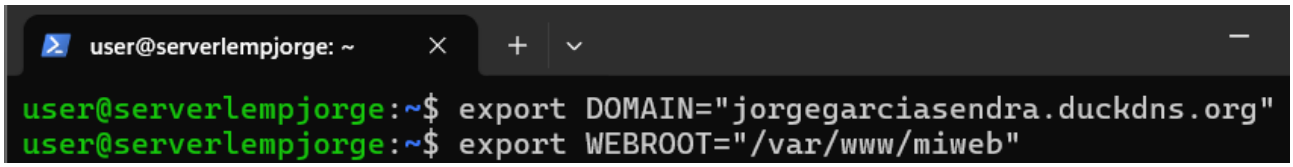
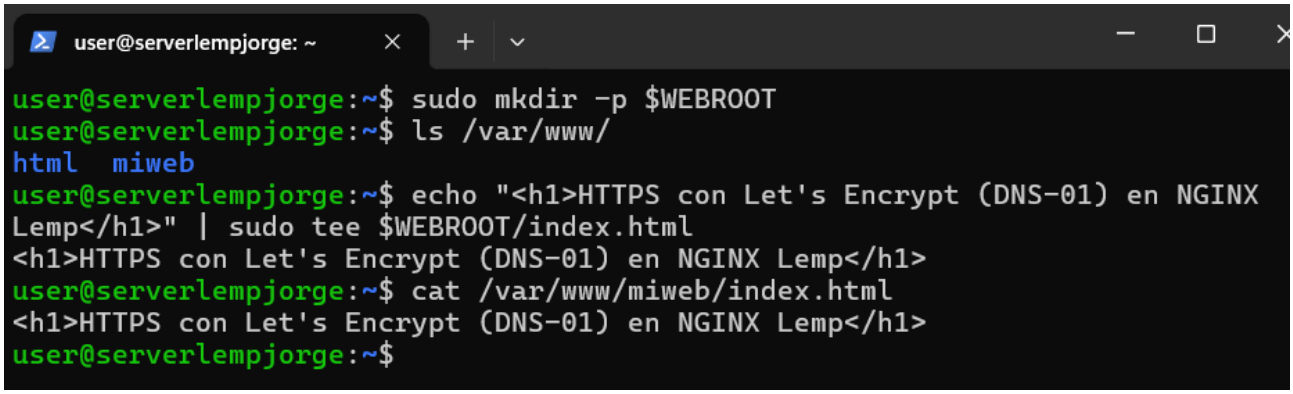
1. Creamos variables de entorno para no tener que escribirlas

