Apache HTTP Server Web server

START HERE

`apachectl configtest`
Check the config

`httpd -S`
Show vhosts

`apachectl start`
Start the server

`httpd -M`
List loaded modules

`apachectl graceful`
Reload, keep connections

`a2ensite <site>`
Enable a vhost (Debian)

`apachectl stop`
Stop the server

`httpd -V`
Version and paths

Request host and port → match → **VirtualHost** ServerName / Alias → map → **Directory / Location** Require, Options → merge → **.htaccess** per-directory overrides

CONCEPTS

Always run configtest before graceful: a syntax error then leaves the running server untouched.

Debian and Ubuntu call the binary and service apache2 (apache2ctl); RHEL, Fedora and source builds use httpd.

Directive order in config does not matter inside a section, but sections merge: Directory, then .htaccess, then Location.

CONTROL

`apachectl start`
Start the server

`apachectl stop`
Stop, aborting open connections

`apachectl restart`
Stop and start again

`apachectl graceful`
Reload config without dropping requests

`apachectl graceful-stop`
Finish requests, then stop

`apachectl configtest`
Check config syntax
Prints Syntax OK or the file and line of the error; same as httpd -t.

`apachectl -k <start|stop|graceful>`
Pass-through form of the commands

`apachectl status`
Brief status report
Needs mod_status; fullstatus also needs a text browser such as lynx.

`systemctl reload apache2`
Reload via systemd (Debian)

`systemctl reload httpd`
Reload via systemd (RHEL)

`apache2ctl -t`
Config test on Debian

INSPECT

`httpd -S`
Show parsed vhosts and ports

`httpd -M`
List loaded modules

`httpd -V`
Version, MPM, config path

`httpd -v`
Print version only

INSPECT (cont.)

`httpd -l`
List compiled-in modules

`httpd -L`
List directives of static modules

`httpd -t -D DUMP_INCLUDES`
Show all included config files

`httpd -f <file> -t`
Test a specific config file

`httpd -X`
Run in foreground, one worker

`httpd -e debug -E <file>`
Verbose startup errors to a file

`httpd -D <name>`
Define a name for <IfDefine>

DEBIAN HELPERS

`a2enmod <module>`
Enable a module

`a2dismod <module>`
Disable a module

`a2ensite <site>`
Enable a vhost file

`a2dissite <site>`
Disable a vhost file

`a2enconf <conf>`
Enable a config snippet

`a2query -s`
List enabled sites

`a2query -m`
List enabled modules

Run a2* then reload: the tools only manage symlinks in sites-enabled, mods-enabled and conf-enabled.

FILES AND PATHS

`/etc/apache2/apache2.conf`
Main config (Debian)

`/etc/httpd/conf/httpd.conf`
Main config (RHEL)

`/etc/apache2/sites-available/`
Vhost files (Debian)

`/etc/httpd/conf.d/*.conf`
Drop-in vhosts (RHEL)

`/var/log/apache2/error.log`
Error log (Debian)

`/var/log/httpd/error_log`
Error log (RHEL)

`/var/www/html`
Default document root

VIRTUAL HOSTS

`<VirtualHost *:80>`
Open a vhost for a port

`ServerName <host>`
Main hostname of the vhost

`ServerAlias <host> ...`
Additional hostnames

`DocumentRoot <dir>`
Directory served for the vhost

`ErrorLog <file>`
Per-vhost error log

`CustomLog <file> combined`
Per-vhost access log

`Redirect permanent / <url>`
Redirect everything

`Alias /<path> <dir>`
Map a URL path to a directory

`Listen <port>`
Bind a port

ACCESS AND .HTACCESS

`<Directory <dir>>`
Settings for a filesystem path

`Require all granted`
Allow everyone

`Require all denied`
Deny everyone

`Require ip <cidr>`
Allow an address range

`Options -Indexes`
No directory listings

`AllowOverride All`
Honor .htaccess in this directory

`DirectoryIndex index.html`
Default file for a directory

`Header set <name> "<value>"`
Set a response header
Needs mod_headers.

`RewriteRule <regex> <target> [L,R=301]`
Rewrite or redirect URLs
Needs mod_rewrite; RewriteEngine On first.

`RewriteCond %{HTTPS} off`
Condition for the next RewriteRule

`htpasswd -c <file> <user>`
Create password file
Drop -c to add users to an existing file.

In 2.4 use Require, not Order/Allow/Deny; mixing them fails or behaves oddly.

PROXY AND TLS

`a2enmod proxy proxy_http`
Enable reverse proxy modules

`ProxyPass / http://<host>:<port>/`
Forward requests to a backend

`ProxyPassReverse / http://<host>:<port>/`
Rewrite backend redirects

`ProxyPreserveHost On`
Pass the original Host header

`a2enmod ssl`
Enable TLS support

`SSLEngine on`
Turn TLS on in a vhost

`SSLCertificateFile <path>`
Certificate (full chain)

`SSLCertificateKeyFile <path>`
Private key

LOGS AND TUNING

`LogLevel warn`
Set log verbosity

`LogLevel ssl:debug`
Raise one module only

`tail -f /var/log/apache2/error.log`
Watch errors live

`httpd -V | grep -i mpm`
Which MPM is used

`MaxRequestWorkers <n>`
Cap concurrent requests

RECIPES

Safe reload

```
# test, then reload without dropping requests
apachectl configtest &&
apachectl graceful
```

New site on Debian

```
sudo cp <site>.conf /etc/apache2/sites-available/
sudo a2ensite <site>
sudo apache2ctl configtest
sudo systemctl reload apache2
```

Reverse proxy vhost

```
<VirtualHost *:80>
  ServerName <host>
  ProxyPreserveHost On
  ProxyPass /
http://127.0.0.1:<port>/
  ProxyPassReverse /
http://127.0.0.1:<port>/
</VirtualHost>
```

Which vhost answers?

```
httpd -S
curl -sI -H "Host: <host>"
http://127.0.0.1/
```