Hướng dẫn cấu hình Keepalived thực hiện IP Failover cho hệ thống High Avaiability

By

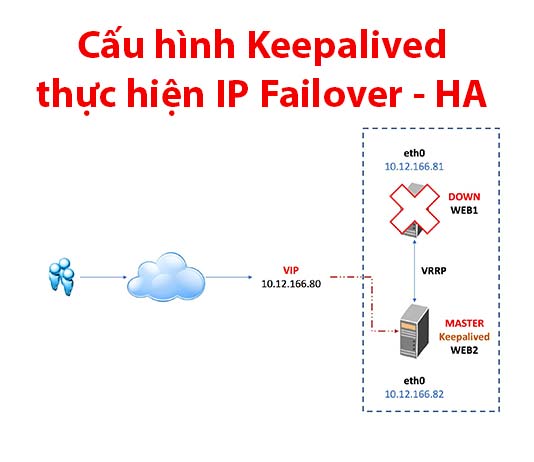
[**Quách Chí Cường**](https://cuongquach.com/author/quach-chi-cuong)

 -

19/02/2018

5428

[**Hướng dẫn cấu hình Keepalived thực hiện IP Failover cho hệ thống High Availability**](https://cuongquach.com/cau-hinh-keepalived-thuc-hien-ip-failover-he-thong-ha.html) | Chào các bạn, trong bài viết ngày hôm nay chúng ta sẽ tìm hiểu thực hành bài Lab liên quan đến mô hình dịch vụ cần ***đảm bảo tính sẵn sàng cao (High Availability – HA)***. Đối với mô hình này thì việc hệ thống bị down là điều cần ưu tiên hạn chế và thực hiện tốt nhất có thể. Hiện tại thì có khá nhiều giải pháp, ứng dụng để đảm bảo tính **HA** cho các dịch vụ trên [Linux](https://cuongquach.com/category/linux). Tuy vậy trong bài viết này chúng ta sẽ chỉ tìm hiểu về **dịch vụ Keepalived** với tính năng **Failover IP** đảm bảo tính năng tự động chuyển đổi**Virtual IP** giữa các máy chủ dịch vụ theo mô hình **Active/Passive**.



**Có thể bạn cũng quan tâm chủ đề khác**  
– [Keepalive là gì ? Tìm hiểu kĩ thuật Keepalive trong hệ thống HA](https://cuongquach.com/keepalive-la-gi-tim-hieu-ki-thuat-keepalive-trong-thong-ha.html)

**Contents**

* [Keepalived là gì ?](https://cuongquach.com/cau-hinh-keepalived-thuc-hien-ip-failover-he-thong-ha.html#Keepalived_la_gi)
* [Cơ chế VRRP trong mô hình HA Keepalived](https://cuongquach.com/cau-hinh-keepalived-thuc-hien-ip-failover-he-thong-ha.html#Co_che_VRRP_trong_mo_hinh_HA_Keepalived)
* [Mô hình Lab Keepalived IP Failover](https://cuongquach.com/cau-hinh-keepalived-thuc-hien-ip-failover-he-thong-ha.html#Mo_hinh_Lab_Keepalived_IP_Failover)
* [1. Cài đặt chương trình Keepalived](https://cuongquach.com/cau-hinh-keepalived-thuc-hien-ip-failover-he-thong-ha.html#1_Cai_dat_chuong_trinh_Keepalived)
* [2. Cài đặt chương trình Web Server](https://cuongquach.com/cau-hinh-keepalived-thuc-hien-ip-failover-he-thong-ha.html#2_Cai_dat_chuong_trinh_Web_Server)
* [3. Cấu hình dịch vụ Keepalived](https://cuongquach.com/cau-hinh-keepalived-thuc-hien-ip-failover-he-thong-ha.html#3_Cau_hinh_dich_vu_Keepalived)
  + [3.1 Cấu hình Keepalived trên WEB1 – MASTER](https://cuongquach.com/cau-hinh-keepalived-thuc-hien-ip-failover-he-thong-ha.html#31_Cau_hinh_Keepalived_tren_WEB1_8211_MASTER)
  + [3.2 Cấu hình Keepalived trên WEB2 – BACKUP](https://cuongquach.com/cau-hinh-keepalived-thuc-hien-ip-failover-he-thong-ha.html#32_Cau_hinh_Keepalived_tren_WEB2_8211_BACKUP)
* [4. Khởi động Keepalived và cấu hình iptables](https://cuongquach.com/cau-hinh-keepalived-thuc-hien-ip-failover-he-thong-ha.html#4_Khoi_dong_Keepalived_va_cau_hinh_iptables)
* [5. Kiểm tra hoạt động Keepalived](https://cuongquach.com/cau-hinh-keepalived-thuc-hien-ip-failover-he-thong-ha.html#5_Kiem_tra_hoat_dong_Keepalived)
  + [5.1 Kiểm tra trường hợp server MASTER die](https://cuongquach.com/cau-hinh-keepalived-thuc-hien-ip-failover-he-thong-ha.html#51_Kiem_tra_truong_hop_server_MASTER_die)
  + [5.2 Kiểm tra trường hợp web server Nginx MASTER die](https://cuongquach.com/cau-hinh-keepalived-thuc-hien-ip-failover-he-thong-ha.html#52_Kiem_tra_truong_hop_web_server_Nginx_MASTER_die)
* [6. Cấu hình Server không khôi phục quyền MASTER](https://cuongquach.com/cau-hinh-keepalived-thuc-hien-ip-failover-he-thong-ha.html#6_Cau_hinh_Server_khong_khoi_phuc_quyen_MASTER)
* [Tổng kết](https://cuongquach.com/cau-hinh-keepalived-thuc-hien-ip-failover-he-thong-ha.html#Tong_ket)

**Keepalived là gì ?**

Chương trình **keepalived** cho phép nhiều máy tính cùng chia sẻ một địa chỉ IP ảo với nhau theo mô hình**Active – Passive** (ta có thể cấu hình thêm một chút để chuyển thành mô hình **Active – Active**nâng cao). Khi người dùng cần truy cập vào dịch vụ, người dùng chỉ cần truy cập vào địa chỉ **IP ảo** dùng chung này thay vì phải truy cập vào những địa chỉ IP thật của các thiết bị kia.

Bạn có thể xem chi tiết tổng quan về Keepalived tại: [**Keepalived là gì ? Tìm hiểu dịch vụ Keepalived High Availability**](https://cuongquach.com/keepalived-la-gi-tim-hieu-dich-vu-keepalived-high-availability.html)

