How To Secure Apache with Let's Encrypt on Ubuntu 18.04

Posted April 27, 2018 152k APACHE LET'S ENCRYPT UBUNTU 18.04

By: Erika Heidi By: Kathleen Juell

Not using **Ubuntu 18.04**? Choose a different version:

- CentOS 7
- Debian 9
- Debian 8

Introduction

Let's Encrypt is a Certificate Authority (CA) that provides an easy way to obtain and install free TLS/SSL certificates, thereby enabling encrypted HTTPS on web servers. It simplifies the process by providing a software client, Certbot, that attempts to automate most (if not all) of the required steps. Currently, the entire process of obtaining and installing a certificate is fully automated on both Apache and Nginx.

In this tutorial, you will use Certbot to obtain a free SSL certificate for Apache on Ubuntu 18.04 and set up your certificate to renew automatically.

This tutorial will use a separate Apache virtual host file instead of the default configuration file. We recommend creating new Apache virtual host files for each domain because it helps to avoid common mistakes and maintains the default files as a fallback configuration.

Prerequisites

To follow this tutorial, you will need:

- One Ubuntu 18.04 server set up by following this initial server setup for Ubuntu 18.04 tutorial, including a sudo non-root user and a firewall.
- A fully registered domain name. This tutorial will use **example.com** throughout. You can purchase a domain name on Namecheap, get one for free on Freenom, or use the domain registrar of your choice.
- Both of the following DNS records set up for your server. You can follow this introduction to DigitalOcean DNS for details on how to add them.
    - An A record with `example.com` pointing to your server's public IP address.
    - An A record with `www.example.com` pointing to your server's public IP address.
- Apache installed by following How To Install Apache on Ubuntu 18.04. Be sure that you have a virtual host file for your domain. This tutorial will use `/etc/apache2/sites-available/example.com.conf` as an example.

Step 1 — Installing Certbot

The first step to using Let's Encrypt to obtain an SSL certificate is to install the Certbot software on your server.

Certbot is in very active development, so the Certbot packages provided by Ubuntu tend to be outdated. However, the Certbot developers maintain a Ubuntu software repository with up-to-date versions, so we'll use that repository instead.

First, add the repository:

```
$ sudo add-apt-repository ppa:certbot/certbot
```


You'll need to press `ENTER` to accept.

Install Certbot's Apache package with `apt`:

```
$ sudo apt install python-certbot-apache
```

Certbot is now ready to use, but in order for it to configure SSL for Apache, we need to verify some of Apache's configuration.

Step 2 — Set Up the SSL Certificate

Certbot needs to be able to find the correct virtual host in your Apache configuration for it to automatically configure SSL. Specifically, it does this by looking for a `ServerName` directive that matches the domain you request a certificate for.

If you followed the virtual host set up step in the Apache installation tutorial, you should have a VirtualHost block for your domain at `/etc/apache2/sites-available/example.com.conf` with the `ServerName` directive already set appropriately.

To check, open the virtual host file for your domain using `nano` or your favorite text editor:

```
$ sudo nano /etc/apache2/sites-available/example.com.conf
```

Find the existing `ServerName` line. It should look like this:

/etc/apache2/sites-available/example.com.conf

```
...
ServerName example.com;
...
```

If it does, exit your editor and move on to the next step.

If it doesn't, update it to match. Then save the file, quit your editor, and verify the syntax of your configuration edits:

```
$ sudo apache2ctl configtest
```

If you get an error, reopen the virtual host file and check for any typos or missing characters. Once your configuration file's syntax is correct, reload Apache to load the new configuration:

```
$ sudo systemctl reload apache2
```

Certbot can now find the correct VirtualHost block and update it.

Next, let's update the firewall to allow HTTPS traffic.

Step 3 — Allowing HTTPS Through the Firewall

If you have the `ufw` firewall enabled, as recommended by the prerequisite guides, you'll need to adjust the settings to allow for HTTPS traffic. Luckily, Apache registers a few profiles with `ufw` upon installation.

