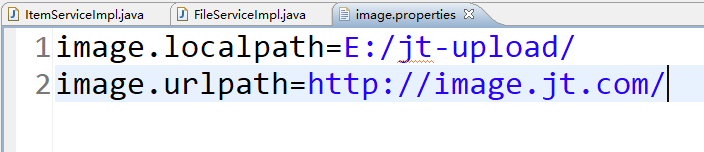
# 反向代理

## 图片回显

### 编辑pro文件



### 编辑Service

@Service

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

//该数据从spring容器中动态获取数据

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

**private** String localPath; //"E:/jt-upload/";

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

**private** String urlPath;

/\*\*

\* 文件上传实现思路

\* 1.校验文件类型 jpg|png|gif....

\* 2.校验是否为恶意程序

\* 3.为了防止图片检索速度慢,采用分文件存储 yyyy/MM/dd/

\* 4.防止文件重名 UUID + 随机数(3)

\* 5.实现文件上传

\*/

@Override

**public** PicUploadResult upload(MultipartFile uploadFile) {

PicUploadResult result = **new** PicUploadResult();

//1.获取文件名称 abc.jpg

String fileName = uploadFile.getOriginalFilename();

//将字符全部小写

fileName = fileName.toLowerCase();

**if**(!fileName.matches("^.+\\.(jpg|png|gif)$")) {

result.setError(1); //不是图片

**return** result;

}

//2.校验文件是否为恶意程序

**try** {

BufferedImage image =

ImageIO.*read*(uploadFile.getInputStream());

**int** width = image.getWidth();

**int** height = image.getHeight();

**if**(width==0 || height==0) {

result.setError(1); //一定不是图片

**return** result;

}

//3.实现分文件存储

String dateDir =

**new** SimpleDateFormat("yyyy/MM/dd")

.format(**new** Date());

// E:/jt-upload/2019/01/29

String localPathDir = localPath + dateDir;

//判断文件夹是否存在

File fileDir = **new** File(localPathDir);

**if**(!fileDir.exists()) {

fileDir.mkdirs(); //创建文件夹

}

//4.定义文件名称

String uuid = UUID.*randomUUID*()

.toString()

.replace("-","");

**int** random = **new** Random().nextInt(1000);

String fileType =

fileName.substring

(fileName.lastIndexOf("."));

//形成文件名称

String realName = uuid + random + fileType;

//E:/jt-upload/2019/01/29/abc.jpg

String localPathReal = localPathDir + "/" + realName;

//实现文件上传

uploadFile.transferTo(**new** File(localPathReal));

//定义url

String url = urlPath + dateDir + "/" + realName;

result.setUrl(url);

} **catch** (Exception e) {

e.printStackTrace();

result.setError(1); //文件上传失败

**return** result;

}

**return** result;

}

}

### 问题描述

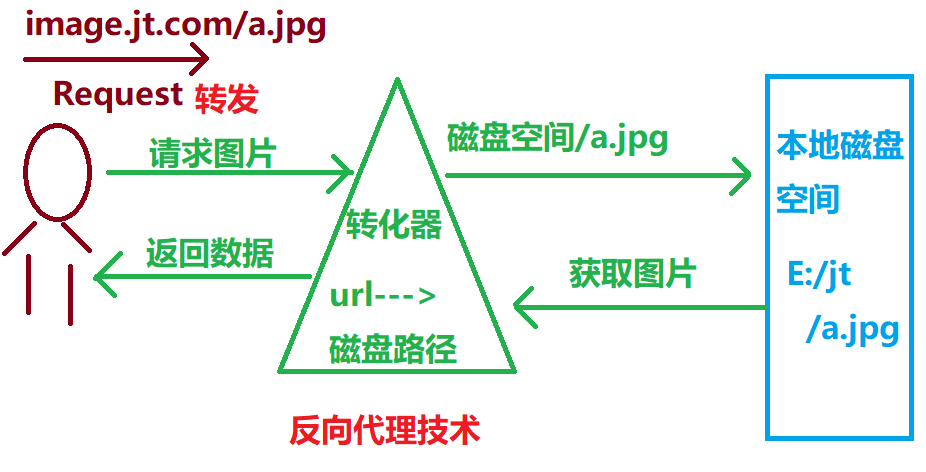
虚拟空间地址:

**http://image.jt.com**/2019/01/30/2f8ac03b372f445ca772060366e55ae9909.jpg

真实的磁盘路径:

**E:\jt-upload**\2019\01\30\2f8ac03b372f445ca772060366e55ae9909.jpg

### 反向代理



1. 用户根据url地址访问图片信息.
2. 转化器根据特定的配置,将请求路径进行转化.之后代替用户发起请求.访问真实的数据.
3. 访问数据后,获取真实的数据,将数据返回给转化器.
4. 由转化器将数据最终返回给用户.

