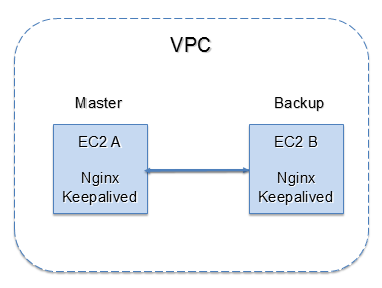
简介

AWS EC2不支持组播和广播，只支持单播。Keepalived是基于VRRP组播的应用，但是在2013年8月5日发布的版本1.2.8中添加了对VRRP单播的支持，所以keepalived可以被应用到AWS中，作为对EC2, HAProxy, LVS等的高可用解决方案。本文以在EC2上搭建Nginx Web服务器为例，展示如何利用keepalived单播特性实现web服务器的高可用，如下图所示：

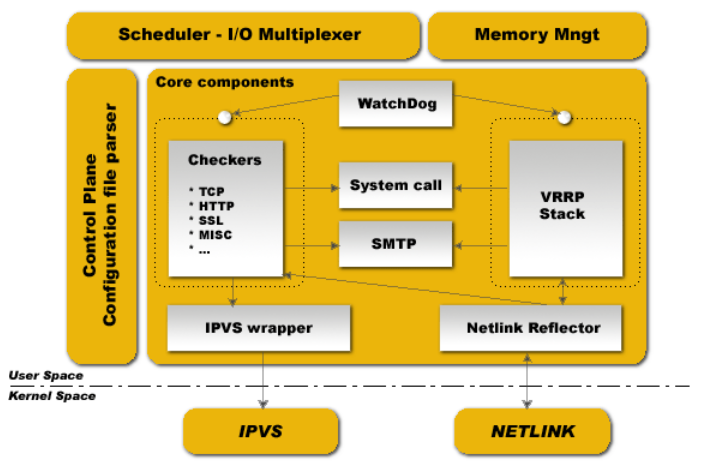


与<http://aws.amazon.com/articles/Amazon-EC2/2127188135977316>文章中提出的高可用方案相比，keepalived能提供更多的好处：

1. 利用keepalived建立起的高可用系统可以防止脑裂 ("brain split", 由于心跳包丢失而造成的故障)。

2. 可以利用vrrp\_script脚本为应用层监控的进程提供很高的灵活性。

Keepalived软件架构



PID

111 Keepalived <-- Parent process monitoring children

112 \\_ Keepalived <-- VRRP child

113 \\_ Keepalived <-- Healthchecking child

1、 WatchDog 负责监控checkers 和VRRP 进程的状况。

2、 Checkers 负责真实服务器的健康检查healthchecking.

**3、 VRRP Stack 负责负载均衡器之间的失败切换FailOver。**

4、 IPVS wrapper 用来发送设定的规则到内核ipvs 代码。

5、 Netlink Reflector 用来设定 vrrp 的vip 地址等。

Keepalived + Nginx 环境搭建

**步骤1：创建安全组**

在默认VPC (10.0.1.0/16)中创建安全组，名为”Keepalvied\_HA\_SG”，运行任意(0.0.0.0/0)源IP地址入站访问SSH和HTTP协议。

**步骤2：创建IAM角色 (IAM Role)**

创建EC2 IAM角色，让EC2实例有权限接管VIP。

1. 在IAM服务中 – 角色 – 创建角色
2. 设置角色为”Keepalived\_HA\_Role”
3. 选择角色类型 – Amazon EC2
4. 设置权限 – 选择策略模板 – Amazon EC2 Full Access

**步骤3：启动2个EC2实例**

1. 启动Amazon Linux EC2实例”Keepalived\_HA\_1”

设置实例详情：

设置子网为10.0.1.0/20

设置IAM角色为”Keepalived\_HA\_Role”

设置网络接口主IP地址为10.0.1.101。并且添加辅助私有IP地址10.0.1.100，这个辅助私有IP地址将在发生故障时在两个实例间切换。

配置安全组：

选择已经存在的安全组”Keepalived\_HA\_SG”

1. 启动Amazon Linux EC2实例”Keepalived\_HA\_2”

设置实例详情：

设置子网为10.0.1.0/20

设置IAM角色为”Keepalived\_HA\_Role”

设置网络接口主IP地址为10.0.1.102

配置安全组：

选择已经存在的安全组”Keepalived\_HA\_SG”

