



# HAProxy Server Configuration Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

## Session - 47 Agenda:

### HAProxy Server Configuration Using Ansible:

To configure HAProxy server we would require 3 node machines; out of these three node machines one node is used as HAProxy server while the other two nodes will be used as load-balancer (using round robin approach) for balancing the load.

#### Lab Environment:

```
192.168.229.128 haproxy-server  
192.168.229.129 nginx-node01  
192.168.229.131 nginx-node02
```

Let's create an Ansible playbook to setup the HAProxy server at the managed node(s).

```
$ vim haproxy-server.yml
```

```
---  
- name: HAProxy Server Configuration Playbook  
  hosts: localhost  
  become: true  
  tasks:  
    - name: Setting up the static hostname in the server machine.  
      hostname:  
        name: haproxy-server  
        use: systemd  
  
    - name: Making entries in the /etc/hosts file for the server hostnames & IP Addresses  
      blockinfile:  
        dest: /etc/hosts  
        block: |  
          192.168.229.128 haproxy-server  
          192.168.229.129 nginx-node01  
          192.168.229.131 nginx-node02  
      insertafter: EOF  
  
    - name: Installing HAProxy packages in the machine.  
      dnf:  
        name: haproxy  
        state: latest  
  
    - name: Copying the /etc/haproxy/haproxy.cfg file using ansible jinja template.  
      template:  
        src: haproxy.cfg.j2  
        dest: /etc/haproxy/haproxy.cfg  
        force: true  
  
    - name: Making changes in the /etc/rsyslog.conf file.  
      replace:  
        dest: /etc/rsyslog.conf  
        regexp: '^#module(load="imudp")'  
        replace: 'module(load="imudp")'
```



# HAProxy Server Configuration Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

```
- name: Making changes in the /etc/rsyslog.conf file.
  replace:
    dest: /etc/rsyslog.conf
    regexp: '^#input(type="imudp" port="514")'
    replace: 'input(type="imudp" port="514")'

- name: Creating haproxy.conf file for the rsyslog.
  copy:
    dest: "/etc/rsyslog.d/haproxy.conf"
    content: |
      local2.=info /var/log/haproxy-access.log
      local2.notice /var/log/haproxy-info.log

- name: Restarting & enabling the rsyslog service.
  service:
    name: rsyslog
    state: restarted
    enabled: yes

- name: Turning on the haproxy_connect_any SELinux boolean.
  command: setsebool -P haproxy_connect_any 1

- name: Allowing HTTP traffic in the firewall.
  firewalld:
    service: http
    zone: public
    permanent: true
    immediate: true
    state: enabled

- name: Starting & enabling the haproxy service.
  service:
    name: haproxy
    state: started
    enabled: yes

- hosts: node1
  become: true
  tasks:
    - name: Setting up the static hostname in the node1 machine.
      hostname:
        name: nginx-node01
        use: systemd

    - name: Making entries in the /etc/hosts file for the server hostnames & IP Addresses
      blockinfile:
        dest: /etc/hosts
        block: |
          192.168.229.128 haproxy-server
          192.168.229.129 nginx-node01
```



# HAProxy Server Configuration Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

192.168.229.131 nginx-node02

insertafter: EOF

```
- name: Installing nginx packages in the machine.
  dnf:
    name: nginx
    state: latest

- name: Copying the image file to the /usr/share/nginx/html/ directory.
  ansible.builtin.copy:
    src: /home/vikasnehra/NehraClassesLogo.png
    dest: /usr/share/nginx/html/NehraClassesLogo.png
    mode: '0644'

- name: Creating the index.html file for node1.
  copy:
    dest: "/usr/share/nginx/html/index.html"
    content: |
      
      <h1>Nehra Classes Are Awesome.</h1>
      <i>This page is hosted on node1 machine using nginx.</i>

- name: Allowing HTTP traffic in the firewall.
  firewalld:
    service: http
    zone: public
    permanent: true
    immediate: true
    state: enabled

- name: Starting & enabling the nginx service.
  service:
    name: nginx
    state: started
    enabled: yes

- hosts: node3
  become: true
  tasks:
    - name: Setting up the static hostname in the node3 machine.
      hostname:
        name: nginx-node02
        use: systemd

    - name: Making entries in the /etc/hosts file for the server hostnames & IP Addresses
      blockinfile:
        dest: /etc/hosts
        block: |
          192.168.229.128 haproxy-server
          192.168.229.129 nginx-node01
```