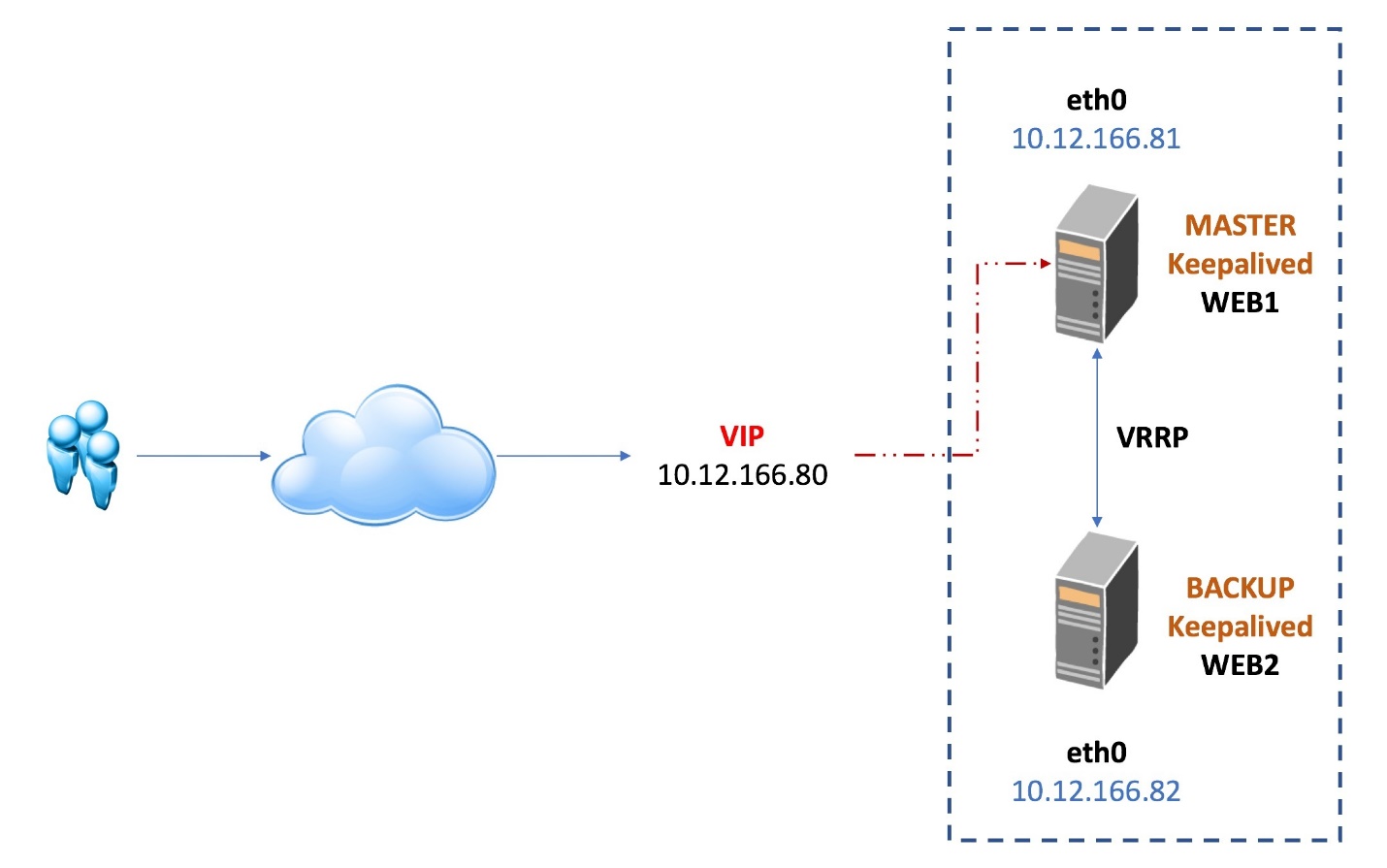
**Cơ chế VRRP trong mô hình HA Keepalived**

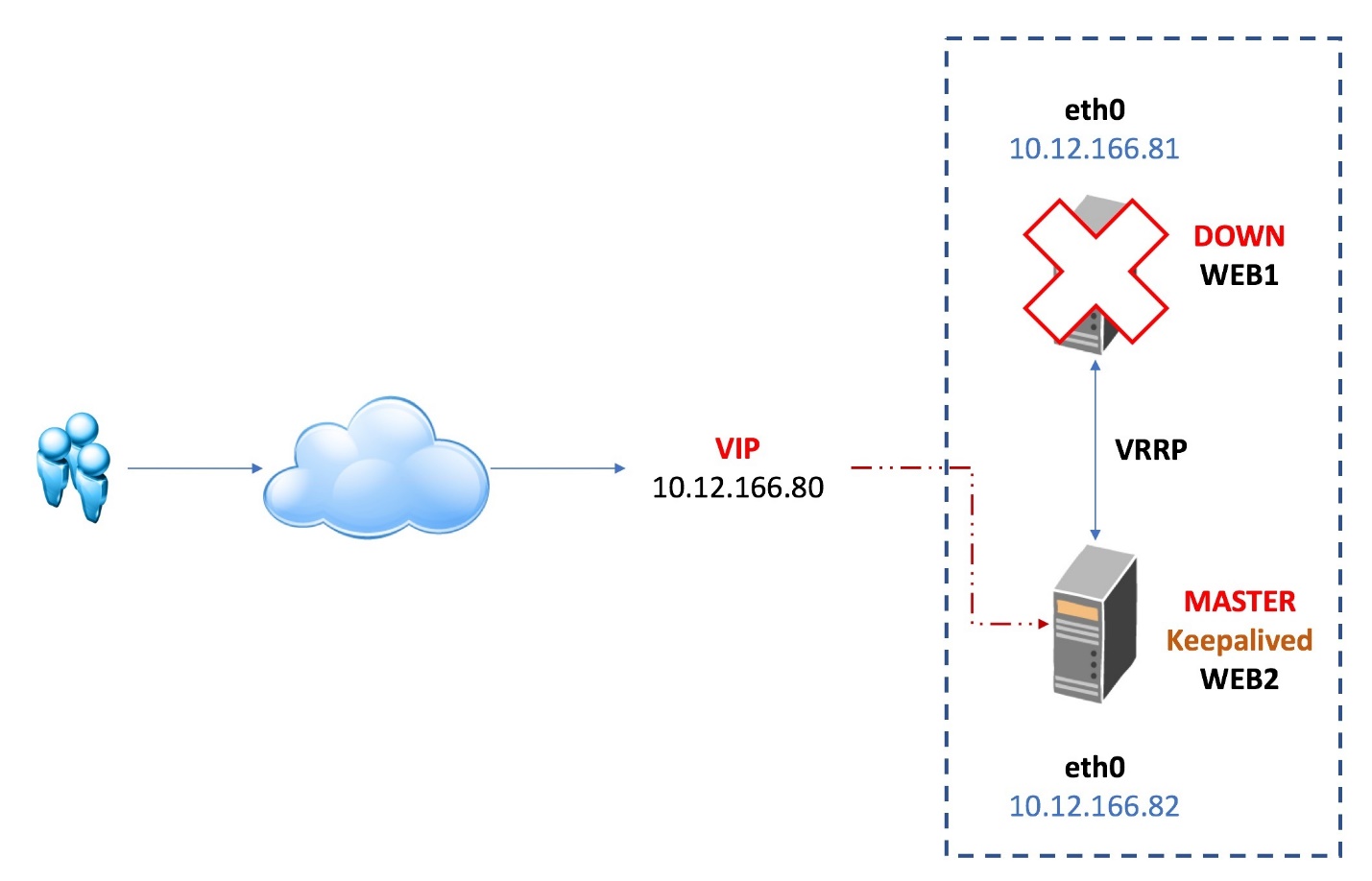
Như đã nói ở trên, các router/server vật lý dùng chung **VIP** sẽ có 2 trạng thái là **MASTER/ACTIVE** và **BACKUP/SLAVE**. Cơ chế failover được xử lý bởi giao thức VRRP, khi khởi động dịch vụ, toàn bộ các server dùng chung VIP sẽ gia nhập vào một nhóm multicast. Nhóm multicast này dùng để gởi/nhận các gói tin quảng bá VRRP. Các router/server sẽ quảng bá độ ưu tiên **(priority)** của mình, server với độ ưu tiên cao nhất sẽ được chọn làm **MASTER**. Một khi nhóm đã có 1 **MASTER** thì MASTER này sẽ chịu trách nhiệm gởi các gói tin quảng bá VRRP định kỳ cho nhóm multicast.

