IP Virtual Server (IPVS) and Keepalive

ICD - Infraestruturas e Centros de Dados

2020/2021

The main goal of this guide is to understand how to use and configure IPVS and Keepalive at in Linux systems.

For the exercises described next, the following tools must be installed,

- VirtualBox - `https://www.virtualbox.org`

while useful documentation is available at:

- IPVS- `http://www.linuxvirtualserver.org`
- Keepalive - `https://keepalived.readthedocs.io/en/latest/#`

Steps

VM Deployment and Configuration

1. Clone the Template VM (generate new MAC) done at the Warmup exercise and launch it.

2. With *nmtui*:

   Change the hostname to *director*

   Change the IP of the network cards to *10.0.0.6/24* and *10.10.10.6/24*. The identifiers of the cards should be *enps0s8* and *enp0s9*, respectively (this can change across systems).

   Ensure that IPV6 is disabled, while the cards are in manual mode for IPV4.

   Run *nmcli connection down enp0sX* and *nmcli connection up enp0sX* for refreshing the configurations of the network interfaces.

3. Install the following packages:

   ```
   yum install net-tools telnet tcpdump ipvsadm  keepalived  haproxy
   ```

IPVS Setup

1. Configure IPVS routing by running:

```
ipvsadm -A -t 10.10.10.6:80 -s rr
ipvsadm -a -t 10.10.10.6:80 -r 10.0.0.10 -m
ipvsadm -a -t 10.10.10.6:80 -r 10.0.0.11 -m
```

2. Activate IP forwarding by:

   Running *echo 1 > /proc/sys/net/ipv4/ip_forward*.

   Edit */etc/sysctl.d/ip_forward.conf* and add *net.ipv4.ip_forward = 1*

3. Run the command *ipvsadm* to check the routing rules.

Web Servers Setup

1. Clone 2x the VM template done at the Warmup exercise and launch the VMs.
2. Change the IPs to *10.0.0.10/24* and *10.0.0.11/24*, as well as the names to *ws1* and *ws2*. The second host-only network driver is not needed in these VMs.
3. Restart network interfaces if necessary.
4. At both VMs install the following packages.

```
yum install net-tools telnet tcpdump httpd
```

5. Start service httpd *systemctl start httpd*.
6. Enable service httpd *systemctl enable httpd*.
7. With *nmtui*:

   Disable the NAT network interface (*enp0s3* - the name may change).

   Add the IP address *10.0.0.6* as the gateway for (*enp0s8* - the name may change) in both VMs.

   Run *nmcli connection down enp0s8* and *nmcli connection up enp0s8* for refreshing the configurations of the network interfaces.

Testing

1. At both VMs write a different message at the */var/www/html/index.html* file (page being served by the web servers).
2. At the Host machine access the website (*10.10.10.6:80*). Try both with your browser and the command curl to check the differences.
3. Access it multiple times to see the requests load balancing between web servers.

Keepalive

1. Change the gateway IP address to *10.0.0.8* (*enp0s8* - the name may change) at the web server VMs. Remember that the NAT interfaces need to be turned off.

   Run *nmcli connection down enp0s8* and *nmcli connection up enp0s8* for refreshing the configurations of the network interfaces.

2. Clone the *director* VM and launch it.

   Change the hostname to *director_backup*

   Change the IP of the network cards to *10.0.0.7/24* and *10.10.10.7/24*. The identifiers of the cards should be *enps0s8* and *enp0s9*, respectively (this can change across systems).

3. On both the director and director_backup VMs, apply the following configuration at */etc/keepalived/keepalived.conf*. Note that the director VM state is MASTER and the director_backup VM state should be BACKUP.

```
vrrp_instancerrp_sync_group VG1 {
        group {
                RH_EXT
                RH_INT
        }
}
vrrp_instance RH_EXT {
        state MASTER #(BACKUP at director_backup VM)
        interface enp0s9
        virtual_router_id 50
        priority 50
        advert_int 1
        authentication {
                auth_type PASS
                auth_pass password123
        }
        virtual_ipaddress {
                10.10.10.8/24
        }
}
vrrp_instance RH_INT {
        state MASTER #(BACKUP at director_backup VM)
        interface enp0s8
        virtual_router_id 50
        priority 50
        advert_int 1
        authentication {
                auth_type PASS
```

```
                auth_pass password123
        }
        virtual_ipaddress {
                10.0.0.8/24
        }
}

virtual_server 10.10.10.8 80 {
        delay_loop 6
        lb_algo rr
        lb_kind NAT
        protocol TCP
        real_server 10.0.0.10 80 {
                TCP_CHECK {
                        connect_timeout 10
                }
        }
        real_server 10.0.0.11 80 {
                TCP_CHECK {
                        connect_timeout 10
                }
        }
}
```

4. Activate IP forwarding on both VMs by (ignore if you have done this already):

   Running *echo 1 > /proc/sys/net/ipv4/ip_forward*.

   Edit */etc/sysctl.d/ip_forward.conf* and add *net.ipv4.ip_forward = 1*

5. Restart Keepalived service *sudo systemctl restart keepalived.service*.

Testing

1. At both the *director* and *director_backup* VMs, run *ip add* to list the available network interfaces. Check that the virtual ip addresses were created by keepalive at the *director* VM.

2. Also, at both VMs, run the command *ipvsadm* to check that keepalive created appropriate routing rules.

3. At the Host machine access the website (*10.10.10.8:80*). Try both with your browser and the command curl to check the differences.

   Access it multiple times to see the requests load balancing between web servers.

4. Power off the *director* VM and check that the service is now being supported by *director_backup* VM.

Note that the virtual ip addresses were created by keepalive at the *director_backup* VM.

Learning outcomes Experiment IPVS and KeepAlive deployment and configuration. Assess how to load balance requests across stateless services. Assess how to deploy fault-tolerant load balancers.