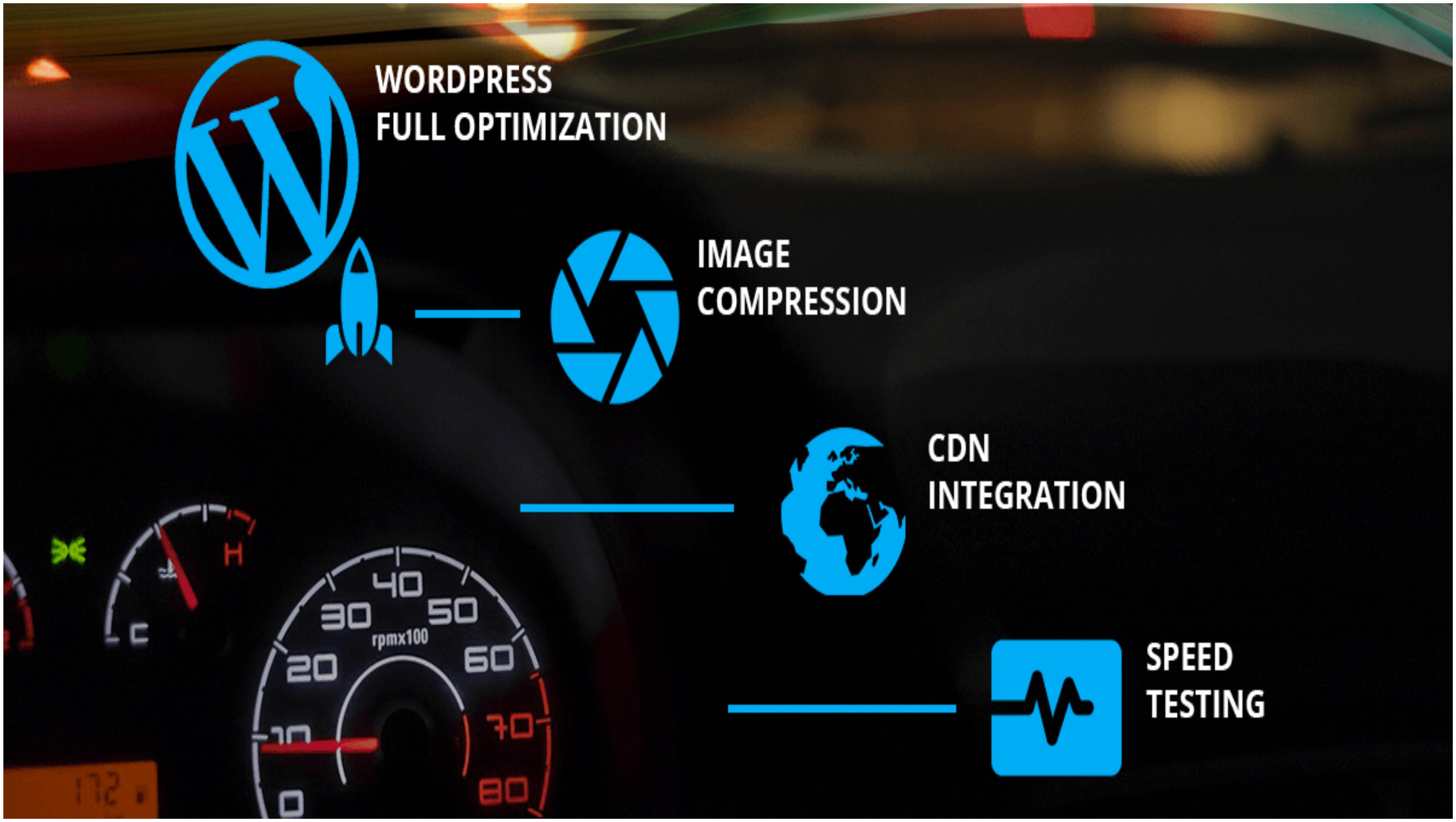

WORDPRESS
FULL OPTIMIZATION
IMAGE
COMPRESSION
CDN
INTEGRATION
SPEED
TESTING

OPTIMIZING WORDPRESS

Site and Page Load Speed

“ THERE IS NO PERFECT WEBSITE “

The Goal is Improvement

BACKUP YOUR WEBSITE
(DATABASE AND SITE'S FILES)

My workflow – redundant backups

ManageWP → Amazon S3

UpDraftPlus → Dropbox

- myToolKit
  - **ManageWP**
  - **UpDraftPlus Premium**
  - Duplicator Pro

- Manual backup
  - FTP file download – database dump
- Other popular options
  - All In One Migration
  - BackUpBuddy

BASIC CONCEPTS OF A WEBSITE

- **Domain** – Registrar, DNS – nameservers, DNSSEC, zone records (A, @,MX, CName)
- **Web hosting** – php, RAM memory, server cores, file storage, server functions
- **CMS** - Browser "asks" – Server "sends" – HTML - CSS
- **WordPress** install – wp-config.php, .htaccess, wp-content/
- Customize **Look and Feel** – themes, CSS (stylesheets)
- Additional **functionality** – plugins & custom coding – php, JavaScript, HTML
- **Security** – SSL, server & code controls, web firewall, user roles, access restriction, strong passwords, malware scanning, constant monitoring, vigilant upgrade and backup policies.
- **Performance/Optimization** – improving response time & page load speed.

ELEMENTS OF WEB PAGE

HTML, scripts, stylesheets, fonts, images, icons, code libraries, ads, tracking information, security authentication, session ids, database queries, analytics codes, geo location data, IP lookup, reverse DNS lookup, packet IDs, checksums, etc.

WHAT IS OPTIMIZATION?

Maximizing the delivery of content – the communication between a user's browser & the server where the content is stored.

