

实验三：使用 Nginx 实现负载均衡

一、实验目的

- 1、了解负载均衡的基本原理；
- 2、掌握使用 Apache 发布网站；
- 3、掌握使用 Nginx 实现负载均衡。

二、实验学时

2 学时

三、实验类型

综合性

四、实验需求

1、硬件

每人配备计算机 1 台。

2、软件

Windows 操作系统，安装 Oracle VM VirtualBox 软件，安装 Mobaxerm 软件。

3、网络

本地主机与虚拟机能够访问互联网，不使用 DHCP 服务。

4、工具

无

五、实验任务

- 1、完成安装 Apache HTTP Server；
- 2、完成通过 Apache 发布内部网站；
- 3、完成使用 Nginx 以负载均衡方式发布内部网站；
- 4、完成负载均衡服务的测试。

六、实验环境

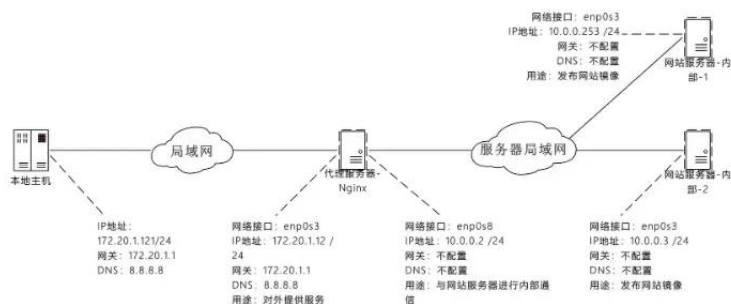
- 1、本实验需要 VM 3 台；
- 2、本实验 VM 配置信息如下表所示；

虚拟机配置	操作系统配置
虚拟机名称：VM-Lab-03-Task-01-172.20.1.12 内存：1GB	主机名：Lab-03-Task-01 IP 地址：172.20.1.12



CPU: 1 颗, 1 核心 虚拟磁盘: 20GB 网卡: 1 块, 桥接 1 块, 内部网络	子网掩码: 255.255.255.0 网关: 172.20.1.1 DNS: 8.8.8.8 IP 地址: 10.0.0.1 子网掩码: 255.255.255.0 DNS: 不配置
虚拟机名称: VM-Lab-03-Task-02-10.0.0.2 内存: 1GB CPU: 1 颗, 1 核心 虚拟磁盘: 20GB 网卡: 1 块, 内部网络	主机名: Lab-03-Task-02 IP 地址: 10.0.0.2 子网掩码: 255.255.255.0 网关: 不配置 DNS: 不配置
虚拟机名称: VM-Lab-03-Task-03-10.0.0.3 内存: 1GB CPU: 1 颗, 1 核心 虚拟磁盘: 20GB 网卡: 1 块, 内部网络	主机名: Lab-03-Task-03 IP 地址: 10.0.0.3 子网掩码: 255.255.255.0 网关: 不配置 DNS: 不配置

3、本实验拓扑图。



4、本实验操作演示视频。

本实验为视频集的第 3 集: <https://www.bilibili.com/video/BV1h14y1k7Gc?p=3>

七、实验内容及步骤

1、内部网站服务器-1 的配置, 并发布网站

在 VM-Lab-03-Task-02-10.0.0.2 主机上操作, 进行内部网站服务器的建设和网站发布。

- (1) 在 VirtualBox 中创建内部网站服务器的虚拟机。
- (2) 安装 Apache, 并完成防火墙的配置。

```
# 通过 dnf 在线工具安装 Apache HTTP Server
# 安装 Apache HTTP Server
[root@Lab-03-Task-02 ~]# dnf -y install httpd
```

```
# 启动 httpd 服务,并设置为自动启动
[root@Lab-03-Task-02 ~]# systemctl start httpd
[root@Lab-03-Task-02 ~]# systemctl enable httpd

# 查看 httpd 服务状态
[root@Lab-03-Task-02 ~]# systemctl status httpd

# 开启防火墙端口
[root@Lab-03-Task-02 ~]# firewall-cmd --permanent --add-port=80/tcp
[root@Lab-03-Task-02 ~]# firewall-cmd --permanent --add-port=81/tcp
[root@Lab-03-Task-02 ~]# firewall-cmd --permanent --add-port=8080/tcp
[root@Lab-03-Task-02 ~]# firewall-cmd --reload
[root@Lab-03-Task-02 ~]# firewall-cmd --list-all
```