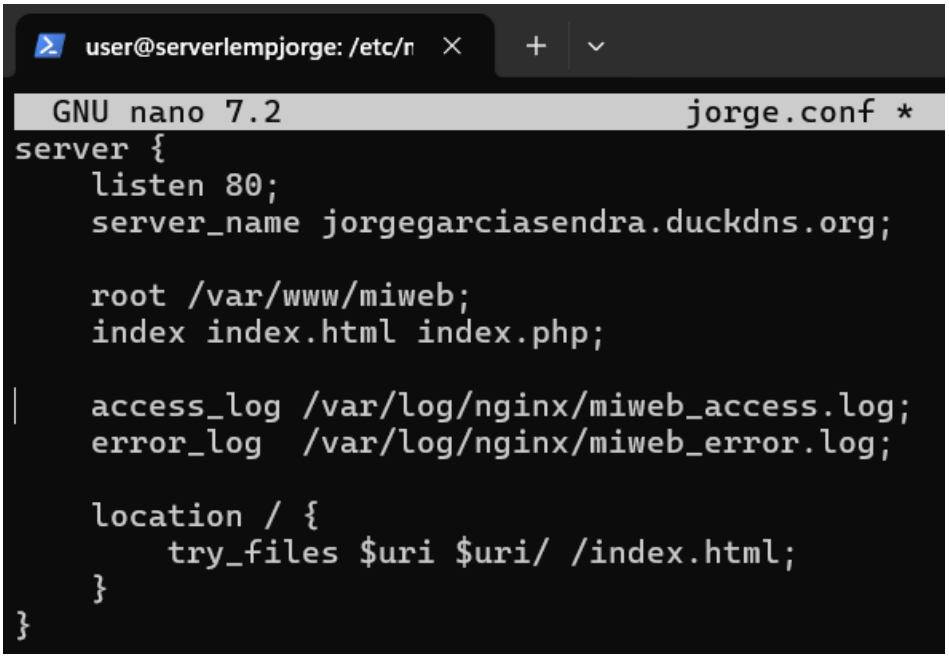


2. Creamos el subdominio en duckDNS

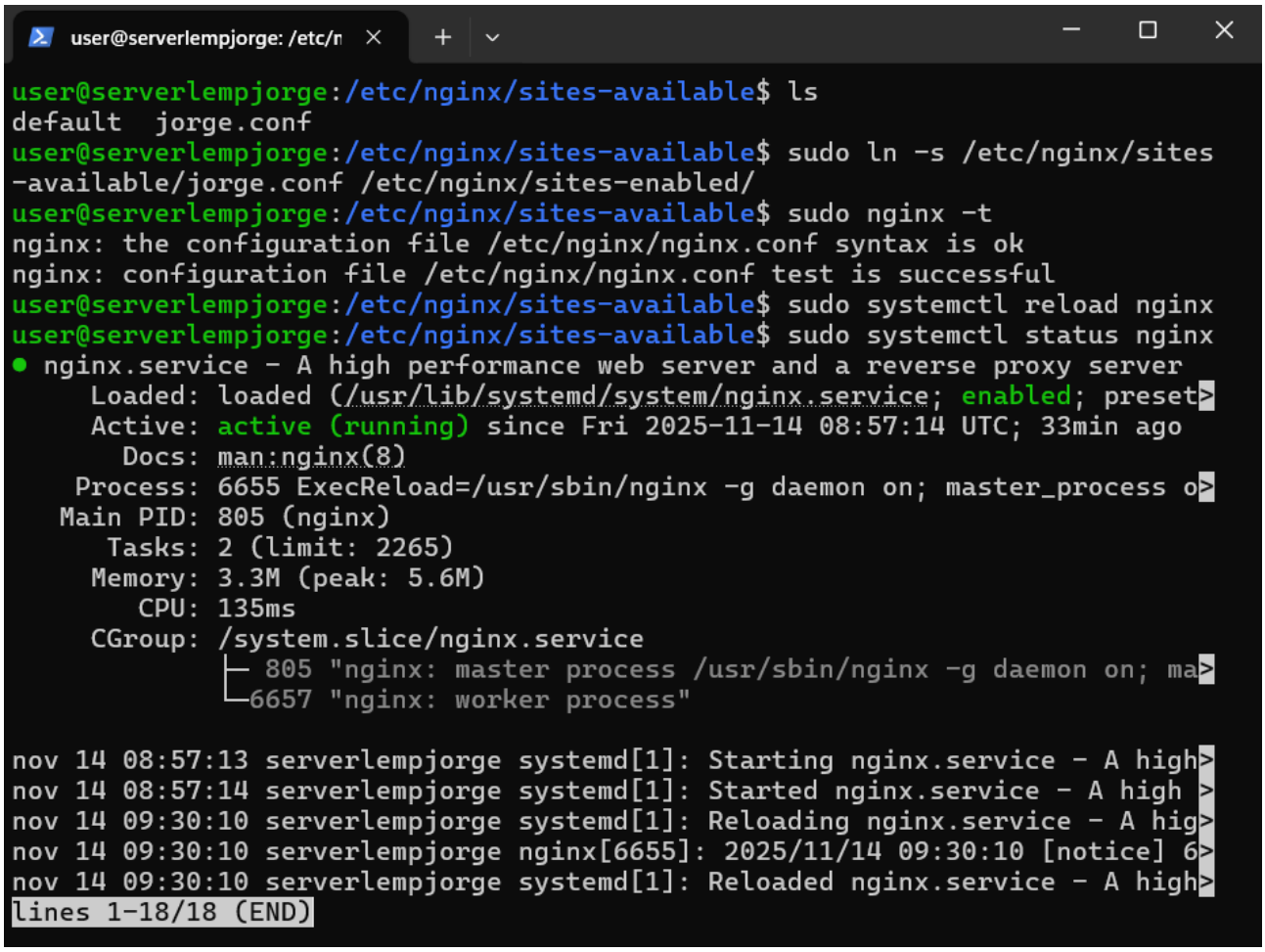
3. Creamos la carpeta donde se aloja la web y el contenido de la pagina principal



