



TÌM HIỂU VỀ KEEPALIVED

TRẦN ĐỨC THỂ

Nội dung tìm hiểu

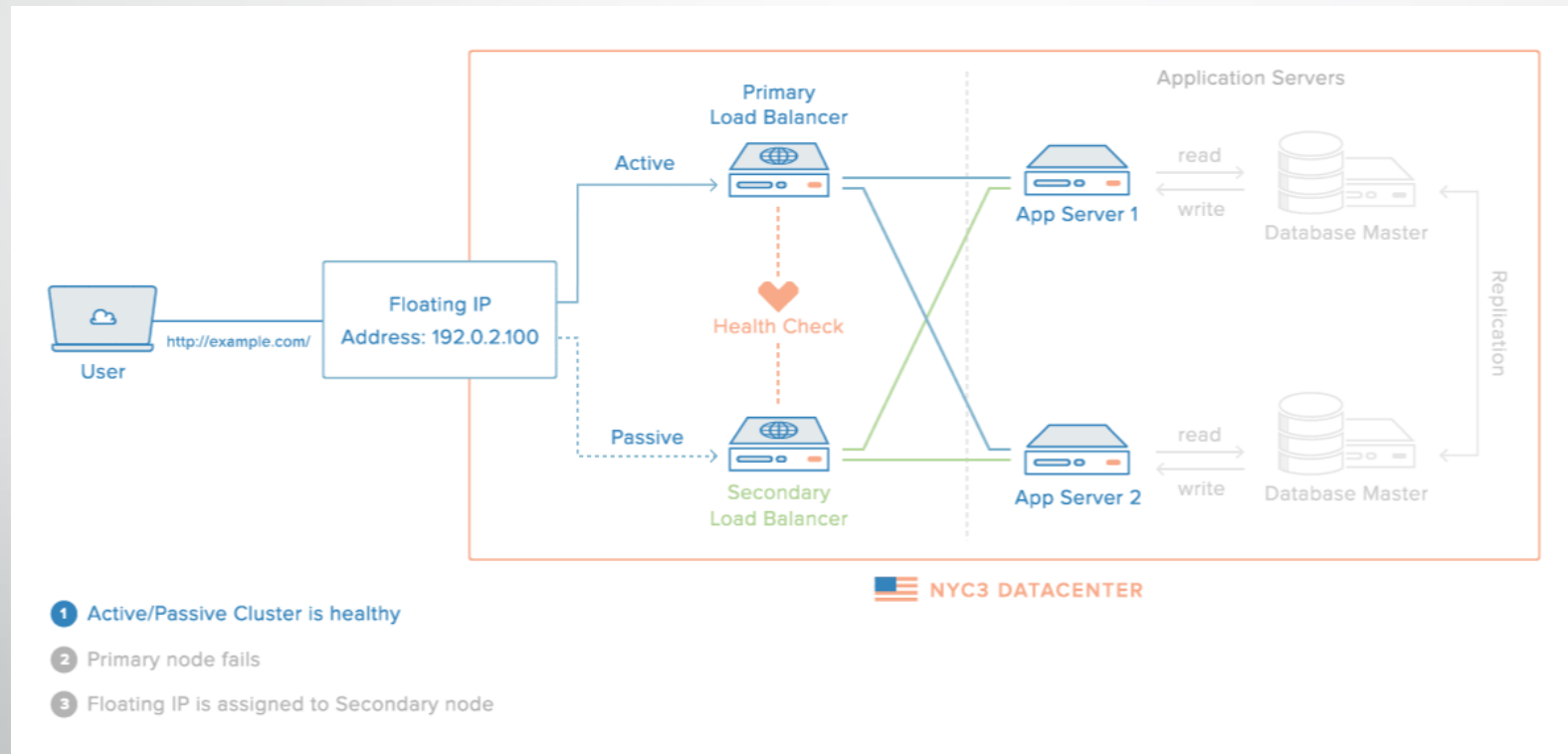
1. Tổng quan về Keepalived
2. Các cách cài đặt Keepalived trên Centos 7
3. Kiến trúc hoạt động
4. Cấu trúc file cấu hình
5. Demo tính năng

