# Nginx

## 业务需求

### 动态获取配置文件的值

1. 定义配置文件

image.localPath=E:/jt-upload/

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

1. 引入外部配置文件

<bean class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">

<property name="locations">

<!--Resource[] locations; -->

<list>

<value>classpath:/property/jdbc.properties</value>

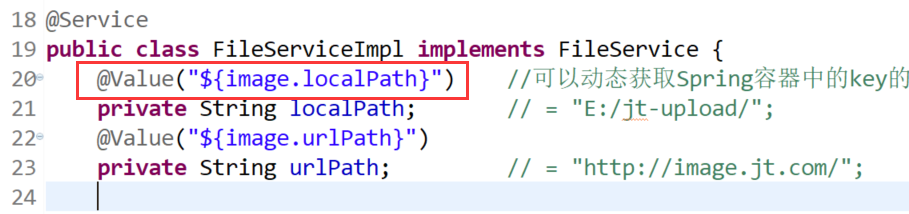
<value>classpath:/property/imageurl.properties</value>

</list>

</property>

</bean>

1. 动态取值



## 实现域名反向代理

### 业务需求

说明:要求用户访问manage.jt.com时,访问后台服务器.

实现规则:使用nginx的反向代理技术

实现步骤:

1. 修改hosts文件
2. 修改nginx配置文件

#后台管理系统

server {

listen 80;

server\_name manage.jt.com;

location / {

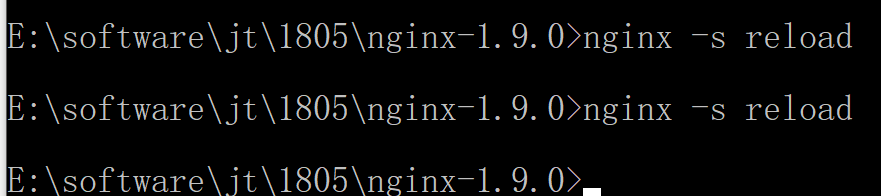
#实现服务器代理

proxy\_pass http://localhost:8091;

