

# nginx服务部署

原创

GeorgeKai

2018-02-04 09:05:10

评论(0)

791人阅读

作者: Georgekai  
归档: 学习笔记  
2018/2/2

## nginx服务部署

### 1.1 常用web软件了解

#### 1.1.1 web服务主流软件地址

[https://w3techs.com/technologies/overview/web\\_server/all](https://w3techs.com/technologies/overview/web_server/all)

#### 1.1.2 常用来提供动态服务的软件

PHP: .php结尾的文件, 大中小网站都有适合, 动态网页语言PHP程序的解析容器, 一般配合apache或nginx解析 动态程序

Tomcat: 适用于中小企业, 不适合并发量高的环境

Resin: 适用于大型企业, 适合并发量高的环境

IIS: .asp结尾的文件, windows下web服务软件

### 1.2 Nginx介绍

1. Nginx ( "engine x" ) 是一个开源的, 支持高性能、高并发的WWW服务器和代理服务软件

2. Nginx因具有高并发(特别是静态资源)、占用系统资源少等特性, 且功能丰富而逐渐流行起来。

3. Nginx可以运行在UNIX、Linux、BSD、Mac OS X、Solaris, 以及Microsoft Windows等操作系统中。

4. 当前流行的Nginx Web组合被称为LNMP或LEMP

5. L==Linux N/E==Nginx M==MySQL P==php 网站的代码程序 ==搭建出一个完整的网站

#### 1.2.1 官方网站: <http://nginx.org/>

#### 1.2.2 软件版本的选择 (建议)

当前时间往前推1年到半年之间的版本相对稳定(近期的可能有bug)

#### 1.2.3 模块使用(中文版): <http://manual.51yip.com/nginx/>

### 1.3 nginx软件特性说明

1) 支持高并发: 能支持几万并发连接(特别是静态小文件业务环境)

2) 资源消耗少: 在3万并发连接下, 开启10个Nginx线程消耗的内存不到200MB

PS: (测试软件工具: [ab](#), [JMeter](#), [Webbench](#), [LoadRunner](#), [http\\_load](#), [tcpcopy](#))

3) 可以做HTTP反向代理及加速缓存、即负载均衡功能(4层以及7层),

内置对RS节点服务器健康检查功能, 这相当于专业的Haproxy软件或LVS(4层)的功能。

PS: LVS等软件没有健康检查功能

4) 具备Squid等专业缓存软件等的缓存功能。(memcache/redis)

5) 支持异步网络I/O事件模型epoll (Linux 2.6+)

PS: apache使用的模型:select, 性能没有nginx好

### 1.4 为什么Nginx总体性能比Apache高

Ngimx:使用最新的epoll（linux 2.6内核）模型，和kqueue（freebsd）异步网络I/O模型  
特点比喻：找女朋友，宿管阿姨查看登记信息，快速查找人员信息，这是epoll模型  
Apache:使用的是传统的select模型  
特点比喻：找女朋友，宿管阿姨一个一个带你去找，去问，这是select模型

表 7-1 Apache select 和 Nginx epoll 的技术对比

指 标	select	epoll
性能	随着连接数的增加性能急剧下降。处理成千上万并发连接数，性能很差	随着连接数的增加，性能基本上没有下降。处理成千上万并发连接时，性能很好
连接数	连接数有限制，处理的最大连接数不超过 1024，如果要处理的连接数超过 1024 个，则需要修改 FD_SETSIZE 宏，并重新编译	连接数无限制
内在处理机制	线性轮询	回调 callback
开发复杂性	低	中

@51CTO博客

1.5 nginx软件的编译安装步骤

PS：也可以参考官方文档：从源头构建nginx：http://nginx.org/en/docs/configure.html

1.5.1先 检查软件安装的系统环境

```
cat /etc/redhat-release
uname -r
```

1.5.2 下载nginx软件——1.12.2 复制链接地址（统一位置进行下载）

```
mkdir -p /server/tools
cd /server/tools
wget -q http://nginx.org/download/nginx-1.12.2.tar.gz
```

1.5.3 安装nginx的依赖包（pcre-devel openssl-devel）

```
yum install -y pcre-devel openssl-devel
PS：不安装pcre，不能使用rewrite模块（rewrite需要识别perl语言的正则表达式）
PS：openssl：对网站远程访问进行加密
```

1.5.4 编译安装软件的步骤

提前条件：如果系统没有安装兼容程序库、开发工具的，请安装一下基础依赖包

```
yum install gcc gcc-c++ automake autoconf -y
```

