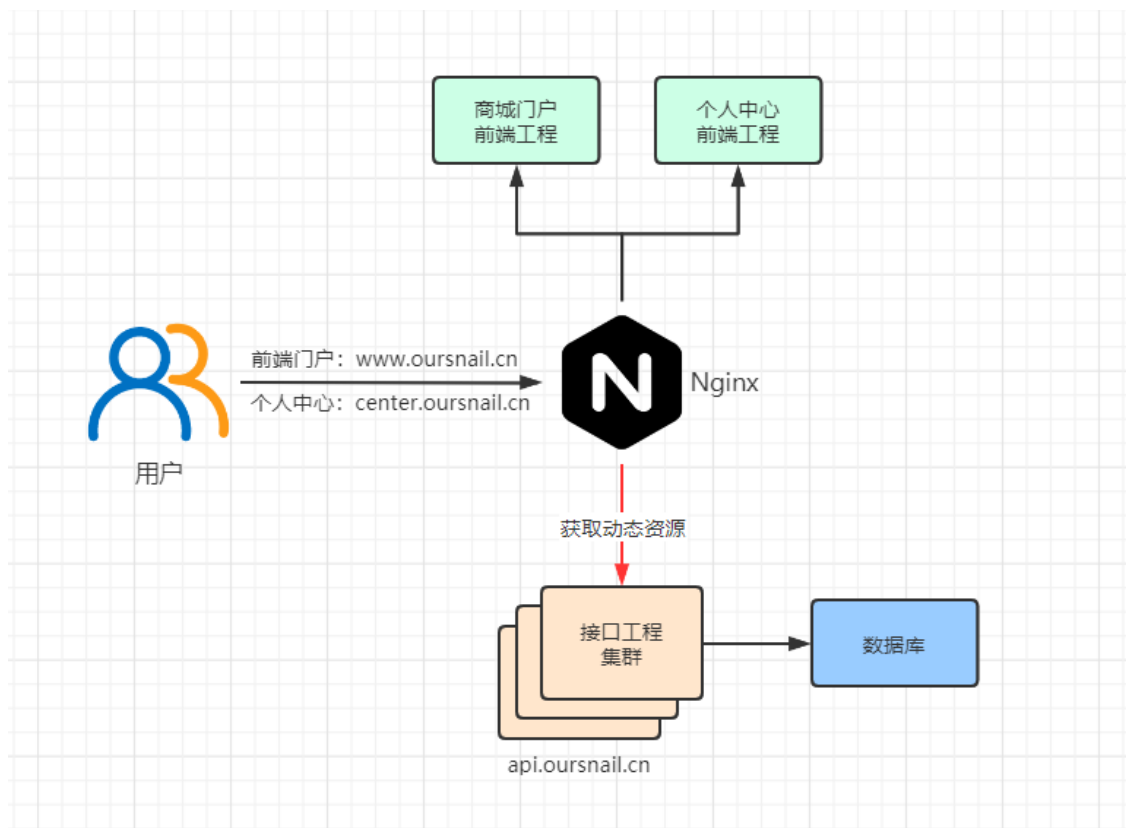
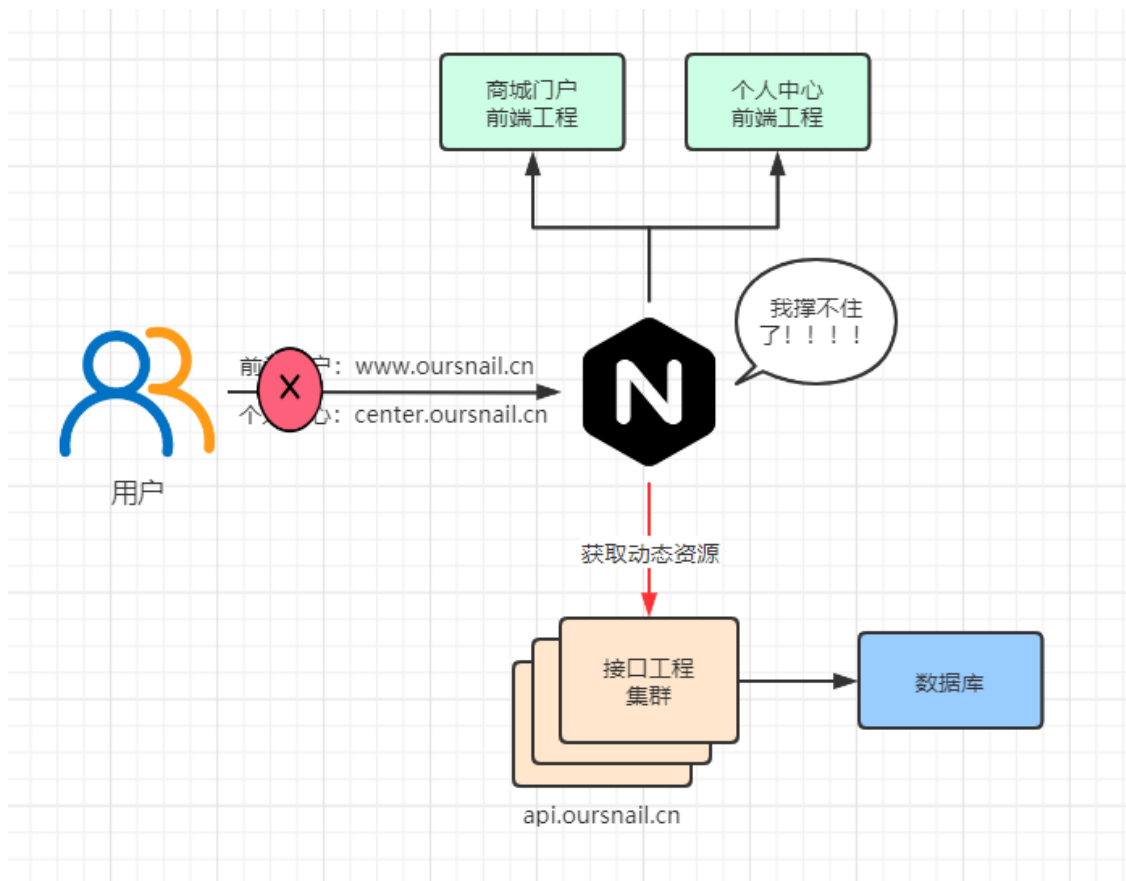