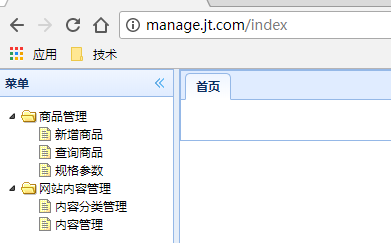
}

}

1. 重启nginx



1. 项目访问

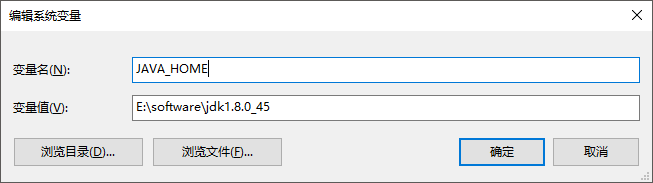


# Nginx高级应用

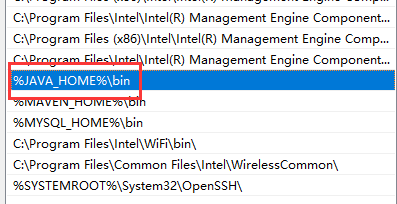
## 搭建tomcat集群

### 配置JDK

1. 配置环境变量

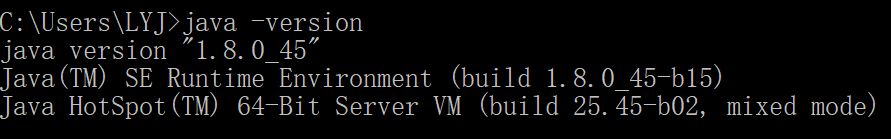


1. 配置变量添加path



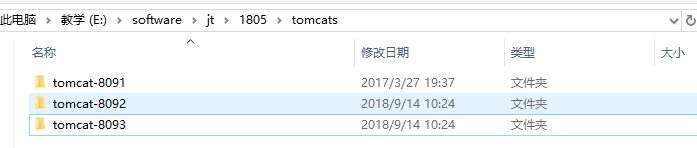
Win7: ;%JAVA\_HOME%/bin

1. 检查JDK

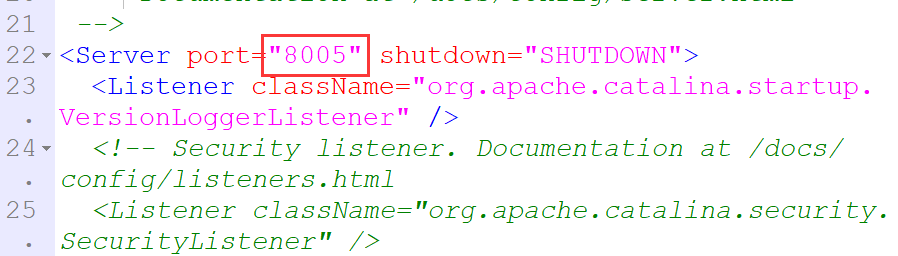


### 部署tomcat集群

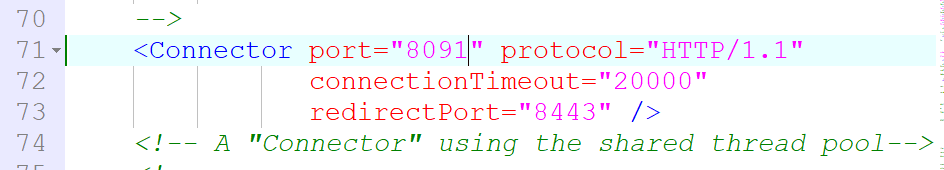
1. 准备3台tomcat



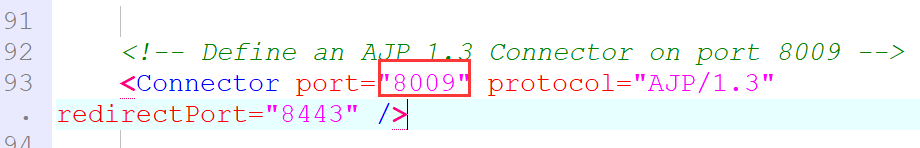
1. 修改配置-8005端口



1. 修改配置-8080端口



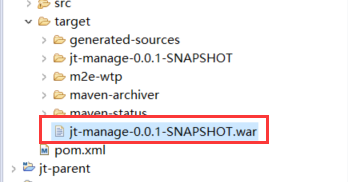
1. 修改配置-8009端口



### 项目准备

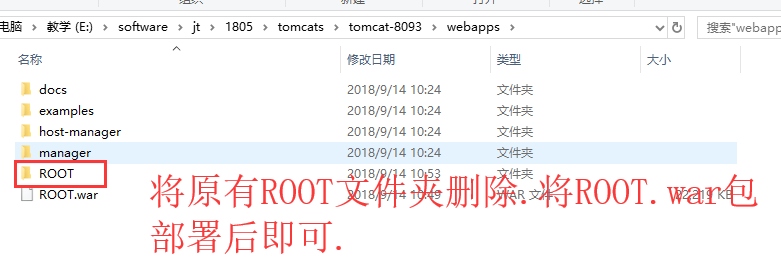
1. 将项目打包

打包顺序:jt-parent---jt-common—jt-manage

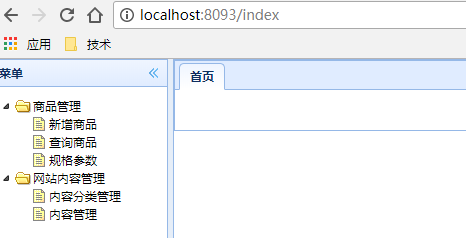


修改项目war包名称ROOT.war

1. 将war包部署到tomcat中.

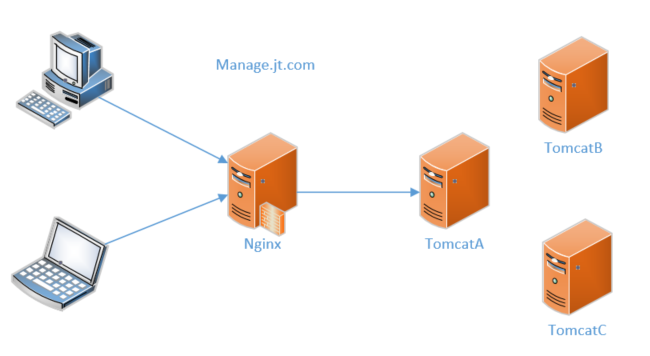


1. 项目访问



## 负载均衡策略

### 架构图设计



说明:根据nginx的配置,可以实现负载均衡.通过搭建tomcat服务器集群.可以极大的提高网站的并发量.

