# 实现文件上传回显

## 业务分析

### 关于路径说明

1. 文件上传将文件保存到本地具体的磁盘空间中.但是该文件只能允许自己通过物理磁盘路径进行访问.
2. 为了让全部用户都能正常访问,所以必须准备一个**虚拟的网络路径**.要求用户通过虚拟网络路径能够正常的访问真实的图片信息.

**虚拟路径**~~~~~~~~~~**真实磁盘路径**

## 业务分析--如何实现

本地磁盘路径:

**D:\1-JT\images\2019\07\08\6accfc0ef0ef477683e4b92a2c5b755a.jpg**

准备虚拟路径:

**http://image.jt.com\2019\07\08\6accfc0ef0ef477683e4b92a2c5b755a.jpg**

说明:

要求本地磁盘根目录是固定的.与域名形成绑定的关系.图片之后的路径信息完全一致.

## 文件上传代码更新

### 编辑pro文件

说明:在原有基础之上拼接虚拟路径

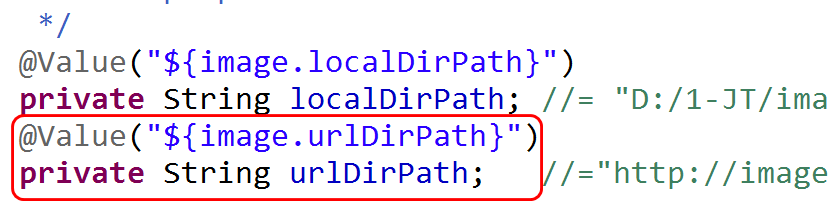
#key=value key value都是字符串

image.localDirPath=D:/1-JT/images/

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

### 编辑Service层

说明:在文件上传的service中添加数据注入



编辑Service实现类 完成虚拟路径映射

//8.编辑虚拟路径数据返回

//http://image.jt.com/yyyy/MM/dd/uuid.jpg

String realUrlPath = urlDirPath+datePathDir+"/"+realName;

ui\_Image.setUrl(realUrlPath);

### 文件上传测试

实现文件上传之后更改文件路径信息.将[www.jt.com](http://www.jt.com)替换为本地根目录检查图片实现是否正确.



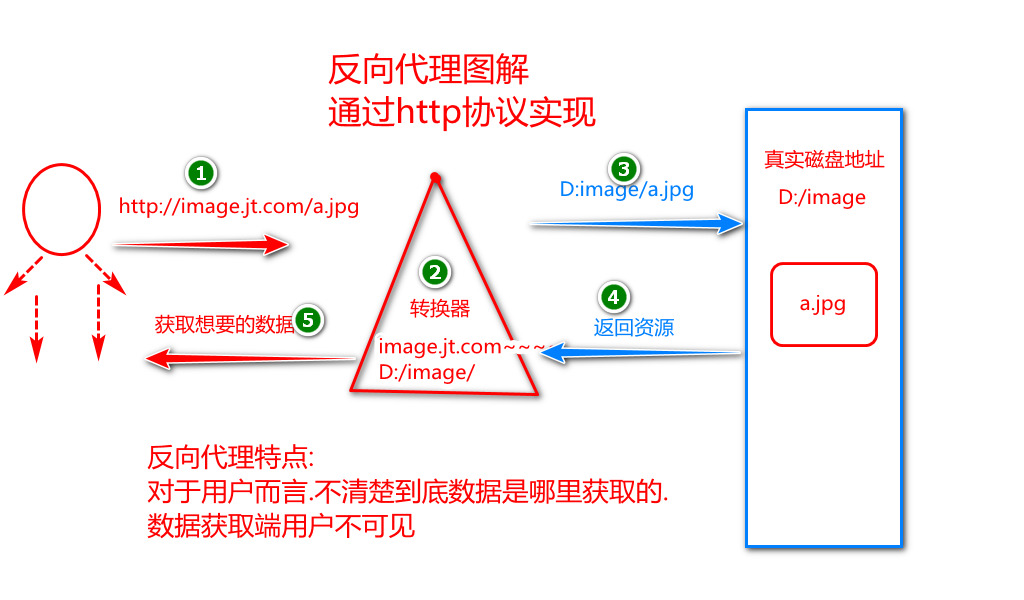
如果显示成功,则一切正确.如果显示不成功则说明路径拼接问题.检查代码

## 反向代理

### 概念

用户发起正常的url请求地址.经过代理.将路径进行转化.代替用户去请求真实资源的过程.称之为反向代理.