- 1. 解压到当前目录nginx-1.12.2.tar.gz

```
tar xf nginx-1.12.2.tar.gz
```
- 2. 进入nginx-1.12.2目录

```
cd nginx-1.12.2/
```
- 3. 创建www虚拟用户  
PS：让www用户专门管理nginx服务，root权限太大，不安全

```
useradd -M -s /sbin/nologin www
```
- 3. 指定安装路径，配置参数（下面有参数解释）

```
./configure --prefix=/application/nginx-1.12.2 --user=www --group=www --with-http_stub_status_module --with-http_ssl_module
```
- 4. 编译（翻译）  
PS：编译过程实质是将各种程序语言转换为系统可以识别的二进制信息

```
make
```
- 5. 编译安装

```
make install
```
- 6. 创建程序目录软连接  
PS：避免版本经常变化，到时候只需更改软连接，不需要更改代码程序

```
ln -s /application/nginx-1.12.2/ /application/nginx
```

7. 启动网站服务

```
/application/nginx/sbin/nginx
```

8. 检查服务是否启动成功

```
ps -ef |grep nginx
```

9. 测试访问

```
lynx 10.0.0.7
curl 10.0.0.7
```

浏览器访问方式

## 1.6 编辑安装配置参数说明

```
./configure
--prefix=PATH          --指定软件程序安装的路径信息
--user=USER             --创建一个虚拟用户，用于管理nginx服务的worker进程
--group=GROUP           --创建一个虚拟用户组，用于管理nginx服务的worker进程
--with-http_ssl_module  --让nginx服务可以支持https访问
--with-http_stub_status_module --便于监控软件监视nginx服务运行状态
```

以下参数可参考（默认有上面的即可）：

```
--sbin-path=/usr/local/nginx/nginx
--conf-path=/usr/local/nginx/nginx.conf
--pid-path=/usr/local/nginx/nginx.pid
--with-pcre=../pcre-8.41
--with-zlib=../zlib-1.2.11
```

## 1.7 nginx命令参数

```
/application/nginx/sbin/nginx          --启动nginx
/application/nginx/sbin/nginx -s reload --平滑重启
/application/nginx/sbin/nginx -s stop  --停止服务
/application/nginx/sbin/nginx -t      --配置文件语法检查
/application/nginx/sbin/nginx -h      --查看参数使用帮助信息
/application/nginx/sbin/nginx -V      --查看软件编译安装配置参数信息
```

### 1.7.1 nginx 查看软件编译安装配置参数信息（-V）

```
[root@web01 application]# /application/nginx/sbin/nginx -V
nginx version: nginx/1.10.3
built by gcc 4.4.7 20120313 (Red Hat 4.4.7-18) (GCC)
built with OpenSSL 1.0.1e-fips 11 Feb 2013
TLS SNI support enabled
configure arguments: --prefix=/application/nginx-1.10.3 --user=www --group=www --with-
http_stub_status_module --with-http_ssl_module
```

**PS：**一般用于查看之前配置nginx的参数

### 1.7.2 nginx 查看参数使用帮助信息（-h）

```
[root@web01 application]# /application/nginx/sbin/nginx -h
nginx version: nginx/1.10.3
Usage: nginx [-?hvVtTq] [-s signal] [-c filename] [-p prefix] [-g directives]
```

Options:

```
-?, -h      : this help
-v          : show version and exit
-V          : show version and configure options then exit
-t          : test configuration and exit
-T          : test configuration, dump it and exit
-q          : suppress non-error messages during configuration testing
-s signal   : send signal to a master process: stop, quit, reopen, reload
-p prefix   : set prefix path (default: /application/nginx-1.10.3/)
-c filename : set configuration file (default: conf/nginx.conf)
-g directives : set global directives out of configuration file
```

## 1.8 部署nginx软件过程中可能遇到的问题

### 1. 软件依赖包未正确安装问题---PCRE依赖包没有安装

```
./configure: error: the HTTP rewrite module requires the PCRE library.
You can either disable the module by using --without-http_rewrite_module
```

option, or install the PCRE library into the system, or build the PCRE library statically from the source with nginx by using `--with-pcre=<path>` option.

解决方法: `yum install pcre pcre-devel -y`

## 2. 软件依赖包未正确安装问题---OPENSSL依赖包没有安装

`./configure: error: SSL modules require the OpenSSL library.`

You can either do not enable the modules, or install the OpenSSL library into the system, or build the OpenSSL library statically from the source with nginx by using `--with-openssl=<path>` option.