- Rendering of a webpage
- Server response time.
- Page load speed.

WHY DO WE NEED OPTIMIZATION ?

- **User Experience** - Happier visitors
  - Google states, that 50% of users expect a mobile site to load in 2 seconds and 53% users are likely to abandon the page, if it's loading longer than 3 seconds.
- **Improved search engine rankings**
  - A fast website can drastically improve your Search Engine rankings.
  - A slow website will lower your search engine ranking.
  - Slow site - leads to fewer visitors.
- **Lower bounce rate**
  - A fast loading site increases chances a visitor will stay on Your page longer
- **Better conversion rates**

OPTIMIZATION CONCEPTS

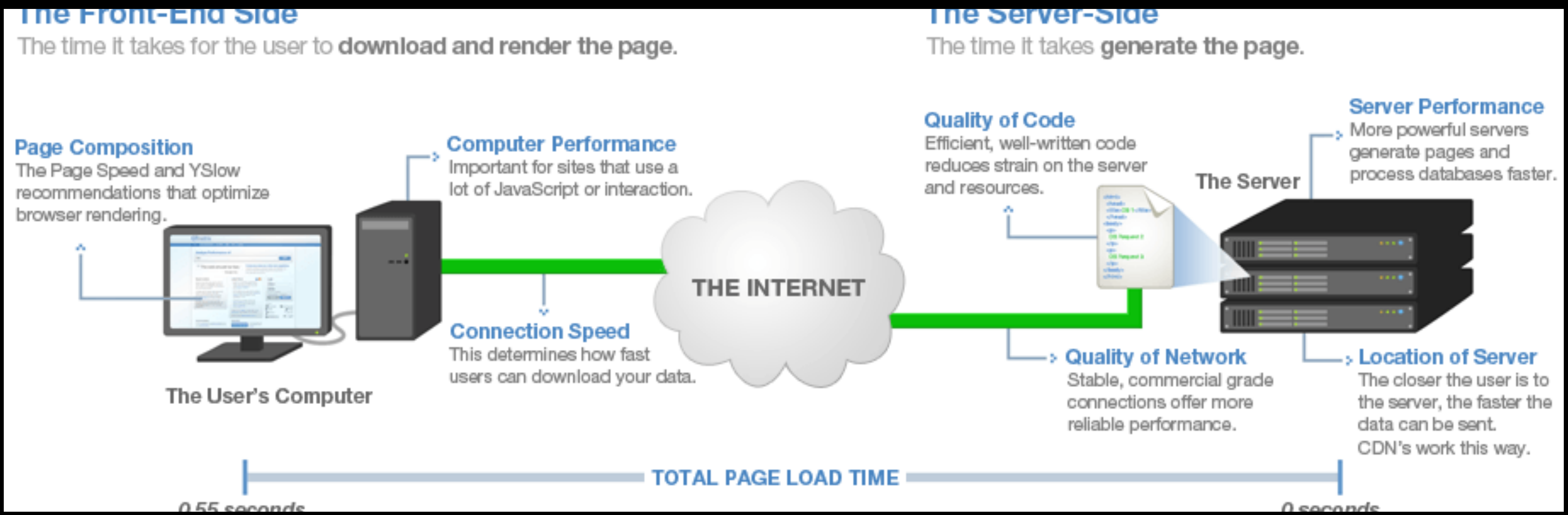
- The browser requests information from a website hosted on a server.
- Techniques used to improve **page load speed** & **server response time.**
- **Server Response** - The server sends information packets back to the browser.
- A webpage may consist of hundreds of “packets” of information being sent to your browser from website and other servers that contribute to the rendering of the webpage in your browser.
- **Page Load** – Information packets are used to render a webpage.
- HTML, scripts, stylesheets, fonts, images, icons, code libraries, ads, tracking information, security authentication, session ids, database queries, analytics codes, geo location data, IP lookup, reverse DNS lookup, packet IDs, checksums, etc.

“ OBAMA FUNDRAISING CAMPAIGN INCREASED DONATION CONVERSION BY 14% WITH SITE OPTIMIZATION, DECREASING THE PAGE LOAD TIME FROM 5 SECONDS TO 2 SECONDS. ”

http://kylerush.net/blog/meet-the-obama-campaigns-250-million-fundraising-platform/

PAGE SPEED

- A measurement of how fast the content on your page loads.
- Page speed can be described in either
  - **page load time**
    - (the time it takes to fully display the content on a specific page)
  - **time to first byte**
    - (how long it takes for your browser to receive the first byte of information from the server).
- No matter how you measure it, a faster page speed is better.


The Front-End Side
The time it takes for the user to download and render the page.
The Server-Side
The time it takes generate the page.
Page Composition
The Page Speed and YSlow recommendations that optimize browser rendering.
Computer Performance
Important for sites that use a lot of JavaScript or interaction.
Connection Speed
This determines how fast users can download your data.
The User's Computer
THE INTERNET
Quality of Code
Efficient, well-written code reduces strain on the server and resources.
The Server
Server Performance
More powerful servers generate pages and process databases faster.
Quality of Network
Stable, commercial grade connections offer more reliable performance.
Location of Server
The closer the user is to the server, the faster the data can be sent. CDN's work this way.
TOTAL PAGE LOAD TIME
0.55 seconds
0 seconds

PAGE LOAD TIME

ALWAYS MEASURE PERFORMANCE

- **Google Page Speed Insights**
  - analyzes your page’s content and suggests the areas you can improve
  - https://developers.google.com/speed/pagespeed/insights/
- **GT Metrix**
  - https://gtmetrix.com
  - https://kinsta.com/blog/gtmetrix-speed-test/
- **Pingdom**
  - https://www.pingdom.com
- WebPageTest
  - https://www.webpagetest.org
- ManageWP – Performance option

SERVER PERFORMANCE

- A good indicator of your Server-side performance is the time it takes to generate the HTML page (**page generation time**).
- This is labeled as **“Waiting” time** on the first element in the waterfall graph (also known as the “**time to first byte**”).
- Ideally this time should be kept under **one second** (or as low as possible).