### 图解



## Nginx nginx安装使用

### Nginx介绍

*Nginx* (engine x) 是一个高性能的[HTTP](https://baike.baidu.com/item/HTTP)和[反向代理](https://baike.baidu.com/item/%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86/7793488)web服务器，同时也提供了IMAP/POP3/SMTP[服务](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1/100571)。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)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，在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。

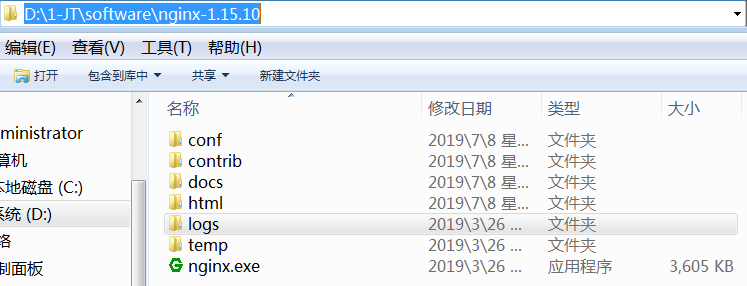
概述:Nginx是由俄罗斯人使用C语言负责开发的.是一个轻量级(内核小.占用内存少2M),并发能力强(3-5万/秒)的web服务器/**反向代理服务器**/电子邮件服务器.

### Nginx官网介绍



### Nginx安装

说明:将nginx安装文件导入本地磁盘要求,不要放置C盘/不要有中文路径.之后将文件解压.



注意事项:

1. nginx启动会占用**80**端口!!!!

如果启动异常 请检查tomcat环境变量配置.

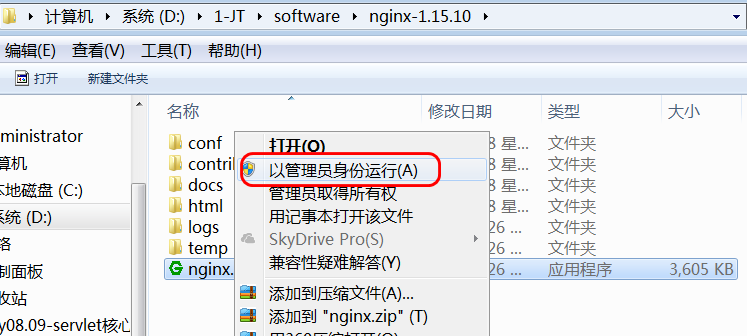
高级系统设置 path路径中检查tomcat是否配置.

2.Nginx只能启动一次.不能多次启动.如果启动多次.则nginx之前正常的启动将被破坏. 启动/关闭.

### nginx启动说明

1. 起步阶段

第一次启动时以超级管理员身份运行.目的获取权限.



1. 关于进程项说明

说明:每次启动nginx都会启动2个进程.

守护进程:防止主进程意外关闭.

主进程: nginx主要服务项.

手动关闭方式: 先关闭守护再关闭主进程.

### Nginx命令

前提:命令执行时需要在**nginx的根目录**中运行!!!!

1. 启动命令 start nginx
2. 关闭命令 nginx -s stop
3. 重启命令 nginx -s reload 必须先开再重启

启动测试:浏览器输入localhost:80检查效果.



