

实验 03：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

安装 VMware WorkStation Pro 或 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，虚拟机网络不使用 DHCP 服务。

4、工具

无

五、实验任务

- 1、完成安装 Apache HTTP Server；

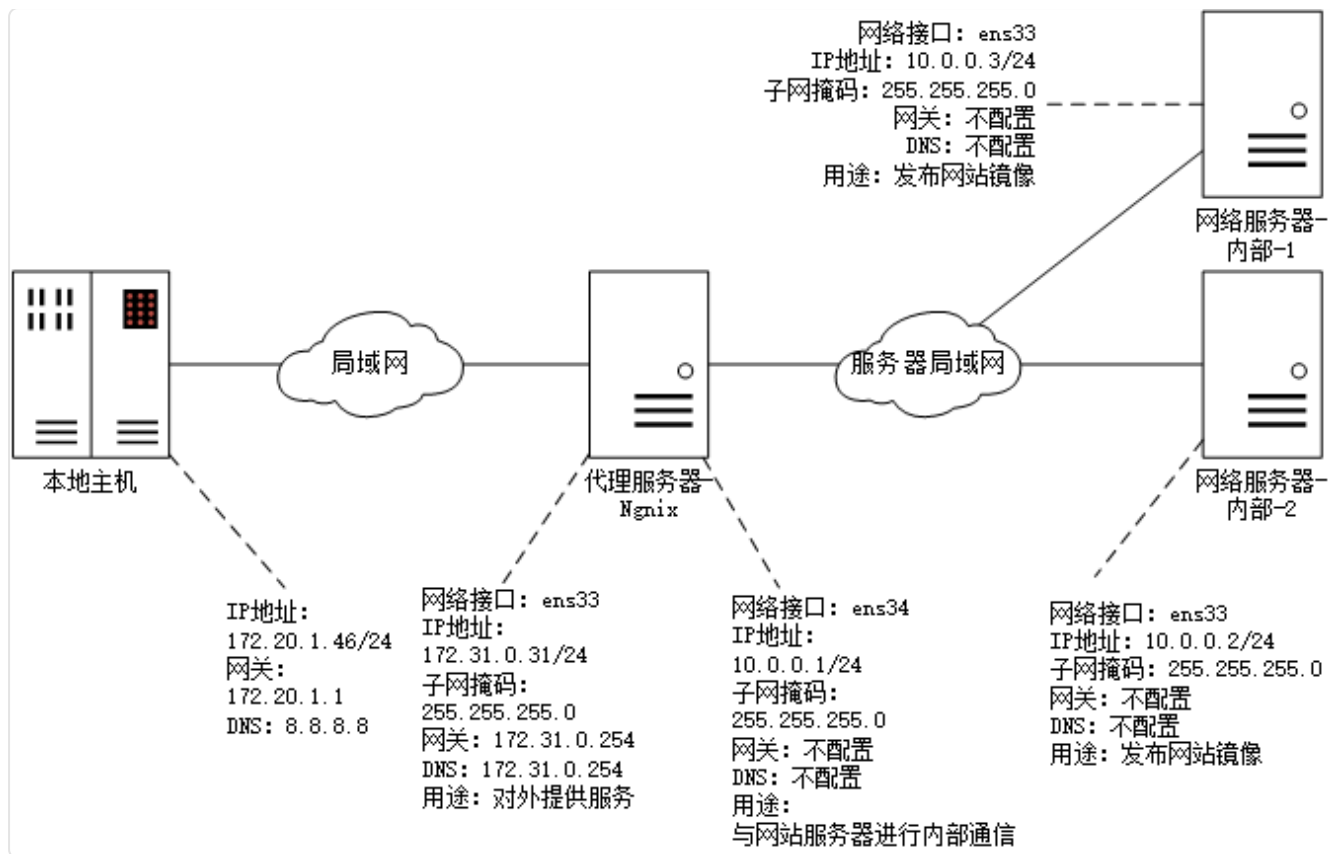
- 2、完成通过 Apache 发布内部网站；
- 3、完成使用 Nginx 以负载均衡方式发布内部网站；
- 4、完成负载均衡服务的测试。

