

目 录

[关于版块](#)

[Keepalived](#)

[Heartbeat](#)

关于版块

关于Keepalived

简介：Keepalived是基于vrrp协议的一款高可用软件。在主服务器和备份服务器上面部署相同的服务配置，使用一个虚拟IP地址对外提供服务，当主服务器出现故障时，虚拟IP地址会自动漂移到备份服务器。

关于Heartbeat

简介：Heartbeat是Linux-HA项目中的一个组件，它实现了一个高可用集群系统。心跳服务和集群通信是高可用集群的两个关键组件，在Heartbeat项目里，由heartbeat模块实现了这两个功能。

组成：

- heartbeat：节点间通信检测模块
- ha-logd：集群事件日志服务
- CCM（Consensus Cluster Membership）：集群成员一致性管理模块
- LRM（Local Resource Manager）：本地资源管理模块
- Stonith Daemon：使出现问题的节点从集群环境中脱离
- CRM（Cluster resource management）：集群资源管理模块
- Cluster policy engine：集群策略引擎
- Cluster transition engine：集群转移引擎

注：Heartbeat自2015年后就不提供更新了，一般情况下推荐使用keepalived。

本版块维护人员

版主：子木

QQ：1242119478

Keepalived

安装

基础环境：服务器：test11-192.168.37.11，test12-192.168.37.12

系统版本：centos7.6

软件版本：keepalived 1.3.5

直接yum安装：`yum install keepalived -y`

最新版安装的可以到官网下载源码包编译安装：<https://www.keepalived.org/download.html>

配置

主服务器：test11

```
[root@test11 ~]#vim /etc/keepalived/keepalived.conf
global_defs {
    .
    .
    router_id test11          #不同的keepalived.conf，此ID要唯一，ID名自行
    定义
    .
    .
}
vrrp_instance VI_1 {          #实例名为VI_1，相同实例的备节点要和这个相同
    state MASTER               #状态为MASTER，备节点状态需要为BACKUP
    interface enp0s3           #注意：默认为eth0，centos7的网卡名可能不是eth0
    , 修改成实际的网卡名
    virtual_router_id 51       #实例ID为51，keepalived.conf里唯一，但备节点
    要相同
    priority 150               #优先级为150，备节点优先级必须比此数字低
    advert_int 1               #主备通讯时间间隔1秒
    authentication {
        auth_type PASS        #PASS认证类型，此参数备节点设置和主节点相同
        auth_pass 1111        #密码是1111，此参数备节点设置和主节点相同
    }
    virtual_ipaddress {
        192.168.37.9/24        #虚拟ip地址（VIP，一个尚未占用的内网ip即可）
    }
}
```

备服务器：test12

```
[root@test12 ~]# vim /etc/keepalived/keepalived.conf
```

```

global_defs {
.
.
    router_id test12          #不同的keepalived.conf, 此ID要唯一, ID名自行
    定义
.
.
}
vrrp_instance VI_1 {
    state BACKUP
    interface enp0s3
    virtual_router_id 51
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        192.168.37.9/24
    }
}
}

```

注意：虽然centos7防火墙默认用的是firewall,但这里需要执行 `iptables -F` 清除一下防火墙虚拟ip才能ping通

在test11和test12上启动keepalived, 执行命令: `systemctl start keepalived`

简单模拟keepalived主服务器挂掉后, vip有没有漂移到备服务器上:

- 在test11上停止keepalived服务: `systemctl stop keepalived`
- 在test11上执行ip a查看vip已经消失了
- 在test12上挂靠ip a查看上面设置的vip 192.168.37.9已漂移至keepalived备服务器test12上来了

双主双实例

简介: keepalived支持多实例多业务双向主备模式, 只要在配置文件止新增一个实例就可以, 即A业务在负载1服务器上主模式, 在负载2上是备模式, 而B业务在负载1是备模式, 在负载2是主模式。

服务器: test11

```

[root@test11 ~]# vim /etc/keepalived/keepalived.conf
global_defs {
.
.
    router_id test11
.
.
}
vrrp_instance VI_1 {

```

```

#跟上面test11的一样
}
vrrp_instance VI_2 {
    state BACKUP
    interface enp0s3
    virtual_router_id 52
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        192.168.37.10/24
    }
}

```

服务器：test12

```

[root@test12 ~]# vim /etc/keepalived/keepalived.conf
global_defs {
.
router_id test12
.
}
vrrp_instance VI_1 {
    #跟test12的上面的一样
}
vrrp_instance VI_2 {
    state MASTER
    interface enp0s3
    virtual_router_id 52
    priority 150
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        192.168.37.10/24
    }
}

```

Nginx负载均衡配合Keepalived服务案例

- 把test11上的nginx配置的负载均衡配置拷贝到test12的nginx配置文件路径上，并用rsync定时同步。

- 客户端把负载均衡代理的域名都解析到 VIP:192.168.37.9 上来。
- 为了解决单个nginx负载重的问题，可以根据业务需求使用上面的双主双实例模式。

Keepalived高可用服务器“裂脑”问题

简介：由于某些原因，导致两台高可用服务器在指定时间内，无法检测到对方的心跳消息，各自取得资源及服务的所有权，而那时的两台高可用服务器对都还活着并在正常运行，这样会导致同一个IP或服务在两端同时存在而发生冲突，最严重的是两台主机占用同一个VIP地址，当用户写入数据时可能会分别写入到两端，这可能会导致服务器两端的数据不一致或造成数据丢失，这种情况被称为裂脑。

原因：

- 高可用服务器对之间心跳线链路发生故障，导致无法通信。
- 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，导致心跳失败
- 其他服务配置不当，如心跳方式不同，心跳广播冲突，软件Bug等。

