# 一、OpenResty 环境搭建

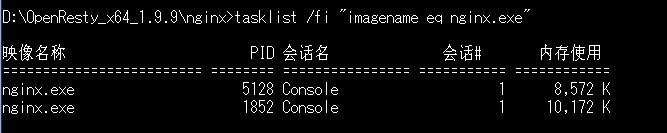
**1、**Windows平台

进入D:\OpenResty\_x64\_1.9.9\nginx

启动：start nginx

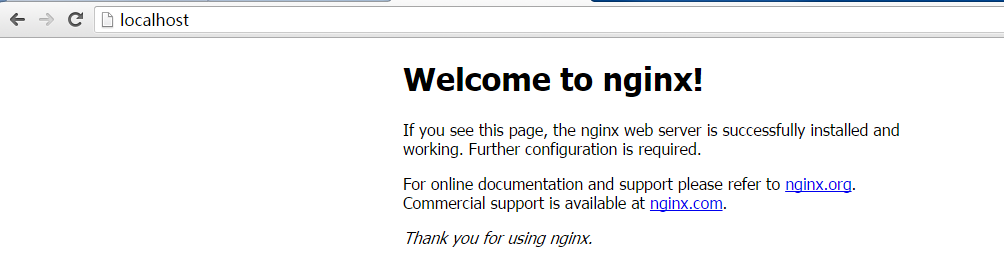
关闭：taskkill /F /IM nginx.exe > nul 或 nginx -s stop

检查状态：tasklist /fi "imagename eq nginx.exe"

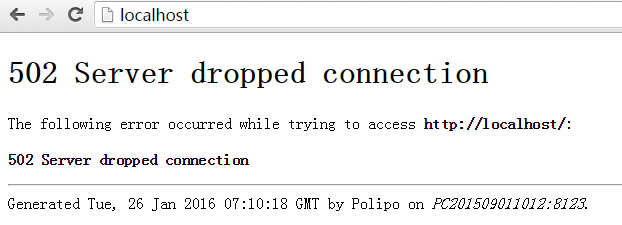


一个是master进程，另一个是worker进程

IE输入localhost验证加载页面：



注，如果用VPN上网可能出错



设置环境变量：

为了不用在安装目录下进行启动，可以通过设置环境变量来简化操作。(重启生效)

Window:

为系统变量PATH加入D:\OpenResty\_x64\_1.9.9\nginx

Linux

打开文件/etc/profile,文件追加入 export PATH=$PATH:/opt/openresty/nginx/sbin

1. nginx/sbin目录添加到PATH

# 二、Hello World

根目录创建测试代码

D:\OpenResty\_x64\_1.9.9\test

D:\OpenResty\_x64\_1.9.9\test\conf

D:\OpenResty\_x64\_1.9.9\test\logs

D:\OpenResty\_x64\_1.9.9\test\conf\nginx.conf 代码块：



**nginx -p D:/OpenResty\_x64\_1.9.9/test/ -c conf/nginx.conf**

错误1：



解决方法，手动增加目录：D:\OpenResty\_x64\_1.9.9\test\temp\client\_body\_temp

热部署：**nginx -p D:/OpenResty\_x64\_1.9.9/test/ -s reload**

访问静态文件：

location ~ \.(gif|jpg|png|js|css)$ {

#root D:/OpenResty\_x64\_1.9.9/test/res/;

root webroot/res/;

}

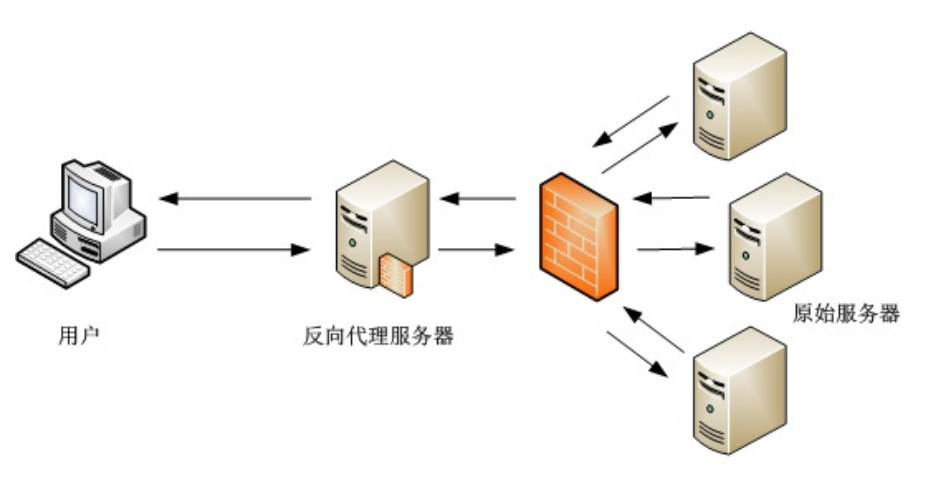
注：绝对路径使用方法：root D:/OpenResty\_x64\_1.9.9/test/res/;