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-01-172.31.0.31 内存：1GB CPU：1 颗，1 核心 虚拟磁盘：20GB 网卡： 1 块 1 块，内部网络	主机名：Lab-03-Task-01 IP 地址：172.31.0.31 子网掩码：255.255.255.0 网关：172.31.0.254 DNS：172.31.0.254 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。

本实验操作演示视频为视频集的第3集：<https://www.bilibili.com/video/BV1b1421t7aa?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 VMware WorkStation Pro 中创建内部网站服务器的虚拟机，并临时将虚拟机临时调整为可联网状态，方便安装应用服务，具体操作步骤请参考《实验 01：安装与基本配置》。
- (2) 虚拟机调整为可联网状态后，安装 Apache，并完成防火墙的配置。

Shell

```
1 # 通过yum在线工具安装Apache HTTP Server
2 # 安装Apache HTTP Server
3 [root@Lab-03-Task-02 ~]# yum -y install httpd
4
5 # 启动httpd服务,并设置为自动启动
6 [root@Lab-03-Task-02 ~]# systemctl start httpd
7 [root@Lab-03-Task-02 ~]# systemctl enable httpd
8
9 # 查看httpd服务状态
10 [root@Lab-03-Task-02 ~]# systemctl status httpd
11
12 # 开启防火墙端口
13 [root@Lab-03-Task-02 ~]# firewall-cmd --permanent --add-port=80/tcp
14 [root@Lab-03-Task-02 ~]# firewall-cmd --permanent --add-port=81/tcp
15 [root@Lab-03-Task-02 ~]# firewall-cmd --permanent --add-port=8080/tcp
16 [root@Lab-03-Task-02 ~]# firewall-cmd --reload
17 [root@Lab-03-Task-02 ~]# firewall-cmd --list-all
```



提醒：

1. 内网服务器创建完成后，临时将虚拟机调整为可联网状态，安装配置完服务后再调整至内网服务器。具体步骤如下：
 - a. 选择指定的虚拟机“10.0.0.2”，鼠标右键，依次选择【设置(S)...】->【硬件】->【网络适配器】，勾选“NAT 模式(N): 用于共享主机的 IP 地址”，保存设置。
 - b. 开启虚拟机，在系统中修改网络接口配置信息，修改内容如下：
IP 地址：172.31.0.32
子网掩码：255.255.255.0
网关：172.31.0.254
DNS：172.31.0.254
 - c. 重启网络接口，使得配置生效，此时虚拟机为可联网状态。
2. 将其他虚拟机临时调整为可联网状态，均可参考以上步骤，注意 IP 地址不冲突。

(3) 创建内部网站。

将虚拟机网络按照规划调整后，开展后续操作。创建的网站与发布方式结果如下：

表 3-1 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

Shell

```

1 # Site-1的内容创建：使用/var/www/html存储网站内容，撰写网站首页。
2 [root@Lab-03-Task-02 ~]# echo "<h1>Site-1:Publish using IP</h1>" > /var/www/html/index.html
3
4 # Site-2的内容创建：创建sitefolder2目录，撰写网站首页。
5 [root@Lab-03-Task-02 ~]# mkdir /var/www/html/sitefolder2
6 [root@Lab-03-Task-02 ~]# echo "<h1>Site-2:Release using port 81</h1>" > /var/www/html/sitefolder2/index.html
7
8 # Site-3的内容创建：创建sitefolder3目录，撰写网站首页。
9 [root@Lab-03-Task-02 ~]# mkdir /var/www/html/sitefolder3
10 [root@Lab-03-Task-02 ~]# echo "<h1>Site-3:Release using port 8080</h1>" > /var/www/html/sitefolder3/index.html
11
12 # Site-4的内容创建：创建sitefolder4目录，撰写网站首页。
13 [root@Lab-03-Task-02 ~]# mkdir /var/www/html/sitefolder4
14 [root@Lab-03-Task-02 ~]# echo "<h1>Site-4:Publish using virtual directory/site4</h1>" > /var/www/html/sitefolder4/index.html
15
16 # Site-5的内容创建：创建sitefolder5目录，撰写网站首页。
17 [root@Lab-03-Task-02 ~]# mkdir /var/www/html/sitefolder5
18 [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。

Shell

