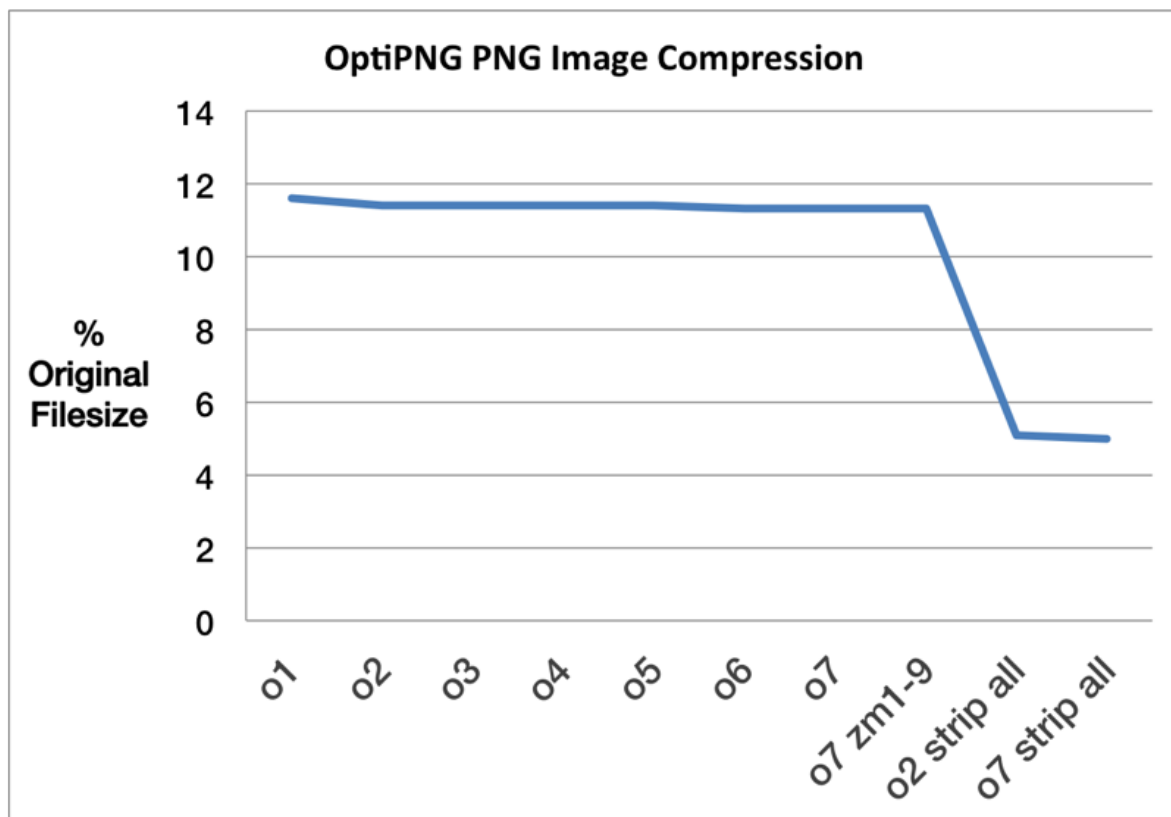


Optimize PNG images by compressing on GNU / Linux, FreeBSD server to Improve Website overall Performance

Author : admin



If you own a website with some few hundreds of .PNG images like 10 000 / 15 000 png images and the website shows to perform slow in [Google PageSpeed Insights](#) and is slow to open when Google Searched or Shared on **Facebook** / **Twitter** etc. then one recommended step to boost up the website opening speed is to compress (optimize) the .PNG pictures without losing the images quality to both save space and account bandwidth you could use **optipng** even though this is not the only tool available to help you optimize and reduce the size of your images, some few other tools you might like to check out if you have more time are:

- a.) [pngcrush](#) – optimizes PNG (Portable Network Graphics) files.
- b.) [pngnq](#) – tool for optimizing PNG (Portable Network Graphics) images. It is a tool for quantizing PNG images in RGBA format.
- c.) [pngquant](#) – PNG (Portable Network Graphics) image optimising utility. It is a command-line utility for converting 24/32-bit PNG images to paletted (8-bit) PNGs.

1. Install and Compress / optimize PNG / GIF / PNM / TIFF file format with optipng

[OPTIPNG tool](#) recompresses the .PNG images to a smaller size without losing any quality information, besides PNG file format it also supports (BMP, GIF, PNM and TIFF) image format.

