war
3. 修改tomcat服务器中的配置文件server.xml 修改业务端口8080.
4. 利用tomcat服务器中bin目录下的start.bat文件启动服务.

#### SpringBoot项目发布

1. 将项目打包 选中jt项目指挥maven install
2. 利用java命令 启动服务.

java –jar xxxx.war

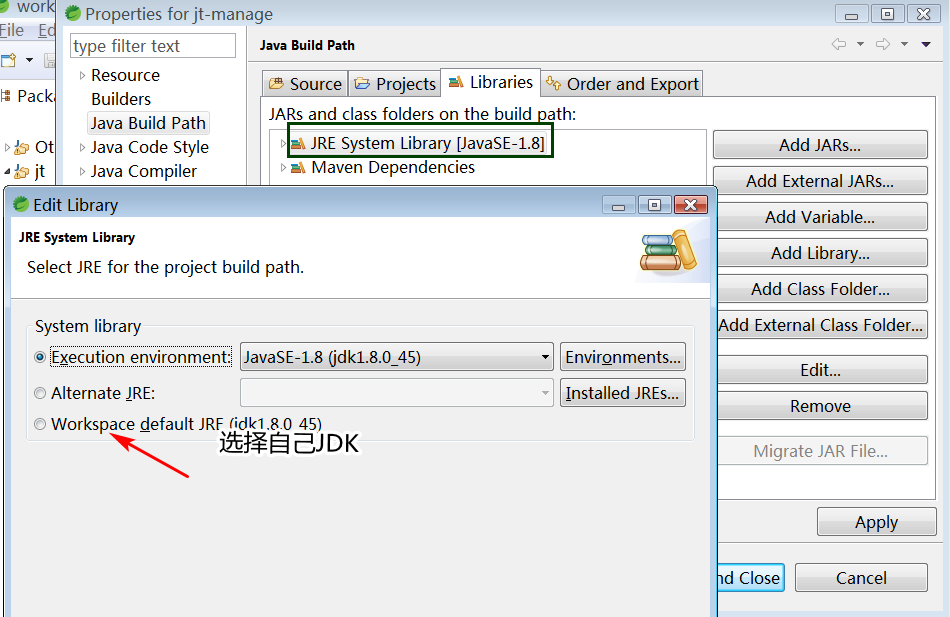
1. 部署启动服务器时 ,.修改YML文件中的端口号即可.

### 关于Maven打包异常问题

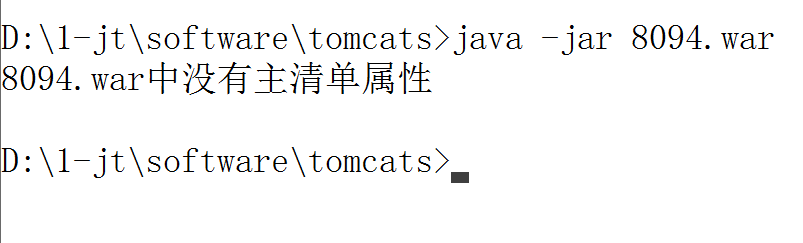
#### 打包插件报错

切换阿里云镜像服务器.之后强制更新maven.

#### 报JDK/JRE异常



#### 启动时没有主启动类



打包时没有添加bulid标签导致的

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

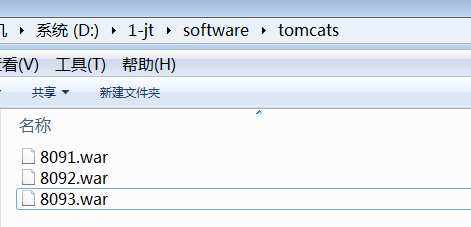
<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

### 集群部署测试

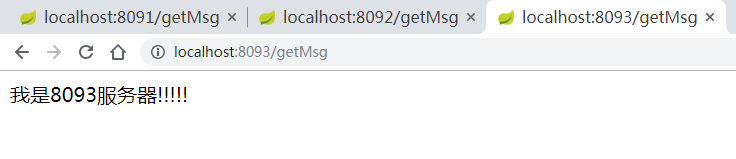


当部署服务器8091/8092/8093之后,分别部署测试

Java –jar 8091.war

Java –jar 8092.war

Java –jar 8093.war 启动测试



## Nginx实现负载均衡

### L轮询策略

策略说明:根据配置文件顺序,依次访问后台服务器.