You can see the current setting by typing:

```
$ sudo ufw status
```

It will probably look like this, meaning that only HTTP traffic is allowed to the web server:

```
Output
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
Apache                     ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
Apache (v6)                ALLOW       Anywhere (v6)
```

To additionally let in HTTPS traffic, allow the Apache Full profile and delete the redundant Apache profile allowance:

```
$ sudo ufw allow 'Apache Full'
$ sudo ufw delete allow 'Apache'
```

Your status should now look like this:

```
$ sudo ufw status
```

```
Output
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
Apache Full                ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
Apache Full (v6)           ALLOW       Anywhere (v6)
```

Next, let's run Certbot and fetch our certificates.

Step 4 — Obtaining an SSL Certificate

Certbot provides a variety of ways to obtain SSL certificates through plugins. The Apache plugin will take care of reconfiguring Apache and reloading the config whenever necessary. To use this plugin, type the following:

```
$ sudo certbot --apache -d example.com -d www.example.com
```

This runs `certbot` with the `--apache` plugin, using `-d` to specify the names you'd like the certificate to be valid for.

If this is your first time running `certbot`, you will be prompted to enter an email address and agree to the terms of service. After doing so, `certbot` will communicate with the Let's Encrypt server, then run a challenge to verify that you control the domain you're requesting a certificate for.

If that's successful, `certbot` will ask how you'd like to configure your HTTPS settings:

```
Output
Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.
------------------------------------------------------------------------------
1: No redirect - Make no further changes to the webserver configuration.
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for
new sites, or if you're confident your site works on HTTPS. You can undo this

change by editing your web server's configuration.
------------------------------------------------------------------------------
Select the appropriate number [1-2] then [enter] (press 'c' to cancel):
```

Select your choice then hit `ENTER`. The configuration will be updated, and Apache will reload to pick up the new settings. `certbot` will wrap up with a message telling you the process was successful and where your certificates are stored:

```
Output
IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at:
   /etc/letsencrypt/live/example.com/fullchain.pem
   Your key file has been saved at:
   /etc/letsencrypt/live/example.com/privkey.pem
   Your cert will expire on 2018-07-23. To obtain a new or tweaked
   version of this certificate in the future, simply run certbot again
   with the "certonly" option. To non-interactively renew *all* of
   your certificates, run "certbot renew"
 - Your account credentials have been saved in your Certbot
   configuration directory at /etc/letsencrypt. You should make a
   secure backup of this folder now. This configuration directory will
   also contain certificates and private keys obtained by Certbot so
```

```
  making regular backups of this folder is ideal.
- 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
```

Your certificates are downloaded, installed, and loaded. Try reloading your website using `https://` and notice your browser's security indicator. It should indicate that the site is properly secured, usually with a green lock icon. If you test your server using the SSL Labs Server Test, it will get an **A** grade.

Let's finish by testing the renewal process.

Step 5 — Verifying Certbot Auto-Renewal

Let's Encrypt's certificates are only valid for ninety days. This is to encourage users to automate their certificate renewal process. The `certbot` package we installed takes care of this for us by adding a renew script to `/etc/cron.d`. This script runs twice a day and will automatically renew any certificate that's within thirty days of expiration.

To test the renewal process, you can do a dry run with `certbot`:

```
$ sudo certbot renew --dry-run
```

If you see no errors, you're all set. When necessary, Certbot will renew your certificates and reload Apache to pick up the changes. If the automated renewal process ever fails, Let's Encrypt will send a message to the email you specified, warning you when your certificate is about to expire.

Conclusion

In this tutorial, you installed the Let's Encrypt client `certbot`, downloaded SSL certificates for your domain, configured Apache to use these certificates, and set up automatic certificate renewal. If you have further questions about using Certbot, their documentation is a good place to start.

By: Erika Heidi By: Kathleen Juell

Related Tutorials

How To Sync and Share Your Files with Seafile on Debian 9

How To Install YunoHost on Debian 9

How To Ensure Code Quality with SonarQube on Ubuntu 18.04

How To Use Traefik as a Reverse Proxy for Docker Containers on Debian 9

How To Install and Configure an Apache ZooKeeper Cluster on Ubuntu 18.04

24 Comments

Leave a comment...