4. Creamos el ServerBlock, en servername ponemos el dominio del duckdns

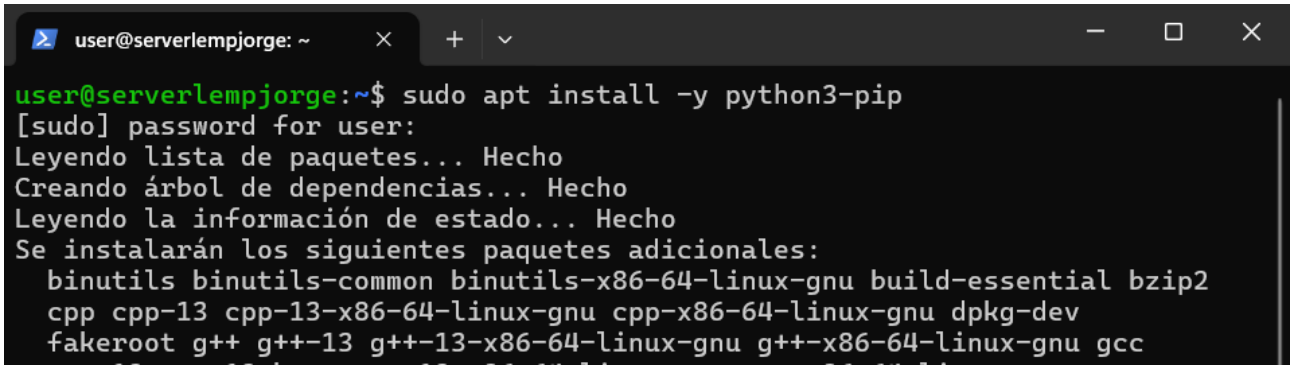


5. Activamos el sitio y recargamos el Nginx



6. Instalamos el pip, y el certbot con la extensión para duck-dns, actualmente es la manera mas facil de hacerlo y la mas rapida.

Instalación pip



Instalacion certbot con extensión para duck dns.

```
user@serverlempjorge:~$ sudo pip3 install certbot-dns-duckdns --break-system
-packages
Collecting certbot-dns-duckdns
  Downloading certbot_dns_duckdns-1.7.0-py3-none-any.whl.metadata (17 kB)
Collecting certbot<6.0,>=1.18.0 (from certbot-dns-duckdns)
  Downloading certbot-5.1.0-py3-none-any.whl.metadata (7.0 kB)
Requirement already satisfied: requests<3.0,>=2.20.0 in /usr/lib/python3/dis
t-packages (from certbot-dns-duckdns) (2.31.0)
Collecting dnspython<3.0,>=2.0.0 (from certbot-dns-duckdns)
  Downloading dnspython-2.8.0-py3-none-any.whl.metadata (5.7 kB)
```

7. Creación de certificados para securizar la pagina web

Creamos la carpeta donde se guardará toda la información de los certificados

```
user@serverlempjorge:~$ sudo mkdir -p /etc/letsencrypt
user@serverlempjorge:~$
```

