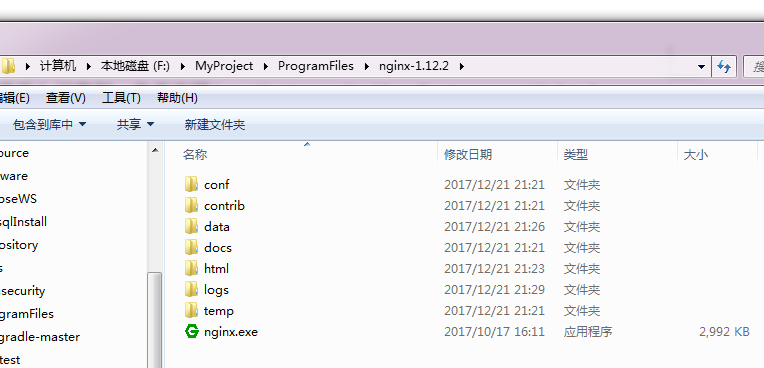
**说明：以下一些示例是参考官方提供的例子实现的，但是官网上的有点瑕疵，准确性不够，以实际案例为准。**

# 一、自己搭建的入门案例(Serving Static Content)

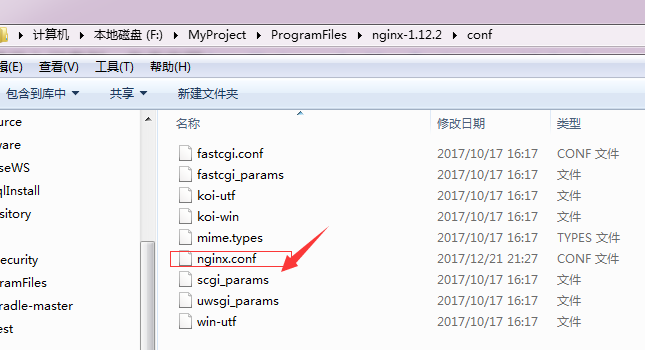
参考官网<http://nginx.org/en/docs/beginners_guide.html>



1. nginx目录结构(data文件夹是我自己加的，里面是我自己的静态资源，包括html和images静态资源)



关键配置文件：



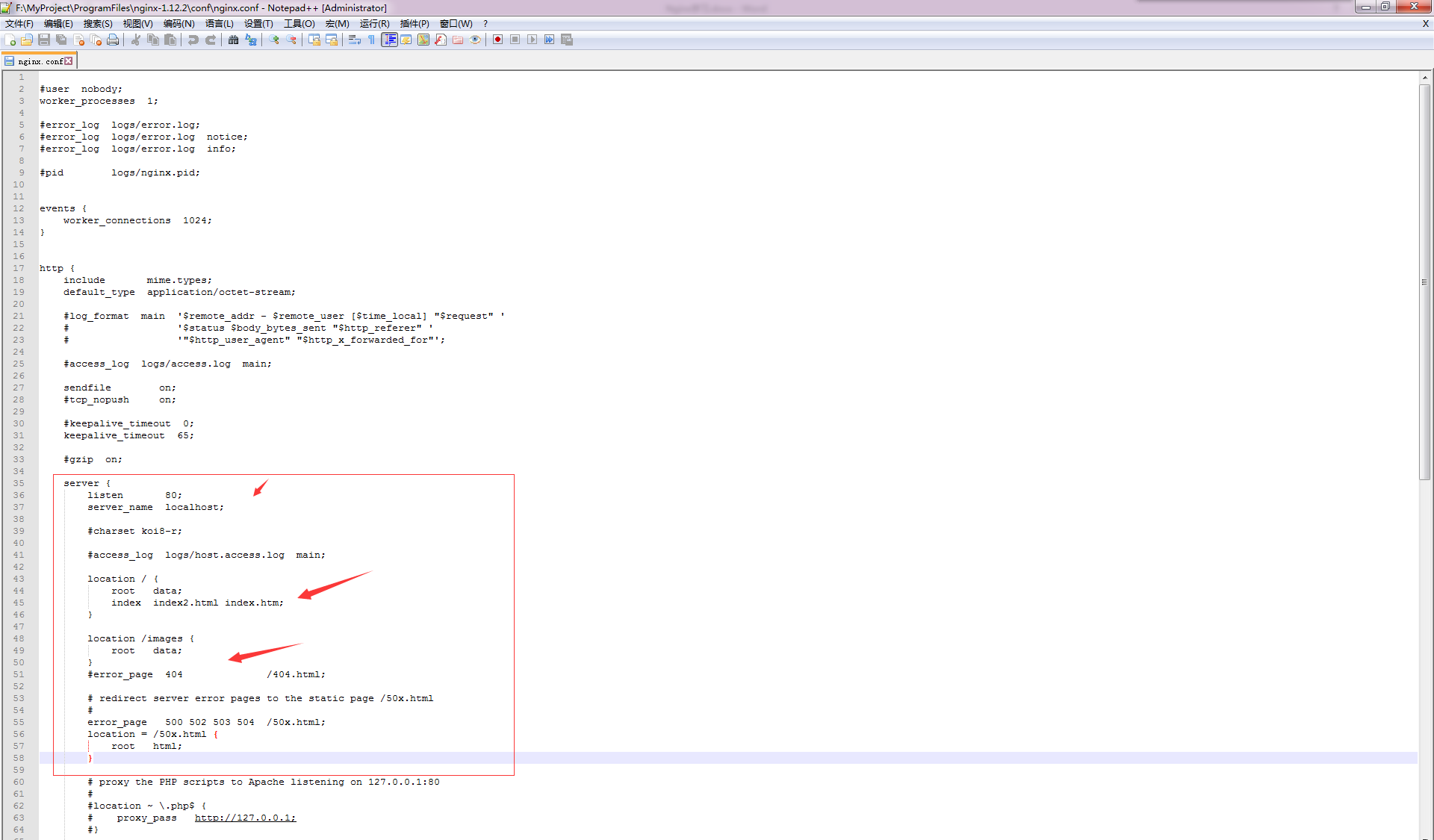
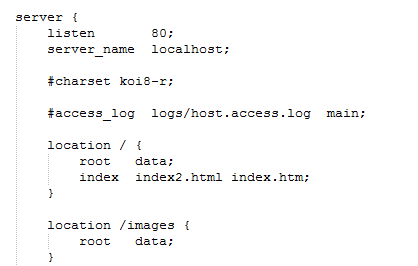
2．Windows cmd命名行操作nginx：