Log In to Comment

jureq *June 4, 2018*

0 certbot is now in standard ubuntu repository. No need to add ppa repository.
Use python3-certbot-apache plugin from standard reepository.

robert98983 *June 7, 2018*

0 Hi,

I have got two sites running on VirtualHosts. (/var/www/site1 and /var/www/site2)
I want to get a SSL for them.
When I try to `sudo apache2ctl configtest` I get this warning:

AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 192.168.2.8. Set the 'ServerName' directive globally to suppress this message
Syntax OK

What can I do to fix this?

Greetings.

yesitsme *June 14, 2018*

0 How would I add wildcard dns?

fpschalk *June 16, 2018*

0 I followed your procedure and it worked like a charm except ... I have a WebSocket app behind my Nginx reverse proxy server. My app runs flawlessly in Chrome but Firefox won't even load it. The issue revolves around communicating with ws over https. Firefox ran the app flawlessly prior to my encrypting my Nginx reverse proxy server. How do you suggest I proceed? Encrypting my Haskell server and Cycle.js front end would be difficult - for me anyway.

solprogroup *July 19, 2018*

0 Hi There,

Thanks the tutorial. However, I tried installing certbot with the instructions given but it kept showing error messages:

```
root@solpro:~# sudo add-apt -repository ppa:cerbot/certbot
sudo: add-apt: command not found
root@solpro:~#  -repository ppa:cerbot/certbot
-repository: command not found
root@solpro:~#  ppa:cerbot/certbot
-bash: ppa:cerbot/certbot: No such file or directory
root@solpro:~# cerbot/certbot
-bash: cerbot/certbot: No such file or directory
root@solpro:~# sudo apt install python-certbot-apache
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package python-certbot-apache
```

```
root@solpro:~# sudo add-apt-repository ppa:cerbot/certbot
Cannot add PPA: 'ppa:~cerbot/ubuntu/certbot'.
ERROR: '~cerbot' user or team does not exist.
```

What am I doing wrong?

---

anonjihen *November 8, 2018*

1 I don't know if you ever figured this out, but I thought I would reply for you or anyone else who might read this.

In the first line of your code:

```
root@solpro:~# sudo add-apt -repository ppa:cerbot/certbot
```

You have an extra space between 'apt -repository.' It should be 'add-apt-repository' as all one word, like this:

```
root@solpro:~# sudo add-apt-repository ppa:cerbot/certbot
```

Small typos like that are common!

---

rmarkjr81 *July 31, 2018*

0 Yes the part to install certbot does not work.
So when I run this: sudo add-apt-repository ppa:certbot/certbot

I get this:
Hit:1 http://archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://ppa.launchpad.net/certbot/certbot/ubuntu bionic InRelease
Get:3 http://archive.ubuntu.com/ubuntu bionic-security InRelease [83.2 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Fetched 172 kB in 5s (31.4 kB/s)
Reading package lists... Done

everything sees to be ok so far .. however when I run this: sudo apt install python-certbot-apache

I get this:
The following packages have unmet dependencies:
python-certbot-apache : Depends: python3-certbot-apache but it is not going to be installed
E: Unable to correct problems, you have held broken packages.

I have done everything prior to this on the main How to setup Owncloud on Ubuntu 18.04 ....which by the way is very time consuming just to be stuck at a dead end now. I am using Ubuntu 18.04 server.

any help with this would be great thanks!

dnsgonz *August 16, 2018*

0 Instead of using PPA I used this: "sudo apt install python-certbot-apache"

pcpgjanssen *September 6, 2018*

2 For anyone getting the following error on the installation of the Certbot's Apache packages in Ubuntu 18.04

```
The following packages have unmet dependencies:
python-certbot-apache : Depends: python3-certbot-apache but it is not going to be installed
E: Unable to correct problems, you have held broken packages.
```

You will have to add the "Universe" repository using the following command:

```
sudo add-apt-repository universe
```

Also, make sure your system is up to date (sudo apt update + sudo apt upgrade).

yuliyan *October 20, 2018*

0 You are missing before step one a very important step:

```
sudo apt-get install -y software-properties-common
```

GLHF

hefese *October 30, 2018*

0 You say as a prerequisite; "A fully registered domain name. This tutorial will use example.com throughout. You can purchase a domain name on Namecheap, get one for free on Freenom, or use the domain registrar of your choice."

