NGINX Assignment

Title:

Set Up an HA Reverse Proxy with Load Balancing and Web Hosting

---

Overview

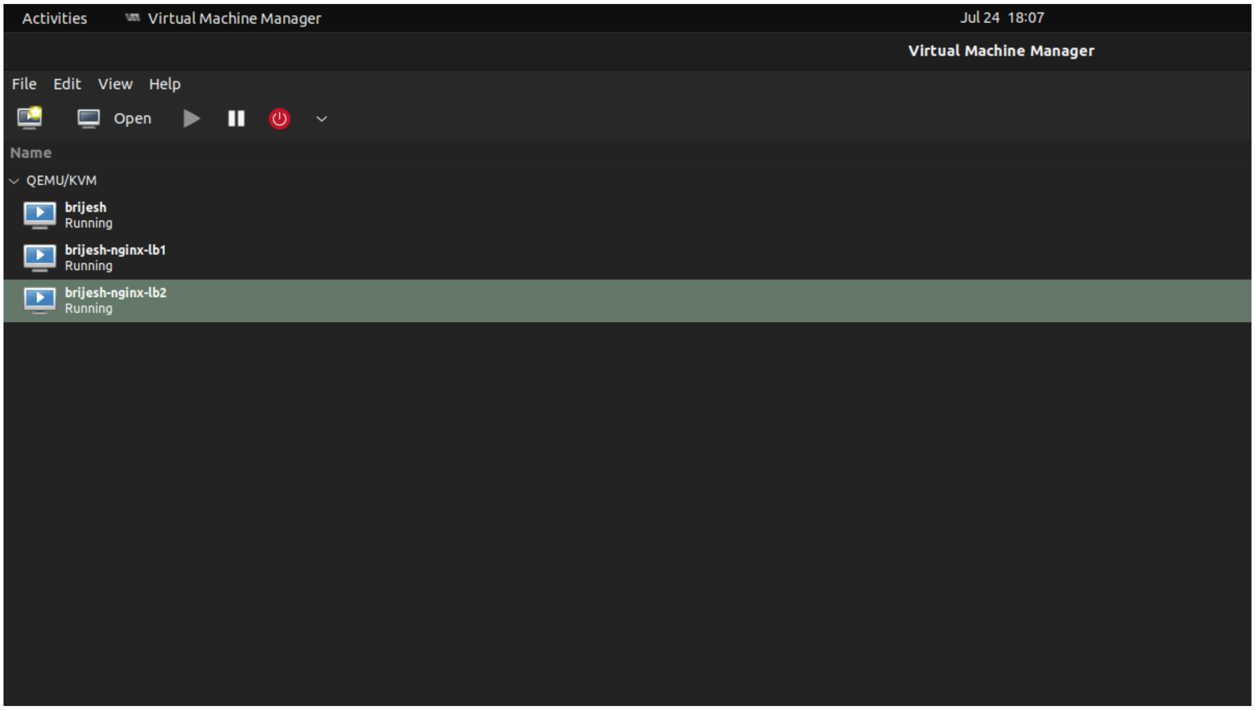
This assignment simulates a production-like High Availability (HA) environment using NGINX and Keepalived. It demonstrates how to:

- Deploy backend web servers serving different content.
- Deploy frontend load balancers using NGINX reverse proxy and round-robin load balancing.
- Implement high availability using Keepalived with VRRP and a floating Virtual IP (VIP).

Assumption: All nodes are on the same subnet and can reach each other.

Node Details

Setup 3 VM with Ubuntu 22.04 lts OS-



brijesh
Running
brijesh-nginx-lb1
Running
brijesh-nginx-lb2
Running

1- 1 vm we will use for 2 web servers
2- 2 vm we use for separate 2 load balancers i.e LB1 and LB2.