# HAProxy Server Configuration Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

192.168.229.131 nginx-node02

insertafter: EOF

- name: Installing nginx packages in the machine.  
dnf:
    - name: nginx
    - state: latest
  
  - name: Copying the image file to the /usr/share/nginx/html/ directory.  
ansible.builtin.copy:
    - src: /home/vikasnehra/NehraClassesLogo.png
    - dest: /usr/share/nginx/html/NehraClassesLogo.png
    - mode: '0644'
  
  - name: Creating the index.html file for node2.  
copy:
    - dest: "/usr/share/nginx/html/index.html"
    - content: |  
  
<h1>Nehra Classes Are Awesome.</h1>  
<i>This page is hosted on node2 machine using nginx.</i>
  
  - name: Allowing HTTP traffic in the firewall.  
firewalld:
    - service: http
    - zone: public
    - permanent: true
    - immediate: true
    - state: enabled
  
  - name: Starting & enabling the nginx service.  
service:
    - name: nginx
    - state: started
    - enabled: yes
- ...

We would require the ansible.posix collection which we can install from Ansible Galaxy.

\$ ansible-galaxy collection install ansible.posix

We would also require the community.general collection which we can install from Ansible Galaxy.

\$ ansible-galaxy collection install community.general

Now, we can execute the ansible playbook to setup the HAProxy server at the managed node(s).

\$ ansible-playbook haproxy-server.yml

Login to haproxy server and run the curl command couple of times to see whether traffic is distributed in round-robin way.

