NAME

certbot - Automatically configure HTTPS using Let's Encrypt

SYNOPSIS

The objective of Certbot, Let's Encrypt, and the ACME (Automated Certificate Management Environment) protocol is to make it possible to set up an HTTPS server and have it automatically obtain a browser-trusted certificate, without any human intervention. This is accomplished by running a certificate management agent on the web server.

This agent is used to:

- Automatically prove to the Let's Encrypt CA that you control the website
- Obtain a browser-trusted certificate and set it up on your web server
- Keep track of when your certificate is going to expire, and renew it
- Help you revoke the certificate if that ever becomes necessary.

OPTIONS

```
usage:
  certbot [SUBCOMMAND] [options] [-d DOMAIN] [-d DOMAIN] ...

Certbot can obtain and install HTTPS/TLS/SSL certificates.  By default,
it will attempt to use a webserver both for obtaining and installing the
certificate. The most common SUBCOMMANDS and flags are:

obtain, install, and renew certificates:
    (default) run   Obtain & install a certificate in your current webserver
    certonly        Obtain or renew a certificate, but do not install it
    renew           Renew all previously obtained certificates that are near expiry
    enhance         Add security enhancements to your existing configuration
   -d DOMAINS       Comma-separated list of domains to obtain a certificate for

  --apache          Use the Apache plugin for authentication & installation
  --standalone      Run a standalone webserver for authentication
  --nginx           Use the Nginx plugin for authentication & installation
  --webroot         Place files in a server's webroot folder for authentication
  --manual          Obtain certificates interactively, or using shell script hooks
```

```
  -n              Run non-interactively
 --test-cert      Obtain a test certificate from a staging server
 --dry-run        Test "renew" or "certonly" without saving any certificates to disk

manage certificates:
   certificates   Display information about certificates you have from Certbot
   revoke         Revoke a certificate (supply --cert-name or --cert-path)
   delete         Delete a certificate (supply --cert-name)
   reconfigure    Update a certificate's configuration (supply --cert-name)

manage your account:
   register       Create an ACME account
   unregister     Deactivate an ACME account
   update_account Update an ACME account
   show_account   Display account details
 --agree-tos      Agree to the ACME server's Subscriber Agreement
  -m EMAIL        Email address for important account notifications

optional arguments:
 -h, --help           show this help message and exit
 -c CONFIG_FILE, --config CONFIG_FILE
                      path to config file (default: /etc/letsencrypt/cli.ini
                      and ~/.config/letsencrypt/cli.ini)
 -v, --verbose        This flag can be used multiple times to incrementally
                      increase the verbosity of output, e.g. -vvv. (default:
                      0)
 --max-log-backups MAX_LOG_BACKUPS
                      Specifies the maximum number of backup logs that
                      should be kept by Certbot's built in log rotation.
                      Setting this flag to 0 disables log rotation entirely,
                      causing Certbot to always append to the same log file.
                      (default: 1000)
 -n, --non-interactive, --noninteractive
                      Run without ever asking for user input. This may
                      require additional command line flags; the client will
                      try to explain which ones are required if it finds one
                      missing (default: False)
 --force-interactive  Force Certbot to be interactive even if it detects
                      it's not being run in a terminal. This flag cannot be
                      used with the renew subcommand. (default: False)
```

-d DOMAIN, --domains DOMAIN, --domain DOMAIN
Domain names to include. For multiple domains you can use multiple -d flags or enter a comma separated list of domains as a parameter. All domains will be included as Subject Alternative Names on the certificate. The first domain will be used as the certificate name, unless otherwise specified or if you already have a certificate with the same name. In the case of a name conflict, a number like -0001 will be appended to the certificate name. (default: Ask)

--eab-kid EAB_KID Key Identifier for External Account Binding (default: None)

--eab-hmac-key EAB_HMAC_KEY
HMAC key for External Account Binding (default: None)

--cert-name CERTNAME Certificate name to apply. This name is used by Certbot for housekeeping and in file paths; it doesn't affect the content of the certificate itself. To see certificate names, run 'certbot certificates'. When creating a new certificate, specifies the new certificate's name. (default: the first provided domain or the name of an existing certificate on your system for the same domains)