#设定负载均衡策略 名称不要加"\_"线 1.默认规则 轮询

upstream jt-windows {

server 127.0.0.1:8091;

server 127.0.0.1:8092;

server 127.0.0.1:8093;

}

#后台管理服务器 用户访问manage.jt.com时访问localhost:8091

server {

listen 80;

server\_name manage.jt.com;

location / {

#代理路径

proxy\_pass http://**jt-windows**;

}

}

### 权重配置

策略说明:尽可能让性能高的服务器多处理请求.

upstream jt-windows {

server 127.0.0.1:8091 weight=6;

server 127.0.0.1:8092 weight=3;

server 127.0.0.1:8093 weight=1;

}

### IPHASH

问题描述:用户信息保存到Seesion中时,由于负载均衡的配置.导致用户**每次访问的服务器都不同**,导致用户信息不能共享.

策略说明:使用IPHASH策略.将用户**IP地址**与**tomcat服务器进行绑定**.保证每次访问都是相同的tomcat.

配置:

upstream jt-windows {

ip\_hash;

server 127.0.0.1:8091;

server 127.0.0.1:8092;

server 127.0.0.1:8093;

}

缺点:

1. 用户的IP地址可能发生变化,这时需要重新登录.
2. 如果后台服务器宕机.则影响一部分用户不能正常访问.
3. 用户负载不均.

总结:IPhash工作中不常使用. 测试操作时可能用到.

实际开发中使用SSO(单点登录)实现用户数据共享

SSO:只要用户登录其中一台服务器,则可能访问相互之间信任的全部服务器.

### 常用属性

#### down属性

如果后台服务器宕机.则可以通过down属性标识.Nginx负载均衡时不会再访问该服务器.

#设定负载均衡策略 名称不要加"\_"线 1.默认规则 轮询

upstream jt-windows {

#ip\_hash;

server 127.0.0.1:8091 **down**;

server 127.0.0.1:8092;

server 127.0.0.1:8093;

}

#### backup属性

说明:一般使用backup标识备用机.正常情况下用户不会访问备用机.当主机宕机或者主机遇忙时才会访问备用机.

upstream jt-windows {

#ip\_hash;

server 127.0.0.1:8091 down;

server 127.0.0.1:8092 down;

server 127.0.0.1:8093 **backup**;

}

### 服务器上线步骤

1. 将项目打包部署 jar/war
2. 根据服务器数量设计上线计划.
3. 现将部署的服务器在nginx中down.
4. 之后将项目进行部署.测试.
5. 将down属性去掉部署其他的tomcat服务器.

问题:如果在nginx重启时用户发起请求.问如何处理???

说明:一般用户请求访问服务器时都会有超市时间.3-5秒

而Nginx启动的时间不足1秒.启动之后继续为用户提供服务.

### Tomcat高可用

高可用:当服务器发生故障时,能够**自动的**实现故障的迁移.

实现说明:

在tomcat服务器之后标识2个属性.一个是失败的次数.一个是延时时间.

如果用户访问服务器.当访问失败的次数达到最大失败次数.则在一个延时周期内,不会再次请求该服务器.直到下一个周期.

