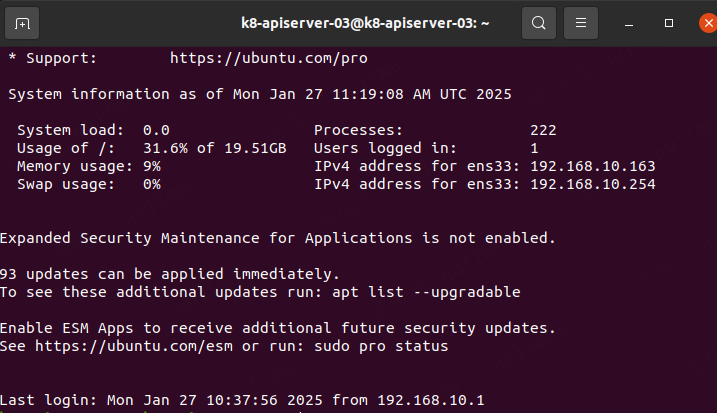
**一键部署负载均衡器haproxy/keepalived**

|  |
| --- |
| Bash #!/bin/bash ################################MASTER################################### #set variables #keppalived: export ROUTER\_ID="VRRP-01" export APISERVER\_VIP="192.168.10.254/24" export AUTH\_PASS="Kubernetes" export STATE="MASTER" export INTERFACE="ens33" export ROUTER\_ID="51" export PRIORITY="101" #haproxy: export APISERVER\_DEST\_PORT="6443" export APISERVER\_SRC\_PORT="6443" export APISERVER\_SUMS="3" export APISERVER\_IDS=("k8-master-05" "k8-master-06" "k8-master-07") export APISERVER\_ADDRESSES=("192.168.10.160" "192.168.10.161" "192.168.10.162") #apiserver: export HOST="192.168.10.163" export MASK="24" export GetWay="192.168.10.2" export DnsServers="8.8.8.8, 114.114.114.114"  if [ ! -f "/etc/netplan/50-cloud-init.yaml" ];then  sudo touch /etc/netplan/50-cloud-init.yaml fi sudo cat <<EOF> /etc/netplan/50-cloud-init.yaml  network:  version: 2  renderer: networkd  ethernets:  ens33:  dhcp4: no  addresses:   - ${HOST}/${MASK}  routes:  - to: default  via: ${GetWay}  nameservers:  addresses: [${DnsServers}] EOF sudo netplan apply  #install sudo apt update  sudo apt install -y haproxy keepalived  if [ ! -f "/etc/keepalived/keepalived.conf"]; then  sudo touch /etc/keepalived/keepalived.conf fi  sudo cat <<EOF> /etc/keepalived/keepalived.conf  ! /etc/keepalived/keepalived.conf ! Configuration File for keepalived global\_defs {  router\_id ${ROUTER\_ID}  script\_user root  enable\_script\_security } vrrp\_script check\_apiserver {  script "/etc/keepalived/check\_apiserver.sh"  interval 3  weight -2  fall 10  rise 2 }  vrrp\_instance VI\_1 {  state ${STATE}  interface ${INTERFACE}  virtual\_router\_id ${ROUTER\_ID}  priority ${PRIORITY}  authentication {  auth\_type PASS  auth\_pass ${AUTH\_PASS}  }  virtual\_ipaddress {  ${APISERVER\_VIP}  }  track\_script {  check\_apiserver  } } EOF   if [ ! -f "/etc/keepalived/check\_apiserver.sh"]; then  sudo touch /etc/keepalived/check\_apiserver.sh fi sudo cat <<EOF> /etc/keepalived/check\_apiserver.sh #!/bin/bash errorExit() {  echo "\*\*\* $\*" 1>&2  exit 1 }  curl -sfk --max-time 2 https://localhost:${APISERVER\_DEST\_PORT}/healthz -o /dev/null || errorExit "Error GET https://localhost:${APISERVER\_DEST\_PORT}/healthz" EOF sudo chmod +x /etc/keepalived/check\_apiserver.sh  if [ ! -f "/etc/haproxy/haproxy.cfg"]; then  sudo touch /etc/haproxy/haproxy.cfg fi sudo cat <<EOF> /etc/haproxy/haproxy.cfg  # /etc/haproxy/haproxy.cfg  #---------------------------------------------------------------------  # Global settings  #---------------------------------------------------------------------  global  log stdout format raw local0  daemon   #---------------------------------------------------------------------  # common defaults that all the 'listen' and 'backend' sections will  # use if not designated in their block  #---------------------------------------------------------------------  defaults  mode http  log global  option httplog  option dontlognull  option http-server-close  option forwardfor except 127.0.0.0/8  option redispatch  retries 1  timeout http-request 10s  timeout queue 20s  timeout connect 5s  timeout client 35s  timeout server 35s  timeout http-keep-alive 10s  timeout check 10s   #---------------------------------------------------------------------  # apiserver frontend which proxys to the control plane nodes  #---------------------------------------------------------------------  frontend apiserver  bind \*:${APISERVER\_DEST\_PORT}  mode tcp  option tcplog  default\_backend apiserverbackend   #---------------------------------------------------------------------  # round robin balancing for apiserver  #---------------------------------------------------------------------  backend apiserverbackend  option httpchk   http-check connect ssl  http-check send meth GET uri /healthz  http-check expect status 200   mode tcp  balance roundrobin    #add apiserver  #server HOST1\_ID HOST\_ADDRESS:APISERVER\_SRC\_PORT check verify none EOF  for ((i=0;i<${APISERVER\_SUMS};i++)) do   HOST\_ID=${APISERVER\_IDS[i]}  HOST\_ADDRESS=${APISERVER\_ADDRESSES[i]}  sudo sed -i '$ a \ server '${HOST\_ID}' '${HOST\_ADDRESS}':'${APISERVER\_SRC\_PORT}' check verify none' /etc/haproxy/haproxy.cfg  done   sudo systemctl enable haproxy --now sudo systemctl enable keepalived --now |

