Magento Cheat Sheet

Magento Open Source / Adobe Commerce 2.4.9 (GA 2026-05-12) · Mage-OS 3.6.0 (2026-10-14)
Checked 2026-10-27 · 2.4.10 is still alpha (2.4.10-alpha1, 2026-10-13)

EDITIONS: WHAT EACH ALLOWS

Open Source	Free code base (OSL-3.0, AFL-3.0), self-hosted, in-process PHP only. No I/O Events, Upgrade Compatibility Tool, B2B or extended-support patches. `repo.magento.com` asks for Marketplace keys.
Mage-OS	Community distribution, `repo.mage-os.org`, no keys; PHP `~8.3||~8.4||~8.5`; no eventing modules.
Commerce on-prem	Licensed, you operate it. In- and out-of-process (events, webhooks, App Builder), B2B, extended support.
Commerce on Cloud	PaaS, single-tenant, Fastly included, manual updates, enforced deadlines.
Cloud Service	SaaS, versionless, **out-of-process only**, no Luma, Edge Delivery storefront, Adobe IMS.

VERSIONS, SUPPORT, ENFORCEMENT

Line	GA	Std. end	Ext. end
2.4.9	2026-05-12	2029-05-31	n/p
2.4.8	2025-04-08	2028-05-31	n/p
2.4.7	2024-04-09	2027-05-31	2028-05-31
2.4.6	2023-03-14	2026-08-11	2027-08-31
2.4.5	2022-08-09	2025-08-12	2026-08-11
2.4.4	2022-04-12	2025-04-12	2026-04-14

n/p = not published; extended support is Adobe Commerce only. Latest: 2.4.8-p5, 2.4.7-p10, 2.4.6-p15; 2.4.9 has no `-pN`, only monthly builds (`2.4.9-2026-oct`).

Cloud enforcement: dependencies by **2026-10-30** or 2027-05-31; 2.4.4/2.4.5 upgrade by 2027-06-01, 2.4.6/2.4.7 by 2028-06-01. Admin or traffic is suspended.

Patches: `-pN`; isolated patch (needs latest `-pN`); hotfix (any supported version); individual patch.

REQUIREMENTS (ON-PREM, LATEST PATCH)

	2.4.9	2.4.8-p5	2.4.7-p10
PHP	8.5 (8.4 upgrade only)	8.4, 8.3	8.3, 8.2
DB	MariaDB 12.3, MySQL 8.4	MariaDB 11.4/11.8, MySQL 8.4	MariaDB 10.11/11.8
Search	OpenSearch 3	OpenSearch 3, ES 8	OS 2.19/3, ES 8
Cache	Valkey 9	Valkey 8.1	Valkey 8.1
Other	RabbitMQ 4.3 or Artemis 2, Composer 2.10, Varnish 8, nginx 1.30		

2.4.9: native MVC replaces Laminas MVC, `symfony/cache` replaces `Zend_Cache`, HugeRTE replaces TinyMCE.

INSTALL

```
composer config -g http-basic.repo.magento.com
  "<public-key>" "<private-key>"
composer create-project \
  --repository-url=https://repo.magento.com/ \
  magento/project-community-edition=2.4.9 \
  /var/www/shop
# Commerce: magento/project-enterprise-edition
composer create-project \
  --repository-url=https://repo.mage-os.org/ \
  mage-os/project-community-edition=3.6.0 \
  /var/www/shop
bin/magento setup:install \
  --base-url=http://localhost/ \
  --db-host=db --db-name=magento \
  --db-user=magento --db-password='<pw>' \
  --admin-user=<user> --admin-password='<pw>' \
  --search-engine=opensearch \
  --opensearch-host=search \
  --cache-backend=valkey \
  --cache-backend-valkey-server=cache
# also --admin-email, -firstname, -lastname
```

Run `composer check-platform-reqs` first; `memory_limit=2G`, OPcache on, docroot `pub/`. A random Admin URI is chosen unless `--backend-frontname`; read it with `bin/magento info:adminuri`.