#设定负载均衡策略 名称不要加"\_"线 1.默认规则 轮询

upstream jt-windows {

#ip\_hash;

server 127.0.0.1:8091 max\_fails=1 fail\_timeout=60s;

server 127.0.0.1:8092 max\_fails=1 fail\_timeout=60s;

server 127.0.0.1:8093 max\_fails=1 fail\_timeout=60s;

}

#后台管理服务器 用户访问manage.jt.com时访问localhost:8091

server {

listen 80;

server\_name manage.jt.com;

location / {

#代理路径

proxy\_pass <http://jt-windows>;

#链接服务器超时

proxy\_connect\_timeout 2;

#读取服务器资源超时

proxy\_read\_timeout 2;

#向服务器发送数据超时

proxy\_send\_timeout 2;

}

}

}

# Linux

## Linux概述

### Linux介绍

Linux是一套免费使用和自由传播的[类Unix](https://baike.baidu.com/item/%E7%B1%BBUnix)[操作系统](https://baike.baidu.com/item/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F/192)，是一个基于[POSIX](https://baike.baidu.com/item/POSIX)和[UNIX](https://baike.baidu.com/item/UNIX)的多用户、[多任务](https://baike.baidu.com/item/%E5%A4%9A%E4%BB%BB%E5%8A%A1/1011764)、支持[多线程](https://baike.baidu.com/item/%E5%A4%9A%E7%BA%BF%E7%A8%8B/1190404)和多[CPU](https://baike.baidu.com/item/CPU)的操作系统。它能运行主要的UNIX工具软件、应用程序和网络协议。它支持[32位](https://baike.baidu.com/item/32%E4%BD%8D/5812218)和[64位](https://baike.baidu.com/item/64%E4%BD%8D)硬件。Linux继承了[Unix](https://baike.baidu.com/item/Unix)以[网络](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C/143243)为核心的设计思想，是一个性能稳定的多用户网络操作系统。

Linux操作系统诞生于1991 年10 月5 日（这是第一次正式向外公布时间）。Linux存在着许多不同的Linux版本，但它们都使用了[Linux内核](https://baike.baidu.com/item/Linux%E5%86%85%E6%A0%B8)。Linux可安装在各种计算机硬件设备中，比如[手机](https://baike.baidu.com/item/%E6%89%8B%E6%9C%BA/6342)、[平板电脑](https://baike.baidu.com/item/%E5%B9%B3%E6%9D%BF%E7%94%B5%E8%84%91/1348389)、[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8/108294)、视频游戏控制台、台式计算机、大型机和超级计算机。

严格来讲，Linux这个词本身只表示Linux内核，但实际上人们已经习惯了用Linux来形容整个基于Linux内核，并且使用[GNU](https://baike.baidu.com/item/GNU)工程各种工具和数据库的操作系统。

### Linux发展史

1.1965 年，Bell 实验室、MIT、GE（通用电气公司）准备开发 Multics 系统，为了同时支持 300 个终端访问主机，但是 1969 年失败了；

（刚开始并没有鼠标、键盘，输入设备只有卡片机，因此如果要测试某个程序，则需要将读卡纸插入卡片机，如果有错误，还需要重新来过； Multics：Multiplexed Information and Computing Service）

2. 1969 年，Ken Thompson（C语言之父）利用汇编语言开发了 FIle Server System（Unics，即 UNIX 的原型）

（因为汇编语言对于硬件的依赖性，因此只能针对特定硬件； 只是为了移植一款“太空旅游”的游戏；）

3. 1973 年，Dennis Ritchie 和 Ken Thompson 发明了 C 语言，而后写出了 UNIX 的内核

（将 B 语言改成 C 语言，由此产生了 C 语言之父； 90% 的代码是 C 语言写的，10% 的代码用汇编写的，因此移植时只要修改那 10% 的代码即可；）

4. 1977 年，Berkeley 大学的 Bill Joy 针对他的机器修改 UNIX 源码，称为BSD（Berkeley Software Distribution）

