

MÓDULO 4: SERVIDORES WEB (APACHE E NGINX)

ÍNDICE

Conceitos de Servidores Web

Apache - Instalação e Configuração

Nginx - Instalação e Configuração

Virtual Hosts / Server Blocks

SSL/TLS e HTTPS

1. CONCEITOS DE SERVIDORES WEB

O que é um Servidor Web?

Servidor web é um software que recebe requisições HTTP/HTTPS e retorna conteúdo (HTML, imagens, arquivos) aos clientes (navegadores).

Principais Servidores Web

Apache vs Nginx

Apache (`httpd`):

- Tradicional, maduro
- `.htaccess` dinâmico
- Módulos extensivos
- Melhor para conteúdo dinâmico

Nginx:

- Alto desempenho
 - Baixo consumo de memória
 - Excelente para arquivos estáticos
 - Proxy reverso eficiente
-

2. APACHE - INSTALAÇÃO

Ubuntu/Debian

```
# Instalar Apache
sudo apt update
sudo apt install apache2 -y

# Verificar status
sudo systemctl status apache2

# Verificar versão
apache2 -v

# Testar configuração
sudo apache2ctl configtest
```

CentOS/RHEL

```
sudo yum install httpd -y
sudo systemctl start httpd
sudo systemctl enable httpd
```

Estrutura de Diretórios

```
/etc/apache2/          # Configurações (Debian)
    apache2.conf        # Config principal
    ports.conf          # Portas
    sites-available/    # Sites disponíveis
    sites-enabled/      # Sites ativos
    mods-available/     # Módulos disponíveis
    mods-enabled/       # Módulos ativos

/var/www/html/          # Documentos web
/var/log/apache2/
    access.log          # Logs
    error.log
```

CentOS/RHEL:

```
/etc/httpd/conf/httpd.conf  # Config principal
/var/www/html/                # Documentos
/var/log/httpd/                # Logs
```

3. CONFIGURAÇÃO BÁSICA DO APACHE

Arquivo Principal (apache2.conf)

```
sudo nano /etc/apache2/apache2.conf

# DIRETÓRIO RAIZ
ServerRoot "/etc/apache2"

# TIMEOUT
Timeout 300

# KEEPALIVE
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 5

# MPM (Multi-Processing Module) - Event
<IfModule mpm_event_module>
    StartServers          2
    MinSpareThreads      25
    MaxSpareThreads      75
    ThreadLimit          64
    ThreadsPerChild      25
    MaxRequestWorkers    150
    MaxConnectionsPerChild 0
</IfModule>

# DIRETÓRIO PADRÃO
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>

# SEGURANÇA
ServerTokens Prod
ServerSignature Off
TraceEnable Off

# LOGS
ErrorLog ${APACHE_LOG_DIR}/error.log
LogLevel warn
```

Configuração de Portas

```
sudo nano /etc/apache2/ports.conf
```

```
Listen 80

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

4. VIRTUAL HOSTS (APACHE)

Criar Virtual Host

```
# Criar diretório do site
sudo mkdir -p /var/www/sitel.com.br/public_html

# Criar página de teste
sudo nano /var/www/sitel.com.br/public_html/index.html

<!DOCTYPE html>
<html>
<head>
    <title>Site 1</title>
</head>
<body>
    <h1>Bem-vindo ao Site 1</h1>
    <p>Virtual Host funcionando!</p>
</body>
</html>

# Ajustar permissões
sudo chown -R www-data:www-data /var/www/sitel.com.br
sudo chmod -R 755 /var/www/sitel.com.br
```

Configuração do Virtual Host

```
sudo nano /etc/apache2/sites-available/sitel.com.br.conf

<VirtualHost *:80>
```

```

ServerName sitel.com.br
ServerAlias www.sitel.com.br
ServerAdmin admin@sitel.com.br

DocumentRoot /var/www/sitel.com.br/public_html

<Directory /var/www/sitel.com.br/public_html>
    Options -Indexes +FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>

