**Keepalived高可用**

## 1.1 keepalived是什么？

　　Keepalived软件起初是专为LVS负载均衡软件设计的，用来管理并监控LVS集群系统中各个服务节点的状态，后来又加入了可以实现高可用的VRRP功能。因此，Keepalived除了能够管理LVS软件外，还可以作为其他服务（例如：Nginx、Haproxy、MySQL等）的高可用解决方案软件。

　　Keepalived软件主要是通过VRRP协议实现高可用功能的。VRRP是Virtual Router RedundancyProtocol(虚拟路由器冗余协议）的缩写，VRRP出现的目的就是为了解决静态路由单点故障问题的，它能够保证当个别节点宕机时，整个网络可以不间断地运行。

　　所以，Keepalived 一方面具有配置管理LVS的功能，同时还具有对LVS下面节点进行健康检查的功能，另一方面也可实现系统网络服务的高可用功能。

## 1.2 keepalived服务的三个重要功能

　　管理LVS负载均衡软件

　　实现LVS集群节点的健康检查中

　　作为系统网络服务的高可用性（failover）

## 1.3 Keepalived高可用故障切换转移原理

　　Keepalived高可用服务对之间的故障切换转移，是通过 VRRP (Virtual Router Redundancy Protocol ,虚拟路由器冗余协议）来实现的。

　　在 Keepalived服务正常工作时，主 Master节点会不断地向备节点发送（多播的方式）心跳消息，用以告诉备Backup节点自己还活看，当主 Master节点发生故障时，就无法发送心跳消息，备节点也就因此无法继续检测到来自主 Master节点的心跳了，于是调用自身的接管程序，接管主Master节点的 IP资源及服务。而当主 Master节点恢复时，备Backup节点又会释放主节点故障时自身接管的IP资源及服务，恢复到原来的备用角色。

那么，什么是VRRP呢？

**VRRP** ,全 称 Virtual Router Redundancy Protocol ,中文名为虚拟路由冗余协议 ，VRRP的出现就是为了解决静态踣甶的单点故障问题，VRRP是通过一种竞选机制来将路由的任务交给某台VRRP路由器的。

## Keepalived的工作原理：

Keepalived高可用对之间是通过VRRP通信的，因此，我们从 VRRP开始了解起：

　　 1) VRRP,全称 Virtual Router Redundancy Protocol,中文名为虚拟路由冗余协议，VRRP的出现是为了解决静态路由的单点故障。

　　 2) VRRP是通过一种竟选协议机制来将路由任务交给某台 VRRP路由器的。

　　 3) VRRP用 IP多播的方式（默认多播地址（224.0\_0.18))实现高可用对之间通信。

　　 4) 工作时主节点发包，备节点接包，当备节点接收不到主节点发的数据包的时候，就启动接管程序接管主节点的开源。备节点可以有多个，通过优先级竞选，但一般 Keepalived系统运维工作中都是一对。

　　 5) VRRP使用了加密协议加密数据，但Keepalived官方目前还是推荐用明文的方式配置认证类型和密码。

介绍完 VRRP,接下来我再介绍一下 Keepalived服务的工作原理：

　　Keepalived高可用对之间是通过 VRRP进行通信的， VRRP是遑过竞选机制来确定主备的，主的优先级高于备，因此，工作时主会优先获得所有的资源，备节点处于等待状态，当主挂了的时候，备节点就会接管主节点的资源，然后顶替主节点对外提供服务。

　　在 Keepalived服务对之间，只有作为主的服务器会一直发送 VRRP广播包,告诉备它还活着，此时备不会枪占主，当主不可用时，即备监听不到主发送的广播包时，就会启动相关服务接管资源，保证业务的连续性.接管速度最快可以小于1秒。

## 2.2 脑裂

    　　在高可用（HA）系统中，当联系2个节点的“心跳线”断开时，本来为一整体、动作协调的HA系统，就分裂成为2个独立的个体。由于相互失去了联系，都以为是对方出了故障。两个节点上的HA软件像“裂脑人”一样，争抢“共享资源”、争起“应用服务”，就会发生严重后果——或者共享资源被瓜分、2边“服务”都起不来了；或者2边“服务”都起来了，但同时读写“共享存储”，导致数据损坏（常见如数据库轮询着的联机日志出错）。

对付HA系统“裂脑”的对策，目前达成共识的的大概有以下几条：

1）添加冗余的心跳线，例如：双线条线（心跳线也HA），尽量减少“裂脑”发生几率；

2）启用磁盘锁。正在服务一方锁住共享磁盘，“裂脑”发生时，让对方完全“抢不走”共享磁盘资源。但使用锁磁盘也会有一个不小的问题，如果占用共享盘的一方不主动“解锁”，另一方就永远得不到共享磁盘。现实中假如服务节点突然死机或崩溃，就不可能执行解锁命令。后备节点也就接管不了共享资源和应用服务。于是有人在HA中设计了“智能”锁。即：正在服务的一方只在发现心跳线全部断开（察觉不到对端）时才启用磁盘锁。平时就不上锁了。