TESTING, UPGRADES, PERFORMANCE

- **Tests:** `dev/tests/unit`, `integration`, `api-functional`, `bin/magento dev:tests:run static`, Jasmine, MFTF 3.6.0 (`Test/Mftf`).
- **Upgrade:** back up, `maintenance:enable`, Composer update, `setup:upgrade`, `setup:di:compile`, `indexer:reindex`, `maintenance:disable`. Search: `config:set catalog/search/engine opensearch`; cache: `setup:config:set --cache-backend=valkey`.
- **Speed:** production mode, Valkey (cache, page cache, sessions), Varnish 8 or Fastly, indexers on schedule, real cron.

bin/magento: DAILY COMMANDS

Info	`--version`, `setup:db:status`, `deploy:mode:show`
Modes	`deploy:mode:set developer|production`
Modules	`module:enable|disable Vendor_Module`, then `setup:upgrade`
Build	`setup:upgrade`, `setup:di:compile`, `setup:static-content:deploy`
Cache	`cache:status`, `cache:clean` (by tag), `cache:flush` (empties storage), `cache:enable|disable [type]`
Indexers	`indexer:status`, `indexer:reindex [id]`, `indexer:set-mode schedule|realtime`
Cron	`cron:install [--force]`, `cron:remove`, `cron:run --group=<id>`
Queues	`queue:consumers:start <name> --max-messages=1`
Config	`config:set`, `config:show`, `setup:config:set`
Other	`maintenance:enable|disable`, `encryption:key:change`

Custom command: extend Symfony `Command`, add it to the `commands` argument of `Magento\Framework\Console\CommandListInterface` in `di.xml`.

MODULE SKELETON

```
app/code/Vendor/Module/
  registration.php  composer.json
  etc/module.xml di.xml config.xml
      events.xml acl.xml webapi.xml
      crontab.xml db_schema.xml
      db_schema_whitelist.json
      schema.graphqls csp_whitelist.xml
      queue*.xml  frontend/routes.xml
      adminhtml/{routes,menu,system}.xml
  Setup/Patch/{Data,Schema}/
  Controller/ Block/ Model/ Api/ Plugin/
  Observer/ ViewModel/
  view/{frontend,adminhtml}/
    {layout,templates,web}
<module name="Vendor_Module"><sequence>
  <module name="Magento_Catalog"/>
</sequence></module>
```

Without `registration.php` a Composer module is never registered. `setup_version` is legacy; use patches.

di.xml, PLUGINS, EVENTS

```
<preference for="Vendor\Api\FooInterface"
  type="Vendor\Model\Foo"/>
<type name="Magento\Quote\Model\Quote">
  <plugin name="vendor_check"
    type="Vendor\Plugin\Check" sortOrder="10"/>
</type>
<type name="Vendor\Model\Counter"
  shared="false"/>
<virtualType name="VendorLog"
  type="Monolog\Logger"/>
<event name="checkout_submit_all_after">
  <observer name="vendor_placed"
    instance="Vendor\Observer\Placed"/>
</event>
```

Plugins wrap public, non-final, non-static methods of ObjectManager-built objects: `beforeX` (arguments), `afterX` (result), `aroundX` (to stop the chain; without `$proceed` it hides later plugins). Lower `sortOrder` runs first. Observers implement `ObserverInterface::execute()`; a repeated observer name merges. `shared="false"` acts when injected.

SCHEMA AND PATCHES

`etc/db_schema.xml` is merged across modules and applied by `setup:upgrade`. **Adding and changing need no whitelist; removing does.** Commands: `setup:db-declaration:generate-whitelist --module-name=Vendor_Module`, `setup:db-declaration:generate-patch`.

- Generate the whitelist while the element is still in the XML; commit it each release; never with a DB prefix.
- Rename: `onCreate="migrateDataFrom(old)"`.
- Disabling a module drops its whitelisted tables on the next `setup:upgrade`.

Patches implement `DataPatchInterface` or `SchemaPatchInterface` (`Magento\Framework\Setup\Patch`): `apply()`, static `getDependencies()`, `getAliases()`; they run once and are recorded in `patch_list`.

FRONTEND: LUMA THEME, LAYOUT, TEMPLATES

```
app/design/frontend/Vendor/name/
  registration.php theme.xml
  etc/view.xml requirejs-config.js
  web/css/source/_extend.less _theme.less
  Magento_Theme/layout/default.xml
  Magento_Theme/templates/html/
    copyright.phtml
<theme><title>My Theme</title>
  <parent>Magento/luma</parent></theme>
```

- Templates and static files use **fallback** (theme, parents, module). Layout files merge by handle; `layout/override/base/` replaces a module file.
- Handle = file name (`catalog_product_view.xml`); `referenceContainer` adds to a container. Entry points `styles-m.less`, `styles-l.less`.
- Production mode needs `setup:static-content:deploy`. Escape output in `.phtml`.

Alternatives: Hyvä (Tailwind CSS + Alpine.js, OSL-3.0/AFL-3.0), PWA Studio 14.5.1 (React, GraphQL), Commerce Storefront on Edge Delivery (Cloud Service).