### 轮询

说明:根据配置文件的顺序.依次访问tomcat服务器.

配置策略:

#负载均衡配置策略 轮询

upstream jt {

server localhost:8091;

server localhost:8092;

server localhost:8093;

}

#后台管理系统

server {

listen 80;

server\_name manage.jt.com;

location / {

#实现服务器代理

#proxy\_pass http://localhost:8091;

proxy\_pass http://jt;

}

}

### 权重策略

说明:因为公司的服务器的处理性能是由差别的,如果在采用轮询策略时,会导致服务器负载不均,长时间负载不均会导致高负载机器有宕机的风险.

采用权重配置:

#负载均衡配置策略 轮询

upstream jt {

server localhost:8091 weight=6;

server localhost:8092 weight=3;

server localhost:8093 weight=1;

}

### IP\_hash策略

1.需求说明:因为现在服务器采用集群的部署方式,对于需要权限的操作,需要用户登录后进行.但是集群中的tomcat没法实现Session共享.导致用户重复登录.如何解决????

2.解决策略:

url重写技术. 将用户信息拼接到http请求中 效率太低.

IP\_hash

3.IP\_Hash说明:

根据用户的IP地址进行hash运算.将计算的结果绑定其中某一台服务器.

4.配置策略:

IP\_Hash的级别要比轮询和权重高.配置后,其他策略将不生效.

#负载均衡配置策略 轮询

upstream jt {

**ip\_hash;**

server localhost:8091 weight=6;

server localhost:8092 weight=3;

server localhost:8093 weight=1;

}

5.专业术语:

**Session黏着**

6.IP\_hash中问题:

1.由于ip地址的原因.经过hash值计算.可能会导致负载不均.

2.如果IP地址绑定的服务器宕机,那么很多的用户访问受限.

3.安全性不好.

7.改进方式:

新单点登录(SSO策略)

### 备用机机制

说明:为了让nginx出现问题后,能够继续提供服务.采用备用机制.正常情况下备用不起作用.当主机遇忙时或者主机宕机时.这时备用机生效.

#负载均衡配置策略 轮询

upstream jt {

#ip\_hash;

server localhost:8091 weight=6;

server localhost:8092 weight=3 down;

server localhost:8093 weight=1 backup;

}

### 项目上线步骤

1. 将项目进行打包.XXXX.war包
2. 根据服务器数量.分批上线
3. 需要将上线的服务器在nginx中先做下线处理down
4. 将项目进行部署.启动服务器/经过项目测试后,之后上线.

一般上线晚上8-5

### Nginx中健康检测

1. 设定超时时间

说明:当nginx实现反向代理时,如果后台服务器没有及时的响应,那么必须等到超时间后.访问下一台服务器.但是超时时间默认3分钟.需要手动修改超时时间.

proxy\_connect\_timeout 3;

proxy\_read\_timeout 3;

proxy\_send\_timeout 3;

1. 设定超时时间的优点

当访问后台的服务器.如果到了指定的超时时间,那么则自动访问下一台服务器.

缺点:

因为配置权重或者轮询.定期都会连接故障机.导致用户查询效率低.

1. 检测检测实现

说明:

当在指定的周期内,如果发现后台服务器不能正常访问.那么在该周期内不会再将请求发往该故障机.直到下个周期再次检测.

配置:

max\_fails=1 在指定的周期内,检测失败的次数

fail\_timeout=60s; 代表周期时间 单位秒

upstream jt {

#ip\_hash;