概括:由第三方代替用户访问数据的过程. (请求的转发)

## Nginx

### Nginx介绍

*Nginx* (engine x) 是一个高性能的[HTTP](https://baike.baidu.com/item/HTTP" \t "_blank)和[反向代理](https://baike.baidu.com/item/%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86/7793488" \t "_blank)服务，也是一个IMAP/POP3/SMTP服务。Nginx是由伊戈尔·赛索耶夫为[俄罗斯](https://baike.baidu.com/item/%E4%BF%84%E7%BD%97%E6%96%AF/125568" \t "_blank)访问量第二的Rambler.ru站点（俄文：Рамблер）开发的，第一个公开版本0.1.0发布于2004年10月4日。

其将[源代码](https://baike.baidu.com/item/%E6%BA%90%E4%BB%A3%E7%A0%81)以类BSD许可证的形式发布，因它的稳定性、丰富的功能集、示例配置文件和低系统资源的消耗而[闻名](https://baike.baidu.com/item/%E9%97%BB%E5%90%8D/2303308)。2011年6月1日，nginx 1.0.4发布。

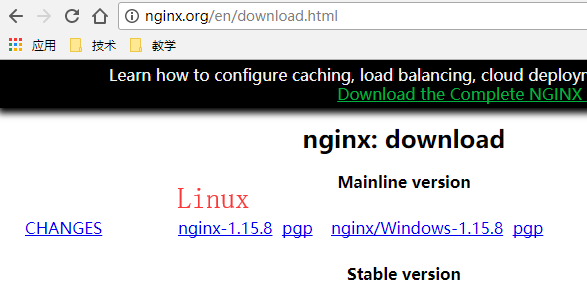
Nginx是一款[轻量级](https://baike.baidu.com/item/%E8%BD%BB%E9%87%8F%E7%BA%A7/10002835)的[Web](https://baike.baidu.com/item/Web/150564) 服务器/[反向代理](https://baike.baidu.com/item/%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86/7793488)服务器及[电子邮件](https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E9%82%AE%E4%BB%B6/111106)（IMAP/POP3）代理服务器，并在一个BSD-like 协议下发行。其特点是**占有内存少**，[**并发**](https://baike.baidu.com/item/%E5%B9%B6%E5%8F%91/11024806)能力强，事实上nginx的并发能力确实在同类型的网页服务器中表现较好，中国大陆使用nginx网站用户有：百度、[京东](https://baike.baidu.com/item/%E4%BA%AC%E4%B8%9C/210931)、[新浪](https://baike.baidu.com/item/%E6%96%B0%E6%B5%AA/125692)、[网易](https://baike.baidu.com/item/%E7%BD%91%E6%98%93/185754)、[腾讯](https://baike.baidu.com/item/%E8%85%BE%E8%AE%AF/112204)、[淘宝](https://baike.baidu.com/item/%E6%B7%98%E5%AE%9D/145661)等。

并发能力: 5万/秒 实际3万/秒 底层开发语言C语言

占用内存: 3-4M

### Nginx下载

网址: <http://nginx.org/en/download.html>



### Nginx命令

1. 启动

以超级管理员权限运行

1. 关于2个进程描述

Nginx启动时,会启动2个进程.一个是主进程/一个是守护进程.

主进程:主要负责实现业务功能的

守护进程:否则主进程意外关闭.

1. Nginx启动命令

前提:执行命令,必须在exe所在的根目录中运行.

Nginx启动占用80端口.