5. 1979 年，UNIX 发布 System V，用于个人计算机；

1984 年，因为 UNIX 规定：“不能对学生提供源码”，Tanenbaum 老师自己编写兼容于 UNIX 的Minix，用于教学；

6. 1984 年，Stallman 开始 GNU（GNU's Not Unix）项目，创办 FSF（Free Software Foundation）基金会；

（产品：GCC、Emacs、Bash Shell、GLIBC； 倡导“自由软件”； GNU 的软件缺乏一个开放的平台运行，只能在 UNIX 上运行； 自由软件指用户可以对软件做任何修改，甚至再发行，但是始终要挂着 GPL 的版权； 自由软件是可以卖的，但是不能只卖软件，而是卖服务、手册等；）

7. 1985 年，为了避免 GNU 开发的自由软件被其他人用作专利软件，因此创建 GPL（General Public License）版权声明；

1988 年，MIT 为了开发 GUI，成立了 XFree86 的组织；

**1991 年，芬兰赫尔辛基大学的研究生** Linus Torvalds 基于 gcc、bash 开发了针对 386 机器的 **Linux 内核**；

8. 1994 年，Torvalds 发布 Linux-v1.0；

1996 年，Torvalds 发布 Linux-v2.0，确定了 Linux 的吉祥物：企鹅；

如图-9所示

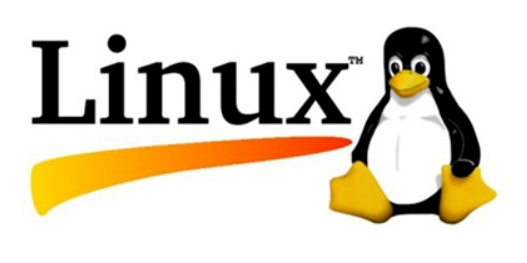


图- 9

### Linux特点

1. 系统开源并且免费
2. 对硬件要求很低
3. 系统稳定性强
4. 系统安全性更好

### Linux系统主流产品

如图-10所示



图- 10

## Linux命令

### cd命令集

ifconfig 检查IP地址 ip addr

pwd 查找当前文件位置

cd命令是linux中最基本的命令语句，必须熟练掌握

cd / 返回根目录

cd ~ 用户主目录

cd . 当前目录

cd ..返回到上一级目录

cd /usr/ 进入到usr目录

cd – 返回上一个目录

cd 直接回家

### ls目录和文件

ls 展现文件目录

ll 展现文件详情

ls –l 详细格式，文件权限，时间

ll 和ls –l作用相同

ls \*.txt 查看所有的txt类型文档

### 目录操作

mkdir 创建目录

mkdir a 创建 a目录

mkdir -p a/b 创建 a目录，并在a目录里创建b目录

mkdir -m 777 c 创建一个权限为777的C目录

rmdir 删除目录（如果目录里有文件，则不能用此命令）

### Vi/vim创建/查看/编辑文件

命令行：Esc切换到命令行模式。

编辑模式：

按i，在光标前开始编辑

按a，在光标后开始编辑

按o，在当前行的下一行开始编辑

按u 撤销之前的操作

底行模式：按 shift+：冒号。

:q! 不保存退出

:wq 保存退出

:/world 从当前光标处，向上查找world关键字

:?world 从当前光标处，向后查找world关键字

### 删除文件

rm 删除文件

rm n.txt 提示y删除n放弃

rm –f n.txt 不提示

rm –rf dirname 不提示递归删除目录下所以内容

rm –rf \* 删除所有文件

rm –rf /\* 删除所有子目录所有和文件

### 复制和移动文件

cp复制文件

cp nginx.conf n.txt

cp –R tomcat1 tomcat2                #复制整个目录

mv 修改文件名，移动文件

mv n.txt m.txt

### 浏览文件

cat 输出文件所有的内容

more 输出文档所有的内容，分页输出，空格浏览下一屏，q退出

less 用法和more相同，只是通过PgUp、PgOn键来控制

tail 用于显示文件后几号，使用频繁

tail -10 nginx.conf 查看nginx.conf的最后10行

tail –f nginx.conf 动态查看日志，方便查看日志新增的信息

ctrl+c 结束查看

### 打包命令

tar命令位于/bin目录下，它能够将用户所指定的文件或目录打包成一个文件，但不做压缩。一般Linux上常用的压缩方式是选用tar将许多文件打包成一个文件，再以gzip压缩命令压缩成name.tar.gz的文件。

