### nginx安装环境

nginx是C语言开发，建议在linux上运行，本教程使用Centos6.5作为安装环境。

* gcc

安装nginx需要先将官网下载的源码进行编译，编译依赖gcc环境，如果没有gcc环境，需要安装gcc：yum install gcc-c++

* PCRE

PCRE(Perl Compatible Regular Expressions)是一个Perl库，包括 perl 兼容的正则表达式库。nginx的http模块使用pcre来解析正则表达式，所以需要在linux上安装pcre库。

**yum install -y pcre pcre-devel**

注：pcre-devel是使用pcre开发的一个二次开发库。nginx也需要此库。

* zlib

zlib库提供了很多种压缩和解压缩的方式，nginx使用zlib对http包的内容进行gzip，所以需要在linux上安装zlib库。

**yum install -y zlib zlib-devel**

* openssl

OpenSSL 是一个强大的安全套接字层密码库，囊括主要的密码算法、常用的密钥和证书封装管理功能及SSL协议，并提供丰富的应用程序供测试或其它目的使用。

nginx不仅支持http协议，还支持https（即在ssl协议上传输http），所以需要在linux安装openssl库。

**yum install -y openssl openssl-devel**

### 编译安装

第一步：将nginx-1.8.0.tar.gz拷贝至linux服务器。

第二步：解压：tar -zxvf nginx-1.8.0.tar.gz

cd nginx-1.8.0

第三步：创建一个makefile。

参数设置如下：

1. 解压后会有一个
2. 文件
3. 参数设置如下：(注意先：mkdir **/var/temp/nginx** -p)

然后复制下面的内容，执行下面命令，会生成一个makefile文件夹，有了这个文件夹，执行make命令

./configure \

--prefix=/usr/local/nginx \

--pid-path=/var/run/nginx/nginx.pid \

--lock-path=/var/lock/nginx.lock \

--error-log-path=/var/log/nginx/error.log \

--http-log-path=/var/log/nginx/access.log \

--with-http\_gzip\_static\_module \

--http-client-body-temp-path=/var/temp/nginx/client \

--http-proxy-temp-path=/var/temp/nginx/proxy \

--http-fastcgi-temp-path=/var/temp/nginx/fastcgi \

--http-uwsgi-temp-path=/var/temp/nginx/uwsgi \

--http-scgi-temp-path=/var/temp/nginx/scgi

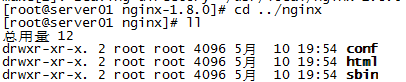
**注意：上边将临时文件目录指定为/var/temp/nginx，需要在/var下创建temp及nginx目录**

1. 编译安装

make

make install

安装成功查看安装目录 ：



## 启动nginx

cd /usr/local/nginx/sbin/

./nginx

查询nginx进程：



15098是nginx主进程的进程id，15099是nginx工作进程的进程id

**注意：执行./nginx启动nginx，这里可以-c指定加载的nginx配置文件，如下：**

**./nginx -c /usr/local/nginx/conf/nginx.conf**

**如果不指定-c，nginx在启动时默认加载conf/nginx.conf文件，此文件的地址也可以在编译安装nginx时指定./configure的参数（--conf-path= 指向配置文件（nginx.conf））**

<http://localhost/>查看欢迎页面

## 停止nginx

方式1，快速停止：

cd /usr/local/nginx/sbin

./nginx -s stop

此方式相当于先查出nginx进程id再使用kill命令强制杀掉进程。

方式2，完整停止(建议使用)：

cd /usr/local/nginx/sbin

./nginx -s quit

此方式停止步骤是待nginx进程处理任务完毕进行停止。

## 重启nginx

方式1，先停止再启动（建议使用）：

对nginx进行重启相当于先停止nginx再启动nginx，即先执行停止命令再执行启动命令。

如下：

./nginx -s quit

./nginx

方式2，重新加载配置文件：

当nginx的配置文件nginx.conf修改后，要想让配置生效需要重启nginx，使用-s reload不用先停止nginx再启动nginx即可将配置信息在nginx中生效，如下：

./nginx -s reload

## 安装测试

nginx安装成功，启动nginx，即可访问虚拟机上的nginx：

## Nginx实现虚拟机

可以实现在同一台服务运行多个网站，而且网站之间互相不干扰。

同一个服务器可能有一个ip，网站需要使用80端口。网站的域名不同。

区分不同的网站有三种方式：

1. ip区分
2. 端口区分
3. 域名区分

### Ip区分虚拟主机

需要一台服务器绑定多个ip地址。

**方法二：**

1. 将/etc/sysconfig/network-scripts/ifcfg-eth0文件复制一份，命名为ifcfg-eth0:1

修改eth0:1其中内容：

DEVICE=eth0:1

IPADDR=192.168.25.103

其他项不用修改

2、重启系统

### 配置nginx基于ip地址的虚拟主机

#### Nginx的配置文件

|  |
| --- |
| #user nobody;  worker\_processes 1;  #error\_log logs/error.log;  #error\_log logs/error.log notice;  #error\_log logs/error.log info;  #pid logs/nginx.pid;  events {  worker\_connections 1024;  }  http {  include mime.types;  default\_type application/octet-stream;  #log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '  # '$status $body\_bytes\_sent "$http\_referer" '  # '"$http\_user\_agent" "$http\_x\_forwarded\_for"';  #access\_log logs/access.log main;  sendfile on;  #tcp\_nopush on;  #keepalive\_timeout 0;  keepalive\_timeout 65;  #gzip on;  server { #一个Server就是一个虚拟主机  listen 80;  server\_name localhost;  #charset koi8-r;  #access\_log logs/host.access.log main;  location / {  root html;  index index.html index.htm;  }  }  } |

#### 基于ip的虚拟主机配置

|  |
| --- |
| server {  listen 80;  server\_name 192.168.25.141;  #charset koi8-r;  #access\_log logs/host.access.log main;  location / {  root html-141;  index index.html index.htm;  }    }  server {  listen 80;  server\_name 192.168.25.100;  #charset koi8-r;  #access\_log logs/host.access.log main;  location / {  root html-100;  index index.html index.htm;  }    } |

Nginx重新加载配置文件。

### 基于端口的虚拟主机

|  |
| --- |
| server {  listen 81;  server\_name 192.168.25.141;  #charset koi8-r;  #access\_log logs/host.access.log main;  location / {  root html-81;  index index.html index.htm;  }    }  server {  listen 82;  server\_name 192.168.25.141;  #charset koi8-r;  #access\_log logs/host.access.log main;  location / {  root html-82;  index index.html index.htm;  }    } |

### 基于域名的虚拟主机

最有用的虚拟主机配置方式。

一个域名只能绑定一个ip地址，一个ip地址可以被多个域名绑定。

2、返回ip地址

3、访问ip地址的80端口

4、响应html网页

1、根据域名换取ip地址

客户端

浏览器

www.baidu.com

百度服务器

61.135.169.121

Dns服务器

根据域名换ip地址

可以修改host文件实现域名访问。

#### 设置域名和ip的映射关系

修改window的hosts文件：（C:\Windows\System32\drivers\etc）

#### 基于域名的虚拟主机配置

|  |
| --- |
| server {  listen 80;  server\_name www.itheima.com;  #charset koi8-r;  #access\_log logs/host.access.log main;  location / {  root html-www;  index index.html index.htm;  }    }  server {  listen 80;  server\_name hehe.itheima.com;  #charset koi8-r;  #access\_log logs/host.access.log main;  location / {  root html-hehe;  index index.html index.htm;  }    } |

修改配置文件后，需要nginx重新加载配置文件。