**安装依赖库（ubtunu安装）**

sudo apt-get install gcc

sudo apt-get install libpcre3 libpcre3-dev

sudo apt-get install zlib1g zlib1g-dev

sudo apt-get install openssl

sudo apt-get install libssl-dev

**安装nginx**

下载nginx

wget <http://nginx.org/download/nginx-1.8.0.tar.gz>

解压

tar -xvzf nginx-1.8.0.tar.gz

进入nginx目录

编译配置（默认配置）

./configure

编译

make

如果出现”-Werror=implicit-fallthrough=”错误，请使用如下命令编译：

make CFLAGS='-Wno-implicit-fallthrough'

安装

sudo make install

**运行命令**

nginx默认安装时启动文件是/usr/local/nginx/sbin/nginx

**启动**

使用默认配置文件启动

sudo /usr/local/nginx/sbin/nginx

使用指定的配置文件启动

sudo /usr/local/nginx/sbin/nginx -c xxx.conf

启动默认使用80端口

**查看是否启动**

进程查看

ps aux|grep nginx

curl访问

curl -v -o /dev/null <http://localhost/index.html>

**停止**

sudo nginx -s stop # 强制停止

sudo nginx -s quit # 处理玩所有连接后停止

sudo nginx -s reload # 重启

如启动时指定了配置文件，在停止时也应指定，如：

sudo nginx -s stop -c xxx.conf

**编译配置**

**添加模块**

./configure --add-module=Path;

Path第三方模块的源码路径

**安装路径**

./configure --prefix=Path

默认值是/usr/loacl/nginx/

**Nginx配置**

Nginx配置文件结构如下：

... #全局块

events { #events块

...

}

http #http块

{

... #http全局块

server #server块

{

... #server全局块

location [PATTERN] #location块

{

...

}

location [PATTERN]

{

...

}

}

server

{

...

}

... #http全局块

}

**全局域（根）配置**

进程配置

worker\_processer number | auto

能够启动的worker进程的数目，一般与cpu核心数相等，或使用auto自动分配

日志配置

error\_log file | stderr level

File：日志文件路径，默认logs/error.log

Stderr：标准错误

Level：记录级别，具有如下几个级别debug|info|notice|warn|error|crit|alert|emerg

**http配置**

http

{

# include 导入另一个配置文件

include mime.types; # 文件扩展名与文件类型映射表

default\_type application/octet-stream; # 默认文件类型，默认为text/plain

keepalive\_timeout 65; # 连接超时时间，默认为75s，可以在http，server，location块。

upstream mysvr { # 配置上游服务器，用于反向代理

server 127.0.0.1:7878; # 上游服务器1

server 192.168.10.120:3333; # 上游服务器2

server 192.168.10.121:3333 backup; # 备用服务器

}

error\_page 404 https://www.baidu.com; #错误页

server {

keepalive\_requests 120; # 单连接请求上限次数

listen 4545; # 监听端口

server\_name 127.0.0.1; # 设置当前服务器的监听的地址

# location [=|~|~\*|^~|@] uri {...}

# =完全匹配，~为区分大小写，~\*不区分大小写

# uri可以为正则或url

# 如 location = /api/abc { ... }，完全匹配 /api/abc

# 如 location /api/ { ... }，匹配 /api/\*.\*

location ~\* /abc/ {

# root /dir1; # 根目录，如 /abc/1 将映射到 /dir1/abc/1

# alias /dir2; # 如 /abc/kkk 将映射到 /dir2/kkk 文件

# index vv.txt; #设置默认页

proxy\_pass http://mysvr; # 请求转向 mysvr 定义的服务器列表

deny 127.0.0.1; #拒绝的ip

allow 172.18.5.54; #允许的ip

}

}

}

**events配置**

events {

worker\_connections 1024; # worker进程的最大连接数

}

**变量**

配置文件可以使用变量，一些 常用的变量如下：

$uri ：当前请求的uri，不含?后面的参数

$args：当前请求的参数，即?后面的字符串

$request\_uri：当前请求的完整uri

$arg\_xxx：当前请求的参数值，xxx为参数名

$http\_xxx：当前请求的头部对应的值，xxx为头部名

设置变量

set $varname value;

**一个反向代理配置示例**

events {

  worker\_connections  1024;  ## Default: 1024

}

http

{

default\_type application/octet-stream; # 默认文件类型，默认为text/plain

keepalive\_timeout 65;  # 连接超时时间，默认为75s，可以在http，server，location块。

    upstream mysvr {    # 配置上游服务器，用于反向代理

      server 192.168.31.169:8080;        # 上游服务器1

    }

    server {

        keepalive\_requests 120; # 单连接请求上限次数

        listen 80;            # 监听端口

        server\_name localhost; # 监听地址

        # 将所有请求转发至mysvr提供的服务器

        location / {

            proxy\_pass http://mysvr;    # 请求转向 mysvr 定义的服务器列表

        }

    }

}