QUESTION:

Set up a cluster of Linux servers running a web server (e.g., Apache) with a load balancer (e.g., HAProxy) for redundancy and scalability.

Demonstrate expertise in system administration, high availability (HA) concepts, load balancing, and web server configuration.

PREREQUISITES:

Choose your Linux Distribution: Select a suitable Linux distribution for your servers (e.g., Ubuntu, CentOS). Ensure all 3 servers use the same distribution.

Server Configuration: Make sure all servers (3 Servers) have the same hardware specifications and network connectivity.

Domain Name and IP Addresses: Have a registered domain name and static IP addresses for each server and the load balancer.

e.g.,

3 Ubuntu distribution 22.04/24.04 VPS (Virtual Private Server)

-VPS 1: Install HAProxy

IP1:172.31.89.194

root_password: Ex@123

-VPS 2,3: Use as a web Server

IP2:172.31.92.29

root_password: Ex@123

IP3:172.31.95.220

root_password: Ex@123

ANSWER:

VPS 1:

- To Add/change Hostname:

#hostnamectl set-hostname hostname
e.g., #hostname set-hostname Proxy

- To configure /etc/hosts file

```
#vim /etc/hosts
```

Output:

```
127.0.0.1 localhost
172.31.24.203   ws1 …….. (Add Webserver 1 IP and Hostname)
172.31.28.32    ws2 …….. (Add Webserver 2 IP and Hostname)
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
~
```

After changes in /etc/hosts file to apply below any one command to save configuration.

```
#systemctl restart systemd-hostnamed
```

Or

```
#systemctl restart dnsmasq.service
```

Or

```
#/bin/systemctl restart system.hostnamed
```

```
#ping IP Address     ……e.g., #ping 172.31.24.203 …. # (Webserver1)
#ping IP Address     ……e.g., #ping 172.31.28.32 …. # (Webserver2)
```

Or

```
#Ping Hostname ……e.g., #ping ws1
#ping Hostname ……e.g., #ping ws2
```

- Update & upgrade VPS.

```
#apt update -y && apt upgrade -y
```

- Installing HAProxy:

It is a command used to display information about the HAProxy package available in your system's package repository.

```
#sudo apt show haproxy
```

Then install HAProxy as you normally would.

```
#sudo apt install -y haproxy
```

Afterwards, you can double check the installed version number with the following command.

```
#haproxy -v
```

Output:

```
HAProxy version 2.8.5-1ubuntu3 2024/04/01 - https://haproxy.org/
Status: long-term supported branch - will stop receiving fixes around Q2 2028.
Known bugs: http://www.haproxy.org/bugs/bugs-2.8.5.html
Running on: Linux 6.8.0-1009-aws #9-Ubuntu SMP Fri May 17 14:39:23 UTC 2024 x86_64
```

- Configuring the load balancer

Once installed, HAProxy should have a template for configuring the load balancer. Open the configuration file, for example, using nano with the command underneath.

```
#sudo nano /etc/haproxy/haproxy.cfg
```

Or

```
#sudo Vim /etc/haproxy/haproxy.cfg
```

Configuration File:

```
global
    log /dev/log    local0
    log /dev/log    local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

    # Default SSL material locations
    ca-base /etc/ssl/certs
    crt-base /etc/ssl/private

    # See: https://ssl-config.mozilla.org/#server=haproxy&server-version=2.0.3&config=intermediate
    ssl-default-bind-ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-
POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384
    ssl-default-bind-ciphersuites
TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
    ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets

defaults
    log    global
    mode    http
    option  httplog
    option  dontlognull
    timeout connect 5000
```

```
    timeout client  50000
    timeout server  50000
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
    errorfile 504 /etc/haproxy/errors/504.http

frontend http_front
  bind *:80
  stats uri /haproxy?stats
  default_backend http_back

backend http_back
  balance roundrobin
  server <server1 name> <private IP 1>:80 check
  server <server2 name> <private IP 2>:80 check
```

e.g.,

```
frontend http_front
  bind *:80
  stats uri /haproxy?stats
  default_backend http_back

backend http_back
  balance roundrobin
  server ws1          172.31.24.203:80 check
  server ws2          172.31.28.32:80 check
```

Esc:wq

After making the configurations, save the file and restart HAProxy with the next command.

```
#sudo systemctl restart haproxy
```

VPS 2: Apache2 installation and Hosting, Applying all below commands on VPS2.

Update webservers.

```
#apt update -y && apt upgrade -y
```

- To Add/change Hostname:

```
#hostnamectl set-hostname hostname
```

e.g., #hostname set-hostname Proxy.