相对路径使用方法：root webroot/res/; **不可：/webroot/res/，后面查下root在不同平台的路径**

# 三、API应用

# 四、反向代理

反向代理：是指以[代理服务器](http://baike.baidu.com/view/751.htm)来接受internet上的连接请求，然后将请求转发给内部网络上的服务器，并将从服务器上得到的结果返回给internet上请求连接的客户端，此时代理服务器对外就表现为一个反向代理服务器。



反向代理典型应用场景：是将防火墙后面的服务器提供给internet用户访问，加强安全防护。还可以为后端的多台服务器提供负载均衡，或为后端较慢的服务器提供缓冲服务。

反向代理服务器是介于Internet客户端和原始服务器的中间处理服务器，负责如何将Internet请求转发给原始服务器，再如何将原始服务器响应转发给Internet客户端。反向代理服务器起到过滤器作用，起到安全，高效，稳定。。。作用。

|  |  |  |
| --- | --- | --- |
| 配置参数 | 说明 | 示例 |
| proxy\_pass | 用来将请求反向代理到URL参数指定到服务器上 |  |
| proxy\_set\_header | 转发原始请求中的Host头部 | Proxy\_set\_header Host $host |
| Proxy\_redirect |  |  |
| Proxy\_set\_body |  |  |
| Proxy\_limit\_rate |  |  |
|  |  |  |

# 十、我对OpenResty的理解

可靠性高：核心框架模块设计（简单，稳定）

可扩展性：模块间耦合度极低（第三方模块庞大）

热部署：master管理进程与worker工作进程的分离设计

文件结构：

**nginx.conf**是主配置文件，默认配置去掉注释之后的内容如下图所示：

**worker\_process** # 表示工作进程的数量，一般设置为cpu的核数

**worker\_connections** # 表示每个工作进程的最大连接数

**server**{} # 块定义了虚拟主机

**listen** # 监听端口

**server\_name**# 监听域名

**location** {}# 是用来为匹配的 URI 进行配置，URI 即语法中的“/uri/”

**location** /{}# 匹配任何查询，因为所有请求都以 / 开头

**root**# 指定对应uri的资源查找路径，这里html为相对路径，完整路径为#/opt/nginx-1.7.7/html/

**index**# 指定首页index文件的名称，可以配置多个，以空格分开。如有多# 个，按配置顺序查找。

# 附录：

nginx常用命令

nginx -s stop 强制关闭   
nginx -s quit 安全关闭   
nginx -s reload 改变配置文件的时候，重启nginx工作进程，来时配置文件生效   
nginx -s reopen 打开日志文件

待解问题列表：

|  |  |  |
| --- | --- | --- |
| 问题描述 |  |  |
| using inherited sockets from "D:\OpenResty\_x64\_1.9.9\nginx\nginx.exe" |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# ngx\_http\_proxy\_module模块

the ngx\_http\_proxy\_module module allows passing requests to another server.

Example Configuration

location / {

proxy\_pass http://localhost:8000;

proxy\_set\_header Host $host;

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

}

## Directives

# Location 区配规则

## 语法：location [=|~|~\*|^~]/uri/{…}

|  |  |  |
| --- | --- | --- |
| 符号 | 含 义 |  |
| **=** | 开头表示精确匹配 |  |
| **^~** | 开头表示uri以某个常规字符串开头，理解为匹配url路径即可。nginx不对url做编码，因此请求为 /static/20%/aa ，可以被规则 ^~/static/ /aa 匹配到（注意是空格） |  |
| **~** | 开头表示区分大小写的正则匹配 |  |
| **~\*** | 开头表示不区分大小写的正则匹配 |  |
| **!~** | 区分大小写不匹配 |  |
| **!~\*** | 不区分大小写不匹配 |  |
| **/** | 通用匹配，任何请求都会匹配到。 |  |

### 区配顺序：

首先区配 =

其次区配 ^~

其次是按文件中顺序的正则区配

最后是交给/通用区配

当有区配成功时，停止区配，按当前区配规则处理请求

示例：

规则C: ^~ /static/

localhost/static/a.html

localhost/static/c.png

规则D: ~ \.(gif|jpg|png|js|css)$

Localhost/a.gif

规则E:~\* \.png$

规则F: !~ \.xhtml$

规则G: !~\* \.xhtml$