DESIGN FOR SPEED

- **Page load speed** is an important factor in page ranking.
- Optimize your template to do as small an amount of database calls as necessary. (here’s where a bloated theme hurts your performance)
- Install a caching plugin.
- More magic with a CDN.
- Underpaying for hosting, is not wise. If you actually want to succeed with your link-bait actions, and want your blog to sustain high loads, go for a good hosting package.

BEHIND THE CURTAIN

DEVTOOLS

- **Timeline**
- **Network**
- **Performance**
- **Console**

- Before the browser can render the page, it needs to construct the DOM and CSSOM trees
- Browser DEVTOOLS allows us to capture and inspect the construction and processing costs of DOM and CSSOM.

CHROME DEVELOPER TOOLS

- View → Developer → Developer Tools
- Network tab – Disable Cache checkbox activated
- Resources sorted in order requested
- Performance Summary
  - Total number of requested resources
  - Total file size that was transferred
  - Loading times
- Audits tab

ERROR LOG

- **ENABLE LOGGING LINK**
  - **define('WP_DEBUG_LOG', true);**
- **OUTPUT LOGGING ON PAGE LINK**
  - **define('WP_DEBUG_DISPLAY', true);**
- More information in the WP_DEBUG codex
  - https://codex.wordpress.org/Debugging_in_WordPress
- Error log is by default saved to /wp-content/debug.log

BOOT CAMP

THE FIRST THING TO DO IS ACTUALLY WORK ON YOUR WORDPRESS INSTALLATION BEFORE YOU QUICKLY JUMP TO ADDING A CACHING PLUGIN OR EVEN DITCHING YOUR HOST FOR A NEW HOST.

CLEAN UP YOUR SITE

- Delete Unused plugins
- Delete Unused themes
- Delete Old Post Revisions
- Remove Unused Media
  - Media Cleaner
  - https://wordpress.org/plugins/media-cleaner/
- Remove Unused Tags
- Delete Pending Comments – Spam
- Cleanup Database – WP DBManager – WP Optimize

START WITH YOUR PLUGINS

- Disable all your plugins first.
- If possible switch to a default theme like the TwentyX series.
- Is your site still slow?
- Could be your hosting company.
- If it speeds up, start turning on plugins one by one to pin-point which one might be the culprit.
- Do the same with your theme(s).

- And **BACKUP** – **BACKUP** - **BACKUP**

IF YOUR SPEED ISSUES ARE PLUGIN RELATED

- Consult with the plugin developer
- Switch to a different plugin from a known trusted author with the same functionality.

CHECK YOUR BROWSER'S INSPECTOR NETWORK TAB

- Sometimes there's a rogue URL or render blocking Javascript that could be causing the slow down.
- It might have nothing to do with your WP or host and entirely a bad JS reference or script.

CHECK HOSTING

- Make sure you're taking full advantage of your hosting's capability.
- Make sure you're using PHP 7.X,
- Make sure your memory usage is configured properly
- Make sure the proper PHP extensions are available.
- This gets technical and should be something a good host will help you with.

CONSIDER A NEW HOST

- Whether fully managed like WP Engine or more on the unmanaged but WP optimized host - if you like the technical stuff.
- Just make sure they offer a solid performance with a different value proposition that “cheap” pricing.
- Let the new (potential) hosting company know what issues you're facing.
- They’ll should be able to go pretty far in helping you.

ANY QUESTIONS?

BASIC TERMINOLOGY

- Optimize the delivery of static resources
  - Caching – page, object, browser, database, static
  - Render Blocking - Defer javascript loading
- Image Optimization
- Minification (minify)
- File compression
- Server improvements - PHP, memory

OPTIMIZATION TECHNIQUES

- **HOSTING UPGRADE**
- **CACHING**
  - Leverage browser caching
  - Page Caching | The Key to Internet-Scale Traffic is Reverse-Proxy Caching
  - Object Caching Speeds Up Dynamic Page views
- **ASSET MANAGEMENT** (images/CSS/JS)
  - reduce impact or conflict of various aspects of a bloated site with many features.
  - Merge, **MINIFICATION**, async, include/exclude, etc.
  - Defer **RENDER-BLOCKING** JavaScript
  - Optimize images
- **DATABASE** Performance
  - Query Performance
  - Removed Query Strings from Static Resources
- Enable **GZIP Compression**
- Upgrade **php** → **7.x**

OPTIMIZATION CHECKLIST

- Choose a FAST and Reliable Web Host
- Opt for a Fast WordPress Theme
- Enable **gZIP** Compression (to compress your data files)
- Enable **page cache**
- Enable **database cache**
- Enable **object cache**
- Enable Leverage **Browser Caching**
- Optional - Use a Content Delivery Network (CDN)
- **Optimize Images** Automatically – **ShortPixel** or Imagify
- **Minify** Your WordPress Website (optimization or minify plugin)
- Add Lazy Load to Your Images & Videos
- (Optional) Optimize Your WordPress Database
  - post revisions, spam, drafts, tables, etc. → **WP Optimize**, WP-DB Manager
  - Delete or Control Post Revisions
  - Disable Pingbacks and Trackbacks - reduce the excessive load which your website receives every time a blog mentions your site
- **Test/measure your site after every step.**

KNOW YOUR HOSTING

- Shared services vs VPS - user loading – how many sites on server
- Server cores – allocated memory – SSD drives – datacenter location
- Caching – server side caching services
  - Memcache
  - Redis
- Proprietary caching is used by some Managed WP hosting
  - SiteGround
  - WPEngine
  - getFlywheel
- Cloud based
- CDN
- Your Hosting decision goes hand-in-hand with learning how to best optimize your WordPress site.

