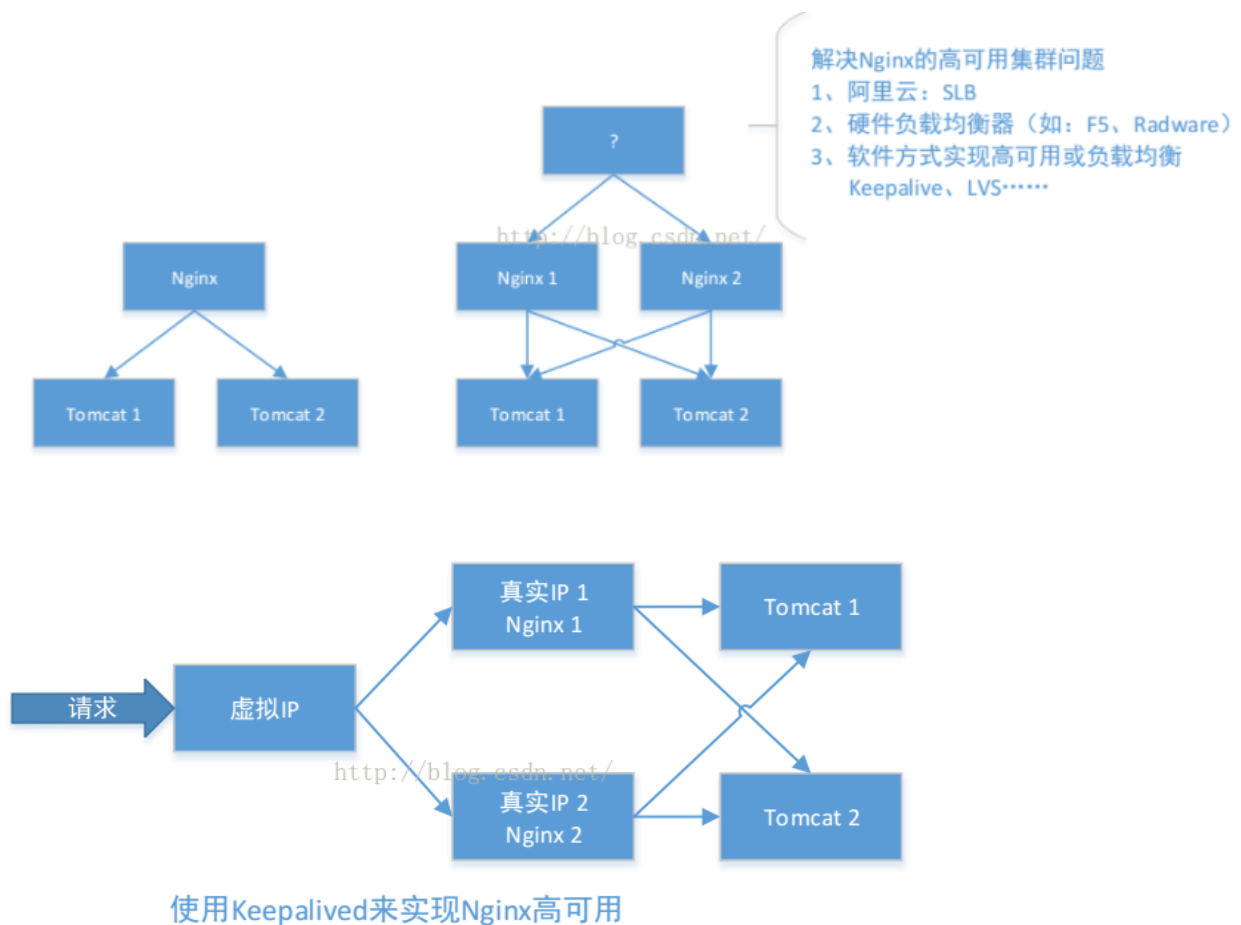


Keepalived

Keepalived是一个基于VRRP协议来实现的服务高可用方案。VRRP协议（虚拟路由冗余协议——Virtual Router Redundancy Protocol，简称VRRP），是由IETF提出的解决局域网中配置静态网关出现单点失效现象的路由协议，1998年已推出正式的RFC2338协议标准。VRRP广泛应用在边缘网络中，它的设计目标是支持特定情况下IP数据流量失败转移不会引起混乱，允许主机使用单路由器，以及即使在实际第一跳路由器使用失败的情形下仍能够维护路由器间的连通性。

大白话来说就是，VRRP协议允许一台机器可以拥有一个或者多个虚拟IP。在高可用的一组机器中，有一个master，多个slave，对外提供一个虚IP，通过虚IP访问master，slave负责监控master，如果master宕机，则选举其中一个slave接管master，虚IP绑定到新的master上（俗称IP漂移），从而实现了高可用。