1. Tổng quan về Keepalived

- ☐ Keepalived là một chương trình dịch vụ trên Linux cung cấp khả năng tạo độ sẵn sàng cao (High Availability) cho hệ thống dịch vụ và khả năng cân bằng tải (Load Balancing) đơn giản
- ☐ Tiến trình dịch vụ của Keepalived khi khởi chạy trên Linux sẽ tạo ra 3 tiến trình cơ bản bao gồm:
 - Một tiến trình cha có tên gọi là **watchdog**, sản sinh ra 2 tiến trình con kế tiếp. Tiến trình cha sẽ quản lý theo dõi hoạt động của tiến trình con
 - Hai tiến trình con:
 - Một chịu trách nhiệm cho **VRRP framework**
 - Một chịu trách nhiệm cho **Health checking** (kiểm tra tình trạng sức khỏe)

1. Tổng quan về Keepalived

- ❑ Keepalived sẽ gom nhóm các máy chủ dịch vụ nào tham gia cụm HA, khởi tạo một **Virtual Server** đại diện cho một nhóm thiết bị đó với một **Virtual IP (VIP)** và một địa chỉ MAC vật lý của máy chủ dịch vụ đang giữ Virtual IP đó



2. Các cách cài đặt Keepalived trên VIVAS CentOS 7

- Cài qua yum

```
# yum install keepalived
```

- Cài qua source code

```
# wget https://www.keepalived.org/software/keepalived-2.2.8.tar.gz | tar xz
```

```
# cd keepalived-2.2.8
```

```
# ./configure
```

```
--prefix= . . . . Thư mục cài đặt keepalived
```

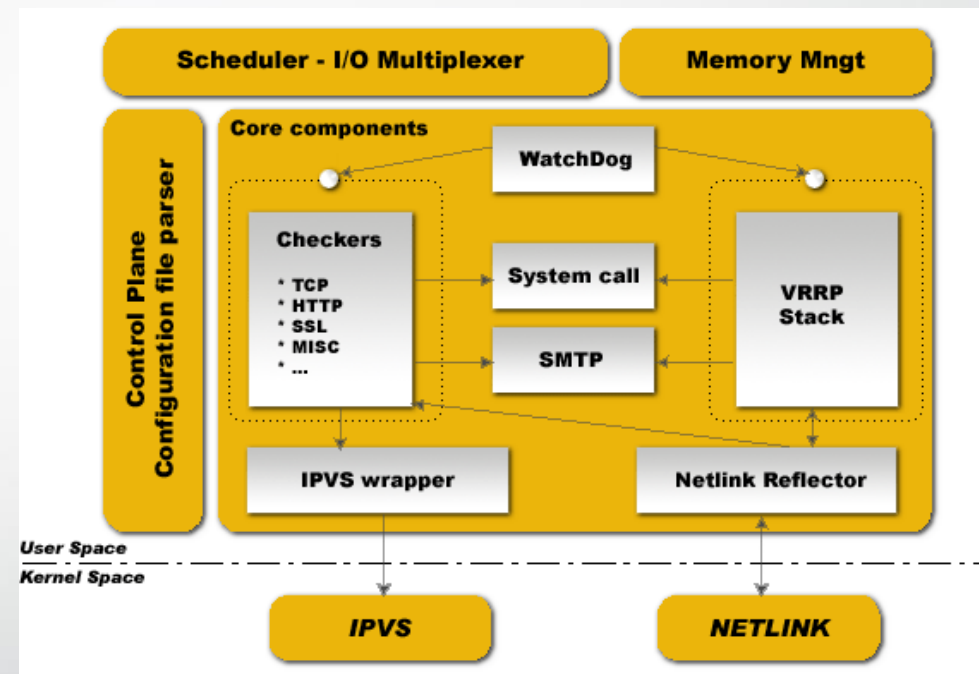
```
--sysconfdir= . . . .Thư mục cấu hình keepalived
```

```
# make && make install
```

3. Kiến trúc hoạt động

Một số thành phần chính:

- Thư viện framework **Watchdog** sẽ sản sinh ra các tiến trình con cho hoạt động giám sát tình trạng (VRRP và Healthchecking)
- **Checkers** là một trong những chức năng chính của Keepalived. Chúng có trách nhiệm kiểm tra sức khỏe của các server backup khác trong mô hình mạng Load Balancing.
- Dùng giao thức **SMTP** để thực hiện gửi email, hỗ trợ công việc quản trị.
- **System Call** là framework cung cấp khả năng thực thi các tập lệnh hệ thống. Chức năng này chủ yếu được sử dụng trong checker MISC. Trong VRRP, framework nó cung cấp khả năng thực thi các tập lệnh bổ sung trong quá trình chuyển trạng thái giao thức



3. Kiến trúc hoạt động

Một số thành phần chính:

- **VRRP (Virtual Router Redundancy Protocol):** là một giao thức tập trung vào việc đảm bảo tính sẵn sàng của các thiết bị định tuyến ảo
 - VRRP tạo ra một gateway dự phòng từ một nhóm các server. Node active được gọi là master server, tất cả các server còn lại đều trong trạng thái backup. Server master là server có độ ưu tiên cao nhất trong nhóm VRRP.
 - Chỉ số nhóm của VRRP thay đổi từ 0 đến 255; độ ưu tiên của router thay đổi từ 1 cho đến 254 (254 là cao nhất, mặc định là 100).
 - Các gói tin quảng bá của VRRP được gửi mỗi chu kỳ một giây. Các server backup có thể học các chu kỳ quảng bá từ server master.
 - Nếu có server nào có độ ưu tiên cao hơn độ ưu tiên của server master thì server đó sẽ chiếm quyền.

4. Cấu trúc trong keepalived.conf

➤ Global Definitions

Được sử dụng để xác định các cài đặt chung ảnh hưởng đến toàn bộ quá trình chạy của Keepalived

```
Global_defs {
```

```
    notification_email {
```

```
        email
```

```
        email
```

```
    }
```

```
    notification_email_from email
```

```
    smtp_server host
```

```
    smtp_connect_timeout num
```

```
}
```

#Địa chỉ email được sử dụng để gửi thông báo khi xảy ra sự kiện

#Địa chỉ email được sử dụng làm người gửi cho email thông báo

#Địa chỉ máy chủ SMTP được sử dụng để gửi email thông báo

#Thời gian chờ kết nối tới máy chủ SMTP

4. Cấu trúc trong keepalived.conf

➤ **vrrp_instance**

Đại diện cho một nhóm máy chủ (cluster) hoạt động cùng nhau để cung cấp tính sẵn sàng cao cho một Virtual IP Address (VIP) cụ thể

```
vrrp_instance string {                                #Xác định tên vrrp_instance
    state MASTER|BACKUP                                #Xác định trạng thái ban đầu
    interface string                                  #Xác định tên của giao diện mạng mà VRRP sẽ sử dụng để hoạt động
    virtual_router_id num                             #Xác định định danh cho vrrp_instance
    priority num                                       #Mức ưu tiên của máy chủ trong cụm VRRP
    advert_int num                                    #Xác định khoảng thời gian giữa các thông điệp VRRP được gửi đi
    authentication {                                #Cài đặt xác thực cho giao thức VRRP
        auth_type PASS|AH                            #Loại xác thực (PASS: xác thực mật khẩu, AH: xác thực IPSEC-AH)
        auth_pass string                             #Mật khẩu xác thực
    }
    virtual_ipaddress {                              #Địa chỉ VIP của VRRP
        ip
    }
}
```

4. Cấu trúc trong keepalived.conf

➤ **vrrp_server**

Khai báo cấu hình load balancing cho keepalived

virtual_server ip port {

delay_loop num #Thời gian chờ giữa các vòng lặp kiểm tra trạng thái Real Server (RS)

lb_algo rr/wrr/lc/wlc/sh/dh/lbrc #Chọn thuật toán cân bằng tải

lb_kind NAT/DR/TUN #Xác định loại cân bằng tải

persistence_timeout num #Thiết lập thời gian timeout cho tính liên tục trong cân bằng tải

protocol UDP/TCP #Giao thức mạng sử dụng cho VS

real_server ip port {

weight num #trọng số (weight) của RS trong cân bằng tải

...

}

connect_timeout num #Thời gian chờ kết nối tối đa khi kiểm tra RS

nb_get_retry num #Số lần thử lại tối đa khi kiểm tra RS không thành công

delay_before_retry num #Thời gian chờ giữa các lần thử lại kiểm tra RS không thành công

THANK YOU