



# Guião para Configuração do Nginx

## Instalação

```
sudo apt update
sudo apt install nginx -y
sudo systemctl start nginx
sudo systemctl enable nginx
```

Depois de fazerem estes comandos vão ao browser e coloquem no url: `http://localhost` .

## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

Muito provavelmente não têm a firewall ligada mas para assegurar que corre tudo bem:

```
# Firewall
sudo ufw allow 'Nginx Full'
sudo ufw reload
```

```
caeta@ubuntu-vm:~$ php -v
PHP 8.3.6 (cli) (built: Sep 30 2024 15:17:17) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.6, Copyright (c) Zend Technologies
    with Zend OPcache v8.3.6, Copyright (c), by Zend Technologies
```

## Configuração para interação HTML-PHP

```
# Para instalar o php
sudo apt install php-fpm
sudo nano /etc/nginx/sites-available/default

# Adicionar index.php na linha do index
index index.php;

# Já existe por default
location / {
    try_files $uri $uri/ =404;
}

# Adicionar esta secção
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    fastcgi_pass unix:/var/run/php/php8.3-fpm.sock;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    include fastcgi_params;
}
```

```
# Para verificar a syntax
sudo nginx -t

# Dar restart ao nginx
sudo systemctl reload nginx
```

```
cd /var/www/html/
nano index.php
rm index.html # Para não haver dúvidas que o index.php é o main index
```

```
<?php
// Criar uma cookie com o nome "user" com o valor "SR2"
setcookie("user", "SR2", time() + 3600, "/"); // Expira em 3600 segundos/1 hora
?>
<!DOCTYPE html>
<html>
<head>
    <title>Projeto A</title>
```

```
</head>
<body>
  <h1>[SR2]: Projeto A, exemplo de ficheiro html+php</h1>
  <p>Data e Tempo atuais para verificar o funcionamento do php: <?php
echo date('Y-m-d H:i:s'); ?></p>
</body>
</html>
```

## Criação de um website seguro com HTTPS

Para obtermos um website seguro com https iremos usar SSL.

```
sudo nano /etc/nginx/nginx.conf
```

Dentro do ficheiro de configuração do nginx não vão ter este bloco do server. Vão ter copiar este bloco de código para dentro do http.

```
http {
    (...)

    server {
        listen 443 ssl;
        server_name localhost; # or use your local IP (127.0.
0.1)

        ssl_certificate /etc/ssl/certs/selfsigned.crt;
        ssl_certificate_key /etc/ssl/private/selfsigned.key;

        root /var/www/html;
        index index.php;

        location / {
            try_files $uri $uri/ =404;
        }

        location ~ /\.php$ {
            include snippets/fastcgi-php.conf;
            fastcgi_pass unix:/var/run/php/php7.4-fpm.sock; #
Adjust PHP version if needed
            fastcgi_param SCRIPT_FILENAME $document_root$fastc
gi_script_name;
```

```

        include fastcgi_params;
    }
}

(...)
}

```

```

# Para verificar a syntax
sudo nginx -t

# Dar restart ao nginx
sudo systemctl reload nginx

```

## Logs

```
sudo nano /etc/nginx/nginx.conf
```

```

http {
    # Logs
    access_log /var/log/nginx/access.log;

    # Limite de conexões simultâneas
    limit_conn_zone $binary_remote_addr zone=conn_limit:10m;

    # Rate limiting 5r/s
    limit_req_zone $binary_remote_addr zone=req_limit:10m rate=5r/
s;

    server {
        # Depois do ssl_certificate

        # Limit the number of concurrent connections per IP
        limit_conn conn_limit 1;

        # Limitar o número de pedidos por ip a 5 com burst ativa
do
        limit_req zone=req_limit burst=5 nodelay;

        # Timeout da conexão: 65 Seconds
        keepalive_timeout 65;

    }
}

```

```
(...)  
  
}
```

Para verificar que os logs estão a funcionar utilizei uma tool chamada siege que envia vários pedidos em simultâneo, neste caso 10.

```
siege -c10 -r1 https://localhost/  
  
# ver os logs de acesso  
sudo tail -f /var/log/nginx/access.log  
  
# ver os logs de erro  
sudo tail -f /var/log/nginx/error.log
```