|  |
| --- |
| Bash #!/bin/bash ################################BACKUP################################### #set variables #keppalived: export ROUTER\_ID="VRRP-01" export APISERVER\_VIP="192.168.10.254/24" export AUTH\_PASS="Kubernetes" export STATE="MASTER" export INTERFACE="ens33" export ROUTER\_ID="50" export PRIORITY="100" #haproxy: export APISERVER\_DEST\_PORT="6443" export APISERVER\_SRC\_PORT="6443" export APISERVER\_SUMS="3" export APISERVER\_IDS=("k8-master-05" "k8-master-06" "k8-master-07") export APISERVER\_ADDRESSES=("192.168.10.160" "192.168.10.161" "192.168.10.162") #apiserver: export HOST="192.168.10.164" export MASK="24" export GetWay="192.168.10.2" export DnsServers="8.8.8.8, 114.114.114.114"  if [ ! -f "/etc/netplan/50-cloud-init.yaml" ];then  sudo touch /etc/netplan/50-cloud-init.yaml fi sudo cat <<EOF> /etc/netplan/50-cloud-init.yaml  network:  version: 2  renderer: networkd  ethernets:  ens33:  dhcp4: no  addresses:   - ${HOST}/${MASK}  routes:  - to: default  via: ${GetWay}  nameservers:  addresses: [${DnsServers}] EOF sudo netplan apply  #install sudo apt update  sudo apt install -y haproxy keepalived  if [ ! -f "/etc/keepalived/keepalived.conf"]; then  sudo touch /etc/keepalived/keepalived.conf fi  sudo cat <<EOF> /etc/keepalived/keepalived.conf  ! /etc/keepalived/keepalived.conf ! Configuration File for keepalived global\_defs {  router\_id ${ROUTER\_ID}  script\_user root  enable\_script\_security } vrrp\_script check\_apiserver {  script "/etc/keepalived/check\_apiserver.sh"  interval 3  weight -2  fall 10  rise 2 }  vrrp\_instance VI\_1 {  state ${STATE}  interface ${INTERFACE}  virtual\_router\_id ${ROUTER\_ID}  priority ${PRIORITY}  authentication {  auth\_type PASS  auth\_pass ${AUTH\_PASS}  }  virtual\_ipaddress {  ${APISERVER\_VIP}  }  track\_script {  check\_apiserver  } } EOF   if [ ! -f "/etc/keepalived/check\_apiserver.sh"]; then  sudo touch /etc/keepalived/check\_apiserver.sh fi sudo cat <<EOF> /etc/keepalived/check\_apiserver.sh #!/bin/bash errorExit() {  echo "\*\*\* $\*" 1>&2  exit 1 }  curl -sfk --max-time 2 https://localhost:${APISERVER\_DEST\_PORT}/healthz -o /dev/null || errorExit "Error GET https://localhost:${APISERVER\_DEST\_PORT}/healthz" EOF sudo chmod +x /etc/keepalived/check\_apiserver.sh   if [ ! -f "/etc/haproxy/haproxy.cfg"]; then  sudo touch /etc/haproxy/haproxy.cfg fi sudo cat <<EOF> /etc/haproxy/haproxy.cfg  # /etc/haproxy/haproxy.cfg  #---------------------------------------------------------------------  # Global settings  #---------------------------------------------------------------------  global  log stdout format raw local0  daemon   #---------------------------------------------------------------------  # common defaults that all the 'listen' and 'backend' sections will  # use if not designated in their block  #---------------------------------------------------------------------  defaults  mode http  log global  option httplog  option dontlognull  option http-server-close  option forwardfor except 127.0.0.0/8  option redispatch  retries 1  timeout http-request 10s  timeout queue 20s  timeout connect 5s  timeout client 35s  timeout server 35s  timeout http-keep-alive 10s  timeout check 10s   #---------------------------------------------------------------------  # apiserver frontend which proxys to the control plane nodes  #---------------------------------------------------------------------  frontend apiserver  bind \*:${APISERVER\_DEST\_PORT}  mode tcp  option tcplog  default\_backend apiserverbackend   #---------------------------------------------------------------------  # round robin balancing for apiserver  #---------------------------------------------------------------------  backend apiserverbackend  option httpchk   http-check connect ssl  http-check send meth GET uri /healthz  http-check expect status 200   mode tcp  balance roundrobin    #add apiserver  #server HOST1\_ID HOST\_ADDRESS:APISERVER\_SRC\_PORT check verify none EOF  for ((i=0;i<${APISERVER\_SUMS};i++)) do   HOST\_ID=${APISERVER\_IDS[i]}  HOST\_ADDRESS=${APISERVER\_ADDRESSES[i]}  sudo sed -i '$ a \ server '${HOST\_ID}' '${HOST\_ADDRESS}':'${APISERVER\_SRC\_PORT}' check verify none' /etc/haproxy/haproxy.cfg  done   sudo systemctl enable haproxy --now sudo systemctl enable keepalived --now |



|  |
| --- |
| Bash #!/bin/bash sudo systemctl disable keepalived.service sudo systemctl disable haproxy.service sudo apt remove -y keepalived haproxy sudo rm -rf /etc/keepalived sudo rm -rf /etc/haproxy |