WHAT IS A CDN?

- CDN = content delivery network
- A CDN caches your site as static content.
- Static content is files like HTML, CSS, JavaScript and image files that stay the same for every user.
- By making these static files available on a CDN with many servers all around the world, you can get them to your visitors super fast.
- Content delivered quickly & efficiently, based on geographic location.

WHY USE CDN?

- A CDN speeds up your site - Optimizes
  - Caching, Compression, Asynchronous JavaScript Loading
- A CDN reduces bandwidth
- A CDN can add a layer of security to your site
- A CDN is scalable and improves availability and uptime

HOSTING GOTCHAS

- Optimization techniques can cause an increase in the use of memory resources on the hosting server and disk writes (disk i/o).
- Consumes more RAM on your server, thus shared host provider may frown upon the consistent usage of system memory if you have a high traffic site.
- Guess what the shared host provider typically says about it?
- Your stuff is consuming an unusual high amount of system memory, we cannot allow this!

CACHING

CACHING

- **BROWSER CACHE** – cheap way to cache by saving information in the user's browser so they don't have to re-download. It's very fast since files are loaded locally from user's browser but tricky to force it to load updated pages.
- **OBJECT CACHE** – built for speeding up already-optimized sites (max speed and minimal features), or for sites requiring 100% dynamic pages. It works by reducing common database calls by storing these "objects" in memory or disk.
- **STATIC/FULL-PAGE CACHING** – reduces server load tremendously (increased efficiency serves more visitors simultaneously). Can be done at server-level (LiteSpeed Cache, Varnish) which is faster, or at php-level which is slower.
- *Disk: Basic* and *Disk: Enhanced* are two method that works great under shared hosting plan. However, if the disk activity on your server is high and it's not SSD, you may achieve worse result.

DELIVER FILES FASTER

- Configure HTTP caching – browser caching
- Enable GZIP compression
- Serving Resources from Multiple Domains
- Set up a CDN
- Remove query string from static resources
  - having a query string in your CSS or JS means the browser treats them as dynamically generated content and the browser assumes it will be different every time. Therefore, these files will be never be cached. NEVER!

