keepalived 提供 kube-apiserver 对外服务的 VIP
keepalived 是一主多备运行模式，故至少两个 LB 节点
keepalived 在运行过程中周期检查本机的 haproxy 进程状态，如果检测到 haproxy 进程异常，则触发重新选主的过程，VIP 将飘移到新选出来的主节点，从而实现 VIP 的高可用
所有组件（如 kubeclt、apiserver、controller-manager、scheduler 等） 都通过 VIP 和 haproxy 监听的 8443 端口访问 kube-apiserver 服务。

keepalived 是一主（master） 多备（backup） 运行模式，故有两种类型的配置文件。master 配置文件只有一份，backup 配置文件视节点数目而定，对于本文档而言，规划如下：
master: 172.16.103.184
backup：172.16.103.245、172.16.103.246

# master 配置文件：

```
global_defs {
   router_id  lb-master
}

vrrp_instance VI-kube-master {
   state MASTER
   priority 120
   dont_track_primary
   interface eth0
   virtual_router_id 68
   advert_int 3
   virtual_ipaddress {
       172.16.102.100
   }
}
```

VIP 所在的接口（interface ${VIP_IF}） 为 eth0 ；
使用 killall -0 haproxy 命令检查所在节点的 haproxy 进程是否正常。如果异常则将权重减少（-30） ,从而触发重新选主过程；
router_id、virtual_router_id 用于标识属于该 HA 的 keepalived 实例，如果有多套

keepalived HA，则必须各不相同；

# backup 配置文件：

```
global_defs {
   router_id  lb-backup
}

vrrp_instance VI-kube-master {
   state BACKUP
   priority 110
   dont_track_primary
   interface eth0
   virtual_router_id 68
   advert_int 3
   virtual_ipaddress {
      172.16.102.100
   }
}
```

VIP 所在的接口（interface ${VIP_IF}） 为 eth0 ；
使用 killall -0 haproxy 命令检查所在节点的 haproxy 进程是否正常。如果异常则将权重减少（-30） ,从而触发重新选主过程；
router_id、virtual_router_id 用于标识属于该 HA 的 keepalived 实例，如果有多套keepalived HA，则必须各不相同；
priority 的值必须小于 master 的值；

```
systemctl restart keepalived
systemctl status keepalived|grep Active
```

确保状态为 active (running) ，否则查看日志，确认原因：

```
journalctl -u keepalived
```

查看 VIP 所在的节点，确保可以 ping 通 VIP：