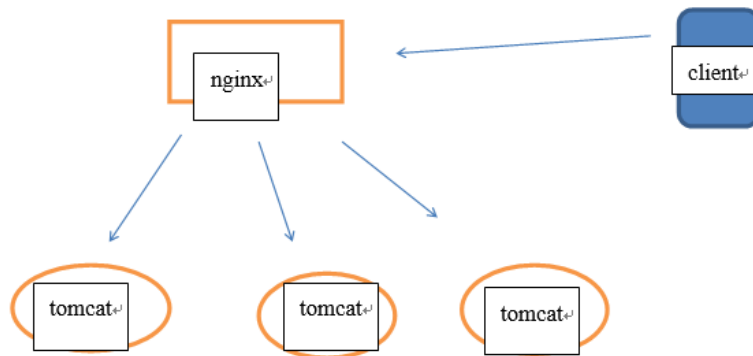


## Nginx

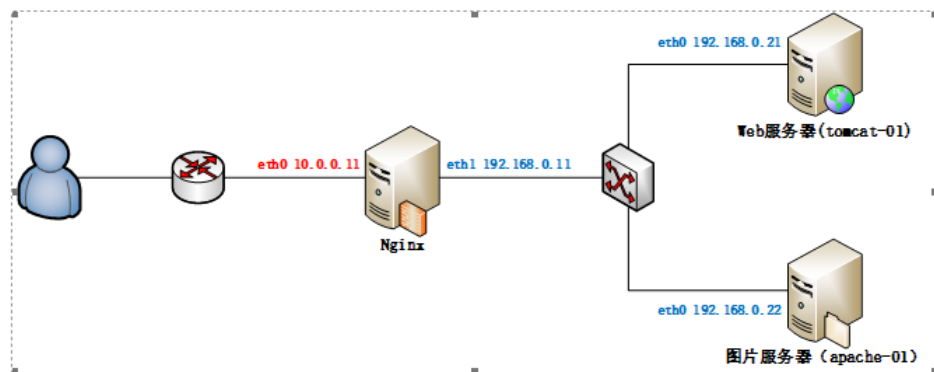
### 1.nginx简介

Nginx是一个自由、开源、高性能及轻量级的HTTP服务器及反转代理服务器。Nginx以其高性能、稳定、功能丰富、配置简单及占用系统资源少而著称。Nginx 超越 Apache 的高性能和稳定性，使得国内使用 Nginx 作为 Web 服务器的网站也越来越多。

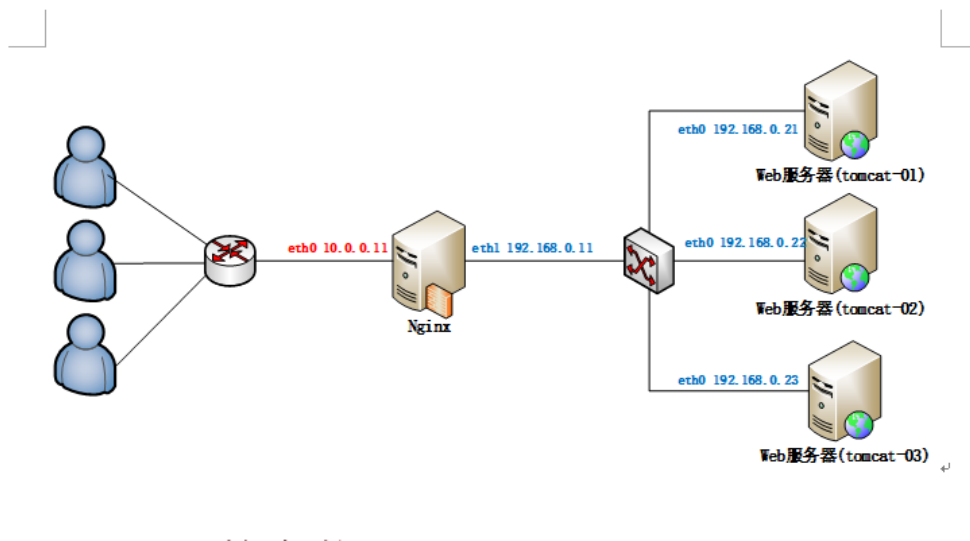


### 2. 基础功能

1. 反向代理（Reverse Proxy）方式是指以代理服务器来接受internet上的连接请求，然后将请求转发给内部网络上的服务器，并将从服务器上得到的结果返回给internet上请求连接的客户端，此时代理服务器对外就表现为一个服务器。



2. 负载均衡（Load Balance），是指建立在现有网络结构之上，并提供了一种廉价有效透明的方法扩展网络设备和服务器的带宽、增加吞吐量、加强网络数据处理能力、提高网络的灵活性和可用性。其原理就是数据流量分摊到多个服务器上执行，减轻每台服务器的压力，多台服务器共同完成工作任务，从而提高了数据的吞吐量。