--dry-run Perform a test run of the client, obtaining test (invalid) certificates but not saving them to disk. This can currently only be used with the 'certonly' and 'renew' subcommands. Note: Although --dry-run tries to avoid making any persistent changes on a system, it is not completely side-effect free: if used with webserver authenticator plugins like apache and nginx, it makes and then reverts temporary config changes in order to obtain test certificates, and reloads webservers to deploy and then roll back those changes. It also calls --pre-hook and --post-hook commands if they are defined because they may be necessary to accurately simulate renewal. --deploy-hook commands are not called. (default: False)

--debug-challenges After setting up challenges, wait for user input before submitting to CA. When used in combination with the '-v' option, the challenge URLs or FQDNs and their expected return values are shown. (default: False)

--preferred-chain PREFERRED_CHAIN
Set the preferred certificate chain. If the CA offers multiple certificate chains, prefer the chain whose topmost certificate was issued from this Subject Common Name. If no match, the default offered chain will be used. (default: None)

--preferred-challenges PREF_CHALLS
A sorted, comma delimited list of the preferred challenge to use during authorization with the most preferred challenge listed first (Eg, "dns" or "http,dns"). Not all plugins support all challenges. See https://certbot.eff.org/docs/using.html#plugins for details. ACME Challenges are versioned, but if you pick "http" rather than "http-01", Certbot will select the latest version automatically. (default: [])

--issuance-timeout ISSUANCE_TIMEOUT
This option specifies how long (in seconds) Certbot will wait for the server to issue a certificate. (default: 90)

--user-agent USER_AGENT
Set a custom user agent string for the client. User agent strings allow the CA to collect high level statistics about success rates by OS, plugin and use case, and to know when to deprecate support for past Python versions and flags. If you wish to hide this information from the Let’s Encrypt server, set this to "". (default: CertbotACMEClient/2.5.0 (certbot; OS_NAME OS_VERSION) Authenticator/XXX Installer/YYY (SUBCOMMAND; flags: FLAGS) Py/major.minor.patchlevel). The flags encoded in the user agent are: --duplicate, --force-renew, --allow-subset-of-names, -n, and whether any hooks are set.

--user-agent-comment USER_AGENT_COMMENT
Add a comment to the default user agent string. May be used when repackaging Certbot or calling it from another tool to allow additional statistical data to be collected. Ignored if --user-agent is set. (Example: Foo-Wrapper/1.0) (default: None)

automation:

Flags for automating execution & other tweaks

--keep-until-expiring, --keep, --reinstall
If the requested certificate matches an existing certificate, always keep the existing one until it is due for renewal (for the 'run' subcommand this means reinstall the existing certificate). (default: Ask)

--expand If an existing certificate is a strict subset of the requested names, always expand and replace it with the additional names. (default: Ask)

--version show program's version number and exit

--force-renewal, --renew-by-default
If a certificate already exists for the requested domains, renew it now, regardless of whether it is near expiry. (Often --keep-until-expiring is more appropriate). Also implies --expand. (default: False)

--renew-with-new-domains
If a certificate already exists for the requested certificate name but does not match the requested domains, renew it now, regardless of whether it is near expiry. (default: False)

--reuse-key When renewing, use the same private key as the existing certificate. (default: False)

--no-reuse-key When renewing, do not use the same private key as the existing certificate. Not reusing private keys is the default behavior of Certbot. This option may be used to unset --reuse-key on an existing certificate. (default: False)

--new-key When renewing or replacing a certificate, generate a new private key, even if --reuse-key is set on the existing certificate. Combining --new-key and --reuse-key will result in the private key being replaced and then reused in future renewals. (default: False)

--allow-subset-of-names
When performing domain validation, do not consider it a failure if authorizations can not be obtained for a strict subset of the requested domains. This may be useful for allowing renewals for multiple domains to succeed even if some domains no longer point at this system. This option cannot be used with --csr.

