Lesson 2.4: The response cache and HTTPS certificates

Learners prove a response-cache hit, explain what the cache never stores and how to clear it, then switch on HTTPS with a certificate pair and read it with exp:velocity ssl show. 45 minutes.

Source file:
`doc/eng-us/help/trainers/hosting-operations/module-02-velocity/response-cache-and-https.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/trainers/hosting-operations/module-02-velocity/response-cache-and-https.md

Two things Velocity does that a traditional PHP-FPM setup leaves to other tools: answering repeat anonymous pages from its own response cache, and serving HTTPS with its own certificates. Support staff meet both every week ("my edit does not show", "the certificate warning"), so this lesson gives learners the checks and the answers.

At a glance

Course	Hosting and operations
Module	Module 2: Velocity
Duration	45 minutes (20 presenting, 25 exercise)
Level	Intermediate
Audience	System administrators and hosting support staff
Prerequisites	Lesson 2.3; `openssl` on the learner's machine

Learning objectives

After this lesson, learners can:

- show a response-cache hit with two requests and read `X-Cache` and `Age`;
- list what the response cache never stores, including any page for a signed-in visitor;
- clear the response cache and say when a restart is not needed;
- switch on HTTPS with a certificate pair and read the served certificate with `exp:velocity ssl show`;
- explain the self-signed stand-in, live certificate swaps and the risk of a long HSTS period.

Trainer's notes

How to present it

Source: Velocity response cache, Velocity HTTPS certificates and Chapter 8, section 8.3.9.

1. **What the cache does** (4 minutes). Rendered anonymous pages are answered inside the

server process before a worker is chosen: no fork, no PHP, no database. Conditional requests get `304`; an expired page is still served while one request renders the replacement (`StaleWhileRevalidate`, 60 s).

2. **What it never stores** (3 minutes). Signed-in visitors are never served from it; requests with the session cookie or an `Authorization` header; responses that set a cookie or say `no-store` or `private`; anything but `GET`; non-200 answers (a 404 only when `NotFoundSeconds` is above 0).
3. **Freshness** (3 minutes). `DefaultTtl` (30 s) is how old a page a visitor may see after an edit. Clear with `exp:velocity cache clear`; after a template change no restart is needed. The browser navigation cache means a published change can appear on the second view.
4. **HTTPS** (6 minutes). With a pair of PEM files: set `[HTTPSSettings] Certificate`, `Key` and `Enabled true`, restart. The pair is checked before use; a pair that fails is never shown, a self-signed certificate stands in. A renewed file is swapped in live within a minute. Let's Encrypt and other sources are set in a snippet in `conf-available/`; port 80 must reach the server for the HTTP-01 challenge.
5. **HSTS and backups** (4 minutes). A browser that has seen `Strict-Transport-Security` refuses plain HTTP for the whole period; raise `HSTSMaxAge` to a year only when every name is on HTTPS for good. Back up the `ssl/` directory; only the ACME account keys cannot be made again.

Common questions

A customer says signed-in users see an old page. Not the response cache: signed-in visitors never get it. Look at the view cache or the HTTP cache.

Why are 404s not cached by default? `NotFoundSeconds=0` sends every 404 to PHP, so `404.log` and the security log see each one. Keep it at 0 where security monitoring reads those logs.

How early should I worry about expiry? A 90-day certificate is renewed when about 30 days are left; fewer than 14 days left means renewal has been failing for two weeks. Alert on `daysLeft` below 14.

Does the shipped configuration redirect HTTP to HTTPS? `velocity.ini` has no setting that redirects one to the other; both listen at once. Make the site's own addresses HTTPS ones before raising HSTS.

Pitfalls

- **Testing while signed in.** A browser with an administration session never sees a cache hit. Use `curl`.
- **ACME in class.** Let's Encrypt needs public DNS and port 80 from the internet. Use a self-made pair in class, as in the exercise, and show ACME on the projector only.
- **HSTS on a training host name.** Do not set `HSTSMaxAge` above its shipped value on a name learners reuse.

Hands-on exercise

Use your own training installation, served by Velocity on the shipped ports (8088 and 8080).

1. Request the front page twice, without signing in:

   Shell

   ```
   curl -sI http://127.0.0.1:8088/ | grep -iE 'x-cache|age'
   curl -sI http://127.0.0.1:8088/ | grep -iE 'x-cache|age'
   ```

2. Show the cache's state: `php bin/php/console exp:velocity cache stats`.
3. Clear it: `php bin/php/console exp:velocity cache clear --apply`, then repeat step 1 once.
4. Make a self-signed pair for the exercise in a directory of your installation:

   Shell

   ```
   mkdir -p var/training-tls
   openssl req -x509 -newkey rsa:2048 -nodes -days 30 -subj "/CN=training.example.invalid" \
     -keyout var/training-tls/key.pem -out var/training-tls/cert.pem
   ```

5. Point Velocity at the pair and switch HTTPS on:

   Shell

   ```
   php bin/php/console exp:velocity config set --apply HTTPSSettings Certificate var/training-tl
   s/cert.pem
   php bin/php/console exp:velocity config set --apply HTTPSSettings Key var/training-tls/key.pe
   m
   php bin/php/console exp:velocity config set --apply HTTPSSettings Enabled true
   php bin/php/console exp:velocity restart
   ```

6. Read `php bin/php/console exp:velocity status` and find the HTTPS address.
7. Show the served certificate: `php bin/php/console exp:velocity ssl show`.
8. Request the site over HTTPS, accepting the self-made certificate: `curl -skI https://127.0.0.1:8080/ | head -1`.
9. Switch HTTPS off again with `config set HTTPSSettings Enabled false` and restart.

Expected result

- Step 1: the second answer carries `X-Cache: HIT` and an `Age` header.
- Step 3: the first request after clearing is not a hit; the next one is again.
- Step 7 lists the certificate for `training.example.invalid` with its expiry date and days left.
- Step 8 prints `HTTP/2 200` or a redirect.
- The trainer checks by asking for the days left that `ssl show` printed.

Quiz

1. Which header shows that a page came from the response cache?
2. A signed-in editor sees an old page. Is the response cache the cause?
3. What does Velocity serve when the configured certificate has expired? A. nothing, HTTPS

stops B. the expired certificate C. a self-signed stand-in D. plain HTTP on port 443

4. A renewal tool replaces the certificate file. What must the operator do?
5. Which file in the `ssl/` directory cannot be made again and must be backed up?

Answers

1. `X-Cache: HIT`, together with an `Age` header.
2. No. Signed-in visitors are never served from it; look at the view cache or the HTTP cache.
3. **C.** A pair that fails the check is never shown; the self-signed certificate stands in until the real one is usable (Velocity HTTPS certificates).
4. Nothing: a changed, valid pair is swapped in live within `watchInterval` (60 seconds), without a restart.
5. The ACME account keys (`acme/account-<ca>.pem`).

Further reading

- Velocity response cache: settings and support answers.
- Velocity HTTPS certificates: every certificate source.
- Velocity HTTPS and certificates: the feature page.
- Certificate expiry: the runbook.
- TLS and HTTPS: HTTPS for hosts in general.

Related pages

- Module 2: Velocity
- Hosting and operations: course overview
- Trainers: start here
- 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.