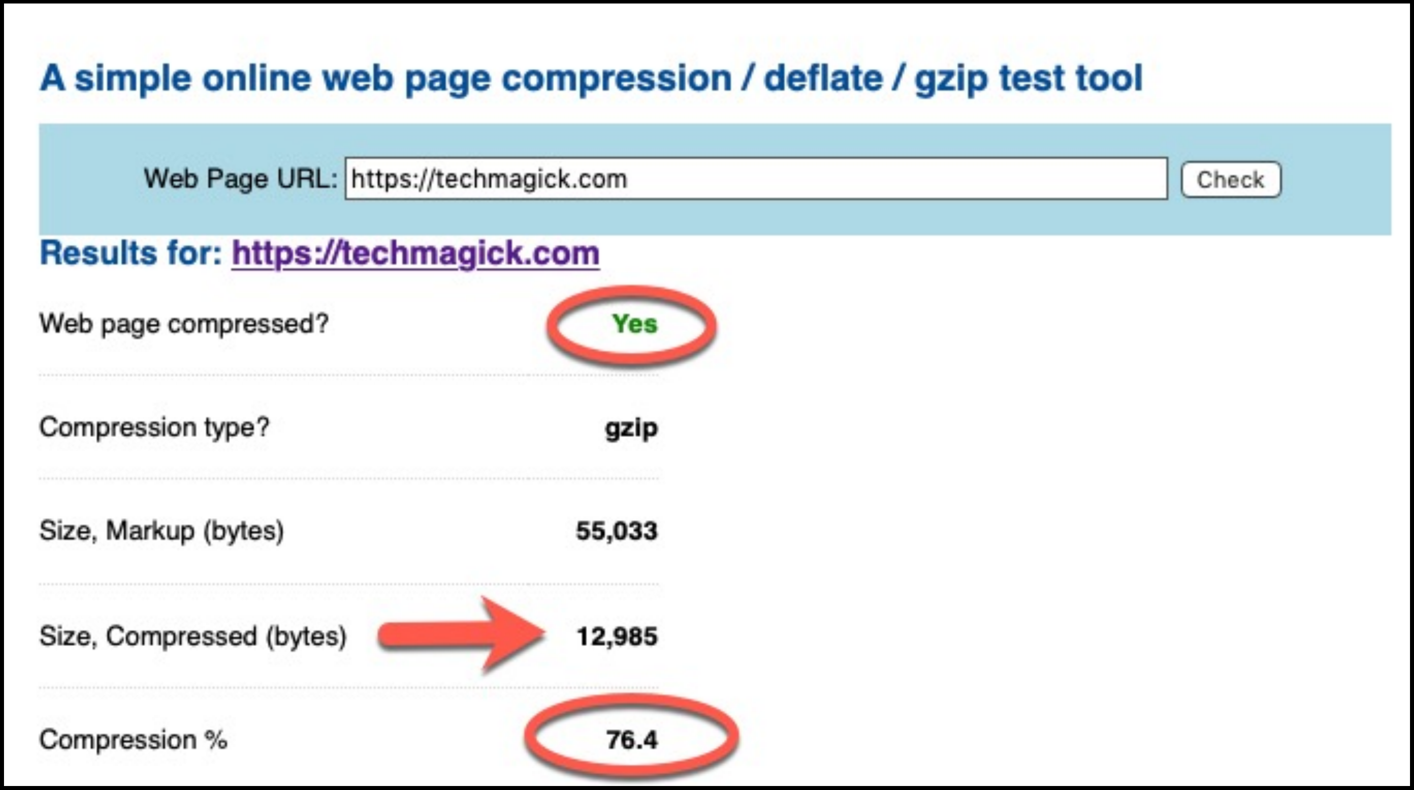
OPTIMIZING CSS & JAVASCRIPT

- CSS and JavaScript in External Files – better for efficient caching
- Minifying CSS and JavaScript
- OPTIMIZE CSS LOADING
  - CSS is included right at the top
  - Don't Use @import
  - Only CSS that is really necessary is delivered
- OPTIMIZE JAVASCRIPT LOADING
  - JavaScripts are "blocking" resources by default
  - Load Scripts at the bottom of HTML
  - Load Scripts with ´async´ attribute
  - Don’t load third-party libraries multiple times!
  - load JavaScript libraries from Google’s Hosted Libraries CDN

PAGE COMPRESSION

- Apache actually has two compression options:
- **mod_deflate** is easier to set up and is standard.
- **mod_gzip** seems more powerful: you can pre-compress content.
- https://betterexplained.com/articles/how-to-optimize-your-site-with-gzip-compression/
- Online GZIP testing: http://www.gidnetwork.com/tools/gzip-test.php
-

GZIP COMPRESSION TEST



A simple online web page compression / deflate / gzip test tool

Web Page URL: https://techmagick.com Check

Results for: https://techmagick.com

Web page compressed?	Yes
Compression type?	gzip
Size, Markup (bytes)	55,033
Size, Compressed (bytes)	12,985
Compression %	76.4

GZIP COMPRESSION WITH W3 TOTAL CACHE

- Go to the *W3 Total Cache Plugin Settings page*
- Navigate to *Browser Cache page*
- *Check the Enable HTTP Compression option*
- *Click on Save changes button* and you are done.

ENABLING GZIP COMPRESSION IN APACHE

```
# BEGIN GZIP COMPRESSION
<IfModule mod_gzip.c>
mod_gzip_on Yes
mod_gzip_dechunk Yes
mod_gzip_item_include file \.(html?|txt|css|js|php|pl)$
mod_gzip_item_include handler ^cgi-script$
mod_gzip_item_include mime ^text/.*
mod_gzip_item_include mime ^application/x-javascript.*
mod_gzip_item_exclude mime ^image/.*
mod_gzip_item_exclude rspheader ^Content-Encoding:.*gzip.*
</IfModule>
# END GZIP COMPRESSION
```

ENABLING DEFLATE COMPRESSION IN APACHE

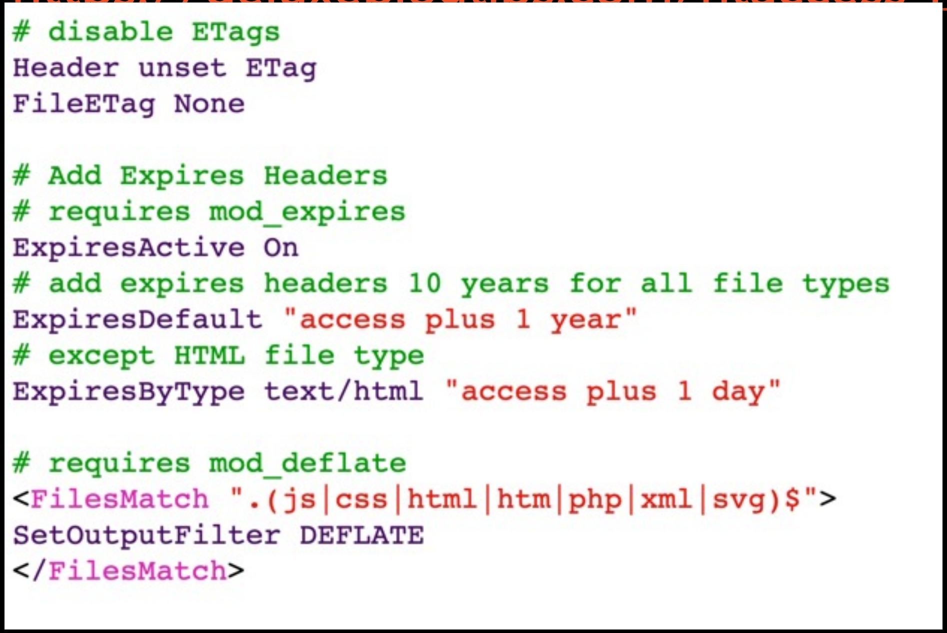
- Add the following to your .htaccess file:

```
# compress text, html, javascript, css, xml:
AddOutputFilterByType DEFLATE text/plain
AddOutputFilterByType DEFLATE text/html
AddOutputFilterByType DEFLATE text/xml
AddOutputFilterByType DEFLATE text/css
AddOutputFilterByType DEFLATE application/xml
AddOutputFilterByType DEFLATE application/xhtml+xml
AddOutputFilterByType DEFLATE application/rss+xml
AddOutputFilterByType DEFLATE application/javascript
AddOutputFilterByType DEFLATE application/x-javascript

# Or, compress certain file types by extension:
<files *.html>
SetOutputFilter DEFLATE
</files>
```

.HTACCESS
RULES FOR BETTER WEBSITE PERFORMANCE

- https://wordpress.org/plugins/rewrite-rules-inspector/
- https://deluxebloggtips.com/htaccess-rules-website-performace/



```
# disable ETags
Header unset ETag
FileETag None

# Add Expires Headers
# requires mod_expires
ExpiresActive On
# add expires headers 10 years for all file types
ExpiresDefault "access plus 1 year"
# except HTML file type
ExpiresByType text/html "access plus 1 day"

# requires mod_deflate
<FilesMatch ".(js|css|html|htm|php|xml|svg)$">
SetOutputFilter DEFLATE
</FilesMatch>
```

Mod_deflate is an Apache module which allows output from your web server to be compressed before being sent to the client.

