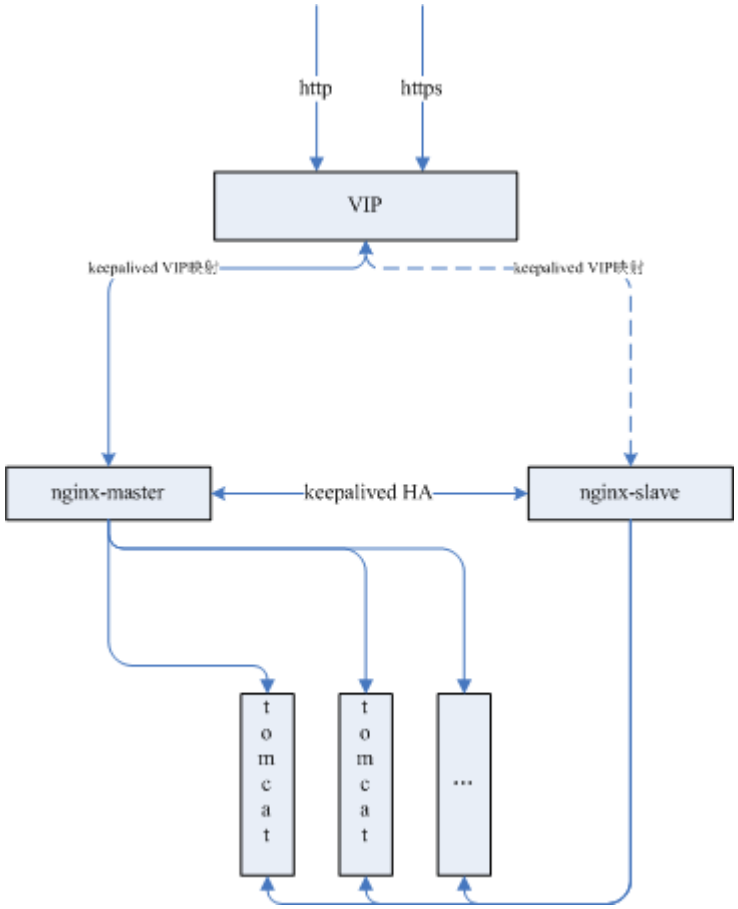
1 Nginx 负载均衡

1.1 规划

192.168.0.221：nginx + keepalived　　master
192.168.0.222：nginx + keepalived　　backup
192.168.0.223：tomcat
192.168.0.224：tomcat
虚拟 ip(VIP):192.168.0.200，对外提供服务的 ip，也可称作浮动 ip
各个组件之间的关系图如下：



1.2 启动两台或多台单体服务（223、224）

如用 java –jar xxx 或 tomcat 中的 startup.sh 启动一个网站或服务
这时：http://192.168.0.223:8080/myweb
http://192.168.0.224:8080/myweb

1.3 配置 nginx master(221)和 slave(222)

221 和 222 配置完全一样

```
#设定 http 服务器,利用它的反向代理功能提供负载均衡支持
http
{
    #添加 tomcat 列表，真实应用服务器都放在这
    upstream tomcat_pool
    {
        #server tomcat 地址:端口号  weight 表示权值，权值越大，被分配的几率越大;
         server 192.168.0.223:8080 weight=4 max_fails=2 fail_timeout=30s;
        server 192.168.0.224:8080 weight=4 max_fails=2 fail_timeout=30s;
    }
    server
    {
        listen          80;          #监听端口
        server_name    localhost;

    #默认请求设置
    location / {
        proxy_pass http://tomcat_pool;      #转向 tomcat 处理
    }
}
```