But although I have a web application and a domain name, because it is in an hosting server, I cannot intervene the configurations of server. Just testing purpose, how can I simulate and experience this process in my "local" system? Is there any way?

manndavidjapan *November 6, 2018*

1 I think I followed the tutorial carefully and didn't notice the error messages that the other users experienced, but I just get 'Unable to connect' messages. Can you point out where I might have gone wrong.

umbertofilippo *November 10, 2018*

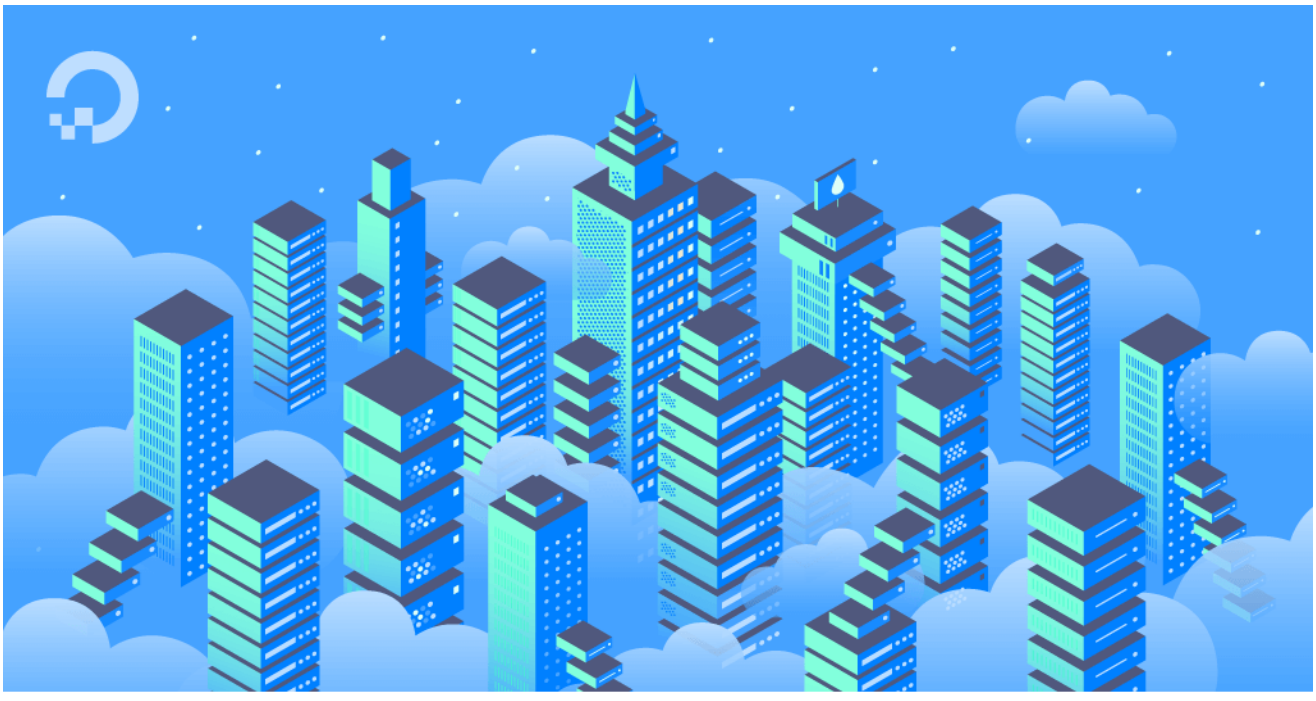
Same here. The only thing I can add is that my site is new so i chose option 2 asking certbot to redirect traffic over HTTPS. As soon as I did this, my website running fine on HTTP became unreachable, giving a timeout error, even if I try accessing it with the same URL as before... Hopefully, I'll figure out what went wrong and will update here with the solution.

