

## How to enable output compression (gzipfile content compression) in nginx webserver

**Author :** admin

I have [recently installed and configured a Debian Linux server with nginx](#). Since then I've been testing around different ways to optimize the nginx performance.

In my nginx quest, one of the most crucial settings which dramatically improved the end client performance was **enabling the so called output compression** which in Apache based servers is also known as **content gzip compression**.

In Apache webserver the content gzip compression is provided by a server module called **mod\_deflate**.

The *output compression nginx settings* saves a lot of bandwidth and though it adds up a bit more load to the server, the plain text files like *html, xml, js and css's* download time reduces drastically as they're streamed to the browser in gzip compressed format.

This little improvement in download speed also does impact the overall end user browser experience and therefore improves the browsing speed experience with websites.

If you have already had experience nginx you already know it is a bit fastidious and you have to be very careful with it's configuration, however thankfully enabling the gzip compression was actually rather easier than I thought.

Here is **what I added in my nginx config to enable output compression**:

```
## Compression
gzip on;
gzip_buffers 16 8k;
gzip_comp_level 9;
gzip_http_version 1.1;
gzip_min_length 0;
gzip_vary on;
```

Important note here is that need to add this code in the nginx configuration block starting with:

```
http {
....
## Compression
gzip on;
gzip_buffers 16 8k;
gzip_comp_level 9;
gzip_http_version 1.1;
gzip_min_length 0;
gzip_vary on;
```

In order to load the gzip output compression as a next step you need to restart the nginx server, either by it's init script if you use one or by killing the old nginx server instances and starting up the nginx server binary again:

I personally use an init script, so restarting nginx for me is done via the cmd:

```
debian:~# /etc/init.d/nginx restart
```

Restarting nginx: nginx.

Now to test if the output gzip compression is enabled for nginx, you can simply use **telnet**

```
hipo@linux:~$ telnet your-nginx-webserver-domain.com 80
```

Escape character is '^']'.

After the **Escape character is set '^']'** appears on your screen type in the blank space:

```
HEAD / HTTP/1.0
```

and press enter twice.

The output which should follow should look like:

```
HTTP/1.1 200 OK
```

```
Server: nginx
```

```
Date: Fri, 08 Apr 2011 12:04:43 GMT
```

```
Content-Type: text/html
```

```
Content-Length: 13
```

```
Last-Modified: Tue, 22 Mar 2011 15:04:26 GMT
```

```
Connection: close
```

```
Vary: Accept-Encoding
```

```
Expires: Fri, 15 Apr 2011 12:04:43 GMT
```

```
Cache-Control: max-age=604800
```

```
Accept-Ranges: bytes
```

The whole transaction with telnet command issued and the nginx webserver output should look like so:

```
hipo@linux:~$ telnet your-nginx-webserver-domain.com 80
```

```
Trying xxx.xxx.xxx.xxx...
```

```
Connected to your-nginx-webserver-domain.com
```

```
.Escape character is '^']'.
```

```
HEAD / HTTP/1.0
```

```
HTTP/1.1 200 OK
```

```
Server: nginx
```

```
Date: Fri, 08 Apr 2011 12:04:43 GMT
```

```
Content-Type: text/html
```

```
Content-Length: 13
```

```
Last-Modified: Tue, 22 Mar 2011 15:04:26 GMT
```

Connection: close

Vary: Accept-Encoding

Expires: Fri, 15 Apr 2011 12:04:43 GMT

Cache-Control: max-age=604800

Accept-Ranges: bytes

The important message in the returned output which confirms your nginx output compression is properly configured is:

Vary: Accept-Encoding

If this message is returned by your nginx server, this means your nginx now will distribute it's content to it's clients in compressed format and apart from the browsing boost a lot of server and client bandwidth will be saved.