# Velocity HTTPS certificates

How Velocity serves HTTPS for a hosted site: certificates from a file or a panel, Let's Encrypt and other ACME CAs, archives and PKCS#12 bundles, the self-signed stand-in, live swap on renewal, HSTS, monitoring expiry…

Source file: `doc/eng-us/help/hosting/servers/velocity-https-certificates.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/hosting/servers/velocity-https-certificates.md

This page is for system administrators at hosting companies who let Velocity serve HTTPS for customer sites. It answers: which certificate sources Velocity accepts, how to configure each, what happens on renewal and failure, how to watch expiry across many sites, and what must be backed up. The sources are Chapter 8, section 8.3.9 and Velocity HTTPS and certificates.

## Contents

1. What you can rely on
2. Choosing a certificate source
3. A certificate you already have
4. Let's Encrypt and other ACME CAs
5. Strict-Transport-Security
6. Watching expiry
7. What to back up
8. Troubleshooting
9. Checklist

## What you can rely on

From the feature page:

- HTTPS comes up before plain HTTP, so there is never a moment when the server answers HTTP but not HTTPS.
- A broken or expired certificate never reaches a visitor: the pair is checked (parses, not expired, key belongs to it) before use, and a **self-signed certificate** stands in until the real one is usable again (the engine's `fallback`, default `self-signed`; `none` switches HTTPS off instead).
- The server reads a private copy of the pair, so a file being rewritten elsewhere cannot break a handshake. Every `watchInterval` seconds (60) the source is checked and a changed, valid pair is swapped in live, with no restart and no dropped connection.
- The certificate that covers each domain is chosen per connection (SNI).
- HTTP/2 comes with it.

## Choosing a certificate source

| Source (`mode`) | Use it when | Configured in |
|---|---|---|
| `manual` / `files` | the panel, certbot or a CA gives you PEM files | `velocity.ini [HTTPSSettings]` |
| `letsencrypt` (also `acme`) | Velocity should obtain and renew the certificate itself | a snippet in `/etc/vc/conf-available/` |
| `certbot` | certbot already manages the certificate | a snippet |
| `archive` | the CA delivers a `.zip`, `.tar.gz`, `.7z` or similar | a snippet |
| `pkcs12` | you have a `.p12` / `.pfx` bundle | a snippet |
| `remote` | the certificate is downloaded from a URL | a snippet |
| `self-signed` | development only | a snippet |

`velocity.ini` exposes only the file pair; the other modes are set under `Q.web.https` in the engine's configuration, for which a file in `conf-available/` is the place (it belongs to the administrator and is merged before the generated site file). In that case leave `[HTTPSSettings] Enabled` at `false`.

## A certificate you already have

Any PEM certificate from a CA or a hosting panel (leaf with chain, and the private key):

**Shell**

```
php bin/php/console exp:velocity config set --apply HTTPSSettings Certificate /etc/ssl/example.com/fullchain.pem --allow-root-user
php bin/php/console exp:velocity config set --apply HTTPSSettings Key /etc/ssl/example.com/privkey.pem --allow-root-user
php bin/php/console exp:velocity config set --apply HTTPSSettings Enabled true --allow-root-user
php bin/php/console exp:velocity restart --allow-root-user
php bin/php/console exp:velocity ssl show --allow-root-user
curl -sI https://example.com/ | head -1
```

- HTTPS is switched on only when `Enabled` is `true` **and** both files exist; a missing path leaves the server on plain HTTP and `status` says what switches it on.
- The server process reads the key as root when started as root, so the key can stay `0600 root`.
- A renewal by your own tool (or the panel) is picked up within a minute, without a restart.
- On the shipped ports HTTPS is `8080` and HTTP `8088`, both listening; `velocity.ini` has no setting that redirects one to the other. Make sure the addresses the site generates are HTTPS (`site.ini [SiteSettings] SiteURL` and the siteaccesses' URLs).