REST, GRAPHQL, TOKENS

REST	`/rest/V1/...`, `/rest/<store-view>/V1/...`, `/rest/all/V1/...`
Async, bulk	`/rest/<store>/async/V1/...`, `/rest/<store>/async/bulk/V1/...`: HTTP 202 + `bulk_uuid`; `GET /V1/bulk/{uuid}/status`
GraphQL	`POST /graphql`, optional `Store` header; `schema.graphqls` + resolver

`webapi.xml` resource: `anonymous`, `self` or an ACL id. Paging and filters: `searchCriteria[filter_groups][0][filters][0][field\|value\|condition_type]`, `[sortOrders]`, `[pageSize]`, `[currentPage]`.

Customer	`POST /V1/integration/customer/token` (`username`, `password`) or GraphQL `generateCustomerToken`; revoke: `/V1/integration/customer/revoke-customer-token`
Admin	`POST /V1/integration/admin/token`; with 2FA: `/V1/tfa/provider/<provider>/authenticate`
Integration	OAuth 1.0a HMAC-SHA256 signed requests
Cloud Service	Adobe IMS OAuth 2

Header `Authorization: Bearer <token>`. JWT `HS256`, 60 min (`webapi/jwtauth/*_expiration`). GraphQL limits: 10 aliases, 1,048,576 query characters, complexity 300, depth 20.

OUT-OF-PROCESS OPTIONS

App Builder	Node.js actions (Adobe I/O Runtime), `aio` CLI; any edition
I/O Events	Commerce to app via cron; Adobe Commerce 2.4.4+, not Open Source
Webhooks	Synchronous HTTPS; reply continues, stops or changes data
Admin UI SDK	Menus, pages, columns; 2.4.5+, not Open Source
App Management	`app.commerce.config`; hosted Commerce only
API Mesh	Merges GraphQL, REST, JSON into one endpoint

CLI: `events:subscribe <code> --fields=<name>`, `events:list:all`, `events:info <code>`.
Event code: `observer.<event>` or `plugin.<class.path>.<method>`.

SECURITY CHECKLIST

- **Patchfirst.** APSB26-146 (2026-09-07): CVE-2026-75650, score 10.0, unauthenticated code execution, hotfix `VULN-39341` on every supported line, then rotate keys.
- Non-default Admin URI (`--backend-frontname`, random by default): less noise, not a control.
- Keep `Magento_TwoFactorAuth` on (Google Authenticator, Duo, Authy, U2F); retry limit 10, lockout 300 s since 2.4.8.
- CSP: payment pages restrict mode since 2.4.7, others report-only; whitelist hosts per policy in `etc/csp_whitelist.xml`.
- Code: escape output, form keys, ACL on every admin route, encrypted config for secrets.
- Scan: Security Scan Tool, `composer audit`, `vendor/bin/phpcs --standard=Magento2`.
- Incident: keep logs, patch, rebuild clean, `encryption:key:change` and `encryption:data:re-encrypt core_config_data sales_order_payment`; rotate admin, integration, OAuth, gateway and DB secrets.

CLOUD DEPLOYMENT: MANY PODS ON AMAZON EKS (CHECKED 2026-10-29)

Database. RDS for MySQL 8.4 or MariaDB 12.3, or Aurora MySQL 3 (listed by Adobe for 2.4.9); `env.php` reads the environment with TLS; `mg_app` and a migration user; split database is deprecated since 2.4.2.

State placement. Five services: RDS, OpenSearch 3.x, two Valkey groups (cache and sessions), Amazon MQ for RabbitMQ 4.3, S3 for media (remote storage driver, `remote-storage:sync`); Varnish StatefulSet for the full page cache; `config.php` committed, secrets in the environment; code, DI and static content built into the image.

Cron. One CronJob running `cron:run --group=<id> --bootstrap=standaloneProcessStarted=1` per group; consumers as Deployments running `queue:consumers:start` scaled by KEDA, `cron_consumers_runner` off; indexers on Update by Schedule.

Release steps. Gate with `setup:db:status`; pre-upgrade Job `setup:upgrade --keep-generated` (or `app:config:import`); pause consumers and cron; rolling update behind Varnish; static content and DI are built into the image, so nothing is compiled at run time.

Key commands. `bin/magento setup:db:status`, `indexer:status`, `queue:consumers:list`, `cache:status`, `config:show web/secure/base_url`, `helm upgrade pocketcoffee ./pocketcoffee-magento`.