## Nginx反向代理配置

### 入门案例

server {

listen 80; #要求监听80端口

server\_name localhost; #监听域名

#具体反向代理的配置信息

location / {

#root是关键字 代表文件夹

root html;

#index是关键字 代表系统默认访问页面

index index.html index.htm;

}

}

### 实现图片反向代理

需求:将域名为image.jt.com----转化为具体文件路径D:\1-JT\images

1.编辑nginx配置文件:

#1.图片服务器

server {

listen 80;

server\_name image.jt.com;

location / {

#代理路径

root D:/1-JT/images;

}

}

2.遇到问题

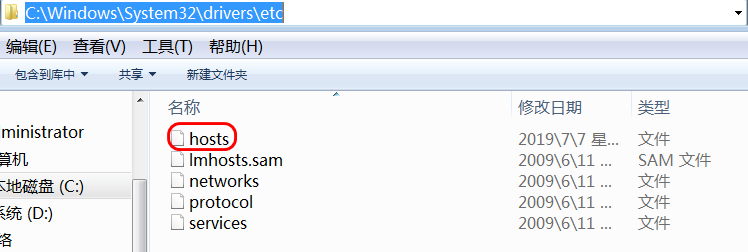
因为image.jt.com这个域名我们没有购买.如果只有本机自己测试时使用.则可以通过修改**hosts**文件实现该功能.

### 关于Hosts文件说明

说明:由于本机需要某些特定的环境作为测试.

则一般在hosts文件中配置**域名与IP**的映射关系.

文件路径:



修改hosts文件

#127.0.0.1 localhost

127.0.0.1 www.164.com

127.0.0.1 image.jt.com

127.0.0.1 manage.jt.com

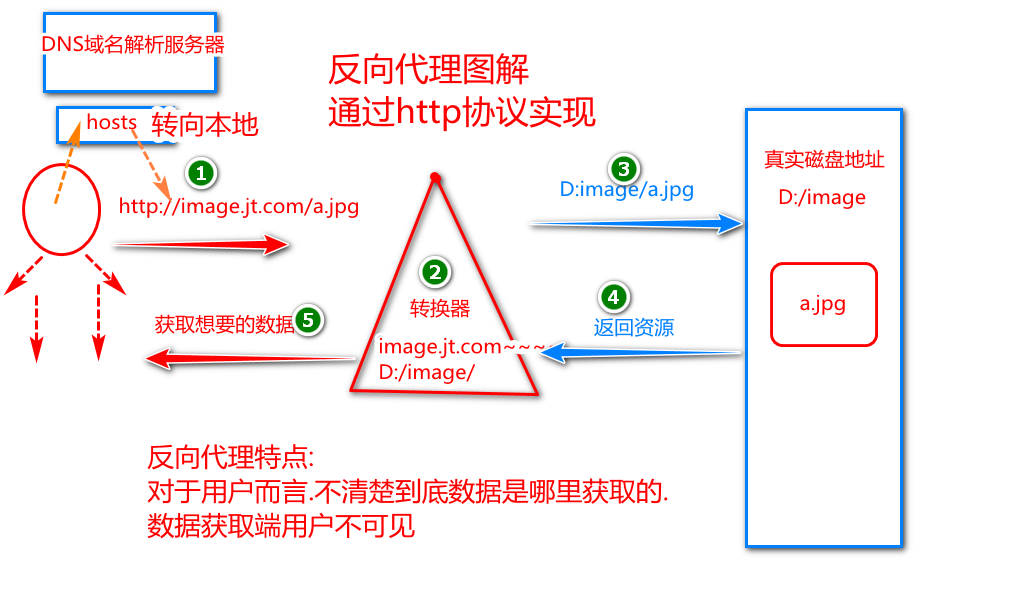
127.0.0.1 www.jt.com

127.0.0.1 sso.jt.com

127.0.0.1 cart.jt.com

127.0.0.1 order.jt.com

### hosts文件图解



## nginx实现域名代理

### 业务说明

当用户访问具体网址时例如www.jt.com/www.taobao.com.但是用户访问的服务器是具体的tomcat服务器.需要标识端口.部署多台tomcat服务器形成集群.这时需要使用nginx反向代理技术.