Nếu vì một sự cố gì đó mà các server **BACKUP** không nhận được các gói tin quảng bá từ **MASTER** trong một khoảng thời gian nhất định thì cả nhóm sẽ bầu ra một **MASTER** mới. **MASTER** mới này sẽ tiếp quản địa chỉ **VIP** của nhóm và gởi các gói tin **ARP** báo là nó đang giữ địa chỉ VIP này. Khi **MASTER** cũ hoạt động bình thường trở lại thì router này có thể lại trở thành**MASTER** hoặc trở thành **BACKUP** tùy theo cấu hình độ ưu tiên của các router.

**Mô hình Lab Keepalived IP Failover**

Trong mô hình lab của bài này, ta sẽ có **2 Nginx Web Server** (bạn có thể đổi thành HAProxy tùy ý) phục vụ xử lý request HTTP Web cơ bản. Hai Nginx **WEB1** và **WEB2** này sẽ được cấu hình dùng chung một VIP là **10.12.166.80**. Bình thường thì VIP này sẽ do node **Master** phụ trách, node **Backup** sẽ ở trạng thái chờ.

Khi có sự cố xảy ra với node **Master** như die server hay dịch vụ die thì node **Backup** sẽ nhận lấy **VIP** này và chịu trách nhiệm xử lý tiếp nội dung dịch vụ đang chạy cụ thể ở bài lab này là Nginx Web Server.



**1. Cài đặt chương trình Keepalived**

Hiện tại chương trình **Keepalived** đã có trên các repository mặc định của **CentOS** nên bạn dễ dàng cài đặt với lệnh ‘**yum**‘. Điểm trừ duy nhất là phiên bản **Keepalived** trên các Repository thường không phải bản mới nhất. Bạn cần thực hiện cài đặt chương trình Keepalived trên cả 2 con **WEB1** và **WEB2**, với các lệnh yum dưới đây sẽ cài đặt chương trình hỗ trợ và cả ‘**Keepalived**‘.



|  |  |
| --- | --- |
| 1  2  3 | # yum groupinstall -y "Development Tools"  # yum install -y gcc kernel-headers kernel-devel curl gcc openssl-devel libnl3-devel net-snmp-devel psmisc ipset-libs  # yum install -y keepalived |

**2. Cài đặt chương trình Web Server**

Chúng ta sẽ thực hiện cài đặt chương trình **Web Server Nginx**([Hướng dẫn cài đặt Nginx trên CentOS](https://cuongquach.com/huong-dan-cai-dat-nginx-tren-centos.html)) trên cả 2 server để test chức năng cơ bản của máy chủ web trên cả 2 Web Server sẽ được cấu hình. Gói chương trình “**nginx**” nằm ở EPEL Repository (**Đọc thêm**: [Hướng dẫn cài đặt EPEL Repository trên CentOS](https://cuongquach.com/cai-dat-epel-repository-tren-centos-rhel-5-6-7.html)).



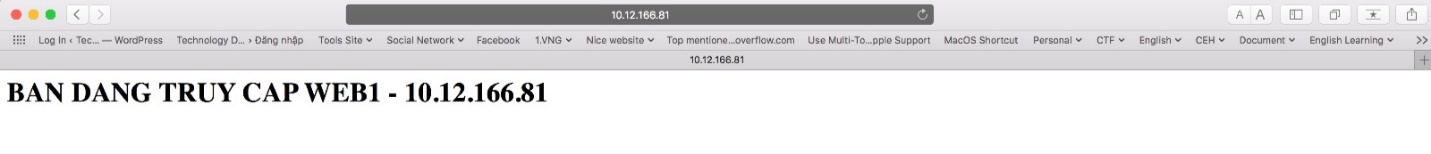
|  |  |
| --- | --- |
| 1  2  3 | # yum install -y epel-release  # yum install -y nginx  # systemctl start nginx |

Do nhu cầu chỉ là thực hiện lab test chức năng**IP** **Failover** với web server cơ bản, nên chúng ta không cần quan tâm đến việc tối ưu hoá cấu hình Nginx.

**+ WEB1**  
Đối với máy chủ WEB1 chúng ta sẽ tạo 1 file **.html** ở thư mục web global của Nginx để truy cập test với nội dung cơ bản thể hiện được thông tin cho bạn biết là bạn đang truy cập máy chủ **Web Server WEB1**.



|  |  |
| --- | --- |
| 1  2  3 | [root@web1]# rm -rf /usr/share/nginx/html/\*  [root@web1]# vi /usr/share/nginx/html/index.html  <h1> BAN DANG TRUY CAP WEB1 - 10.12.166.81 </h1> |

[](https://cuongquach.com/resources/images/2018/02/keepalived-web1-access.jpg)

**+ WEB2**



|  |  |
| --- | --- |
| 1  2  3 | [root@web2]# rm -rf /usr/share/nginx/html/\*  [root@web2]# vi /usr/share/nginx/html/index.html  <h1> BAN DANG TRUY CAP WEB2 - 10.12.166.82 </h1> |



**3. Cấu hình dịch vụ Keepalived**

Cấu hình cho phép gắn địa chỉ IP ảo lên card mạng và IP Forward.



|  |  |
| --- | --- |
| 1  2  3 | # echo "net.ipv4.ip\_nonlocal\_bind = 1" >> /etc/sysctl.conf  # echo "net.ipv4.ip\_forward = 1" >> /etc/sysctl.conf  # sysctl -p |

**Cú pháp cấu hình Keepalived cơ bản**



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36 | global\_defs {     notification\_email {          admin@cuongquach.com     }     notification\_email\_from keepalived-web1@cuongquach.com     smtp\_server x.x.x.x     smtp\_connect\_timeout 30    }    vrrp\_script chk\_haproxy {          script "command"          interval <time>          weight <n>  }    vrrp\_instance string {      state MASTER|BACKUP      interface string      mcast\_src\_ip @IP      virtual\_router\_id num      priority num      advert\_int num      smtp\_alert      authentication {          auth\_type PASS|AH          auth\_pass string      }      virtual\_ipaddress { # Block limited to 20 IP addresses          @IP          @IP      }      notify\_master "/path\_to\_script/script\_fault.sh <arg\_list>"      notify\_backup "/path\_to\_script/script\_fault.sh <arg\_list>"      notify\_fault "/path\_to\_script/script\_fault.sh <arg\_list>"  } |

Để cấu hình dịch vụ keepalived, ta cần phải chỉnh sửa file **/etc/keepalived/keepalived.conf**. Một số block cấu hình đáng chú ý trong file này như sau:

* **global\_defs**: cấu hình thông tin toàn cục (global) cho keepalived như gởi email thông báo tới đâu, tên của cluster đang cấu hình.
* **vrrp\_script**: chứa script, lệnh thực thi hoặc đường dẫn tới script kiểm tra dịch vụ (Ví dụ: nếu dịch vụ này down thì keepalived sẽ tự chuyển VIP sang 1 server khác).
* **vrrp\_instance**: thông tin chi tiết về 1 server vật lý trong nhóm dùng chung VRRP. Gồm các thông tin như interface dùng để liên lạc của server này, độ ưu tiên để, virtual IP tương ứng với interface, cách thức chứng thực, script kiểm tra dịch vụ….

