Get Certbot

- System Requirements
- Installation
- Snap (Recommended)
- Alternative 1: Docker
- Alternative 2: Pip
- Alternative 3: Third Party Distributions
- Certbot-Auto [Deprecated]

Table of Contents

System Requirements

- Linux, macOS, BSD and Windows
- Recommended root access on Linux/BSD/Required Administrator access on Windows
- Port 80 Open

> [!NOTE] Certbot is most useful when run with root privileges, because it is then able to automatically configure TLS/SSL for Apache and nginx. *Certbot is meant to be run directly on a web server*, normally by a system administrator. In most cases, running Certbot on your personal computer is not a useful option. The instructions below relate to installing and running Certbot on a server.

Installation

Unless you have very specific requirements, we kindly suggest that you use the installation instructions for your system found at https://certbot.eff.org/instructions.

Snap (Recommended)

Our instructions are the same across all systems that use Snap. You can find instructions for installing Certbot through Snap can be found at https://certbot.eff.org/instructions by selecting your server software and then choosing "snapd" in the "System" dropdown menu.

Most modern Linux distributions (basically any that use systemd) can install Certbot packaged as a snap. Snaps are available for x86_64, ARMv7 and ARMv8 architectures. The Certbot snap provides an easy way to ensure you have the latest version of Certbot with features like automated certificate renewal preconfigured.

If you unable to use snaps, you can use an alternate method for installing `certbot`.

Alternative 1: Docker

Docker is an amazingly simple and quick way to obtain a certificate. However, this mode of operation is unable to install certificates or configure your webserver, because our installer plugins cannot reach your webserver from inside the Docker container.

Most users should use the instructions at certbot.eff.org. You should only use Docker if you are sure you know what you are doing and have a good reason to do so.

You should definitely read the `where-certs` section, in order to know how to manage the certificates manually. Our ciphersuites page provides some information about recommended ciphersuites. If none of these make much sense to you, you should definitely use the installation method recommended for your system at certbot.eff.org, which enables you to use installer plugins that cover both of those hard topics.

If you're still not convinced and have decided to use this method, from the server that the domain you're requesting a certificate for resolves to, install Docker, then issue a command like the one found below. If you are using Certbot with the `Standalone` plugin, you will need to make the port it uses accessible from outside of the container by including something like `-p 80:80` or `-p 443:443` on the command line before `certbot/certbot`.

```
sudo docker run -it --rm --name certbot \
            -v "/etc/letsencrypt:/etc/letsencrypt" \
            -v "/var/lib/letsencrypt:/var/lib/letsencrypt" \
            certbot/certbot certonly
```

Running Certbot with the `certonly` command will obtain a certificate and place it in the directory `/etc/letsencrypt/live` on your system. Because Certonly cannot install the certificate from within Docker, you must install the certificate manually according to the procedure recommended by the provider of your webserver.

There are also Docker images for each of Certbot's DNS plugins available at https://hub.docker.com/u/certbot which automate doing domain validation over DNS for popular providers. To use one, just replace `certbot/certbot` in the command above with the name of the image you want to use. For example, to use Certbot's plugin for Amazon Route 53, you'd use `certbot/dns-route53`. You may also need to add flags to Certbot and/or mount additional directories to provide access to your DNS API credentials as specified in the `DNS plugin documentation <dns_plugins>`.

For more information about the layout of the `/etc/letsencrypt` directory, see `where-certs`.

Alternative 2: Pip

Installing Certbot through pip is only supported on a best effort basis and when using a virtual environment. Instructions for installing Certbot through pip can be found at https://certbot.eff.org/instructions by selecting your server software and then choosing "pip" in the "System" dropdown menu.

Alternative 3: Third Party Distributions

Third party distributions exist for other specific needs. They often are maintained by these parties outside of Certbot and tend to rapidly fall out of date on LTS-style distributions.

Certbot-Auto [Deprecated]

We used to have a shell script named `certbot-auto` to help people install Certbot on UNIX operating systems, however, this script is no longer supported.

Please remove `certbot-auto`. To do so, you need to do three things:

1. If you added a cron job or systemd timer to automatically run certbot-auto to renew your certificates, you should delete it. If you did this by following our instructions, you can delete the entry added to /etc/crontab by running a command like sudo sed -i '/certbot-

auto/d' /etc/crontab.

2. Delete the certbot-auto script. If you placed it in /usr/local/bin like we recommended, you can delete it by running `sudo rm /usr/local/bin/certbot-auto.
3. Delete the Certbot installation created by certbot-auto by running sudo rm -rf /opt/eff.org.

□□□ #2
□ □□□ □□□ 2026-10-31 13:02:37 UTC
□ □□□ □□□ 2026-10-31 13:02:38 UTC