```
1 # 配置文件的具体路径：/etc/httpd/conf/httpd.conf
2 # 查看Apache配置文件中实际用到的内容，完成修改配置信息。
3 [root@Lab-03-Task-02 ~]# cat /etc/httpd/conf/httpd.conf | grep -vE "#|^$"
4 # -----httpd.conf文件-----
5 ServerRoot "/etc/httpd"
6 Listen 80
7 Include conf.modules.d/*.conf
8 User apache
9 Group apache
10 ServerAdmin root@localhost
11 <Directory />
12     AllowOverride none
13     Require all denied
14 </Directory>
15 DocumentRoot "/var/www/html"
16 <Directory "/var/www">
17     AllowOverride None
18     Require all granted
19 </Directory>
20 <Directory "/var/www/html">
21     Options Indexes FollowSymLinks
22     AllowOverride None
23     Require all granted
24 </Directory>
25 <IfModule dir_module>
26     DirectoryIndex index.html
27 </IfModule>
28 <Files ".ht*">
29     Require all denied
30 </Files>
31 # 此处省略了部分提示信息
32 # -----
```

(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。

Shell

```
1 # 使用端口方式发布Site-2、Site-3。
2 # 创建Site-2、Site-3的配置文件，并存入到/etc/httpd/conf.d目录下。
3 [root@Lab-03-Task-02 ~]# vi /etc/httpd/conf.d/port-site-2.conf
4 # 将以下内容写入配置文件
5 # -----port-site-2.conf文件-----
6 Listen 81
7 <VirtualHost *:81>
8     DocumentRoot "/var/www/html/sitefolder2"
9 </VirtualHost>
10 # -----port-site-2.conf文件-----
11
12 [root@Lab-03-Task-02 ~]# vi /etc/httpd/conf.d/port-site-3.conf
13 # 将以下内容写入配置文件
14 # -----port-site-3.conf文件-----
15 Listen 8080
16 <VirtualHost *:8080>
17     DocumentRoot "/var/www/html/sitefolder3"
18 </VirtualHost>
19 # -----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。

Shell

```
1 # 使用虚拟目录方式发布Site-4、Site-5。
2 # 创建Site-4、Site-5的配置文件，并存入到/etc/httpd/conf.d目录下。
3 [root@Lab-03-Task-02 ~]# vi /etc/httpd/conf.d/alias-site-4.conf
4 # 将以下内容写入配置文件
5 # -----alias-site-4.conf文件-----
6 Alias /site4 "/var/www/html/sitefolder4"
7 <Directory "/var/www/html/sitefolder4">
8     Options Indexes FollowSymLinks
9     AllowOverride None
10    Require all granted
11 </Directory>
12 # -----alias-site-4.conf文件-----
13
14 [root@Lab-03-Task-02 ~]# vi /etc/httpd/conf.d/alias-site-5.conf
15 # 将以下内容写入配置文件
16 # -----alias-site-5.conf文件-----
17 Alias /site5 "/var/www/html/sitefolder5"
18 <Directory "/var/www/html/sitefolder5">
19     Options Indexes FollowSymLinks
20     AllowOverride None
21     Require all granted
22 </Directory>
23 # -----alias-site-5.conf文件-----
```

(7) 网站测试。

Shell

```
1 # 重新载入httpd服务的配置文件
2 [root@Lab-03-Task-02 ~]# systemctl reload httpd
3 #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) 创建内部网站服务器-2。
- (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 3-2 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
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	基于虚拟主机
Site-5	http://10.0.0.3/site5	/var/www/html/sitefolder5	基于虚拟主机



提醒：
安装配置完服务后必须要将内部网站服务器 -1、网站服务器 -2 的网络接口配置和网络适配器修改至原样。

3、安装并配置 Nginx 实现负载均衡

在 VM-Lab-03-Task-01-172.31.0.31 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 3-3 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。

Shell