O output mostra 2 pedidos bem sucedidos(1: Primeiro pedido, 2: Pedido que enche o buffer de burst para bloquear os próximos 8 pedidos) dos 10 enviados. Isto significa que os logs estão a funcionar corretamente.

```
caeta@ubuntu-vm:~$ siege -c10 -r1 https://localhost/  
New configuration template added to /home/caeta/.siege  
Run siege -C to view the current settings in that file  
  
{  
    "transactions": 2,  
    "availability": 20.00,  
    "elapsed_time": 0.04,  
    "data_transferred": 0.00,  
    "response_time": 0.13,  
    "transaction_rate": 50.00,  
    "throughput": 0.05,  
    "concurrency": 6.50,  
    "successful_transactions": 2,  
    "failed_transactions": 8,  
    "longest_transaction": 0.03,  
    "shortest_transaction": 0.01  
}
```

```

2024/12/02 17:46:50 [error] 3736#3736: *10 limiting connections by zone "conn_limit", client: 127.0.0.1, server: localhost, request: "GET / HTTP/1.1", host: "localhost"
2024/12/02 17:46:50 [error] 3736#3736: *11 limiting connections by zone "conn_limit", client: 127.0.0.1, server: localhost, request: "GET / HTTP/1.1", host: "localhost"
2024/12/02 17:46:50 [error] 3736#3736: *12 limiting connections by zone "conn_limit", client: 127.0.0.1, server: localhost, request: "GET / HTTP/1.1", host: "localhost"
2024/12/02 17:46:50 [error] 3737#3737: *10 limiting connections by zone "conn_limit", client: 127.0.0.1, server: localhost, request: "GET / HTTP/1.1", host: "localhost"
2024/12/02 17:46:50 [error] 3737#3737: *17 limiting requests, excess: 5.860 by zone "req_limit", client: 127.0.0.1, server: localhost, request: "GET / HTTP/1.1", host: "localhost"
2024/12/02 17:46:50 [error] 3737#3737: *16 limiting requests, excess: 5.860 by zone "req_limit", client: 127.0.0.1, server: localhost, request: "GET / HTTP/1.1", host: "localhost"
2024/12/02 17:46:50 [error] 3736#3736: *13 limiting requests, excess: 5.855 by zone "req_limit", client: 127.0.0.1, server: localhost, request: "GET / HTTP/1.1", host: "localhost"
2024/12/02 17:46:50 [error] 3737#3737: *20 limiting requests, excess: 5.850 by zone "req_limit", client: 127.0.0.1, server: localhost, request: "GET / HTTP/1.1", host: "localhost"