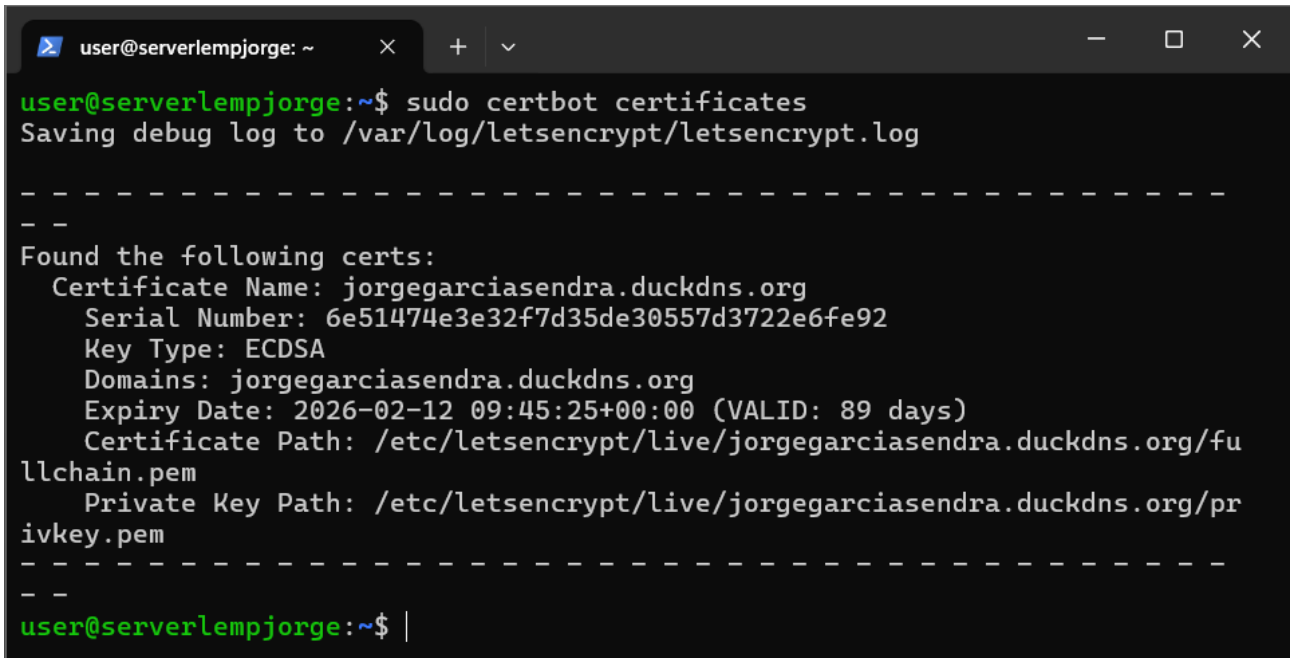
Creamos el archivo de configuración con el token de autenticación de DuckDNS y establecemos los permisos necesarios

```
user@serverlempjorge: ~
user@serverlempjorge:~$ echo "dns_ducksdns_token=736ebd7a-1f3e-409d-9b4e-7b1
ae44a6c5c" | sudo tee /etc/letsencrypt/duckdns.ini
dns_ducksdns_token=736ebd7a-1f3e-409d-9b4e-7b1ae44a6c5c
user@serverlempjorge:~$ sudo chmod 600 /etc/letsencrypt/duckdns.ini
user@serverlempjorge:~$
```

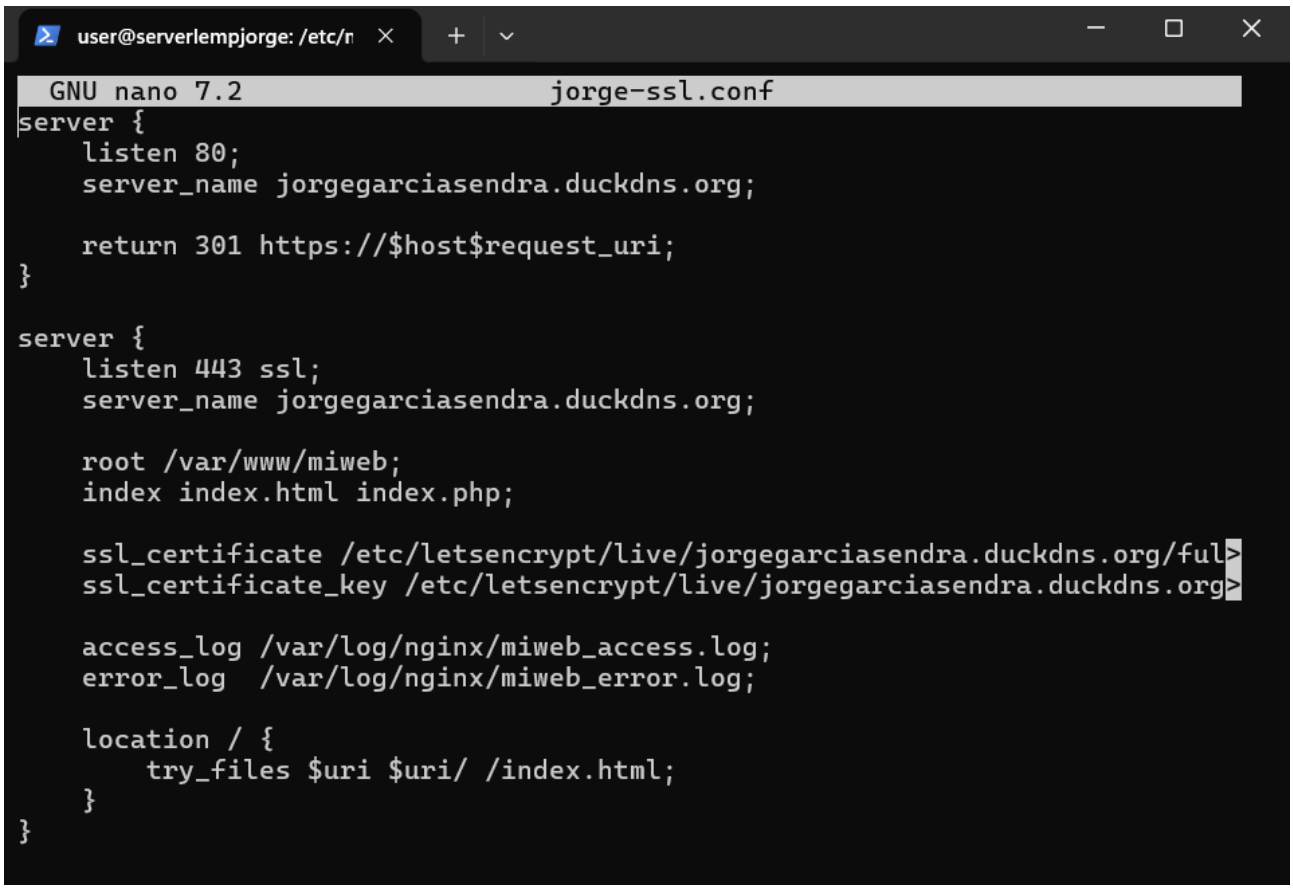
Ejecutamos certbot para obtener un certificado SSL valido

```
user@serverlempjorge: ~
user@serverlempjorge:~$ sudo certbot certonly --authenticator dns-duckdns --
dns-duckdns-credentials /etc/letsencrypt/duckdns.ini --dns-duckdns-propagati
on-seconds 60 -d jorgegarciasendra.duckdns.org
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for jorgegarciasendra.duckdns.org
Waiting 60 seconds for DNS changes to propagate

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/jorgegarciasendra.duckdns.org
/fullchain.pem
Key is saved at:         /etc/letsencrypt/live/jorgegarciasendra.duckdns.org
/privkey.pem
This certificate expires on 2026-02-12.
These files will be updated when the certificate renews.

NEXT STEPS:
- The certificate will need to be renewed before it expires. Certbot can aut
omatically renew the certificate in the background, but you may need to take
 steps to enable that functionality. See https://certbot.org/renewal-setup f
or instructions.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- -
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
 * Donating to EFF:                    https://eff.org/donate-le
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- -
user@serverlempjorge:~$
```

