

```
1: #
2: # This is the main Apache HTTP server configuration file. It contains the
3: # configuration directives that give the server its instructions.
4: # See <URL:http://httpd.apache.org/docs/2.4/> for detailed information.
5: # In particular, see
6: # <URL:http://httpd.apache.org/docs/2.4/mod/directives.html>
7: # for a discussion of each configuration directive.
8: #
9: # Do NOT simply read the instructions in here without understanding
10: # what they do. They're here only as hints or reminders. If you are unsure
11: # consult the online docs. You have been warned.
12: #
13: # Configuration and logfile names: If the filenames you specify for many
14: # of the server's control files begin with "/" (or "drive:" for Win32), the
15: # server will use that explicit path. If the filenames do *not* begin
16: # with "/", the value of ServerRoot is prepended -- so 'log/access_log'
17: # with ServerRoot set to '/www' will be interpreted by the
18: # server as '/www/log/access_log', where as '/log/access_log' will be
19: # interpreted as '/log/access_log'.
20: #
21: #
22: # ServerRoot: The top of the directory tree under which the server's
23: # configuration, error, and log files are kept.
24: #
25: # Do not add a slash at the end of the directory path. If you point
26: # ServerRoot at a non-local disk, be sure to specify a local disk on the
27: # Mutex directive, if file-based mutexes are used. If you wish to share the
28: # same ServerRoot for multiple httpd daemons, you will need to change at
29: # least PidFile.
30: #
31: ServerRoot "/etc/httpd"
32: #
33: #
34: # Listen: Allows you to bind Apache to specific IP addresses and/or
35: # ports, instead of the default. See also the <VirtualHost>
36: # directive.
37: #
38: # Change this to Listen on specific IP addresses as shown below to
39: # prevent Apache from glomming onto all bound IP addresses.
40: #
41: #Listen 12.34.56.78:80
42: Listen 80
43: #
44: #
45: # Dynamic Shared Object (DSO) Support
46: #
47: # To be able to use the functionality of a module which was built as a DSO you
48: # have to place corresponding 'LoadModule' lines at this location so the
49: # directives contained in it are actually available before they are used.
50: # Statically compiled modules (those listed by `httpd -l`) do not need
51: # to be loaded here.
52: #
53: # Example:
54: # LoadModule foo_module modules/mod_foo.so
55: #
56: Include conf.modules.d/*.conf
57: #
58: #
59: # If you wish httpd to run as a different user or group, you must run
60: # httpd as root initially and it will switch.
61: #
62: # User/Group: The name (or #number) of the user/group to run httpd as.
63: # It is usually good practice to create a dedicated user and group for
64: # running httpd, as with most system services.
65: #
66: User apache
```

```
67: Group apache
68:
69: # 'Main' server configuration
70: #
71: # The directives in this section set up the values used by the 'main'
72: # server, which responds to any requests that aren't handled by a
73: # <VirtualHost> definition. These values also provide defaults for
74: # any <VirtualHost> containers you may define later in the file.
75: #
76: # All of these directives may appear inside <VirtualHost> containers,
77: # in which case these default settings will be overridden for the
78: # virtual host being defined.
79: #
80:
81: #
82: # ServerAdmin: Your address, where problems with the server should be
83: # e-mailed. This address appears on some server-generated pages, such
84: # as error documents. e.g. admin@your-domain.com
85: #
86: ServerAdmin root@localhost
87:
88: #
89: # ServerName gives the name and port that the server uses to identify itself.
90: # This can often be determined automatically, but we recommend you specify
91: # it explicitly to prevent problems during startup.
92: #
93: # If your host doesn't have a registered DNS name, enter its IP address here.
94: #
95: #ServerName www.example.com:80
96:
97: #
98: # Deny access to the entirety of your server's filesystem. You must
99: # explicitly permit access to web content directories in other
100: # <Directory> blocks below.
101: #
102: <Directory />
103:     AllowOverride none
104:     Require all denied
105: </Directory>
106:
107: #
108: # Note that from this point forward you must specifically allow
109: # particular features to be enabled - so if something's not working as
110: # you might expect, make sure that you have specifically enabled it
111: # below.