Backend Web Servers on VM 1

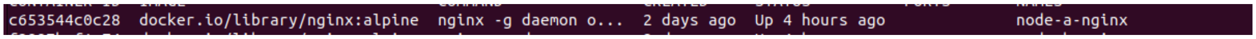
We are using Podman containers to run two separate nginx containers. And using the host network to access it.

1- container command-

```
podman run -d --name node-a-nginx   --network host   -v
~/node-a/html:/usr/share/nginx/html:ro   -v
~/node-a/conf/nginx.conf:/etc/nginx/nginx.conf:ro   docker.io/library/nginx:alpine
```

Web Server 1-
Nginx.conf

```
=====================================================
events {}

http {
   server {
        listen 8081;
        location / {
            root /usr/share/nginx/html;
            index index.html;
        }
    }
}
=====================================================
```



```
c653544c0c28  docker.io/library/nginx:alpine  nginx -g daemon o...  2 days ago  Up 4 hours ago              node-a-nginx
```

2- container command-

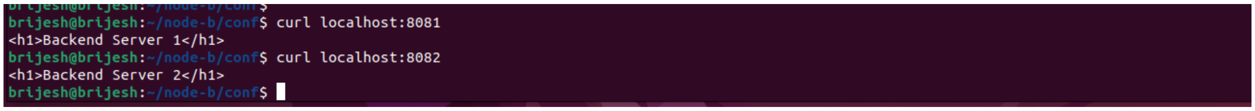
```
podman run -d --name node-a-nginx   --network host   -v
~/node-a/html:/usr/share/nginx/html:ro   -v
~/node-b/conf/nginx.conf:/etc/nginx/nginx.conf:ro   docker.io/library/nginx:alpine
```

Web Server 1-

Nginx.conf

```
===================================================
events {}

http {
    server {
        listen 8082;
        location / {
            root /usr/share/nginx/html;
            index index.html;
        }
    }
}
=====================================================
```

Output of the both web server using curl:-



```
brijesh@brijesh:~/node-b/conf$ curl localhost:8081
<h1>Backend Server 1</h1>
brijesh@brijesh:~/node-b/conf$ curl localhost:8082
<h1>Backend Server 2</h1>
brijesh@brijesh:~/node-b/conf$ 
```

- **Node A (Container)** - IP: 192.168.122.240:8081
  - Serves: "Beckend Server 1"
- **Node B (Container)** - IP: 192.168.122.240:8082
  - Serves: "Beckend Server 2"

Setup Loadbalancer using Nginx-

Load Balancer Nodes (LB Tier)

- **LB1** - IP: 192.168.122.17
- **LB2** - IP: 192.168.122.137
- **VIP** - Floating IP assigned by Keepalived: 192.168.122.100

LB-1 – On Node 1

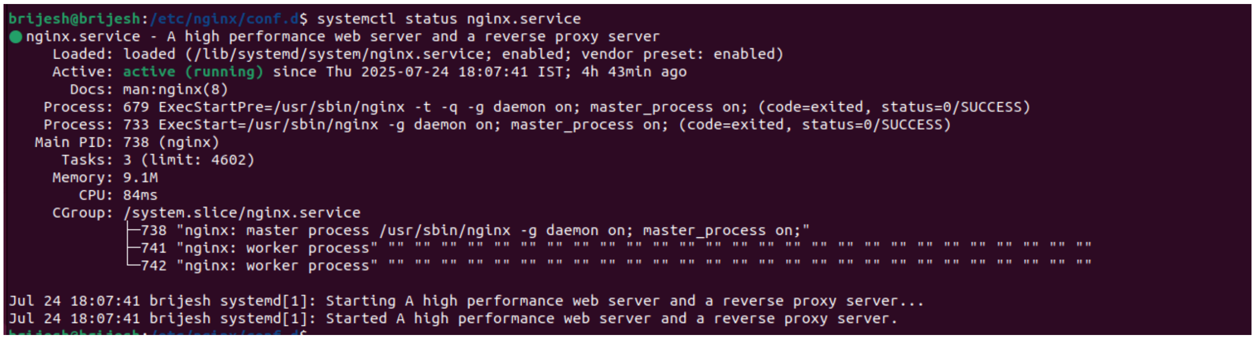
Install Nginx-

```
===========================

sudo apt install -y nginx

============================
```

