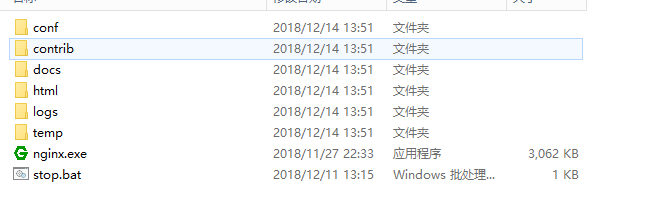
在我们搭建一个网站的时候，往往有时候会加载更多的图片，如果都从tomcat服务器来获取静态资源，这样会增加我们服务器的负载，使得服务器运行 速度非常慢，这时我们可以使用nginx服务器来加载这些静态资源，这样就可以实现负载均衡，为我们的Tomcat服务器减压了。一般大型网站都这么干，他们有单独的图片服务器，这里我们在本地利用nignx来搭建一个简单的图片服务器。

一、安装nignx

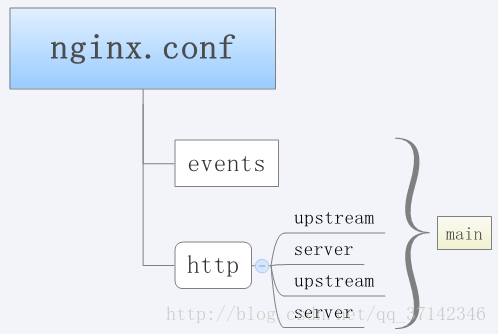
nignx是绿色版本的，只要到官网下载解压既可启动，解压目录图如下所示：



这里我们可以通过命令行start nignx.exe来启动服务器，也可以通过bat批处理文件来启动，打开批处理文件简单处理一下就 可以启动了：

上面的两种方式都可以启动服务器，当我们启动一下服务器我们来简单测试一下，在浏览器输入localhost访问，你可以看见一个简单nignx服务器欢迎页面。

下面，我们着重讲解一下nignx的配置文件，在conf目录下，打开nignx.conf文件，nginx.conf由多个块组成，最外面的块是main，main包含events和http,http包含多个upstream和多个server，server又包含多个location：



main（全局设置）、server（虚拟主机设置）、upstream（负载均衡服务器设置）和 location（URL匹配特定位置的设置）。

1.main块设置的指令将影响其他所有设置；

2.server块的指令主要用于指定主机和端口；

3.upstream指令主要用于负载均衡，设置一系列的后端服务器；

4.location块用于匹配网页位置。

这四者之间的关系式：server继承main，location继承server，upstream既不会继承其他设置也不会被继承。

在这四个部分当中，每个部分都包含若干指令，这些指令主要包含Nginx的主模块指令、事件模块指令、HTTP核心模块指令，同时每个部分还可以使用其他HTTP模块指令，例如Http SSL模块、HttpGzip Static模块和Http Addition模块等。

通过上面的简单的讲解我们了解了一下nignx的配置文件，现在我们来具体看一下它的内容并且开始配置：

server {

listen 80;

server\_name localhost;

#charset koi8-r;

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

location / {

root html;

index index.html index.htm;

}

#error\_page 404 /404.html;

# redirect server error pages to the static page /50x.html

#

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

# proxy the PHP scripts to Apache listening on 127.0.0.1:80

#

#location ~ \.php$ {

# proxy\_pass http://127.0.0.1;

#}

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000

#

#location ~ \.php$ {

# root html;

# fastcgi\_pass 127.0.0.1:9000;

# fastcgi\_index index.php;

# fastcgi\_param SCRIPT\_FILENAME /scripts$fastcgi\_script\_name;

# include fastcgi\_params;

#}

# deny access to .htaccess files, if Apache's document root

# concurs with nginx's one

#

#location ~ /\.ht {

# deny all;

#}

}

上面这是配置文件中的一部分，我们看一下location这个节点，它下面的root表示该站点的根目录，root html表示根目录下有一个html文件夹，当你访问上面配置的域名（laocahost）时，它默认访问跟目录下的html文件中的index.html页面，这样你应该就明白了怎么样配置一个自定义的图片服务器了吧，我们可以自定一个server节点，将其localhost节点指定为我们要访问的图片地址域名，这样我们就可以轻松访问我们的静态资源了。

location / {

root html;

index index.html index.htm;

}

二、搭建图片服务器

经过上面的介绍我们应该明白了它的大概原理，下面我们来实战配置一下。

然后我们复制nignx.conf文件的server节点，改为如下配置：

gzip on;

upstream node\_server {

server 127.0.0.1:8091;

server 127.0.0.1:8092;

}

server {

listen 8090;

server\_name localhost;

#charset koi8-r;

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

location / {

proxy\_pass http://node\_server;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

}

location /html {

root html;

index index.html index.htm;

}

}

我们将域名改为了我们自定义的图片域名，将root目录改为了我们的图片存放目录，这样我们启动nignx服务器就可以访问图片资源了，这里加入该目录下有一张图片为test.jpg。当你在浏览器中输入localhost/test.jpg就可以访问我们的图片资源了。

至此，我们就搭建了一个简单的图片服务器。