(default: False)
--agree-tos Agree to the ACME Subscriber Agreement (default: Ask)
--duplicate Allow making a certificate lineage that duplicates an existing one (both can be renewed in parallel) (default: False)
-q, --quiet Silence all output except errors. Useful for automation via cron. Implies --non-interactive. (default: False)

security:
Security parameters & server settings

--rsa-key-size N Size of the RSA key. (default: 2048)
--key-type {rsa,ecdsa}
Type of generated private key. Only *ONE* per invocation can be provided at this time. (default: ecdsa)
--elliptic-curve N The SECG elliptic curve name to use. Please see RFC 8446 for supported values. (default: secp256r1)
--must-staple Adds the OCSP Must-Staple extension to the certificate. Autoconfigures OCSP Stapling for supported setups (Apache version >= 2.3.3). (default: False)
--redirect Automatically redirect all HTTP traffic to HTTPS for the newly authenticated vhost. (default: redirect enabled for install and run, disabled for enhance)
--no-redirect Do not automatically redirect all HTTP traffic to HTTPS for the newly authenticated vhost. (default: redirect enabled for install and run, disabled for enhance)
--hsts Add the Strict-Transport-Security header to every HTTP response. Forcing browser to always use SSL for the domain. Defends against SSL Stripping. (default: None)
--uir Add the "Content-Security-Policy: upgrade-insecure-requests" header to every HTTP response. Forcing the browser to use https:// for every http:// resource. (default: None)
--staple-ocsp Enables OCSP Stapling. A valid OCSP response is stapled to the certificate that the server offers during TLS. (default: None)

--strict-permissions Require that all configuration files are owned by the current user; only needed if your config is somewhere unsafe like /tmp/ (default: False)

--auto-hsts Gradually increasing max-age value for HTTP Strict Transport Security security header (default: False)

testing:

The following flags are meant for testing and integration purposes only.

--run-deploy-hooks When performing a test run using '--dry-run' or 'reconfigure', run any applicable deploy hooks. This includes hooks set on the command line, saved in the certificate's renewal configuration file, or present in the renewal-hooks directory. To exclude directory hooks, use --no-directory-hooks. The hook(s) will only be run if the dry run succeeds, and will use the current active certificate, not the temporary test certificate acquired during the dry run. This flag is recommended when modifying the deploy hook using 'reconfigure'. (default: False)

--test-cert, --staging
Use the staging server to obtain or revoke test (invalid) certificates; equivalent to --server https://acme-staging-v02.api.letsencrypt.org/directory (default: False)

--debug Show tracebacks in case of errors (default: False)

--no-verify-ssl Disable verification of the ACME server's certificate. The root certificates trusted by Certbot can be overriden by setting the REQUESTS_CA_BUNDLE environment variable. (default: False)

--http-01-port HTTP01_PORT
Port used in the http-01 challenge. This only affects the port Certbot listens on. A conforming ACME server will still attempt to connect on port 80. (default: 80)

--http-01-address HTTP01_ADDRESS
The address the server listens to during http-01 challenge. (default:)