计算机生成了可选文字:
C：\Users\LYJnetstat
¯aon
1n
str
TCP
5680
TCP
5680
·：]：8093
LISTENING
LISTENING
C：\Users\LYJtaskkill/f/pid5680
成功：己终止
C：\Users\LYJ>

启动: start nginx

重启: nginx -s reload

关闭: nginx -s stop

### Nginx入门案例

server {

listen 80;

server\_name localhost;

#charset koi8-r;

#access\_log logs/host.access.log main;

location / {

#root 代表文件夹 /开头 绝对

root html;

#默认跳转首页

index index.html index.htm;

}

}

## Nginx实现图片回显

### 修改HOSTS文件

说明:本机中的HOSTS文件定义了域名与IP的映射关系.

路径: C:\Windows\System32\drivers\etc

文件内容:

127.0.0.1 image.jt.com

127.0.0.1 manage.jt.com

127.0.0.1 www.jt.com

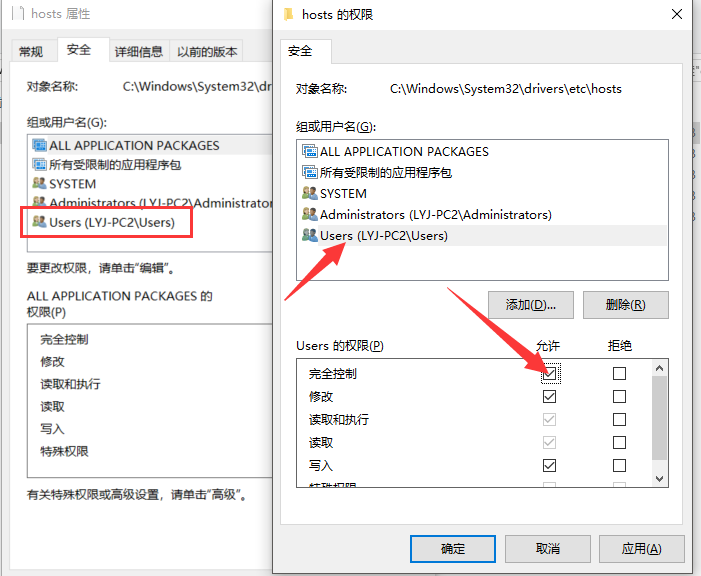
127.0.0.1 sso.jt.com

127.0.0.1 cart.jt.com

127.0.0.1 order.jt.com

127.0.0.1 solr.jt.com

192.168.126.148 www.easymall.com



修改完成后保存

### 修改Nginx配置文件

#编辑图片服务器

server {

listen 80;

server\_name image.jt.com;

location / {

#拦截image.jt.com实现反向代理

root E:/jt-upload;

}

}

修改完成后,重启Nginx

nginx -s reload

## 反向代理-域名

### 需求

访问京淘后台需要通过域名 http://manage.jt.com/index访问首页

### 编辑HOSTS文件

127.0.0.1 manage.jt.com

### 编辑nginx配置文件

#编辑后台管理服务器

server {

listen 80;

server\_name manage.jt.com;

location / {

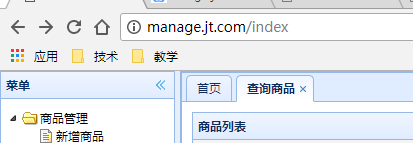
#反向代理到url

proxy\_pass http://localhost:8091;