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

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

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

}

# 项目Linux部署

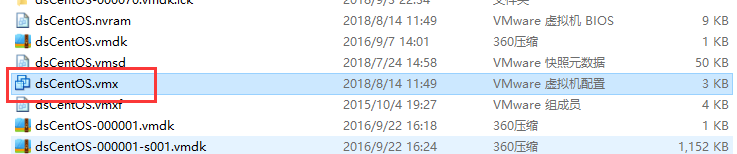
## 虚拟机

### 虚拟机介绍

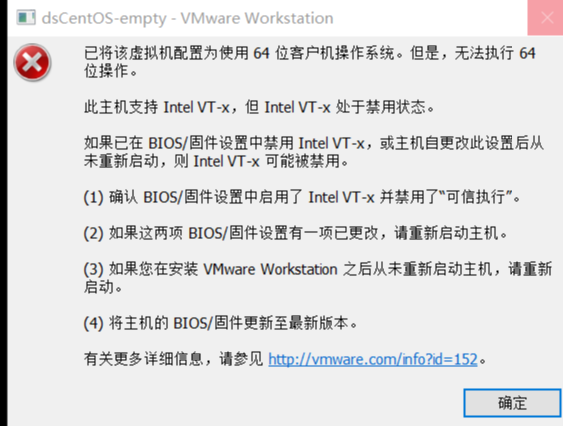
虚拟系统通过生成现有[操作系统](https://baike.baidu.com/item/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F/192)的全新虚拟[镜像](https://baike.baidu.com/item/%E9%95%9C%E5%83%8F/1574)，它具有真实[windows](https://baike.baidu.com/item/windows)系统完全一样的功能，进入虚拟系统后，所有操作都是在这个全新的独立的虚拟系统里面进行，可以独立安装运行[软件](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6/12053)，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" \t "_blank)，[VirtualBox](https://baike.baidu.com/item/VirtualBox" \t "_blank)，[Virtual pc](https://baike.baidu.com/item/Virtual%20pc" \t "_blank)）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" \t "_blank)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" \t "_blank)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。

### 虚拟机使用

1. Linux系统虚拟机使用,以虚拟机方式打开

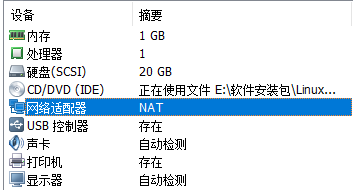


1. 虚拟化设置 进入bios系统中修改虚拟化设置即可



### 虚拟机环境介绍

1. 虚拟机设置



以上的配置,当虚拟机启动时,会向操作系统**割裂**对应的资源.

### 虚拟机网络配置

1. 桥接模式

直接连接物理网络. (直接插网线).

使用桥接模式后,虚拟机相当于在当前局域网中的一台机器.独占IP

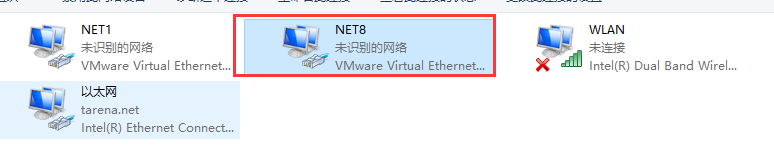
192.168.1.100,192.168.1.101 通信靠 路由器/交换机.

特点:在同一个局域网中的用户,可以互相通信.

1. NAT模式

相当于在PC机中开辟了一块网络空间.该空间只允许PC机自己访问.其他机器拒绝连接.

该模式必须有网卡的支持.

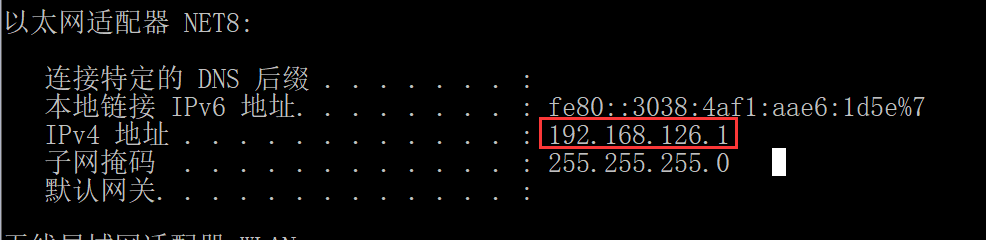


启动机器检测IP地址:

命令:ifconfig

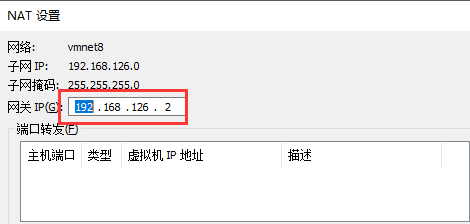
**192.168.126.160**

检查网卡配置

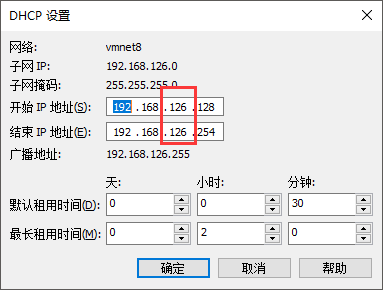


### 修改网卡配置

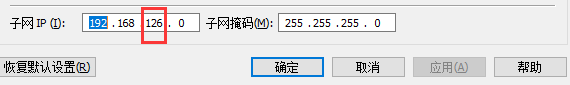
1.修改为网卡的网段 只改126



2.DHCP



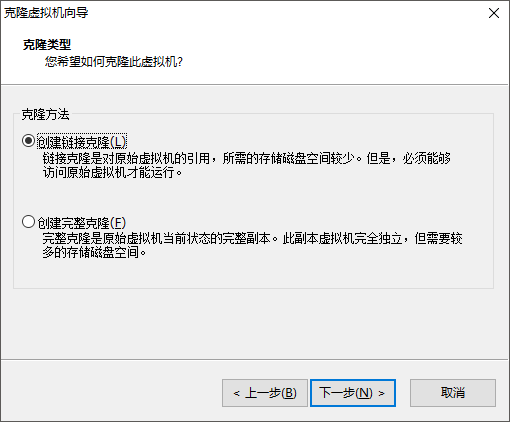
3.



修改完成后重启虚拟机

### 克隆虚拟机

说明:克隆虚拟机目的是为了防止误操作.损坏原有Linux系统.导致重新安装



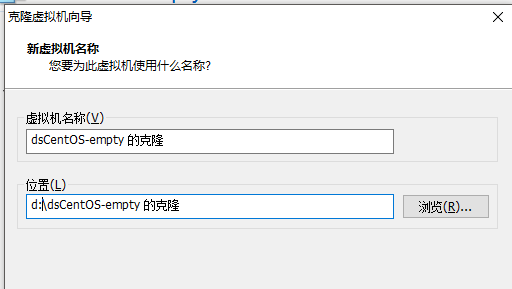
1. 创建链接克隆:

优点:占用磁盘少

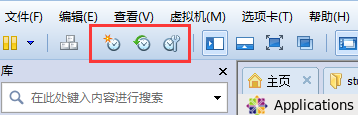
缺点:必须依赖原有虚拟机

1. 创建完整克隆

将原有文件全部复制,所占用空间较大



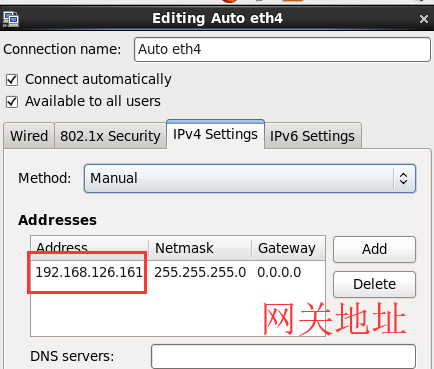
### 拍摄快照



1. 创建快照
2. 回退快照
3. 快照管理



### 设定固定IP



## Xshell安装

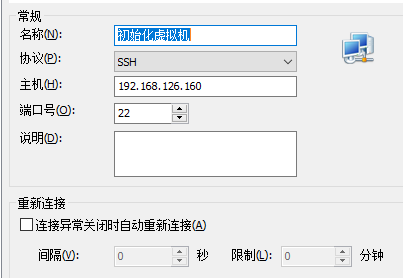
### Xshell介绍

Xshell [1]  是一个强大的安全终端模拟软件，它支持SSH1, SSH2, 以及Microsoft Windows 平台的TELNET 协议。**Xshell 通过互联网到远程[主机](https://baike.baidu.com/item/%E4%B8%BB%E6%9C%BA/455151" \t "_blank)的安全连接**以及它创新性的设计和特色帮助用户在复杂的网络环境中享受他们的工作。

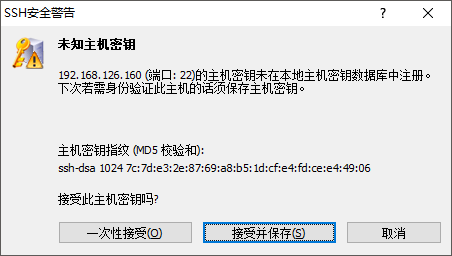
Xshell可以在Windows界面下用来访问远端不同系统下的服务器，从而比较好的达到远程控制终端的目的。除此之外，其还有丰富的外观配色方案以及样式选择。