```

O output deve ser qualquer coisa semelhante a isto:

```

root@ubuntu-vm:~/nginx/sites-available# sudo tail -f /var/log/nginx/access.log
127.0.0.1 - - [02/Dec/2024:11:27:40 +0000] "GET / HTTP/1.1" 200 210 "-" Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:129.0) Gecko/20100101 Firefox/129.0"
127.0.0.1 - - [02/Dec/2024:11:27:50 +0000] "GET / HTTP/1.1" 200 210 "-" Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:129.0) Gecko/20100101 Firefox/129.0"
127.0.0.1 - - [02/Dec/2024:11:27:51 +0000] "GET / HTTP/1.1" 200 208 "-" Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:129.0) Gecko/20100101 Firefox/129.0"
127.0.0.1 - - [02/Dec/2024:11:27:51 +0000] "GET / HTTP/1.1" 200 208 "-" Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:129.0) Gecko/20100101 Firefox/129.0"
127.0.0.1 - - [02/Dec/2024:11:27:52 +0000] "GET / HTTP/1.1" 200 208 "-" Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:129.0) Gecko/20100101 Firefox/129.0"
127.0.0.1 - - [02/Dec/2024:11:45:30 +0000] "GET / HTTP/1.1" 200 209 "-" Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:129.0) Gecko/20100101 Firefox/129.0"
127.0.0.1 - - [02/Dec/2024:11:45:30 +0000] "GET / HTTP/1.1" 200 209 "-" Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:129.0) Gecko/20100101 Firefox/129.0"
127.0.0.1 - - [02/Dec/2024:11:45:30 +0000] "GET /favicon.ico HTTP/1.1" 404 134 "https://localhost/" Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:129.0) Gecko/20100101 Firefox/129.0"
127.0.0.1 - - [02/Dec/2024:11:46:42 +0000] "GET / HTTP/1.1" 200 210 "-" Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:129.0) Gecko/20100101 Firefox/129.0"
127.0.0.1 - - [02/Dec/2024:11:46:42 +0000] "GET /favicon.ico HTTP/1.1" 404 134 "https://localhost/" Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:129.0) Gecko/20100101 Firefox/129.0"
127.0.0.1 - - [02/Dec/2024:13:07:49 +0000] "HEAD / HTTP/1.1" 200 0 "-" curl/8.5.0"

root@ubuntu-vm:~/nginx/sites-available# sudo tail -f /var/log/nginx/error.log
2024/12/02 11:14:04 [error] 4561#4561: *6 directory index of "/var/www/html/" is forbidden, client: 127.0.0.1, server: _, request: "GET / HTTP/1.1", host: "localhost"
2024/12/02 11:15:04 [error] 4561#4561: *6 directory index of "/var/www/html/" is forbidden, client: 127.0.0.1, server: _, request: "GET / HTTP/1.1", host: "localhost"
2024/12/02 11:15:21 [error] 4561#4561: *7 directory index of "/var/www/html/" is forbidden, client: 127.0.0.1, server: _, request: "GET / HTTP/1.1", host: "localhost"
2024/12/02 11:15:22 [error] 4561#4561: *6 directory index of "/var/www/html/" is forbidden, client: 127.0.0.1, server: _, request: "GET / HTTP/1.1", host: "localhost"
2024/12/02 11:15:33 [error] 4561#4561: *9 directory index of "/var/www/html/" is forbidden, client: 127.0.0.1, server: _, request: "GET / HTTP/1.1", host: "localhost"
2024/12/02 11:23:38 [crit] 2461#2461: *10 connect() to unix:/var/run/php/php7.4-fpm.sock failed (2: No such file or directory) while connecting to upstream, client: 127.0.0.1, server: _, request: "GET / HTTP/1.1", upstream: "fastcgi://unix:/var/run/php/php7.4-fpm.sock", host: "localhost"
2024/12/02 11:23:32 [crit] 2461#2461: *10 connect() to unix:/var/run/php/php7.4-fpm.sock failed (2: No such file or directory) while connecting to upstream, client: 127.0.0.1, server: _, request: "GET / HTTP/1.1", upstream: "fastcgi://unix:/var/run/php/php7.4-fpm.sock", host: "localhost"
2024/12/02 11:27:25 [crit] 2461#2461: *13 connect() to unix:/var/run/php/php7.4-fpm.sock failed (2: No such file or directory) while connecting to upstream, client: 127.0.0.1, server: _, request: "GET / HTTP/1.1", upstream: "fastcgi://unix:/var/run/php/php7.4-fpm.sock", host: "localhost"
2024/12/02 11:27:26 [crit] 2461#2461: *13 connect() to unix:/var/run/php/php7.4-fpm.sock failed (2: No such file or directory) while connecting to upstream, client: 127.0.0.1, server: _, request: "GET / HTTP/1.1", upstream: "fastcgi://unix:/var/run/php/php7.4-fpm.sock", host: "localhost"

```

## Cookies!

Existem duas formas para criação de cookies em nginx: com Código (PHP, js, etc) ou diretamente no ficheiro de configuração do nginx que editamos anteriormente. Considerarei o método de php mais simples e eficaz. Para tal, temos de editar o ficheiro `index.php`.

```
nano /var/www/html/index.php
```

Adicionem esta secção de php para adicionar uma cookie ao vosso web server. Podem mudar os placeholders para qualquer valor.

```

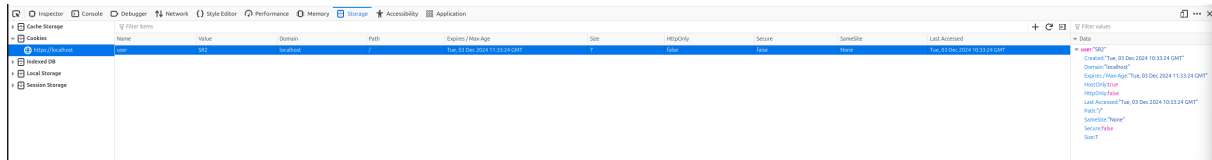
<?php
// Criar uma cookie com o nome "user" com o valor "SR2"
setcookie("user", "SR2", time() + 3600, "/"); // Expira em 3600 segundos/1 hora
?>

<!DOCTYPE html>

<html>
<head>
    <title>Projeto A</title>
</head>
<body>
    <h1>[SR2]: Projeto A, exemplo de ficheiro html+php</h1>
    <p>Data e Tempo atuais para verificar o funcionamento do php: <?php echo date('Y-m-d H:i:s'); ?></p>
</body>
</html>

```

Se formos às developer tools do browser conseguimos verificar que a cookie foi criada. Podemos de seguida remover/adicionar novas cookies também a partir do browser.



