

Installing and configuring Lego ACME client

Lego is a Let's Encrypt ACME client written in Go that supports APIs provided by many DNS providers. I find it easier to install and use compared to the official Let's Encrypt certbot client.

Installing Lego

Lego is hosted on GitHub. Check their [releases page](#) for the latest version. As of writing the current version is 4.4.0, so this can be downloaded via:

```
wget https://github.com/go-acme/lego/releases/download/v4.4.0/lego_v4.4.0_linux_amd64.tar.gz
```

The download is a gzip compressed tar file, so we will need to run these commands to extract it:

```
gzip -d lego_v4.4.0_linux_amd64.tar.gz && tar -xvf lego_v4.4.0_linux_amd64.tar
```

Lego is an executable file, but if it's not located within /bin or /usr/bin, we will need to specify the full path later. This can be remedied by moving it to /usr/bin:

```
sudo mv lego /usr/bin/
```

Requesting your certificates

Creating the script

First, we will use nano to create a new file within /usr/bin:

```
sudo nano /usr/bin/request-certs-lego
```

Then, paste in the following code:

```
#!/bin/bash

mkdir -p /etc/lego

GANDIV5_API_KEY=YOUR-KEY-HERE \
```

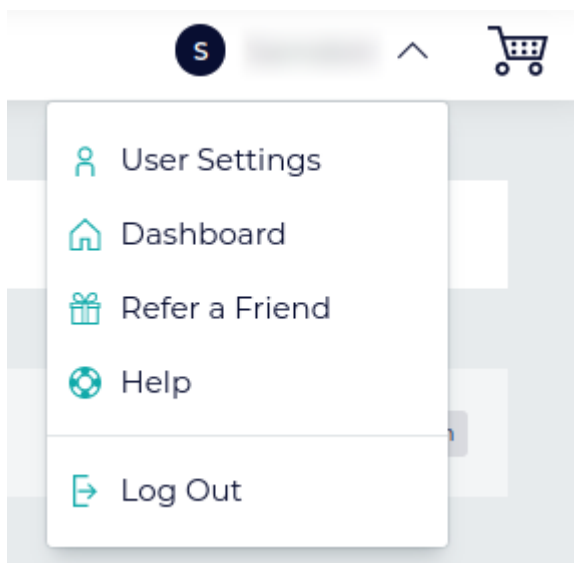
```
lego --email="someone@example.com" --path="/etc/lego" -a --dns gandiv5 -d YOUR.DOMAIN -d
\*.YOUR.DOMAIN run

systemctl restart nginx
```

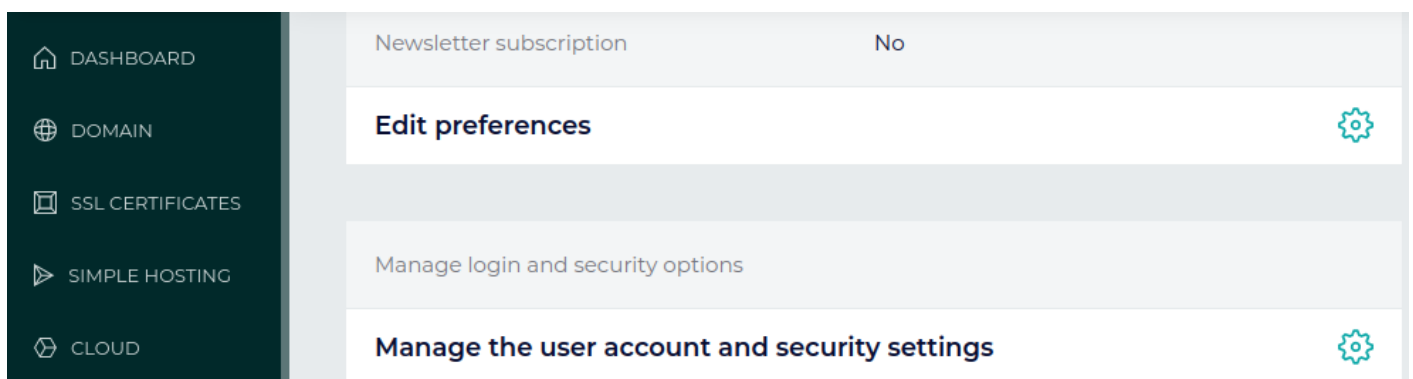
Now, modify the email and domain fields to match your setup. If you have multiple domains, copy the whole line and populate it with the other domain.

Retrieving your Gandi.net API key

Now, [login](#) to your Gandi.net account and click on the arrow to the right of your username, then click on **User Settings**:



Once in user settings, click on **Manage the user account and security settings**:



Go to the security section, and click on **Regenerate the API key**. The key will then appear on screen. Copy this to your clipboard.

Security

Edit your account and security settings, as well as third-party applications and authorizations.

Settings

Security

Authorized apps

My apps

Account security

Password	Last updated March 11, 2021. Change password
Two-factor authentication	Enable TOTP
Security key	Manage your security key authentication
Production API key	Last updated March 14, 2021. Regenerate the API key
Restrict Login IP Addresses	Add an IP restriction

Copy the API key you see here into the key field within the script.

Making the script executable

For security reasons, Linux won't let you run a bash script without first modifying it to make it executable. To do this, we will use `chmod`:

```
sudo chmod +x /usr/bin/request-certs-lego
```

Running the script

Now, we can run the script as root:

```
sudo request-certs-lego
```

You should see an output similar to this:

(EXAMPLE OUTPUT)

Using the certificates

If your website was previously set to use the certbot plugin or other SSL provider, you will need to modify the website definitions to point to the new Lego certificates, here's an example for NGINX:

```
ssl_certificate /etc/lego/certificates/YOUR.DOMAIN.crt;  
ssl_certificate_key /etc/lego/certificates/YOUR.DOMAIN.key;
```

Renewing the certificates

Creating the renewal script

As with the request script, we will use nano to create a new file within /usr/bin:

```
sudo nano /usr/bin/renew-certs-lego
```

Paste in the following code, modifying to match your email address & domains:

```
#!/bin/bash  
  
GANDIV5_API_KEY=YOUR-KEY-HERE \  
lego --email="someone@example.com" --path="/etc/lego" -a --dns gandiv5 -d YOUR.DOMAIN -d  
\*.YOUR.DOMAIN renew  
  
systemctl restart nginx
```

Making the script executable

For security reasons, Linux won't let you run a bash script without first modifying it to make it executable. To do this, we will use chmod:

```
sudo chmod +x /usr/bin/renew-certs-lego
```

Running the script

Now, we can run the script as root:

```
sudo renew-certs-lego
```

Automating renewal with cron

Now that we've confirmed the renewal script is working, we can call it via cron.d to make it run every month. To do this, we need to create a new file under /etc/cron.d/

```
sudo nano /etc/cron.d/lego-cert-renewal
```

Once nano is opened, paste in the following line:

```
0 1 1 * * root /usr/bin/renew-certs-lego > /root/renew-certs-lego-log 2>&1
```

0 means 0 minutes

1 means the first hour

1 means the first day of the month

* means every month

* means every year

root is the user the script will run as

> /root/renew-certs-lego-log is the location of the log file

2>&1 directs the output of the cron job to the log file

Revision #4

Created 2021-08-17 10:15:10 UTC by Sam

Updated 2021-08-17 13:09:04 UTC by Sam