ErrorLog ${APACHE_LOG_DIR}/sitel-error.log
CustomLog ${APACHE_LOG_DIR}/sitel-access.log combined
</VirtualHost>

```

Ativar Site

```

# Habilitar site
sudo a2ensite sitel.com.br.conf

# Desabilitar site padrão (opcional)
sudo a2dissite 000-default.conf

# Testar configuração
sudo apache2ctl configtest

# Recarregar Apache
sudo systemctl reload apache2

# Testar
curl -H "Host: sitel.com.br" http://localhost

```

5. MÓDULOS DO APACHE

Gerenciar Módulos

```

# Listar módulos habilitados
apache2ctl -M

# Habilitar módulo
sudo a2enmod rewrite
sudo a2enmod ssl
sudo a2enmod headers

# Desabilitar módulo
sudo a2dismod status

```

```

# Módulos importantes
sudo a2enmod rewrite      # URL rewrite
sudo a2enmod ssl           # HTTPS
sudo a2enmod headers       # Headers HTTP
sudo a2enmod proxy         # Proxy
sudo a2enmod proxy_http    # Proxy HTTP
sudo a2enmod deflate        # Compressão

# Recarregar após mudanças
sudo systemctl reload apache2

```

.htaccess - Reescrita de URL

```

sudo nano /var/www/site1.com.br/public_html/.htaccess

# Habilitar Rewrite
RewriteEngine On

# Redirecionar www para não-www
RewriteCond %{HTTP_HOST} ^www\.(.*)$ [NC]
RewriteRule ^(.*)$ http://%1/$1 [R=301,L]

# Forçar HTTPS
RewriteCond %{HTTPS} off
RewriteRule ^(.*)$ https:// %{HTTP_HOST}%{REQUEST_URI} [L,R=301]

# URL amigável - remover .html
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME}.html -f
RewriteRule ^(.*)$ $1.html [L]

# Bloquear acesso a arquivos
<FilesMatch "\.(htaccess|htpasswd|ini|log|sh|sql)$">
    Require all denied
</FilesMatch>

```

6. NGINX - INSTALAÇÃO

Ubuntu/Debian

```

# Instalar Nginx
sudo apt update
sudo apt install nginx -y

# Verificar status

```

```
sudo systemctl status nginx

# Verificar versão
nginx -v

# Testar configuração
sudo nginx -t
```

Estrutura de Diretórios

```
/etc/nginx/
    ■■■ nginx.conf          # Config principal
    ■■■ sites-available/    # Sites disponíveis
    ■■■ sites-enabled/      # Sites ativos
    ■■■ conf.d/              # Configurações adicionais
    ■■■ snippets/            # Trechos reutilizáveis

/var/www/html/                  # Documentos web
/var/log/nginx/
    ■■■ access.log
    ■■■ error.log
```

7. CONFIGURAÇÃO BÁSICA DO NGINX

nginx.conf

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

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;

events {
    worker_connections 768;
    use epoll;
    multi_accept on;
}

http {
    # BÁSICO
    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
```

```

# SEGURANÇA
server_tokens off;
client_max_body_size 20M;

# MIME TYPES
include /etc/nginx/mime.types;
default_type application/octet-stream;

# LOGS
access_log /var/log/nginx/access.log;
error_log /var/log/nginx/error.log;

# COMPRESSÃO GZIP
gzip on;
gzip_vary on;
gzip_proxied any;
gzip_comp_level 6;
gzip_types text/plain text/css text/xml text/javascript
            application/json application/javascript application/xml+rss;

# VIRTUAL HOSTS
include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;
}

```

8. SERVER BLOCKS (NGINX)

Criar Server Block

```

# Criar diretório
sudo mkdir -p /var/www/sitel.com.br/html

# Criar página
sudo nano /var/www/sitel.com.br/html/index.html

```

```

<!DOCTYPE html>
<html>
<head>
    <title>Site 1 - Nginx</title>
</head>
<body>
    <h1>Nginx Server Block Funcionando!</h1>
</body>
</html>

