6. High-Available Reverse Proxy

Introduction

Please ensure that you have already set up web servers in BOTH web01 and web02 as haproxy is dependent on them.

**Objective:**

At the end of this guide, you will learn how to configure ha-prx01 and ha-prx02 as high-available reverse proxies using haproxy and keepalived.

**Login**

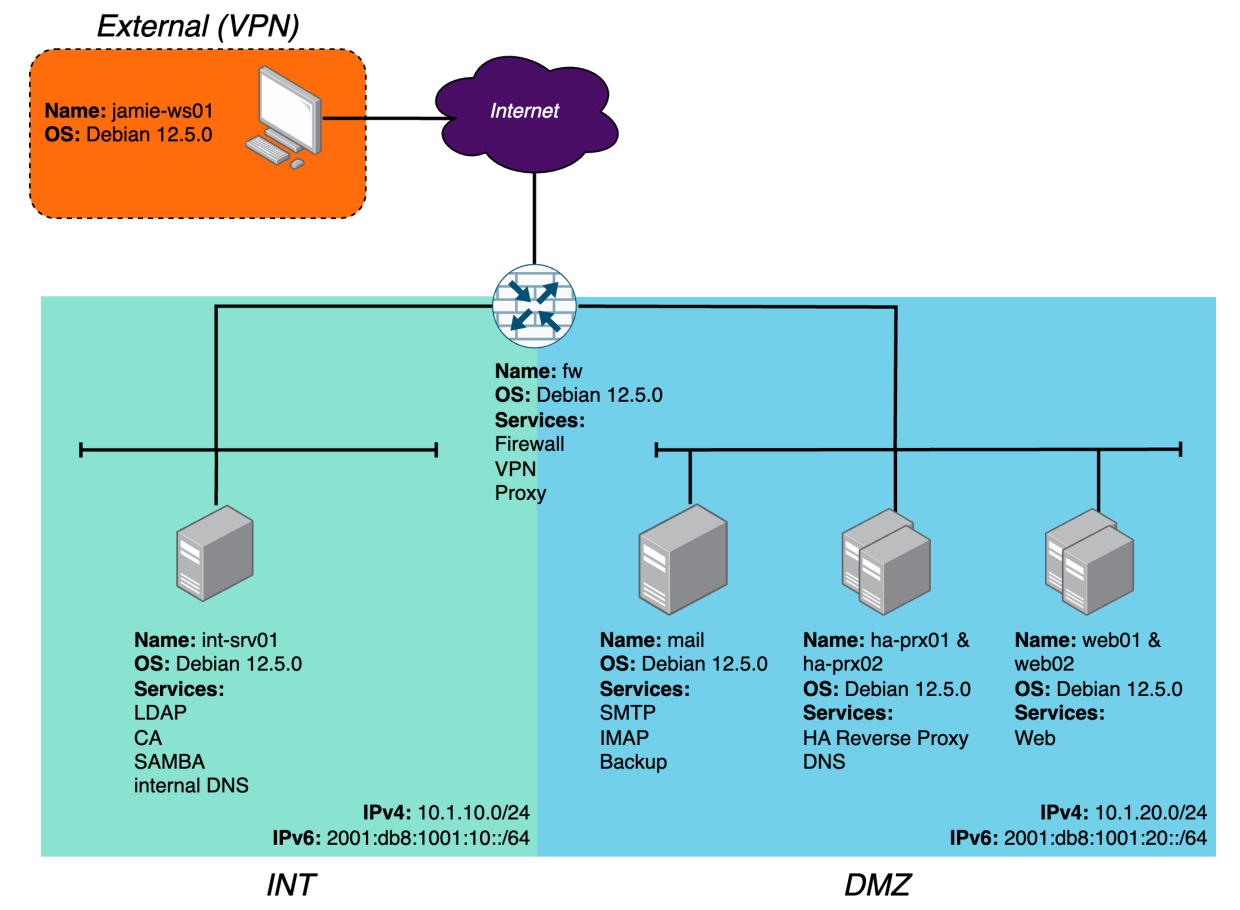
The login credential for all server and client machines:

Username: root / user

Password: Skill39@Lyon

**Network Topology**

This will be the network topology that will be referenced for setting up the infrastructure.