-c 创建一个新的tar文件

-v 显示运行过程的信息

-f 指定文件名

-z 调用gzip压缩命令进行压缩

-t 查看压缩文件的内容

-x 解开tar文件

tar –cvf n.tar ./\* 压缩当前目录下的所有文件和目录，文件名为n.tar

tar –xvf n.tar 解压压缩包中的文件到当前目录（如果长时间未解压成功 Ctrl+C推出）

tar –cvzf m.tar ./\* 解压m.tar文件到当前目录

### grep命令

grep root /etc/passwd 在文件中查找关键字root

grep root /etc/passwd –-color         高亮显示

grep root /etc/passwd –A5 –B5         高亮显示，A后5行，B前5行

grep -n root /etc/passwd 查找并显示行数

grep -v root /etc/passwd 取反，查出不含root的数据

## 虚拟机安装和使用

### 虚拟机介绍

虚拟系统通过生成现有[操作系统](https://baike.baidu.com/item/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F/192)的全新虚拟[镜像](https://baike.baidu.com/item/%E9%95%9C%E5%83%8F/1574)，它具有真实[windows](https://baike.baidu.com/item/windows)系统完全一样的功能，进入虚拟系统后，所有操作都是在这个全新的独立的虚拟系统里面进行，可以独立安装运行[软件](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6/12053)，保存数据，拥有自己的独立桌面，不会对真正的系统产生任何影响 ，而且具有能够在现有系统与虚拟镜像之间灵活切换的一类操作[系统](https://baike.baidu.com/item/%E7%B3%BB%E7%BB%9F/479832)。虚拟系统和传统的虚拟机（[Parallels Desktop](https://baike.baidu.com/item/Parallels%20Desktop) ，[Vmware](https://baike.baidu.com/item/Vmware)，[VirtualBox](https://baike.baidu.com/item/VirtualBox)，[Virtual pc](https://baike.baidu.com/item/Virtual%20pc)）不同在于：虚拟系统不会降低电脑的性能，启动虚拟系统不需要像启动[windows](https://baike.baidu.com/item/windows)系统那样耗费时间，运行程序更加方便快捷；虚拟系统只能模拟和现有[操作系统](https://baike.baidu.com/item/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F/192)相同的环境，而虚拟机则可以模拟出其他种类的操作系统；而且虚拟机需要模拟底层的硬件指令，所以在[应用程序](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F/5985445)运行速度上比虚拟系统慢得多。

### 虚拟机配置

如图-11所示

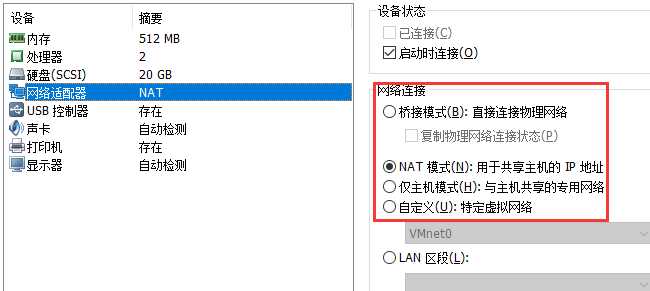


图- 11

内存:一般最大为真实内存的3/4, 一般为512M-1G

硬盘:一般默认20G

如果上述的设置一旦确定,则直接向操作系统割裂对应数据的资源.

网络模式:桥接模式/NAT模式

## 网络模式

### 桥接模式

相当于将虚拟机直接接入局域网中,这时会独占IP地址.可以与本机通过局域网通信.

