

< 2018年12月 >						
日	一	二	三	四	五	六
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

公告

昵称：MacoLee

园龄：2年7个月

粉丝：36

关注：8

+加关注

搜索

找找看

谷歌搜索

- 常用链接
- 我的随笔

我的评论

我的参与

最新评论

我的标签

- 随笔分类
- Git(1)

python(23)

python之路(1)

Web前端(5)

算法(1)

Keepalived安装使用详解

简介

Keepalived是一个基于VRRP协议来实现的服务高可用方案，可以利用其来避免IP单点故障，类似的工具还有heartbeat、corosync、pacemaker。

但是它一般不会单独出现，而是与其它负载均衡技术（如lvs、haproxy、nginx）一起工作来达到集群的高可用。

Keepalived的作用是检测服务器的状态，如果有一台web服务器死机，或工作出现故障，Keepalived将检测到，并将有故障的服务器从系统中剔除，同时使用其他服务器代替该服务器的工作；

当服务器工作正常后Keepalived自动将服务器加入到服务器群中，这些工作全部自动完成，不需要人工干涉，需要人工做的只是修复故障的服务器。

安装



```
#下载
wget http://www.keepalived.org/software/keepalived-1.2.23.tar.gz
#解压
tar -zxvf keepalived-1.2.23.tar.gz
cd keepalived-1.2.23
#安装
./configure --prefix=/usr/local/keepalived #prefix指定安装目录
make
make install
```



运维(38)
随笔档案
2018年8月 (1)
2017年4月 (1)
2017年3月 (1)
2017年2月 (1)
2017年1月 (1)
2016年12月 (2)
2016年11月 (3)
2016年10月 (1)
2016年9月 (15)
2016年8月 (14)
2016年7月 (20)
2016年6月 (14)
积分与排名
积分 - 85799
排名 - 5054
最新评论
1. Re:Pycharm如何导入Django项目
@dalyday首先,你要在pycharm中打开你的django项目,然后再操作...
--MacoLee
2. Re:Pycharm如何导入Django项目
第二步箭头1处不显示项目名称,麻烦能指点下
--dalyday
3. Re:Ansible之playbook
天道酬勤, -i的用法
--Michael2397
4. Re:Python之Fabric模块
@runs_once #查看本地系统信息,当有多台主机时只运行一次
这句没看懂 查看本地系统信息,怎么会有多台主机呢???
--Lemon_乐
5. Re:HAProxy安装配置详解
赞
--_BLUE
阅读排行榜
1. linux下进程、端口号相互查看方法(36706)

配置
【/usr/local/keepalived/etc/keepalived/keepalived.conf】
<div><pre>global_defs { notification_email { #指定keepalived在发生事情的时候,发送邮件告知,可以有多个地址,每行一个。 acassen@firewall.loc failover@firewall.loc sysadmin@firewall.loc } notification_email_from Alexandre.Cassen@firewall.loc #指定发件人 smtp_server 127.0.0.1 #发送邮件的smtp地址 smtp_connect_timeout 30 #超时时间 router_id LVS_DEVEL #运行keepalived的机器的一个标识,多个节点标识可以相同,也可以不同 vrrp_skip_check_adv_addr vrrp_strict vrrp_garp_interval 0 vrrp_gna_interval 0 } vrrp_instance VI_1 { state MASTER #指定当前节点为主节点 备用节点上设置为BACKUP即可 interface eth0 #绑定虚拟IP的网络接口 virtual_router_id 51 #VRRP组名,两个节点的设置必须一样,以指明各个节点属于同一VRRP组 priority 100 #主节点的优先级(1-254之间),备用节点必须比主节点优先级低 advert_int 1 authentication { #设置验证信息,两个节点必须一致 auth_type PASS auth_pass 1111 } virtual_ipaddress { #指定虚拟IP,两个节点设置必须一样 192.168.1.21/24 } } </pre></div>
【/usr/local/keepalived/etc/sysconfig/keepalived】

2. HAProxy安装配置详解(28054)
3. Nmap命令的29个实用范例(20559)
4. LVS安装使用详解(18316)
5. linux下查看进程运行的时间(18282)
6. linux系统中rsync+inotify实现服务器之间文件实时同步(10439)
7. Pychram如何导入Django项目(8910)
8. Django数据库设计中字段为空的方式(8327)
9. Saltstack系列3: Saltstack常用模块及API(7642)
10. Keepalived安装使用详解(6000)

- 评论排行榜
1. Pychram如何导入Django项目(2)
2. HAProxy安装配置详解(1)
3. Python之Fabric模块(1)
4. Ansible之playbook(1)