}

}

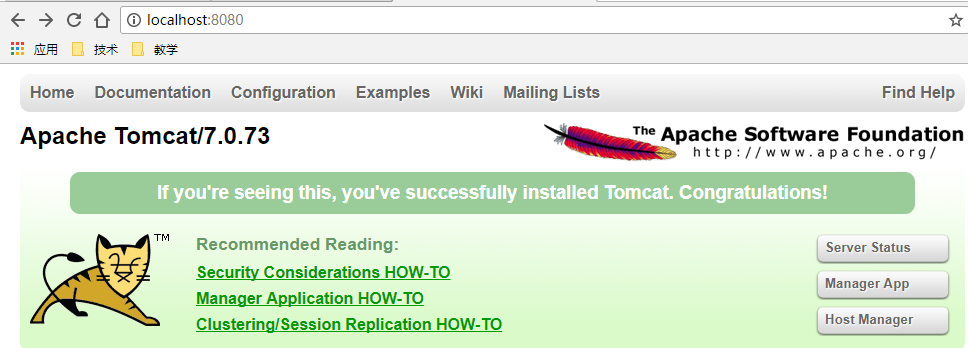
### 页面效果



## Nginx负载均衡

### 准备多台tomcat

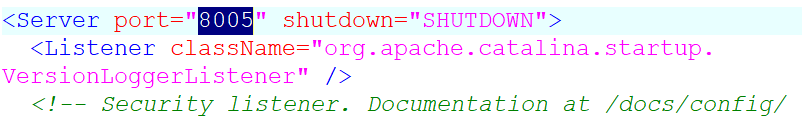
1. 复制tomcat服务器.并且执行start.bat命令.检查tomcat服务器是否正常运行.
2. 运行结果



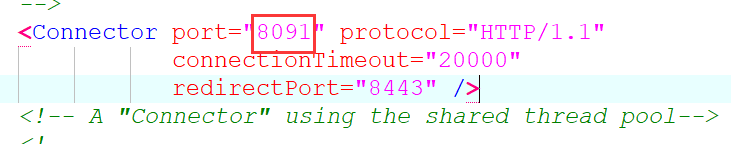
1. 如果运行不了检查 java-home必配/catalina配置(移除)

### 修改端口

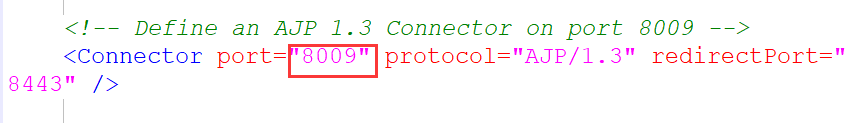
1.修改8005端口



2.修改8080端口

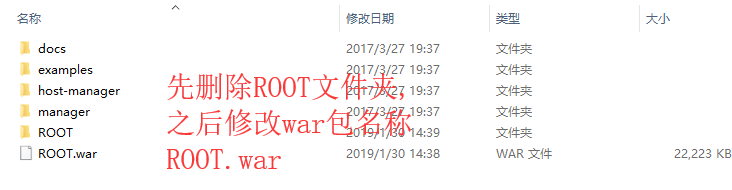


3.修改8009端口

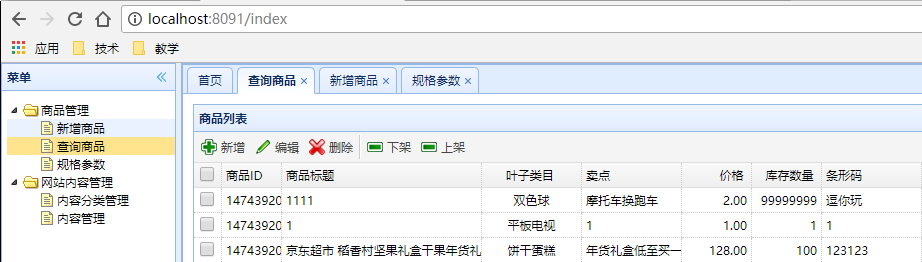


### 项目启动测试

1. 将项目maven-clean-🡪clean---maven-install,之后将项目发布到tomcat服务器中.



1. 启动测试



### 轮询策略

说明:根据配置文件中配置tomcat的顺序.依次访问tomcat.

配置说明:

#配置负载均衡策略

upstream jt {

server 127.0.0.1:8091;

server 127.0.0.1:8092;

server 127.0.0.1:8093;

}

#编辑后台管理服务器

server {

listen 80;

server\_name manage.jt.com;

location / {

#反向代理到url

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

proxy\_pass http://jt;

}

}

### 权重策略

说明:可以根据服务器处理能力的不同,为高性能的处理器.多分配请求.通过数字代表权重的大小

配置:

#配置负载均衡策略 默认策略:轮询

upstream jt {

server 127.0.0.1:8091 weight=6;

server 127.0.0.1:8092 weight=3;

server 127.0.0.1:8093 weight=1;

}

### IP\_HASH策略

问题:因为配置负载均衡策略后,有多台tomcat共同为用户提供服务,如何实现用户的数据共享呢???

因为tomcat之间不能共享Session .