解决方法: `yum install openssl openssl-devel -y`

## 1.9 nginx软件启动过程中可能遇到的问题

1. 启动Nginx时如下报错 “`nginx:[emerg]getpwnam(“nginx”)failed`”

解答: 这是因为没有对应的Nginx服务用户, 执行`useradd nginx -s /sbin/nologin -M`创建

## 1.10 nginx软件程序目录结构

```
ls -l /application/nginx/
drwxr-xr-x 2 root root 4096 Sep  8 21:22 conf      ---配置文件保存目录
drwxr-xr-x 2 root root 4096 Sep  8 21:22 html      ---站点目录
drwxr-xr-x 2 root root 4096 Sep  8 21:52 logs      ---nginx服务相关日志文件保存目录（错误日志 访问日志）
drwxr-xr-x 2 root root 4096 Sep  8 21:22 sbin      ---服务命令保存目录（只有一个nginx命令文件）
```

### 1.10.1 nginx配置文件目录conf下的主要文件说明:

`mime.types`        ---媒体资源类型文件

`nginx.conf`        ---nginx服务主配置文件

`nginx.conf.default` ---nginx服务主配置文件的默认配置（模板）

PS: `vimdiff nginx.conf nginx.conf.default` ---可以比较两个文件之间的区别（实用性广泛）

## 1.11 测试环境模拟搭建网站

### 1.11.1 精简化nginx.conf主配置文件内容

`grep -Ev "#|^$" nginx.conf.default > nginx.conf`

PS: 过滤掉包含#号的行, 和空行

### 1.11.2 编辑主配置文件前先备份

`cp nginx.conf {,.bak}`

### 1.11.3 编辑主配置文件

`vim /application/nginx/conf/nginx.conf`

`worker_processes 1;`                               ---worker主进程数（相当于服务员）进程数越多效率越高

`events {`                                       ---事件区块开始

`worker_connections 1024;`               ---每个主进程的最大连接数（想当于客户）

`}`   ---事件区块结束

`http {`                                       ---http区块开始

`include mime.types;`                   ---支持的媒体类型文件

`default_type application/octet-stream;` ---默认支持的媒体类型文件

`sendfile on;`                           ---高效传输模式（基于磁盘block传输方式, 要快许多）

`keepalive_timeout 65;`               ---连接超时（65s内没有数据传输即断开）

`server {`                               ---第一个server区块（就是一个网站（虚拟主机），可以有多个）

`listen 80;`                       ---监听端口

`server_name www.georgekai.com;`   ---提供服务的域名

`location / {`                       ---第一个location（管理部分网站资源文件）

`root html/www;`               ---站点根目录（存放首页文件、图片...）

`index georgekai.html index.htm;` ---网站默认首页文件（空格分割多个文件）

`}`                                   ---第一个server区块结束

`error_page 500 502 503 504 404 /kai.jpg;` ---对应的http状态码, 使用kai.jpg回应客户

`location = /kai.jpg {`               ---第二个server区块...（第二个网站（虚拟主机））

`root html;`                   .....同上.....

`}`

```
}  
}
```

#### 1.11.4 创建站点目录，生成首页文件

##### 1. 创建站点目录

```
mkdir /application/nginx/html/www
```

##### 2. 生成首页文件（html语言）

```
vim georgekai.html
```

```
<html>
```

```
<meta charset="utf-8">
```

```
<head>
```

```
<title>网站标题</title>
```

```
</head>
```

```
<body>
```

网站主体内容

```
<table border=1>
```

```
<tr> <td>01</td> <td>oldboy</td> </tr>
```

```
<tr> <td>02</td> <td>oldgril</td> </tr>
```

```
<tr> <td>03</td> <td>olddog</td> </tr>
```

```
</table>
```

```
<a href="http://blog.oldboyedu.com">
```

```

```

```
</a>
```

```
</body>
```

PS：以上内容是html语言格式，也可随便写点内容做测试

#### 1.11.5 检查配置文件语法并重启nginx服务

```
/application/nginx/sbin/nginx -t
```

```
/application/nginx/sbin/nginx -s reload
```