(3) 创建内部网站。

创建的网站与发布方式结果如下：

表 1 内部网站服务器-1 发布的网站一览表

网站名称	访问路径	存放目录	发布方式
Site-1	http://10.0.0.2	/var/www/html	基于 IP 地址
Site-2	http://10.0.0.2:81	/var/www/html/sitefolder2	基于端口
Site-3	http://10.0.0.2:8080	/var/www/html/sitefolder3	基于端口
Site-4	http://10.0.0.2/site4	/var/www/html/sitefolder4	基于虚拟目录
Site-5	http://10.0.0.2/site5	/var/www/html/sitefolder5	基于虚拟目录

Site-1 的内容创建：使用/var/www/html 存储网站内容，撰写网站首页。

```
[root@Lab-03-Task-02 ~]# echo "<h1>Site-1:Publish using IP</h1>" > /var/www/html/index.html
```

Site-2 的内容创建：创建 sitefolder2 目录，撰写网站首页。

```
[root@Lab-03-Task-02 ~]# mkdir /var/www/html/sitefolder2
[root@Lab-03-Task-02 ~]# echo "<h1>Site-2:Release using port 81</h1>" > /var/www/html/sitefolder2/index.html
```

Site-3 的内容创建：创建 sitefolder3 目录，撰写网站首页。

```
[root@Lab-03-Task-02 ~]# mkdir /var/www/html/sitefolder3
[root@Lab-03-Task-02 ~]# echo "<h1>Site-3:Release using port 8080</h1>" > /var/www/html/sitefolder3/index.html
```

Site-4 的内容创建：创建 sitefolder4 目录，撰写网站首页。

```
[root@Lab-03-Task-02 ~]# mkdir /var/www/html/sitefolder4
[root@Lab-03-Task-02 ~]# echo "<h1>Site-4:Publish using virtual directory/site4/</h1>" > /var/www/html/sitefolder4/index.html
```

Site-5 的内容创建：创建 sitefolder5 目录，撰写网站首页。

```
[root@Lab-03-Task-02 ~]# mkdir /var/www/html/sitefolder5
```

```
[root@Lab-03-Task-02 ~]# echo "<h1>Site-5:Publish using virtual directory/site5</h1>" > /var/www/html/sitefolder5/index.html
```

(4) 基于 IP 地址发布网站

修改 Apache 配置文件的信息，以默认方式发布网站 Site-1。

```
# 配置文件的具体路径: /etc/httpd/conf/httpd.conf
# 查看 Apache 配置文件中实际用到的内容, 完成修改配置信息。
[root@Lab-03-Task-02 ~]# cat /etc/httpd/conf/httpd.conf | grep -vE "#|^$codeholder_2amp;quot;
# -----httpd.conf 文件-----
ServerRoot "/etc/httpd"
Listen 80
Include conf.modules.d/*.conf
User apache
Group apache
ServerAdmin root@localhost
<Directory />
    AllowOverride none
    Require all denied
</Directory>
DocumentRoot "/var/www/html"
<Directory "/var/www">
    AllowOverride None
    Require all granted
</Directory>
<Directory "/var/www/html">
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>
<Files ".ht*">
    Require all denied
</Files>
# 此处省略了部分提示信息
# -----
```

(5) 基于端口发布网站。

在/etc/httpd/conf.d 文件夹下创建 Site-2、Site-3 的配置文件。

```
# 使用端口方式发布 Site-2、Site-3。
# 创建 Site-2、Site-3 的配置文件, 并存入到/etc/httpd/conf.d 目录下。
[root@Lab-03-Task-02 ~]# vi /etc/httpd/conf.d/port-site-2.conf
# 将以下内容写入配置文件
# -----port-site-2.conf 文件-----
Listen 81
```

```

<VirtualHost *:81>
    DocumentRoot "/var/www/html/sitefolder2"
</VirtualHost>
# -----port-site-2.conf 文件-----

[root@Lab-03-Task-02 ~]# vi /etc/httpd/conf.d/port-site-3.conf
# 将以下内容写入配置文件
# -----port-site-3.conf 文件-----
Listen 8080
<VirtualHost *:8080>
    DocumentRoot "/var/www/html/sitefolder3"
</VirtualHost>
# -----port-site-3.conf 文件-----

```

(6) 基于虚拟目录发布网站。

在/etc/httpd/conf.d 目录下创建 Site-4、Site-5 的配置文件。

```

# 使用虚拟目录方式发布 Site-4、Site-5。
# 创建 Site-4、Site-5 的配置文件，并存入到/etc/httpd/conf.d 目录下。
[root@Lab-03-Task-02 ~]# vi /etc/httpd/conf.d/alias-site-4.conf
# 将以下内容写入配置文件
# -----alias-site-4.conf 文件-----
Alias /site4 "/var/www/html/sitefolder4"
<Directory "/var/www/html/sitefolder4">
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
# -----alias-site-4.conf 文件-----

[root@Lab-03-Task-02 ~]# vi /etc/httpd/conf.d/alias-site-5.conf
# 将以下内容写入配置文件
# -----alias-site-5.conf 文件-----
Alias /site5 "/var/www/html/sitefolder5"
<Directory "/var/www/html/sitefolder5">
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
# -----alias-site-5.conf 文件-----

