

实验四：代理服务器

一、实验目的

- 1、理解正向代理服务器和反向代理服务器的工作原理；
- 2、掌握搭建反向代理服务器的方法；
- 3、掌握使用方向代理实现负载均衡。

二、实验学时

2 学时

三、实验类型

创新性

四、实验需求

1、硬件

每人配备计算机 1 台，不低于双核 CPU、8G 内存、500GB 硬盘。

2、软件

Windows 操作系统，安装 VirtualBox 虚拟化软件，安装 Putty 管理终端软件。

3、网络

计算机使用固定 IP 地址接入局域网，并支持对互联网的访问，虚拟主机可通过 NAT 方式访问互联网。

4、工具

无。

五、实验任务

- 1、完成 Nginx 反向代理服务器部署安装；
- 2、完成反向代理服务器实现内部 Web 服务器的负载均衡；
- 3、完成对内部 Web 服务器的性能测试。

六、实验内容及步骤

1、应用场景设计

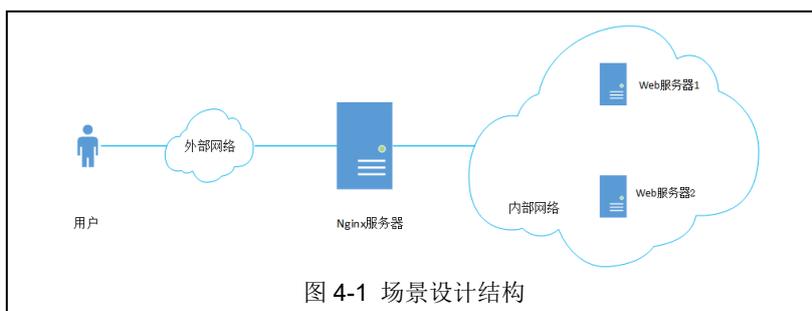
(1) 场景描述

本实验使用 3 台 CentOS 虚拟主机，其中 2 台为 Web 服务器，进行 http 业务发布，1 台为 Nginx 服务器，进行代理服务实现，并通过 VirtualBox 软件中网卡模式的选择，实现内外网的分类，其场景应用结构如图 4-1 所示。

3 台 CentOS 虚拟主机的 IP 地址地址配置如表 4-1 所示。

表 4-1 虚拟主机 IP 地址配置

序号	虚拟主机名称	网络配置	网关	网卡模式	描述
1	Web 服务器 1	10.0.2.14/24	10.0.2.1	桥接	业务网卡
2	Web 服务器 2	10.0.2.15/24	10.0.2.1	桥接	业务网卡
3	Nginx 服务器	172.16.124.102/24	172.16.124.1	桥接	代理服务
4		10.0.2.102/24	10.0.2.1		内部业务通信



(2) 网络配置

Nginx 服务器需要配置 2 个 IP 地址，一个用于代理服务，供外网访问；一个用于内部业务通信，本次实验的虚拟主机网卡名为“ifcfg-enp0s3”。

其“代理服务”网络配置如下所示，如图 4-2 所示。

```
#vi /etc/sysconfig/network-scripts/ifcfg-enp0s3
TYPE=Ethernet
BOOTPROTO=static
IPADDR=172.16.124.102
PREFIX=24
GATWAY=172.16.124.1
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
NAME=enp0s3
DEVICE=enp0s3
ONBOOT=yes
PROXY_METHOD=none
BROWSER_ONLY=no
```

```
[root@localhost ~]# vi /etc/sysconfig/network-scripts/ifcfg-enp0s3
TYPE=Ethernet
BOOTPROTO=static
IPADDR=172.16.124.102
PREFIX=24
GATWAY=172.16.124.1
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
```

图 4-2 配置 Nginx 服务器 IP-1

将网卡配置文件 ifcfg-enp0s3 进行复制并重命名为 ifcfg-enp0s3:1，编辑 ifcfg-enp0s3:1 文件进行“内部业务通信”网络配置，其操作命令如下。

```
# cd /etc/sysconfig/network-scripts/
# cp ifcfg-enp0s3 ifcfg-enp0s3:1
```

```
# vi ifcfg-enp0s3:1
TYPE=Ethernet
BOOTPROTO=static
IPADDR=10.0.2.102
PREFIX=24
GATWAY=10.0.2.1
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
NAME=enp0s3
DEVICE=enp0s3
ONBOOT=yes
PROXY_METHOD=none
BROWSER_ONLY=no
##启动网卡
# ifup ifcfg-enp0s3:1
##重启网络
# systemctl restart network
```

Nginx 服务器配置完成后，通过如下命令查看 IP 地址配置，网络配置结果如图 4-3 所示。

```
# ip addr
```

```
[root@localhost network-scripts]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:9f:0a:78 brd ff:ff:ff:ff:ff:ff
    inet 172.16.124.102/24 brd 172.16.124.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet 10.0.2.102/24 brd 10.0.2.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::1ecc:b38e:8999:f845/64 scope link
        valid_lft forever preferred_lft forever
```