**Chú thích cấu hình block vrrp\_instance**  
– Trong các phần giải thích dưới, router sẽ đồng nghĩa với máy chủ dịch vụ .

* **state (MASTER|BACKUP)**: chỉ trạng thái **MASTER** hoặc **BACKUP** được sử dụng bởi máy chủ. Nếu là MASTER thì máy chủ này có nhiệm vụ nhận và xử lý các gói tin từ host đi lên. Nếu con MASTER tèo, những con BACKUP này sẽ dựa vào 1 cơ chế bầu chọn và nhảy lên làm Master.
* **interface**: chỉ định cổng mạng nào sẽ sử dụng cho hoạt động IP Failover – VRRP
* **mcast\_src\_ip**: địa chỉ IP thực của card mạng Interface của máy chủ tham gia vào VRRP. Các gói tin trao đổi giữa các VRRP Router sử dụng địa chỉ thực này.
* **virtual\_router\_id**: định danh cho các router (ở đây là máy chủ dịch vụ) thuộc cùng 1 nhóm VRRP. Hiểu nôm na là 1 router có thể tham gia nhiều nhóm VRRP (các nhóm hoạt động động lập nhau), và **VRRP-ID** là tên gọi của từng nhóm.
* **priority**: chỉ định độ ưu tiên của VRRP router (tức độ ưu tiên máy chủ dịch vụ trong quá trình bầu chọn MASTER). Các VRRP Router trong cùng một VRRP Group tiến hành bầu chọn Master sử dụng giá trị priority đã cấu hình cho máy chủ đó. Priority có giá trị từ 0 đến 255. Nguyên tắc có bản: Priority cao nhất thì nó là Master, nếu priority bằng nhau thì IP cao hơn là Master.
* **advert\_int**: thời gian giữa các lần gởi gói tin VRRP advertisement (đơn vị giây).
* **smtp\_alert**: kích hoạt thông báo bằng email SMTP khi trạng thái MASTER có sự thay đổi.
* **authentication**: chỉ định hình thức chứng thực trong VRRP. ‘*auth\_type*‘, sử dụng hình thức mật khẩu plaintext hay mã hoá AH. ‘*auth\_pass*‘, chuỗi mật khẩu chỉ chấp nhận 8 kí tự.
* **virtual\_ipaddress**: Địa chỉ IP ảo của nhóm VRRP đó (Chính là địa chỉ dùng làm gateway cho các host). Các gói tin trao đổi, làm việc với host đều sử dụng địa chỉ ảo này.
* **notify\_master**: chỉ định chạy shell script nếu có sự kiện thay đổi về trạng thái MASTER.
* **notify\_backup**: chỉ định chạy shell script nếu có sự kiện thay đổi về trạng thái BACKUP.
* **notify\_fault**: chỉ định chạy shell script nếu có sự kiện thay đổi về trạng thái thất bại (fault).

**3.1 Cấu hình Keepalived trên WEB1 – MASTER**

Giờ chúng ta sẽ cấu hình cho máy chủ **WEB1** làm máy chủ **MASTER** trong mô hình Failover IP giữa 2 WEB Server. Tức máy chủ WEB1 sẽ là người xử lý các request HTTP đến thông qua Virtual IP ‘**10.12.166.80**‘ và khi mà máy chủ WEB1 chết thì máy chủ WEB2 sẽ tự động lên làm **MASTER** và đảm nhận xử lý **IP Virtual**. Ta sẽ cho độ ưu tiên ‘**priority**‘ là 100 và state là ‘**MASTER**‘. Có **Virtual\_router\_id** giống nhau ví dụ là 50 .



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | [root@web1]# vi /etc/keepalived/keepalived.conf    vrrp\_script chk\_nginx {          script "killall -0 nginx"          interval 2          weight 4  }    vrrp\_instance VI\_1 {      state MASTER      interface eth0      mcast\_src\_ip 10.12.166.81      virtual\_router\_id 50      priority 100      advert\_int 1      authentication {          auth\_type AH          auth\_pass cuongquach      }      virtual\_ipaddress {          10.12.166.80      }      track\_script      {          chk\_nginx      }  } |

**3.2 Cấu hình Keepalived trên WEB2 – BACKUP**

Ta sẽ cho độ ưu tiên ‘**priority**‘ là 98 và state là ‘**BACKUP**‘. Có **Virtual\_router\_id** giống nhau ví dụ là 50 .



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | [root@web2]# vi /etc/keepalived/keepalived.conf    vrrp\_script chk\_nginx {          script "killall -0 nginx"          interval 2          weight 4  }    vrrp\_instance VI\_1 {      state BACKUP      interface eth0      mcast\_src\_ip 10.12.166.82      virtual\_router\_id 50      priority 98      advert\_int 1      authentication {          auth\_type AH          auth\_pass cuongquach      }      virtual\_ipaddress {          10.12.166.80      }      track\_script      {          chk\_nginx      }  } |

**4. Khởi động Keepalived và cấu hình iptables**

Bạn cần khởi động Keepalived trên cả 2 máy chủ dịch vụ WEB.



|  |  |
| --- | --- |
| 1  2 | # systemctl start keepalived  # systemctl enable keepalived |

Cấu hình firewall rule cho phép **VRRP** giao tiếp sử dụng dãi địa chỉ multicast **224.0.0.0/8** qua giao thức **VRRP (112)** trên card mạng mà chúng ta cấu hình **Keepalived** hoạt động. Nếu bạn sử dụng**AH (51)** khi cấu hình mật khẩu chứng thực 2 bên thì thêm rule cho giao thức AH nữa.



|  |  |
| --- | --- |
| 1  2  3  4  5 | # iptables -I INPUT -i eth0 -d 224.0.0.0/8 -p vrrp -j ACCEPT  # iptables -I INPUT -i eth0 -d 224.0.0.0/8 -p ah -j ACCEPT  # iptables -I OUTPUT -o eth0 -s 224.0.0.0/8 -p vrrp -j ACCEPT  # iptables -I OUTPUT -o eth0 -s 224.0.0.0/8 -p ah -j ACCEPT.  # service iptables save |

**5. Kiểm tra hoạt động Keepalived**

Chúng ta sẽ kiểm tra xem trên server **WEB1** đảm nhận vai trò **MASTER** đã nhận được IP ảo ‘**10.12.166.80**‘ hay chưa nhé. Trong trường hợp server **MASTER** down, thì địa chỉ ảo VIP **10.12.166.80** sẽ được tự động gán cho máy chủ WEB2 BACKUP lên làm **MASTER**.