```

```

# Permissões
sudo chown -R www-data:www-data /var/www/sitel.com.br

```

```
sudo chmod -R 755 /var/www/sitel.com.br
```

Configuração do Server Block

```
sudo nano /etc/nginx/sites-available/sitel.com.br
```

```
server {
    listen 80;
    listen [::]:80;

    server_name sitel.com.br www.sitel.com.br;

    root /var/www/sitel.com.br/html;
    index index.html index.htm index.php;

    # LOGS
    access_log /var/log/nginx/sitel-access.log;
    error_log /var/log/nginx/sitel-error.log;

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

    # PHP (se necessário)
    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
    }

    # NEGAR ACESSO A .htaccess
    location ~ /\.ht {
        deny all;
    }

    # CACHE ESTÁTICO
    location ~* \.(jpg|jpeg|png|gif|ico|css|js)$ {
        expires 365d;
        add_header Cache-Control "public, immutable";
    }
}
```

Ativar Server Block

```
# Criar link simbólico
sudo ln -s /etc/nginx/sites-available/sitel.com.br /etc/nginx/sites-enabled/

# Testar configuração
sudo nginx -t

# Recarregar Nginx
```

```
sudo systemctl reload nginx

# Testar
curl http://localhost
```

9. SSL/TLS - HTTPS

Certificado Let's Encrypt (Gratuito)

```
# Instalar Certbot
sudo apt install certbot python3-certbot-apache -y # Apache
sudo apt install certbot python3-certbot-nginx -y # Nginx

# Obter certificado - Apache
sudo certbot --apache -d sitel.com.br -d www.sitel.com.br

# Obter certificado - Nginx
sudo certbot --nginx -d sitel.com.br -d www.sitel.com.br

# Renovação automática (já configurada)
sudo certbot renew --dry-run

# Listar certificados
sudo certbot certificates
```

SSL Manual (Certificado Próprio)

```
# Gerar certificado autoassinado
sudo mkdir /etc/ssl/private
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
    -keyout /etc/ssl/private/sitel.key \
    -out /etc/ssl/certs/sitel.crt
```

Virtual Host Apache com SSL

```
<VirtualHost *:443>
    ServerName sitel.com.br
    ServerAlias www.sitel.com.br
    DocumentRoot /var/www/sitel.com.br/public_html

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/sitel.crt
    SSLCertificateKeyFile /etc/ssl/private/sitel.key
```

```

# SEGURANÇA SSL
SSLProtocol all -SSLv3 -TLSv1 -TLSv1.1
SSLCipherSuite ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256
SSLHonorCipherOrder off

# HSTS
Header always set Strict-Transport-Security "max-age=31536000"

ErrorLog ${APACHE_LOG_DIR}/sitel-ssl-error.log
CustomLog ${APACHE_LOG_DIR}/sitel-ssl-access.log combined
</VirtualHost>

```

Server Block Nginx com SSL

```

server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;

    server_name sitel.com.br www.sitel.com.br;

    # CERTIFICADOS SSL
    ssl_certificate /etc/ssl/certs/sitel.crt;
    ssl_certificate_key /etc/ssl/private/sitel.key;

    # CONFIGURAÇÃO SSL
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384;
    ssl_prefer_server_ciphers off;

    # SSL SESSION
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 10m;

    # HSTS
    add_header Strict-Transport-Security "max-age=31536000" always;

    root /var/www/sitel.com.br/html;
    index index.html index.php;

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

# Redirecionar HTTP para HTTPS
server {
    listen 80;
    server_name sitel.com.br www.sitel.com.br;
    return 301 https://$server_name$request_uri;
}

```

10. PROXY REVERSO (NGINX)

Nginx como Proxy para Apache

```
server {
    listen 80;
    server_name sitel.com.br;

    location / {
        proxy_pass http://127.0.0.1:8080;
        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;
    }

    # Cache estático no Nginx
    location ~* \.(jpg|jpeg|png|gif|ico|css|js)$ {
        root /var/www/sitel.com.br/html;
        expires 365d;
    }
}

# Alterar Apache para porta 8080
sudo nano /etc/apache2/ports.conf
# Mudar: Listen 8080

sudo nano /etc/apache2/sites-available/sitel.com.br.conf
# Mudar: <VirtualHost *:8080>

sudo systemctl restart apache2
sudo systemctl reload nginx
```