```
1 # 安装了必要的依赖库和工具
2 [root@Lab-03-Task-01 ~]# yum install -y gcc wget tar make zlib-devel pc
  re-devel openssl-devel
3 # 官网下载Nginx 源码包
4 [root@Lab-03-Task-01 ~]# wget https://nginx.org/download/nginx-1.18.0.t
  ar.gz
5
6 # 解压下载的源码包
7 [root@Lab-03-Task-01 ~]# tar -xzvf nginx-1.18.0.tar.gz
8 # 使用 ./configure 命令配置 Nginx, 并添加 http_stub_status_module 模块
9 [root@Lab-03-Task-01 ~]# cd nginx-1.18.0
10 [root@Lab-03-Task-01 nginx-1.18.0]# ./configure --with-http_stub_status
  _module
11 # 编译并安装 Nginx
12 [root@Lab-03-Task-01 nginx-1.18.0]# make
13 [root@Lab-03-Task-01 nginx-1.18.0]# make install
14 # 添加环境变量, 启动 Nginx
15 [root@Lab-03-Task-01 nginx-1.18.0]# cd /usr/local/nginx
16 [root@Lab-03-Task-01 nginx]# vi /etc/profile
17 ----- profile -----
18 # 添加内容
19 export PATH="/usr/local/nginx/sbin:$PATH"
20 -----
21 [root@Lab-03-Task-01 nginx]# source ~/.bashrc
22 [root@Lab-03-Task-01 nginx]# ./sbin/nginx
```

(2) 实现轮询模式的负载均衡。

```
1 # 配置Nginx实现负载均衡服务
2 # 在cd /usr/local/nginx/conf.d/目录下进行配置文件修改
3 # 轮询负载均衡
4 # 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。
5 [root@Lab-03-Task-01 nginx]# mkdir /usr/local/nginx/conf.d
6 [root@Lab-03-Task-01 nginx]# vi /usr/local/nginx/conf.d/port-site1.conf
7 ----- port-site1.conf -----
8 upstream load1{
9     server 10.0.0.2:80;
10    server 10.0.0.3:80;
11 }
12 server{
13     listen 80;
14     server_name www.a.com;
15     location / {
16         proxy_pass http://load1;
17         proxy_set_header Host $host:80;
18     }
19 }
20
21 -----
22 # 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
23 [root@Lab-03-Task-01 nginx]# vi /usr/local/nginx/conf.d/port-site2.conf
24 ----- port-site2.conf -----
25 upstream load2{
26     server 10.0.0.2:81;
27     server 10.0.0.3:81;
28 }
29 server{
30     listen 80;
31     server_name www.b.com;
32     location / {
33         proxy_pass http://load2;
34         proxy_set_header Host $host:80;
35     }
36
37 }
38 -----
39 # 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
40 [root@Lab-03-Task-01 nginx]# vi /usr/local/nginx/conf.d/port-site3.conf
41 ----- port-site3.conf -----
42 upstream load3{
```

```

43         server 10.0.0.2:8080;
44         server 10.0.0.3:8080;
45     }
46     server{
47         listen 80;
48         server_name www.c.com;
49         location / {
50             proxy_pass http://load3;
51             proxy_set_header Host $host:80;
52         }
53
54     }
55     -----
56 # 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
57 [root@Lab-03-Task-01 nginx]# vi /usr/local/nginx/conf.d/port-site4.conf
58 ----- port-site4.conf -----
59 upstream load4{
60     server 10.0.0.2;
61     server 10.0.0.3;
62 }
63 server{
64     listen 80;
65     server_name www.d.com;
66     location / {
67         proxy_pass http://load4/site4/;
68         proxy_set_header Host $host:80;
69     }
70
71 }
72 -----
73 # 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
74 [root@Lab-03-Task-01 nginx]# vi /usr/local/nginx/conf.d/port-site5.conf
75 ----- port-site5.conf -----
76 upstream load5{
77     server 10.0.0.2;
78     server 10.0.0.3;
79 }
80 server{
81     listen 80;
82     server_name www.e.com;
83     location / {
84         proxy_pass http://load5/site5/;
85         proxy_set_header Host $host:80;
86     }

```

