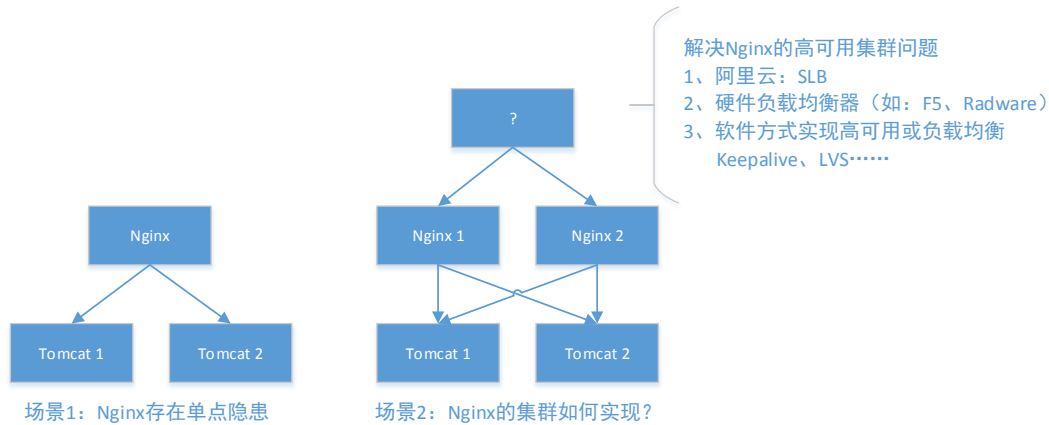


高可用架构篇

Keepalived + Nginx 实现高可用 Web 负载均衡

一、场景需求 (请看视频解说):

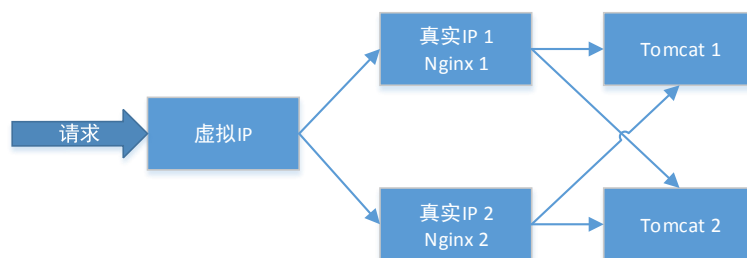


二、Keepalived 简要介绍

Keepalived 是一种高性能的服务器高可用或热备解决方案, Keepalived 可以用来防止服务器单点故障的发生, 通过配合 Nginx 可以实现 web 前端服务的高可用。

Keepalived 以 VRRP 协议为实现基础, 用 VRRP 协议来实现高可用性 (HA)。VRRP (Virtual Router Redundancy Protocol) 协议是用于实现路由器冗余的协议, VRRP 协议将两台或多台路由器设备虚拟成一个设备, 对外提供虚拟路由器 IP (一个或多个), 而在路由器组内部, 如果实际拥有这个对外 IP 的路由器如果工作正常的话就是 MASTER, 或者是通过算法选举产生, MASTER 实现针对虚拟路由器 IP 的各种网络功能, 如 ARP 请求, ICMP, 以及数据的转发等; 其他设备不拥有该虚拟 IP, 状态是 BACKUP, 除了接收 MASTER 的 VRRP 状态通告信息外, 不执行对外的网络功能。当主机失效时, BACKUP 将接管原先 MASTER 的网络功能。

VRRP 协议使用多播数据来传输 VRRP 数据, VRRP 数据使用特殊的虚拟源 MAC 地址发送数据而不是自身网卡的 MAC 地址, VRRP 运行时只有 MASTER 路由器定时发送 VRRP 通告信息, 表示 MASTER 工作正常以及虚拟路由器 IP (组), BACKUP 只接收 VRRP 数据, 不发送数据, 如果一定时间内没有接收到 MASTER 的通告信息, 各 BACKUP 将宣告自己成为 MASTER, 发送通告信息, 重新进行 MASTER 选举状态。



