# nginx安装部署

安装nginx，同时配置nginx作为tomcat的代理 操作系统：SLE 11 sp3

目前确权平台和监控大屏的静态文件在183.131.202.165，/var/www目录下

## 安装nginx

目前官网只给出了两种安装方案：(1)系统自带包管理安装; (2)从源码编译。两种方案都需要root权限。 按照方案(1)进行安装。

1. 配置dns
   服务器一开始没有配置dns地址，无法使用包管理工具
   在`/etc/resolv.conf`中添加一行：`nameserver 114.114.114.114`
2. 通过包管理工具安装nginx

> SLE的包管理工具是zypper，一般来说用linux的保管里安装软件是很简单的，但是服务器上zypper的配置有点奇怪，因此过程稍微复杂了一点。具体如下:

```
# 添加repo
$ zypper addrepo
https://download.opensuse.org/repositories/server:http/SLE_11_SP4/server:http.repo
$ zypper refresh

# 这个时候直接安装会有问题，有些基础包的repo被配置在cd上了
$ zypper install nginx
.......
Failed to mount cd:///?devices=/dev/disk/by-id/scsi-1ATA_QEMU_DVD-ROM_QM00001,/dev/sr0 on :
Mounting media failed
Please insert medium [SUSE-Linux-Enterprise-Server-11-SP3 11.3.3-1.138] #1 and type 'y' to
continue or 'n' to cancel the operation. [yes/no] (no):
......

# 查看可用的repo
$ zypper lr
# | Alias   | Name  | Enabled | Refresh
--+---------+-------+---------+--------
1 | SUSE-Linux-Enterprise-Server-11-SP3 11.3.3-1.138 | SUSE-Linux-Enterprise-Server-11-SP3
11.3.3-1.138 | Yes     | No
2 | server_http                                      | Webservers and tools around it
(SLE_11_SP4)      | Yes     | No

# 就是“SUSE-Linux-Enterprise-Server-11-SP3 11.3.3-1.138”这个repo，需要禁用掉
$ zypper mr -d "SUSE-Linux-Enterprise-Server-11-SP3 11.3.3-1.138"

# 再次查看
# 可以看到Enabled状态已经是No了
$ zypper lr
# | Alias   | Name   | Enabled | Refresh
--+---------+--------+---------+--------
```

```
1 | SUSE-Linux-Enterprise-Server-11-SP3 11.3.3-1.138 | SUSE-Linux-Enterprise-Server-11-SP3
11.3.3-1.138 | No     | No
2 | server_http                                     | Webservers and tools around it
(SLE_11_SP4)      | Yes     | No

# 再次尝试安装nginx
# 这次会提示某个dependency无法获取（应该是之前在光盘里的...）
# 但是似乎不安装直接(选2)似乎也能安装成功，就暂时没管了...
$ zypper install nginx
......
Problem: nothing provides libgd.so.2()(64bit) needed by nginx-1.13.6-89.1.x86_64
 Solution 1: do not install nginx-1.13.6-89.1.x86_64
 Solution 2: break nginx-1.13.6-89.1.x86_64 by ignoring some of its dependencies
......
```

# nginx基本操作

操作nginx需要root权限

```
# 启动
nginx
# 关闭
nginx -s quit
# 强制关闭
nginx -s stop
# 重新加载配置
nginx -s reload
```

# nginx配置

- 根据官方文档，nginx的配置文件可能在三个文件夹下`/usr/local/nginx/conf`、`/etc/nginx`或`/usr/local/etc/nginx.`。
  - 当前操作系统下，配置文件在`/etc/nginx`下
- 该目录下已有一个默认的配置文件和备份：`nginx.conf.default`，`nginx.conf`
  - 其中已经有了部分配置，nginx可以在一个配置文件中配置多个服务
  - 每支持一个服务就需要在`http`下添加一个`server`
  - 服务监听的端口，静态文件路径等信息在`server`下进行配置

```
http {
    server {
    }
}
```

配置项的相关说明详见文档

## 静态文件

- 在`server`下配置

```
server {
    # 监听端口
    listen 80;
    # server名称
    server_name localhost;

    location / {
        # 静态文件根目录
        root   /var/www/server;
        # 访问该路径时，需要寻找的index文件，这项可以省略
        # `index`不配置时，默认搜索根目录下的index.html, index.htm, index.php
        index  index.html index.htm;
    }
}
```