HA即High Availability，即高可用的意思，到目前为止，我们商城的整体架构如下（本节基于笔者之前自己所做的一个商城项目而写，不过核心并不是讲商城怎么做，所以不需要担心）：



我们可以通过Nginx实现动静分离，可一定程度上提高用户体验；并且可通过Nginx实现负载均衡，提高网站后端的整体性能和可靠性。

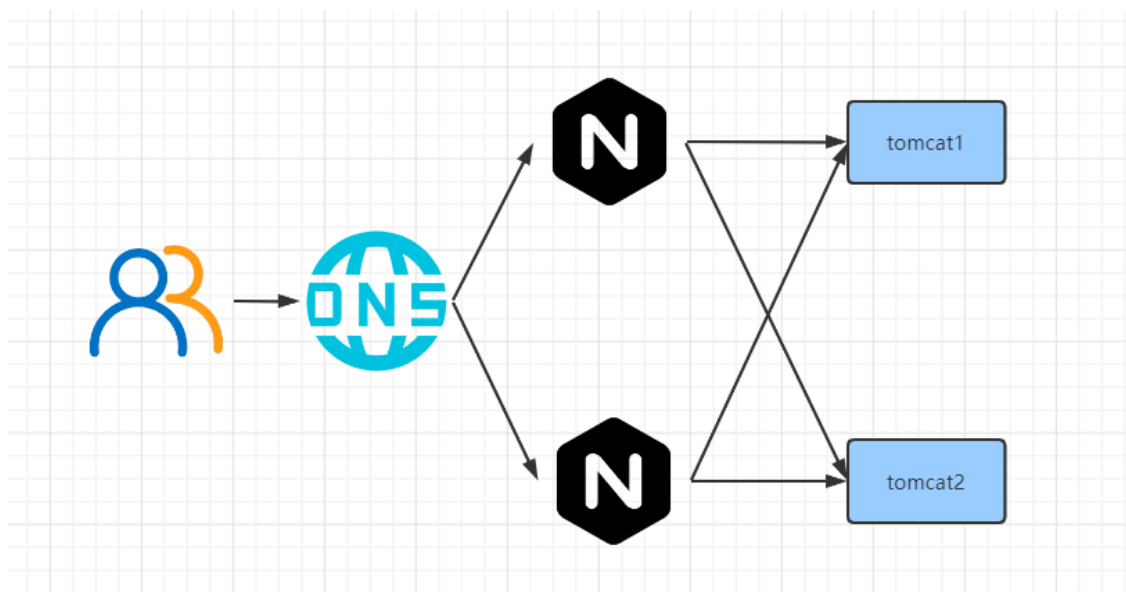
不过，我们目前使用的还是单节点Nginx，**Nginx如果挂了呢？甚至是Nginx这台主机宕机了呢？**后端的集群就成了摆设：