**步骤4：为实例配置弹性IP (EIP)**

1. 分配三个IP
2. EIP 1映射到实例”Keepalived\_HA\_1”的主私有IP地址10.0.1.101
3. EIP 2映射到实例”Keepalived\_HA\_2”的主私有IP地址10.0.1.102
4. EIP 3映射到实例”Keepalived\_HA\_1”的辅助私有IP地址10.0.1.100。该EIP地址将做为VIP地址，对外提供高可用。

**步骤5：在2个EC2实例上安装Nginx服务**

1. 在实例"Keepalived\_HA\_1"上安装Nginx Web Server

$ sudo yum install nginx -y

$ sudo chkconfig nginx on

$ sudo service nginx start

# echo "server A" > /usr/share/nginx/html/index.html

$ sudo service nginx reload

1. 在实例"Keepalived\_HA\_2"上安装Nginx Web Server

$ sudo yum install nginx -y

$ sudo chkconfig nginx on

$ sudo service nginx start

# echo "server B" > /usr/share/nginx/html/index.html

$ sudo service nginx reload

**步骤6：在2个EC2实例上安装keepalived服务**

因为需要安装高于1.2.8版本的keepalived，在下面的步骤中利用源代码安装比较新的版本。

1. 在EC2实例"Keepalived\_HA\_1"上安装keepalived

$ sudo yum -y install gcc make openssl openssl-devel

$ sudo mkdir -p /usr/local/src/hasoft

$ cd /usr/local/src/hasoft

$ sudo wget http://www.keepalived.org/software/keepalived-1.2.13.tar.gz

$ sudo tar -zxvf keepalived-1.2.13.tar.gz

$ cd keepalived-1.2.13

$ sudo ./configure --prefix=/usr/local/keepalived

$ sudo make

$ sudo make install

$sudo cp /home/ec2-user/keepalived-1.2.13/bin/keepalived /etc/sysconfig/

$ sudo cp /home/ec2-user/keepalived-1.2.13/keepalived/etc/init.d/keepalived.init /etc/init.d/

sudo cp /home/ec2-user/keepalived-1.2.13/bin/keepalived /usr/sbin/

sudo service keepalived start

1. 在EC2实例"Keepalived\_HA\_2"上安装keepalived

按照与上面"Keepalived\_HA\_1"相同的步骤在"Keepalived\_HA\_1"安装keepalived

**步骤7：在2个EC2实例上配置keepalived**

1. 在EC2实例"Keepalived\_HA\_1"上做如下配置，该EC2实例初始状态为MASTER节点
2. 编辑keepalived配置文件/etc/keepalived/keepalived.conf，如下

[ec2-user@ip-10-0-1-101 keepalived]$ cat /etc/keepalived/keepalived.conf

! Configuration File for keepalived

vrrp\_script nginx\_check

{

# script to watch the nginx process

script "pidof nginx"

interval 1

weight -60 # 如果nginx服务出现故障，本节点VRRP的优先级减60，从默认的150变成90

}

vrrp\_instance VI\_1 {

state MASTER # 节点初始状态为MASTER

interface eth0

virtual\_router\_id 51

priority 150 # MASTER节点设置成150, BACKUP节点设置成100

unicast\_src\_ip 10.0.1.101

unicast\_peer {

10.0.1.102

}

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

track\_script

{

nginx\_check

}

notify\_master "/etc/keepalived/vrrp.sh > /etc/keepalived/vrrp.sh.log 2>&1"

}

1. 创建脚本/etc/keepalived/vrrp.sh，该脚本负责VIP地址的切换。

注意：在该脚本中会用到awscli命令行工具，该工具的安装和配置方法如文档http://docs.amazonaws.cn/cli/latest/userguide/cli-chap-getting-set-up.html

本文使用的Amazon Linux系统默认安装了awscli命令行工具，只需要配置AWS Access Key ID和AWS Secret Access Key即可。

 $ cat vrrp.sh

#!/bin/bash

set -ex

attachmentid=`aws ec2 describe-network-interface-attribute --network-interface-id eni-8189b2d8 --attribute attachment --region cn-north-1 --output text | grep ATTACHMENT| cut -f 3`

instanceid=`/usr/bin/curl --silent http://169.254.169.254/latest/meta-data/instance-id`

aws ec2 detach-network-interface --attachment-id $attachmentid --region cn-north-1

result=`aws ec2 describe-network-interface-attribute --network-interface-id eni-8189b2d8 --attribute attachment --query 'Attachment.Status' --output text --region cn-north-1`

while [ "$result" != "None" ]

do

sleep 1

result=`aws ec2 describe-network-interface-attribute --network-interface-id eni-8189b2d8 --attribute attachment --query 'Attachment.Status' --output text --region cn-north-1`

done

aws ec2 attach-network-interface --network-interface-id eni-8189b2d8 --instance-id $instanceid --device-index 2 --region cn-north-1