- nginx运行时会有至少两个进程同时运行——一个master和至少一个worker
  - master进行调度，而worker是实际服务的提供者
  - 可以看到，master以root用户运行，而worker以nginx用户运行（在centOS下似乎是nobody）

```
$ ps -ef | grep nginx
root      8958     1  0 Feb28 ?        00:00:00 nginx: master process
nginx    28252  8958  0 15:53 ?        00:00:00 nginx: worker process
root     31092 28825  0 19:15 pts/3    00:00:00 grep nginx
```

- 因为用户权限的关系，除了修改配置文件，还需要设置静态文件根目录的所有权和权限
  - nginx需要至少拥有文件的读权限，**否则服务会出现`403`错误**
  - 为了配置方便，将所有文件的权限设为755，同时所有权转到nginx用户下
    - 不过似乎可以进行更细致的配置，文件夹权限为755，文件权限为644
  - 命令如下
    - `chmod -R 755 /var/www/server`
    - `chown -R nginx:nginx /var/www/server`

## 作为tomcat代理

- 目前，tomcat监听8070端口，所有对`/ownership_webtransfer`的访问都需要转发到tomcat进行处理
- 配置如下：

```
    location /ownership_webtransfer {
        proxy_pass http://127.0.0.1：8070/ownership_webtransfer;
    }
```

- 这样配置之后遇到一个问题：所有经过nginx转发的post请求都会返回`403`错误
  - 经过排查之后发现是由请求`header`中的`Origin`字段引起的
    - ***可能是一个跨域问题，但不确定***
  - 需要在转发时把`header`中的`Origin`字段去掉

        - `proxy_set_header Origin "";`
        - 头字段被设置为`""`时，这个字段将不会被传递到被代理服务
            - If the value of a header field is an empty string then this field will not be passed to a proxied server
- 另外，nginx还可以为server设置group用作负载均衡
    - 这里仅简单地用作别名，具体说明见官网文档
    - 这段配置与`server`同级
    - ```
      # upstream $group_name ...
      upstream tomcat {
          server localhost:8070 max_fails=3 fail_timeout=5;
      }
      ```
- 代理最终配置如下：

```
location /ownership_webtransfer {
    proxy_set_header Origin "";
    proxy_pass http://tomcat/ownership_webtransfer;
}
```

- 代理配置完成之后所有来自浏览器的请求都由nginx处理，tomcat只接受来自nginx的请求
    - 修改配置文件`{tomcat_home}/conf/server.xml`，在`<Connector>`中添加`address`限制
    - ```
      <Connector port="8070" protocol="HTTP/1.1"
                 connectionTimeout="20000"
                 address="127.0.0.1"
                 redirectPort="8443" />
      ```

## ssl

默认配置文件中已经有了被注释掉的例子，生成证书然后直接填空就可以了

- 证书生成
    - nginx只接受certification和key的组合，不接受keystore形式

```
mkdir /etc/nginx/ssl
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/ssl/nginx.key -out
/etc/nginx/ssl/nginx.crt
```

- 配置证书

```
server {
    listen       8080 ssl;
    server_name  localhost;

    ssl_certificate      /etc/nginx/ssl/cert.crt;
```

```
    ssl_certificate_key  /etc/nginx/ssl/cert.key;

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;

    ssl_session_cache    shared:SSL:1m;
    ssl_session_timeout  5m;

    ssl_ciphers  HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers  on;

    ##############
    # other confs

}
```

## server部分完整配置

```
upstream tomcat {
    server localhost:8070 max_fails=3 fail_timeout=5;
}

server {
    listen       8080 ssl;
    server_name  localhost;

    ssl_certificate      /etc/nginx/ssl/cert.crt;
    ssl_certificate_key  /etc/nginx/ssl/cert.key;

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;

    ssl_session_cache    shared:SSL:1m;
    ssl_session_timeout  5m;

    ssl_ciphers  HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers  on;

    location / {
        root   /var/www;
        index  index.html index.htm;
    }

    location /ownership_webtransfer {
        proxy_set_header Origin "";
        proxy_pass http://tomcat/ownership_webtransfer;
    }
}
```

## log

详细配置可以参考这里

- 默认配置下log位于`/var/log/nginx/`
  - 可以在`nginx.conf`中配置，全局或者`http`字段中都可以
    - `error_log $log_file $log_level`
    - `access_log $log_file $log_level`
- log格式默认为

```
log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                  '$status $body_bytes_sent "$http_referer" '
                  '"$http_user_agent" "$http_x_forwarded_for"';
```