Apache configuration file - .htaccess

What is the .htaccess file?

.htaccess is the Apache web server configuration file (filename starts with a dot for a hidden file), which is placed within the root directory of the webspace (e.g. /var/www/vhosts/website/htdocs/.htaccess) of the website and read and loaded by the web server.

This file can alter the existing configuration and it is commonly used to set the 301 redirect, the PHP interpreter version, custom error pages, password protected pages and many other.

You can only have one .htaccess configuration file per directory. There is no need to copy the .htaccess file to subdirectories as .htaccess file works recursively (additional directives apply for main directory and all subdirectories).

.htaccess file is not compatible with other Linux web servers such as nginx or lighttpd.

Syntax for this file is generally the same to what you would add to the main Apache configuration file.

Any user, who is able to modify the webspace can add their own .htaccess file, this may be a security concern. It also makes the life of the system administrator harder, as he may not be aware of the changes, making it difficult to find out why there is a conflicting configuration in place, which usually results in a 503 Service Unavailable Error.

How to disable the .htaccess file?

Add the directive below to the Apache configuration file:

```
AllowOverride None
```

As per Apache documentation, if possible it is recommended to add additional configuration directly to the main configuration file of the web server. This may not be possible when multiple websites are sharing one web-server (shared hosting, reseller accounts). Due to the fact that .htaccess is parsed on each request, it may slow down the website noticeably, especially when .htaccess has many lines, or there are multiple configuration files in subdirectories.

.htaccess file examples:

Block a specific IP address from accessing your webspace:

To block an IP address from connecting to your website (in our example we will use 8.8.8.8 (Google's public DNS resolver), you need to create a .htaccess with the code below, try it with your local IP address:

```
Order Deny,Allow Deny from 8.8.8.8
```

Apache web server does not need to be restarted or reloaded to parse the new configuration and it would work as soon as the page is reloaded.

How to deny all except for one IP address:

This will block all IPs from accessing the webspace except 8.8.8.8.

```
order deny,allow deny from all allow from 8.8.8.8
```

(perhaps replace the 8.8.8.8 IP with your own.)

How to do a 301 redirect to https:

You can use the .htaccess to redirect users to the https version of your website (will show a padlock if you have an SSL installed). In the below example we are redirect anyone using serverdown.co.uk or www.serverdown.co.uk to visit https://serverdown.co.uk/ Adapt yours according to your needs.

```
RewriteEngine On
RewriteCond %{SERVER_PORT} !=443
RewriteRule ^(.*)$ https://serverdown.co.uk/$1 [R=301,L]
```