1. 给vrrp.sh文件添加执行权限

[ec2-user@ip-10-0-1-101 keepalived]$ sudo chmod a+x vrrp.sh

2. 在EC2实例"Keepalived\_HA\_2"上做如下配置，该EC2实例初始状态为BACKUP节点

1) 编辑keepalived配置文件/etc/keepalived/keepalived.conf，如下：

[ec2-user@ip-10-0-1-102 keepalived]$ cat /etc/keepalived/keepalived.conf

! Configuration File for keepalived

vrrp\_script nginx\_check

{

#script to watch the nginx process

script "pidof nginx"

interval 1

weight -60

}

vrrp\_instance VI\_1 {

state BACKUP # 节点初始状态为BACKUP

interface eth0

virtual\_router\_id 51

priority 100

unicast\_src\_ip 10.0.1.102

unicast\_peer {

10.0.1.101

}

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

track\_script

{

nginx\_check

}

notify\_master "/etc/keepalived/vrrp.sh > /etc/keepalived/vrrp.sh.log 2>&1"

}

2) 创建脚本文件/etc/keepalived/vrrp.sh

$ cat vrrp.sh

#!/bin/bash

set -ex

attachmentid=`aws ec2 describe-network-interface-attribute --network-interface-id eni-8189b2d8 --attribute attachment --region cn-north-1 --output text | grep ATTACHMENT| cut -f 3`

instanceid=`/usr/bin/curl --silent http://169.254.169.254/latest/meta-data/instance-id`

aws ec2 detach-network-interface --attachment-id $attachmentid --region cn-north-1

result=`aws ec2 describe-network-interface-attribute --network-interface-id eni-8189b2d8 --attribute attachment --query 'Attachment.Status' --output text --region cn-north-1`

while [ "$result" != "None" ]

do

sleep 1

result=`aws ec2 describe-network-interface-attribute --network-interface-id eni-8189b2d8 --attribute attachment --query 'Attachment.Status' --output text --region cn-north-1`

done

aws ec2 attach-network-interface --network-interface-id eni-8189b2d8 --instance-id $instanceid --device-index 2 --region cn-north-1

1. 为脚本vrrp.sh添加执行权限

[ec2-user@ip-10-0-1-102 keepalived]$ sudo chmod a+x vrrp.sh

1. 在2个实例上都启动keepalived服务

$ sudo service keepalived start

**步骤8：测试“在Nginx服务出现故障时的Master和Backup节点的选举过程”**

1. 准备另一个观察实例，在该实例上查看Nginx Web服务上的内容。

在初始状态下，利用curl工具查看VIP(EIP 3，辅助IP对应的EIP，在本例子中为54.223.190.243

)的内容，如下：

$ curl 54.223.190.243

server A

2. 在EC2实例"Keepalived\_HA\_1"上, 利用tcpdump查看VRRP 数据包如下：

sudo tcpdump -n port not 22

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode

listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes

07:53:59.835674 IP 10.0.1.101 > 10.0.1.102: VRRPv2, Advertisement, vrid 51, prio 150, authtype simple, intvl 1s, length 16

07:54:00.836785 IP 10.0.1.101 > 10.0.1.102: VRRPv2, Advertisement, vrid 51, prio 150, authtype simple, intvl 1s, length 16

From the capture, I can observe that master node is sending VRRP unicast packet to backup node with priority 150.

通过抓包结果可知，Master节点向Backup节点周期性地发送Advertisement报文，优先级为150。

3. 在Master节点"Keepalived\_HA\_1"上手动停止Nginx服务，并观察日志

1) 停止节点"Keepalived\_HA\_1"上的Nginx服务

sudo service nginx stop

Stopping nginx: [ OK ]

2) 在节点"Keepalived\_HA\_1"上查看keepalived日志

sudo tail -f /var/log/messages

......