BLOCKING SCRIPT RESOURCES

- Before the browser can render a page it has to build the DOM tree by parsing the HTML markup.
    - DOM = Document Object Model
- Whenever the parser encounters a script it has to stop and execute it before it can continue parsing the HTML.
- HTML references a blocking external JavaScript file in the above-the-fold portion of your page
- In the case of an external script the parser is also forced to wait for the resource to download, which may incur one or more network roundtrips and delay the time to first render of the page.

RENDER-BLOCKING JAVASCRIPT

- Properly enqueued JavaScript shows up in the head section of your HTML document.
- On the internet as in nature, the main thing about a head is that it's above a body—and this means something fairly serious for site speed, because JavaScript can be render-blocking.
- Long JavaScript files in your head can delay your browser from displaying page content, because its default behavior is first to interpret the JS files themselves.
- In other words, JS is blocking the browser's crucial function of rendering the page out for the user to actually see.
- "Render-blocking" comes from a web browser's default behavior: It wants to completely receive and process everything that's come higher up in the page, before it moves any further down.
- The result can be slow sites and frustrated users.

BLOCKING CSS RESOURCES

- Before the browser can render content it must process all the style and layout information for the current page.
  - Process HTML markup and build the Data Object Model (DOM) tree.
  - Process CSS markup and build the CSSOM tree.
  - Combine the DOM and CSSOM into a render tree.
  - Run layout on the render tree to compute geometry of each node.
  - Paint the individual nodes to the screen.
- As a result, the browser will block rendering until external stylesheets are downloaded and processed, which may require multiple roundtrips and delay the time to first render.

MINIFICATION

- **MINIFICATION**
  - Removes unnecessary white spaces and extra comment lines in the source of the HTML webpage, JavaScript and stylesheet files to achieve a smaller size for these components.
  - That means files can be downloaded faster when the browser requests!
- COMBINING
  - What it does is it processes and merges multiple JavaScript files or CSS files into just one or two files.
- **NOTE: Minification and Combining often break your website**
- Some JavaScript or CSS just doesn't like to be combined/minified or due to misconfiguration.
- Hence this process is a trial and error test to figure out a way to make it work and hope everything play nice with each other.
  - Useful plugin combination: WP Super Cache + Autoptimize

CACHING PLUGINS

- **W3 Total Cache**
  - GZIP compression to optimize web browser rendering.
  - Minification and concatenation of HTML, CSS and JavaScript files.
  - Support for Content Delivery Networks (CDN).
  - Compatible with CloudFlare.
- WP Rocket
  - Page caching is immediately activated.
  - Google Fonts Optimization minimizes HTTP requests.
  - JavaScript files are deferred till the page is rendered.
  - Integrates seamlessly with CloudFlare.
  - Image lazy load

CACHING PLUGIN #2

- Swift Performance Lite
  - Page caching works out of the box.
  - compatible with WooCommerce, bbPress, Cloudflare and Varnish.
  - CSS and Javascript optimization
  - Database Optimization
  - Plugin Organizer -- i.e. Contact 7 loading rules
- WP Super Cache (use with Au
  - Generates/caches static HTML files
  - It offers three options for decreasing load times:
    - Use mod_rewrite to deliver static pages
    - Serve static pages using PHP
    - Use a legacy caching mode that caches pages for users who are logged in
  - Page compression and dynamic caching.
  - Support for Content Delivery Networks (CDN).
  - Caching for visitors using a mobile device.
  - Scheduler to manage deletion and re-caching at given intervals.

CACHING PLUGINS

- W3 Total Cache
  - https://wordpress.org/plugins/w3-total-cache/
- Swift Performance Lite
  - https://wordpress.org/plugins/swift-performance-lite/
- Autoptimize
  - https://wordpress.org/plugins/autoptimize/
- WP Super Cache
  - https://wordpress.org/plugins/wp-super-cache/
- WP Rocket
  - https://wp-rocket.me
- Breeze – WordPress Cache Plugin
  - https://wordpress.org/plugins/breeze/

W3 TOTAL CACHE
A COMPLETE OPTIMIZATION TOOL

- First & foremost W3 Total Cache allows your dynamic WordPress website to be served via static HTML pages.
- These static pages don’t require additional database calls.
- There is no PHP for the server to process, and as such they load much faster.
- Whenever you change a page or update a blog post – the cached files get regenerated.

W3 TOTAL CACHE

- Steep learning curve
- You either understand how it works, or ended up hating it.
- A majority of people would give W3 Total Cache a bad rap because it makes their site slower.
- In many cases, it breaks the whole thing then runs away.
- Turned out, the issue lies with the hosting service that they were using, especially under shared host environment, not the plugin itself.
- W3 Total Cache has a bunch of advanced options that if not configured carefully, could become a two-headed knife problem to your website performance.
- With that in mind, if you are currently using a slow shared host plan use WP Super Cache instead.