- To configure /etc/hosts file

```
#vim /etc/hosts
```

Output:

```
127.0.0.1 localhost
172.31.31.32   proxy …….. (Add Proxy IP and Hostname)
172.31.28.32    ws2 …….. (Add Webserver 2 IP and Hostname)
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
~
```

After changes in /etc/hosts file to apply below any one command to save configuration.

```
#systemctl restart systemd-hostnamed
```

Or

```
#systemctl restart dnsmasq.service
```

Or

```
#/bin/systemctl restart system.hostnamed
```

```
#ping IP Address    ……e.g., #ping 172.31.31.32…. # (Proxy)
```

```
#ping IP Address    ……e.g., #ping 172.31.28.32 …. # (Webserver2)
```

Or

```
#Ping Hostname ……e.g., #ping ws1
```

```
#ping Hostname ……e.g., #ping ws2
```

This command is used to install the Apache web server on a Debian-based Linux system (like Ubuntu, Debian, etc.).

```
#apt install apache2 -y
```

This command ensures that the Apache web server starts automatically when your system boots up.

```
#systemctl enable apache2
```

This command initiates the Apache web server on your system.

```
#systemctl start apache2
```

```
#echo "Welcome from Webserver1 server running fine" | sudo tee /var/www/html/index.html
```

Command Explanation:

1. This is a comment symbol in Bash, indicating the line is for human readability and not executed by the system.
2. echo "Welcome from Webserver1 server running fine": This part creates a text string containing the message "Welcome From Webserver1 server running fine".
3. |: This is a pipe character, which sends the output of the previous command (the text string) to the next command. sudo: This command executes the following command with superuser privileges.
4. tee /var/www/html/index.html: This command writes the input (the text string) to the specified file /var/www/html/index.html and also outputs it to the terminal.

output:

"Welcome From Webserver1 server running fine".

VPS 3: Apache2 installation and Hosting, Applying all below commands on VPS3.

Update webservers.

```
#apt update -y && apt upgrade -y
```

- To Add/change Hostname:

```
#hostnamectl set-hostname hostname
```

e.g., #hostname set-hostname Proxy

- To configure /etc/hosts file

```
#vim /etc/hosts
```

Output:

```
127.0.0.1 localhost
172.31.24.203   ws1 ……. (Add Webserver 1 IP and Hostname)
172.31.31.32   proxy ……. (Add Proxy IP and Hostname)
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
~
```

After changes in /etc/hosts file to apply below any one command to save configuration.

```
#systemctl restart systemd-hostnamed
```

Or

```
#systemctl restart dnsmasq.service
```

Or

```
#/bin/systemctl restart system.hostnamed
```

```
#ping IP Address    ……e.g., #ping 172.31.31.32…. # (proxy)
#ping IP Address    ……e.g., #ping 172.31.24.203 …. # (Webserver1)
```

Or

```
#Ping Hostname ……e.g., #ping ws1
#ping Hostname ……e.g., #ping ws2
```

This command is used to install the Apache web server on a Debian-based Linux system (like Ubuntu, Debian, etc.).

```
#apt install apache2 -y
```

This command ensures that the Apache web server starts automatically when your system boots up.

```
#systemctl enable apache2
```

This command initiates the Apache web server on your system.

```
#systemctl start apache2
```