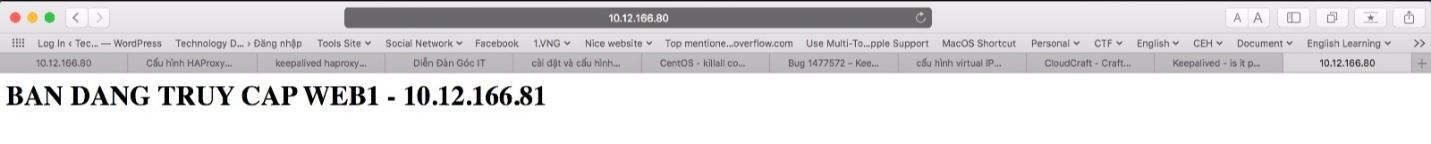
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | [root@web1]# ip a sh eth0  2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1400 qdisc pfifo\_fast state UP qlen 1000      link/ether 00:0c:29:92:26:7b brd ff:ff:ff:ff:ff:ff      inet 10.12.166.81/24 brd 10.12.166.255 scope global eth0         valid\_lft forever preferred\_lft forever      inet 10.12.166.80/32 scope global eth0         valid\_lft forever preferred\_lft forever      inet6 fe80::20c:29ff:fe92:267b/64 scope link         valid\_lft forever preferred\_lft forever |

Kiểm tra log hoạt động của dịch vụ Keepalived.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | [root@web1]# tail -n100 /var/log/messages  Feb 19 15:35:48 localhost Keepalived\_vrrp[15089]: VRRP\_Instance(VI\_1) removing protocol VIPs.  Feb 19 15:35:48 localhost Keepalived\_vrrp[15089]: Using LinkWatch kernel netlink reflector...  Feb 19 15:35:48 localhost Keepalived\_vrrp[15089]: VRRP sockpool: [ifindex(2), proto(112), unicast(0), fd(10,11)]  Feb 19 15:35:49 localhost Keepalived\_vrrp[15089]: VRRP\_Instance(VI\_1) Transition to MASTER STATE  Feb 19 15:35:50 localhost Keepalived\_vrrp[15089]: VRRP\_Instance(VI\_1) Entering MASTER STATE  Feb 19 15:35:50 localhost Keepalived\_vrrp[15089]: VRRP\_Instance(VI\_1) setting protocol VIPs.  Feb 19 15:35:50 localhost Keepalived\_vrrp[15089]: Sending gratuitous ARP on eth0 for 10.12.166.80  Feb 19 15:35:50 localhost Keepalived\_vrrp[15089]: VRRP\_Instance(VI\_1) Sending/queueing gratuitous ARPs on eth0 for 10.12.166.80  Feb 19 15:35:50 localhost Keepalived\_vrrp[15089]: Sending gratuitous ARP on eth0 for 10.12.166.80  Feb 19 15:35:50 localhost Keepalived\_vrrp[15089]: Sending gratuitous ARP on eth0 for |

Truy cập địa chỉ Virtual IP **10.12.166.80** thì bạn sẽ thấy index html **Nginx Web Server MASTER** hiện tại là 10.12.166.81.



**5.1 Kiểm tra trường hợp server MASTER die**

Trường hợp **WEB1 Nginx down nguyên server** (*bạn thử off server WEB1 để test*), thì WEB2 sẽ kiểm tra **VRRP** thấy thất bại sẽ tự bầu chọn **WEB2** lên làm **MASTER** máy chủ. Kiểm tra địa chỉ IP bên server WEB2.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | [root@web2]# ip a sh eth0  2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1400 qdisc pfifo\_fast state UP qlen 1000  link/ether 00:0c:29:49:08:77 brd ff:ff:ff:ff:ff:ff  inet 10.12.166.82/24 brd 10.12.166.255 scope global eth0  valid\_lft forever preferred\_lft forever  inet 10.12.166.80/32 scope global eth0  valid\_lft forever preferred\_lft forever  inet6 fe80::20c:29ff:fe49:877/64 scope link  valid\_lft forever preferred\_lft forever |

Truy cập địa chỉ **10.12.166.80** sẽ ra html nội dung**WEB2**.



**5.2 Kiểm tra trường hợp web server Nginx MASTER die**

Trường hợp dịch vụ**Nginx** chết trên server **MASTER** (server không chết) thì ở phần cấu hình Keepalived có cấu hình “**vrrp\_script – chk\_nginx**” dùng để kiểm tra Process ID của dịch vụ Nginx. Nếu dịch vụ Nginx chết thì Keepalived sẽ trừ trọng số priority đi 4 (theo cấu hình) lúc này *104 – 4 = 100* , độ ưu tiên của con **MASTER WEB1** thấp hơn độ ưu tiên của WEB2 (*100 + 2 = 102*) nên VRRP tiến hành bầu chọn lại **MASTER**. Server **WEB2** sẽ lên làm **MASTER** và *giữ VIP*, còn server WEB1 sẽ chuyển sang trạng thái **BACKUP**.

Chúng ta coi log trong cùng thời điểm trên 2 WEB Server khi dịch vụ **Nginx** trên Master dừng hoạt động.

**+ WEB1**



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | [root@web1]# systemctl stop nginx  [root@web1]# tail -n 10 /var/log/messages  Feb 19 22:59:48 web1 Keepalived\_vrrp[26772]: VRRP\_Script(chk\_nginx) failed  Feb 19 22:59:48 web1 Keepalived\_vrrp[26772]: VRRP\_Instance(VI\_1) Changing effective priority from 104 to 100  Feb 19 22:59:49 web1 Keepalived\_vrrp[26772]: VRRP\_Instance(VI\_1) Received advert with higher priority 102, ours 100  Feb 19 22:59:49 web1 Keepalived\_vrrp[26772]: VRRP\_Instance(VI\_1) IPSEC-AH : Syncing seq\_num - Decrement seq  Feb 19 22:59:49 web1 Keepalived\_vrrp[26772]: VRRP\_Instance(VI\_1) Entering BACKUP STATE  Feb 19 22:59:49 web1 Keepalived\_vrrp[26772]: VRRP\_Instance(VI\_1) removing protocol VIPs. |

**+ WEB2**



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | [root@web2]# tail -n 10 /var/log/messages  Feb 19 22:59:50 web2 Keepalived\_vrrp[8425]: VRRP\_Instance(VI\_1) forcing a new MASTER election  Feb 19 22:59:51 web2 Keepalived\_vrrp[8425]: VRRP\_Instance(VI\_1) Transition to MASTER STATE  Feb 19 22:59:52 web2 Keepalived\_vrrp[8425]: VRRP\_Instance(VI\_1) Entering MASTER STATE  Feb 19 22:59:52 web2 Keepalived\_vrrp[8425]: VRRP\_Instance(VI\_1) setting protocol VIPs.  Feb 19 22:59:52 web2 Keepalived\_vrrp[8425]: Sending gratuitous ARP on eth0 for 10.12.166.80  Feb 19 22:59:52 web2 Keepalived\_vrrp[8425]: VRRP\_Instance(VI\_1) Sending/queueing gratuitous ARPs on eth0 for 10.12.166.80 |

Sau đó nếu dịch vụ Nginx trên **WEB1** hoạt động trở lại, thì trọng số priority trên WEB1 sẽ khôi phục lại như lúc ban đầu. Lúc này priority **WEB1 (104)** sẽ lớn hơn **(102)** của **WEB2**, nên hoạt động bầu chọn **MASTER** lại diễn ra. **WEB1** sẽ khôi phục lại quyền làm **MASTER** và nắm giữa VIP. WEB2 sẽ quay về làm BACKUP.

**6. Cấu hình Server không khôi phục quyền MASTER**

Trong một số trường hợp, bạn không muốn server **WEB1** vừa chết xong quay lại chiếm quyền làm **MASTER** trong giữa 2 máy chủ **Keepalived**.