MINIFICATION PLUGINS

- Position your scripts and CSS in the most optimal location for a faster sequential load
- Autoptimize
  - aggregate, minify and cache scripts and styles, injects CSS in the page head by default but can also inline critical CSS and defer the aggregated full CSS, moves and defers scripts to the footer and minifies HTML.
- Swift performance lite
  - combines and minifies the CSS files
  - generates the Critical CSS for each page automatically.
  - combine/minify Javascripts –
  - move Javascripts to footer without any conflict.
  - unique Async Execute solution run Javascripts individually as soon as a chunk has been downloaded
- W3 total cache
  - Minification of posts and pages and feeds
  - Minification of inline, embedded or 3rd party JavaScript
  - Minification of inline, embedded or 3rd party CSS

HEARTBEAT – WP API

- repetitive background tasks that greatly affect the overall performance of the server.
- Heartbeat Control plugin
- Swift Performance Lite
- W3 Total Cache

OPTIMIZE DATABASE

- Insure db tables use InnoDB storage engine
  - a lot of older sites are still using the MyISAM storage
  - InnoDB performs better[66] and is more reliable
  - https://wordpress.org/plugins/wp-optimize/
  - https://wordpress.org/plugins/revision-control/
- Delete Unused Plugins
  - One of the easiest and often most effective ways to reduce the clutter in your database is to delete any plugins you aren't using.
- Post Revisions - Limit the number of saved revisions
  - Add to wp-config.php
  - to disable post revision completely.
    - **define('WP_POST_REVISIONS', false);**
  - To modify the number of revisions that are kept per post/page:
    - **define('WP_POST_REVISIONS', 3);**

IMPROVE DATABASE QUERYING WITH A SEARCH INDEX

- Relevanssi – a better search engine
- Replaces the standard wordpress search
- Configurable features and options
- An improved fork of “SearchWP” plugin

WP OPTIMIZE MAXIMIZE DB EFFICIENCY.

WPOptimize

- Optimizes without the need for manual queries
- Removes all unnecessary data
  - trashed/unapproved/spam comments, stale data)
  - pingbacks, trackbacks and expired transient options
- Clean Up/Compact/de-fragment MySQL tables
- Control which optimizations you wish to carry out
- Retains a set number of weeks' data during clean-ups

MANAGEWP

- Another tool for database cleanup
- Delete revisions, comments
- Remove unused plugins and themes
- Performance monitoring options
- Backups – easy roll-backs

ANALYSIS THEME

- Is your site using a multipurpose theme but then only utilizing 1% of the theme's features or none at all.
- For basic WordPress users a more minimal theme is the way to go.
- In most scenarios huge multipurpose themes just slow a site down
- **Theme Function Snippets**
  - a collection of code modifications.
    - Remove Dash Icons For Regular Visitors
    - Remove Jetpack Device Px For Regular Visitors
    - Remove Emoji Icons
    - Remove Query String From Static Resources
    - Display Enqueued Scripts On The Page
    - Remove Local Jquery And Link To Googleapis Server Instead
    - Remove Woocommerce Ajax Cart Fragments
    - WP Cron And Heartbeat
    - https://www.custompcguide.net/badass-wordpress-optimization-tips-and-tears/#wordpress-theme-functions

SCRIPT LOADING

- prevent certain scripts from loading on specific pages and posts is to utilize a WordPress plugin like Gonzalez[86] or Plugin Organizer[87].
- the popular Contact Form 7 WordPress plugin loads it’s CSS and JavaScript file on all pages, when it is only needed on pages with a contact form.
- Use wp_dequeue_script()[83]
- an example of how to utilize the function[84] with Contact Form 7.
- Contact Form 7 documentation on how to load the JavaScript and CSS only when necessary[85]

THE FIX: DEFER AND ASYNC YOUR JAVASCRIPT

- Both methods let the browser load a JS resource “as time permits,” while attending to other things (like page rendering) as well.
- This means that you *can’t* rely on a deferred or asynced JavaScript file being in place prior to page render, as you could without these attributes—but the advantage is that the file won’t slow the speed at which the page becomes visible to users.
- In general, you can defer JavaScript files that rely on user interactions, like clicks and mouse hovers—and files that fix layout details, like center or hide a set of element. Again, these rely on a loaded page to work anyway (which is why they’re almost all going to be wrapped in jQuery(document).ready(function() { }), or else they’re liable not to work), so you should be safe to get the page out beforehand.

WHEN TO USE ASYNC INSTEAD OF DEFER

- You use async when you're linking directly to an *external* JavaScript library. That link would look something like: <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>. Notice how it's a link to the full URL, and the JavaScript will get pulled in
- enqueueing *external* JS is a lot less common, at least for us, since most of our enqueued JS is in themes and plugins that host their own code. At any rate, the code for async is precisely the same as the code for defer—but with the two words switched out. So if you do happen to have a lot of externally hosted enqueued scripts, getting them asynced is a very similar technical process to the one we've just covered.
- **WHICH SCRIPTS TO DEFER AND ASYNC**
  - Don't do anything to jQuery. jQuery (handle jQuery) is a key dependency for many other JS files, and you want to let it load early.
  - Any file that's wrapped in a jQuery(document).ready(function() { }) call should be fine to defer. That code basically says "Wait until the entire document object model (DOM) loads," so racing to get the JavaScript file loaded in the head doesn't serve much purpose.
  - In general, you can defer JavaScript files that rely on user interactions, like clicks and mouse hovers—and files that fix layout details, like center or hide a set of element. Again, these rely on a loaded page to work anyway (which is why they're almost all going to be wrapped in jQuery(document).ready(function() { }), or else they're liable not to work), so you should be safe to get the page out beforehand.
  - It's, unfortunately, impossible to use this method for JavaScript files that have been added some way other than the generally correct method of enqueueing them. This is another reason to prefer thkat method over other ways of loading scripts that may appear to work fine at first glance.