### 编辑配置文件

#配置后台管理服务器

server {

listen 80;

server\_name manage.jt.com;

location / {

#代理服务器路径

proxy\_pass http://localhost:8091;

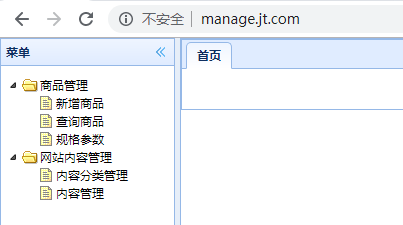
}

}

编辑完成之后,重启nginx服务器.

nginx -s reload

### 域名测试

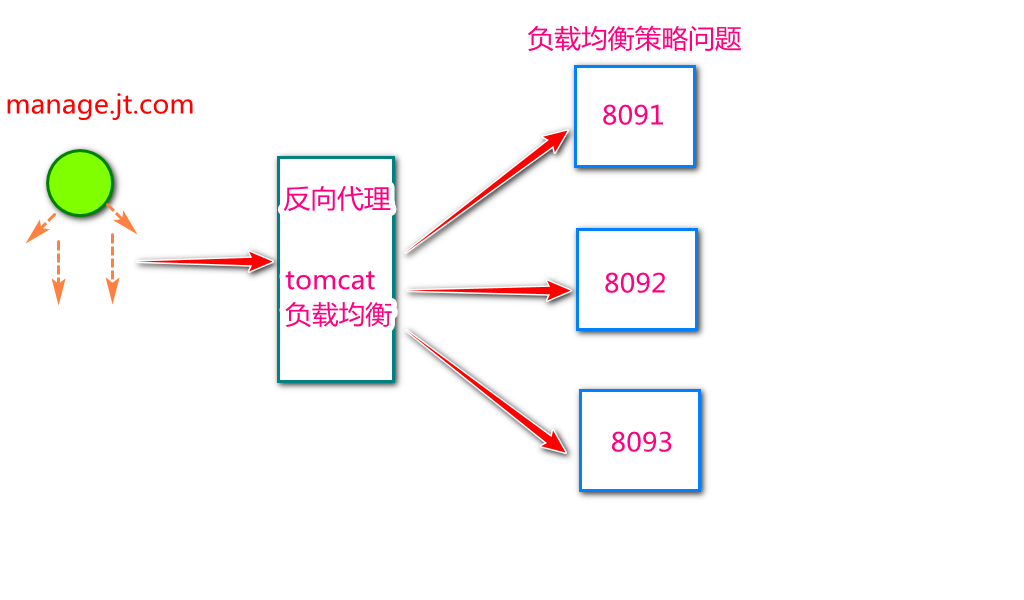


### 关于Nginx启动异常问题

1. 频繁启动导致出现配置不生效. 检查进程项 将多余服务停止.
2. nginx启动完成之后,最好执行一个重启命令.检查程序启动最终是否正确.
3. nginx没有效果. 检查文件的安装位置是否在C盘 是否有中文路径.

## Nginx实现负载均衡

### 图解



### 准备3台tomcat服务器

#### 编辑indexController

说明:根据端口号分别修改输出信息

/\*\*

\* 测试负载均衡

\*/

@ResponseBody

@RequestMapping("/getMsg")

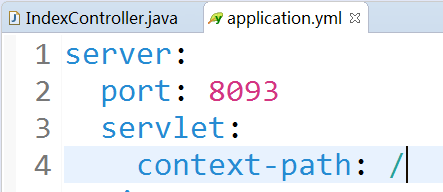
**public** String getMsg() {

**return** "我是8093服务器!!!!!!!";

}

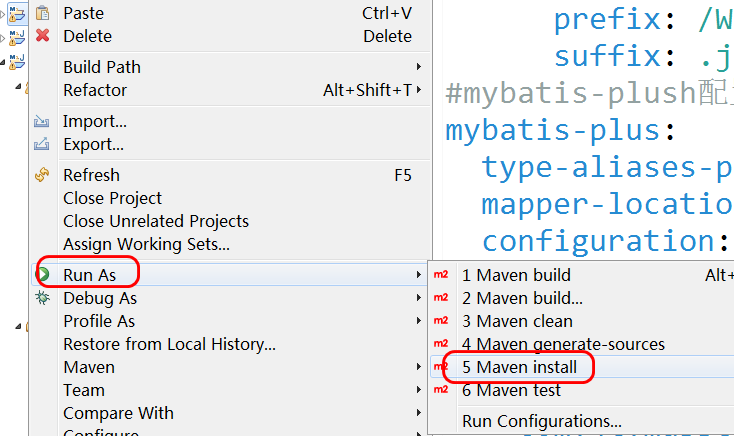
#### 编辑YML配置文件

说明:根据8091-8093不同的端口.修改yml配置信息



#### 项目打包

将修改后的服务项项目打包.第一次将jt整体打包,之后可以简化只打包jt-manage.