特点:所有的在局域网中的IP都可以访问虚拟机.

### NAT模式

相当于在网络环境中会开辟一块全新的网络空间.只有主机能够访问.

特点:NAT模式只有主机访问,其他人通过IP地址禁止访问.

如图-12所示

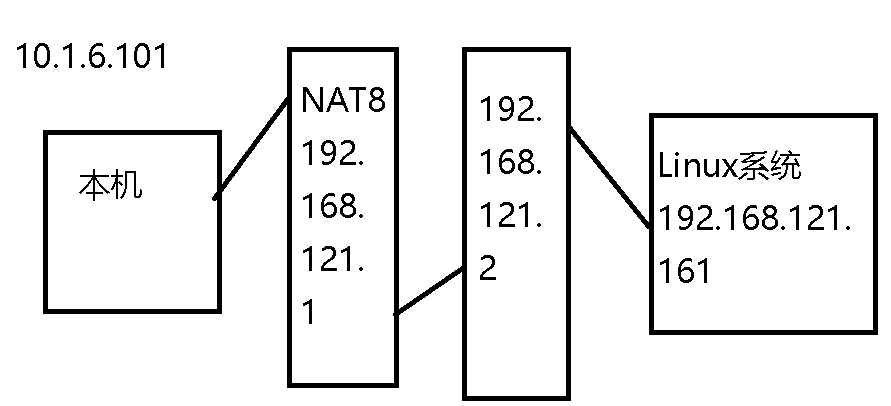


图- 12

## 虚拟机克隆

说明:JT虚拟机中安装了很多插件和环境.为了保护环境不被破坏.所以需要克隆虚拟机备份数据.之后的操作全部在备份的机器上运行.

如图-13所示

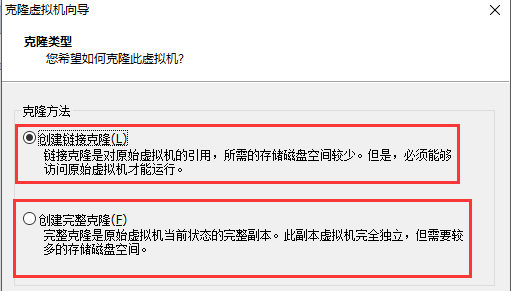


图- 13

如图-14所示

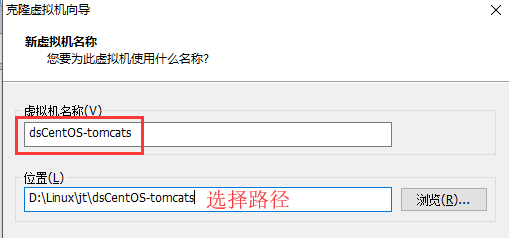
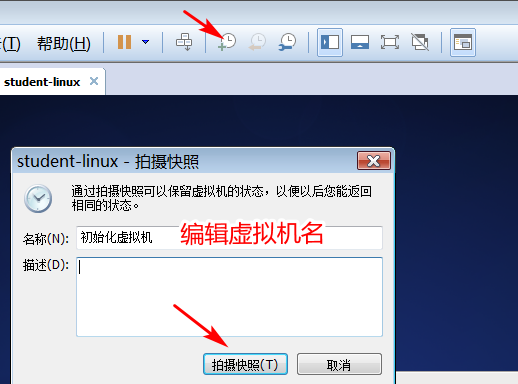


