**nginx**

1. **nginx 简介**

nginx (engine x) 是一个高性能的HTTP和反向代理服务器，也是一个IMAP/POP3/SMTP服务器。Nginx是由伊戈尔·赛索耶夫为俄罗斯访问量第二的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。2011年6月1日，nginx 1.0.4发布。

Nginx是一款轻量级的Web 服务器/反向代理服务器及电子邮件（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、新浪、网易、腾讯、淘宝等。

1. **反向代理**

首先要明白什么是代理，什么是反向代理，客户端使用的是代理，而服务端使用的是反向代理。

反向代理的实现规则

(1)首先需要一个负载均衡设备来处理用户的请求，将用户的请求再分发到空闲的服务器上

(2)服务器得到响应之后，处理数据并返回对应的响应数据到负载均衡设备

(3)负载均衡设备最后再将服务器返回的响应数据返回给用户

再举一个现实中的场景，比较容易理解一些

当我们拨打一个集团客户电话的时候，比如保险公司电话、10086移动客服电话、10000电信客服电话、110、119等等，可能拨打的号码后面对应的客服有几十个甚至几百个，但是你不会知道到底是谁接了电话，是谁在帮助你解决问题，因为整个分配过程是由系统自动完成，你可能每一次拨打，对应接听的客服都不一样，这里所拨打的号码就是反向代理。

反向代理的好处在于隐藏了真实的服务端，当我们请求某一个网址的时候，就像拨打集团号码一样，背后可能有成千上万台服务器为我们服务，但具体是哪一台，用户不会知道，用户也不关心这些，用户关心的只是能不能获取到自己想看到的内容而已，所以用户输入的网址就是反向代理服务器，反向代理服务器会帮我们把请求转发到真实的服务器那里去。

在反向代理服务中，nginx因为其良好的自身性能、稳定性和特性，占据全球领域大部分业务场景，所以国内的互联网巨头，几乎都是使用nginx完成的反向代理服务。

1. **nginx安装前序**

nginx是C语言开发的，尽量建议在Linux上部署运行，当然，也可以安装 Windows 版本，这里只演示怎样在Linux环境中使用。

下面的所有安装过程，为了简化安装步骤，建议使用超级管理员来完成，否则会出现权限不够的错误提示。

1. **PCRE pcre-devel安装**

PCRE(Perl Compatible Regular Expressions) 是一个Perl库，包括 perl 兼容的正则表达式库。nginx 的 http 模块使用 pcre 来解析正则表达式，所以需要在 linux 上安装 pcre 库，pcre-devel 是使用 pcre 开发的一个二次开发库。nginx也需要此库。命令：

yum install -y pcre pcre-devel

**(PS：yum命令会根据Linux当前环境中缺少的内容进行网络下载，所以首先确定是否能上网，否则会出现错误状态，后面不再标注说明)**

1. **zlib-gzip安装**

zlib 库提供了很多种压缩和解压缩的方式， nginx 使用 zlib 对 http 包的内容进行 gzip ，所以需要在 Linux上安装 zlib 库，执行以下命令安装：

yum install -y zlib zlib-devel

1. **gcc C++安装**

nginx 需要在本地Linux环境中进行编译，编译依赖 gcc C++环境，如果没有 gcc 环境，则需要执行以下命令进行安装：

yum install gcc-c++

1. **OpenSSL安装**

OpenSSL 是一个强大的安全套接字层密码库，囊括主要的密码算法、常用的密钥和证书封装管理功能及 SSL 协议，并提供丰富的应用程序供测试或其它目的使用。

nginx 不仅支持 http 协议，还支持 https（即在ssl协议上传输http），所以需要在 Linux 安装 OpenSSL 库。

yum install -y openssl openssl-devel

1. **解压nginx压缩包**

Nginx官方下载地址，建议使用稳定版本，也就是Stable version版本

https://nginx.org/en/download.html

下载完成之后，copy到Linux中，然后到对应的目录通过命令解压缩

tar -zxvf nginx-\*\*\*\*.tar.gz

1. **配置nginx编译环境**

通过cd命令进入解压完成的nginx目录中，通过命令开始配置

./configure

这里需要注意的时候，Linux系统会自动检查需要使用哪些配置项来完成这次编辑，并自动加入对应的内容，不需要手动配置，这次编译的时候，Linux系统会自动加入pcre、zlib、openssl、gcc进行对应编译操作

1. **开始nginx编译**

在解压完成的nginx目录中，首先执行编译命令

make

等执行编译完毕之后，再执行安装命令

make install

到这里nginx已经编译、安装、并部署到Linux系统中，下面我们来测试下是否已经成功

1. **开始运行nginx**

如果前面全部安装成功，这时候输入下面的命令可以到nginx的目录中

cd /usr/local/nginx/sbin/

然后通过命令启动nginx

./nginx

启动以后，不会有任何提示，现在我们需要打开一个网页来访问下就知道是否运行成功，因为nginx默认需要80端口支持，前面知识已经讲述过，80是不需要输入端口号的，所以只需要输入

http://电脑IP地址

这个时候如果成功，应该看到下面的效果



1. **退出运行nginx**

一般情况下，服务器不会轻易关闭，如果一定要关闭，输入命令

./nginx -s quit

1. **开启外部访问权限**

跟前面讲述在Linux上部署java-web项目一样，我们需要手动开启80端口对外访问权限

首先开启外部网络访问80端口

firewall-cmd --permanent --add-port=80/tcp

然后重启防火墙

firewall-cmd --reload

这时候，在外部同局域网电脑也可以正常访问了

1. **nginx与tomcat的配置**

在修改配置文件之前，一定要记得备份一份原始文件到本地，避免修改错误，导致重新安装nginx

进入到nginx的配置文件中，直接打开它，操作命令

vi /usr/local/nginx/conf/nginx.conf

然后修改下图中的两个位置



server\_name localhost;

location / {

root html;

index index.html index.html;

}

然后为下面的效果：



server\_name localhost:8080;

location / {

proxy\_pass <http://localhost:8080>;

}

接着，需要重启nginx，不需要重启tomcat，然后在访问服务器IP，直接就跳转到tomcat中了

1. **nginx与SpringBoot的配置**

因为SpringBoot中自带tomcat，所以如果在没有修改SpringBoot中tomcat的端口的情况下，可以直接使用上面修改好的nginx，在启动SpringBoot前，记得退出前面加载的tomcat，否则端口会冲突，最后通过服务器IP访问SpringBoot中的内容，如果访问到代表成功