

Node.js: how to manage the Let's Encrypt SSL certificates for nginx

We can easily handle Let's Encrypt SSL certificates for nginx with Node.js.

First, we need to get the SSL certificate for our domain:

```
sudo certbot --nginx -d example.com
```

Once gained root privileges (`sudo -i`) we can copy the certificate files to the appropriate directory plus restoring the correct ownership on these files:

```
cp /etc/letsencrypt/live/example.com/fullchain.pem  
/home/example/app/fullchain.pem  
chown example:example /home/example/app/fullchain.pem  
cp /etc/letsencrypt/live/example.com/privkey.pem  
/home/example/app/privkey.pem  
chown example:example /home/example/app/privkey.pem
```

Now we can use the certificate in Node.js:

```
'use strict';  
  
const express = require('express');  
const fs = require('fs');  
const https = require('https');
```

```
const port = process.env.PORT || 8080;
const app = express();

const sslOptions = {
  key: fs.readFileSync('privkey.pem'),
  cert: fs.readFileSync('fullchain.pem')
};

https.createServer(sslOptions, app).listen(port);
```

Finally, we need a cronjob for the root user (`crontab -e`) with the following commands:

```
+ + + + + /usr/bin/certbot renew --quiet
+ + + + + cat
/etc/letsencrypt/live/example.com/fullchain.pem >
/home/example/app/fullchain.pem && chown
example:example /home/example/app/fullchain.pem
+ + + + + cat
/etc/letsencrypt/live/example.com/privkey.pem >
/home/example/app/privkey.pem && chown
example:example /home/example/app/privkey.pem
```

Replace the + signs with the appropriate time values and restart the cron daemon. **Important:** the second and third commands should **not** be executed at the same time of the first one. We don't know in advance how much time it will take to certbot to renew the certificates, so make sure that such commands will run just after a few minutes.