策略A:COOKIE+拦截器(常用策略)

策略介绍:

根据用户访问的IP地址+算法最终确定唯一一台tomcat服务器!!!

#配置负载均衡策略 默认策略:轮询 IPhass#绑定服务器访问

upstream jt {

**ip\_hash;**

server 127.0.0.1:8091 weight=6;

server 127.0.0.1:8092 weight=3;

server 127.0.0.1:8093 weight=1;

}

缺点:

1. 使用IPHash容易造成负载不均现象(数据倾斜)
2. 使用IPhash时,如果服务器宕机,则直接影响用户的使用

### 服务优化策略

1. down属性

说明:如果某台服务器宕机,则可以通过down属性,标识该服务器已经宕机.则nginx不会再将请求发往故障机!

1. backup属性

说明:正常情况下,请求不会发往backup标识的备用机.但是当主机全部遇忙或者宕机时,备用机生效!

#配置负载均衡策略 默认策略:轮询 IPhass#绑定服务器访问

upstream jt {

#ip\_hash;

server 127.0.0.1:8091 weight=6 down;

server 127.0.0.1:8092 weight=3 down;

server 127.0.0.1:8093 weight=1 backup;

}

1. 配置超时策略

设定超时时间.当用户在指定的时间内,不能正确的访问服务器.,则会自动的跳转到下一台.

#编辑后台管理服务器

server {

listen 80;

server\_name manage.jt.com;

location / {

#反向代理到url

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

proxy\_pass http://jt;

#配置超时 单位秒

**proxy\_connect\_timeout 3;**

**proxy\_read\_timeout 3;**

**proxy\_send\_timeout 3;**

}

}

### Nginx健康检测

说明:通过nginx健康检测,在一定的时间范围内(周期),如果这台服务器不能正常访问.则在该周期内,不会再将请求发往该机器.一直到下一个周期.

server localhost:8090 max\_fails=1 fail\_timeout=60s;

max\_fails:最大的失败次数.

fail\_timeout=60s; 60秒内不允许访问该服务器

配置项:

#配置负载均衡策略 默认策略:轮询 IPhass#绑定服务器访问

upstream jt {

#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;

}

# 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" \t "_blank)[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" \t "_blank)的多用户、[多任务](https://baike.baidu.com/item/%E5%A4%9A%E4%BB%BB%E5%8A%A1/1011764" \t "_blank)、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" \t "_blank)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" \t "_blank)以[网络](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C/143243" \t "_blank)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" \t "_blank)。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" \t "_blank)、[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" \t "_blank)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

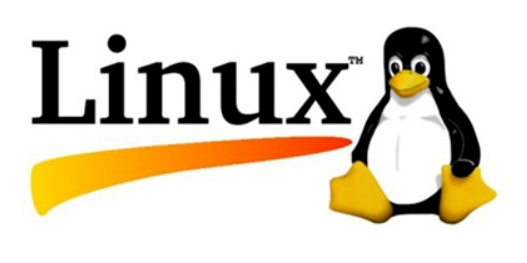


图- 9

### 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

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 –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

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" \t "_blank)，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" \t "_blank)。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