- 推荐排行榜
1. HAProxy安装配置详解(5)
2. LVS安装使用详解(4)
3. Nmap命令的29个实用范例(3)
4. linux下查看进程运行的时间(2)
5. linux下进程、端口号相互查看方法(2)
6. Python之Rpyc模块(1)
7. Saltstack系列3: Saltstack常用模块及API(1)
8. CentOS下puppet安装(1)

```
KEEPALIVED_OPTIONS="-D -f /usr/local/keepalived/etc/keepalived/keepalived.conf" #指定keepalived配置文件路径
```

因为我们使用非默认路径（/usr/local）安装keepalived，需要设置一些软链接以保证keepalived能正常启动

```
ln -s /usr/local/keepalived/sbin/keepalived /usr/bin #将keepalived主程序加入到环境变量
ln -s /usr/local/keepalived/etc/rc.d/init.d/keepalived /etc/init.d/ #keepalived启动脚本，放到/etc/init.d/目录下就可以使用service命令便捷调用
ln -s /usr/local/keepalived/etc/sysconfig/keepalived /etc/sysconfig/ #keepalived启动脚本变量引用文件，默认文件路径是/etc/sysconfig/，也可以不做软链接，直接修改启动脚本中文件路径即可
```

启动

```
service keepalived start|stop|restart
chkconfig keepalived on
```

默认的配置文件中，指定了虚拟IP :192.168.1.21，可使用ip addr（或ip a）命令验证之。

应用


keepalived两种模式：主-备、主-主

主-备

虚拟IP: 192.168.1.21 主节点: 192.168.1.23 备用节点: 192.168.1.24

主配置文件

【 /usr/local/keepalived/etc/keepalived/keepalived.conf】



```
global_defs {
    notification_email { #指定keepalived在发生事情的时候，发送邮件告知，可以有多个地址，每行一个。
        acassen@firewall.loc
        failover@firewall.loc
```

```

        sysadmin@firewall.loc
    }

    notification_email_from Alexandre.Cassen@firewall.loc #指定发件人
    smtp_server 127.0.0.1 #发送email的smtp地址
    smtp_connect_timeout 30 #超时时间
    router_id LVS_DEVEL #运行keepalived的机器的一个标识, 多个节点标识可以相同, 也可以不同
    vrrp_skip_check_adv_addr
    vrrp_strict
    vrrp_garp_interval 0
    vrrp_gna_interval 0
}

vrrp_instance VI_1 {
    state MASTER #指定当前节点为主节点 备用节点上设置为BACKUP即可
    interface eth0 #绑定虚拟IP的网络接口
    virtual_router_id 51 #VRRP组名, 两个节点的设置必须一样, 以指明各个节点属于同一VRRP组
    priority 100 #主节点的优先级(1-254之间), 备用节点必须比主节点优先级低
    advert_int 1
    authentication { #设置验证信息, 两个节点必须一致
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress { #指定虚拟IP, 两个节点设置必须一样
        192.168.1.21/24
    }
}

```



备配置文件

【/usr/local/keepalived/etc/keepalived/keepalived.conf】



```

global_defs {
    notification_email { #指定keepalived在发生事情的时候, 发送邮件告知, 可以有多个地址, 每行一个。
        acassen@firewall.loc
        failover@firewall.loc
        sysadmin@firewall.loc
    }
}

```

```

notification_email_from Alexandre.Cassen@firewall.1oc #指定发件人
smtp_server 127.0.0.1 #发送邮件的smtp地址
smtp_connect_timeout 30 #超时时间
router_id LVS_DEVEL #运行keepalived的机器的一个标识,多个节点标识可以相同,也可以不同
vrrp_skip_check_adv_addr
vrrp_strict
vrrp_garp_interval 0
vrrp_gna_interval 0
}

vrrp_instance VI_1 {
    state BACKUP #指定当前节点为主节点 备用节点上设置为BACKUP即可
    interface eth0 #绑定虚拟IP的网络接口
    virtual_router_id 51 #VRRP组名,两个节点的设置必须一样,以指明各个节点属于同一VRRP组
    priority 99 #主节点的优先级(1-254之间),备用节点必须比主节点优先级低
    advert_int 1
    authentication { #设置验证信息,两个节点必须一致
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress { #指定虚拟IP,两个节点设置必须一样
        192.168.1.21/24
    }
}

```



主-主

虚拟IP: 192.168.1.21、192.168.1.22 A节点: 192.168.1.23 B节点: 192.168.1.24

主备模式的缺点就是始终只有一台机器位于工作状态,另外一台机器永远是备用状态,存在资源浪费之问题。

双主模式允许两台机器均处于工作状态并互相作为备份。搭建keepalived双方模式的要素:



1. 必须有两个虚拟IP, 分别绑定至两个节点上

2. 每个节点作为某个虚拟IP的主节点，并同时作为另外一个虚拟IP的备用节点。

3. 当某个节点产生故障时，两个虚拟IP自动绑定至正常节点上

也就是说，两个节点的配置应该是交叉的，对同个虚拟IP，交叉互为主备。



节点A配置文件

【/usr/local/keepalived/etc/keepalived/keepalived.conf】



```
global_defs {
    notification_email { #指定keepalived在发生事情的时候，发送邮件告知，可以有多个地址，每行一个。
        acassen@firewall.loc
        failover@firewall.loc
        sysadmin@firewall.loc
    }
    notification_email_from Alexandre.Cassen@firewall.loc #指定发件人
    smtp_server 127.0.0.1 #发送email的smtp地址
    smtp_connect_timeout 30 #超时时间
    router_id LVS_DEVEL #运行keepalived的机器的一个标识,多个节点标识可以相同，也可以不同
    vrrp_skip_check_adv_addr
    vrrp_strict
    vrrp_garp_interval 0
    vrrp_gna_interval 0
}


vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 51 #本机两个vrrp_instance组的此值不能相同，但对应备用节点的此值必须相同
    priority 100 #对应备用节点值应该比此值小
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
}
```

```

    }
    virtual_ipaddress {
        192.168.1.21/24
    }
}

vrrp_instance VI_2 {
    state BACKUP
    interface eth0
    virtual_router_id 52 #本机两个vrrp_instance组的此值不能相同，但对应备用节点的此值必须相同
    priority 99 #主节点的值应该比此值大
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        192.168.1.22/24
    }
}

```



节点B配置文件

【/usr/local/keepalived/etc/keepalived/keepalived.conf】



```

global_defs {
    notification_email { #指定keepalived在发生事情的时候，发送邮件告知，可以有多个地址，每行一个。
        acassen@firewall.loc
        failover@firewall.loc
        sysadmin@firewall.loc
    }

    notification_email_from Alexandre.Cassen@firewall.loc #指定发件人
    smtp_server 127.0.0.1 #发送email的smtp地址
    smtp_connect_timeout 30 #超时时间
    router_id LVS_DEVEL #运行keepalived的机器的一个标识,多个节点标识可以相同，也可以不同
    vrrp_skip_check_adv_addr

```

```

vrrp_strict
vrrp_garp_interval 0
vrrp_gna_interval 0
}

vrrp_instance VI_1 {
    state BACKUP
    interface eth0
    virtual_router_id 51 #本机两个vrrp_instance组的此值不能相同，但对应备用节点的此值必须相同
    priority 99 #对应主节点值应该比此值大
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        192.168.1.21/24
    }
}

vrrp_instance VI_2 {
    state MASTER
    interface eth0
    virtual_router_id 52 #本机两个vrrp_instance组的此值不能相同，但对应备用节点的此值必须相同
    priority 100 #对应备用节点的值应该比此值小
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        192.168.1.22/24
    }
}

```



两个节点实际都处于正常工作状态，但是无法接收到彼此的组播通知，这时两个节点均强行绑定虚拟IP，导致不可预料的后果。

这时就需要设置仲裁，即每个节点必须判断自身的状态（应用服务状态及自身网络状态），要实现这两点可使用自定义shell脚本实现，通过周期性地检查自身应用服务状态，并不断ping网关（或其它可靠的参考IP）均可。当自身服务异常、或无法ping通网关，则认为自身出现故障，就应该移除掉虚拟IP(停止keepalived服务即可)。

主要借助keepalived提供的vrrp_script及track_script实现：

在keepalived的配置文件最前面加入以下代码，定义一个跟踪脚本：

```
vrrp_script check_local { #定义一个名称为check_local的检查脚本
    script "/usr/local/keepalived/bin/check_local.sh" #shell脚本的路径
    interval 5 #运行间隔
}
```

再在vrrp_instance配置中加入以下代码使用上面定义的检测脚本：

```
track_script {
    check_local
}
```

我们在/usr/local/keepalived/bin/check_local.sh定义的检测规则是：

1. 自身web服务故障（超时，http返回状态不是200）
2. 无法ping通网关
3. 产生以上任何一个问题，均应该移除本机的虚拟IP(停止keepalived实例即可)

但这里有个小问题，如果本机或是网关偶尔出现一次故障，那么我们不能认为是服务故障。更好的做法是如果连续N次检测本机服务不正常或连接N次无法ping通网关，才认为是故障产生，才需要进行故障转移。