关闭防火墙

```
setenforce 0  
systemctl stop firewalld
```

安装Nginx (230与231)

```
sudo yum install -y yum-utils  
  
sudo cat >>/etc/yum.repos.d/nginx.repo<<-'EOF'  
[nginx-stable]  
name=nginx stable repo  
baseurl=http://nginx.org/packages/centos/$releasever/$basearch/
```

```

gpgcheck=1
enabled=1
gpgkey=https://nginx.org/keys/nginx_signing.key
module_hotfixes=true

[nginx-mainline]
name=nginx mainline repo
baseurl=http://nginx.org/packages/mainline/centos/$releasever/$basearch/
gpgcheck=1
enabled=0
gpgkey=https://nginx.org/keys/nginx_signing.key
module_hotfixes=true
EOF

#安装nginx
sudo yum install -y nginx
#启动nginx
systemctl start nginx
#系统启动nginx自动启动
systemctl enable nginx

```

KeepAlived Master(230)

安装keepalived

```

yum install -y keepalived

```

check_nginx脚本

Keepalived会定时执行 “ps -C nginx --no-heading|wc -l” 命令，
如果返回0，代表Nginx挂了，然后尝试重启，如果重启失败，停止keepalived触发故障转移
如果返回大于0，代表Nginx正常运行，啥都不干~

```

sudo cat >/etc/keepalived/check_nginx.sh<<- 'EOF'
#!/bin/bash
counter=$(ps -C nginx --no-heading|wc -l)
if [ "${counter}" = "0" ]; then
systemctl start nginx
sleep 2
counter=$(ps -C nginx --no-heading|wc -l)
if [ "${counter}" = "0" ]; then
systemctl stop keepalived
fi
fi
EOF

chmod 755 /etc/keepalived/check_nginx.sh

```

keepalived.conf master配置文件

```

sudo cat >/etc/keepalived/keepalived.conf<<- 'EOF'
! Configuration File for keepalived
# 全局配置, 路由ID, 固定不变
global_defs {
router_id LVS_DEVEL
}
# 定义Nginx状态脚本
vrrp_script chk_nginx {
script "/etc/keepalived/check_nginx.sh"
# 间隔时间, 单位为秒, 默认1秒
interval 2
# 权重, 当脚本成功或失败对当前节点的优先级是增加还是减少
weight -5
}

#VRRP实例
vrrp_instance VI_1 {
# 主节点
state MASTER
# 绑定的网卡, 使用ifconfig命令查看获取
interface ens33
# 虚拟路由id, 保证相同
virtual_router_id 51
# 优先级, 抢占模式下优先级高的称为主
priority 101
# 指定发送VRRP通告的间隔。单位是秒。
advert_int 2
# 安全认证用的密码, 自定义即可
authentication {
auth_type PASS
auth_pass 1111
}
# 对外暴露的VIP地址
virtual_ipaddress {
192.168.31.240
}
# 指定Nginx执行状态脚本
track_script {
chk_nginx
}
}
EOF

```

启动KeepAlived

```

systemctl start keepalived
tail -f /var/log/messages

```

KeepAlived Backup(231)

其他步骤完全相同, 只有keepalived.conf有细微差别, 过程略

```

yum install -y keepalived

```

```

sudo cat >/etc/keepalived/check_nginx.sh<<- 'EOF'

```

```
#!/bin/bash
counter=$(ps -C nginx --no-heading|wc -l)
if [ "${counter}" = "0" ]; then
systemctl start nginx
sleep 2
counter=$(ps -C nginx --no-heading|wc -l)
if [ "${counter}" = "0" ]; then
systemctl stop keepalived
fi
fi
EOF
chmod 755 /etc/keepalived/check_nginx.sh
```

keepalived.conf slave配置文件

```
sudo cat >/etc/keepalived/keepalived.conf<<-'EOF'
! Configuration File for keepalived
global_defs {
router_id LVS_DEVEL
}

vrrp_script chk_nginx {
script "/etc/keepalived/check_nginx.sh"
interval 2
weight -5
}

vrrp_instance VI_1 {
# 初始角色Backup
state BACKUP
interface ens33
virtual_router_id 51
# 优先级比master低
priority 100
advert_int 2
authentication {
auth_type PASS
auth_pass 1111
}
virtual_ipaddress {
192.168.31.240
}
track_script {
chk_nginx
}
}
EOF
```

```
systemctl start keepalived
tail -f /var/log/messages
```