IMAGE OPTIMIZATION

- Image Compression
  - Lossless compression - reduce image size without any quality loss.
  - Lossy compression means that some data from the original image file is lost.
  - Most compression tools will let you choose the degree of compression that will be used on your images.
- Image Resizing
- IMAGE CDN - ***Cloudinary*** *offers a free service*

WORKING WITH IMAGES

- Getting Rid of Images
    - Reduce number of images by:
    - with CSS
    - Using web fonts
    - Eliminating images that you don’t need
- Use The Right Image Format
    - SVG is best suited for geometric shapes
        - convert logos, icons, diagrams, drawings, text, etc. into SVG
    - JPG for photos & screen shots
    - PNG for transparency & maximum quality
- Include Responsive Images
- Optimize Images

OPTIMIZING IMAGES

- Choose the best file format for each of your images
- A lossy format like **JPEG --** "quality" setting
- **PNG --** scale down to a smaller color palette
- **SVG – minify the content - SVGO tool**

- Loading the Right Images
  - create *separate* images for those small thumbnail previews and the big originals.

TOOLS

- ShortPixel Image Optimizer - ShortPixel Adaptive Images (image CDN)
- Imagify
- Lazy Load by WP Rocket
- a3 Lazy Load
- SMUSH
- EWWW Image Optimizer
- TinyPNG
- Autoptimize
- WP -Optimize
  - Image Optimization: removes orphaned images from your WordPress site, plus images of a certain, pre-defined size.
- Data URI
  - The main benefit to using data URIs is that you can reduce the number of HTTP requests that your site needs to make to load the page. Each individual file referenced in your CSS or HTML code will create a new HTTP request. By using data URIs, you're actually embedding the file data directly within your HTML or CSS file, so there's no need to make a HTTP request to fetch the resource.

SHORTPIXEL – IMAGE OPTIMIZATION

- ShortPixel Image Optimizer
  - A cloud-based image optimization service
  - compress all your past images and PDF documents with a single click.
  - aggressive image optimization algorithm
  - optimizing images as you upload them - new images are automatically resized/rescaled and optimized on the fly, in the background.
  - can also bulk optimize all your images in one
  - free version of ShortPixel limited to 100 images/month.
- ShortPixel Adaptive Images
  - ShortPixel Adaptive Images replaces all your website’s pics with properly sized, smartly-cropped, optimized images
  - serves them from ShortPixel’s global CDN.
    serves WebP images to the right browsers auto-magically

LAZY LOADING

- Lazy loading delays the loading of images in the browser until the user needs to see it.
- As the reader scrolls, images load asynchronously on demand.
- Images that must appear ‘above the fold’, or when the web page first appears are loaded straight away.
- Images which follow ‘below the fold’, however, are not yet visible to the user.
  - They do not have to be immediately loaded into the browser.
  - They can be loaded later – or lazy loaded – only if and when the user scrolls down and it becomes necessary to show them.

LAZY LOAD

- You can also control which images you want to lazy load, and which ones you don't!
- Start lazy-loading after the above-the-fold images have loaded, loading all of the images independent of user interaction.

LAZY LOADING BEST PRACTICES

- Best Practice - Control which images you want to lazy load, and which ones you don't!
- Lazy-loading images above the fold can make loading visibly slower, both technically and for human perception.
- Don't use Lazy-loading for the main page hero image.
- Start lazy-loading after the above-the-fold images have loaded, loading all of the images independent of user interaction.
- Screen readers (ARIA), some search bots and any users with JavaScript disabled will not be able to view images lazy loaded with JavaScript.
- Work around: a <noscript> fallback.
  - https://www.robinosborne.co.uk/2016/05/16/lazy-loading-images-dont-rely-on-javascript/

WP OFFLOAD S3 LITE + AMAZON CLOUDFRONT

- “offloads” media files - automatically copies images, videos, documents, and any other media added through WordPress’ media uploader to Amazon S3, DigitalOcean Spaces or Google Cloud Storage.
- It then automatically replaces the URL to each media file with their respective Amazon S3, DigitalOcean Spaces or Google Cloud Storage URL
- The pro upgrade has an upload tool to handle existing
- Integrates with EWWW Image Optimizer media files.

EWWW IMAGE OPTIMIZER

- will automatically optimize new images that you upload
- can also optimize all the images that you have already uploaded
- optionally convert your images to the best file format.
- You can choose pixel perfect compression or high compression options that are visually lossless.

CACHE ENABLER

- **Cache Enabler** has a menu checkbox option for caching WebP images to be served if available and the current user's browser supports them.
- Cache Enabler requires the use of a sister program, ShortPixel

- **WebP** is a recent image format from Google aiming to offer lower file-sizes for lossless and lossy compression at an acceptable visual quality. It includes support for alpha-channel transparency and animation.

IMAGE PROCESSING CDNS

- **Cloudinary and imgix**
- **Cloudinary** and **imgix** are two established image processing CDNs.

- Image CDNs like **Cloudinary** and **Imgix** both support controlling image density to serve the best density to users from a single canonical source.

RESOLVING CACHE PROBLEMS

- backup your htaccess file
- clean-up settings left behind by uninstalled cache plugins
- If your site breaks, just go into your FTP and rename the cache plugin folder.
- The default repair steps is to restore htaccess, delete “cache” directory and cache files in your “wp-contents” directory.

GET HELP

Facebook groups, support forums, WordPress codex and documentation.

Online research – technical blogs, etc.

Last resort if you get stuck, find someone in this group who you can consult with toward getting the SPEED YOU DESERVE!

MY GOTO SETUP

- **Theme** → Astra, BeaverBuilder Theme, GeneratePress
- Page Builder → Beaver Builder Pro
- **Backups** → UpdraftPlus Premium, ManageWP, Duplicator Pro
- **Optimization** → WP 3 Total Cache or Swift Performance Lite
- **Image Optimization** → ShortPixel Image Optimizer, ShortPixel Adaptive Images , Lazy Load
- Security → iThemes, WordFence