

HTTP/2

HTTP/2 is a major revision of the HTTP network protocol and it focuses on performance improvements. Its goal is to reduce the latency as well as to make the web applications faster by allowing multiple concurrent requests between the web browser and the server across a single TCP connection.

- HTTP/2 is binary, instead of textual.
- It is fully multiplexed, sending multiple requests in parallel over a single TCP connection.
- It uses header compression HPACK to reduce overhead.
- It allows servers to “push” responses proactively into client caches instead of waiting for a new request for each resource
- It uses the new ALPN extension which allows for faster-encrypted connections since the application protocol is determined during the initial connection.
- It reduces additional round trip times (RTT), making your website load faster without any optimization.
- Domain sharding and asset concatenation is no longer needed with HTTP/2.

HTTP/2 introduces other improvements, more details: [HTTP/2 RFC7540](https://tools.ietf.org/html/rfc7540)

Apache Server

Minimum supported version: Apache 2.4.24

Enable HTTP/2 on Apache:

```
# a2enmod http2
# systemctl restart apache2
```

Add Protocols h2 h2c http/1.1 to the server configuration or if you are enabling HTTP/2 for an individual virtual host, then you need to add the Protocols under respective VirtualHost.

- h2 – instructing Apache to support HTTP/2 protocol over SSL/TLS
- h2c – instructing Apache to support HTTP/2 over TCP
- http/1.1 – if a client doesn't accept HTTP/2 then serve the request over HTTP/1.1

```
# systemctl reload apache2
```

Problem if HTTP/2 modul doesn't load

Grund: Das Apache Modul mpm_prefork wird nicht von HTTP/2 unterstützt und ist ein deb. von libapache2-mod-php7.3

Error-log:

```
[Wed Aug 28 16:01:59.902395 2019] [http2:warn] [pid 16731] AH10034: The mpm
```

module (prefork.c) is not supported by mod_http2. The mpm determines how things are processed in your server. HTTP/2 has more demands in this regard and the currently selected mpm will just not do. This is an advisory warning. Your server will continue to work, but the HTTP/2 protocol will be inactive.

Lösung: Umbau von libapache2-mod-php7.3 zu libapache2-mod-fcgid

```
# apt-get install libapache2-mod-fcgid php7.3-fpm
# a2dismod php7.3 mpm_prefork
# a2enmod actions mpm_event fcgid proxy_fcgi alias setenvif
# systemctl restart apache2

# a2enconf php7.3-fpm
# systemctl reload apache2
```

Reverse Proxy

Before starting enable: proxy_http2 module (# a2enmod proxy_http2 && systemctl reload apache2)

The examples below demonstrate how to configure HTTP/2 for backend connections for a reverse proxy.

```
HTTP/2 (TLS)
ProxyPass "/app" "h2://app.example.com"
ProxyPassReverse "/app" "https://app.example.com"
```

```
HTTP/2 (cleartext)
ProxyPass "/app" "h2c://app.example.com"
ProxyPassReverse "/app" "http://app.example.com"
```

The schemes to configure above in ProxyPassReverse for reverse proxying h2 (or h2c) protocols are the usual https (resp. http) as expected/used by the user agent.

NGINX Server

TODO

Weiteres

Validate if server is using the HTTP/2 connection

```
# openssl s_client -alpn h2 -connect 127.0.0.1:443 -status | grep protocol
```

<https://scotthelme.co.uk/http-2-is-here/>

- <https://www.rosehosting.com/blog/how-to-enable-http2-in-nginx-on-ubuntu-and-centos/>
- https://wiki.mikejung.biz/HTTP_2.0

Last update: **2019/09/09 08:30**