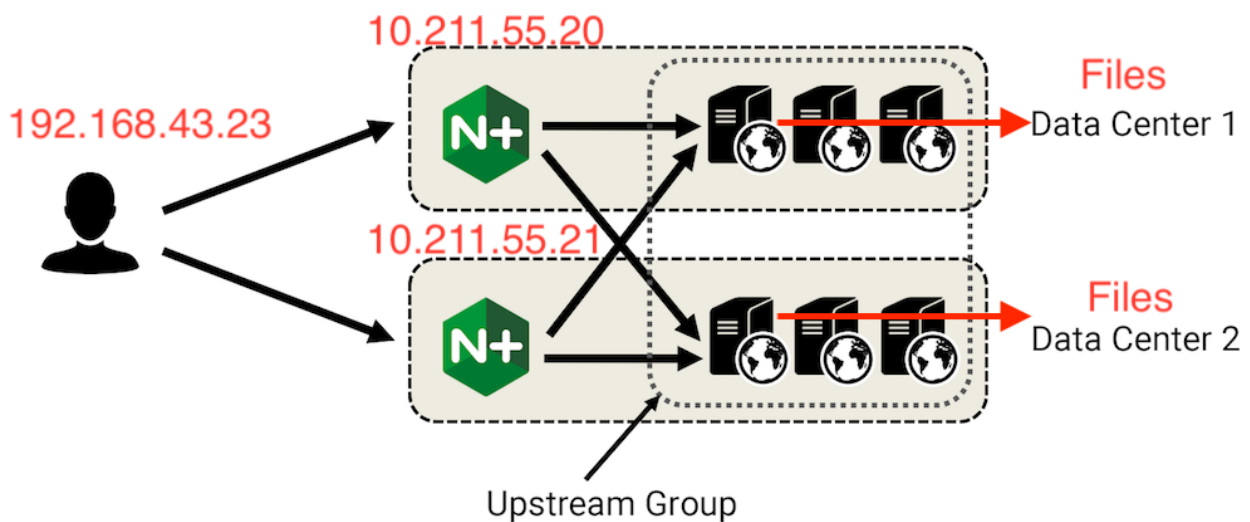


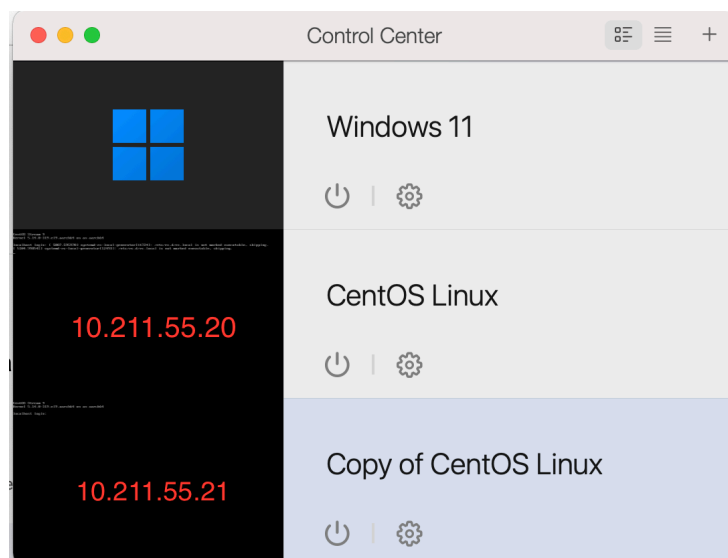
Understood. To follow the provided instructions and create the files/directory using the same name and case as provided in the task steps, please provide me with the specific names and case instructions for the files/directory you want to create.

Load Balancer using Nginx

Agenda



Step 1) Two machine



Step 2) Configure the base Nginx server Note: We can use port else we can leave

Nginx on macOS:

1. Install Nginx:

```
brew install nginx
```

📄 > \$ _

2. Start Nginx:

```
brew services start nginx
```

📄 > \$ _

3. Stop Nginx:

```
brew services stop nginx
```

📄 > \$ _

4. Restart Nginx:

```
brew services restart nginx
```

📄 > \$ _

5. Check the status of Nginx:

```
brew services list
```

📄 > \$ _

6. Uninstall Nginx:

```
brew uninstall nginx
```

📄 > \$ _

These commands allow you to install, start, stop, restart, check the status, and uninstall Nginx using Homebrew on macOS.

```
/opt/homebrew/etc/nginx git:(stable) (8.566s)
brew info nginx
==> nginx: stable 1.25.2 (bottled), HEAD
HTTP(S) server and reverse proxy, and IMAP/POP3 proxy server
https://nginx.org/
/opt/homebrew/Cellar/nginx/1.25.2 (26 files, 2.4MB) *
  Poured from bottle using the formulae.brew.sh API on 2023-09-11 at 16:33:43
From: https://github.com/Homebrew/homebrew-core/blob/HEAD/Formula/n/nginx.rb
License: BSD-2-Clause
==> Dependencies
Required: openssl@3 ✓, pcre2 ✓
==> Options
--HEAD
    Install HEAD version
==> Caveats
Docroot is: /opt/homebrew/var/www

The default port has been set in /opt/homebrew/etc/nginx/nginx.conf to 8080 so that
nginx can run without sudo.

nginx will load all files in /opt/homebrew/etc/nginx/servers/.

To restart nginx after an upgrade:
  brew services restart nginx
Or, if you don't want/need a background service you can just run:
  /opt/homebrew/opt/nginx/bin/nginx -g daemon\ off\;
```

```

http {
    include      mime.types;
    default_type application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #              '$status $body_bytes_sent "$http_referer" '
    #              '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    sendfile      on;
    #tcp_nopush   on;

    #keepalive_timeout  0;
    keepalive_timeout  65;

    #gzip  on;
    upstream myapp {
        server 10.211.55.20;
        server 10.211.55.21;
    # Add more servers if needed
    }

    server {
        listen 8080;
        server_name localhost; # Replace with your actual domain name

        location / {
            proxy_pass http://myapp;
            #proxy_set_header Host $host;
            #proxy_set_header X-Real-IP $remote_addr;
        }
    }
}

```

Step 3) Configure Both the above mentioned machines (in criss-cross manner).

Make sure that you have configured to each machine.

```

lusy@localhost.localdomain /usr/share/nginx/static (0.017s)
cd /etc/nginx/

lusy@localhost.localdomain ~ (0.02s)
ls
conf.d      fastcgi.conf.default  koi-utf  mime.types.default  scgi_params
default.d   fastcgi_params        koi-win  nginx.conf           scgi_params.default
fastcgi.conf  fastcgi_params.default  mime.types  nginx.conf.default  uwsgi_params

lusy@localhost.localdomain /etc/nginx (0.017s)
cd conf.d/

lusy@localhost.localdomain /etc/nginx (0.019s)
cat default.nginx.conf
upstream myapp{

    server 10.211.55.20;

}

server {
    listen 80; # Listen on port 80 (HTTP)

    server_name 10.211.55.21; # Replace with your domain name

    # Location block for the root directory
    location / {
        root /usr/share/nginx/static; # Replace with the actual path to your web files
        index index.html index.htm;
        proxy_pass http://myapp;
    }
}

```

Step 4) Test it. (Hosting file)

```
lusy@localhost.localdomain:/etc/nginx/conf.d (0.018s)
cd /usr/share/nginx/static

lusy@localhost.localdomain /etc/nginx/conf.d (0.02s)
ls
index.html

lusy@localhost.localdomain /usr/share/nginx/static (0.018s)
cat index.html
<!DOCTYPE html>
<html>
<head>
  <title>Load Balancer</title>
</head>
<body>
  <h1>Load Balancer</h1>

  <h2>Servers</h2>
  <p>Below are the IP addresses of two backend servers:</p>

  <ul>
    <li>Server 1: <span id="server1-ip"></span></li>
    <li>Server 2: <span id="server2-ip"></span></li>
  </ul>

  <script type="text/javascript">
    // Replace these placeholder values with the actual server IPs
    var server1IP = "OFF";
    var server2IP = "10.211.55.21";

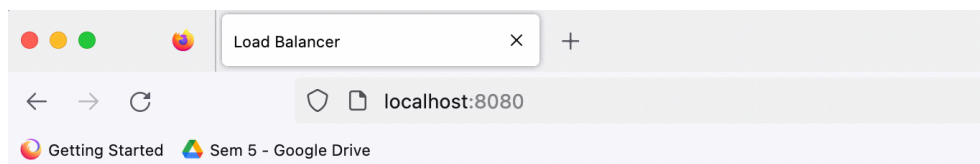
    // Set the server IP addresses in the HTML
    document.getElementById("server1-ip").textContent = server1IP;
    document.getElementById("server2-ip").textContent = server2IP;
  </script>
</body>
</html>

lusy@localhost.localdomain /usr/share/nginx/static
```

To each machine I have changed to verify later

Text

Step 4) Test it. (Acting like Load balancer)

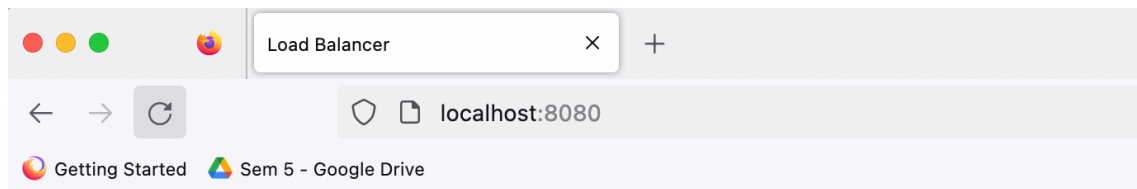


Load Balancer

Servers

Below are the IP addresses of two backend servers:

- Server 1: OFF
- Server 2: 10.211.55.21



Load Balancer

Servers

Below are the IP addresses of two backend servers:

- Server 1: 10.211.55.20
- Server 2: OFF