EDIT 1

First of all, I think one has to adapt the config file for the VirtualHost proposed in the previous Digital Ocean tutorial to include the SSL port (443) instead of the standard HTTP one (80).
I tried to modify mine like this (see below) as proposed in the official Apache docs, restarted apache, but still no website. Will hopefully come back soon!

```
LoadModule ssl_module modules/mod_ssl.so

<VirtualHost *:443>
    ServerAdmin admin@example.com
    ServerName example.com
    ServerAlias www.example.com
    DocumentRoot /var/www/example.com/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    SSLEngine on
    SSLCertificateFile /etc/letsencrypt/live/www.example.org/fullchain.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/www.example.org/privkey.pem
</VirtualHost>
```



How To Install the Apache Web Server on Ubuntu 18.04

by Justin Ellingwood
by Kathleen Juell

The Apache HTTP server is the most widely-used web server in the world. It provides many powerful features including dynamically loadable modules, robust media support, and extensive integration with other popular software. In this guide, we'll explain how to install an

umbertofilippo *November 10, 2018*

1 **SOLUTION**

Ok, in my case I had to add a rule for port HTTPS to my service group in the Amazon EC2 console (I am using an Amazon EC2 instance). Hope this is helpful!

I also tried to overwrite my conf file with the original found on the same Digital Ocean tutorial I pointed out before, having my VirtualHost pointing to port 80 again, and everything seems to be running flawlessly!
So I guess the tutorial is fine, just remember to add port HTTPS rule Inbound in the Security Group if you have an Amazon EC2 instance like me. Cheers!

wheels745 *December 18, 2018*

0 This also worked for me! For lightsail, you want to go to networking, then just add a rule for https

LividPython *November 12, 2018*

0 How would I make it so my website would use https automatically rather then http?

fredrbus *November 17, 2018*

0 It will prompt you when certbot is fetching the certificates.

If you're not doing this step, check out this article in the Apache Wiki.

marhabaworld *November 15, 2018*

0 i seem to be getting an error - i had certbot already installed however it got expired 2 days back - now i am trying to go through this process however i seem to be getting error and not able to renew the ssl.

fredrbus *November 17, 2018*

0 If your certificates are expired, you can run

```
certbot renew
```

If that doesn't work, read this.

EDIT:
To avoid doing this again, you can set up automatic renewal.

aubzp *November 28, 2018*

0 I successfully installed the certificate on a wordpress site and it works except for the fact that I now have no images, css so all that is displayed on screen is text, unstyled text! WTF?

ktcw *November 28, 2018*

0 I'm suspecting that these page assets are hardlinked to their old http URLs? They should be pointed to the new https URLs. Alternatively, you can configure your web server to automatically redirect all http traffic to https.

computertechs *December 14, 2018*

0 I noticed that I started to get a warning message on my droplets from certbot stating that TLS-SNI-01 authentication was deprecated and would stop working soon:

https://community.letsencrypt.org/t/february-13-2019-end-of-life-for-all-tls-sni-01-validation-support/74209

You can force a new certificate to be issued using a supported authentication type with this command:

```
certbot certonly --cert-name yourcertname --force-renewal -a webroot -w /your/webroot/direct
```

https://community.letsencrypt.org/t/how-to-change-certbot-verification-method/56735

sherwinrheycondez *December 29, 2018*

0 How to do this in multiple sites. I already have the first one setup but I get and error when I do it again on another domain name.

jeffo12 *January 12, 2019*

0 I found your instructions to be the best on the net. Thank you!
I have a server up and running in multi-site config ubuntu 18 using Let's encrypt ssl. The ssl is up and running for 2 months. I ran the dry-run renewal and got errors that seem to be timing out when fetching .wellknown/acme-challenge/ file. I backed up a few steps and also did apache2 config test and got error below. When I first installed ssl the dry run ran perfectly. firewall is set properly as per above instructions too.

Error: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK

My etc/hosts file says
127.0.0.1 localhost
127.0.1.1 myusername

I checked the mysitename.conf file in sites-enabled directory and everything is perfect on it.
What else can I check.