但这么做的缺点是，如果脚本检测到故障产生，并停止掉了keepalived服务，那么当故障恢复后，keepalived是无法自动恢复的。

还可以利用独立的脚本以秒级的间隔检查自身服务及网关连接性，再根据故障情况控制keepalived的运行或是停止。

在每个节点运行shell脚本（check_service.sh）检测本机的服务是否正常，一旦检测到服务异常时，停止掉本机的keepalived，如此虚拟IP自动转移到备用机器之上，如每隔3秒检测一次本机服务状态，如果连接3次检测失败，则停止掉keepalived实例。同时如果本机服务是正常的，但是keepalived没有启动（故障恢复之后），则启动keepalived，以达到故障恢复之目的。

check_service.sh文件的内容



```
#!/bin/bash

pidfile=/var/lock/subsys/`basename $0`.pid
if [ -f $pidfile ] && [ -e /proc/`cat $pidfile` ] ; then
    exit 1
fi

trap "rm -fr $pidfile ; exit 0" 1 2 3 15
echo $$ > $pidfile

maxfails=3
fails=0
success=0

while [ 1 ]
do
    /usr/bin/wget --timeout=3 --tries=1 http://127.0.0.1/ -q -O /dev/null
    if [ $? -ne 0 ] ; then
        let fails=${fails+1}
        success=0
    else
        fails=0
        let success=${success+1}
    fi

    if [ $fails -ge $maxfails ] ; then
        fails=0
        success=0

        #check keepalived is running ? try to stop it
        service keepalived status | grep running
        if [ $? -eq 0 ] ; then
            logger -is "local service fails $maxfails times ... try to stop keepalived."
            service keepalived stop 2>&1 | logger
        fi
    fi
fi
```

```
if [ $success -gt $maxfails ] ; then
    #check keepalived is stopped ? try to start it
    service keepalived status | grep stopped
    if [ $? -eq 0 ] ; then
        logger -is "service changes normal, try to start keepalived ."
        service keepalived start
    fi
    success=0
fi
sleep 3
done
```



两个节点上均应运行此脚本，请将此脚本加入到**cron**任务中（此程序已经作了单实例运行机制，加入计划任务的作用就是防止脚本意外中断后检测功能失效），可实现的功能：

如果本地服务连续三次检测失败，就尝试停止**keepalived**服务（如果**keepalived**处于运行状态）

如果本地服务连接三次检测成功，但**keepalived**没有启动，则启动之

关键的执行点，均已经记录到系统日志中（**/var/log/messages**）

执行**crontab -e**，加入以下内容：

```
*/1 * * * * /root/check_service.sh
```

停止掉本机的**keepalived**，稍过一会，就会**keepalived**服务被自动启动了（这是因为本地服务检测正常）

停止掉本机的**nginx**，稍过一会，就会发现**keepalived**服务也被停止掉了

再启动**nginx**，稍过一会，发现**keepalived**也被正常启动，并绑定了正确的虚拟IP

参考资料：

<http://zhangxugg-163-com.iteye.com/blog/1665419>

分类: [运维](#)

标签: [运维](#), [shell](#), [负载均衡](#), [keepalived](#)

好文要顶

关注我

收藏该文



MacoLee

关注 - 8

粉丝 - 36

+加关注

« 上一篇: [Django缓存使用方法](#)

» 下一篇: [HAProxy安装配置详解](#)

posted on 2016-09-08 15:53 [MacoLee](#) 阅读(6000) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问](#) 网站首页。

【推荐】超50万VC++源码: 大型组态工控、电力仿真CAD与GIS源码库!

【活动】华为云12.12会员节全场1折起 满额送Mate20

【活动】华为云会员节云服务特惠1折起

【推荐】服务器100%基准CPU性能, 1核1G首年168元, 限时特惠!

腾讯云

腾讯云AMD云服务器

节省IT成本30%

1核1G AMD机型**0.57元/天起**

立即抢购

相关博文:

· [窗口融合技术](#)

· [>>用.NET CF进行HardReset!](#)

- 开源项目—[OpenPOP.NET](#) 已经发布v1.3a版本！
- [贾静雯相关视频下载来源汇总](#)
- [Is This My Destiny?](#)



最新新闻：

- [为什么中国至今没有像样的国产操作系统？](#)
- [微软可能正在打造普通人能买到的HoloLens](#)
- [健身器材越来越智能了，我们还需要健身教练吗？](#)
- [社交江河，风浪四起](#)
- [如何在Unity实现从纹理中生成法线贴图？](#)

» [更多新闻...](#)

Powered by:

[博客园](#)

Copyright © MacoLee