Default.conf

```
===================================================

upstream backend {

    server 192.168.122.240:8081;

    server 192.168.122.240:8082;

}


server {

    listen 80;
    location / {
        proxy_pass http://backend;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
            proxy_next_upstream error timeout invalid_header http_500 http_502 http_503 http_504;
    }
}
```

==



```
brijesh@brijesh:/etc/nginx/conf.d$ systemctl status nginx.service
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2025-07-24 18:07:41 IST; 4h 43min ago
       Docs: man:nginx(8)
    Process: 679 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 733 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 738 (nginx)
      Tasks: 3 (limit: 4602)
     Memory: 9.1M
        CPU: 84ms
     CGroup: /system.slice/nginx.service
             ├─738 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─741 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
             └─742 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""

Jul 24 18:07:41 brijesh systemd[1]: Starting A high performance web server and a reverse proxy server...
Jul 24 18:07:41 brijesh systemd[1]: Started A high performance web server and a reverse proxy server.
brijesh@brijesh:/etc/nginx/conf.d$
```

LB-2 – On Node 2

Install Nginx-

============================

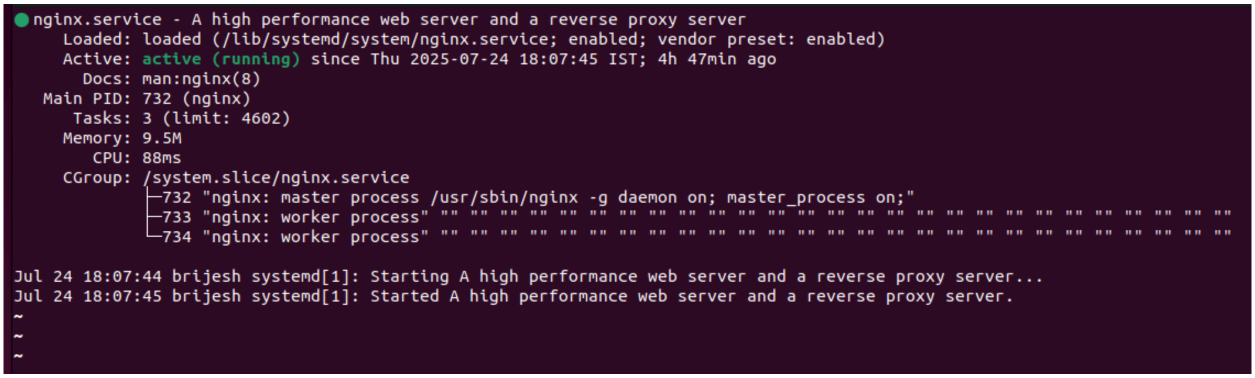
```
sudo apt install -y nginx
```

=============================

Default.conf

==

```
upstream backend {
    server 192.168.122.240:8081;
    server 192.168.122.240:8082;
}

server {
    listen 80;
    location / {
        proxy_pass http://backend;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
            proxy_next_upstream error timeout invalid_header http_500 http_502 http_503 http_504;
    }
}
```



```
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2025-07-24 18:07:45 IST; 4h 47min ago
       Docs: man:nginx(8)
   Main PID: 732 (nginx)
      Tasks: 3 (limit: 4602)
     Memory: 9.5M
        CPU: 88ms
     CGroup: /system.slice/nginx.service
             ├─732 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─733 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
             └─734 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""

Jul 24 18:07:44 brijesh systemd[1]: Starting A high performance web server and a reverse proxy server...
Jul 24 18:07:45 brijesh systemd[1]: Started A high performance web server and a reverse proxy server.
~
~
~
```

Note- Remove LB1 and LB2 nginx static web pages by using below command and then reload Nginx to apply changes-

