



链滴

Centos 安装 Nginx 反向代理 Tomcat 配置 Https

作者: [pencilso](#)

原文链接: <https://ld246.com/article/1531880146749>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

安装Nginx

```
yum install nginx
```

创建log目录

```
mkdir /usr/share/nginx/log
```

启动命令

```
service nginx start
```

关闭命令

```
service nginx stop
```

重载配置

```
service nginx reload
```

更改配置文件

nginx 配置文件在 /etc/nginx/nginx.conf

更改了配置文件后 需要重载一下配置

```
##### 每个指令必须有分号结束。#####  
#user administrator administrators; #配置用户或者组，默认为nobody nobody。  
#worker_processes 2; #允许生成的进程数，默认为1  
#pid /nginx/pid/nginx.pid; #指定nginx进程运行文件存放地址  
error_log log/error.log debug; #制定日志路径，级别。这个设置可以放入全局块，http块，serve  
块，级别以此为：debug|info|notice|warn|error|crit|alert|emerg  
events {  
    accept_mutex on; #设置网路连接kulaidian序列化，防止惊群现象发生，默认为on  
    multi_accept on; #设置一个进程是否同时接受多个网络连接，默认为off  
    #use epoll; #事件驱动模型，select|poll|kqueue|epoll|resig|dev/poll|eventport  
    worker_connections 1024; #最大连接数，默认为512  
}  
http {  
  
    #开启GZIP压缩  
    gzip on;  
    gzip_min_length 1k;  
    gzip_buffers 4 16k;
```

```

#gzip_http_version 1.0;
gzip_comp_level 4;
gzip_types text/plain application/javascript text/css application/xml text/javascript applic
tion/x-httpd-php image/jpeg image/gif image/png;
gzip_vary off;
gzip_disable "MSIE [1-6]\.";

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
#access_log off; #取消服务日志
log_format myFormat '$remote_addr-$remote_user [$time_local] $request $status $body_
ytes_sent $http_referer $http_user_agent $http_x_forwarded_for'; #自定义格式
access_log log/access.log myFormat; #combined为日志格式的默认值
sendfile on; #允许sendfile方式传输文件, 默认为off, 可以在http块, server块, location块。
sendfile_max_chunk 100k; #每个进程每次调用传输数量不能大于设定的值, 默认为0, 即不设上
。
keepalive_timeout 65; #连接超时时间, 默认为75s, 可以在http, server, location块。

error_page 404 https://www.baidu.com; #错误页
# 配置拦截80端口 然后转发到https
server {
    listen 80;
    server_name www.example.com example.com pro.example.com;#多域名用空格隔开 这里
举例三个域名
    rewrite ^(.*)$ https://$host$1 permanent;
}
#监听来着prod.example.com 的请求 然后再代理向Tomcat的端口 这里我的Tomcat端口为8080
server {
    listen 443;
    server_name prod.example.com;#监听prod的二级域名
    ssl on;
    ssl_certificate /usr/local/centos_x64/nginx_https/214544061980920.pem;
    ssl_certificate_key /usr/local/centos_x64/nginx_https/214544061980920.key;
    ssl_session_timeout 5m;
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!M
5:!ADH:!RC4;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers on;
    location / {
        proxy_pass http://127.0.0.1:8080; #来自请求交给tomcat处理
        proxy_redirect off;
        proxy_set_header Host $host; #后端的Web服务器可以通过X-Forwarded-For>获取用
真实IP
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        client_max_body_size 10m; #允许客户端请求的最大单文件字节数
        client_body_buffer_size 128k; #缓冲区代理缓冲用户端请求的最大字节数
        proxy_connect_timeout 90; #nginx跟后端服务器连接超时时间(代理连接超时)
        proxy_read_timeout 90; #连接成功后, 后端服务器响应时间(代理接收超时)
        proxy_buffer_size 4k; #设置代理服务器 (nginx) 保存用户头信息的缓冲区大小
        proxy_buffers 6 32k; #proxy_buffers缓冲区, 网页平均在32k以下的话>, 这样设置
        proxy_busy_buffers_size 64k; #高负荷下缓冲大小 (proxy_buffers*2)
        proxy_temp_file_write_size 64k; #设定缓存文件夹大小, 大于这个值, 将从upstream服务
传

```

```

    }
}
#监听来自 www.example.com example.com 两个域名的请求 转向到静态的Html文件目录
server {
    listen 443;
    server_name www.example.com example.com;
    ssl on;
    ssl_certificate /usr/local/centos_x64/nginx_https/214544061980920.pem;
    ssl_certificate_key /usr/local/centos_x64/nginx_https/214544061980920.key;
    ssl_session_timeout 5m;
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!M
5:!ADH:!RC4;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers on;
    location / {
        root /web/; #默认Html目录
        index index.html index.htm;#首页文件
    }
}
}
}

```