解决：

- 同时使用串口电缆和以太网电缆连接，同时使用两条心跳线路，一条坏了，另一个依然可以传播心跳
- 做好对裂脑的监控报警，在发生时人为第一时间介入仲裁

解决多组keepalived服务器在一个局域网的冲突问题

当在同一个局域网内部署了多组keepalived服务器，且未使用专门的心跳线通信时，可能会发生高可用接管严重故障问题。Keepalive高可用功能是通过VRRP协议实现的，VRRP协议默认通过IP多播的形式实现高可用对之间的通信，如果同一个局域网内存在多组keepalived服务器对，就会造成IP多播地址冲突问题，导致接管错乱，不同组的keepalived都会使用默认的224.0.0.18作为多播地址。此时解决办法是，在同组的keepalived服务器所有的配置文件指定独一无二的多播地址，配置如下：

```
global_defs {
router_id test11
vrrp_mcast_group4 224.0.0.19
}
```

解决高可用服务只针对物理服务器的问题

默认情况下，keepalived软件仅仅在对方机器宕机或者keepalived停掉的时候才会接管业务。但实际工作中，有业务服务停止而keepalived服务还在工作的情况，这就会导致用户访问的VIP无法找到对应的服务，那么，如何解决业务服务宕机可以将VIP漂移到备节点上呢

例如可以定时任务监控nginx，服务有问题的就把keepalived停掉：

```
[root@test11 ~]# cat /scripts/chk_nginx_proxy.sh
#!/bin/bash

nginx_status=`curl -I -m 5 -s -w "%{http_code}\n" -o /dev/null 192.168.37.11`
```

Keepalived

```
if [ $nginx_status -ne 200 ];then
    systemctl stop keepalived.service
fi
```

Heartbeat

安装

基础环境：主服务器：test11-192.168.37.11；从服务器：test12-192.168.37.12

系统版本：centos6.x

注：heartbeat不提供centos7的yum安装包，如果需要使用，推荐在centos6.x下安装使用

在主从服务器上添加hosts：

```
vim /etc/hosts
192.168.37.11 Master.HA
192.168.37.12 Slave.HA
```

配置主、从两台服务ssh互信，如下：

test11:

```
ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa
ssh-copy-id -i .ssh/id_rsa.pub root@Slave.HA
```

test12:

```
ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa
ssh-copy-id -i .ssh/id_rsa.pub root@Master.HA
```

直接yum安装： `yum install -y epel-release,heartbeat,libnet`

配置：

`cp /usr/share/doc/heartbeat-3.0.4/{authkeys,ha.cf,haresources} /etc/ha.d/`

去掉下面注释

```
vim /etc/ha.d/authkeys
auth 3
3 md5 Hello!
```

修改权限为600，不然heartbeat无法启动： `chmod 600 /etc/ha.d/authkeys`

修改/etc/ha.d/haresource,在后面追加（nginx为监测那个服务，如果是Apache的，更改为：httpd）：

```
Master.HA IPaddr::192.168.139.10/24/eth0:0 nginx
```

修改/etc/ha.d/ha.cf:


```

debugfile    /var/log/ha-debug    #调试日志
logfile      /var/log/ha-log    #日志文件，和上面的调试记录的信息类似
logfacility   local0        #日志的类型，级别
keepalive    2             #单位是秒，2 s探测一次
deadtime     30            #如果30 s没有响应，则认为死亡
warntime     10            #10 s内没有响应，则警告一次
initdead     60            #服务初始化时间
udpport      694           #ha使用的端口
ucast        eth0 192.168.37.12 #组播地址，通过eth0接口来探测从主机是否正
常
auto_failback on          #如果主恢复正常，是否回切
node         Master.HA    #主机节点
node         Slave.HA     #从节点
ping         192.168.37.1  #仲裁地址，来判断主从是否正常，一般会选择
网关
respawn      hacluster    /usr/lib64/heartbeat/ipfail
#以hacluster的身份来执行后面的脚本，respawn是脚本的执行方式，这个脚本主要用来判断网
络是否正常

```

复制上面的3个配置文件到从设备对应的/etc/ha.d目录下

```
scp /etc/ha.d/{authkeys,ha.cf,haresources} 192.168.37.12:/etc/ha.d/
```

从设备上的配置文件，只需要修改/etc/ha.d/ha.cf

```

debugfile    /var/log/ha-debug
logfile      /var/log/ha-log
logfacility   local0
keepalive    2
deadtime     30
warntime     10
initdead     60
udpport      694
ucast        eth0 192.168.37.11 #这里填写主服务器的ip，和主服务器上的配
置正好相反
auto_failback on
node         Master.HA
node         Slave.HA
ping         192.168.139.2
respawn      hacluster    /usr/lib64/heartbeat/ipfail

```

启动服务：启动主，再启动从。启动主服务时，虚拟ip要过一会才可以看到。

报错：

启动过程中如果出现这人错误的ERROR: Client child command [/usr/lib/heartbeat/ipfail] is not executable

解决：把/etc/ha.d/ha.cf下的

```
respawn hacluster /usr/lib/heartbeat/ipfail
```

更改为

```
respawn hacluster /usr/lib64/heartbeat/ipfail
```

测试:

在主从上分别修改两个index页面的内容，以作区分，这里公供测试

a、echo Master... > /usr/share/nginx/html/index.html

b、echo Slave... >/usr/share/nginx/html/index.html

初次在浏览器中输入：192.168.37.10，看到的是Master

主设备停掉heartbeat,或关机，看到的是Slave

主设备重启hearbeat或重新开机，看到的是Master