图 4-3 Nginx 服务器网络配置

参照表 4-1，完成 3 台虚拟化主机的部署，完成 2 台 Web 服务器的网络配置，并将配置命令填写到表 4-2 中。

表 4-2 Web 服务器网络配置

--

2、部署 Apache 服务器

在 2 台 CentOS 的 Web 服务器上安装 Apache 服务，可参照《实验三：网站服务器》实验指导书中 Apache 安装过程，进行软件安装、服务启动操作。

3、部署 Nginx 服务器

(1) 配置安装源

本次实验获取 Nginx 软件包，是通过引用官方的 RPM 源进行获取，其操作命令如下，配置过程如图 4-4 所示。

```
##导入 Nginx 官方 GPG 密码，如果没有该密码，下载 RPM 包是会出现错误
# rpm --import http://nginx.org/keys/nginx_signing.key
##加入 Nginx 的 RPM 包软件
# rpm -ivh http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7ngx.noarch.rpm
```

```
[root@MiWiFi-R3P-srv ~]# rpm --import http://nginx.org/keys/nginx_signing.key
[root@MiWiFi-R3P-srv ~]# rpm -ivh http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7ngx.noarch.rpm
Retrieving http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7ngx.noarch.rpm
Preparing... ##### [100%]
Updating / installing...
 1:nginx-release-centos-7-0.el7ngx ##### [100%]
```

图 4-4 配置 Nginx 安装源

(2) 安装 Nginx

加入 Nginx 的安装源后，输入以下命令进行 Nginx 的安装，其安装过程如图 4-5 所示。

```
# yum install nginx
```

```
[root@MiWiFi-R3P-srv ~]# yum install nginx
Loaded plugins: fastestmirror
base | 3.6 kB 00:00
extras | 3.4 kB 00:00
nginx | 2.9 kB 00:00
updates | 3.4 kB 00:00
(1/2): extras/7/x86_64/primary_db | 166 kB 00:00
(2/2): nginx/x86_64/primary_db | 31 kB 00:03
Loading mirror speeds from cached hostfile
 * base: mirrors.163.com
 * extras: mirrors.tuna.tsinghua.edu.cn
 * updates: mirror.lzu.edu.cn
Resolving Dependencies
--> Running transaction check
--> Package nginx.x86_64 1:1.12.2-1.el7_4.ngx will be installed
--> Finished Dependency Resolution
```

图 4-5 安装 Nginx

(3) Nginx 服务配置

① 防火墙配置

开启防火墙的 TCP/80 端口，从而可远程访问 Nginx 的 Web 服务，其命令如下所示。

```
##添加 TCP/80 端口永久访问
# firewall-cmd --zone=public --add-port=80/tcp --permanent
##重新加载 firewalld 防火墙规则
# firewall-cmd --reload
```

② 服务启动

完成安装后，需重启 Nginx 服务，其操作命令如下，

```
##启动 Nginx 服务
# systemctl start nginx
```

```
##设置 Nginx 服务开机自启动
# systemctl enable nginx
```

(4) Nginx 测试

用户在本机浏览器中输入 `http://IP 地址` (IP 地址为 Nginx 虚拟化主机外部服务地址), 查看是否能够访问 Nginx 欢迎界面, 如图 4-6 所示, 以验证 Nginx 服务是否安装成功。



图 4-6 测试 Nginx 服务

4、负载均衡的实现

本次实验所使用的负载均衡算法是轮询算法, 在轮询算法中 `Weight` 越大, 通过 Nginx 代理分配访问的几率就越大。

本次实验设置 Web 服务器 1 的 `Weight` 值为 1, Web 服务器 2 值为 2, 简单地说, 例如通过 Nginx 服务器进行三次业务访问, 一次请求转发给 Web 服务器 1, 两次请求转发给 Web 服务器 2, 具体配置如下所示。

(1) 修改配置文件

编辑 Nginx 服务的默认配置文件 `default.conf`, 该文件存放的目录为 `/etc/nginx/conf.d`, 将配置文件中内容修改成如下所示, 编辑完成后保存退出。

```
# vi /etc/nginx/conf.d/default.conf
server {
##设置监听端口为 80
    listen    80;
    server_name localhost;

    .....
    location / {
        root    /usr/share/nginx/html;
        index  index.html index.htm;
        proxy_pass http://fzjh;
    }

    .....
}
```

编辑 Nginx 服务的配置文件 `nginx.conf`, 该文件存放的目录为 `/etc/nginx`, 在该配置文件中进行负载均衡的配置, 将配置文件中部分内容修改成如下所示, 编辑完成后保存退出。

```
# vi /etc/nginx/nginx.conf
.....
http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;
    log_format main '$remote_addr - $remote_user [$time_local]
"$request" '
        '$status $body_bytes_sent "$http_referer" '

```

```

"$http_user_agent" "$http_x_forwarded_for";
    access_log /var/log/nginx/access.log main;
sendfile        on;
#tcp_nopush     on;
keepalive_timeout 65;
#gzip on;
##设置负载均衡, 并设置轮询的依据
upstream fzhj {
    server 10.0.2.14:80 weight=1;
    server 10.0.2.15:80 weight=2;
}
.....