使用Keepalived来实现Nginx高可用



三、方案规划

VIP	IP	主机名	Nginx 端口	默认主从
192.168.1.50	192.168.1.51	edu-proxy-01	88	MASTER
	192.168.1.52	edu-proxy-02	88	BACKUP

CentOS 6.6 x64

keepalived-1.2.18.tar.gz

nginx-1.6.2.tar.gz

四、安装 Nginx

1、安装编译 Nginx 所需的依赖包

```
# yum install gcc gcc-c++ make automake autoconf libtool pcre pcre-devel zlib zlib-devel  
openssl openssl-devel
```

2、上传 Nginx(nginx-1.6.2.tar.gz)到 /usr/local/src 目录

3、编译安装 Nginx

```
# cd /usr/local/src/  
# tar -zxvf nginx-1.6.2.tar.gz  
# cd nginx-1.6.2  
# ./configure --prefix=/usr/local/nginx  
# make && make install
```

4、配置 Nginx

```
# vi /usr/local/nginx/conf/nginx.conf  
user root;  
worker_processes 1;  
#error_log logs/error.log;  
#error_log logs/error.log notice;  
#error_log logs/error.log info;  
#pid logs/nginx.pid;  
events {  
    worker_connections 1024;  
}  
http {  
    include 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 logs/access.log main;  
    sendfile on;  
    #tcp_nopush on;  
    #keepalive_timeout 0;  
    keepalive_timeout 65;
```



```
#gzip on;
server {
    listen      88;
    server_name localhost;
    #charset koi8-r;
    #access_log logs/host.access.log main;
    location / {
        root    html;
        index   index.html index.htm;
    }
    #error_page 404          /404.html;
    # redirect server error pages to the static page /50x.html
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root    html;
    }
}
}
```

修改 Nginx 欢迎首页内容（用于后面测试，用于区分两个节点的 Nginx）:

```
# vi /usr/local/nginx/html/index.html
```

192.168.1.51 中的标题加 1

```
<h1>Welcome to nginx! 1</h1>
```

192.168.1.52 中的标题加 2

```
<h1>Welcome to nginx! 2</h1>
```

5、系统防火墙打开对应的端口 88

```
# vi /etc/sysconfig/iptables
```

```
## Nginx
```

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 88 -j ACCEPT
```

```
# service iptables restart
```

6、测试 Nginx 是否安装成功

```
# /usr/local/nginx/sbin/nginx -t
```

```
nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
```

```
nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful
```

7、启动 Nginx

```
# /usr/local/nginx/sbin/nginx
```

重启 Nginx

```
# /usr/local/nginx/sbin/nginx -s reload
```

8、设置 Nginx 开机启动

```
# vi /etc/rc.local
```



龙果学院微信公众号: ron-coo

加入:

[/usr/local/nginx/sbin/nginx](#)

9、分别访问两个 Nginx



五、安装 Keepalived (<http://www.keepalived.org/download.html>)

1、上传或下载 keepalived (keepalived-1.2.18.tar.gz) 到 [/usr/local/src](#) 目录

2、解压安装

```
# cd /usr/local/src
# tar -zxvf keepalived-1.2.18.tar.gz
# cd keepalived-1.2.18
# ./configure --prefix=/usr/local/keepalived
# make && make install
```

3、将 keepalived 安装成 Linux 系统服务:

因为没有使用 keepalived 的默认路径安装 (默认是 [/usr/local](#)) , 安装完成之后, 需要做一些工作
复制默认配置文件到默认路径

```
# mkdir /etc/keepalived
# cp /usr/local/keepalived/etc/keepalived/keepalived.conf /etc/keepalived/
```

复制 keepalived 服务脚本到默认的地址