112: #
113:
114: #
115: # DocumentRoot: The directory out of which you will serve your
116: # documents. By default, all requests are taken from this directory, but
117: # symbolic links and aliases may be used to point to other locations.
118: #
119: DocumentRoot "/var/www/html"
120:
121: #
122: # Relax access to content within /var/www.
123: #
124: <Directory "/var/www">
125:     AllowOverride None
126:     # Allow open access:
127:     Require all granted
128: </Directory>
129:
130: # Further relax access to the default document root:
131: <Directory "/var/www/html">
132:     #
```

```
133: # Possible values for the Options directive are "None", "All",
134: # or any combination of:
135: #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
136: #
137: # Note that "MultiViews" must be named *explicitly* --- "Options All"
138: # doesn't give it to you.
139: #
140: # The Options directive is both complicated and important. Please see
141: # http://httpd.apache.org/docs/2.4/mod/core.html#options
142: # for more information.
143: #
144: Options Indexes FollowSymLinks
145:
146: #
147: # AllowOverride controls what directives may be placed in .htaccess files.
148: # It can be "All", "None", or any combination of the keywords:
149: #   Options FileInfo AuthConfig Limit
150: #
151: AllowOverride None
152:
153: #
154: # Controls who can get stuff from this server.
155: #
156: Require all granted
157: </Directory>
158:
159: #
160: # DirectoryIndex: sets the file that Apache will serve if a directory
161: # is requested.
162: #
163: <IfModule dir_module>
164:     DirectoryIndex index.html
165: </IfModule>
166:
167: #
168: # The following lines prevent .htaccess and .htpasswd files from being
169: # viewed by Web clients.
170: #
171: <Files ".ht*">
172:     Require all denied
173: </Files>
174:
175: #
176: # ErrorLog: The location of the error log file.
177: # If you do not specify an ErrorLog directive within a <VirtualHost>
178: # container, error messages relating to that virtual host will be
179: # logged here. If you *do* define an error logfile for a <VirtualHost>
180: # container, that host's errors will be logged there and not here.
181: #
182: ErrorLog "logs/error_log"
183:
184: #
185: # LogLevel: Control the number of messages logged to the error_log.
186: # Possible values include: debug, info, notice, warn, error, crit,
187: # alert, emerg.
188: #
189: LogLevel warn
190:
191: <IfModule log_config_module>
192:     #
193:     # The following directives define some format nicknames for use with
194:     # a CustomLog directive (see below).
195:     #
196:     LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
197:     LogFormat "%h %l %u %t \"%r\" %>s %b" common
198:
```

```
199: <IfModule logio_module>
200:     # You need to enable mod_logio.c to use %I and %O
201:     LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %I %O"
combinedio
202: </IfModule>
203:
204: #
205: # The location and format of the access logfile (Common Logfile Format).
206: # If you do not define any access logfiles within a <VirtualHost>
207: # container, they will be logged here. Contrariwise, if you *do*
208: # define per-<VirtualHost> access logfiles, transactions will be
209: # logged therein and *not* in this file.
210: #
211: #CustomLog "logs/access_log" common
212:
213: #
214: # If you prefer a logfile with access, agent, and referer information
215: # (Combined Logfile Format) you can use the following directive.
216: #
217: CustomLog "logs/access_log" combined
218: </IfModule>
219:
220: <IfModule alias_module>
221:     #
222:     # Redirect: Allows you to tell clients about documents that used to
223:     # exist in your server's namespace, but do not anymore. The client
224:     # will make a new request for the document at its new location.
225:     # Example:
226:     # Redirect permanent /foo http://www.example.com/bar
227:
228:     #
229:     # Alias: Maps web paths into filesystem paths and is used to
230:     # access content that does not live under the DocumentRoot.
231:     # Example:
232:     # Alias /webpath /full/filesystem/path
233:     #
234:     # If you include a trailing / on /webpath then the server will
235:     # require it to be present in the URL. You will also likely
236:     # need to provide a <Directory> section to allow access to
237:     # the filesystem path.
