

生产环境搭建高可用Harbor (PG13故障自动切换篇)

原创 青牛踏雪 Kubernetes技术栈

各个概述：

在 [生产环境搭建高可用Harbor \(包括恢复演练实操\)](#) 篇幅中已经提到，为了简单部署，只是讲解了：

基于阿里云的SLB四层实现了对2台ECS部署了Harbor服务的负载均衡与高可用；

使用2台ECS搭建了主从复制模式的PG13，以保障数据安全；

使用阿里云NFS做镜像数据共享；

复用已有的阿里云RDS (Redis) 的4个index做缓存与会话共享；

模拟PG13发生主节点故障后手动切换从节点的演练以及灾难性恢复；

但其实对于最核心也最重要的数据层PG13来讲并没有实现真正的高可用，只是对数据做了一层基于PG的XLGO复制，保障主节点宕机后，可以快速手动将从节点提升为主节点；就这个问题，今天我们来聊聊，如何使用开源组件keepalived结合阿里云的EIP+HAVIP让PG13可以实现HA自动切换，从而成为真正意义的高可用、负载均衡的Harbor。

就官方的高可用部署方案而言，推荐把所有服务都分布在k8s集群中，对于各个服务可用性与性能来讲，肯定是提高了，但是这样也一定会对你使用的资源有所要求与消耗，并且增加了使用与技术维护成本，同时对k8s小白也非常的不友好；另外，对于任何系统架构设计，能解耦的，绝不把鸡蛋放在一个篮子里，要相信墨菲定律~

而我们这里所讲的方案，就是简单原则，化繁为简，学以致用；在控制成本的的前提下，以最小资源环境，使用大家都熟知的技术手段设计一套**高可用，负载均衡**，并且可以将资源利用率完全发挥到极致！

准备工作：

阿里云专有网络 高可用开源组件

EIP

HAVIP

keepalived xxx.xxx.xxx.xxx

实现目标：

keepalived 是什么，这里就不过多叙述了，自行百度与谷歌吧！只对如何使用阿里云的EIP+HAVIP与keepalived来实现PG13数据库主节点故障以后，实现自动切换从节点，以达到高可用目的，使数据库服务对Harbor继续提供访问~

PS:

因为阿里云现在已经不支持宿主机ECS创建绑定VIP，但是！阿里云把这个功能集成在了专用网络里，起名为HAVIP！阿西吧~所以，我们只能在阿里云的专用网络中申请HAVIP来配合keepalived做高可用。

申请时注意事项:

1. 需要在ECS所属区域中的交换机申请HAVIP，切记不要选错，比如我的ECS在北京区域某个交换机中，那么我就需要把区域切换到北京，然后在这个交换机中申请HAVIP，交换机的id可以在ECS详情中查看。
2. 阿里云HAVIP只支持单播，所以要使用unicast_src_ip参数，且要使用unicast_peer配置对端的设备IP。
3. HaVip仅支持绑定到同一个交换机下的ECS实例，每个HaVip最多可以绑定两台ECS实例。

申请阿里云HAVIP

申请链接 <https://vpcnext.console.aliyun.com/vpc/cn-shanghai/havips>



安装、配置keepalived

在每个ECS安装keepalived 安装命令：`apt install keepalived -y`，我的是ubuntu18.04，如果是CentOS，请使用yum install安装。

配置主节点keepalived

```
# 配置 主节点 keepalived ECS253
vim /etc/keepalived/keepalived.conf
! Configuration File for keepalived

global_defs {
    notification_email {
        acassen@firewall.loc
        failover@firewall.loc
        sysadmin@firewall.loc
    }
    notification_email_from Alexandre.Cassen@firewall.loc
    smtp_server 192.168.200.1
    smtp_connect_timeout 30
    router_id LVS_DEVEL
    vrrp_skip_check_adv_addr
    vrrp_garp_interval 0
    vrrp_gna_interval 0
}
vrrp_instance VI_1 {
    state MASTER          # 设置ECS253实例为主实例

    interface eth0        # ECS网卡: eth0
    virtual_router_id 51
    nopreempt
    priority 100           # 设置优先级，数字越大，优先级越高.
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 111111
    }
    unicast_src_ip 172.19.48.253 # ECS实例的私网IP地址: 172.19.48.253
    unicast_peer {
        172.19.48.254          # 对端ECS实例的私网IP地址: 172.19.48.254
    }
    virtual_ipaddress {
        172.19.48.100          # VIP地址: 172.19.48.100
    }
}
```

```

172.19.48.188      # HA VIP的IP地址: 172.19.48.188
}
notify_master "/etc/keepalived/script/pg13_action.sh MASTER"
notify_backup "/etc/keepalived/script/pg13_action.sh BACKUP"
notify_fault "/etc/keepalived/script/pg13_action.sh FAULT"
notify_stop "/etc/keepalived/script/pg13_action.sh STOP"
garp_master_delay 1
garp_master_refresh 5

track_interface {
    eth0      # ECS网卡为eth0
}
}