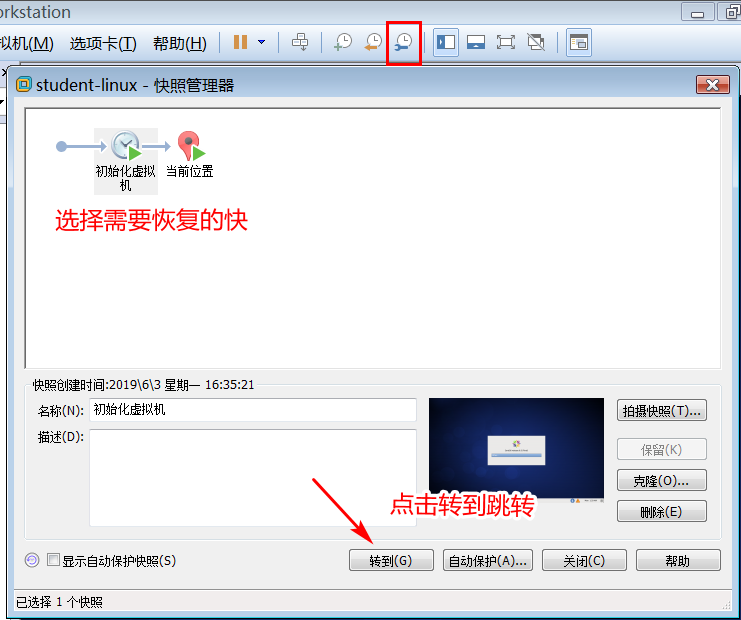
图- 14

## 虚拟机快照

### 拍摄快照



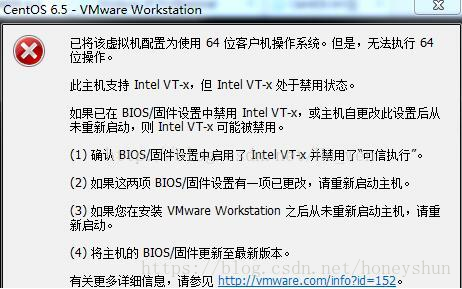
### 回退快照



## 虚拟机启动异常

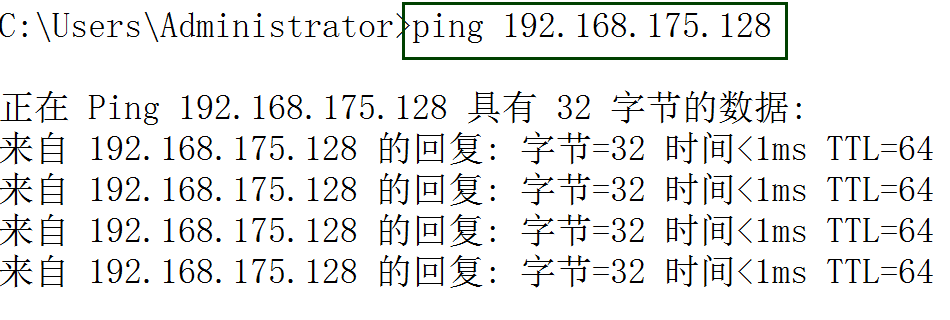
### 虚拟化错误

bOIS系统中开启虚拟化设置. F10保存退出.

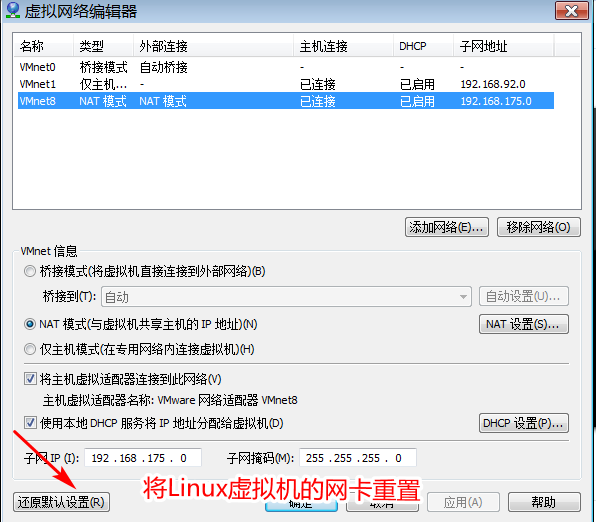


### xShell链接虚拟机失败

1. 检查ping LinuxIP地址



1. 重置网卡



## LinuxIP划分和链接

### IP划分

本机IP:192.168.126.**1**

虚拟机IP:192.168.126.180

说明:以上的IP地址表示在同一个网络空间内.如果前三项不一致,则按照以下方式修改.