If you don't have optipng installed on your server you can;

a.) install it on Redhat RPM based Linux distributions lets say CentOS Linux use:

```
[root@centos: ~]# yum install epel-release  
[root@centos: ~]# yum install optipng
```

Note that, You will need to first enable epel repo on centos 7

b.) If instead you're on a **Debian GNU / Linux**

```
debian:~# apt-get install optipng
```

c.) **FreeBSD** users can install it from *FreeBSD ports* with:

```
freebsd# cd /usr/ports/graphics/optipng  
freebsd# make install clean
```

optipng syntax is quite self explanatory
optipng [options] **what-ever-file.png**

You can get a full list of possible command options with **-?** command, here is a list:

```
debian:~# optipng -?
```

Synopsis:

```
optipng [options] files ...
```

Files:

Image files of type: PNG, BMP, GIF, PNM or TIFF

Basic options:

- ?**, **-h**, **-help** show this help
- o** optimization level (0-7) [default: 2]
- v** run in verbose mode / show copyright and version info

General options:

- backup**, **-keep** keep a backup of the modified files
- clobber** overwrite existing files
- fix** enable error recovery
- force** enforce writing of a new output file
- preserve** preserve file attributes if possible
- quiet**, **-silent** run in quiet mode
- simulate** run in simulation mode
- out** write output file to
- dir** write output file(s) to
- log** log messages to
- stop option switch parsing

Optimization options:

- f** PNG delta filters (0-5) [default: 0,5]
- i** PNG interlace type (0-1)
- zc** zlib compression levels (1-9) [default: 9]
- zm** zlib memory levels (1-9) [default: 8]
- zs** zlib compression strategies (0-3) [default: 0-3]
- zw** zlib window size (256,512,1k,2k,4k,8k,16k,32k)
- full** produce a full report on IDAT (might reduce speed)
- nb** no bit depth reduction
- nc** no color type reduction
- np** no palette reduction

-nx no reductions
-nz no IDAT recoding

Editing options:

-snip cut one image out of multi-image or animation files
-strip strip metadata objects (e.g. "all")

Optimization levels:

-o0	-o1 -nx -nz	(0 or 1 trials)
-o1	-zc9 -zm8 -zs0 -f0	(1 trial)
(or...)	-zc9 -zm8 -zs1 -f5	(1 trial)
-o2	-zc9 -zm8 -zs0-3 -f0,5	(8 trials)
-o3	-zc9 -zm8-9 -zs0-3 -f0,5	(16 trials)
-o4	-zc9 -zm8 -zs0-3 -f0-5	(24 trials)
-o5	-zc9 -zm8-9 -zs0-3 -f0-5	(48 trials)
-o6	-zc1-9 -zm8 -zs0-3 -f0-5	(120 trials)
-o7	-zc1-9 -zm8-9 -zs0-3 -f0-5	(240 trials)
-o7 -zm1-9	-zc1-9 -zm1-9 -zs0-3 -f0-5	(1080 trials)

Notes:

The combination for -o1 is chosen heuristically.

Exhaustive combinations such as "-o7 -zm1-9" are not generally recommended.

Examples:

optipng file.png	(default speed)
optipng -o5 file.png	(slow)
optipng -o7 file.png	(very slow)

Just running it with, lets say -o7 arguments is enough for optipng to compress your image and reduce some 15 to 30% of picture size

optipng -o7 what-ever-image-you-have.png

```
root@pcfreak:/var/www/images# optipng -backup -o7 zend2-powered-php.png.bak
** Processing: zend2-powered-php.png.bak
114x76 pixels, 3x8 bits/pixel, RGB
Reducing image to 8 bits/pixel, 65 colors in palette
Input IDAT size = 2531 bytes
Input file size = 2588 bytes

Trying:
  zc = 9  zm = 9  zs = 0  f = 0      IDAT size = 1679
  zc = 9  zm = 8  zs = 0  f = 0      IDAT size = 1679
  zc = 8  zm = 9  zs = 0  f = 0      IDAT size = 1679
  zc = 8  zm = 8  zs = 0  f = 0      IDAT size = 1679

Selecting parameters:
  zc = 8  zm = 8  zs = 0  f = 0      IDAT size = 1679

Output IDAT size = 1679 bytes (852 bytes decrease)
Output file size = 1943 bytes (645 bytes = 24.92% decrease)
```

2. Compress images without losing quality recursively inside directory and subdirectories with optipng

a.) To optimize all pictures inside a single directory (without sub-directories) on remote server you can run, below command:

```
cd whatever-dir/
for i in *.png; do optipng -o6 -quiet -keep -preserve -dir optimized -log optipng-compress.log "$i"; done
```

As you can see a log is being written on what the command has done and the originals of the optimized images is going to be preserved, the optimize level is 6 is the PNG encoding level.

```
cd /var/www/your-site/images/
find . -type f -iname "*.png" -print0 | xargs -I {} -0 optipng -o6 -keep -preserve -log optipng-compress.log "{}"
```

This command is pretty handy to run on own dedicated server, if you don't have one just do it on your Linux computer at home or if you don't own a PC with Linux install any Deb / RPM based Linux inside VirtualBox or VMWare Virtual Machine and do it there, then upload to your [Hosting Provider](#) / **Amazon EC2 etc and Enjoy the increased website performance :)**