Comprobamos que el certificado se ha creado correctamente



8. Creamos el serverblock del HTTPS



Activamos el sitio y hacemos un reload

```
user@serverlempjorge:/etc/nginx/sites-available$ sudo ln -s /etc/nginx/sites
-available/jorge-ssl.conf /etc/nginx/sites-enabled/
user@serverlempjorge:/etc/nginx/sites-available$
```

```
user@serverlempjorge: /etc/r
user@serverlempjorge:/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
user@serverlempjorge:/etc/nginx/sites-available$ sudo systemctl reload nginx
user@serverlempjorge:/etc/nginx/sites-available$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset>
     Active: active (running) since Fri 2025-11-14 08:57:14 UTC; 1h 59min a>
       Docs: man:nginx(8)
    Process: 7982 ExecReload=/usr/sbin/nginx -g daemon on; master_process o>
   Main PID: 805 (nginx)
      Tasks: 2 (limit: 2265)
     Memory: 3.6M (peak: 6.1M)
        CPU: 202ms
     CGroup: /system.slice/nginx.service
             ├─ 805 "nginx: master process /usr/sbin/nginx -g daemon on; ma>
             └─7983 "nginx: worker process"

nov 14 08:57:13 serverlempjorge systemd[1]: Starting nginx.service - A high>
nov 14 08:57:14 serverlempjorge systemd[1]: Started nginx.service - A high >
nov 14 09:30:10 serverlempjorge systemd[1]: Reloading nginx.service - A hig>
nov 14 09:30:10 serverlempjorge nginx[6655]: 2025/11/14 09:30:10 [notice] 6>
nov 14 09:30:10 serverlempjorge systemd[1]: Reloaded nginx.service - A high>
nov 14 10:56:58 serverlempjorge systemd[1]: Reloading nginx.service - A hig>
nov 14 10:56:58 serverlempjorge nginx[7982]: 2025/11/14 10:56:58 [notice] 7>
nov 14 10:56:58 serverlempjorge systemd[1]: Reloaded nginx.service - A high>
lines 1-21/21 (END)
```

9. Entramos al navegador y comprobamos que funciona correctamente