Apache virtual host

A complete Apache 2.4 virtual host for Exponential 6 with an HTTP to HTTPS redirect, HTTP/2, PHP-FPM through mod_proxy_fcgi, the rules of .htaccess_root in server context, security headers and browser caching.

Source file: `doc/eng-us/examples/web-servers/apache-virtual-host.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/examples/web-servers/apache-virtual-host.md

`doc/eng-us/examples/ezpublish.conf` is a complete virtual host for one installation: port 80 redirects to HTTPS (and serves certificate challenge files), port 443 serves the site over HTTP/2 with PHP-FPM, and the routing rules of `.htaccess_root` sit in the virtual host itself. Apache then reads no `.htaccess` at all (`AllowOverride None`), which saves a file lookup per directory on every request. The file keeps its historical name.

> **Warning**
>
> `.htaccess` **slows down every request.** When `.htaccess` files are enabled (`AllowOverride` anything but `None`), let alone used, Apache looks for them and reads and evaluates them on every request, in every directory of the path, and page loading times go up drastically (measured). The only real fix is to switch the feature off for the virtual host: `AllowOverride None` in the HTTPS virtual host, and the rules in the virtual host's own configuration as mod_rewrite rules. That is far faster in production speed tests and well worth the time it takes to optimize a production configuration. Switch over right away: copy the rules of the example HTTPS virtual host into your virtual host (with the static cache, also `staticcache_offload.conf`), set `AllowOverride None`, run `apachectl configtest`, reload Apache and test the site. The same rules for other servers: nginx, LiteSpeed, Caddy, IIS, Velocity (built in); where virtual host rules go in a control panel: Plesk, cPanel, DirectAdmin, ISPConfig. **On LiteSpeed or OpenLiteSpeed,** the server's own page cache can also answer anonymous visitors and crawlers without PHP: Page caching with LiteSpeed and OpenLiteSpeed.

Install it

1. Load the modules: `rewrite`, `proxy`, `proxy_fcgi`, `ssl`, `headers`, `expires`, and `http2` for HTTP/2 (Debian: `a2enmod rewrite proxy_fcgi ssl http2 headers expires`; Red Hat style systems load most of them by default and ship `mod_http2` and `mod_ssl` as packages of their own).
2. Copy the file to `/etc/apache2/sites-available/example.com.conf` (Debian, then `a2ensite example.com`) or `/etc/httpd/conf.d/example.com.conf` (Red Hat style), and replace `example.com`, `/var/www/example.com`, the PHP-FPM socket and the certificate paths.
3. Create the PHP-FPM pool the socket belongs to (PHP-FPM pool).
4. Get the certificate. With certbot, before the HTTPS host exists, `certbot certonly --webroot -w /var/www/example.com -d example.com -d www.example.com`: the port 80 host serves `/.well-known/acme-challenge/` for exactly that.
5. Check and reload:

Shell

```
apachectl configtest           # Syntax OK
systemctl reload apache2       # httpd on Red Hat style systems
curl -sI https://example.com/ | head -1
```

Using a hosting panel such as Plesk? It writes the virtual host itself: do not edit its files; keep `.htaccess` instead (Apache .htaccess) and see chapter 8, section 8.5.3. There, `./console exp:setup:htaccess --apply` puts `.htaccess_root` into place as `.htaccess` (Setting up .htaccess and config.php from the console).

With this virtual host (`AllowOverride None`) a `.htaccess` in the root is not read, but copying it does no harm and keeps the rules in force should the tree ever be served with `AllowOverride All`. After an upgrade, `./console exp:setup:htaccess` shows the diff between your `.htaccess` and the new `.htaccess_root`, which is also the list of rules to carry over into this virtual host.

The file

The first versions of this file came with eZ Publish, the software Exponential grew out of (1999 - 2014). Its rewrite rules are those of `.htaccess_root`, part of the software (GNU GPL version 2 or later); the documented example around them is free documentation (GFDL 1.3 or later), like this page.