```

(7) 网站测试。

```

# 重新载入 httpd 服务的配置文件
[root@Lab-03-Task-02 ~]# systemctl reload httpd

#通过内部主机进行网站的访问测试

```

2、内部网站服务器-2 的配置，并发布网站

- (1) 克隆内部网站服务器-1。
- (2) 修改网络配置，将 IP 地址修改为 10.0.0.3。
- (3) 发布内容与内部网站服务器-1 内容一致。

创建的网站与发布方式结果如下：

表 2 内部网站服务器-2 发布的网站一览表

网站名称	访问路径	存放目录	发布方式
Site-1	http://10.0.0.3	/var/www/html	基于 IP 地址
Site-2	http://10.0.0.3:81	/var/www/html/sitefolder2	基于端口
Site-3	http://10.0.0.3:8080	/var/www/html/sitefolder3	基于端口
Site-4	http://10.0.0.3/site4	/var/www/html/sitefolder4	基于虚拟目录
Site-5	http://10.0.0.3/site5	/var/www/html/sitefolder5	基于虚拟目录

3、配置 Nginx 实现负载均衡

在 VM-Lab-03-Task-01-172.20.1.12 主机上操作，配置 Nginx 并实现负载均衡服务。

本项目以域名方式发布内部网站服务器上的 5 个网站，通过 Nginx 发布的网站的信息如下：

表 3 对外发布的网站一览表

序号	发布地址	对应内部网站服务器-1 的网站	对应内部网站服务器-2 的网站
1	http://www.a.com	http://10.0.0.2	http://10.0.0.3
2	http://www.b.com	http://10.0.0.2:81	http://10.0.0.3:81
3	http://www.c.com	http://10.0.0.2:8080	http://10.0.0.3:8080
4	http://www.d.com	http://10.0.0.2/site4	http://10.0.0.3/site4
5	http://www.e.com	http://10.0.0.2/site5	http://10.0.0.3/site5