那么就牵涉到如何做Nginx的HA。让我们发车吧。

一、思考

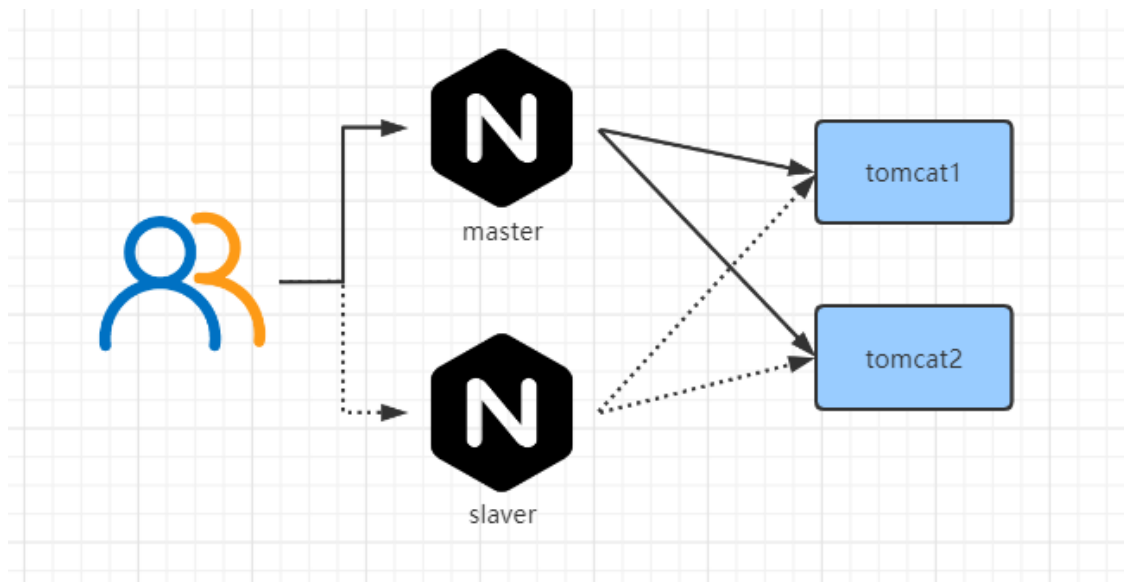
最简单的一个思路是Nginx也做成集群，做域名的DNS负载均衡，如果其中一个节点挂了，仍然会有一部分流量进到该节点，从而导致部分用户不可访问。只有进行DNS解析的修正或修复好该节点才可恢复服务，这里就需要人工介入，且存在问题发现延迟。



严格来说，这并没有做到真正的HA。我们希望，Nginx可以7*24小时是服务可用的。

二、keepalived的概念

可以考虑使用主备节点模式，主挂从上，比如：一个Nginx主节点，一个Nginx备节点，当Nginx主节点宕机后，备节点可立马提供服务。由于是主备关系，那么只有主节点宕机后备节点才会生效，也就是说同一时刻只有一个节点对外提供服务。



如何实现这个小目标呢？那就是大名鼎鼎的keepalived。

Keepalived的作用是检测服务器的状态，如果有一台web服务器宕机，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。

keepalived是以VRRP协议为实现基础的，VRRP全称Virtual Router Redundancy Protocol，即虚拟路由冗余协议。

虚拟路由冗余协议，可以认为是实现路由器高可用的协议。也就是说N台提供相同功能的路由器组成一个路由器组，这个组里面有一个master和多个backup，master上面有一个对外提供服务的vip，master不断向backup发送心跳信息，告诉backup自己还活着，当backup收不到心跳消息时就认为master已经宕机啦，这时就需要根据VRRP的优先级来选举一个backup（根据权重）当master，从而保证高可用。