238:
239:     #
240:     # ScriptAlias: This controls which directories contain server scripts.
241:     # ScriptAliases are essentially the same as Aliases, except that
242:     # documents in the target directory are treated as applications and
243:     # run by the server when requested rather than as documents sent to the
244:     # client. The same rules about trailing "/" apply to ScriptAlias
245:     # directives as to Alias.
246:     #
247:     ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
248:
249: </IfModule>
250:
251: #
252: # "/var/www/cgi-bin" should be changed to whatever your ScriptAliased
253: # CGI directory exists, if you have that configured.
254: #
255: <Directory "/var/www/cgi-bin">
256:     AllowOverride None
257:     Options None
258:     Require all granted
259: </Directory>
260:
261: <IfModule mime_module>
262:     #
263:     # TypesConfig points to the file containing the list of mappings from
```

```
264: # filename extension to MIME-type.
265: #
266: TypesConfig /etc/mime.types
267:
268: #
269: # AddType allows you to add to or override the MIME configuration
270: # file specified in TypesConfig for specific file types.
271: #
272: #AddType application/x-gzip .tgz
273: #
274: # AddEncoding allows you to have certain browsers uncompress
275: # information on the fly. Note: Not all browsers support this.
276: #
277: #AddEncoding x-compress .Z
278: #AddEncoding x-gzip .gz .tgz
279: #
280: # If the AddEncoding directives above are commented-out, then you
281: # probably should define those extensions to indicate media types:
282: #
283: AddType application/x-compress .Z
284: AddType application/x-gzip .gz .tgz
285:
286: #
287: # AddHandler allows you to map certain file extensions to "handlers":
288: # actions unrelated to filetype. These can be either built into the server
289: # or added with the Action directive (see below)
290: #
291: # To use CGI scripts outside of ScriptAliased directories:
292: # (You will also need to add "ExecCGI" to the "Options" directive.)
293: #
294: #AddHandler cgi-script .cgi
295:
296: # For type maps (negotiated resources):
297: #AddHandler type-map var
298:
299: #
300: # Filters allow you to process content before it is sent to the client.
301: #
302: # To parse .shtml files for server-side includes (SSI):
303: # (You will also need to add "Includes" to the "Options" directive.)
304: #
305: AddType text/html .shtml
306: AddOutputFilter INCLUDES .shtml
307: </IfModule>
308:
309: #
310: # Specify a default charset for all content served; this enables
311: # interpretation of all content as UTF-8 by default. To use the
312: # default browser choice (ISO-8859-1), or to allow the META tags
313: # in HTML content to override this choice, comment out this
314: # directive:
315: #
316: AddDefaultCharset UTF-8
317:
318: <IfModule mime_magic_module>
319: #
320: # The mod_mime_magic module allows the server to use various hints from the
321: # contents of the file itself to determine its type. The MIMEMagicFile
322: # directive tells the module where the hint definitions are located.
323: #
324: MIMEMagicFile conf/magic
325: </IfModule>
326:
327: #
328: # Customizable error responses come in three flavors:
329: # 1) plain text 2) local redirects 3) external redirects
```

```
330: #
331: # Some examples:
332: #ErrorDocument 500 "The server made a boo boo."
333: #ErrorDocument 404 /missing.html
334: #ErrorDocument 404 "/cgi-bin/missing_handler.pl"
335: #ErrorDocument 402 http://www.example.com/subscription_info.html
336: #
337: #
338: #
339: # EnableMMAP and EnableSendfile: On systems that support it,
340: # memory-mapping or the sendfile syscall may be used to deliver
341: # files. This usually improves server performance, but must
342: # be turned off when serving from networked-mounted
343: # filesystems or if support for these functions is otherwise
344: # broken on your system.
345: # Defaults if commented: EnableMMAP On, EnableSendfile Off
346: #
347: #EnableMMAP off
348: EnableSendfile on
349:
350: # Supplemental configuration
351: #
352: # Load config files in the "/etc/httpd/conf.d" directory, if any.
353: IncludeOptional conf.d/*.conf
354:
```