```
87
88 }
89 -----
90 # 修改Nginx 配置文件nginx.conf
91 [root@Lab-03-Task-01 nginx]# vi /usr/local/nginx/conf/nginx.conf
92 ----- nginx.conf -----
93 # 在文件中添加以下内容
94 include /usr/local/nginx/conf.d/*.conf;
95 -----
96
97 # 重新加载Nginx
98 [root@Lab-03-Task-01 nginx]# ./sbin/nginx -s reload
```

(3) 完成服务器防火墙的配置。

Shell

```
1 # 添加防火墙策略
2 [root@Lab-03-Task-01 nginx]# firewall-cmd --zone=public --add-port=80/tcp --permanent
3 [root@Lab-03-Task-01 nginx]# firewall-cmd --reload
4 [root@Lab-03-Task-01 nginx]# firewall-cmd --list-all
5 # 配置SELinux
6 [root@Lab-03-Task-01 nginx]# setenforce 0
7 # 修改SELinux配置文件
8 [root@Lab-03-Task-01 nginx]# sed -i "s/SELINUX=enforcing/SELINUX=permissive/g" /etc/selinux/config
9 # 重启Nginx服务器
10 [root@Lab-03-Task-01 nginx]# reboot
```

4、配置 Nginx 状态监控

在 VM-Lab-03-Task-01-172.31.0.31 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。

配置 Nginx 实现监控。

Shell

```
1 # 修改Nginx 配置文件，添加以下内容
2 [root@Lab-03-Task-01 ~]# vi /usr/local/nginx/conf/nginx.conf
3 ----- nginx.conf -----
4 server {
5     listen *:80 default_server;
6     server_name _;
7     location /nginx_status {
8         stub_status on;
9         access_log off;
10        allow 172.31.0.31;
11        #deny all;
12    }
13 }
14 -----
15
16 [root@Lab-03-Task-01 nginx]# ./sbin/nginx -s reload
17
18 # 在本地浏览器上通过访问http://172.31.0.31/nginx_status可查看Nginx的状态信息
```

5、访问测试

在本地主机上配置 hosts 文件（文件访问路径 C:\Windows\System32\drivers\etc\hosts），依次通过浏览器进行域名访问测试。

Shell

```
1 #hosts文件内容
2 172.31.0.31    www.a.com
3 172.31.0.31    www.b.com
4 172.31.0.31    www.c.com
5 172.31.0.31    www.d.com
6 172.31.0.31    www.e.com
```

测试完成后，不同场景下的测试结果如表 4 所示：

表 4 测试结果表

场景	测试步骤	测试结果
场景一	两台内部服务器均正常，进行访问测试。查看 Nginx 状态监控。	两台正常，可以访 载均衡。
场景二	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。	单台故障，可以访 明容灾。
场景三	手动关闭内部网站服务器 -1 和网站服务器 -2，进行网站访问测试。查看 Nginx 状态监控。	全部故障，不可访

八、实验考核

实验考核分为【实验随堂查】和【实验线上考】两个部分。

实验随堂查：每个实验设置 5 个考核点。完成实验任务后，按照考核点要求，学生提交实验成果的截图或录屏视频。通过线上考核平台（如课堂派）进行作答。依据提交成果进行评分。

实验线上考：每个实验设置 5 道客观题。通过线上考核平台（如课堂派）进行作答。系统自动评分。

1、实验随堂查

本实验随堂查设置提交实验成果 -5 个截图 / 视频，具体如下：

题目 1[文件题]：提交通过内部主机访问内部服务器发布的网站截图；

题目 2[文件题]：提交使用域名能够访问到发布的网站截图；

题目 3[文件题]：提交在两台内部服务器均正常的情况下进行访问测试，查看 Nginx 状态监控，结果为两台正常，可以访问，负载均衡的网站截图；

题目 4[文件题]：提交在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，结果为单台故障，可以访问，说明容灾的网站截图；

题目 5[文件题]：提交在手动关闭内部网站服务器 -1 和网站服务器 -2，进行网站访问测试。查看 Nginx 状态监控，结果为全部故障，不可访问的网站截图；

2、实验线上考

本实验线上考共 5 题。

考核题目不对外发布：