

EasyEngine CLI — Course Outline for a WordPress Developer ?

Below is a **chaptered, addressable syllabus** you can reference like `c2.6` (Chapter 2, Subsection 6). It's designed around what you shared from the GitHub docs (site management for **HTML/WP**, **cron**, and **shell**) and also adds the *practical, WordPress-on-a-root-server* knowledge you'll want alongside those commands. ☐

c1) Orientation: What EasyEngine *is* (and why it changes your workflow) ?

c1.1 — EasyEngine mental model (shared hosting vs root server vs containers)

c1.2 — What EasyEngine manages for you (sites, containers, services, SSL)

c1.3 — Core CLI structure: `ee <noun> <verb> [args] [options]`

c1.4 — Local machine vs server: where you run `ee` and what it touches

c1.5 — Learning map: what you'll be able to do by the end of the course

c2) Site Lifecycle Fundamentals (HTML + WordPress) ??

c2.1 — Naming and identifying sites (`<site-name>` conventions, DNS expectations)

c2.2 — Creating sites: the shared mechanics across types

- What `--type` means conceptually
- When `--skip-status-check` matters and when it's risky

c2.3 — `ee site create --type=html`

- c2.3.1 Minimal create
- c2.3.2 Create with SSL (`--ssl=le`, `--ssl=self`)
- c2.3.3 Wildcard SSL (`--wildcard`) and when you actually need it

c2.4 — `ee site create --type=wp` (WordPress install paths)

- c2.4.1 Minimal WP create
- c2.4.2 WP + Let's Encrypt SSL
- c2.4.3 WP + wildcard SSL
- c2.4.4 WP + Redis cache (`--cache`) and what it implies
- c2.4.5 WP multisite (`--mu=subdir` vs `--mu=subdom`)
- c2.4.6 VIP flow (`--vip`) and using a repo containing `wp-content`
- c2.4.7 Skipping steps (`--skip-install`, `--skip-content`) and why/when
- c2.4.8 Version selection (`--version=latest|nightly|x.y.z`)

c2.5 — Managing a site’s runtime state

- c2.5.1 `ee site enable` (and what “enable” means in practice)
- c2.5.2 `ee site disable` (what stops/removes; what persists)
- c2.5.3 `--verify` and dependent “global services”
- c2.5.4 `--force` usage patterns (and cautions)

c2.6 — Updating an existing site: `ee site update`

- c2.6.1 Adding SSL to a non-SSL site (`--ssl=le|self`)
- c2.6.2 Switching to wildcard (`--wildcard`)
- c2.6.3 Safe operational routine (pre-checks, downtime avoidance mindset)

c2.7 — SSL operations: `ee site ssl`

- c2.7.1 Challenge verification concepts
- c2.7.2 Renewal behavior and `--force`
- c2.7.3 Practical renewal troubleshooting checklist

c2.8 — Inspecting & inventorying: `ee site info` and `ee site list`

- c2.8.1 `ee site info` output: credentials, links, what to look for
- c2.8.2 `ee site list` filters (`--enabled`, `--disabled`)
- c2.8.3 Output formats (`--format=table|csv|yaml|json|count|text`)

c2.9 — Reload/restart: `ee site reload` and `ee site restart`

- c2.9.1 Reload vs restart (what you use each for)
- c2.9.2 “Type” variants in docs (`--type=html|wp`) and expected behavior

c2.10 — Deleting sites: `ee site delete`

- c2.10.1 Confirmation and `--yes`
- c2.10.2 Data safety: what to back up *before* deletion

c2.11 — Caching operations: `ee site clean`

- c2.11.1 Clearing both caches (default behavior)
- c2.11.2 `--page` vs `--object` (when to clear which)

c2.12 — Sharing a site online: `ee site share` (ngrok)

- c2.12.1 Create a share link
 - c2.12.2 `--refresh` behavior
 - c2.12.3 Taking it down (`--disable`)
 - c2.12.4 Using `--token` (ngrok token hygiene)
-

c3) WordPress Provisioning Details (Deep Dive) ?

