

nginx的上传模块upload

【转】<https://blog.csdn.net/daa20/article/details/75084282>

一. nginx upload module原理

官方文档: <http://www.grid.net.ru/nginx/upload.en.html>

Nginx upload module通过nginx服务来接受用户上传的文件, 自动解析请求体中存储的所有文件上传到upload_store指定的目录下。这些文件信息从原始请求体中分离并根据nginx.conf中的配置重新组装好上传参数, 交由upload_pass指定的段处理, 从而允许处理任意上传文件。每个上传文件中的file字段值被一系列的upload_set_form_field指令值替换。每个上传文件的内容可以从\$upload_tmp_path变量读取, 或者可以将文件转移到目的目录下。上传的文件移除可以通过upload_cleanup指令控制。如果请求的方法不是POST, 模块将返回405错误(405 Not Allowed), 该错误提示可以通过error_page指令处理。

具体的过程如下:

1. 用户访问能够选择上传文件的页面
2. 用户提交表单
3. 浏览器把文件和有关文件的信息作为请求的一部分发送给服务器
4. 服务器把文件保存到临时存储目录下upload_store
5. upload_pass指定的处理表单提交的php页面将文件从upload_store拷贝到持久存储位置

二.nginx upload module配置参数

upload_pass 指明后续处理的php地址。文件中的字段将被分离和取代, 包含必要的信息处理上传文件。

upload_resumable 是否启动可恢复上传。

upload_store 指定上传文件存放地址(目录)。目录可以散列, 在这种情况下, 在nginx启动前, 所有的子目录必须存在。

upload_state_store 指定保存上传文件可恢复上传的文件状态信息目录。目录可以散列, 在这种情况下, 在nginx启动前, 所有的子目录必须存在。

upload_store_access 上传文件的访问权限, user:r是指用户可读

upload_pass_form_field 从表单原样转到后端的参数, 可以正则表达式表示。:

\$upload_field_name - 原始文件中的字段的名称

upload_pass_form_field "^submit|description|description";

意思是把submit, description这两个字段也原样通过upload_pass传递到后端php处理。如果希望把所有的表单字段都传给后端可以用upload_pass_form_field "^.*\$";

upload_set_form_field 名称和值都可能包含以下特殊变量:

\$upload_field_name 表单的name值

\$upload_content_type 上传文件的类型

\$upload_file_name 客户端上传的原始文件名称

\$upload_tmp_path 文件上传后保存在服务端的位置

upload_aggregate_form_field 可以多使用的几个变量,文件接收完毕后生成的并传递到后端

\$upload_file_md5 文件的MD5校验值

\$upload_file_md5_uc 大写字母表示的MD5校验值

\$upload_file_sha1 文件的SHA1校验值

\$upload_file_sha1_uc 大写字母表示的SHA1校验值

\$upload_file_crc32 16进制表示的文件CRC32值

\$upload_file_size 文件大小

\$upload_file_number 请求体中的文件序号

这些字段值是在文件成功上传后计算的。

upload_cleanup 如果出现400 404 499 500-505之类的错误,则删除上传的文件

upload_buffer_size 上传缓冲区大小

upload_max_part_header_len 指定头部分最大长度字节。

upload_max_file_size 指定上传文件最大大小, 软限制。client_max_body_size硬限制。

upload_limit_rate 上传限速, 如果设置为0则表示不限制。

upload_max_output_body_len 超过这个大小, 将报403错(Request entity too large)。

upload_tame_arrays 指定文件字段名的方括号是否删除

upload_pass_args 是否转发参数。

三. nginx配置



```
# wget http://www.nginx.org/download/nginx-1.2.2.tar.gz

# wget http://www.grid.net.ru/nginx/download/nginx_upload_module-2.2.0.tar.gz

# tar zxvf nginx_upload_module-2.2.0.tar.gz -c ../software/

# tar zxvf nginx_upload_module-2.2.0.tar.gz -C ../software/

# ./configure --prefix=/usr/local/nginx --add-module=../nginx_upload_module-2.2.0 --
with-http_secure_link_module

# make
```

```
# make install

# vi nginx.conf

user www-data;

worker_processes 20;

error_log logs/error.log notice;

working_directory /usr/local/nginx;

events {

worker_connections 1024;

}

http {

include mime.types;

default_type application/octet-stream;

root /www/web/upload;

server {

listen 80;

server_name 192.168.41.129;


error_page 405 =200 @405; //处理405错误

location / {

index index.html index.htm index.php;

}

location @405

{

root /www/web/upload;

}

location ~ /\.php$ {
```

```
try_files $uri /404.html;

fastcgi_pass 127.0.0.1:9000;

fastcgi_index index.php;

fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;

include /etc/nginx/fastcgi_params;
}

client_max_body_size 100m;

# 上传页面提交到这个location

location /upload {

# 文件上传以后转交给后端的php代码处理

upload_pass @test;

# 上传文件的临时存储位置，目录是散列的，应该存在子目录0 1 2 3 4 5 6 7 8 9

upload_store /www/web/upload/tmp 1;

upload_store_access user:r;

# 设置请求体的字段

upload_set_form_field "${upload_field_name}_name" $upload_file_name;

upload_set_form_field "${upload_field_name}_content_type" $upload_content_type;

upload_set_form_field "${upload_field_name}_path" $upload_tmp_path;

# 指示后端关于上传文件的md5值和文件大小

upload_aggregate_form_field "${upload_field_name}_md5" $upload_file_md5;

upload_aggregate_form_field "${upload_field_name}_size" $upload_file_size;

# 指示原样转到后端的参数，可以用正则表达式表示

upload_pass_form_field "^submit$|^description$";

upload_pass_args on;

}

# 将请求转到后端的地址处理
```

```
location @test {  
  
    rewrite ^(.*)$ /test.php last;  
  
}  
  
}  
  
}
```