Bạn sẽ lưu ý một số cấu hình sau, có thể tự mình trải nghiệm cấu hình :

* **STATE** của các server đều là **BACKUP.**
* Cấu hình thêm ‘**nopreempt**‘.
* Các priority chênh nhau một ít.

**+ Cấu hình Keepalved trên Web1**



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | vrrp\_script chk\_nginx {          script "killall -0 nginx"          interval 2          weight 4  }    vrrp\_instance VI\_1 {      state BACKUP      interface eth0      mcast\_src\_ip 10.12.166.81      virtual\_router\_id 50      priority 51      advert\_int 1      nopreempt      authentication {          auth\_type AH          auth\_pass cuongquach      }      virtual\_ipaddress {          10.12.166.90      }      track\_script      {          chk\_nginx      }  } |

**+ Cấu hình Keepalived trên Web2**



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | vrrp\_script chk\_nginx {          script "killall -0 nginx"          interval 2          weight 4  }    vrrp\_instance VI\_1 {      state BACKUP      interface eth0      mcast\_src\_ip 10.12.166.82      virtual\_router\_id 50      priority 50      advert\_int 1      nopreempt      authentication {          auth\_type AH          auth\_pass cuongquach      }      virtual\_ipaddress {          10.12.166.90      }      track\_script      {          chk\_nginx      }  } |

**Tổng kết**

Vậy là bạn đã biết cách để thực hiện [**cấu hình Keepalived cho hoạt động IP Failover với Virtual IP**](https://cuongquach.com/cau-hinh-keepalived-thuc-hien-ip-failover-he-thong-ha.html) nhằm đảm bảo tính sẵn sàng cao High Availability cho các dịch vụ trên Linux rồi phải không nào ! Bạn có thể áp dụng bài lab này với các dịch vụ khác như FTP, HAProxy, Squid,…

High Availability - Hướng dẫn triển khai Haproxy Keepalive cho Apache trên CentOS 7

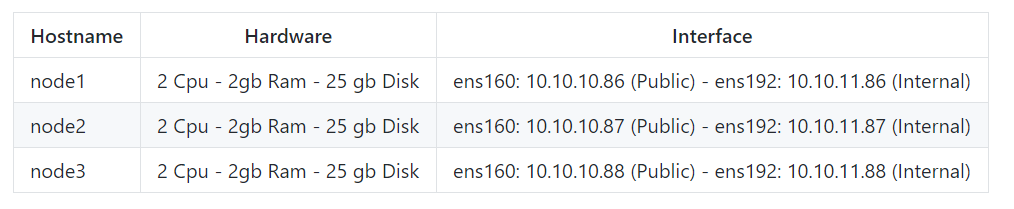
Tổng quan[#](https://blog.cloud365.vn/linux/haproxy-keepalived-apache/#tong-quan)

**HAProxy** viết tắt của High Availability Proxy, là công cụ mã nguồn mở nổi tiếng ứng dụng cho giải pháp cân bằng tải TCP/HTTP cũng như giải pháp máy chủ Proxy (Proxy Server). HAProxy có thể chạy trên các mỗi trường Linux, Solaris, FreeBSD. Công dụng phổ biến nhất của HAProxy là cải thiện hiệu năng, tăng độ tin cậy của hệ thống máy chủ bằng cách phân phối khối lượng công việc trên nhiều máy chủ (như Web, App, cơ sở dữ liệu). HAProxy hiện đã và đang được sử dụng bởi nhiều website lớn như GoDaddy, GitHub, Bitbucket, Stack Overflow, Reddit, Speedtest.net, Twitter và trong nhiều sản phẩm cung cấp bởi Amazon Web Service.

Dịch vụ **keepalived** sử dụng với mục đích tạo ra virtual ip address (IP VIP) cho hệ thống. Tiến trình **keepalived** có thể tự động giám sát dịch vụ hoặc hệ thống và có khả năng chịu lỗi cho hệ thống khi dịch vụ hoặc hệ điều hành xảy ra vấn đề. Trong bài hướng dẫn, tôi sẽ sử dụng **keepalived** để tăng tính sẵn sàng cho dịch vụ cân bằng tải.

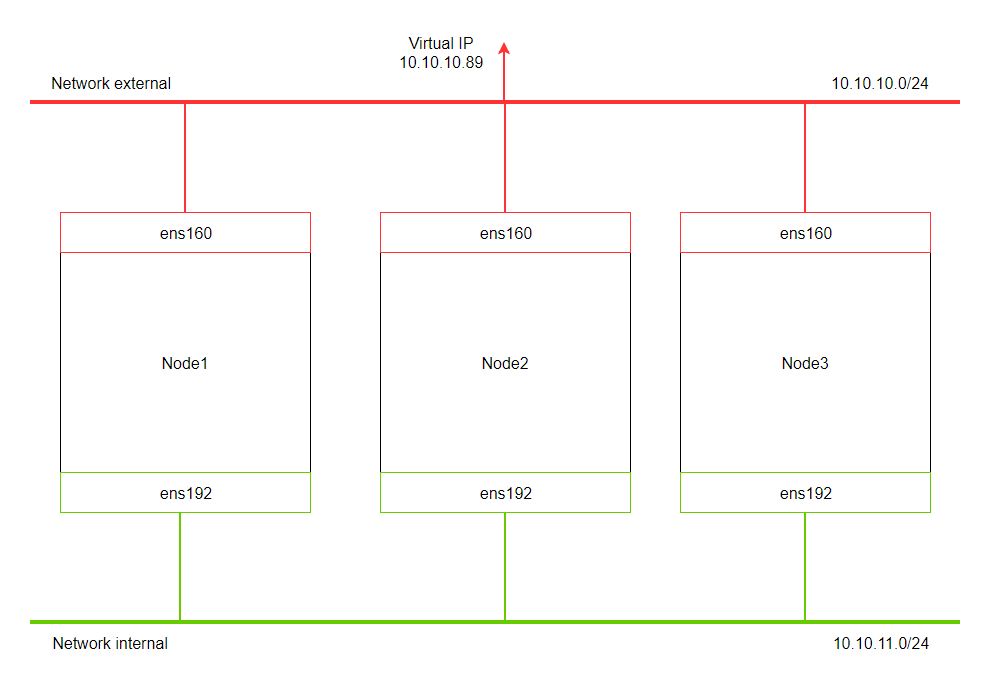
Phần 1. Chuẩn bị[#](https://blog.cloud365.vn/linux/haproxy-keepalived-apache/#phan-1-chuan-bi)

