

基于 CentOS 7 的 NGINX 安装与配置

修订记录:

版本号	修订人	修订日期	修订内容
V1.0	朱嘉宁	2019 年 1 月 14 日	初稿

目 录

一、软件版本	1
二、编译安装	1
2.1 安装依赖软件	1
2.2 新建用户	1
2.3 安装 Nginx	1
2.4 配置环境变量	1
2.5 启动、停止和重启 nginx	1
三、基本配置	1
3.1 搭建服务	1
3.2 多服务设置	2
3.3 禁用 IP 登陆	2
四、高级设置	3
4.1 反向代理与负载均衡	3
4.2 动静分离	3

一、软件版本

Nginx 版本: 1.14.2

二、编译安装

2.1 安装依赖软件

需要的软件有三个: zlib、pcre、openssl, 通过编译安装或是 yum 安装均可。编译安装注意修改类库位置。yum 安装的情况下, openssl-devel 里包含 zlib-devel 和 pcre-devel, 可以直接使用命令:

```
yum install -y openssl-devel
```

进行安装。

此外, 编译过程需要 gcc-c++, 使用命令:

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

2.2 新建用户

为保证安全, 建议用专门的用户组和用户来执行程序。使用命令:

```
groupadd nginx
```

```
useradd -g nginx -s /sbin/nologin -M nginx
```

2.3 安装 Nginx

完成上一步后解压安装包, 进入解压后的目录进行编译。在此我们设置用户组和用户, 使用命令:

```
./configure \  
--group=nginx \  
--user=nginx \  
--with-http_ssl_module
```

编译完成后进行安装, 使用命令:

```
make && make install
```

安装好的 Nginx 位于 /usr/local/nginx 下。

2.4 配置环境变量

为启停方便, 可以把 Nginx 的有个信息写入环境变量, 新建 /etc/profile.d/nginx.sh 文件, 写入以下内容:

```
NGINX_HOME=/usr/local/nginx  
PATH=$NGINX_HOME/sbin:$PATH
```

然后使用命令:

```
source /etc/profile
```

来更新环境变量。

2.5 启动、停止和重启 nginx

启动 Nginx: nginx

停止 Nginx: nginx -s stop

重启 Nginx: nginx -s reload

三、基本配置

配置文件为 conf/nginx.conf, 修改这个文件即可。也可以新建文件然后在 nginx.conf 中用 include 来引用。修改 nginx.conf 后, 必须重启 Nginx 才能使修改生效。

3.1 搭建服务

在 nginx.conf 中有 http 段, 其中有 server 段。一个 server 段都是一个服务, 段中内容主要包含要监听的端口、域名和服务器中的网页文件地址。来看下面的示例:

原文	注释
server { listen 80;	监听 80 端口

server_name localhost;	域名
location / {	
root html;	目录地址
index index.html index.htm;	首页文件
}	

3.2 多服务设置

往往在一个 IP 地址下会有不同的服务，我们可以通过在 http 段中添加多个 server 段来实现。只要两个 server 段中的监听端口或域名有一个不同，那它们就是两个不同的服务。示例如下：

```
server {
    listen 80;
    server_name localhost;
    location / {
        root html/1;
        index index.html index.htm;
    }
}

server {
    listen 81;
    server_name localhost;
    location / {
        root html/2;
        index index.html index.htm;
    }
}

server {
    listen 81;
    server_name 3.com;
    location / {
        root html/1;
        index index.html index.htm;
    }
}
```

3.3 禁用 IP 登陆

为防止 IP 被未授权域名绑定，通常会设置禁止通过 IP 访问，而只允许通过设定好的域名访问。可以在 nginx.conf 中添加一个 server 段：

```
server {
    listen 9000;
    server_name _;
    return 403;
}
```

四、高级设置

4.1 反向代理与负载均衡

通常，反向代理与负载均衡会一同出现在我们的服务器配置中。反向代理用于隐藏后面的真正的服务器，负载均衡用于平衡服务器的压力。它们使用的是 `upstream` 模块，安装时默认安装。来看下面的示例：

```
upstream test {
    server 192.168.101.31 weight=1 max_fails=2 fail_timeout=30s;
    server 192.168.101.32 weight=1 max_fails=2 fail_timeout=30s;
}
server {
    listen 80;
    server_name    localhost;
    location / {
        proxy_pass    http://test;
    }
}
```

示例中，定义了一个 `upstream` 段，取名 `test`。段内有 192.168.101.31 和 192.168.101.32 两台服务器（默认 80 端口）。后面接着的是权重、失败次数和失败时长。

后面的 `server` 段中，`location` 的设置不再是本地路径，而是用 `proxy_pass` 来表示代理，后面接 “`http://test;`”，表示用 `test` 里面的地址。

4.2 动静分离

```
upstream dynamic_server {
    server 192.168.101.31:8080 weight=1 max_fails=2 fail_timeout=30s;
    server 192.168.101.32:8080 weight=1 max_fails=2 fail_timeout=30s;
}
upstream static_server {
    server 127.0.0.1:80;
}
server {
    listen 80;
    server_name    localhost;
    location ~ .*\. (js|css|html|html|gif|jpg|jpeg|png|bmp|swf|ioc|rar|zip|txt|flv|mid|doc|ppt|pdf|xls|mp3|wma)$ {
        proxy_pass    http://static_server;
    }
    location ~ .*$ {
        proxy_pass    http://dynamic_server;
    }
}
```