--https-port HTTPS_PORT
Port used to serve HTTPS. This affects which port

```
                    Nginx will listen on after a LE certificate is
                    installed. (default: 443)
  --break-my-certs      Be willing to replace or renew valid certificates with
                    invalid (testing/staging) certificates (default:
                    False)

paths:
  Flags for changing execution paths & servers

  --cert-path CERT_PATH
                    Path to where certificate is saved (with certonly
                    --csr), installed from, or revoked (default: None)
  --key-path KEY_PATH   Path to private key for certificate installation or
                    revocation (if account key is missing) (default: None)
  --fullchain-path FULLCHAIN_PATH
                    Accompanying path to a full certificate chain
                    (certificate plus chain). (default: None)
  --chain-path CHAIN_PATH
                    Accompanying path to a certificate chain. (default:
                    None)
  --config-dir CONFIG_DIR
                    Configuration directory. (default: /etc/letsencrypt)
  --work-dir WORK_DIR   Working directory. (default: /var/lib/letsencrypt)
  --logs-dir LOGS_DIR   Logs directory. (default: /var/log/letsencrypt)
  --server SERVER       ACME Directory Resource URI. (default:
                    https://acme-v02.api.letsencrypt.org/directory)

manage:
  Various subcommands and flags are available for managing your
  certificates:

  certificates          List certificates managed by Certbot
  delete                Clean up all files related to a certificate
  renew                 Renew all certificates (or one specified with --cert-
                    name)
  revoke                Revoke a certificate specified with --cert-path or
                    --cert-name
  reconfigure           Update renewal configuration for a certificate
                    specified by --cert-name
```

```
run:
  Options for obtaining & installing certificates

certonly:
  Options for modifying how a certificate is obtained

  --csr CSR           Path to a Certificate Signing Request (CSR) in DER or
                      PEM format. Currently --csr only works with the
                      'certonly' subcommand. (default: None)

renew:
  The 'renew' subcommand will attempt to renew any certificates previously
  obtained if they are close to expiry, and print a summary of the results.
  By default, 'renew' will reuse the plugins and options used to obtain or
  most recently renew each certificate. You can test whether future renewals
  will succeed with '--dry-run'. Individual certificates can be renewed with
  the '--cert-name' option. Hooks are available to run commands before and
  after renewal; see https://certbot.eff.org/docs/using.html#renewal for
  more information on these.

  --pre-hook PRE_HOOK   Command to be run in a shell before obtaining any
                      certificates. Unless --disable-hook-validation is
                      used, the command's first word must be the absolute
                      pathname of an executable or one found via the PATH
                      environment variable. Intended primarily for renewal,
                      where it can be used to temporarily shut down a
                      webserver that might conflict with the standalone
                      plugin. This will only be called if a certificate is
                      actually to be obtained/renewed. When renewing several
                      certificates that have identical pre-hooks, only the
                      first will be executed. (default: None)
  --post-hook POST_HOOK
                      Command to be run in a shell after attempting to
                      obtain/renew certificates. Unless --disable-hook-
                      validation is used, the command's first word must be
                      the absolute pathname of an executable or one found
                      via the PATH environment variable. Can be used to
                      deploy renewed certificates, or to restart any servers
                      that were stopped by --pre-hook. This is only run if
                      an attempt was made to obtain/renew a certificate. If
```

multiple renewed certificates have identical post-hooks, only one will be run. (default: None)

--deploy-hook DEPLOY_HOOK
Command to be run in a shell once for each successfully issued certificate. Unless --disable-hook-validation is used, the command's first word must be the absolute pathname of an executable or one found via the PATH environment variable. For this command, the shell variable $RENEWED_LINEAGE will point to the config live subdirectory (for example, "/etc/letsencrypt/live/example.com") containing the new certificates and keys; the shell variable $RENEWED_DOMAINS will contain a space-delimited list of renewed certificate domains (for example, "example.com www.example.com") (default: None)

--disable-hook-validation
Ordinarily the commands specified for --pre-hook/--post-hook/--deploy-hook will be checked for validity, to see if the programs being run are in the $PATH, so that mistakes can be caught early, even when the hooks aren't being run just yet. The validation is rather simplistic and fails if you use more advanced shell constructs, so you can use this switch to disable it. (default: False)

--no-directory-hooks Disable running executables found in Certbot's hook directories during renewal. (default: False)

--disable-renew-updates
Disable automatic updates to your server configuration that would otherwise be done by the selected installer plugin, and triggered when the user executes "certbot renew", regardless of if the certificate is renewed. This setting does not apply to important TLS configuration updates. (default: False)

--no-autorenew Disable auto renewal of certificates. (default: False)

certificates:
List certificates managed by Certbot

delete:
Options for deleting a certificate

revoke:
Options for revocation of certificates

--reason {unspecified,keycompromise,affiliationchanged,superseded,cessationofoperation}
Specify reason for revoking certificate. (default: unspecified)
--delete-after-revoke
Delete certificates after revoking them, along with all previous and later versions of those certificates. (default: None)
--no-delete-after-revoke
Do not delete certificates after revoking them. This option should be used with caution because the 'renew' subcommand will attempt to renew undeleted revoked certificates. (default: None)

register:
Options for account registration

--register-unsafely-without-email
Specifying this flag enables registering an account with no email address. This is strongly discouraged, because you will be unable to receive notice about impending expiration or revocation of your certificates or problems with your Certbot installation that will lead to failure to renew. (default: False)
-m EMAIL, --email EMAIL
Email used for registration and recovery contact. Use comma to register multiple emails, ex: u1@example.com,u2@example.com. (default: Ask).
--eff-email Share your e-mail address with EFF (default: None)
--no-eff-email Don't share your e-mail address with EFF (default: None)