```

本次实验通过设置 Weight 的轮询算法实现业务的负载均衡, 请查找 Nginx 的其他负载均衡调度算法, 并将算法实现填写到表 4-3 中。

表 4-3 负载均衡算法



(2) 服务重启

完成配置文件编辑后, 需重启 Nginx 服务, 其操作命令如下。

```
# systemctl restart nginx
```

(3) 访问测试

在本地浏览器中输入 Nginx 服务地址, 检测能否访问到内部 Web 服务器发布的内容, 以验证代理服务是否配置成功, 访问结果如图 4-7 所示, 该页面为内部 Web 服务器 1 的网站访问界面。



图 4-7 测试负载均衡发布网页

5、性能测试

(1) 安装测试工具

在 Nginx 接收到请求之后，需把请求分发到后端 Web 服务集群中，需记录 Nginx 的分发日志来分析每台 Web 服务器处理的请求数目。

ab 是 apache 自带的压力测试工具，可以对 Apache、Nginx、Tomcat、IIS 等 Web 服务器进行模拟用户访问，进而进行压力测试，安装 ab 命令如下所示。

```
# yum install httpd-tools
```

(2) 修改配置文件

通过修改 Nginx 配置文件 nginx.conf，将 Nginx 的访问日志进行记录，为后续测试结果查看提供参考，将其配置文件部分内容修改为如下所示。

```
http {
    include      /etc/nginx/mime.types;
    default_type application/octet-stream;
    log_format   main
        '$remote_user [$time_local] $http_x_Forwarded_for $remote
        _addr $request '
        '$http_x_forwarded_for '
        '$upstream_addr '
        'ups_resp_time: $upstream_response_time '
        'request_time: $request_time';
    #设置 Nginx 的访问日志
    access_log   /var/log/nginx/access.log main;
    sendfile     on;
    #tcp_nopush  on;
    keepalive_timeout 65;
    #gzip on;
    #设置负载均衡配置
    upstream fzh {
        server 10.0.2.14:80 weight=1;
        server 10.0.2.15:80 weight=2;
    }
    include /etc/nginx/conf.d/*.conf;
}
```

(3) 服务重启

完成配置文件编辑后，需重启 Nginx 服务，其操作命令如下。

```
# systemctl restart nginx
```

(4) 设置测试命令

通过 ab 压力测试工具进行 Nginx 代理业务访问，其测试命令如下所示。

```
##-n 表示请求数，-c 表示并发数
# ab -n 100 -c 10 http://172.16.124.102/html/test
```

(5) 查看测试结果

执行 ab 压力测试后，查看 Nginx 的日志文件，其日志默认存放在 /var/log/nginx/ 目录下，其查看命令如下，部分日志结果如图 4-8 所示，将 Nginx 日志结果进行分析填写到表 4-4 中。

```
# cat /var/log/nginx/access.log
```

```
- [03/Mar/2018:01:51:32 -0500] - 172.16.124.102 GET /html/test.html HTTP/1.0 - 10.0.2.14:80
ups_resp_time: 0.003 request_time: 0.003
- [03/Mar/2018:01:51:32 -0500] - 172.16.124.102 GET /html/test.html HTTP/1.0 - 10.0.2.15:80
ups_resp_time: 0.017 request_time: 0.017
- [03/Mar/2018:01:51:32 -0500] - 172.16.124.102 GET /html/test.html HTTP/1.0 - 10.0.2.15:80
ups_resp_time: 0.016 request_time: 0.016
- [03/Mar/2018:01:51:32 -0500] - 172.16.124.102 GET /html/test.html HTTP/1.0 - 10.0.2.14:80
ups_resp_time: 0.003 request_time: 0.003
- [03/Mar/2018:01:51:32 -0500] - 172.16.124.102 GET /html/test.html HTTP/1.0 - 10.0.2.15:80
ups_resp_time: 0.020 request_time: 0.020
- [03/Mar/2018:01:51:32 -0500] - 172.16.124.102 GET /html/test.html HTTP/1.0 - 10.0.2.15:80
ups_resp_time: 0.018 request_time: 0.018
```

图 4-8 Nginx 日志查看

表 4-4 Nginx 日志分析结果

--

七、实验扩展

1、Nginx 服务器功能

- (1) Nginx 除做反向代理服务器外，可以做正向代理服务器吗？如果能，怎样实现？
- (2) Nginx 和 Tomcat、Apache 的区别是什么？请简要描述。

2、性能测试

- (1) 性能测试工具有哪几种？请简要描述。