主从切换日志，触发故障转移

```
Apr 6 17:17:27 localhost systemd: Started nginx - high performance web server.
Apr 6 17:20:01 localhost systemd: Started Session 48 of user root.
Apr 6 17:25:39 localhost Keepalived_vrrp[62708]: VRRP_Instance(VI_1) Transition to MASTER STATE
Apr 6 17:25:41 localhost Keepalived_vrrp[62708]: VRRP_Instance(VI_1) Entering MASTER STATE
Apr 6 17:25:41 localhost Keepalived_vrrp[62708]: VRRP_Instance(VI_1) setting protocol VIPs.
Apr 6 17:25:41 localhost Keepalived_vrrp[62708]: Sending gratuitous ARP on ens33 for 192.168.31.240
Apr 6 17:25:41 localhost Keepalived_vrrp[62708]: VRRP_Instance(VI_1) Sending/queueing gratuitous ARPs on ens33 for 192.168.31.240
Apr 6 17:25:41 localhost Keepalived_vrrp[62708]: Sending gratuitous ARP on ens33 for 192.168.31.240
Apr 6 17:25:41 localhost Keepalived_vrrp[62708]: Sending gratuitous ARP on ens33 for 192.168.31.240
Apr 6 17:25:41 localhost Keepalived_vrrp[62708]: Sending gratuitous ARP on ens33 for 192.168.31.240
Apr 6 17:25:41 localhost Keepalived_vrrp[62708]: Sending gratuitous ARP on ens33 for 192.168.31.240
Apr 6 17:25:46 localhost Keepalived_vrrp[62708]: Sending gratuitous ARP on ens33 for 192.168.31.240
Apr 6 17:25:46 localhost Keepalived_vrrp[62708]: VRRP_Instance(VI_1) Sending/queueing gratuitous ARPs on ens33 for 192.168.31.240
Apr 6 17:25:46 localhost Keepalived_vrrp[62708]: Sending gratuitous ARP on ens33 for 192.168.31.240
Apr 6 17:25:46 localhost Keepalived_vrrp[62708]: Sending gratuitous ARP on ens33 for 192.168.31.240
Apr 6 17:25:46 localhost Keepalived_vrrp[62708]: Sending gratuitous ARP on ens33 for 192.168.31.240
Apr 6 17:25:46 localhost Keepalived_vrrp[62708]: Sending gratuitous ARP on ens33 for 192.168.31.240
Apr 6 17:25:46 localhost Keepalived_vrrp[62708]: Sending gratuitous ARP on ens33 for 192.168.31.240
```

默认抢占模式下，主节点恢复，低优先级节点自动降级

```
Apr 6 17:27:46 localhost Keepalived_vrrp[62708]: VRRP_Instance(VI_1) Received advert with higher priority 101, ours 100
Apr 6 17:27:46 localhost Keepalived_vrrp[62708]: VRRP_Instance(VI_1) Entering BACKUP STATE
Apr 6 17:27:46 localhost Keepalived_vrrp[62708]: VRRP_Instance(VI_1) removing protocol VIPs.
```

KeepAlived工作原理

第一步：选举出Master

- VRRP备份组中的设备根据优先级选举出Master。Master设备通过发送免费ARP报文，将虚拟MAC地址通知给与它连接的设备或者主机，从而承担报文转发任务。
 - 选举规则：比较优先级的大小，优先级高者当选为Master设备。当两台设备优先级相同时，如果已经存在Master，则其保持Master身份，无需继续

选举；如果不存在Master，则继续比较接口IP地址大小，接口IP地址较大的设备当选为Master设备。

- 第二步：Master设备状态的通告（VRRP备份组状态维持）
 - Master设备周期性地发送VRRP通告报文，在VRRP备份组中公布其配置信息（优先级等）和工作状况。Backup设备通过接收到的VRRP报文来判断Master设备是否工作正常。
 - 当Master设备主动放弃Master地位（如Master设备退出备份组）时，会发送优先级为0的通告报文，用来使Backup设备快速切换成Master设备，而不用等到Master_Down_Interval（默认为3s）定时器超时。这个切换的时间称为Skew_Time（几乎可以堪称0s），计算方式为： $(256 - \text{Backup设备的优先级}) / 256$ ，单位为秒。
 - 当Master设备发生网络故障而不能发送通告报文的时候，Backup设备并不能立即知道其工作状况。等到Master_Down_Interval定时器超时后，才会认为Master设备无法正常工作，从而将状态切换为Master。其中，Master_Down_Interval定时器取值为： $3 \times \text{Advertisement_Interval} + \text{Skew_Time}$ ，单位为秒。其中Advertisement_Interval默认情况下为1S

VRRP主备切换过程

如果Master发生故障，则主备切换的过程

- 当组内的备份设备一段时间（Master_Down_Interval定时器取值为： $3 \times \text{Advertisement_Interval} + \text{Skew_Time}$ ，单位为秒）内没有接收到来自Master设备的报文，则将自己转为Master设备。
- 一个VRRP组里有多台备份设备时，短时间内可能产生多个Master设备，此时，设备将会对收到的VRRP报文中的优先级与本地优先级做比较，从而选取优先级高的设备成为Master。设备的状态变为Master之后，会立刻发送免费ARP来刷新交换机上的MAC表项，从而把用户的流量引到此设备上来，整个过程对用户完全透明。