update_account:
Options for account modification

unregister:
Options for account deactivation.

```
  --account ACCOUNT_ID  Account ID to use (default: None)

install:
  Options for modifying how a certificate is deployed

rollback:
  Options for rolling back server configuration changes

  --checkpoints N       Revert configuration N number of checkpoints.
                        (default: 1)

plugins:
  Options for the "plugins" subcommand

  --init                Initialize plugins. (default: False)
  --prepare             Initialize and prepare plugins. (default: False)
  --authenticators      Limit to authenticator plugins only. (default: None)
  --installers          Limit to installer plugins only. (default: None)

enhance:
  Helps to harden the TLS configuration by adding security enhancements to
  already existing configuration.

show_account:
  Options useful for the "show_account" subcommand:

reconfigure:
  Common options that may be updated with the "reconfigure" subcommand:

plugins:
  Plugin Selection: Certbot client supports an extensible plugins
  architecture. See 'certbot plugins' for a list of all installed plugins
  and their names. You can force a particular plugin by setting options
  provided below. Running --help <plugin_name> will list flags specific to
  that plugin.

  --configurator CONFIGURATOR
                        Name of the plugin that is both an authenticator and
                        an installer. Should not be used together with
                        --authenticator or --installer. (default: Ask)
```

-a AUTHENTICATOR, --authenticator AUTHENTICATOR
Authenticator plugin name. (default: None)
-i INSTALLER, --installer INSTALLER
Installer plugin name (also used to find domains). (default: None)
--apache Obtain and install certificates using Apache (default: False)
--nginx Obtain and install certificates using Nginx (default: False)
--standalone Obtain certificates using a "standalone" webserver. (default: False)
--manual Provide laborious manual instructions for obtaining a certificate (default: False)
--webroot Obtain certificates by placing files in a webroot directory. (default: False)
--dns-cloudflare Obtain certificates using a DNS TXT record (if you are using Cloudflare for DNS). (default: False)
--dns-digitalocean Obtain certificates using a DNS TXT record (if you are using DigitalOcean for DNS). (default: False)
--dns-dnsimple Obtain certificates using a DNS TXT record (if you are using DNSimple for DNS). (default: False)
--dns-dnsmadeeasy Obtain certificates using a DNS TXT record (if you are using DNS Made Easy for DNS). (default: False)
--dns-gehirn Obtain certificates using a DNS TXT record (if you are using Gehirn Infrastructure Service for DNS). (default: False)
--dns-google Obtain certificates using a DNS TXT record (if you are using Google Cloud DNS). (default: False)
--dns-linode Obtain certificates using a DNS TXT record (if you are using Linode for DNS). (default: False)
--dns-luadns Obtain certificates using a DNS TXT record (if you are using LuaDNS for DNS). (default: False)
--dns-nsone Obtain certificates using a DNS TXT record (if you are using NS1 for DNS). (default: False)
--dns-ovh Obtain certificates using a DNS TXT record (if you are using OVH for DNS). (default: False)
--dns-rfc2136 Obtain certificates using a DNS TXT record (if you are using BIND for DNS). (default: False)
--dns-route53 Obtain certificates using a DNS TXT record (if you are using Route53 for DNS). (default: False)

```
  --dns-sakuracloud     Obtain certificates using a DNS TXT record (if you are
                        using Sakura Cloud for DNS). (default: False)

apache:
  Apache Web Server plugin (Please note that the default values of the
  Apache plugin options change depending on the operating system Certbot is
  run on.)

  --apache-enmod APACHE_ENMOD
                        Path to the Apache 'a2enmod' binary (default: None)
  --apache-dismod APACHE_DISMOD
                        Path to the Apache 'a2dismod' binary (default: None)
  --apache-le-vhost-ext APACHE_LE_VHOST_EXT
                        SSL vhost configuration extension (default: -le-
                        ssl.conf)
  --apache-server-root APACHE_SERVER_ROOT
                        Apache server root directory (default: /etc/apache2)
  --apache-vhost-root APACHE_VHOST_ROOT
                        Apache server VirtualHost configuration root (default:
                        None)
  --apache-logs-root APACHE_LOGS_ROOT
                        Apache server logs directory (default:
                        /var/log/apache2)
  --apache-challenge-location APACHE_CHALLENGE_LOCATION
                        Directory path for challenge configuration (default:
                        /etc/apache2)
  --apache-handle-modules APACHE_HANDLE_MODULES
                        Let installer handle enabling required modules for you
                        (Only Ubuntu/Debian currently) (default: False)
  --apache-handle-sites APACHE_HANDLE_SITES
                        Let installer handle enabling sites for you (Only
                        Ubuntu/Debian currently) (default: False)
  --apache-ctl APACHE_CTL
                        Full path to Apache control script (default:
                        apache2ctl)
  --apache-bin APACHE_BIN
                        Full path to apache2/httpd binary (default: None)

dns-cloudflare:
  Obtain certificates using a DNS TXT record (if you are using Cloudflare
```

