# Nginx高级

## Nginx反向代理

### 实现域名反向代理

要求:manage.jt.com/index时,要求用户访问localhost:8091/index

#配置后台管理服务器

server {

listen 80;

server\_name manage.jt.com;

location / {

#发起请求

proxy\_pass http://localhost:8091;

}

}

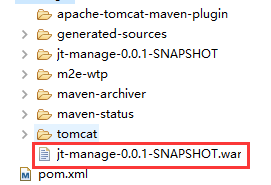
2.修改hosts文件



## 搭建tomcat集群

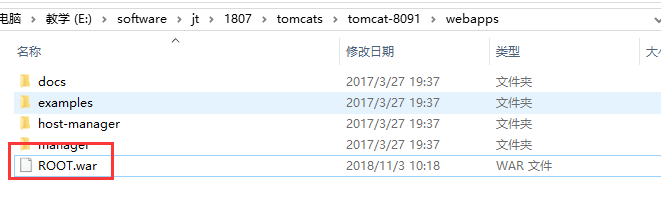
### 将项目打包

说明:将jt-parent/jt-common/jt-manage分别打包



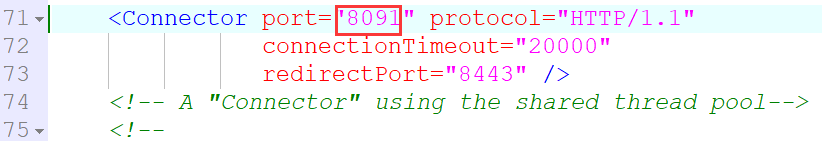
### 部署tomcat服务器

1. 部署jar包

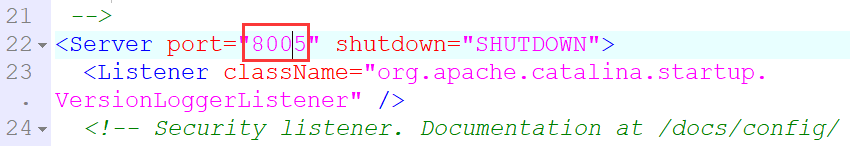


将jt-manage项目打包后,修改名称为ROOT.war.之后部署

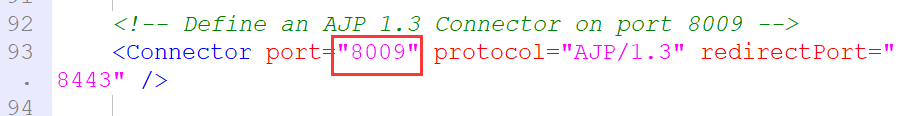
1. 修改端口号
2. 修改8080服务端口



1. 修改8005端口



1. 修改8009端口



测试成功后,部署8092/8093服务器.

## Nginx实现负载均衡

### 轮询方式

特点:根据配置文件的顺序,依次访问tomcat服务器.

Nginx配置:

#配置负载均衡策略 1轮询

upstream jt {

server localhost:8091;

server localhost:8092;

server localhost:8093;

}

#配置后台管理服务器

server {

listen 80;

server\_name manage.jt.com;

location / {

#发起请求

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

proxy\_pass http://jt;

}

}

### 权重方式

说明:根据服务器处理能力的不同,采用权重的策略,处理能力强的服务器,尽可能多处理请求.

#配置负载均衡策略 1轮询 2.权重

upstream jt {

server localhost:8091 weight=6;

server localhost:8092 weight=3;

server localhost:8093 weight=1;

}

### IP\_hash

要求:如何实现用户数据共享???

说明:如果将用户信息数据保存到session域中,可能会造成用户频繁登陆的现象.

解决方案:

1. url重写技术

用户访问nginx <http://manage.jt.com?sessionId=100200300>

Nginx访问真实tomcat服务器

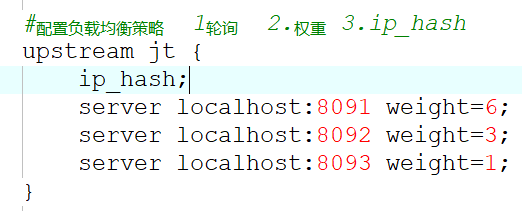
<http://localhost:8093?sessionId=100200300>

早期适用于该方案.但是由于效率过低,后期几乎不用.

特点:该方案可以解决Cookie禁用问题.

1. IP\_hash技术

特点:根据用户的Ip地址之后进行哈希运算.最终绑定到指定的服务器中.



IP\_hash存在的问题:

1. 由于绑定了具体某台服务器,可能会导致某些服务器的负载压力过高.
2. 如果某台服务器宕机,则直接影响访问该服务器的用户.

3.单点登录解决方案(SSO)

核心思想:主要用户登陆一次后,访问其他相关的系统时,无需再次登录.

变向的实现Session共享.

**SSO英文全称Single Sign On**，[单点登录](https://baike.baidu.com/item/%E5%8D%95%E7%82%B9%E7%99%BB%E5%BD%95/4940767)。SSO是在多个应用系统中，用户只需要登录一次就可以访问所有相互信任的应用系统。它包括可以将这次主要的登录映射到其他应用中用于同一个用户的登录的机制。它是目前比较流行的企业业务整合的解决方案之一。

### Nginx中下线机制

说明:如果nginx中部署tomcat服务器宕机,需要手动修改nginx配置文件.