\$ curl 192.168.229.128

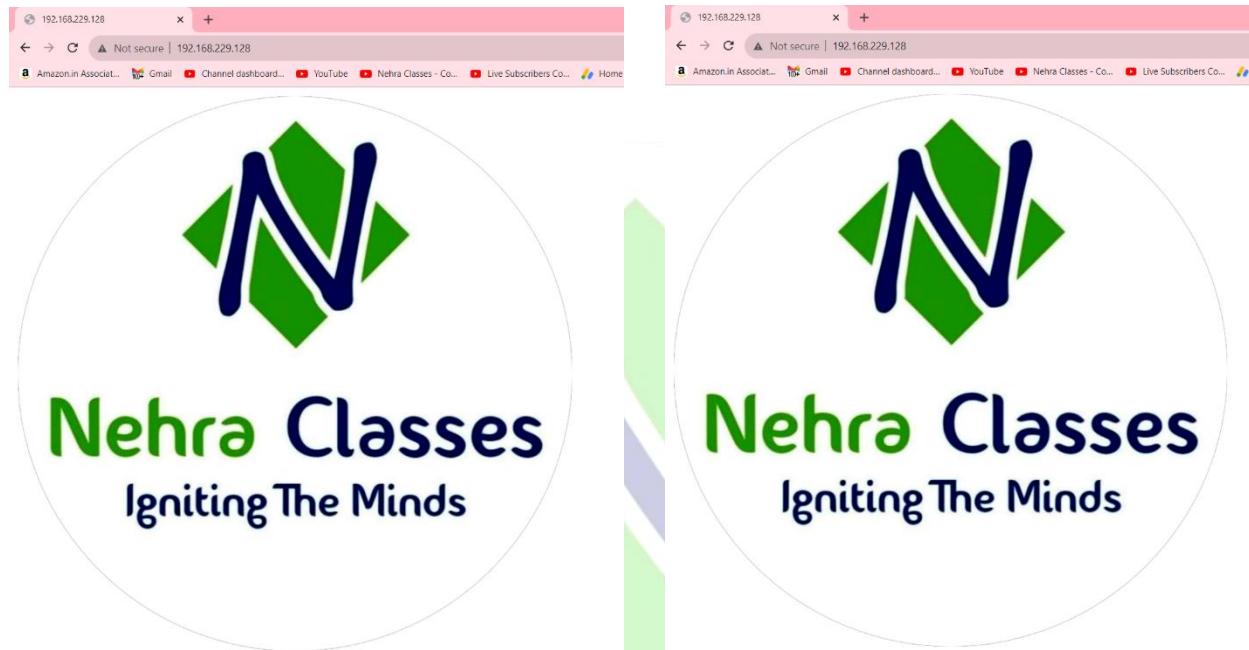


# HAProxy Server Configuration Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

```
$ curl 192.168.229.128
$ curl 192.168.229.128
$ curl 192.168.229.128
```

You can verify the same using any web browser as well.



Nehra Classes Are Awesome.

*This page is hosted on node1 machine using nginx.*

Nehra Classes Are Awesome.

*This page is hosted on node2 machine using nginx.*

Perfect, this confirms HAProxy is working properly as it is distributing traffic between two nodes. Above confirms that HAProxy has configured successfully on RHEL9.

You can view the status of your haproxy via web browser, type URL will be <http://192.168.229.128/haproxy?stats>

Statistics Report for HAProxy																															
HAProxy version 2.4.7-b5e51a5, released 2021/10/04																															
Statistics Report for pid 37489																															
> General process information																															
<p>pid = 37489 (process #1, nproc = 1, nbthread = 4)          uptime = 0d 0h0m27s          system limits: memmax = unlimited; ultimlim = 8041          maxsock = 8041; maxconn = 4000; maxpipes = 0          current conn = 1; current pipes = 0/0; conn rate = 0/sec; bit rate = 0.543 kbps          Running tasks: 0/23; idle = 100 %</p>																															
<p>Display option: <input type="text"/>  <ul style="list-style-type: none"> <li>• Scope: <input type="text"/></li> <li>• Hide DOWN servers</li> <li>• Refresh now</li> <li>• CSV export</li> <li>• JSON export (schema)</li> </ul> </p>																															
http_balancer			Queue			Session rate			Sessions			Bytes			Denied			Errors			Warnings			Server							
Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntime	Thrtie		
Frontend			1	2	-	1	2	3 000	5			2 198	3 853 696	0	0	2						OPEN									
<p>Note: "NOLB"="DRAIN" = UP with load-balancing disabled.</p>																															
nginx_webservers			Queue			Session rate			Sessions			Bytes			Denied			Errors			Warnings			Server							
Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntime	Thrtie		
nginx-node01	0	0	-	0	2	0	1	-	3	3	1m37s	1 263	1 928 908	0	0	0	0	0	0	0	0	7m35s	UP	1/1	Y	-	1	1	46s	-	
nginx-node02	0	0	-	0	1	0	1	-	2	2	1m37s	935	1 924 788	0	0	0	0	0	0	0	0	0	6m52s	UP	1/1	Y	-	1	1	1m28s	-
Backend	0	0	-	0	3	0	1	300	5	5	1m37s	2 198	3 853 696	0	0	0	0	0	0	0	0	0	7m35s	UP	2/2	2	0	0	1	45s	
app			Queue			Session rate			Sessions			Bytes			Denied			Errors			Warnings			Server							
Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntime	Thrtie		
app1	0	0	-	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	8m20s	DOWN	1/1	Y	-	1	1	8m20s	-	
app2	0	0	-	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	8m20s	DOWN	1/1	Y	-	1	1	8m20s	-
app3	0	0	-	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	8m19s	DOWN	1/1	Y	-	1	1	8m19s	-
app4	0	0	-	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	8m19s	DOWN	1/1	Y	-	1	1	8m19s	-
Backend	0	0	-	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	8m19s	

HAProxy server is working as expected.

Thank You