cd 到nginx目录，nginx.exe –s signal

signal: reload,stop,quit等。

启动nginx可dbclick nginx.exe，也可命令行nginx.exe 或 start nginx.exe

1. server location配置

 -> 

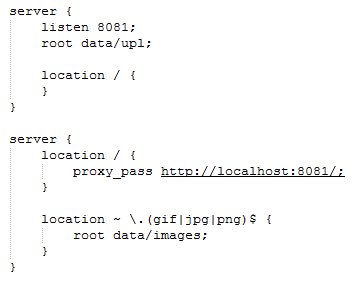
# 二、搭建简单的代理服务器（Simple Proxy Server）

参考官网：<http://nginx.org/en/docs/beginners_guide.html>

我使用的是一台Nginx服务器，提供两个server，一个作为代理服务器，另一个作为被代理服务器。

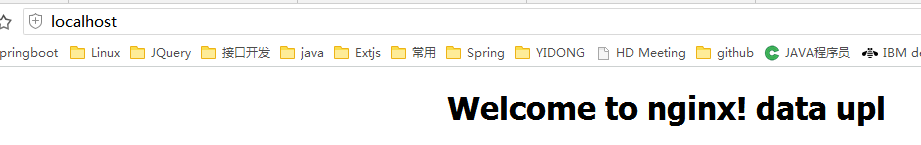


nginx.conf配置

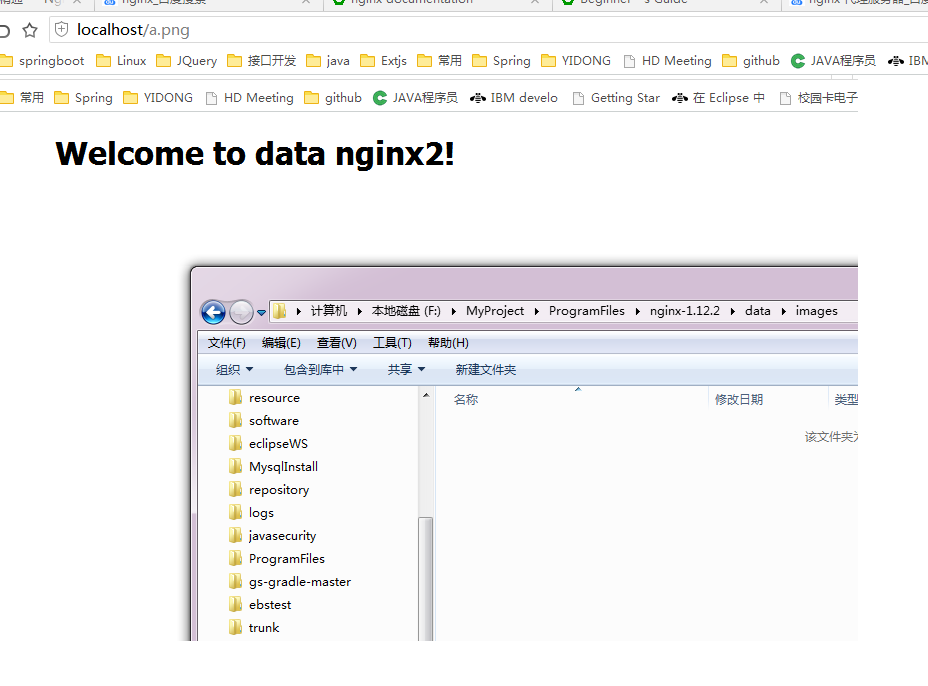


浏览器访问：

1.当浏览器访问localhost时，nginx代理服务器将请求转发给被代理服务器localhost:8081，因其root设置为data/upl，故请求会被代理到localhost:8081/data/upl/index.html



2.当浏览器访问<http://localhost/a.png>时，nginx代理服务器通过正则表达式匹配并过滤到该请求，指向其本地的静态图片资源，最终的请求路径是localhost:80/data/images/a.png并不会将其进行代理转发。(注：localhost默认端口是80)



# 三、了解Nginx

Nginx官网主要维护两个版本：稳定版（生产）stable和开发版Mainline

Nginx两种安装方式：pre-build package安装和compiled from the sources源码编译安装（更灵活，可加入第三方模块）

Nginx采用多进程，异步非阻塞的方式处理请求，在其商用版还包括了多线程处理来阻塞问题。（在NGINX 1.7.11中引入了线程池。默认情况下，NGINX+还没有包含线程池，但是如果你想试试的话，可以联系销售人员，NGINX+ R6是一个已经启用了线程池的构建版本。）

Nginx原理可参考官网 <http://nginx.org/en/docs/beginners_guide.html>

中文可参考 [http://tengine.taobao.org/book/chapter\_02.html#id1](http://tengine.taobao.org/book/chapter_02.html%23id1)

## Compiling and Installing From the Sources

### Installing NGINX Dependencies

Prior to compiling NGINX from the sources, it is necessary to install its dependencies:

* the [PCRE](http://pcre.org/) library – required by NGINX [Core](https://nginx.org/en/docs/ngx_core_module.html) and [Rewrite](https://nginx.org/en/docs/http/ngx_http_rewrite_module.html) modules and provides support for regular expressions:
* $ wget ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-8.41.tar.gz
* $ tar -zxf pcre-8.41.tar.gz
* $ cd pcre-8.41
* $ ./configure
* $ make

$ sudo make install