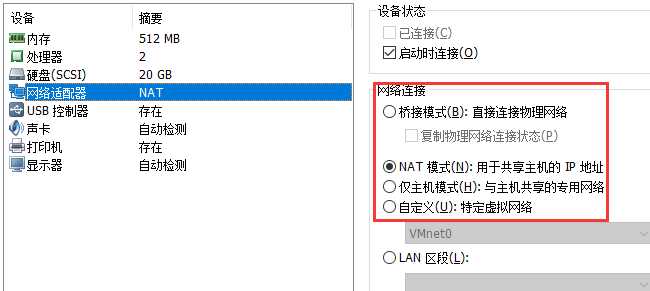


图- 11

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

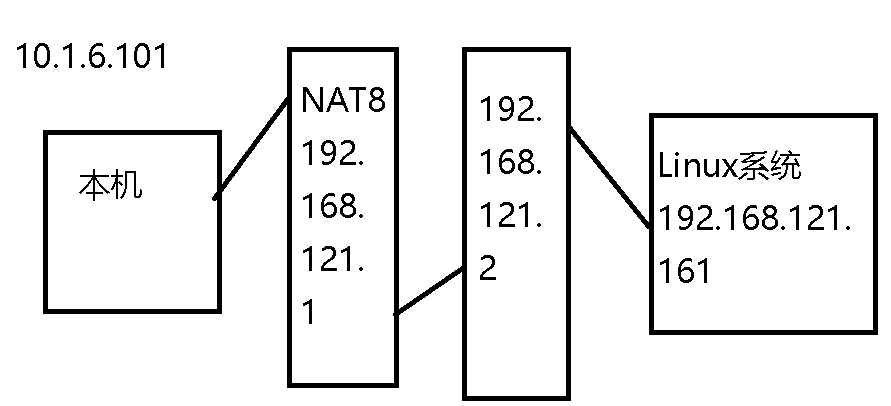


图- 12

## 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

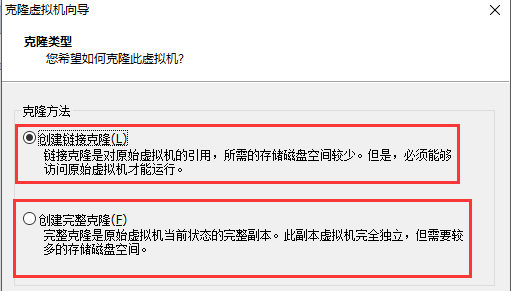


图- 13

如图-14所示

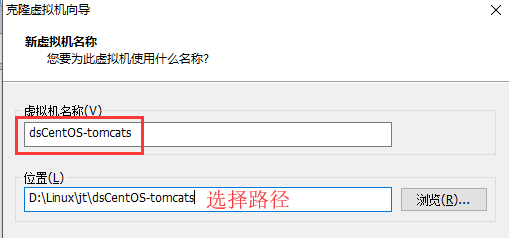


图- 14

## 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

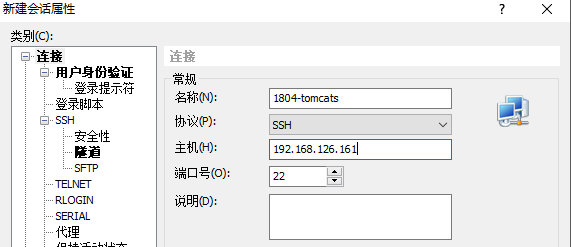


图- 15

## 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

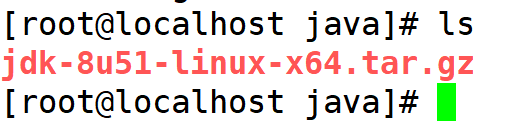


图- 16

### 解压JDK文件

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

如图-17所示

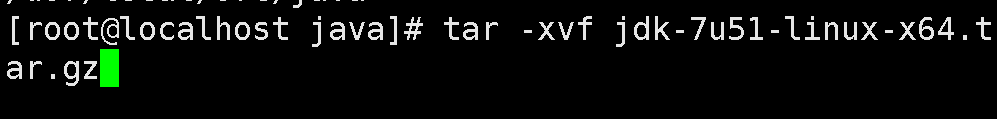


图- 17

### 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

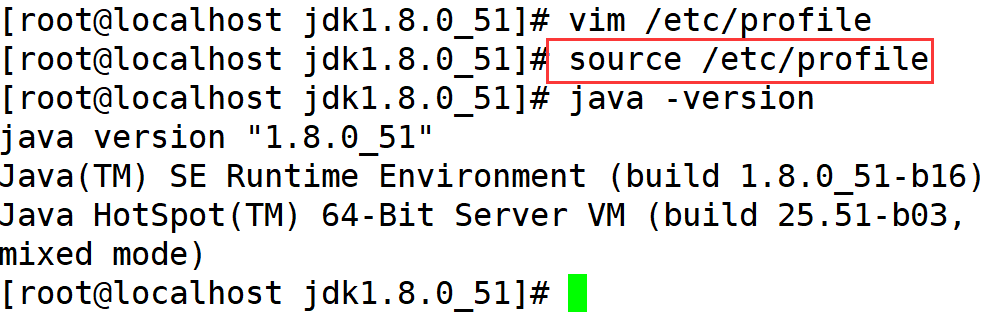


图- 18