## Let's Encrypt and other ACME CAs

Requirements for the HTTP-01 challenge: DNS of every name points to this machine, and port 80 of this server is reachable from the internet (`[ServerSettings] Port 80`). From chapter 8:

Shell

```
cat > /etc/vc/conf-available/https-acme.conf <<'EOF'
{ "Q": { "web": { "https": {
  "port": 443,
  "mode": "letsencrypt",
  "acme": {
    "email": "hostmaster@example.com",
    "domains": ["example.com", "www.example.com"],
    "directory": "letsencrypt-staging"
  }
} } } }
EOF
php bin/php/console exp:velocity conf enable --apply https-acme --allow-root-user
php bin/php/console exp:velocity restart --allow-root-user
php bin/php/console exp:velocity ssl show --allow-root-user
```

- Start with `letsencrypt-staging`: its certificates are not trusted by browsers, but its rate limits are generous. When `ssl show` lists a staging certificate, remove that line (the default is `letsencrypt`) and restart.
- Until the first certificate arrives, the self-signed one is served; the CA certificate replaces it live.
- Renewal runs in the background when a third of the lifetime is left (`renewAt` 0.33), with growing pauses after a failure (5 minutes doubling up to a day), so rate limits are never spent. `exp:velocity ssl renew [host...]` asks for it now.
- `directory` also takes `zerossl`, `google`, `google-staging` or any ACME directory URL; `eab` holds external account binding. `keyType` defaults to `ec256`.
- Wildcards need `challenge: dns-01` with a `dnsHook` program. When another server owns port 80, set `webroot` to a directory that server serves.
- `start` reports that the configuration differs from `velocity.ini` because of the enabled snippet; that is expected.

> **Tip**
>
> On a host with many sites, use one snippet per installation and the customer's domains only. A shared snippet that lists another customer's names would request certificates for them from the wrong machine.

## Strict-Transport-Security

INI

```
[HTTPSSettings]
HSTSMaxAge=300              # seconds; 0 sends no header
HSTSIncludeSubDomains=disabled
HSTSPreload=disabled
```

The header goes out on every HTTPS answer and never over plain HTTP. A browser that has seen it refuses plain HTTP to that host for the whole period, even after you lower the value. Raise it to a year (`31536000`) only when every name of the site is served over HTTPS for good; preload needs `HSTSIncludeSubDomains` and at least a year. Agree this with the customer: it outlives a move to another host.

## Watching expiry

**Shell**

```
php bin/php/console exp:velocity ssl show --allow-root-user          # served certificate, every known one by expiry
php bin/php/console exp:velocity ssl show --json --allow-root-user   # for monitoring: watch daysLeft
```

A 90-day CA certificate is renewed when about 30 days are left, so one with fewer than 14 days left means renewal has been failing for more than two weeks; the last error, which `ssl show` prints with the ACME state, says why. Keep NTP running: certificates are checked against the system clock. The control panel's **SSL** tab shows the same and can renew and reload from a browser. Alert on `daysLeft` below 14 for every customer site; see Certificate expiry.

## What to back up

Back up the `ssl/` directory of the configuration directory the server runs with. Everything in it can be made again except the **ACME account keys** (`acme/account-<ca>.pem`). The layout:

```
ssl/
  self-signed.pem, self-signed.key      the self-signed pair
  active-443-<fp>-<pid>.pem/.key        the copy one server reads (do not edit)
  imported/                             pairs imported from DER, archives, bundles, remote
  acme/account-<ca>.pem                 one account key per CA: keep it
  acme/<first domain>/fullchain.pem, privkey.pem, state.json
```

Private keys are written `0600`, atomically, and never logged.

## Troubleshooting

| Symptom | Cause | Fix |
|---|---|---|
| `status` says HTTPS is off | Enabled is not `true`, or a named file does not exist | check the paths with `ls -l` |
| browser warns about the certificate | the self-signed stand-in is served | `ssl show`; the start-up log names the reason |
| ACME `Invalid response from http://.../.well-known/acme-challenge/...` | port 80 does not reach this server | open port 80, or set `acme.webroot` for the server that owns it |
| `none of the certificates belongs to the private key` | key and certificate from different requests | use the key the certificate was issued for |
| `no private key found (is it encrypted?...)` | encrypted key, no password given | set `keyPassword`, `keyPasswordFile` or `keyPasswordEnv` |

## Checklist

- ☐ Each site serves a CA certificate (`ssl show`), not the self-signed stand-in.
- ☐ ACME tried against staging first; production directory set afterwards.
- ☐ Monitoring alerts on `daysLeft` below 14.

☐ NTP running on the machine.

☐ `ssl/` backed up, including the ACME account keys.

☐ `HSTSMaxAge` raised only with the customer's agreement and every name on HTTPS.

## Related pages

- Velocity for hosts
- TLS and HTTPS
- TLS certificates in panels
- Certificate expiry
- Velocity HTTPS and certificates
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.


> **Become an Exponential Partner**
>
> Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.
>
> Partners: start here | How to apply



Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.