11. COMANDOS DE ADMINISTRAÇÃO

Apache

```
# Gerenciamento
sudo systemctl start apache2
sudo systemctl stop apache2
sudo systemctl restart apache2
sudo systemctl reload apache2
sudo systemctl status apache2

# Configuração
```

```

sudo apache2ctl configtest      # Testar config
sudo apache2ctl -M              # Listar módulos
sudo apache2ctl -S              # Listar virtual hosts

# Sites
sudo a2ensite site.conf        # Habilitar site
sudo a2dissite site.conf        # Desabilitar site

# Módulos
sudo a2enmod rewrite          # Habilitar módulo
sudo a2dismod status           # Desabilitar módulo

# Logs
sudo tail -f /var/log/apache2/access.log
sudo tail -f /var/log/apache2/error.log

```

Nginx

```

# Gerenciamento
sudo systemctl start nginx
sudo systemctl stop nginx
sudo systemctl restart nginx
sudo systemctl reload nginx
sudo systemctl status nginx

# Configuração
sudo nginx -t                      # Testar config
sudo nginx -T                      # Mostrar config completa
sudo nginx -s reload                # Reload graceful

# Logs
sudo tail -f /var/log/nginx/access.log
sudo tail -f /var/log/nginx/error.log

# Processos
ps aux | grep nginx

```

12. SEGURANÇA

Headers de Segurança

Apache:

```

# Habilitar mod_headers
sudo a2enmod headers

# Adicionar no VirtualHost
Header always set X-Frame-Options "SAMEORIGIN"

```

```
Header always set X-Content-Type-Options "nosniff"
Header always set X-XSS-Protection "1; mode=block"
Header always set Referrer-Policy "no-referrer-when-downgrade"
Header always set Content-Security-Policy "default-src 'self'"
```

Nginx:

```
add_header X-Frame-Options "SAMEORIGIN" always;
add_header X-Content-Type-Options "nosniff" always;
add_header X-XSS-Protection "1; mode=block" always;
add_header Referrer-Policy "no-referrer-when-downgrade" always;
add_header Content-Security-Policy "default-src 'self'" always;
```

Proteção contra DDoS

Nginx - Rate Limiting:

```
http {
    limit_req_zone $binary_remote_addr zone=mylimit:10m rate=10r/s;

    server {
        location / {
            limit_req zone=mylimit burst=20 nodelay;
        }
    }
}
```

13. MONITORAMENTO

Apache Status

```
# Habilitar mod_status
sudo a2enmod status

# Configurar
sudo nano /etc/apache2/mods-enabled/status.conf
```

```
<Location /server-status>
    SetHandler server-status
    Require local
    Require ip 192.168.1.0/24
</Location>
```

```
# Acessar
```

```
curl http://localhost/server-status
```

Nginx Status

```
server {  
    listen 80;  
    server_name localhost;  
  
    location /nginx_status {  
        stub_status on;  
        access_log off;  
        allow 127.0.0.1;  
        allow 192.168.1.0/24;  
        deny all;  
    }  
}  
  
# Acessar  
curl http://localhost/nginx_status
```

14. EXERCÍCIOS

Apache: Instale e configure 2 virtual hosts

Nginx: Configure 2 server blocks

SSL: Obtenha certificado Let's Encrypt

Proxy: Configure Nginx como proxy para Apache

Segurança: Implemente headers de segurança

RESUMO

Apache:

- Tradicional, robusto
- .htaccess dinâmico
- Virtual Hosts

Nginx:

- Alto desempenho

- Proxy reverso
- Server Blocks

Comandos:

- `a2ensite/a2dissite` (Apache)
- `nginx -t` (Nginx)
- `certbot` (SSL)

Arquivos:

- `/etc/apache2/sites-available/`
 - `/etc/nginx/sites-available/`
-

Próximo: Módulo 5 - QoS e ACLs