三、虚拟IP

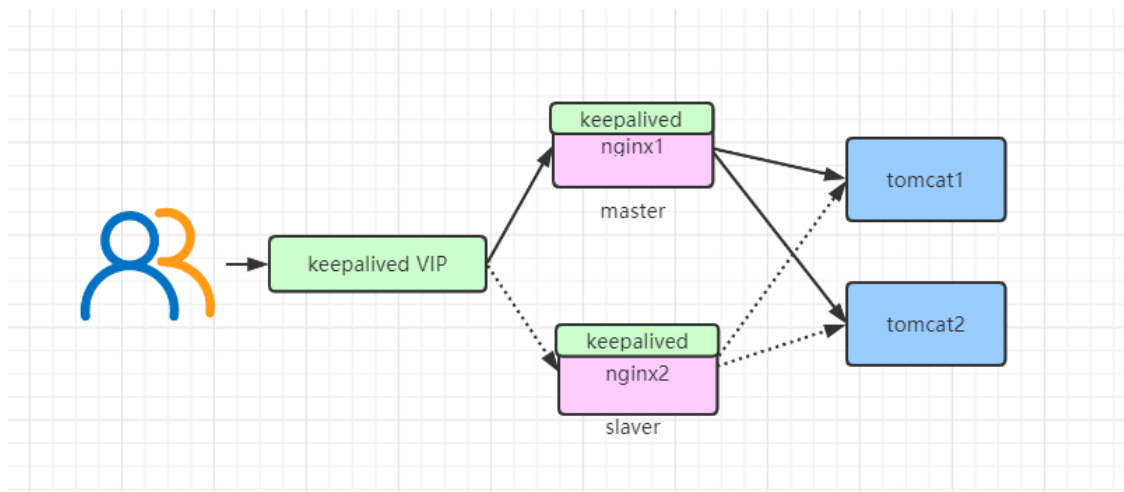
目前好像设想是很好的，但是好像有点不对劲。

按照目前的架构，用户访问域名，通过DNS的解析访问到master，如果master挂了则靠keepalived进行切换至slaver，但是有个问题，master和slaver肯定是两台主机分别部署的服务，当master挂了之后，是否要进行IP的切换呢？

肯定是不希望切换的，我们希望DNS的解析是固定的，那么这个时候就出现了VIP，VIP即Virtual IP Address，VIP用于向客户端提供一个固定的“虚拟”访问地址，以避免后端服务器发生切换时对客户端的影响。

客户端依旧访问域名，DNS只用解析至VIP即可，对于用户来说，无论是使用了master节点还是slaver节点，都看到的是固定的一个VIP。

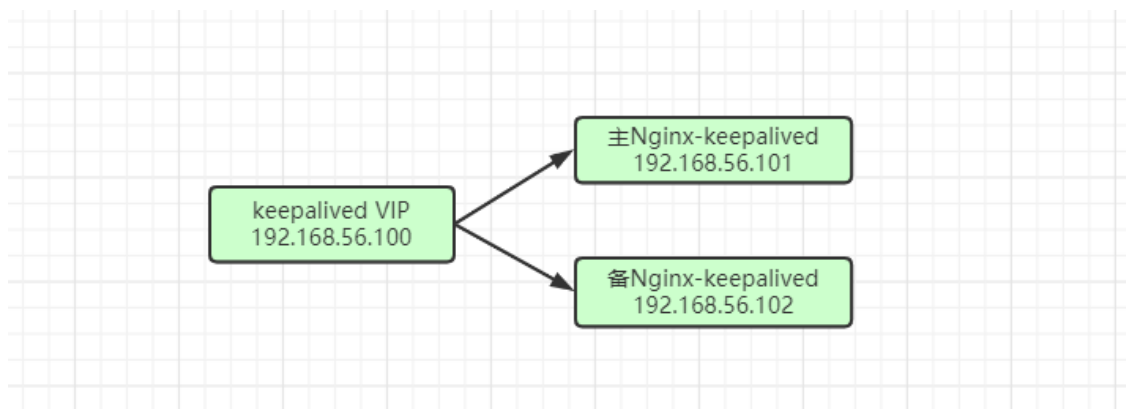
此时我们完整的架构变成了：



初始的时候，VIP被加载在Master的网卡上，所有指向VIP的请求会被发向Master，Slaver服务器处于Standby状态。如果Master出现故障，集群会通过选举算法从可用的Slaver节点中选出一个新的Master节点，并将VIP也迁移到新Master节点的网卡上。这样可以保证服务始终可用，并且对客户端来说访问的IP也不会变化。

好了，至此，我们知道VIP是统一对外的IP，VIP绑定master节点的网卡，从而使得请求都指向master节点；当master宕机时，即可通过keepalived发现，并且进行选举得出新的master节点，这个新的master节点可自动绑定VIP对外服务，也就是说keepalived实现了VIP的绑定（VIP实际上是配置在keepalived配置文件中的）