Phân hoạch

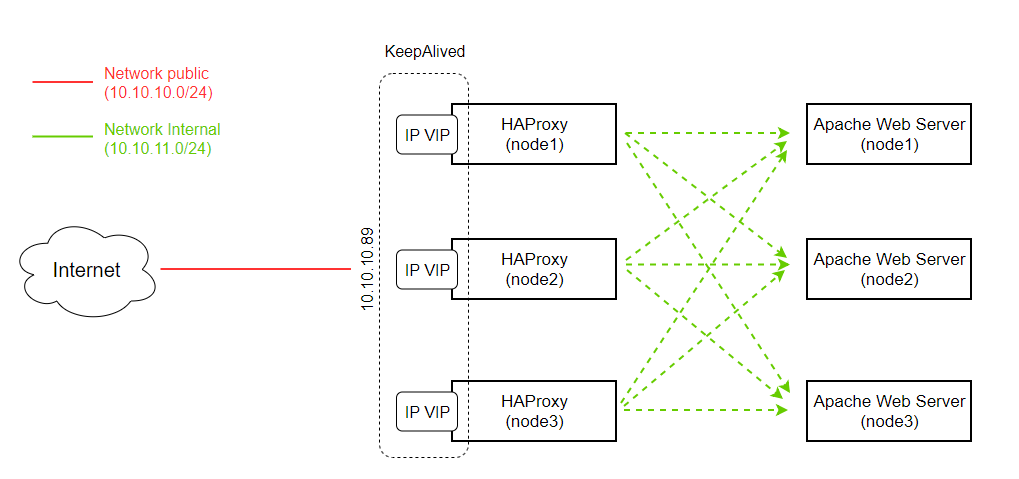


Mô hình

Mô hình triển khai



Mô hình hoạt động



Phần 2. Cầu hình Apache[#](https://blog.cloud365.vn/linux/haproxy-keepalived-apache/#phan-2-cau-hinh-apache)

Thực hiện tại node1

yum install httpd -y

cat /etc/httpd/conf/httpd.conf | grep 'Listen 80'

sed -i "s/Listen 80/Listen 10.10.11.86:8081/g" /etc/httpd/conf/httpd.conf

echo '<h1>Chào mừng tới Blog Cloud365 (Web1)</h1>' > /var/www/html/index.html

systemctl start httpd

systemctl enable httpd

Thực hiện tại node2

yum install httpd -y

cat /etc/httpd/conf/httpd.conf | grep 'Listen 80'

sed -i "s/Listen 80/Listen 10.10.11.87:8081/g" /etc/httpd/conf/httpd.conf

echo '<h1>Chào mừng tới Blog Cloud365 (Web2)</h1>' > /var/www/html/index.html

systemctl start httpd

systemctl enable httpd

Thực hiện tại node3

yum install httpd -y

cat /etc/httpd/conf/httpd.conf | grep 'Listen 80'

sed -i "s/Listen 80/Listen 10.10.11.88:8081/g" /etc/httpd/conf/httpd.conf

echo '<h1>Chào mừng tới Blog Cloud365 (Web3)</h1>' > /var/www/html/index.html

systemctl start httpd

systemctl enable httpd

Phần 2. Triển khai Keepalive[#](https://blog.cloud365.vn/linux/haproxy-keepalived-apache/#phan-2-trien-khai-keepalive)

Cài đặt gói Keepalive trên cả 3 node

yum install keepalived -y

Lưu ý: Tôi sẽ mô tả cách cấu hình trước. Sau phần mô tả sẽ có file cấu hình mẫu keepalive trên cả 3 node

Trong bài, chúng ta sẽ sử dụng keepalive để kiếm tra trạng thái của HAProxy trong mỗi 2 giây, và nếu kiếm tra thành công node sẽ được + 2 điểm.

vrrp\_script chk\_haproxy {

script "killall -0 haproxy" # check the haproxy process

interval 2 # every 2 seconds

weight 2 # add 2 points if OK

}

Tiếp theo chúng ta sẽ tạo block với tên vrrp\_instance. Đây là thành phần chính khi cấu hình HA cho dịch vụ HAProxy. Theo bài, tôi cấu hình cho keepalived kết nối với các dịch vụ tương tự nó thuộc các node khác thông giao diện mạng ens160. Mô hình keepalive là MASTER - SLAVE nên chúng ta cần chỉ định node Master và node Slave. Trong bài tôi sẽ cấu hình node1 làm node master với tham số state băng MASTER và node2, node3 làm node slave với cấu hình state bằng BACKUP.

# Trên Node1

vrrp\_instance VI\_1 {

interface ens160

state MASTER

..

}

# Trên Node2

vrrp\_instance VI\_1 {

interface ens160

state BACKUP

..

}

# Trên Node3

vrrp\_instance VI\_1 {

interface ens160

state BACKUP

..

}

Tiếp theo, chúng ta cần quan tâm tới giá trị priority trên mỗi node. Giá trị ưu tiên lần lượt trên node1 node2 node3 sẽ bằng 101 100 99.

# Trên Node1

vrrp\_instance VI\_1 {

..

priority 101

..

}

# Trên Node2

vrrp\_instance VI\_1 {

..

priority 100

..

}

# Trên Node3

vrrp\_instance VI\_1 {

..

priority 99

..

}

Cụm cần chỉ định một ID chia sẽ giữa các node. Tôi sẽ sử dụng giá trị 51 trong bài.

vrrp\_instance VI\_1 {

..

virtual\_router\_id 51

..

}

Để khai báo IP VIP cho cụm 3 node. Tôi sẽ sử dụng block virtual\_ipaddress, khai báo 1 IP cùng dải ens160. Ở đây tôi chọn ip 10.10.10.89

vrrp\_instance VI\_1 {

..

virtual\_ipaddress {

10.10.10.89/24 # virtual ip address

}

..

}

Cuối cùng là track\_script. track-script giúp keepalived xác định node nào sẽ nắm IP VIP. Như trong bài node1 được cấu hình với độ ưu tiên bằng 101. Nếu node1 kiểm tra dịch vụ HAProxy thành công thì độ ưu tiên của nó không đổi nhưng nếu kiếm tra thất bại thì độ ưu tiên của node1 sẽ giảm đi 2 và thập hơn node2. Khi đó node2 sẽ được chuyển IP VIP sang. Đây là cách dịch vụ keepalived hoạt động.

File cấu hình đầy đủ

Cấu hình keepalived trên node1

echo '

vrrp\_script chk\_haproxy {

script "killall -0 haproxy"

interval 2

weight 2

}

vrrp\_instance VI\_1 {

interface ens160

state MASTER

virtual\_router\_id 51

priority 101

virtual\_ipaddress {

10.10.10.89/24

}

track\_script {

chk\_haproxy

}

}' > /etc/keepalived/keepalived.conf

Cấu hình keepalived trên node2

echo '

vrrp\_script chk\_haproxy {

script "killall -0 haproxy"

interval 2

weight 2

}

vrrp\_instance VI\_1 {

interface ens160

state BACKUP

virtual\_router\_id 51

priority 100

virtual\_ipaddress {

10.10.10.89/24

}

track\_script {

chk\_haproxy

}

}' > /etc/keepalived/keepalived.conf

Cấu hình keepalived trên node3

echo '

vrrp\_script chk\_haproxy {

script "killall -0 haproxy"

interval 2

weight 2

}

vrrp\_instance VI\_1 {

interface ens160

state BACKUP

virtual\_router\_id 51

priority 99

virtual\_ipaddress {

10.10.10.89/24

}

track\_script {

chk\_haproxy

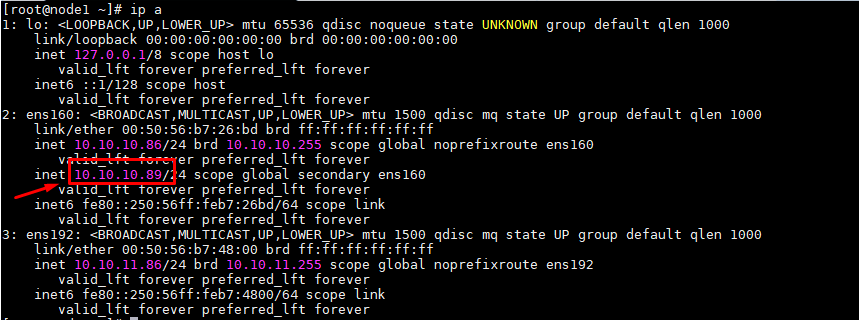
}

}' > /etc/keepalived/keepalived.conf

Khởi động dịch vụ trên 3 node

systemctl start keepalived

Kiểm tra node1, chúng ta sẽ thấy IP VIP do keepalive quản lý



Phần 3. Cài đặt Haproxy bản 1.8[#](https://blog.cloud365.vn/linux/haproxy-keepalived-apache/#phan-3-cai-%C4%91at-haproxy-ban-18)