3）设置仲裁机制。例如设置参考IP（如网关IP），当心跳线完全断开时，2个节点都各自ping一下参考IP，不通则表明断点就出在本端。不仅“心跳”、还兼对外“服务”的本端网络链路断了，即使启动（或继续）应用服务也没有用了，那就主动放弃竞争，让能够ping通参考IP的一端去起服务。更保险一些，ping不通参考IP的一方干脆就自我重启，以彻底释放有可能还占用着的那些共享资源。

### 2.2.1 脑裂产生的原因

一般来说，裂脑的发生，有以下几种原因：

高可用服务器对之间心跳线链路发生故障，导致无法正常通信。

　　　因心跳线坏了（包括断了，老化）。

　　　因网卡及相关驱动坏了，ip配置及冲突问题（网卡直连）。

　　　因心跳线间连接的设备故障（网卡及交换机）。

　　　因仲裁的机器出问题（采用仲裁的方案）。

高可用服务器上开启了 iptables防火墙阻挡了心跳消息传输。

高可用服务器上心跳网卡地址等信息配置不正确，导致发送心跳失败。

其他服务配置不当等原因，如心跳方式不同，心跳广插冲突、软件Bug等。

**提示：** Keepalived配置里同一 VRRP实例如果 virtual\_router\_id两端参数配置不一致也会导致裂脑问题发生。

### 2.2.2 常见的解决方案

在实际生产环境中，我们可以从以下几个方面来防止裂脑问题的发生：

　　同时使用串行电缆和以太网电缆连接，同时用两条心跳线路，这样一条线路坏了，另一个还是好的，依然能传送心跳消息。

　　当检测到裂脑时强行关闭一个心跳节点（这个功能需特殊设备支持，如Stonith、feyce）。相当于备节点接收不到心跳消患，通过单独的线路发送关机命令关闭主节点的电源。

　　做好对裂脑的监控报警（如邮件及手机短信等或值班）.在问题发生时人为第一时间介入仲裁，降低损失。例如，百度的监控报警短倍就有上行和下行的区别。报警消息发送到管理员手机上，管理员可以通过手机回复对应数字或简单的字符串操作返回给服务器.让服务器根据指令自动处理相应故障，这样解决故障的时间更短.