四、ip的规划



我们搭建了一套三节点的虚拟机，192.168.56.100作为VIP，（主）192.168.56.101和（备）192.168.56.102分别部署一个nginx和一个keepalived，来实现keepalived+vip的nginx HA。

五、keepalived安装

（主）192.168.56.101和（备）192.168.56.102两台主机都需要分别安装nginx和keepalived。

keepalived下载官网地址：<https://www.keepalived.org/download.html>

我下载了最新的keepalived-2.2.2.tar.gz版本，上传到服务器上，下面进行解压安装。

第一步解压：`tar -zxvf keepalived-2.2.2.tar.gz`

进入解压后的目录：`cd keepalived-2.2.2`

配置生成makefile：`./configure --prefix=/usr/local/keepalived --sysconf=/etc`

安装：`make && make install`

安装完毕后可来到这个目录下启动：`/usr/local/keepalived`

keepalived.conf配置文件位置：`/etc/keepalived`

六、keepalived的配置和启动

首先是192.168.56.101主机，我们将其作为主节点，修改配置文件/etc/keepalived/keepalived.conf：

```
! Configuration File for keepalived
```

```

global_defs {
    # 路由ID: 当前安装keepalived节点主机的标识符, 全局唯一
    router_id keep_101
}

# 基于VRRP的实例, 就是一个计算机节点
vrrp_instance VI_1 {
    # 表示当前101的nginx是主节点
    state MASTER
    # 绑定的网卡
    interface enp0s8
    # 保证主备节点一致
    virtual_router_id 51
    # 权重/优先级, 当master挂了, 会根据这个值进行选举
    priority 100
    # 主备之间心跳间隔, 默认为1秒
    advert_int 1
    # 认证授权的密钥, 防止非法节点进入路由器组
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    # 虚拟IP
    virtual_ipaddress {
        192.168.56.100
    }
}

```

备节点的配置文件为:

```

! Configuration File for keepalived

global_defs {
    router_id keep_102
}

vrrp_instance VI_1 {
    state BACKUP
    interface enp0s8
    virtual_router_id 51
    priority 80
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
}

```

```

}
virtual_ipaddress {
    192.168.56.100
}
}

```

分别到/usr/local/keepalived/sbin目录下执行./keepalived启动。

```

[root@localhost sbin]# ps -ef | grep keepalived
root      14376      1  0 08:49 ?        00:00:00 /usr/local/keepalived/sbin/keepalived -D
root      14377 14376  0 08:49 ?        00:00:00 /usr/local/keepalived/sbin/keepalived -D
root      14414 1405  0 08:56 pts/0    00:00:00 grep --color=auto keepalived
[root@localhost sbin]#
[root@localhost sbin]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:76:76:d1 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.5/24 brd 10.0.2.255 scope global noprefixroute dynamic enp0s3
        valid_lft 419sec preferred_lft 419sec
    inet6 fe80::8030:d0ae:87e5:b4ba/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:2f:c2:a7 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet 192.168.56.100/32 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::da98:2a0a:1ebc:85ed/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@localhost sbin]#

```

虚拟IP绑定了此主节点的网卡

我们看下备节点的情况，并没有绑定此虚拟IP：

```

[root@localhost keepalived]# ps -ef | grep keepalived
root      21447      1  0 08:22 ?        00:00:00 ./keepalived
root      21448 21447  0 08:22 ?        00:00:00 ./keepalived
root      21648 1842  0 08:58 pts/0    00:00:00 grep --color=auto keepalived
[root@localhost keepalived]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:38:39:87 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.6/24 brd 10.0.2.255 scope global noprefixroute dynamic enp0s3
        valid_lft 413sec preferred_lft 413sec
    inet6 fe80::2e5d:e66:7f35:4163/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:7f:24:6e brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::d1e0:8e86:8288:22fe/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@localhost keepalived]#

```

我在本地hosts中绑定域名：192.168.56.100 www.ha.com

浏览器访问 www.ha.com：

Welcome to MASTER- 192.168.56.101 nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

显然此时应该访问到主节点上，此时我模拟宕机，直接停止主节点的keepalived服务：

```
[root@localhost sbin]# kill 14376
[root@localhost sbin]# ps -ef | grep keepalived
root      14444  1405  0 09:00 pts/0    00:00:00 grep --color=auto
keepalived
```

此时访问 www.ha.com：

Welcome to BACKUP- 92.168.56.102 nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