*Thực hiện trên tất cả các node*

Cài đặt

sudo yum install wget socat -y

wget <http://cbs.centos.org/kojifiles/packages/haproxy/1.8.1/5.el7/x86_64/haproxy18-1.8.1-5.el7.x86_64.rpm>

yum install haproxy18-1.8.1-5.el7.x86\_64.rpm -y

Tạo bản backup cho cấu hình mặc định và chỉnh sửa cấu hình HAproxy

cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.bak

Cầu hình Haproxy

echo 'global

log 127.0.0.1 local2

chroot /var/lib/haproxy

pidfile /var/run/haproxy.pid

maxconn 4000

user haproxy

group haproxy

daemon

stats socket /var/lib/haproxy/stats

defaults

mode http

maxconn 8000

log global

option httplog

option dontlognull

option http-server-close

retries 3

timeout http-request 20s

timeout queue 1m

timeout connect 10s

timeout client 1m

timeout server 1m

timeout http-keep-alive 10s

timeout check 10s

listen stats

bind \*:8080 interface ens160

mode http

stats enable

stats uri /stats

stats realm HAProxy\ Statistics

stats admin if TRUE

listen web-backend

bind \*:80

balance roundrobin

cookie SERVERID insert indirect nocache

mode http

option httpchk

option httpclose

option httplog

option forwardfor

server node1 10.10.11.86:8081 check cookie node1 inter 5s fastinter 2s rise 3 fall 3

server node2 10.10.11.87:8081 check cookie node2 inter 5s fastinter 2s rise 3 fall 3

server node3 10.10.11.88:8081 check cookie node3 inter 5s fastinter 2s rise 3 fall 3' > /etc/haproxy/haproxy.cfg

Cấu hình Log cho HAProxy

sed -i "s/#\$ModLoad imudp/\$ModLoad imudp/g" /etc/rsyslog.conf

sed -i "s/#\$UDPServerRun 514/\$UDPServerRun 514/g" /etc/rsyslog.conf

echo '$UDPServerAddress 127.0.0.1' >> /etc/rsyslog.conf

echo 'local2.\* /var/log/haproxy.log' > /etc/rsyslog.d/haproxy.conf

systemctl restart rsyslog

Bổ sung cấu hình cho phép kernel có thể binding tới IP VIP

echo 'net.ipv4.ip\_nonlocal\_bind = 1' >> /etc/sysctl.conf

Kiểm tra

$ sysctl -p

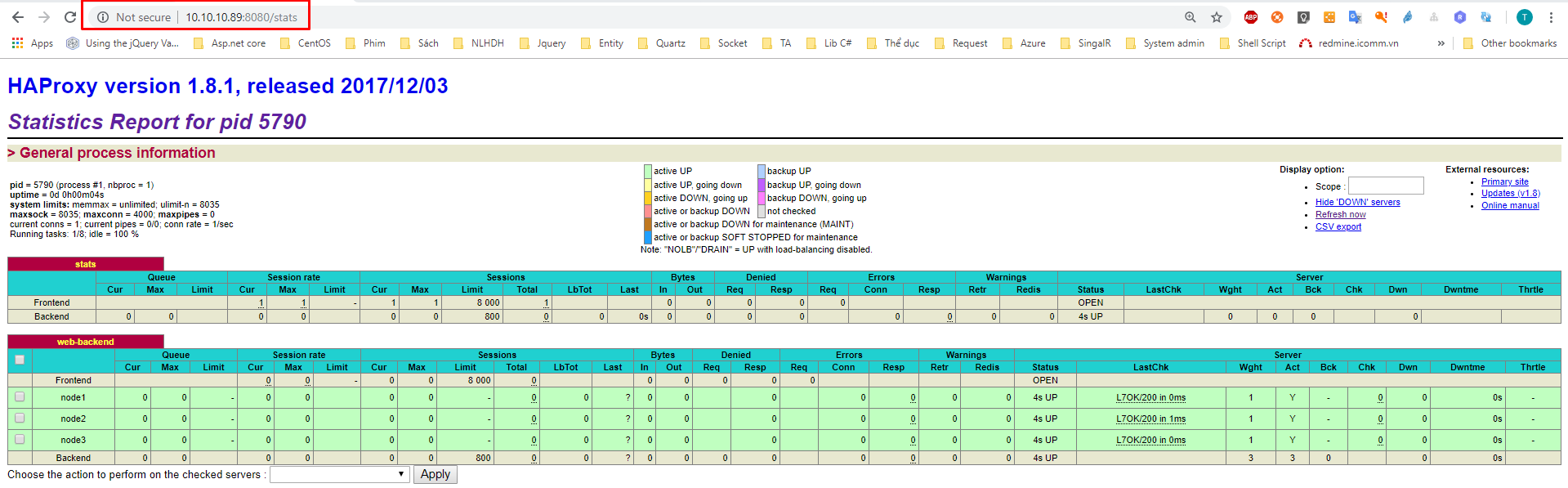
net.ipv4.ip\_nonlocal\_bind = 1

Khởi tạo dịch vụ HAProxy

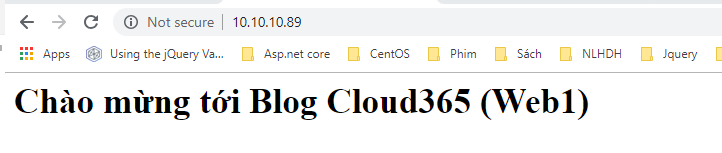
systemctl restart haproxy

systemctl enable haproxy

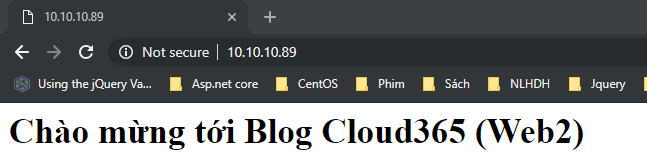
Kết quả



Do tôi cấu hình sticky session trên request vì vậy trong một thời điểm chỉ có thể kết nối tới 1 webserver. Để truy cập tới các webserver còn lại, các bạn có thể tạo phiên ẩn danh và truy cập lại.



Khi truy cập trình ẩn danh



Tổng kết[#](https://blog.cloud365.vn/linux/haproxy-keepalived-apache/#tong-ket)

Đến đây tôi đã giới thiệu cho các bạn sử dụng Keepalived để tăng tính sẵn sàng cho dịch vụ cân bằng tải, cũng như dịch vụ Web. Ngoài ra các bạn có thể tham khảo thêm seri [High Availability](https://blog.cloud365.vn/search/?q=High+Availability) và [tổng quan về HAProxy](https://blog.cloud365.vn/linux/tong-quan-haproxy/) để có thêm thông tin.