## Virtual Directories vs. Physical Paths: Configuração de aliases e caminhos virtuais

Um alias é, resumidamente, uma variável que guarda um caminho. A primeira imagem indica um alias que existe por padrão: root → `/var/www/html`.

```
server {
    listen 443 ssl;

    server_name localhost;

    ssl_certificate /etc/ssl/certs/selfsigned.crt;
    ssl_certificate_key /etc/ssl/private/selfsigned.key;

    # Limit the number of concurrent connections per IP
    limit_conn conn_limit 1;

    # Limit the number of requests per IP (burst allowed)
    limit_req zone=req_limit burst=5 nodelay;

    root /var/www/html;

    # Add index.php to the list if you are using PHP
    index index.php;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
        add_header Set-Cookie "user_session=abc123; HttpOnly; Secure; SameSite=Strict";
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php8.3-fpm.sock; # Update version if needed
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

Escolhi adicionar mais 2 aliases como PoC.

```

server {
    listen 443 ssl;

    server_name localhost;

    ssl_certificate /etc/ssl/certs/selfsigned.crt;
    ssl_certificate_key /etc/ssl/private/selfsigned.key;

    # Limit the number of concurrent connections per IP
    limit_conn conn_limit 1;

    # Limit the number of requests per IP (burst allowed)
    limit_req zone=req_limit burst=5 nodelay;

    root /var/www/html;

    # Add index.php to the list if you are using PHP
    index index.php;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
        add_header Set-Cookie "user_session=abc123; HttpOnly; Secure; SameSite=Strict";
    }

    location /assets/ {
        alias /var/www/static/;
    }

    location /images/ {
        alias /var/www/images/;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php8.3-fpm.sock; # Update version if needed
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}

```

Aliases:

- assets: `/var/www/static/`
- images: `/var/www/images/`

Por default, o utilizador do nginx não tem permissões para aceder a estas diretorias. Temos de dar permissões ao user do nginx para ler os caminhos:

```

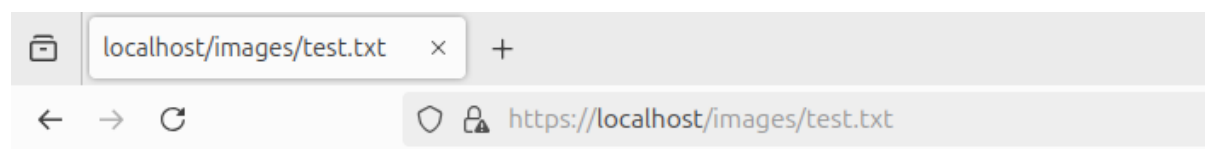
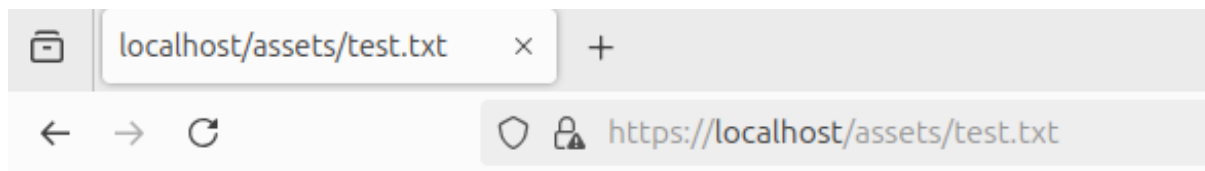
# Dar acesso user do nginx (www-data)
sudo chown -R www-data:www-data /var/www/

# Adicionar conteúdo para depois verificarmos (Não dá para utilizar echo
# diretamente com sudo então utilizei tee)
echo "SR2" | sudo tee /var/www/static/test.txt
echo "Imagem" | sudo tee /var/www/images/test.txt

```



Como podemos verificar, as aliases funcionam perfeitamente



## Fim

---

Obrigado pela vossa atenção.