```

配置备节点 keepalived

```

# 配置 备节点 keepalived ECS254
vim /etc/keepalived/keepalived.conf
! Configuration File for keepalived

global_defs {
    notification_email {
        acassen@firewall.loc
        failover@firewall.loc
        sysadmin@firewall.loc
    }
    notification_email_from Alexandre.Cassen@firewall.loc
    smtp_server 192.168.200.1
    smtp_connect_timeout 30
    router_id LVS_DEVEL
    vrrp_skip_check_adv_addr
    vrrp_garp_interval 0
    vrrp_gna_interval 0
}

vrrp_instance VI_1 {
    state BACKUP      # ECS254实例为备用实例
    interface eth0    # 网卡 : eth0
    virtual_router_id 51
    nopreempt
    priority 10       # 设置优先级, 数字越大, 优先级越高.
    advert_int 1
}

```

```
authentication {
    auth_type PASS
    auth_pass 111111
}
unicast_src_ip 172.19.48.254 # ECS实例的私网IP地址: 172.19.48.254
unicast_peer {
    172.19.48.253 # 对端ECS实例的私网IP地址: 172.19.48.253
}
virtual_ipaddress {
    172.19.48.188 # HaVip的IP地址: 172.19.48.188
}
notify_master "/etc/keepalived/script/pg13_action.sh MASTER"
notify_backup "/etc/keepalived/script/pg13_action.sh BACKUP"
notify_fault "/etc/keepalived/script/pg13_action.sh FAULT"
notify_stop "/etc/keepalived/script/pg13_action.sh STOP"
garp_master_delay 1
garp_master_refresh 5

    track_interface {
        eth0 # ECS网卡为eth0
    }
}
```

因涉及到业务私密性，脚本就不提供了，自己尝试写写吧，有问题可以加我微信咨询。

将HAVIP与主备ECS实例绑定

完成以下操作，将HaVip分别与主备ECS实例绑定。

登录专有网络管理控制台。

在左侧导航栏，单击**高可用虚拟IP**。

在顶部菜单栏处，选择HaVip的地域。

找到步骤一：创建HaVip创建的HaVip实例，单击HaVip实例的ID。

在**绑定资源**区域**ECS实例**处，单击**立即绑定**。



在弹出的对话框中，选择ECS1实例，然后单击**确定**。

重复上述步骤，将HaVip与ECS2实例绑定。

绑定成功后，在**绑定资源**区域，查看HaVip绑定的ECS实例。



将HAVIP与EIP绑定

将HaVip与EIP绑定，绑定后该HaVip可以通过EIP提供公网服务。

登录专有网络管理控制台。

在左侧导航栏，单击**高可用虚拟IP**。

在顶部菜单栏处，选择HaVip的地域。

找到步骤一：创建HaVip创建的HaVip实例，单击HaVip实例的ID。

在**绑定资源**区域，单击**弹性公网IP**右侧的**立即绑定**。



在弹出的对话框中，选择需要绑定的EIP，然后单击**确定**。

绑定成功后，您可以查看绑定的EIP。

高可用虚拟IP						
创建高可用虚拟IP		刷新	自定义			
实例ID	IP地址	状态	绑定实例	专有网络	交换机	操作
harp-1g...	192.168.1.100	已分配	harp-1g...	vpc-1g...	vpc-1g...	

故障演练测试

完成如上操作后，访问已经配置的Harbor域名，因为我们之前已经做了负载均衡，只需要模拟一下将PG13停止掉，便可触发了自动切换，此时，我们已经做好了Harbor的PG13数据库高可用！

在基于DevOps思想建设自动化运维改革的大道上，一直砥砺前行，从未停歇。

道阻且长，行则将至，行而不辍，未来可期。

欢迎搜索 **k8stech** 关注公众号 **Kubernetes技术栈**，定时更新关于运维开发、云原生、**SRE**等文章。




Kubernetes技术栈

最新的k8s技术以及周边技术文章分享





 Kubernetes技术栈