for DNS).

--dns-cloudflare-propagation-seconds DNS_CLOUDFLARE_PROPAGATION_SECONDS
The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 10)
--dns-cloudflare-credentials DNS_CLOUDFLARE_CREDENTIALS
Cloudflare credentials INI file. (default: None)

dns-digitalocean:
Obtain certificates using a DNS TXT record (if you are using DigitalOcean for DNS).

--dns-digitalocean-propagation-seconds DNS_DIGITALOCEAN_PROPAGATION_SECONDS
The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 10)
--dns-digitalocean-credentials DNS_DIGITALOCEAN_CREDENTIALS
DigitalOcean credentials INI file. (default: None)

dns-dnsimple:
Obtain certificates using a DNS TXT record (if you are using DNSimple for DNS).

--dns-dnsimple-propagation-seconds DNS_DNSIMPLE_PROPAGATION_SECONDS
The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 30)
--dns-dnsimple-credentials DNS_DNSIMPLE_CREDENTIALS
DNSimple credentials INI file. (default: None)

dns-dnsmadeeasy:
Obtain certificates using a DNS TXT record (if you are using DNS Made Easy for DNS).

--dns-dnsmadeeasy-propagation-seconds DNS_DNSMADEEASY_PROPAGATION_SECONDS
The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 60)
--dns-dnsmadeeasy-credentials DNS_DNSMADEEASY_CREDENTIALS

DNS Made Easy credentials INI file. (default: None)

dns-gehirn:
Obtain certificates using a DNS TXT record (if you are using Gehirn Infrastructure Service for DNS).

--dns-gehirn-propagation-seconds DNS_GEHIRN_PROPAGATION_SECONDS
The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 30)
--dns-gehirn-credentials DNS_GEHIRN_CREDENTIALS
Gehirn Infrastructure Service credentials file. (default: None)

dns-google:
Obtain certificates using a DNS TXT record (if you are using Google Cloud DNS for DNS).

--dns-google-propagation-seconds DNS_GOOGLE_PROPAGATION_SECONDS
The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 60)
--dns-google-credentials DNS_GOOGLE_CREDENTIALS
Path to Google Cloud DNS service account JSON file. (See https://developers.google.com/identity/protocols/OAuth2ServiceAccount#creatinganaccount forinformation about creating a service account and https://cloud.google.com/dns/access-control#permissions_and_roles for information about therequired permissions.) (default: None)

dns-linode:
Obtain certificates using a DNS TXT record (if you are using Linode for DNS).

--dns-linode-propagation-seconds DNS_LINODE_PROPAGATION_SECONDS
The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 120)
--dns-linode-credentials DNS_LINODE_CREDENTIALS

Linode credentials INI file. (default: None)

dns-luadns:
Obtain certificates using a DNS TXT record (if you are using LuaDNS for DNS).

--dns-luadns-propagation-seconds DNS_LUADNS_PROPAGATION_SECONDS
The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 30)
--dns-luadns-credentials DNS_LUADNS_CREDENTIALS
LuaDNS credentials INI file. (default: None)

dns-nsone:
Obtain certificates using a DNS TXT record (if you are using NS1 for DNS).

