

Below is a compact step-by-step working procedure you can paste into a lab guide. It assumes Apache httpd on the web servers and Nginx as the load balancer.

Network: 192.168.4.0/24

Web1: 192.168.4.191

Web2: 192.168.4.192

LB: 192.168.4.193

NFS: 192.168.4.194

Distro: Rocky Linux 9, SELinux permissive, firewalld disabled (lab only)

1. Common base config (all 4 servers)

As root on each VM:

```
dnf -y update
setenforce 0
sed -i 's/^SELINUX=enforcing/SELINUX=permissive/' /etc/selinux/config

systemctl disable --now firewalld
```

2. NFS server on 192.168.4.194

Install and enable NFS server.

```
dnf -y install nfs-utils
systemctl enable --now nfs-server rpcbind
```

Create export for web content:

```
mkdir -p /srv/www
chown nobody:nobody /srv/www
chmod 0777 /srv/www
```

Export to the two web servers:

```
cat >> /etc/exports << 'EOF'  
/srv/www 192.168.4.191(rw,sync,no_root_squash)  
/srv/www 192.168.4.192(rw,sync,no_root_squash)  
EOF  
  
exportfs -rav
```

Put a simple index page:

```
cat > /srv/www/index.html << 'EOF'  
<!doctype html>  
<html>  
  <head><title>NFS web farm</title></head>  
  <body>  
    <h1>Hello from NFS-backed web farm</h1>  
  </body>  
</html>  
EOF
```

3. Web servers (192.168.4.191 and .192)

3.1 Install Apache httpd

```
dnf -y install httpd  
systemctl enable --now httpd
```

By default, document root is `/var/www/html`.

3.2 Mount NFS share as document root

On each web server:

```
dnf -y install nfs-utils  
mkdir -p /var/www/html  
mount -t nfs 192.168.4.194:/srv/www /var/www/html
```

Make mount persistent:

```
echo '192.168.4.194:/srv/www /var/www/html nfs defaults 0 0' >> /etc/fstab
```

Reload to test:

```
mount -a
systemctl restart httpd
```

Quick check from each web server:

```
curl http://127.0.0.1/
```

You should see the NFS-served HTML.

4. Load balancer server (192.168.4.193) with Nginx

4.1 Install Nginx

```
dnf -y install nginx
systemctl enable --now nginx
```

4.2 Configure HTTP load balancing

Edit `/etc/nginx/nginx.conf` and replace the default `http { ... }` block with e.g.:

```
http {
    upstream webfarm {
        server 192.168.4.191:80;
        server 192.168.4.192:80;
    }

    server {
        listen 80;
        server_name _;

        location / {
            proxy_pass http://webfarm;
            proxy_set_header Host $host;
        }
    }
}
```

```
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}
}
```

Nginx uses round-robin automatically when multiple servers are listed in an `upstream` block.^{[5][6]}

Reload Nginx:

```
nginx -t
systemctl reload nginx
```

5. Test the full chain

From a client machine on the same network:

```
curl http://192.168.4.193/
```

You should see the NFS-hosted page. Refresh multiple times; you can verify load balancing by temporarily placing different test files on each backend (for example, `hostname` output inside the HTML) while still using the shared NFS path, or by checking Apache access logs on both web servers (`/var/log/httpd/access_log`).