配置完成后，两个 nginx 效果一样，但仍只能各自访问：

http://192.168.0.221/myweb

http://192.168.0.222/myweb

当配置 keepalived 之后就有了主从之分了

1.4 配置 keepalived master(221)和 slave(222)

master(221)配置：

```
global_defs {
    notification_email {
        997914490@qq.com
    }
    notification_email_from sns-lvs@gmail.com
    smtp_server smtp.hysec.com
    smtp_connection_timeout 30
    router_id nginx_master          #  设置 nginx master 的 id，在一个网络应该是唯一的
}
vrrp_script chk_http_port {
    script "/usr/local/src/check_nginx_pid.sh"      #最后手动执行下此脚本，以确保此脚本能
```

```
够正常执行
    interval 2                          #（检测脚本执行的间隔，单位是秒）
    weight 2
}
vrrp_instance VI_1 {
    state MASTER            # 指定 keepalived 的角色，MASTER 为主，BACKUP 为备
    interface eth0          # 当前进行 vrrp 通讯的网络接口卡(当前 centos 的网卡)
    virtual_router_id 66    # 虚拟路由编号，主从要一直
    priority 100            # 优先级，数值越大，获取处理请求的优先级越高
    advert_int 1            # 检查间隔，默认为 1s(vrrp 组播周期秒数)
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    track_script {
    chk_http_port           #（调用检测脚本）
    }
    virtual_ipaddress {
        192.168.0.200           # 定义虚拟 ip(VIP)，可多设，每行一个
    }
}
```

slave(222) 配置：

同 221 基本一致，把上面橙色部分的改为：nginx_backup、BACKUP

1.5 nginx 检测脚本 check_nginx_pid.sh

master(221)和 slave(222)完全一致：

```
#!/bin/bash
A=`ps -C nginx --no-header |wc -l`
if [ $A -eq 0 ];then
    /usr/local/nginx/sbin/nginx              #重启 nginx
    if [ `ps -C nginx --no-header |wc -l` -eq 0 ];then    #nginx 重启失败
        exit 1
    else
        exit 0
    fi
else
    exit 0
fi
```

1.6 效果测试

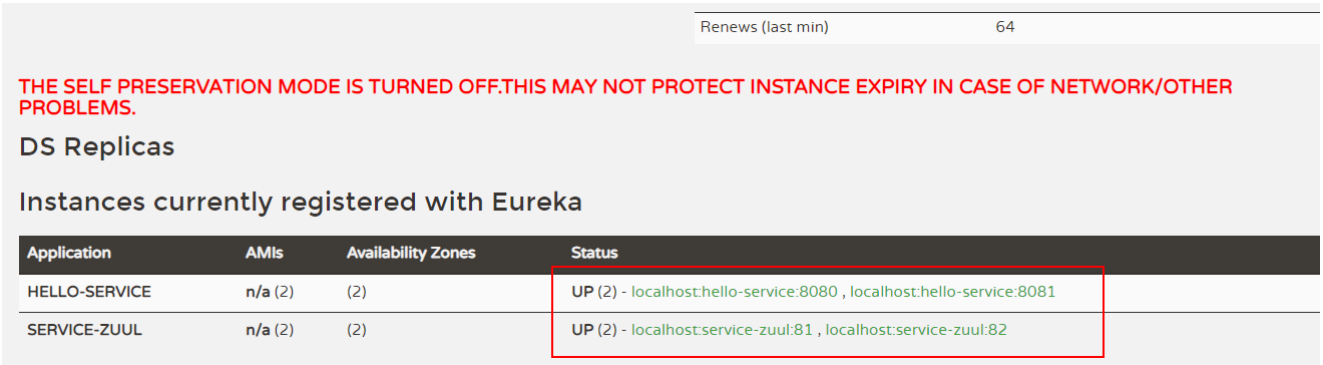
http://192.168.0.200/myweb
会交替的转向 223 和 224
把 master 上的 keepalived 停掉（模拟宕机）
页面访问不受影响

2 Nginx 实现网关 zuul 负载均衡

使用 server.port 模拟两个网关项目,端口号分别为 81,82 -------> 测试网关高可用

使用 server.port 模拟两个 hello-service 项目,端口号分别为 8080,8081 ---->测试 zuul 的路由时,自动负载均衡

查看注册中心,一共有四个服务:



Nginx 配置:

```
upstream    backServer{
            server 127.0.0.1:81;
            server 127.0.0.1:82;
      }
      server {
            listen            80;
            server_name    localhost;
            location / {
                  ### 指定上游服务器负载均衡服务器
                  proxy_pass http://backServer/;
                  index    index.html index.htm;
            }
      }
```