打包完成之后.,将war包拷贝到本地磁盘.路径如下修改war包名称为8091.war/8092.war/8093.war



### 关于jt打包报错说明

[INFO] Scanning for projects...

[ERROR] [ERROR] Some problems were encountered while processing the POMs:

[ERROR] **Child module** D:\1-JT\workspace\jt\abc of D:\1-JT\workspace\jt\pom.xml does **not exist** @

@

[ERROR] The build could not read 1 project -> [Help 1]

[ERROR]

[ERROR] The project com.jt:jt:0.0.1-SNAPSHOT (D:\1-JT\workspace\jt\pom.xml) has 1 error

[ERROR] Child module D:\1-JT\workspace\jt\abc of D:\1-JT\workspace\jt\pom.xml does not exist

[ERROR]

[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.

[ERROR] Re-run Maven using the -X switch to enable full debug logging.

[ERROR]

[ERROR] For more information about the errors and possible solutions, please read the following articles:

[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/ProjectBuildingException

原因:实际项目与jt父级项目中的pom.xml文件不匹配导致pom.xml文件报错.

检查标签与实际项目是否相符.

<modules>

<module>jt-common</module>

<module>jt-manage</module>

</modules>

## 项目集群发布

### 业务说明

准备3个tomcat.war包启动3台tomcat服务器.

方法1:将xxx.war部署到第三方tomcat服务器中,保存到webapp文件夹中即可. 太low

方法2:直接启动war包程序.运行服务.

java -jar xxxx.jar/war 适用于springBoot程序

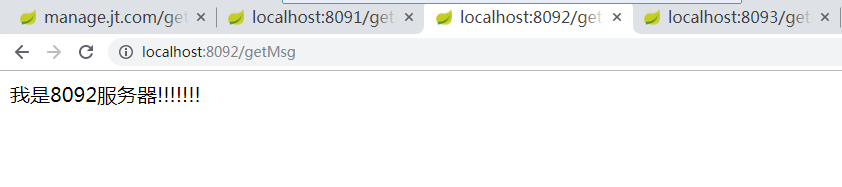
ctrl + c 表示结束运行!!!!

### 启动tomcat服务器命令

java -jar xxxx.jar/war 适用于springBoot程序

ctrl + c 表示结束运行!!!!

分别运行3台tomcat服务器.之后检查服务端口是否正确!!!



## Nginx负载均衡配置

### 轮询策略

说明:根据配置文件顺序,依次访问tomcat服务器.

#准备tomcat\_windows集群 默认:轮询策略

upstream jtWindows {

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 / {

#代理服务器路径

proxy\_pass http://jtWindows;

}

}

### 权重策略

说明:根据物理服务器配置,让高性能服务器尽可能多分配请求.

配置信息:

#准备tomcat\_windows集群 默认:轮询策略 2.权重策略

upstream jtWindows {

server localhost:8091 weight=6;

server localhost:8092 weight=3;

server localhost:8093 weight=1;

}

### IPHASH策略

说明:使用IPhash可以将用户与服务器绑定.实现了数据的共享

但是:一般很少使用.做测试或者小规模使用可以.大型公司禁止使用.

配置步骤:

#准备tomcat\_windows集群 默认:轮询策略 2.权重策略 3.了解 IPhash策略

upstream jtWindows {

ip\_hash;

server localhost:8091 weight=6;

server localhost:8092 weight=3;

server localhost:8093 weight=1;

}

缺点:

1.如果后台服务器宕机.则影响部分用户无法访问服务器.