==

sudo rm /etc/nginx/sites-enabled/default

==

sudo systemctl reload nginx

===

Keepalived Setup on the both LB1 and LB2 Node:-

3. Configure Keepalived (High Availability)

Installation-

===================================

sudo apt install keepalived -y

====================================

LB1 keepalive we use as Master-

LB1 - /etc/keepalived/keepalived.conf

```
==============================================
vrrp_instance VI_1 {
    state MASTER
    interface enp1s0
    virtual_router_id 51
    priority 150
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass secret123
    }
    virtual_ipaddress {
        192.168.122.100
    }
}
===========================================================
sudo systemctl enable --now keepalived
============================================================
```

LB2 we use as Backup-

LB2 - /etc/keepalived/keepalived.conf

===

```
vrrp_instance VI_1 {
    state BACKUP
    interface enp1s0
    virtual_router_id 51
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass boss123
    }
    virtual_ipaddress {
        192.168.122.100
    }
}
```

==

```
sudo systemctl enable --now keepalived
```

===

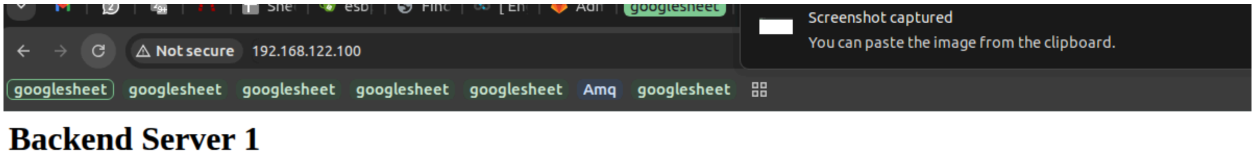
Testing Instructions

1. Test VIP & Load Balancing

```
# From any host or browser:
curl http://192.168.122.100
```

Repeat the curl command multiple times. You should see alternating outputs:

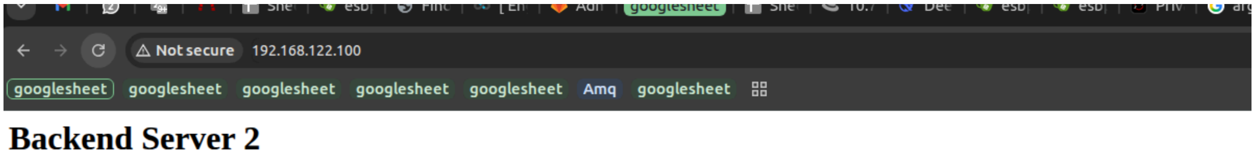
- "Backend Server 1"
- "Backend Server 2"



Screenshot captured
You can paste the image from the clipboard.