具体的配置方法如下：

- (1) 安装 Nginx。
- (2) 完成服务器防火墙的配置。
- (3) 实现轮询负载均衡。

```
# 安装了必要的依赖库和工具
[root@Lab-03-Task-01 ~]# dnf install -y gcc make zlib-devel pcre-devel openssl-devel

# 官网下载 Nginx 源码包
[root@Lab-03-Task-01 ~]# wget https://nginx.org/download/nginx-1.18.0.tar.gz

# 解压下载的源码包
[root@Lab-03-Task-01 ~]# tar -xzf nginx-1.18.0.tar.gz
# 使用 ./configure 命令配置 Nginx，并添加 http_stub_status_module 模块
[root@Lab-03-Task-01 ~]# cd nginx-1.18.0
[root@Lab-03-Task-01 nginx-1.18.0]# ./configure --with-http_stub_status_module
# 编译并安装 Nginx
[root@Lab-03-Task-01 nginx-1.18.0]# make
[root@Lab-03-Task-01 nginx-1.18.0]# sudo make install
```

```
# 添加环境变量, 启动 Nginx
[root@Lab-03-Task-01 nginx-1.18.0]# cd /usr/local/nginx
[root@Lab-03-Task-01 nginx]# vi /etc/profile
----- profile -----
# 添加内容
export PATH="/usr/local/nginx/sbin:$PATH"
-----

[root@Lab-03-Task-01 nginx]# source ~/.bashrc
[root@Lab-03-Task-01 nginx]# ./sbin/nginx
# 配置 Nginx 实现负载均衡服务
# 在 cd /usr/local/nginx/conf.d/目录下进行配置文件修改
# 轮询负载均衡
# 创建配置文件 port-site-1.conf, 通过域名实现内部网站 80 端口的负载。
[root@Lab-03-Task-01 nginx]# mkdir /usr/local/nginx/conf.d
[root@Lab-03-Task-01 nginx]# vi /usr/local/nginx/conf.d/port-site1.conf
----- port-site1.conf -----
upstream load1{
    server 10.0.0.2:80;
    server 10.0.0.3:80;
}
server{
    listen 80;
    server_name www.a.com;
    location / {
        proxy_pass http://load1;
        proxy_set_header Host $host:80;
    }
}

-----

# 创建配置文件 port-site-2.conf, 通过域名实现内部网站 81 端口的负载
[root@Lab-03-Task-01 nginx]# vi /usr/local/nginx/conf.d/port-site2.conf
----- port-site2.conf -----
upstream load2{
    server 10.0.0.2:81;
    server 10.0.0.3:81;
}
server{
    listen 80;
    server_name www.b.com;
    location / {
        proxy_pass http://load2;
        proxy_set_header Host $host:80;
    }
}

-----

# 创建配置文件 port-site-3.conf, 通过域名实现内部网站 8080 端口的负载
[root@Lab-03-Task-01 nginx]# vi /usr/local/nginx/conf.d/port-site3.conf
```

```
----- port-site3.conf -----
upstream load3{
    server 10.0.0.2:8080;
    server 10.0.0.3:8080;
}
server{
    listen 80;
    server_name www.c.com;
    location / {
        proxy_pass http://load3;
        proxy_set_header Host $host:80;
    }
}

-----
# 创建配置文件 port-site-4.conf, 通过域名实现内部网站虚拟目录 site4 的负载
[root@Lab-03-Task-01 nginx]# vi /usr/local/nginx/conf.d/port-site4.conf
----- port-site4.conf -----
upstream load4{
    server 10.0.0.2;
    server 10.0.0.3;
}
server{
    listen 80;
    server_name www.d.com;
    location / {
        proxy_pass http://load4/site4/;
        proxy_set_header Host $host:80;
    }
}

-----
# 创建配置文件 port-site-5.conf, 通过域名实现内部网站虚拟目录 site5 的负载
[root@Lab-03-Task-01 nginx]# vi /usr/local/nginx/conf.d/port-site5.conf
----- port-site5.conf -----
upstream load5{
    server 10.0.0.2;
    server 10.0.0.3;
}
server{
    listen 80;
    server_name www.e.com;
    location / {
        proxy_pass http://load5/site5/;
        proxy_set_header Host $host:80;
    }
}

-----
```



```
# 修改 Nginx 配置文件 nginx.conf
[root@Lab-03-Task-01 nginx]# vi /usr/local/nginx/conf/nginx.conf
----- nginx.conf -----
# 在文件中添加以下内容
include /usr/local/nginx/conf.d/*.conf;
-----

# 重新加载 Nginx
[root@Lab-03-Task-01 nginx]# ./sbin/nginx -s reload

# 添加防火墙策略
[root@Lab-03-Task-01 nginx]# firewall-cmd --zone=public --add-port=80/tcp --per
manent
[root@Lab-03-Task-01 nginx]# firewall-cmd --reload
[root@Lab-03-Task-01 nginx]# firewall-cmd --list-all
# 配置 SELinux
[root@Lab-03-Task-01 nginx]# setenforce 0
# 修改 SELinux 配置文件
[root@Lab-03-Task-01 nginx]# vi /etc/selinux/config
----- config -----
SELINUX=permissive
-----

# 重启 Nginx 服务器
[root@Lab-03-Task-01 nginx]# reboot
```

4、配置 Nginx 状态监控

在 VM-Lab-03-Task-01-172.20.1.12 主机上操作，配置 Nginx 并实现服务监控。

Nginx 提供了一个内置的状态信息监控页面可用于监控 Nginx 的整体访问情况，这个功能由 ngx_http_stub_status_module 模块进行实现。

(1) 配置 Nginx 实现监控。

```
# 修改 Nginx 配置文件, 添加以下内容
[root@Lab-03-Task-01 ~]# vi /usr/local/nginx/conf/nginx.conf
----- nginx.conf -----
server {
    listen *:80 default_server;
    server_name _;
    location /nginx_status {
        stub_status on;
        access_log off;
        allow 172.20.1.12;
        #deny all;
    }
}
-----

[root@Lab-03-Task-01 nginx]# ./sbin/nginx -s reload
```

在本地浏览器上通过访问 http://172.20.1.12/nginx_status 可查看 Nginx 的状态信息

5、访问测试

在本机上配置 Hosts 文件，通过访问域名 www.a.com、www.b.com、www.c.com、www.d.com、www.e.com 来进行以下测试：

- (1) 两台内部服务器均正常，进行访问测试。查看 Nginx 状态监控。
- (2) 手动关闭内部网站服务器-1，进行网站访问测试。查看 Nginx 状态监控。
- (3) 手动关闭内部网站服务器-2，进行网站访问测试。查看 Nginx 状态监控。测试结果应如下：

两台正常，可以访问，负载均衡。
单台故障，可以访问，说明容灾。
全部故障，不可访问。