2.使用iphash会造成严重的负载不均现象.

## NGINX运维配置项

### down属性

问题说明:当nginx实现负载均衡时,当后台服务器宕机,则nginx会等待超时时间.之后访问下一台服务器.但是每次周期都会去访问故障机.

属性说明:

down表示标识故障机.nginx如果遇到用down标识的服务器.则永远不会访问该机器.

upstream jtWindows {

#ip\_hash;

server localhost:8091 down;

server localhost:8092;

server localhost:8093;

}

### backup属性

说明:backup表示备用机机制.正常情况下该机器不参与用户的访问.留作备用.当服务器全部宕机或者主服务器遇忙时.

upstream jtWindows {

#ip\_hash;

server localhost:8091 down;

server localhost:8092 down;

server localhost:8093 backup;

}

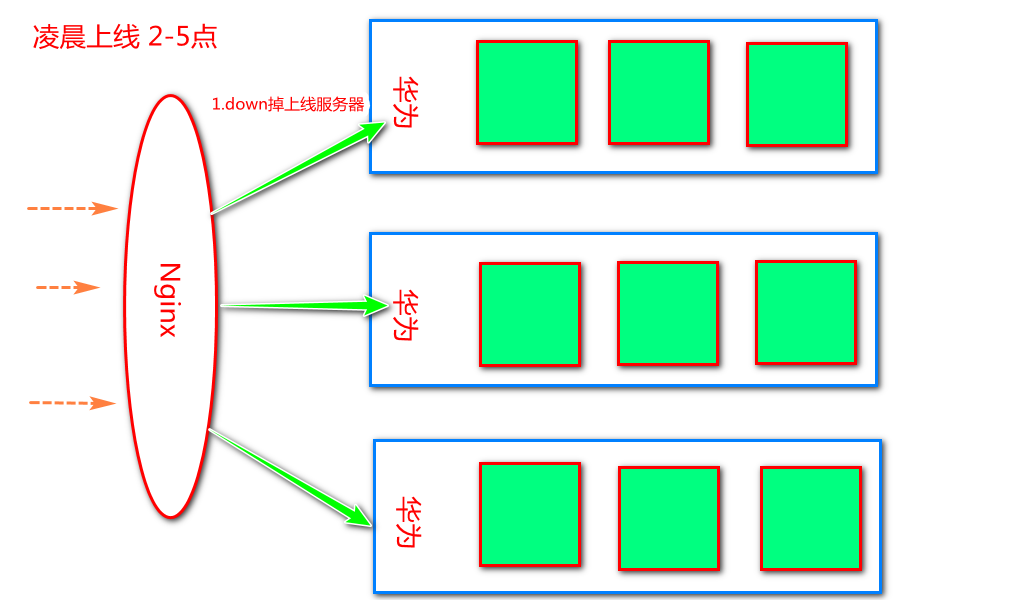
知识扩展:

项目组如何实现项目发布??

场景:公司内部有3台物理设备,每台上配置3台tomcat服务器.项目上线如何部署???? 视频网站

要求平稳过度.

一般通过nginx down属性实现分批部署.之后项目上线,经过测试之后准确无误.之后修改down属性.部署另外的服务器.



### 健康检查/心跳检测

说明:tomcat服务器如果正常可以抗击高并发.如果服务器宕机.则受一定的影响.这时采用备用机可以抗击压力.

问题:备用机要想生效必须满足down标签. 服务之后需要人为的添加down属性.

属性介绍:

max\_fails=1: 表示用户发起请求时链接不到服务器表示失败.

fail\_timeout=60s 如果服务器访问失败次数1次之后则在60秒内不会再访问故障机.

proxy\_connect\_timeout 2; #服务器链接超时时间

proxy\_read\_timeout 2; #读取服务器资源超时时间

proxy\_send\_timeout 2; #向服务器发送资源的超时时间

使用说明:

nginx健康监测是为了在没有程序员表示down属性时,引起频繁访问故障机的情况.当服务器宕机.只需要影响用户2秒之后,在指定的周期中不会再次访问故障机.提升用户体验.