Jan 1 07:56:31 ip-10-0-1-101 Keepalived\_vrrp[27757]: VRRP\_Script(nginx\_check) failed

Jan 1 07:56:33 ip-10-0-1-101 Keepalived\_vrrp[27757]: VRRP\_Instance(VI\_1) Received higher prio advert

Jan 1 07:56:33 ip-10-0-1-101 Keepalived\_vrrp[27757]: VRRP\_Instance(VI\_1) Entering BACKUP STATE

From the output, nginx checking script failed, the priority changed to 90 (150 - 60 = 90), current node received higher prio (100) advertisement from node "Keepalived\_HA\_2", so "Keepalived\_HA\_1" is entring backup state.

通过日志的屏幕输出可知，监控脚本发现Nginx服务失败，将当前的VRRP优先级减60变成90。当前节点发现自己接收到了来自节点"Keepalived\_HA\_2"的更高优先级(100)的advertisement报文，所以"Keepalived\_HA\_1"节点进入Backup状态。

3) 同时在节点"Keepalived\_HA\_2"上查看keepalived日志

[ec2-user@ip-10-0-1-102 ~]$ sudo tail -f /var/log/messages

......

Jan 1 07:56:33 ip-10-0-1-102 Keepalived\_vrrp[22092]: VRRP\_Instance(VI\_1) forcing a new MASTER election

Jan 1 07:56:33 ip-10-0-1-102 Keepalived\_vrrp[22092]: VRRP\_Instance(VI\_1) forcing a new MASTER election

Jan 1 07:56:34 ip-10-0-1-102 Keepalived\_vrrp[22092]: VRRP\_Instance(VI\_1) Transition to MASTER STATE

Jan 1 07:56:35 ip-10-0-1-102 Keepalived\_vrrp[22092]: VRRP\_Instance(VI\_1) Entering MASTER STATE

From the output, the origin backup node changed to master status.

从日志屏幕输出可知，原来的Backup节点"Keepalived\_HA\_2"进入了Master状态，成为Master节点。

4. 在观察实例上，查看Web服务的内容。此时VIP的指向了节点"Keepalived\_HA\_2"

$ curl 54.223.190.243

server B

5. 在节点"Keepalived\_HA\_2"上重新启动Nginx服务，在当前的Backup节点"Keepalived\_HA\_1" 上观察日志

1) 在节点"Keepalived\_HA\_1"上启动Nginx服务

[ec2-user@ip-10-0-1-101 keepalived]$ sudo service nginx start

Starting nginx: [ OK ]

2) 在节点"Keepalived\_HA\_1"上查看keepalived的日志

[ec2-user@ip-10-0-1-101 ~]$ sudo tail -f /var/log/messages

......

Jan 1 08:22:44 ip-10-0-1-101 Keepalived\_vrrp[27757]: VRRP\_Script(nginx\_check) succeeded

Jan 1 08:22:45 ip-10-0-1-101 Keepalived\_vrrp[27757]: VRRP\_Instance(VI\_1) forcing a new MASTER election

Jan 1 08:22:45 ip-10-0-1-101 Keepalived\_vrrp[27757]: VRRP\_Instance(VI\_1) forcing a new MASTER election

Jan 1 08:22:46 ip-10-0-1-101 Keepalived\_vrrp[27757]: VRRP\_Instance(VI\_1) Transition to MASTER STATE

Jan 1 08:22:47 ip-10-0-1-101 Keepalived\_vrrp[27757]: VRRP\_Instance(VI\_1) Entering MASTER STATE

通过日志的屏幕输出可知，监控脚本发现Nginx服务运行正常，将当前的VRRP优先级增加60变回程150。当前节点能感知到自己任然在接收来自节点"Keepalived\_HA\_2"的优先级为100的advertisement报文，低于自己的优先级，会强制新Master节点的选举。当"Keepalived\_HA\_2"节点收到"Keepalived\_HA\_1"的优先级更高的通告后，选举过程完成，"Keepalived\_HA\_1"节点进入Master状态。

3) 在EC2节点"Keepalived\_HA\_2"上查看keepalived日志

[ec2-user@ip-10-0-1-102 ~]$ sudo tail -f /var/log/messages

......

Jan 1 08:22:45 ip-10-0-1-102 Keepalived\_vrrp[22092]: VRRP\_Instance(VI\_1) Received higher prio advert

Jan 1 08:22:45 ip-10-0-1-102 Keepalived\_vrrp[22092]: VRRP\_Instance(VI\_1) Entering BACKUP STATE

从日志屏幕输出可知，原来的Master节点"Keepalived\_HA\_2"进入了Backup状态，重新成为Backup节点。

4. 在观察实例上，查看Web服务的内容。此时VIP的指向了节点"Keepalived\_HA\_1"

$ curl 54.223.190.243

server A