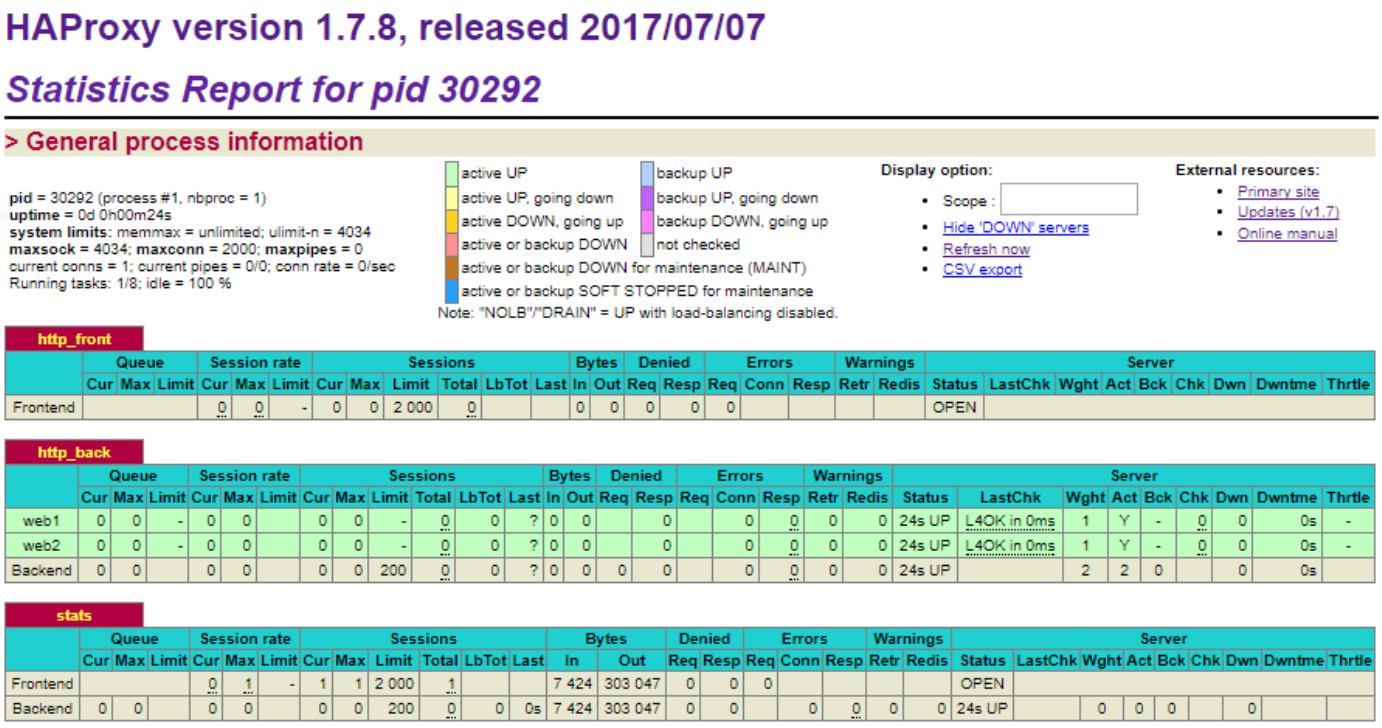
```
#echo "Welcome from Webserver2 server running fine" | sudo tee /var/www/html/index.html
```

- Testing The Setup

With the HAProxy configured and running, open your load balancer server's public IP in a web browser and check that you connect to your backend correctly. The parameter stats uri in the configuration enables the statistics page at the defined address.

```
http://<load balancer public IP>/haproxy?stats
```

When you load the statistics page and all your servers are listed in green, your configuration succeeded!



HAProxy version 1.7.8, released 2017/07/07

Statistics Report for pid 30292

> General process information

pid = 30292 (process #1, nbproc = 1)
uptime = 0d 0h00m24s
system limits: memmax = unlimited; ulimit-n = 4034
maxsock = 4034; **maxconn** = 2000; **maxpipes** = 0
current conns = 1; current pipes = 0/0; conn rate = 0/sec
Running tasks: 1/8; idle = 100 %

active UP
active UP, going down
active DOWN, going up
active or backup DOWN
active or backup DOWN for maintenance (MAINT)
active or backup SOFT STOPPED for maintenance
backup UP
backup UP, going down
backup DOWN, going up
not checked
Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:
- Scope :
- Hide 'DOWN' servers
- Refresh now
- CSV export

External resources:
- Primary site
- Updates (v1.7)
- Online manual

http_front

	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	Dwntme	Thrtle
Frontend				0	0	-	0	0	2 000	0			0	0	0	0	0					OPEN								

http_back

	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	Dwntme	Thrtle
web1	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	24s UP	L4OK in 0ms	1	Y	-	0	0	0s	-
web2	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	24s UP	L4OK in 0ms	1	Y	-	0	0	0s	-
Backend	0	0		0	0		0	0	200	0	0	?	0	0	0	0		0	0	0	0	24s UP		2	2	0		0	0s	

stats

	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	Dwntme	Thrtle
Frontend				0	1	-	1	1	2 000	1			7 424	303 047	0	0	0					OPEN								
Backend	0	0		0	0		0	0	200	0	0	0s	7 424	303 047	0	0		0	0	0	0	24s UP		0	0	0		0		

To check status of Haproxy

```
#sudo systemctl status haproxy
```

&

to Check that your servers are still reporting all green, and then open just the load balancer IP without any port numbers on your web browser.

```
http://<load balancer public IP>/
```

Chrome: Reload the site you get below type of output.

Welcome India from WS1
Welcome India from WS2

Or

#while true; do curl localhost; sleep 1; done
Output:

Welcome India from WS1
Welcome India from WS2
Welcome India from WS1
Welcome India from WS2
Welcome India from WS1
Welcome India from WS2