```
# cp /usr/local/keepalived/etc/rc.d/init.d/keepalived /etc/init.d/
# cp /usr/local/keepalived/etc/sysconfig/keepalived /etc/sysconfig/
# ln -s /usr/local/sbin/keepalived /usr/sbin/
# ln -s /usr/local/keepalived/sbin/keepalived /sbin/
```

设置 keepalived 服务开机启动

```
# chkconfig keepalived on
```



龙果学院微信公众号: ron-coo

4、修改 Keepalived 配置文件

(1) MASTER 节点配置文件 (192.168.1.51)

```
# vi /etc/keepalived/keepalived.conf
! Configuration File for keepalived
```

```
global_defs {
    ## keepalived 自带的邮件提醒需要开启 sendmail 服务。建议用独立的监控或第三方 SMTP
    router_id edu-proxy-01    ## 标识本节点的字条串，通常为 hostname
}

## keepalived 会定时执行脚本并对脚本执行的结果进行分析，动态调整 vrrp_instance 的优先级。如果
## 脚本执行结果为 0，并且 weight 配置的值大于 0，则优先级相应的增加。如果脚本执行结果非 0，并且 weight
## 配置的值小于 0，则优先级相应的减少。其他情况，维持原本配置的优先级，即配置文件中 priority 对应
## 的值。

vrrp_script chk_nginx {
    script "/etc/keepalived/nginx_check.sh"    ## 检测 nginx 状态的脚本路径
    interval 2    ## 检测时间间隔
    weight -20    ## 如果条件成立，权重-20
}

## 定义虚拟路由，VI_1 为虚拟路由的标示符，自己定义名称
vrrp_instance VI_1 {
    state MASTER    ## 主节点为 MASTER，对应的备份节点为 BACKUP
    interface eth1    ## 绑定虚拟 IP 的网络接口，与本机 IP 地址所在的网络接口相同，我的是 eth1
    virtual_router_id 51    ## 虚拟路由的 ID 号，两个节点设置必须一样，可选 IP 最后一段使用，相
    同的 VRID 为一个组，他将决定多播的 MAC 地址
    mcast_src_ip 192.168.1.51    ## 本机 IP 地址
    priority 100    ## 节点优先级，值范围 0-254，MASTER 要比 BACKUP 高
    nopreempt    ## 优先级高的设置 nopreempt 解决异常恢复后再次抢占的问题
    advert_int 1    ## 组播信息发送间隔，两个节点设置必须一样，默认 1s
    ## 设置验证信息，两个节点必须一致
    authentication {
        auth_type PASS
        auth_pass 1111    ## 真实生产，按需求对应该过来
    }
    ## 将 track_script 块加入 instance 配置块
    track_script {
        chk_nginx    ## 执行 Nginx 监控的服务
    }
    ## 虚拟 IP 池，两个节点设置必须一样
    virtual_ipaddress {
        192.168.1.50    ## 虚拟 ip，可以定义多个
    }
}
```



(2)BACKUP 节点配置文件 (192.168.1.52) :

```
# vi /etc/keepalived/keepalived.conf
! Configuration File for keepalived

global_defs {
    router_id edu-proxy-02
}

vrrp_script chk_nginx {
    script "/etc/keepalived/nginx_check.sh"
    interval 2
    weight -20
}

vrrp_instance VI_1 {
    state BACKUP
    interface eth1
    virtual_router_id 51
    mcast_src_ip 192.168.1.52
    priority 90
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    track_script {
        chk_nginx
    }
    virtual_ipaddress {
        192.168.1.50
    }
}
```

5、编写 Nginx 状态检测脚本 `/etc/keepalived/nginx_check.sh` (已在 `keepalived.conf` 中配置)

脚本要求: 如果 nginx 停止运行, 尝试启动, 如果无法启动则杀死本机的 keepalived 进程, keepalived 将虚拟 ip 绑定到 BACKUP 机器上。内容如下:

```
# vi /etc/keepalived/nginx_check.sh
#!/bin/bash
A=`ps -C nginx -no-header |wc -l`
if [ $A -eq 0 ];then
    /usr/local/nginx/sbin/nginx
    sleep 2
    if [ `ps -C nginx --no-header |wc -l` -eq 0 ];then
```



```
killall keepalived  
  
fi  
  
fi
```

保存后, 给脚本赋执行权限:

```
# chmod +x /etc/keepalived/nginx_check.sh
```

6、启动 Keepalived

```
# service keepalived start  
Starting keepalived: [ OK ]
```

7、Keepalived+Nginx 的高可用测试

(1) 关闭 192.168.1.51 中的 Nginx, Keepalived 会将它重新启动

```
# /usr/local/nginx/sbin/nginx -s stop
```

(2) 关闭 192.168.1.51 中的 Keepalived, VIP 会切换到 192.168.1.52 中

```
# service keepalived stop
```

```
[root@edu-proxy-01 network-scripts]# ip add  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000  
    link/ether 00:50:56:8d:54:46 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.1.51/24 brd 192.168.1.255 scope global eth1  
    inet6 fe80::250:56ff:fe8d:5446/64 scope link  
        valid_lft forever preferred_lft forever  
[root@edu-proxy-01 network-scripts]#
```

Keepalived 停止后, 该节点的网络接口中的 VIP 将消失

查看此时 VIP 对应的 MAC, Windows 下使用 CMD 命令查看:



```
C:\Windows\system32\cmd.exe  
Microsoft Windows [版本 6.1.7601]  
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。  
  
C:\Users\Administrator>arp -a 192.168.1.50  
  
接口: 192.168.1.180 --- 0xd  
Internet 地址      物理地址      类型  
192.168.1.50      00-50-56-8d-ce-4e 动态  
  
C:\Users\Administrator>arp -a 192.168.1.52  
  
接口: 192.168.1.180 --- 0xd  
Internet 地址      物理地址      类型  
192.168.1.52      00-50-56-8d-ce-4e 动态  
  
C:\Users\Administrator>
```

说明此时 VIP 已经漂移到物理主机 192.168.1.52 上了

再通过 VIP 来访问 Nginx 集群, 访问到的也是 192.168.1.52





(3) 重新启动 192.168.1.51 中的 Keepalived, VIP 又会切回到 192.168.1.51 中来

`service keepalived start`

查看虚拟 IP 状态

```
管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>arp -a 192.168.1.50

接口: 192.168.1.180 --- 0xd
Internet 地址      物理地址      类型
192.168.1.50      00-50-56-8d-54-46 动态

C:\Users\Administrator>arp -a 192.168.1.51

接口: 192.168.1.180 --- 0xd
Internet 地址      物理地址      类型
192.168.1.51      00-50-56-8d-54-46 动态

C:\Users\Administrator>
```

`ip add`

```
[root@edu-proxy-01 network-scripts]# ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
    link/ether 00:50:56:8d:54:46 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.51/24 brd 192.168.1.255 scope global eth1
    inet 192.168.1.50/32 scope global eth1
    inet6 fe80::250:56ff:fe8d:5446/64 scope link
        valid_lft forever preferred_lft forever
[root@edu-proxy-01 network-scripts]#
```

Keepalived 启动后, 网络接口上又会创建出 VIP 192.168.1.50





Keepalived 服务管理命令:

停止: `service keepalived stop`

启动: `service keepalived start`

重启: `service keepalived restart`

查看状态: `service keepalived status`

其他参考资料:

keepalived 之 vrrp_script 总结: <http://my.oschina.net/hnscswc/blog/158746>

keepalived 双机热备实现故障时发送邮件通知: <http://www.2cto.com/os/201407/317795.html>

基于 keepalived 实现 VIP 转移, lvs, nginx 的高可用: <http://www.tuicool.com/articles/eu26Vz>

最新课程列表、课程下载链接、课程知识互动, 请关注“龙果”微信公众号!



内容持续完善中!



龙果学院微信公众号: ron-coo