

HTTPS en Nginx con Let's Encrypt

Configuraremos variables de entorno

```
export DOMAIN="jaimesimarro.duckdns.org"
export WEBROOT="/var/www/miweb"
```

Instalamos Nginx

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

Creamos el directorio web

```
jaime@server:~$ sudo mkdir -p $WEBROOT
jaime@server:~$ echo "<h1>HTTPS con Let's Encrypt en Nginx – Jaime Simarro</h1>" | sudo tee $WEBROOT/index.html
<h1>HTTPS con Let's Encrypt en Nginx – Jaime Simarro</h1>
jaime@server:~$ cat $WEBROOT/index.html
<h1>HTTPS con Let's Encrypt en Nginx – Jaime Simarro</h1>
jaime@server:~$
```

Creamos la configuración de Nginx

```
GNU nano 7.2                                     /etc/nginx/sites-available/miweb
server {
    listen 80;
    listen [::]:80;

    server_name jaimesimarro.duckdns.org;

    root /var/www/miweb;
    index index.html index.htm;

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

    access_log /var/log/nginx/miweb_access.log;
    error_log /var/log/nginx/miweb_error.log;
}
```

Activamos el sitio

```
sudo ln -s /etc/nginx/sites-available/miweb /etc/nginx/sites-enabled/
sudo systemctl reload nginx
```

Instalamos Certbot y el plugin de DuckDNS

```

jaime@server:~$ sudo snap install --classic certbot
[sudo] password for jaime:
2025-11-14T10:14:26Z INFO Waiting for automatic snapd restart...
certbot 5.1.0 from Certbot Project (certbot-eff✓) installed
jaime@server:~$ sudo ln -sf /snap/bin/certbot /usr/bin/certbot
[sudo] password for jaime:
jaime@server:~$ sudo snap set certbot trust-plugin-with-root=ok
jaime@server:~$ sudo snap install certbot-dns-duckdns
certbot-dns-duckdns v1.7.0 from Marvin Heptner (infinityofspace) installed
jaime@server:~$ |

```

Creamos el archivo con las credenciales de DuckDNS

```

jaime@server:~$ sudo mkdir -p /etc/letsencrypt
jaime@server:~$ echo "dns_duckdns_token=393dc506-d8bb-4f2a-9829-611d94bf99f7" | sudo tee /etc/letsencrypt/duckdns.ini
dns_duckdns_token=393dc506-d8bb-4f2a-9829-611d94bf99f7
jaime@server:~$ sudo chmod 600 /etc/letsencrypt/duckdns.ini
jaime@server:~$ sudo cat /etc/letsencrypt/duckdns.ini
dns_duckdns_token=393dc506-d8bb-4f2a-9829-611d94bf99f7
jaime@server:~$ 

```

Instalamos el certificado

```

sudo ~/.acme.sh/acme.sh --install-cert -d jaimesimarro.duckdns.org \
--key-file /etc/letsencrypt/live/jaimesimarro.duckdns.org/privkey.pem \
--fullchain-file /etc/letsencrypt/live/jaimesimarro.duckdns.org/fullchain.pem \
--reloadcmd "systemctl reload nginx"

```

Creamos el archivo de configuracion HTTPS

```

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

    server_name jaimesimarro.duckdns.org;

    ssl_certificate /etc/letsencrypt/live/jaimesimarro.duckdns.org/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/jaimesimarro.duckdns.org/privkey.pem;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers on;

    root /var/www/miweb;
    index index.html index.htm;

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

    access_log /var/log/nginx/miweb_ssl_access.log;
    error_log /var/log/nginx/miweb_ssl_error.log;
}

```

Activamos el sitio y reiniciamos el nginx

```
jaime@server:/etc/nginx/sites-available$ sudo ln -s /etc/nginx/sites-available/miweb-ssl /etc/nginx/sites-enabled/
jaime@server:/etc/nginx/sites-available$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
jaime@server:/etc/nginx/sites-available$ sudo systemctl reload nginx
Warning: The unit file, source configuration file or drop-ins of nginx.service changed on disk. Run 'systemctl daemon-reload' to reload units.
jaime@server:/etc/nginx/sites-available$ systemctl daemon-reload
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ====
Authentication is required to reload the systemd state.
Authenticating as: jaime
Password:
==== AUTHENTICATION COMPLETE ====

```

Probamos la funcionalidad