### 3. 优势

1. Nginx专为性能优化而开发, 性能是其最重要的考量, 实现上非常注重效率。有报告表明能支持高达 50,000 个并发连接数。
2. Nginx具有很高的稳定性。其它HTTP服务器, 当遇到访问的峰值, 或者有人恶意发起慢速连接时, 也很可能会导致服务器物理内存耗尽频繁交换, 失去响应, 只能重启服务器。例如当前apache一旦上到200个以上进程, web响应速度就明显非常缓慢了。而Nginx采取了分阶段资源分配技术, 使得它的CPU与内存占用率非常低。
3. nginx官方表示保持10,000个没有活动的连接, 它只占2.5M内存, 就稳定性而言, nginx比其他代理服务器更胜一筹。
4. Nginx支持热部署。它的启动特别容易, 并且几乎可以做到7\*24不间断运行, 即使运行数个月也不需要重新启动。你还能够在不间断服务的情况下, 对软件版本进行进行升级。
5. Nginx采用C进行编写, 不论是系统资源开销还是CPU使用效率都高很多。

### 4. 安装

#### 1. 下载nginx

官网: <http://nginx.org/>

#### 2. 上传并解压nginx

```
tar -zxvf nginx-1.8.1.tar.gz -C /usr/local/src
```

#### 3. 编译nginx

进入到nginx源码目录 `cd /usr/local/src/nginx-1.8.1`

#### 4. 检查安装环境,并指定将来要安装的路径

```
./configure --prefix=/usr/local/nginx
```

缺包报错 ./configure: error: C compiler cc is not found

## 5. 使用YUM安装缺少的包

```
yum -y install gcc pcre-devel openssl openssl-devel
```

## 6. 编译安装

```
make && make install
```

## 7. 安装完后测试是否正常：

```
cd usr/local/nginx/sbin  
./nginx
```

查看端口是否有nginx进程监听

```
netstat -ntlp | grep 80
```

如果您正在运行防火墙，请运行以下命令以允许HTTP和HTTPS通信：

```
sudo firewall-cmd --permanent --zone=public --add-service=http  
sudo firewall-cmd --permanent --zone=public --add-service=https  
sudo firewall-cmd --reload
```

## 8.Nginx的启动，重启，关闭

### 1. 启动

```
cd usr/local/nginx/sbin  
./nginx
```

### 2. 重启

```
??????nginx  
    kill -HUP ??????????????  
    ????  
    cd /usr/local/nginx/sbin  
    ./nginx -s reload  
  
?????????  
    nginx -t -c /usr/local/nginx/conf/nginx.conf  
    ??  
    cd /usr/local/nginx/sbin  
    ./nginx -t
```

## 3. 关闭

```
??nginx???
ps -ef | grep nginx
???? kill -QUIT ????
???? kill -TERM ????
???? kill -9 nginx
?nginx.conf???pid?????????logs???
kill -???? '/usr/local/nginx/logs/nginx.pid'
```

## 5. 配置nginx

### 1. 配置反向代理

#### 1.修改nginx配置文件

```
server {
    listen      80;
    server_name  nginx-01.itcast.cn;    #nginx 所在服务器的主机名
    #反向代理的配置
    location / {
        #拦截所有请求
        root html;
        proxy_pass http://192.168.0.21:8080;    #这里是代理走向的目标服务器：tomcat
    }
}
```

#### 2.启动tomcat-01上的tomcat

#### 3.启动nginx-01上的nginx ./nginx

重启: kill -HUP cat /usr/local/nginx/logs/nginx.pid 参考网址:<http://www.cnblogs.com/jianxie/p/3990377.html>

### 2. 动静分离

↵  
#动态资源 index.jsp

```
location ~ .*\.(\.jsp|do|action)$ {  
    proxy_pass http://tomcat-01.itcast.cn:8080;  
}
```

↵  
#静态资源

```
location ~ .*\.(\.html|js|css|gif|jpg|jpeg|png)$ {  
    expires 3d;  
}
```

### 3. 负载均衡

在http这个节下面配置一个叫upstream的，后面的名字可以随意取，但是要和location下的proxy\_pass [http://](#)后的保持一致。

```
http {  
    是在 http 里面的, 已有 http, 不是在 server 里, 在 server 外面  
    upstream tomcats {  
        server shizhan02:8080 weight=1;#weight 表示多少个  
        server shizhan03:8080 weight=1;  
        server shizhan04:8080 weight=1;  
    }  
    #卸载 server 里  
    location ~ .*\.(\.jsp|do|action)$ {  
        proxy_pass http://tomcats;    #tomcats 是后面的 tomcat 服务器组的逻辑组号  
    }  
}
```