We will be using keepalived and haproxy for high-available reverse proxy.

**Install haproxy and keepalived on both ha-prx01 and ha-prx02:**

apt install keepalived haproxy -y

## Keepalived

**Do the following on ha-prx01**

apt install keepalived -y

cp /etc/keepalived/keepalived.conf.sample /etc/keepalived/keepalived.conf

vim /etc/keepalived/keepalived.conf

| global\_defs {  router\_id ha-prx01  vrrp\_skip\_check\_adv\_addr  vrrp\_strict  vrrp\_garp\_interval 0  vrrp\_gna\_interval 0  }  vrrp\_instance VI\_1\_IPv4 {  state MASTER  interface ens33  virtual\_router\_id 20  priority 100  advert\_int 1  authentication {  auth\_type PASS  auth\_pass Skill39@Lyon  }  virtual\_ipaddress {  10.1.20.20/24  }  }  vrrp\_instance VI-1\_IPv6 {  state MASTER  interface ens33  virtual\_router\_id 20  priority 100  advert\_int 1  authentication {  auth\_type PASS  auth\_pass Skill39@Lyon  }  virtual\_ipaddress {  2001:db8:1001:20::20/64  }  } |
| --- |

systemctl restart keepalived

**Copy the file to ha-prx02. You can also redo the above steps in ha-prx02 but copying will be faster.**

scp /etc/keepalived/keepalived.conf [root@10.1.20.22:/etc/keepalived/](about:blank)

**Do note that changes need to be made after copying the file to ha-prx02.**

In ha-prx02, MASTER needs to be changed to BACKUP as shown below:

| global\_defs {  router\_id ha-prx01  vrrp\_skip\_check\_adv\_addr  vrrp\_strict  vrrp\_garp\_interval 0  vrrp\_gna\_interval 0  }  vrrp\_instance VI\_1\_IPv4 {  state BACKUP  interface ens33  virtual\_router\_id 20  priority 100  advert\_int 1  authentication {  auth\_type PASS  auth\_pass Skill39@Lyon  }  virtual\_ipaddress {  10.1.20.20/24  }  }  vrrp\_instance VI-1\_IPv6 {  state BACKUP  interface ens33  virtual\_router\_id 20  priority 100  advert\_int 1  authentication {  auth\_type PASS  auth\_pass Skill39@Lyon  }  virtual\_ipaddress {  2001:db8:1001:20::20/64  }  } |
| --- |

**Restart the service in both ha-prx01 and ha-prx02.**

systemctl restart keepalived -y

## Haproxy

**Do the following in ha-prx01**

scp root@10.1.10.10:/etc/ssl/CA/certs/web.crt /etc/haproxy

scp [root@10.1.10.10:/etc/ssl/CA/certs/ca.crt](about:blank) /etc/haproxy/

scp [root@10.1.10.10:/etc/ssl/CA/private/web.key](about:blank)  /etc/haproxy

cat /etc/haproxy/web.key >> /etc/haproxy/web.crt

cat /etc/haproxy/ca.crt >> /etc/haproxy/web.crt

vim /etc/haproxy/haproxy.cfg

| frontend http-in  bind :::80 v4v6  mode http    frontend https-in  bind :::443 v4v6 ssl crt /etc/haproxy/web.crt  http-response set-header via-proxy “ha-prx01”  default\_backend www    backend www  http-request redirect scheme https code 301 if !{ ssl\_fc }  balance roundrobin  option forwardfor  option http-server-close  http-request set-header via-proxy “ha-prx01”  http-request add-header X-Forwarded-Port https if { ssl\_fc }  server web01.dmz.worldskills.org [2001:db8:1001:20::31]:80 check  server web02.dmz.worldskills.org [2001:db8:1001:20::32]:80 check |
| --- |

**Copy the file to ha-prx02**

scp /etc/haproxy/haproxy.cfg [root@10.1.20.22:/etc/haproxy/haproxy.cfg](about:blank)