### XShell使用

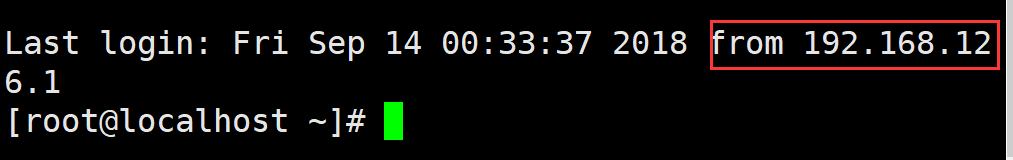
1. 创建连接



2.保存秘钥



3.远程连接



## Linux介绍

### Linux特点

1.Linux运行全部基于文件 /

2.Linux 安全性更好

3.Linux占用内核更小

4.运行更加的稳定. 定期维护 一周维护.

### Linux版本



### Linux命令

1.cd / 回到根目录

2.ls 显示文件夹

3.ll 显示文件详细信息

4.clear 清屏

5.pwd 查看当前文件夹的路径

6.mkdir 创建文件夹

mkdir aa bb cc

7.mv 移动文件/修改文件名称

mv cc ccc

8.rm 删除文件

rm aa.txt

rm -rf 任意文件 强制删除

9.vim aa.txt 创建编辑文件 aa.txt

I:插入

Esc: 退出编译状态

:wq 保存退出

U:恢复之前状态

10.cat aa.txt 查看文件内容

11.cp aa.txt bb.txt 复制文件

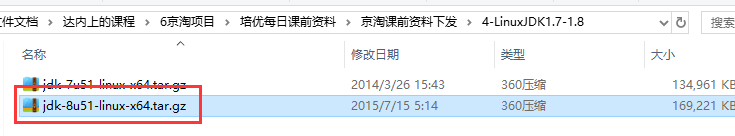
cp -r aa cc 递归复制文件夹

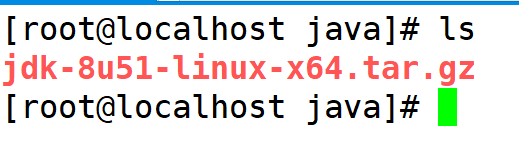
## 安装JDK环境

### 创建文件夹

说明:在/usr/local/src下创建文件java,之后将JDK安装包上传到该文件.

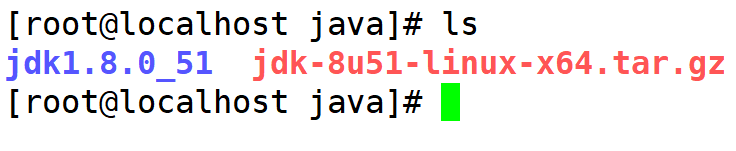
添加JDK文件:





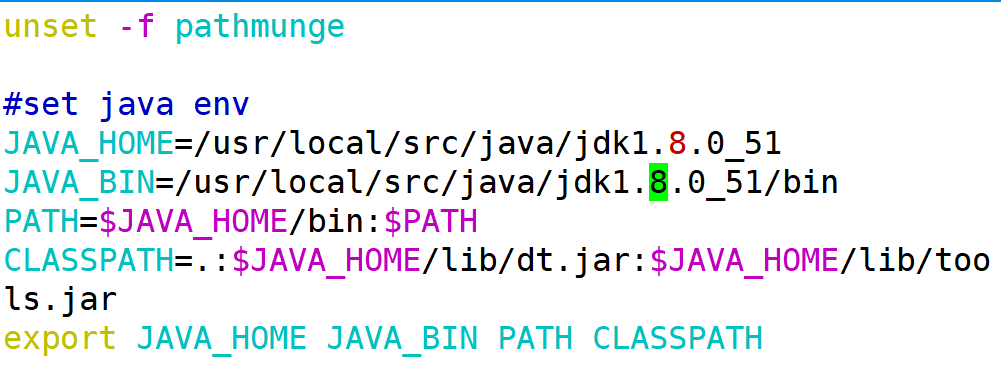
### 解压JDK

命令: tar -xvf jdk-8u51-linux-x64.tar.gz



### 配置环境变量

vim /etc/profile



#set java env

JAVA\_HOME=/usr/local/src/java/jdk1.8.0\_51

JAVA\_BIN=/usr/local/src/java/jdk1.8.0\_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

让JDK环境变量配置生效

source /etc/profile