--dns-nsone-propagation-seconds DNS_NSONE_PROPAGATION_SECONDS
The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 30)
--dns-nsone-credentials DNS_NSONE_CREDENTIALS
NS1 credentials file. (default: None)

dns-ovh:
Obtain certificates using a DNS TXT record (if you are using OVH for DNS).

--dns-ovh-propagation-seconds DNS_OVH_PROPAGATION_SECONDS
The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 120)
--dns-ovh-credentials DNS_OVH_CREDENTIALS
OVH credentials INI file. (default: None)

dns-rfc2136:
Obtain certificates using a DNS TXT record (if you are using BIND for DNS).

--dns-rfc2136-propagation-seconds DNS_RFC2136_PROPAGATION_SECONDS
The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS

record. (default: 60)
--dns-rfc2136-credentials DNS_RFC2136_CREDENTIALS
RFC 2136 credentials INI file. (default: None)

dns-route53:
Obtain certificates using a DNS TXT record (if you are using AWS Route53 for DNS).

dns-sakuracloud:
Obtain certificates using a DNS TXT record (if you are using Sakura Cloud for DNS).

--dns-sakuracloud-propagation-seconds DNS_SAKURACLOUD_PROPAGATION_SECONDS
The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 90)
--dns-sakuracloud-credentials DNS_SAKURACLOUD_CREDENTIALS
Sakura Cloud credentials file. (default: None)

manual:
Authenticate through manual configuration or custom shell scripts. When using shell scripts, an authenticator script must be provided. The environment variables available to this script depend on the type of challenge. $CERTBOT_DOMAIN will always contain the domain being authenticated. For HTTP-01 and DNS-01, $CERTBOT_VALIDATION is the validation string, and $CERTBOT_TOKEN is the filename of the resource requested when performing an HTTP-01 challenge. An additional cleanup script can also be provided and can use the additional variable $CERTBOT_AUTH_OUTPUT which contains the stdout output from the auth script. For both authenticator and cleanup script, on HTTP-01 and DNS-01 challenges, $CERTBOT_REMAINING_CHALLENGES will be equal to the number of challenges that remain after the current one, and $CERTBOT_ALL_DOMAINS contains a comma-separated list of all domains that are challenged for the current certificate.

--manual-auth-hook MANUAL_AUTH_HOOK
Path or command to execute for the authentication script (default: None)
--manual-cleanup-hook MANUAL_CLEANUP_HOOK
Path or command to execute for the cleanup script

(default: None)

nginx:
Nginx Web Server plugin

--nginx-server-root NGINX_SERVER_ROOT
Nginx server root directory. (default: /etc/nginx or /usr/local/etc/nginx)
--nginx-ctl NGINX_CTL
Path to the 'nginx' binary, used for 'configtest' and retrieving nginx version number. (default: nginx)
--nginx-sleep-seconds NGINX_SLEEP_SECONDS
Number of seconds to wait for nginx configuration changes to apply when reloading. (default: 1)

null:
Null Installer

standalone:
Runs an HTTP server locally which serves the necessary validation files under the /.well-known/acme-challenge/ request path. Suitable if there is no HTTP server already running. HTTP challenge only (wildcards not supported).

webroot:
Saves the necessary validation files to a .well-known/acme-challenge/ directory within the nominated webroot path. A seperate HTTP server must be running and serving files from the webroot path. HTTP challenge only (wildcards not supported).

--webroot-path WEBROOT_PATH, -w WEBROOT_PATH
public_html / webroot path. This can be specified multiple times to handle different domains; each domain will have the webroot path that preceded it. For instance: '-w /var/www/example -d example.com -d www.example.com -w /var/www/thing -d thing.net -d m.thing.net' (default: Ask)
--webroot-map WEBROOT_MAP
JSON dictionary mapping domains to webroot paths; this implies -d for each entry. You may need to escape this

from your shell. E.g.: --webroot-map
'{"eg1.is,m.eg1.is":"/www/eg1/", "eg2.is":"/www/eg2"}'
This option is merged with, but takes precedence over,
-w / -d entries. At present, if you put webroot-map in
a config file, it needs to be on a single line, like:
webroot-map = {"example.com":"/var/www"}. (default:
{})

AUTHOR

Certbot