(Still based on your doc excerpt, but expanded into the decisions you'll actually make.)

c3.1 — Admin bootstrap options (`--title`, `--admin-user`, `--admin-pass`, `--admin-email`)

c3.2 — PHP version selection (`--php=5.6|7.2|latest`) and compatibility strategy

c3.3 — Database configuration patterns

- c3.3.1 Defaults vs explicit config (`--dbname`, `--dbuser`, `--dbpass`, etc.)
- c3.3.2 Remote DB host (`--dbhost`) implications
- c3.3.3 `--skip-check` and when it can bite you
- c3.3.4 `--force` and “reset remote database if not empty”

c3.4 — Local DB & local Redis containers (`--local-db`, `--with-local-redis`)

c3.5 — Multisite operations mindset (subdir vs subdom prerequisites)

c3.6 — VIP workflow (`--vip`) as a deployment pattern

c4) Cron Management with `ee cron` ??

c4.1 — What EasyEngine cron controls (site-level vs host-level)

c4.2 — Listing jobs: `ee cron list [<site-name>] [--all]`

c4.3 — Creating jobs: `ee cron create`

- c4.3.1 Site cron vs host cron (`<site-name>` vs `host`)
- c4.3.2 `--command` quoting rules and practical examples
- c4.3.3 `--schedule` formats:
 - Classic Linux cron strings
 - Macros like `@weekly`, `@daily`, etc.
 - `@every <duration>` (e.g., `@every 10m`, combined units)

c4.4 — Running immediately: `ee cron run-now <cron-id>`

c4.5 — Updating jobs: `ee cron update <id>`

- c4.5.1 Changing site, command, schedule, and user

c4.6 — Deleting jobs: `ee cron delete <cron-id>`

c4.7 — WordPress-specific cron recipes (WP-Cron alternatives, due-now runners)

c5) Shell Access with `ee shell` (Your daily driver) ?

c5.1 — Interactive shell into a site container: `ee shell <site-name>`

c5.2 — Choosing the user: `--user=root|www-data|...` (practical safety rules)

c5.3 — Choosing a service: `--service=php` (default) vs `--service=nginx`, etc.

c5.4 — Non-interactive execution: `--command='...'`

c5.5 — TTY considerations: `--skip-tty` (when automation needs it)

c5.6 — Common WP developer workflows via `ee shell`

- Running WP-CLI operations
 - Inspecting logs/configs
 - Quick Nginx tests/reloads
-

c6) Operational Routines (Putting commands into real workflows) ?

c6.1 — “Day 1” routine: create ? SSL ? verify ? info ? share (optional)

c6.2 — “Release” routine: maintenance mode, cache clear, reload/restart

c6.3 — “Incident” routine: disable/enable, verify dependencies, SSL renewal

c6.4 — Safe defaults vs dangerous flags (`--force`, `--skip-check`, `--skip-status-check`)

c6.5 — Auditing & reporting with `ee site list --format=...`

c7) WordPress Performance & Architecture Tuning (EE-centric mindset) ?

c7.1 — Page cache vs object cache: what you clear and why

c7.2 — Redis in practice (`--cache`, `--with-local-redis`)

c7.3 — PHP version strategy for client projects

c7.4 — Multisite scaling considerations

c7.5 — When to prefer local DB container vs global/remote DB

c8) Security & Access Patterns (Root server reality check) ?

c8.1 — SSL choices (`le` vs `self`) and practical environments

c8.2 — Principle of least privilege for `ee shell`
`--user=...`

c8.3 — ngrok sharing: token handling and exposure control

c8.4 — Deletion safeguards and backup checkpoints

c9) Capstone Projects (guided practice paths) ?

c9.1 — Build a production-like WP site with SSL + cron runner + caching

c9.2 — Build a multisite (subdir/subdom) with correct SSL and cron strategy

c9.3 — VIP-style site: `--vip` repo workflow + operational commands

c9.4 — Maintenance toolkit: scripts using `ee shell --command, ee site list --format=json`

c10) Quick Reference Index (command ?topic map) ?

1. c2.3 — `ee site create --type=html`
 2. c2.4 / c3 — `ee site create --type=wp`
 3. c2.10 — `ee site delete`
 4. c2.6 — `ee site update`
 5. c2.5 — `ee site enable|disable`
 6. c2.7 — `ee site ssl`
 7. c2.8 — `ee site info|list`
 8. c2.9 — `ee site reload|restart`
 9. c2.11 — `ee site clean`
 10. c2.12 — `ee site share`
2. **Cron**
 1. c4.3 — `ee cron create`
 2. c4.2 — `ee cron list`
 3. c4.5 — `ee cron update`
 4. c4.6 — `ee cron delete`
 5. c4.4 — `ee cron run-now`
 3. **Shell**
 1. c5 — `ee shell`
-

Next step ?

Tell me where you want to start:

1. `c2.4` (WordPress site creation end-to-end), or
 2. `c5` (shell workflows; fastest way to feel productive), or
 3. Name any code like `c2.6` and I'll "execute" that chapter in lesson format.
-

Revision #2

Created 2026-04-15 16:17:09 UTC by art10m

Updated 2026-04-15 16:17:50 UTC by art10m