我们看下虚拟IP现在是不是绑定在备份节点上：

```

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:76:76:d1 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.5/24 brd 10.0.2.255 scope global noprefixroute dynamic enp0s3
        valid_lft 530sec preferred_lft 530sec
    inet6 fe80::d803:d0ae:37e5:b4b2/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:76:76:d1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::d803:d0ae:37e5:b4b2/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@localhost sbin]#

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:76:76:d1 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.6/24 brd 10.0.2.255 scope global noprefixroute dynamic enp0s3
        valid_lft 363sec preferred_lft 363sec
    inet6 fe80::245d:e60:7f35:4163/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:76:76:d1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet 192.168.56.100/32 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::d1d0:9e9b:8d8b:22fe/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@localhost keepalived]#
```

七、配置keepalived为系统服务

由于想停止keepalived需要用到kill命令，考虑将keepalived注册为linux的服务，这样就可以通过systemctl来启动和关闭了。

来到keepalived的解压目录，找到此目录，我的
是：/home/fossi/keepalived-2.2.2/keepalived/etc

接下来进行文件的拷贝：

```
cp init.d/keepalived /etc/init.d/
```

```
cp sysconfig/keepalived /etc/sysconfig/
```


刷新服务：systemctl daemon-reload

下面即可通过下面的命令进行控制：

- systemctl status keepalived.service 查看当前服务状态
- systemctl enable keepalived.service 设置开机自动启动
- systemctl disable keepalived.service 取消开机自动启动
- systemctl start keepalived.service 启动
- systemctl stop keepalived.service 停止

我们来测试下启动命令：

```
root@localhost/sbin# systemctl start keepalived.service
root@localhost/sbin# systemctl status keepalived.service
● keepalived.service - LVS and VRRP High Availability Monitor
   Loaded: loaded (/usr/lib/systemd/system/keepalived.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2021-05-09 09:05:30 EDT; 4s ago
     Process: 14441 ExecStop:/bin/kill -s QUIT $MAINPID (code=exited, status=1/FAILURE)
    Main PID: 14511 (keepalived)
      CGroup: /system.slice/keepalived.service
              └─14511 /usr/local/keepalived/sbin/keepalived -D
                  └─14512 /usr/local/keepalived/sbin/keepalived -D

May 09 09:05:34 localhost.localdomain keepalived_vrrp[14512]: (VI_1) received lower priority (80) ...ng
May 09 09:05:34 localhost.localdomain keepalived_vrrp[14512]: (VI_1) Receive advertisement timeout
May 09 09:05:34 localhost.localdomain keepalived_vrrp[14512]: (VI_1) Entering MASTER STATE
May 09 09:05:34 localhost.localdomain keepalived_vrrp[14512]: (VI_1) setting VIPs.
May 09 09:05:34 localhost.localdomain keepalived_vrrp[14512]: (VI_1) Sending/queueing gratuitous A...00
May 09 09:05:34 localhost.localdomain keepalived_vrrp[14512]: Sending gratuitous ARP on emp0s8 for...00
May 09 09:05:34 localhost.localdomain keepalived_vrrp[14512]: Sending gratuitous ARP on emp0s8 for...00
May 09 09:05:34 localhost.localdomain keepalived_vrrp[14512]: Sending gratuitous ARP on emp0s8 for...00
May 09 09:05:34 localhost.localdomain keepalived_vrrp[14512]: Sending gratuitous ARP on emp0s8 for...00
May 09 09:05:34 localhost.localdomain keepalived_vrrp[14512]: Sending gratuitous ARP on emp0s8 for...00
Hint: Some lines were ellipsized, use -l to show in full.
root@localhost/sbin# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: emp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:38:39:87 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.5/24 brd 10.0.2.255 scope global noprefixroute dynamic emp0s3
        valid_lft 369sec preferred_lft 369sec
    inet6 fe80::8930:d0ae:87e5:b4ba/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: emp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:38:39:a7 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global noprefixroute emp0s8
        valid_lft forever preferred_lft forever
    inet 192.168.56.100/32 scope global emp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::8930:d0ae:87e5:b4ba/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
root@localhost/sbin#
```