Backend Server 1

On Reload-

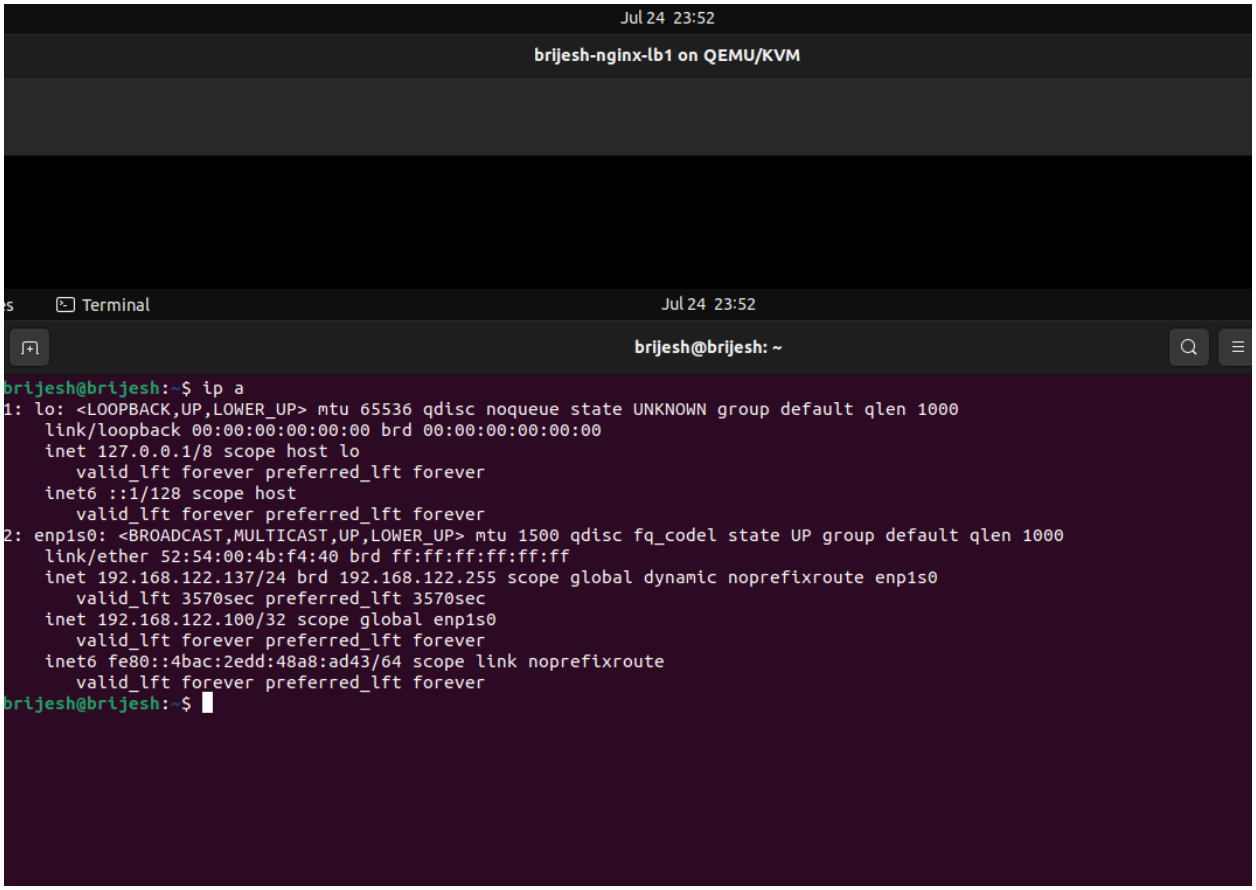


Backend Server 2

2. Simulate Failover

# On LB1:

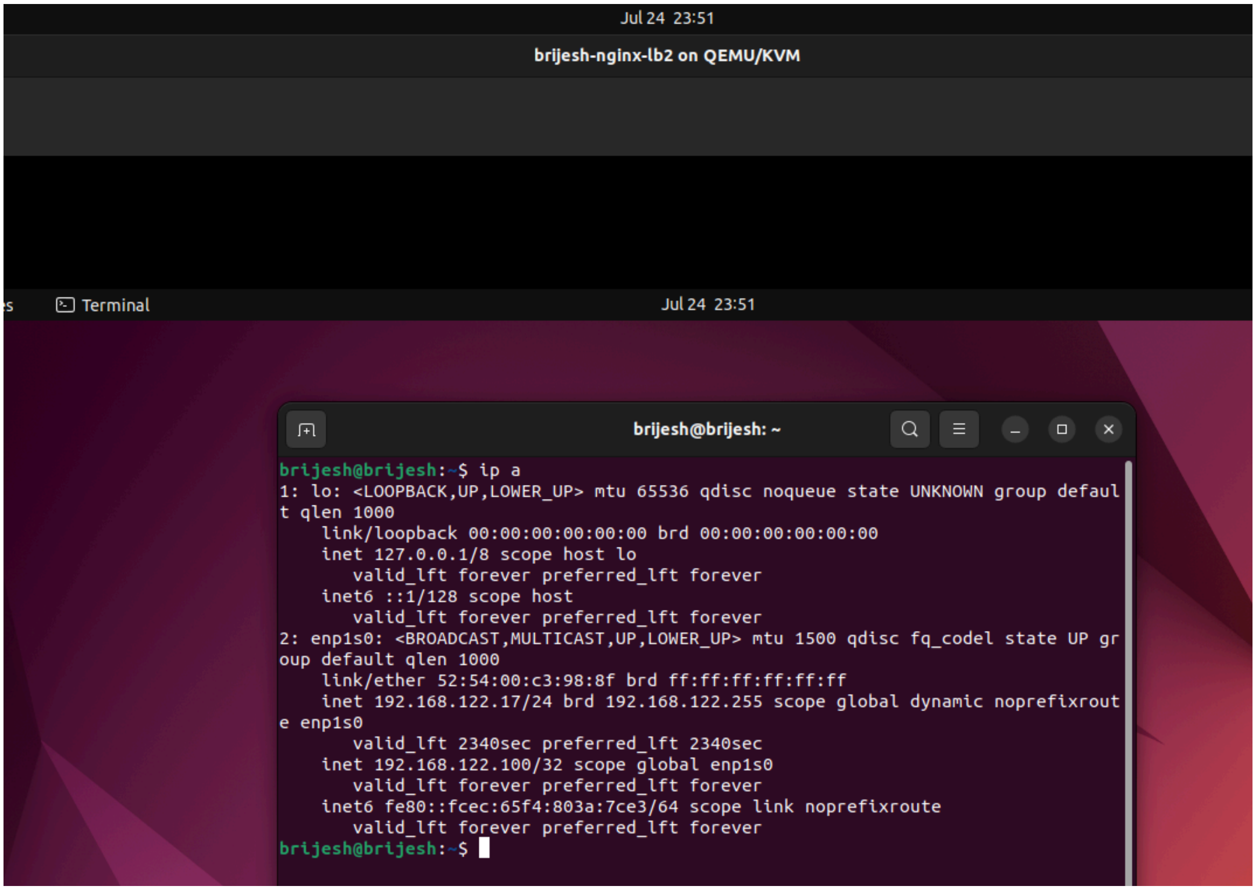
1- ip a



```
brijesh@brijesh:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:4b:f4:40 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.137/24 brd 192.168.122.255 scope global dynamic noprefixroute enp1s0
       valid_lft 3570sec preferred_lft 3570sec
    inet 192.168.122.100/32 scope global enp1s0
       valid_lft forever preferred_lft forever
    inet6 fe80::4bac:2edd:48a8:ad43/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
brijesh@brijesh:~$
```

sudo systemctl stop keepalived

- Run ip a on LB2 to verify VIP 192.168.122.100 has been assigned.



```
brijesh@brijesh:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 52:54:00:c3:98:8f brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.17/24 brd 192.168.122.255 scope global dynamic noprefixrout
e enp1s0
       valid_lft 2340sec preferred_lft 2340sec
    inet 192.168.122.100/32 scope global enp1s0
       valid_lft forever preferred_lft forever
    inet6 fe80::fcec:65f4:803a:7ce3/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
brijesh@brijesh:~$
```

- VIP traffic will now be handled by LB2.

3. Recovery

# On LB1:
sudo systemctl start keepalived

- VIP will automatically fail back to LB1 due to higher priority.

Output

Round-robin Load Balancing

curl http://192.168.122.100
Backend Server 1

curl http://192.168.122.100
Backend server 2

VIP Before & After Failover

- Before failover:
  - LB1 has VIP 192.168.122.100
- After stopping keepalived on LB1:
  - LB2 takes over VIP

Conclusion

This assignment demonstrates a highly available web architecture using:

- NGINX as a reverse proxy and load balancer
- Podman containers as backend nodes
- Keepalived for failover using VRRP and VIP

All components work together to simulate real-world redundancy and load balancing in a multi-tier deployment.