#### 1.11.6 浏览器进行测试访问

10.0.0.7

www.georgekai.com

PS：通过域名访问，需要将以上解析关系添加到本机hosts文件

#### 1.12 nginx主配置文件区域分类及说明

##### 1. main区块

##### 2. event区块

##### 3. http区块

server区块（可以有多个），每个server相当于一个网站

location区块（可以有多个），每个location相当于server的一个模块

```

[root@www conf]# egrep -v "#|^$" nginx.conf.default
worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include      mime.types;
    default_type application/octet-stream;
    sendfile     on;
    keepalive_timeout 65;
    server {
        listen      80;
        server_name localhost;
        location / {
            root      html;

            index      index.html index.htm;
        }
        error_page   500 502 503 504 /50x.html;

        location = /50x.html {
            root      html;
        }
    }
}

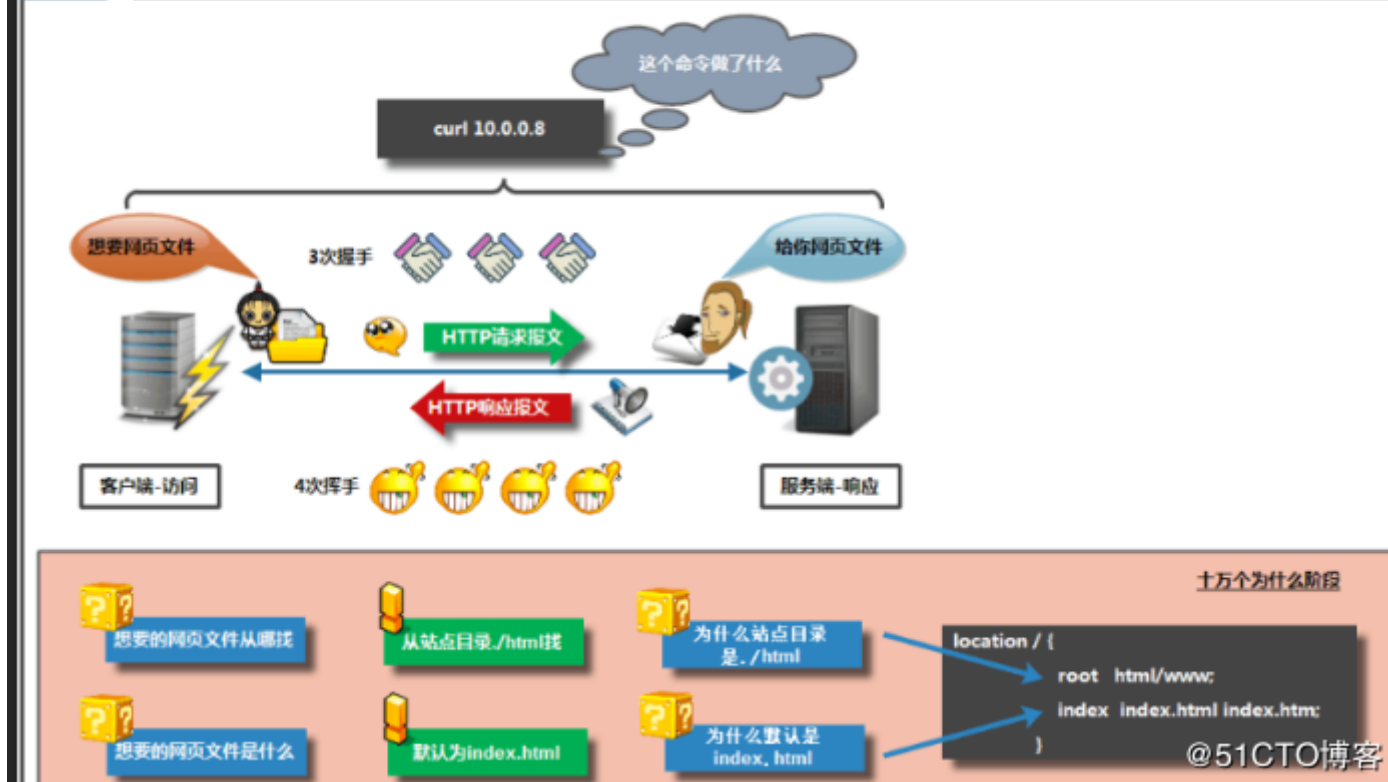
```

<- 去掉包含#号和空行的内容  
 <- worker 进程的数量  
 <- 事件区块开始  
 <- 每个 worker 进程支持的最大连接数  
 <- 事件区块结束  
 <- HTTP 区块开始  
 <- Nginx 支持的媒体类型库文件  
 <- 默认的媒体类型  
 <- 开启高效传输模式  
 <- 连接超时  
 <- 第一个 Server 区块开始，表示一个独立的虚拟主机站点  
 <- 提供服务的端口，默认 80  
 <- 提供服务的域名主机名  
 <- 第一个 location 区块开始  
 <- 站点的根目录，相当于 Nginx 的安装目录  
 <- 默认的首页文件，多个用空格分开  
 <- 第一个 location 区块结果  
 <- 出现对应的 http 状态码时，使用 50x.html 回应客户  
 <- location 区块开始，访问 50x.html  
 <- 指定对应的站点目录为 html

@51CTO博客

### 1.13 什么叫站点目录，什么首页文件

## Nginx基础知识-站点目录与首页文件概念说明



马上过年了，年假期间不会中断更新！！！！小伙伴们可以考虑用ansible部署管理Nginx  
关注微信公共号：linux运维菜鸟之旅

小伙伴们可以关注我的微信公众号：linux运维菜鸟之旅



关注“中国电信天津网厅”公众号，首次绑定可免费领2G流量，为你的学习提供流量！



版权声明：原创作品，如需转载，请注明出处。否则将追究法律责任

---