再测试下停止命令：

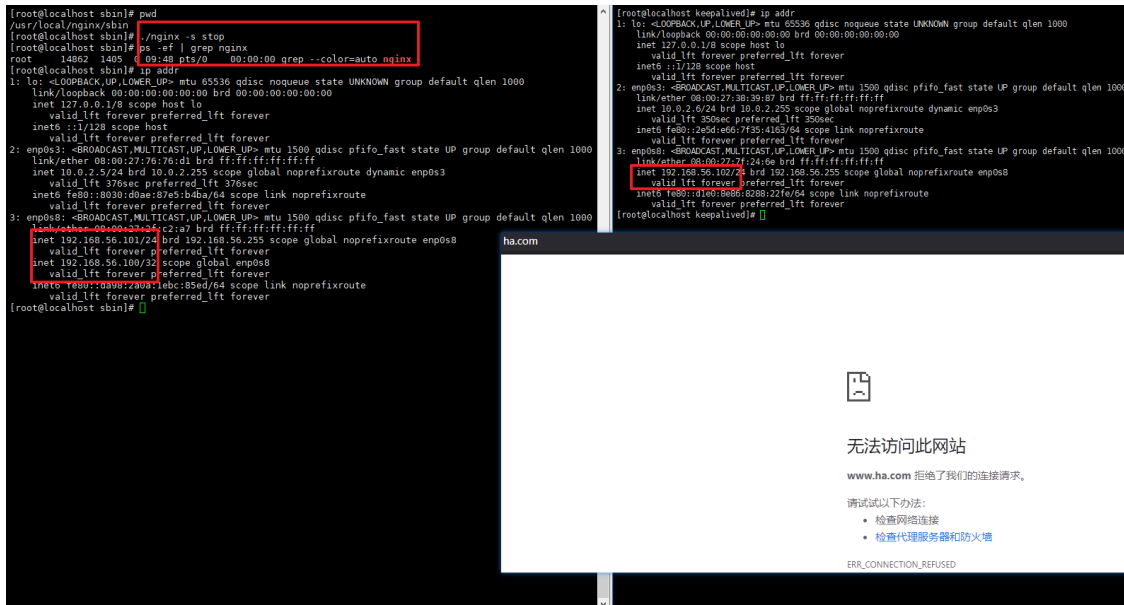
```
root@localhost/sbin# systemctl stop keepalived.service
root@localhost/sbin# systemctl status keepalived.service
● keepalived.service - LVS and VRRP High Availability Monitor
   Loaded: loaded (/usr/lib/systemd/system/keepalived.service; disabled; vendor preset: disabled)
   Active: inactive (dead)

May 09 09:05:39 localhost.localdomain keepalived_vrrp[14512]: Sending gratuitous ARP on emp0s8 for...00
May 09 09:05:39 localhost.localdomain keepalived_vrrp[14512]: Sending gratuitous ARP on emp0s8 for...00
May 09 09:05:39 localhost.localdomain keepalived_vrrp[14512]: Sending gratuitous ARP on emp0s8 for...00
May 09 09:05:39 localhost.localdomain keepalived_vrrp[14512]: Sending gratuitous ARP on emp0s8 for...00
May 09 09:05:52 localhost.localdomain keepalived_vrrp[14512]: (VI_1) sent 0 priority
May 09 09:05:52 localhost.localdomain keepalived_vrrp[14512]: (VI_1) removing VIPs.
May 09 09:05:52 localhost.localdomain keepalived[14511]: Stopping LVS and VRRP High Availability Monitor...
May 09 09:05:52 localhost.localdomain keepalived_vrrp[14512]: Stopped LVS and VRRP High Availability Monitor.
May 09 09:05:53 localhost.localdomain system[1]: Stopped LVS and VRRP High Availability Monitor.
Hint: Some lines were ellipsized, use -l to show in full.
root@localhost/sbin# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: emp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:38:39:87 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.5/24 brd 10.0.2.255 scope global noprefixroute dynamic emp0s3
        valid_lft 372sec preferred_lft 372sec
    inet6 fe80::8930:d0ae:87e5:b4ba/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: emp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:38:39:a7 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global noprefixroute emp0s8
        valid_lft forever preferred_lft forever
    inet 192.168.56.100/32 scope global emp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::8930:d0ae:87e5:b4ba/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
root@localhost/sbin# ps -ef | grep keepalived
root  14511 1405  0 09:07 pts/0    00:00:00 grep --color=auto keepalived
root@localhost/sbin#
```

八、Nginx异常退出的情况

上面说的是主机宕机的情形，那如果仅仅是nginx服务挂了呢？

如果我直接停止主节点上的nginx服务，那么就会出现无法访问的情况。



可以看到，我停止了主节点上的nginx进程，但是虚拟IP仍然是绑定在主节点上的，因此我们也无法访问到页面，我们该如何解决呢？

为保证不间断服务，应当让keepalived可定时检测nginx的运行状态，如果nginx挂了应当尝试自动启动，如果实在启动不了，则应当停止当前主节点的keepalived进程，让用户访问到备节点。

