

实验七：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 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 Mobaxterm 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、实现负载均衡服务的搭建，并发布网站；
- 2、实现负载均衡服务的测试。

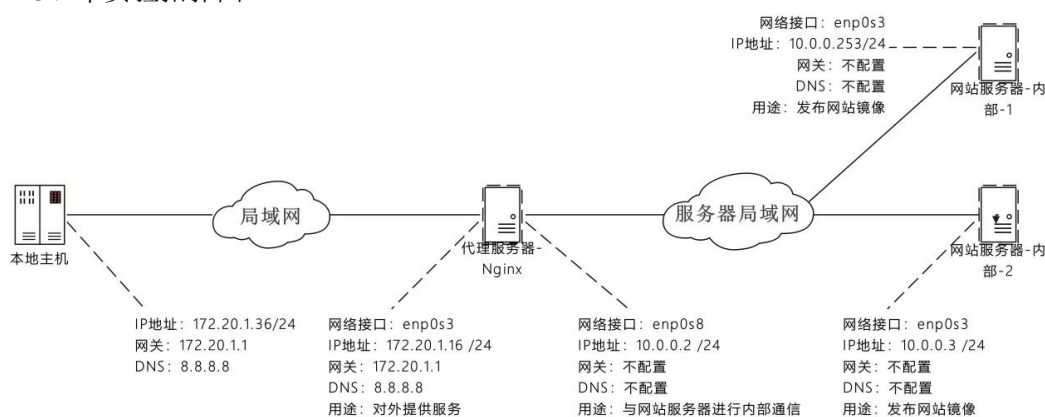
六、实验环境

- 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-07-Task-01-172.20.1.16 内存：1GB CPU：1 颗，1 核心 虚拟磁盘：20GB 网卡：1 块，桥接 1 块，内部网络	主机名：Lab-07-Task-01 IP 地址：172.20.1.16 子网掩码：255.255.255.0 网关：172.20.1.1 DNS：8.8.8.8 IP 地址：10.0.0.2

	子网掩码: 255.255.255.0 DNS: 8.8.8.8
虚拟机名称: VM-Lab-07-Task-02-10.0.0.253 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-07-Task-02 IP 地址: 10.0.0.253 子网掩码: 255.255.255.0 网关: 10.0.0.1 DNS: 8.8.8.8
虚拟机名称: VM-Lab-07-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-07-Task-03 IP 地址: 10.0.0.3 子网掩码: 255.255.255.0 网关: 10.0.0.1 DNS: 8.8.8.8

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 4 集: <https://www.bilibili.com/video/BV1Vh4y1T7EP?p=4>

七、实验内容及步骤

1、内网网站服务器-1 的配置, 并发布网站

在 VM-Lab-02-Task-02-10.0.0.253 主机上操作, 进行内部网站服务器的建设和网站发布。
具体步骤:

- (1) 创建内部网站服务器的 VM, 按照《实验 01: 安装 Linux 操作系统》的内容进行创建。
- (2) 完成服务器防火墙的配置。
- (3) 安装 Apache。
- (4) 参照《实验 04: 使用 Apache 发布网站》的内容, 发布 5 个网站。网站发布的结果为使用 IP 地址发布网站; 使用端口 81、8080 发布网站; 使用虚拟目录 site4、site45 发布网站。