## Xshell

### Xshell介绍

说明:Xshell是远程连接Linux系统的工具.可以进行文件传输.命令的编辑等一列的操作.和操作真实的Linux系统一致.

连接步骤:

1. 获取服务器IP地址
2. 获取服务器用户名和密码
3. Xshell远程连接工具都是加密传输 公钥和私钥

### 远程连接方式

说明:新建会话后连接远程虚拟机

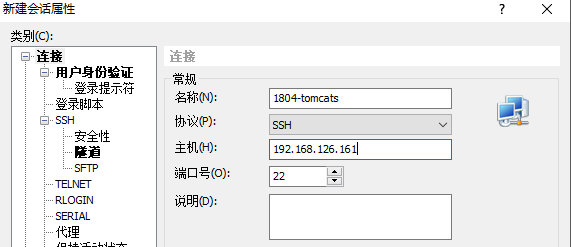


图- 15

## Linux系统安装JDK

### 上传JDK安装包

说明:将JDK安装包上传到指定文件目录下/usr/local/src下

如图-16所示

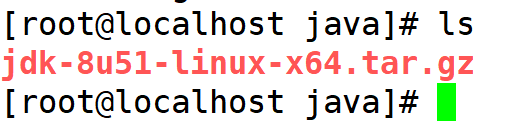


图- 16

### 解压JDK文件

命令: tar -xvf jdk-7u51-linux-x64.tar.gz

如图-17所示

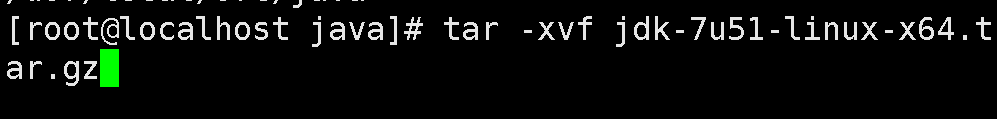


图- 17

### 配置环境变量

说明:修改Linux系统中环境变量需要修改/etc/profile文件

命令:vim /etc/profile

配置文件变量

#set java env

JAVA\_HOME=**/usr/local/src/java/jdk1.7.0\_51**

JAVA\_BIN=/usr/local/src/java/jdk1.7.0\_51/bin

PATH=$JAVA\_HOME/bin:$PATH

CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

export JAVA\_HOME JAVA\_BIN PATH CLASSPATH

环境变量生效:

source /etc/profile

### 检测JDK是否生效

命令:java -version

如图-18所示

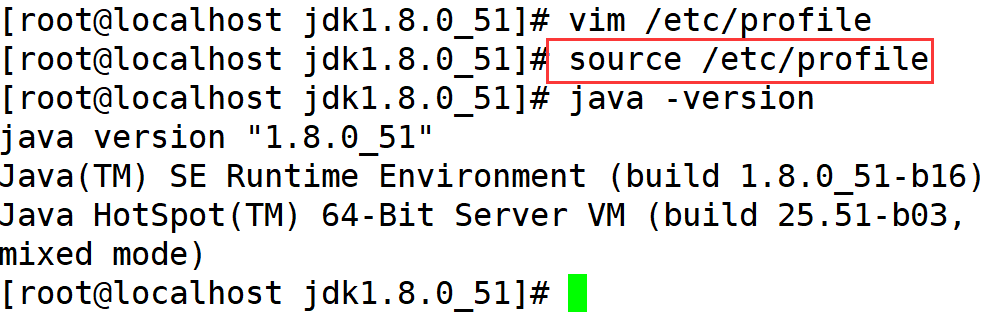


图- 18