在/etc/keepalived目录下新建脚本文件check_nginx_alive_or_not.sh：

```
#!/bin/bash
A=`ps -C nginx --no-header |wc -l`
# 判断nginx是否宕机，如果宕机了，尝试重启
if [ $A -eq 0 ];then
    /usr/local/nginx/sbin/nginx
    # 等待一小会再次检查nginx，如果没有启动成功，则停止keepalived，使其启动备用机
    sleep 3
    if [ `ps -C nginx --no-header |wc -l` -eq 0 ];then
        # 尝试重启nginx失败，则停止keepalived
        killall keepalived
    fi
fi
```

这个脚本很简单，就是检测nginx进程是否还存在，不存在则尝试重启，重启失败则锁行直接关闭keepalived进程，触发主备节点的切换。

我们来手动运行下。

```

[root@localhost keepalived]# ./check_nginx_alive_or_not.sh
-bash: ./check_nginx_alive_or_not.sh: Permission denied
[root@localhost keepalived]#
[root@localhost keepalived]#
[root@localhost keepalived]# chmod +x check_nginx_alive_or_not.sh
[root@localhost keepalived]# ll
total 12
-rwxr-xr-x. 1 root root 431 May  9 09:54 check_nginx_alive_or_not.sh
-rw-r--r--. 1 root root 751 May  9 07:04 keepalived.conf
drwxr-xr-x. 2 root root 4096 May  9 06:31 samples
[root@localhost keepalived]# ps -ef | grep nginx
root      14942   1405  0 09:51 pts/0    00:00:00 grep --color=auto nginx
[root@localhost keepalived]# ./check_nginx_alive_or_not.sh
[root@localhost keepalived]# ps -ef | grep nginx
root      14948      1  0 09:51 ?        00:00:00 nginx: master process /usr/local/nginx/sbin/nginx
nobody    14950  14948  0 09:51 ?        00:00:00 nginx: worker process
root      14955   1405  0 09:51 pts/0    00:00:00 grep --color=auto nginx
[root@localhost keepalived]#

```

发现这个脚本起作用了，那么下面让keepalived定时监听。

配置keepalived监听nginx脚本：

```

vrrp_script check_nginx_alive {
    script "/etc/keepalived/check_nginx_alive_or_not.sh"
    interval 2 # 每隔两秒运行上一行脚本
    #weight 10 # 如果脚本运行失败，则调整权重，可以配置为-10即降低权重
}

```

在 vrrp_instance 中新增监控的脚本：

```

track_script {
    check_nginx_alive # 追踪 nginx 脚本
}

```

即可实现定时2秒执行脚本。不过这里直接配置到crontab也可以吧。整体配置如下：

```

! Configuration File for keepalived

global_defs {
    # 路由ID: 当前安装keepalived节点主机的标识符, 全局唯一
    router_id keep_101
}

vrrp_script check_nginx_alive {
    script "/etc/keepalived/check_nginx_alive_or_not.sh"
    interval 2 # 每隔两秒运行上一行脚本
    #weight 10 # 如果脚本运行失败, 则调整权重, 可以配置为-10即降低权重
}

# 基于VRRP的实例, 就是一个计算机节点
vrrp_instance VI_1 {
    # 表示当前101的nginx是主节点
    state MASTER
    # 绑定的网卡
    interface enp0s8
    # 保证主备节点一致
    virtual_router_id 51
    # 权重/优先级, 当master挂了, 会根据这个值进行选举
    priority 100
    # 主备之间心跳间隔, 默认为1秒
    advert_int 1
    # 认证授权的密钥, 防止非法节点进入路由器组
    authentication {
        auth_type PASS
        auth_pass 1111
    }

    track_script {
        check_nginx_alive # 追踪 nginx 脚本
    }

    # 虚拟IP
    virtual_ipaddress {
        192.168.56.100
    }
}

```

重启Keepalived使得配置文件生效: `systemctl restart keepalived`

可以将2秒定时调大, 然后终结nginx进程进行测试, 我这里测试通过, 脚本达到预期。

九、注意

备用节点的配置应当与主节点的配置相当,

试想, 如果主节点可以抗住10万用户同时请求, 但是备节点因为配置较差只能抗住1万, 那么当主节点宕机时, 备节点因为扛不住也一样会宕机, 就失去了主备的意义。