**After copying the file to ha-prx02, change the header.**

| frontend http-in  bind :::80 v4v6  mode http    frontend https-in  bind :::443 v4v6 ssl crt /etc/haproxy/web.crt  http-response set-header via-proxy “ha-prx02”  default\_backend www    backend www  http-request redirect scheme https code 301 if !{ ssl\_fc }  balance roundrobin  option forwardfor  option http-server-close  http-request set-header via-proxy “ha-prx02”  http-request add-header X-Forwarded-Port https if { ssl\_fc }  server web01.dmz.worldskills.org [2001:db8:1001:20::31]:80 check  server web02.dmz.worldskills.org [2001:db8:1001:20::32]:80 check |
| --- |

**Restart the service on both ha-prx01 and ha-prx02**

systemctl restart haproxy

## NAT

After configuring haproxy and keepalived, we configure NAT on the firewall to forward HTTP, HTTPS and DNS to 10.1.20.20 (virtual IP address which is ha-prx01)

**Do the following in fw**

vim /etc/nftables.conf

table ip nat {

chain postrouting {

type nat hook postrouting priority srcnat;

#interface with public IP

iif “ens33” tcp dport 80 dnat to 10.1.20.20:80;

iif “ens33” tcp dport 443 dnat to 10.1.20.20:443;

iif “ens33” tcp dport 53 dnat to 10.1.20.20:53;

iif “ens33” udp dport 53 dnat to 10.1.20.20:53;

}

chain prerouting {

type nat hook prerouting priority dstnat;

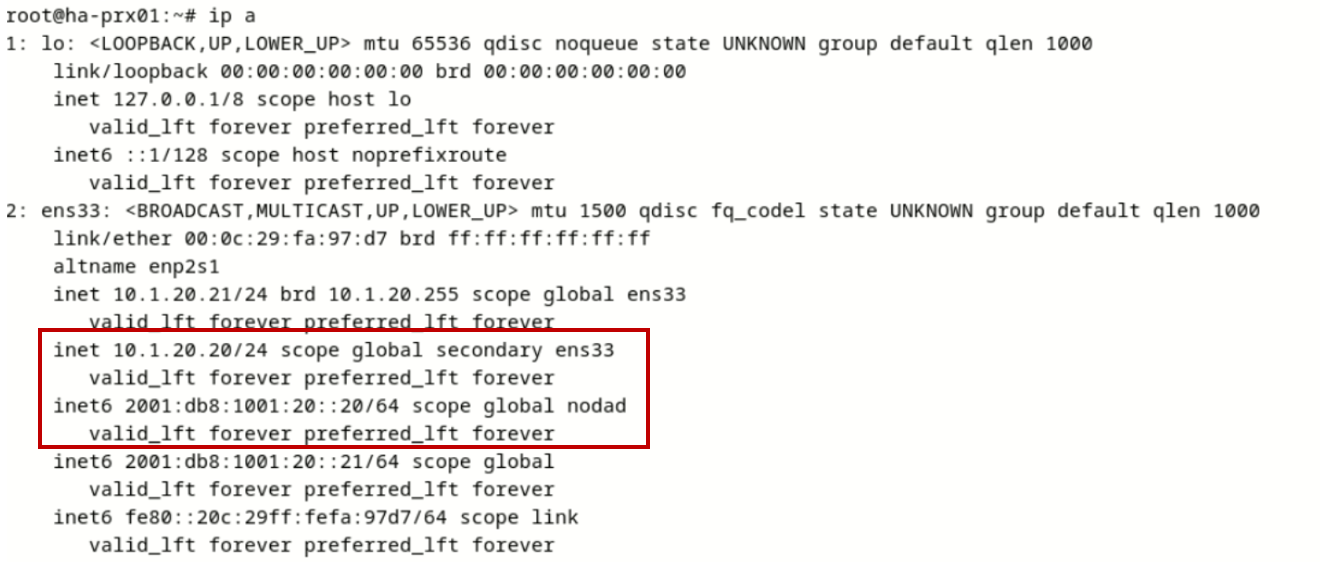
oif “ens33” masquerade;

}

}

systemctl restart nftables.service

Test



Access the web to test if the reverse proxy and port forwarding works