Config

```
# Apache virtual host for Exponential 6 (PHP-FPM, HTTPS, HTTP/2)
# ------------------------------------------------------------------------
# What this is: a complete virtual host for one Exponential installation,
# with the routing rules of .htaccess_root written in server context, so
# Apache needs no .htaccess (AllowOverride None saves a file lookup per
# directory on every request). The rules are the ones of .htaccess_root of
# this release: compare them with .htaccess_root after every upgrade and
# merge new rules. Explained in doc/eng-us/examples/web-servers/apache-virtual-host.md.
#
# Replace before use:
#   example.com, www.example.com      the names of the site
#   /var/www/example.com              the installation root (where index.php is)
#   /run/php-fpm/www.sock             the socket of this site's PHP-FPM pool
#   /etc/ssl/example.com/*.pem        the certificate and its key
#
# Needs the modules: rewrite, proxy, proxy_fcgi, ssl, http2, headers,
# expires (Debian: a2enmod rewrite proxy_fcgi ssl http2 headers expires).
# Check with "apachectl configtest", reload with "systemctl reload apache2"
# (httpd on Red Hat style systems).
#
# In server context (outside <Directory>) a RewriteRule pattern is matched
# against the URL path with its leading slash, and %{REQUEST_FILENAME} is not
# yet a file name: that is why every pattern below starts with "^/" and why
# the script rule tests %{DOCUMENT_ROOT}%{REQUEST_URI}.
#
# License: the first versions of this file came with the software Exponential
# grew out of, eZ Publish, 1999 - 2014. The rewrite rules are those of
# .htaccess_root, part of the software (GNU General Public License version 2
# or later, like the rest of the code). This documented example, its comments
# and optional sections, is free documentation under the GNU Free
# Documentation License 1.3 or later (doc/eng-us/README-LICENSE.md).

# Plain HTTP: only the redirect to HTTPS, and the ACME challenge files a
# certificate client (certbot --webroot) places below .well-known.
```

```
<VirtualHost *:80>
    ServerName example.com
    ServerAlias www.example.com
    DocumentRoot /var/www/example.com

    <Directory /var/www/example.com>
        Options None
        AllowOverride None
        Require all granted
    </Directory>

    RewriteEngine On
    RewriteCond %{REQUEST_URI} !^/\.well-known/acme-challenge/[A-Za-z0-9_-]+$
    RewriteRule ^ https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]

    ErrorLog logs/example.com-error.log
    CustomLog logs/example.com-access.log combined
</VirtualHost>

<VirtualHost *:443>
    ServerName example.com
    ServerAlias www.example.com

    # The installation root, where index.php is
    DocumentRoot /var/www/example.com
    DirectoryIndex index.php

    Protocols h2 http/1.1
    SSLEngine on
    SSLCertificateFile    /etc/ssl/example.com/fullchain.pem
    SSLCertificateKeyFile /etc/ssl/example.com/privkey.pem

    ErrorLog logs/example.com-ssl-error.log
    CustomLog logs/example.com-ssl-access.log combined

    # Uploads: Apache's own limit (0 = none); PHP's upload_max_filesize and
    # post_max_size in the pool decide (doc/eng-us/examples/web-servers/php-fpm-pool.conf)
    LimitRequestBody 0

    <Directory /var/www/example.com>
        Options FollowSymLinks
        # No .htaccess is read: the rules are below
        AllowOverride None
        Require all granted
    </Directory>

    # Every PHP script goes to the PHP-FPM pool of this site. Only the front
    # controllers ever get here: the rules below send nothing else to a .php.
    <FilesMatch "\.php$">
        SetHandler "proxy:unix:/run/php-fpm/www.sock|fcgi://localhost"
    </FilesMatch>

    # The rules of .htaccess_root, in its order, in server context
    <IfModule mod_rewrite.c>
        RewriteEngine On
        # No dot file or dot directory (.git, .env, .htaccess ...) is served or
        # run, whatever folder it lies in; .well-known stays reachable.
        RewriteRule (^|/)\.(?!well-known(/|$)) - [R=404,L]
        # No script runs because its path was asked for, except the front
        # controllers at the root: a .php below them (in the package store,
        # an extension, var/) answers 404 instead of being executed.
        RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -f
        RewriteRule ^/.+/[^/]+\.(php[0-9]?|phtml|phar)$ - [R=404,L]
        # REST API
```

```
        RewriteRule ^/api/ /index_rest.php [L]
        RewriteRule ^/index_rest\.php - [L]
        # SOAP server (soap.ini EnableSOAP=true): index_soap.php since
        # Exponential 6.0.15, soap.php only forwards to it. Uncomment with SOAP on:
        #RewriteRule ^/index_soap\.php - [L]
        RewriteRule ^/([^/]+/)?content/treemenu.* /index_treemenu.php [L]
        RewriteRule ^/var/([^/]+/)?storage/images(-versioned)?/.* - [L]
        # The stored files of PDF exports generated once (Setup > PDF export)
        RewriteRule ^/var/([^/]+/)?storage/pdf/[A-Za-z0-9][A-Za-z0-9._-]*\.pdf$ - [L]
        # Of the originals only the images; other uploads stay behind
        # content/download and its permissions.
        RewriteRule ^/var/([^/]+/)?storage/original/image/.+\.(png|jpe?g|gif|webp|svg)$ - [L]
        RewriteRule ^/var/([^/]+/)?cache/(texttoimage|public)/.* - [L]
        # With site.ini [FileSettings] CacheVarDir=var_cache (multi-site hosting):
        #RewriteRule ^/var_cache/([^/]+/)?cache/(texttoimage|public)/.* - [L]
        RewriteRule ^/design/[^/]+/(stylesheets|images|javascript|fonts)/.* - [L]
        RewriteRule ^/share/icons/.* - [L]
        # Icon themes of extensions (icon.ini [ExtensionSettings] IconExtensions[]): images with
exactly one extension
        RewriteRule ^/extension/[^/]+/icons/[A-Za-z0-9_/-]+\.(svg|png|gif)$ - [L]
        RewriteRule ^/extension/[^/]+/design/[^/]+/(stylesheets|flash|images|lib|javascripts?|fon
ts|vendor|media)/.* - [L]
        RewriteRule ^/packages/styles/.+/(stylesheets|images|javascript)/[^/]+/.* - [L]
        RewriteRule ^/packages/styles/.+/thumbnail/.* - [L]
        # Only the preview images of imported packages
        RewriteRule ^/var/storage/packages/.+\.(png|jpe?g|gif|webp)$ - [L]
        RewriteRule ^/favicon\.ico - [L]
        RewriteRule ^/design/standard/images/favicon\.ico - [L]
        RewriteRule ^/robots\.txt - [L]
        # The site's service workers, served from the root
        RewriteRule ^/index\.js$ - [L]
        RewriteRule ^/sw\.js$ - [L]
        RewriteRule ^/w3c/p3p\.xml - [L]
        # Everything else, settings/, var/, vendor/, kernel/ and composer.json
        # included, is answered by index.php, never sent as a file.
        RewriteRule .* /index.php [L]
    </IfModule>

    <IfModule mod_headers.c>
        # SVG images are sanitized on upload; served from the storage they also
        # run nothing and load nothing when opened directly.
        <FilesMatch "(?i)\.svgz?$">
            Header always set Content-Security-Policy "sandbox; default-src 'none'; style-src 'un
safe-inline'"
            Header always set X-Content-Type-Options "nosniff"
        </FilesMatch>
        # The kernel sends these on its pages (site.ini [HTTPHeaderSettings]
        # SecurityHeaders[]); the same values for the files Apache serves.
        # "setifempty" keeps a header that index.php already sent.
        Header setifempty X-Content-Type-Options "nosniff"
        Header setifempty X-Frame-Options "SAMEORIGIN"
        Header setifempty Referrer-Policy "strict-origin-when-cross-origin"
        Header setifempty Permissions-Policy "camera=(), microphone=(), payment=(), usb=()"
        # HSTS once every name is served over HTTPS for good; start short
        # (doc/eng-us/install/13-security-hardening.md, 13.9). Here or in site.ini
        # SecurityHeaders[Strict-Transport-Security], not both.
        #Header always set Strict-Transport-Security "max-age=300"
    </IfModule>

    # Optional: browser caching of images and design files
    <IfModule mod_expires.c>
        <LocationMatch "^/var/([^/]+/)?storage/images(-versioned)?/">
            # The image URL changes when the image changes
            ExpiresActive On
```

```
            ExpiresDefault "access plus 1 year"
        </LocationMatch>
        <LocationMatch "^/(extension/[^/]+/)?design/[^/]+/(stylesheets|images|javascripts?|fonts|
lib)/">
            # Design files keep their names: keep this short while a design changes
            ExpiresActive On
            ExpiresDefault "access plus 7 days"
        </LocationMatch>
        <LocationMatch "^/(share|extension/[^/]+)/icons/">
            ExpiresActive On
            ExpiresDefault "access plus 7 days"
        </LocationMatch>
        # Packed scripts and styles carry their modification time in the name
        # only with ezjscore.ini [Packer] AppendLastModifiedTime=enabled:
        #<LocationMatch "^/var/([^/]+/)?cache/public/">
        #    ExpiresActive On
        #    ExpiresDefault "access plus 30 days"
        #</LocationMatch>
    </IfModule>

    # A generated static cache (Setup > Cache > Static content cache) can be
    # served without starting PHP: add the rules of
    # doc/eng-us/examples/staticcache_offload.conf, translated to server context,
    # BEFORE the RewriteRule .* /index.php line.
</VirtualHost>
```