2、内网网站服务器-2 的配置, 并发布网站

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

3、配置 Nginx 实现负载均衡

在 VM-Lab-07-Task-01-172.20.1.16 主机上操作，配置 Nginx 并实现负载均衡服务。

- (1) 安装 Nginx。
- (2) 通过修改配置文件，实现轮询负载均衡。

```
# 安装 Nginx
# 使用 dnf 命令安装 Nginx
[root@Lab-07-Task-01 ~]# dnf install -y nginx

# 启动 Nginx 服务
[root@Lab-07-Task-01 ~]# systemctl start nginx
# 查看 Nginx 服务运行状态
[root@Lab-07-Task-01 ~]# systemctl status nginx
# 设置 nginx 服务为开机自启动
[root@Lab-07-Task-01 ~]# systemctl enable nginx

# 配置 Nginx 实现负载均衡服务
# 在 vi /etc/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, 通过域名实现内网网站 80 端口的负载。
[root@Lab-07-Task-01 ~]# vi /etc/nginx/conf.d/port-site1.conf
----- port-site1.conf -----
upstream load1{
    server 10.0.0.253:80;
    server 10.0.0.3:80;
}
server{
    listen 80;
    server_name www.a.demo;
    location / {
        proxy_pass http://load1;
        proxy_set_header Host $host:80;
    }
}

# 创建配置文件 port-site-2.conf, 通过域名实现内网网站 81 端口的负载
[root@Lab-07-Task-01 ~]# vi /etc/nginx/conf.d/port-site2.conf
----- port-site2.conf -----
upstream load2{
    server 10.0.0.253:81;
    server 10.0.0.3:81;
}
server{
    listen 80;
    server_name www.b.demo;
    location / {
        proxy_pass http://load2;
```

```
        proxy_set_header Host $host:80;
    }

}

-----
# 创建配置文件 port-site-3.conf, 通过域名实现内网网站 8080 端口的负载
[root@Lab-07-Task-01 ~]# vi /etc/nginx/conf.d/port-site3.conf
----- port-site3.conf -----
upstream load3{
    server 10.0.0.253:8080;
    server 10.0.0.3:8080;
}
server{
    listen 80;
    server_name www.c.demo;
    location / {
        proxy_pass http://load3;
        proxy_set_header Host $host:80;
    }
}

-----
# 创建配置文件 port-site-4.conf, 通过域名实现内网网站虚拟目录 site4 的负载
[root@Lab-07-Task-01 ~]# vi /etc/nginx/conf.d/port-site4.conf
----- port-site4.conf -----
upstream load4{
    server 10.0.0.253;
    server 10.0.0.3;
}
server{
    listen 80;
    server_name www.d.demo;
    location / {
        proxy_pass http://load4/site4;
        proxy_set_header Host $host:80;
    }
}

-----
# 创建配置文件 port-site-5.conf, 通过域名实现内网网站虚拟目录 site5 的负载
[root@Lab-07-Task-01 ~]# vi /etc/nginx/conf.d/port-site5.conf
----- port-site5.conf -----
upstream load5{
    server 10.0.0.253;
    server 10.0.0.3;
}
server{
    listen 80;
    server_name www.e.demo;
```

```

        location / {
            proxy_pass http://load5/site5;
            proxy_set_header Host $host:80;
        }
    }
}
-----
# 重新加载 Nginx
[root@Lab-07-Task-01 nginx]# systemctl reload nginx

# 添加防火墙策略
[root@Lab-07-Task-01 ~]# firewall-cmd --zone=public --add-port=80/tcp --perman
ent
[root@Lab-07-Task-01 ~]# firewall-cmd --reload
[root@Lab-07-Task-01 ~]# firewall-cmd --list-all
# 配置 SELinux
[root@Lab-07-Task-01 ~]# setenforce 0
# 修改 SELinux 配置文件
[root@Lab-07-Task-01 ~]# vi /etc/selinux/config
----- config -----
SELINUX=permissive
-----
# 重启 Nginx 服务器
[root@Lab-07-Task-01 ~]# reboot

```

4、配置 Nginx 状态监控

在 VM-Lab-07-Task-01-172.20.1.16 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) 使用命令检测当前 Nginx 是否有 status 功能；
- (2) 配置 Nginx 实现监控。

```

# 检测是否有 status 功能
[root@Lab-07-Task-01 ~]# nginx -V 2>&1 | grep -o with-http_stub_status_module
# 输出 ngx_http_stub_status_module 说明有此模块
# 安装 status 模块
[root@Lab-07-Task-01 ~]# ./configure --with-http_stub_status_module

# 修改 Nginx 配置文件，添加以下内容
[root@Lab-07-Task-01 ~]# vi /etc/nginx/nginx.conf
----- nginx.conf -----
server {
    listen *:80 default_server;
    server_name _;
    location /nginx_status {
        stub_status on;
        access_log off;
        allow 172.20.1.16;
    }
}

```

```
#deny all;  
}  
}
```

在本地浏览器上通过访问 http://172.20.1.16/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.demo、www.b.demo、www.c.demo、www.d.demo、www.e.demo 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。