* the [zlib](http://www.zlib.net/) library – required by NGINX [Gzip](https://nginx.org/en/docs/http/ngx_http_gzip_module.html) module for headers compression:
* $ wget http://zlib.net/zlib-1.2.11.tar.gz
* $ tar -zxf zlib-1.2.11.tar.gz
* $ cd zlib-1.2.11
* $ ./configure
* $ make

$ sudo make install

* the [OpenSSL](https://www.openssl.org/) library – required by NGINX SSL modules to support the HTTPS protocol:
* $ wget http://www.openssl.org/source/openssl-1.0.2k.tar.gz
* $ tar -zxf openssl-1.0.2k.tar.gz
* $ cd openssl-1.0.2k
* $ ./configure darwin64-x86\_64-cc --prefix=/usr
* $ make

$ sudo make install

### Downloading the Sources

NGINX provides source files for both stable and mainline versions. The source files can be downloaded from the NGINX Open Source download page at <http://www.nginx.org/en/download.html>.

To download and unpack source files for the latest mainline version, type-in the commands:

$ wget http://nginx.org/download/nginx-1.13.5.tar.gz

$ tar zxf nginx-1.13.5.tar.gz

$ cd nginx-1.13.5

To download and unpack source files for the latest stable version, type-in the commands:

$ wget http://nginx.org/download/nginx-1.12.2.tar.gz

$ tar zxf nginx-1.12.2.tar.gz

$ cd nginx-1.12.2

### Completing the Installation

* Compile and install the build:
* $ make

$ sudo make install

* After the installation is finished, run NGINX Open Source:

$ sudo nginx

# 四、Nginx实际运用，结合工会项目上Nginx阐述

<http://www.linuxidc.com/Linux/2015-07/120738.htm>

Nginx集群实现负载均衡

<https://www.cnblogs.com/xiaocen/p/3718006.html>

<http://blog.csdn.net/wang379275614/article/details/47778201>

Nginx+Tomcat集群负载均衡，Session保持。

<http://blog.csdn.net/e421083458/article/details/30092795>

nginx+keepalived实现双机热备+负载均衡