What differs from .htaccess

The rules are the ones of `.htaccess_root`, in its order, with two changes that server context needs:

- **Every pattern starts with `^/`.** Outside `<Directory>`, `RewriteRule` sees the URL path with its leading slash.
- **The script rule tests `%{DOCUMENT_ROOT}%{REQUEST_URI}`.** In server context `%{REQUEST_FILENAME}` is not yet a file name; the rule must still fire only for a script that exists, so a URL alias that ends in `.php` keeps working.

And one consequence: `[L]` ends the processing here, so `/content/treemenu/...` reaches `index_treemenu.php` directly (under `.htaccess` it ends at `index.php`, see Apache .htaccess).

Everything not on the asset list, `settings/`, `var/log/`, `vendor/`, `kernel/`, `composer.json` included, is rewritten to `/index.php` and never sent. The `<FilesMatch "\.php$">` handler hands `/index.php`, `/index_rest.php` and `/index_treemenu.php` to PHP-FPM; no other script is ever reached, because the rules send nothing else to a `.php`. With SOAP on, uncomment the `index_soap.php` line: the SOAP server is `index_soap.php` since Exponential 6.0.15 (`soap.php` before, now a shim that forwards to it). What to change and how to test it: the SOAP entry script is now index_soap.php.

Headers and caching

- `X-Content-Type-Options`, `X-Frame-Options`, `Referrer-Policy` and `Permissions-Policy` with the values the kernel sends on its pages (`site.ini [HTTPHeaderSettings] SecurityHeaders[]`), set with `setifempty` so a page keeps the kernel's own. HSTS is commented: switch it on once every name is served over HTTPS, here or in `site.ini`, not both (chapter 13, section 13.9).

- Stored image variations are cached for a year (their URL changes when the image changes), design files and icons for a week.

Behind a proxy or load balancer

When TLS ends at a proxy, keep only the port 80 host (without its redirect, with the `<Directory>`, handler and rules of the 443 host), let the proxy send `X-Forwarded-Proto` and the original `Host`, and list the proxy's address in `site.ini` `[HTTPHeaderSettings] TrustedProxies[]`. See chapter 8, section 8.8.

WebDAV

This virtual host does not route WebDAV (off by default). To serve it, add `RewriteRule ^index_webdav\.php - [L]` inside the `<Directory>` block, before the catch-all, in the **HTTPS** host (the port 80 host only redirects), or a virtual host of its own (`webdav.example.com`) with `RewriteRule . /index_webdav.php [L]`. Since 6.0.15 the script is `index_webdav.php`; rules written for `webdav.php` keep working through a forwarding shim until a later major version. Before/after and the test (`curl -X PROPFIND -H 'Depth: 1'`, expect `207 Multi-Status`): The WebDAV entry script is now index_webdav.php.

Check it

Shell

```
curl -sI http://example.com/some/page | grep -i -E '^(HTTP|location)'     # 301 to https://
for p in /.git/config /.env /kernel/classes/ezini.php /vendor/autoload.php; do
  printf '%-30s %s\n' "$p" "$(curl -s -o /dev/null -w '%{http_code}' https://example.com$p)"   # 404 each
done
curl -s https://example.com/settings/site.ini | grep -c '^\[DatabaseSettings\]'   # 0: a page, not the file
curl -sI https://example.com/design/admin4/stylesheets/components.css | grep -i -E '^(HTTP|x-content-type)'
```

How this example was checked: `httpd -t`, then a throwaway Apache 2.4 with this file (ports, paths, socket and a self-signed certificate substituted) answered every request of the test list over HTTPS as expected, redirected plain HTTP and served an ACME challenge file on port 80.

Related pages

- Apache .htaccess: the rules explained one by one.
- PHP-FPM pool: the pool behind the socket.
- Chapter 8, section 8.5: Apache with PHP-FPM.
- Chapter 13: Security hardening.
- `../staticcache_offload.conf`: serving the static content cache without PHP.
- Web server examples and all examples.
- Free documentation: this page is free to copy, translate and improve.

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.