　　当然，在实施高可用方案时，要根据业务实际需求确定是否能容忍这样的损失。对于一般的网站常规业务.这个损失是可容忍的。

## 2.3 如何进行脑裂情况监控

### 2.3.1 在什么服务器上进行监控？

在备服务器上进行监控，可以使用zabbix监控，参考<http://www.cnblogs.com/clsn/p/7885990.html>

### 2.3.2 监控什么信息？

备上面出现vip情况：

1）脑裂情况出现

2）正常主备切换也会出现

**5、添加VIP**

[root@moban ~]# ip addr add 10.10.10.10/24 dev ens33 label ens33:0

[root@moban ~]# ip addr del 10.10.10.10/24 dev ens33 label ens33:0

添加虚拟IP的三种方法

1、ifconfig ens33: 10.10.10.250 netmask 255.255.255.0 up 别名

2、ip addr add 10.10.10.10/24 dev ens33 辅助IP

3、keepalived

**Keepalived-一主一备**

yum -y install keepalived

配置文件

/etc/keepalived/keepalived.conf

systemctl enable keepalived

systemctl list-unit-files|grep enabled

systemctl start keepalived

|  |  |
| --- | --- |
| ! Configuration File for keepalived  global\_defs {  notification\_email {  122123498@qq.com  }  notification\_email\_from 122123498.com  smtp\_server 192.168.200.1  smtp\_connect\_timeout 30  router\_id LVS\_DEVEL  #解决多组keepalived在同一个局域网的冲突问题  vrrp\_mcast\_group4 224.0.0.19  vrrp\_skip\_check\_adv\_addr  #vrrp\_strict  vrrp\_garp\_interval 0  vrrp\_gna\_interval 0  }  #检测nginx是否正常运行  vrrp\_script chk\_nginx\_proxy {  script "/server/scripts/chk\_nginx\_proxy.sh"  interval 2  weight 2  }  vrrp\_instance VI\_1 {  state MASTER  interface ens33  virtual\_router\_id 51  priority 150  advert\_int 1  authentication {  auth\_type PASS  auth\_pass 1111  }  virtual\_ipaddress {  10.10.10.10/24 dev ens33  }  #调用chk\_nginx\_proxy  track\_script {  chk\_nginx\_proxy  }  } | ! Configuration File for keepalived  global\_defs {  notification\_email {  122123498@qq.com  }  notification\_email\_from 122123498.com  smtp\_server 192.168.200.1  smtp\_connect\_timeout 30  #id，不同的keepalived.conf，此ID要唯一，主备不能相同  router\_id LVS\_DEVEL1  #解决多组keepalived在同一个局域网的冲突问题  vrrp\_mcast\_group4 224.0.0.19  vrrp\_skip\_check\_adv\_addr  #vrrp\_strict  vrrp\_garp\_interval 0  vrrp\_gna\_interval 0  }  vrrp\_instance VI\_1 { #实例名字为VI\_1，主备相同  state BACKUP #状态为BACKUP ，主节点需要设为MASTER  interface ens33 #通信接口为ens33，主备相同  virtual\_router\_id 51 #实例ID为51，keepalived.conf里唯一主备相同  priority 100 #优先级为100，备节点的优先级必须低于次数字  advert\_int 1 #检查间隔为1秒  authentication {  auth\_type PASS #PASS认证类型，主备相同  auth\_pass 1111 #密码1111，主备相同  }  virtual\_ipaddress {  10.10.10.10/24 dev ens33 #虚拟VIP .主备相同  }  } |
| **双主keepalive：A业务在lb01上是主，在lb02上是备 而B业务在lb01上是备，在lb02上是主** | |
| vrrp\_instance VI\_2 {  state BACKUP  interface ens33  virtual\_router\_id 55  priority 100  advert\_int 1  authentication {  auth\_type PASS  auth\_pass 1111  }  virtual\_ipaddress {  10.10.10.11/24 dev ens33  } | vrrp\_instance VI\_2 {  state MASTER  interface ens33  virtual\_router\_id 55  priority 150  advert\_int 1  authentication {  auth\_type PASS  auth\_pass 1111  }  virtual\_ipaddress {  10.10.10.10/24 dev ens33  } |

【总结】配置文件修改

Keepalived主备配置文件区别：

01. router\_id 信息不一致

02. state 状态描述信息不一致

03. priority 主备竞选优先级数值不一致

**解决服务监听的网卡上不存在IP地址问题**

如果配置使用listen 10.10.10.10:80 的方式指定IP监听服务，而本地的网卡上没有10.10.10.12这个IP，nginx就会报错

如果要实现双主同时跑不同的业务，配置文件里指定了IP监听，备节点则会因为网卡实际不存在VIP也报错

解决办法：

vim /etc/sysctl.conf

net.ipv4.ip\_nonlocal\_bind = 1

**解决高可用服务只针对物理服务器的问题**

keepalived软件仅仅在对方机器宕机或者keepaived停掉的时候才会接管业务

但是业务停止了而keepalived还在工作，会导致用户访问vip无法找到对应的服务

方法一：监控nginx业务，当有问题的时候停止keepalived服务器，实现VIP漂移

#!/bin/sh

while true

do

if [ `netstat -lntup|grep nginx|wc -l` -ne 1 ];then

systemctl stop keepalived

else

if [ ! `ps -ef|grep keepalived|grep -v grep|wc -l` -gt 1 ];then

systemctl start keepalived

fi

fi

sleep 5

done

方法二：在keepalived配置文件中加入上文中绿色背景的配置文件

即可实现：当nginx挂掉以后主自动停止keepalived 当nginx修复以后，确认服务没有问题、手动启动keepalived

**chk\_nginx\_proxy.sh**

#!/bin/sh

if [ `netstat -lntup|grep nginx|wc -l` -ne 1 ];then

systemctl stop keepalived

fi

**解决多组keepalived服务器在同一个局域网冲突问题**

#解决多组keepalived在同一个局域网的冲突问题

vrrp\_mcast\_group4 224.0.0.19 #默认情况下都会使用此地址作为多播地址

**配置keepalived服务日志**

默认情况下keepalived日志都会记录在系统日志当中var/log/messages

1、sed -i '14 s#KEEPALIVED\_OPTIONS="-D"#KEEPALIVED\_OPTIONS="-D -d -S 0"#g' /etc/sysconfig/keepalived

2、vim /etc/rsyslog.conf

底部加入：local（本地） 0（零）

# keepalived

local0.\* /var/log/keepalived.log

修改大约54行

\*.info;mail.none;authpriv.none;cron.none;local0.none /var/log/messages

完成上面两步骤以后日志就会记录在，设定好的文件里，即：/var/log/keepalived.log

**开发检测keepalived裂脑的脚本**

思路：在备节点上执行脚本、如果可以ping通主节点、并且备节点有VIP就报警。需要人员介入检查是否裂脑

然后整合到zabbix监控里

[root@lb02 /server/scripts]# cat check\_keepalived.sh

#!/bin/sh

lb01\_VIP=10.10.10.10

lb01\_ip=10.10.10.5

while true

do

ping -c 2 -W 3 ${lb01\_ip} &>/dev/null

if [ $? -eq 0 -a `ip add|grep "${lb01\_VIP}"|wc -l` -eq 1 ];then

echo "brain.warning"

else

echo "keepalived is OK"

fi

sleep 5

done