#配置负载均衡策略 1轮询 2.权重 3.ip\_hash

upstream jt {

#ip\_hash;

server localhost:8091 weight=6 **down**;

server localhost:8092 weight=3;

server localhost:8093 weight=1;

}

### 备用机机制

说明:正常情况下,备用的服务器不会为用户提供服务.当主机遇忙时,或者主机全部宕机时.这时备用机采用生效.

#配置负载均衡策略 1轮询 2.权重 3.ip\_hash

upstream jt {

#ip\_hash;

server localhost:8091 weight=6 down;

server localhost:8092 weight=3 down;

server localhost:8093 weight=1 **backup**;

}

### Nginx中健康检测机制

max\_fails=1: 心跳检测时,服务没有正常响应的次数.

fail\_timeout=60s 如果出现宕机现象,则在60秒内,不会再把请求发往宕机的机器.当到了下一个周期时会重新进行心跳检测.如果检测通过,则会正常的访问服务器.

#配置负载均衡策略 1轮询 2.权重 3.ip\_hash

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;

}

配置超时时间:因为nginx默认的超时时间很久,所以需要修改nginx超时时间,当一个请求超过超时时间,则会访问下一台tomcat服务器.

#配置后台管理服务器

server {

listen 80;

server\_name manage.jt.com;

location / {

#发起请求

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

proxy\_pass http://jt;

**proxy\_connect\_timeout 3;**

**proxy\_read\_timeout 3;**

**proxy\_send\_timeout 3;**

}

}

### 项目上线步骤

1. 将项目先clean之后打包.
2. 规划分批部署的方式.
3. 修改nginx配置文件,将部署的服务器进行下线处理. Down,之后重启nginx
4. 关闭tomcat服务器.部署新的war包.
5. 重启tomcat服务器,经过测试无误后将nginx配置文件中的down去掉.

通过脚本文件,实施快速部署.

# Linux部署服务器

## Linux操作系统介绍

### 产品介绍



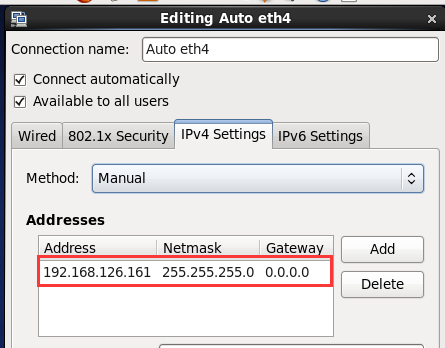
### Linux系统特点

1. Linux开源的
2. Linux运行最稳定的操作系统
3. Linux运行安全性更好
4. Linux操作系统运行内核小

概念:

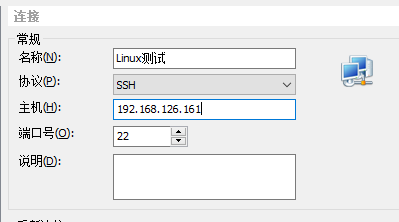
Linux操作系统内部全部都是基于文件的.只要有足够的权限可以删除任意的文件.(自杀)

### 配置虚拟机固定IP

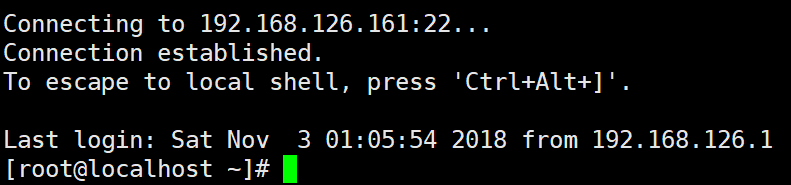


### 远程连接工具Xshell

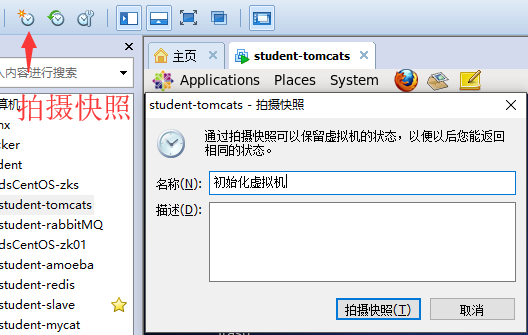
1. 配置连接 输入用户名和密码后关闭



1. Linux系统连接



### 虚拟机快照



### Linux命令

1.ifconfig 查看Ip地址

2.cd / 取到Linux操作系统的根目录

cd /usr 进入到指定目录下 绝对路径

cd usr 进入当前文件夹下的usr目录 相对路径

cd ../ 返回上一目录

3.ls 查询当前文件夹下的文件信息

ll 查询文件的详细信息

4.clear 清屏 /windows cls

5.pwd 查询当前文件所在路径

6.mkdir aa bb cc 同时创建多个文件夹

mkdir -p dd/ee/ff 可以实现递归创建

1. rm -rf 文件名称 删除文件
2. vim a.txt

i: 进入编辑状态

esc: 退出编辑状态

u : 表示撤销

:wq 表示保存退出.

1. cp 需要复制文件 新文件名称 复制单个文件

cp -r aa ee 复制文件夹