Caching: a key to WordPress Performance

Hosting Guides · https://wellbuiltnetworks.com/docs/caching-and-wordpress-performance/

WordPress offers easy page creation, flexibility, and an amazing amount of extensibility with the plugin system. That shouldn't come at the cost of performance and speed, though. Some of the biggest web sites in the world run on WordPress, and yes - they run on great dedicated hardware. But you can make any WordPress installation run *better* if you implement caching correctly.

Here at Wellbuiltnetworks we do all of this for you with our concierge hosting. Here's part of what we do, and why we do it.

Two Types of Cache

We are looking at two types of caching today: *in-memory-caching* (in the form of Redis) and *opcode caching*. These will have different effects on different web sites, but if your host offers them it is well worth the effort to set caching up and test the difference it makes to your web site's performance.

Note that there are performance/caching plugins for WordPress that do amazing things for performance (our favorites are lsCache and WP Rocket) by optimizing your code, but we are talking about actual server-level caching that can be manually enabled that allow these caching plugins to do even more than they can do by default.

Redis, our in-memory cache

Redis might not be available on your shared hosting service, due to security concerns. Redis works well, but it does *not* have robust security features built in. On dedicated servers this is fine; on shared hosting this can present some problems as what Redis does is cache data in memory for fast access, but since that memory is shared amongst all the users of the system, there are risks of having your in-memory data accessed by other sites. More on that later.

What does Redis *do*, anyway?

Redis is a cache that's stored in memory. If you're not a nerd you might not know this, but memory is:

- 1,000 times faster than an SSD
- 100,000 than a hard drive
- Somewhere between 100 and 1,000 times faster than database access

WordPress is complicated to explain, but basically when someone views your web site the web server runs the WordPress code, and that code then loads files from disk (like images) and makes a bunch of database queries. Some of those database queries are unique to each user, but most of them return the same information each time, like the name of your web site, or the words that the user sees on your landing page.

What Redis allows us to do is take those database queries that haven't changed from the last time the query was made and save them in memory so they can be accessed instantaneously the next time someone loads the page.

And *that* makes a small difference in performance if your site isn't under much load, but it makes a *huge* difference in performance for a busy site, or if your site goes viral unexpectedly.

- Page load times can decrease by 30% - 60%
- Database query loads can decrease by 50% - 90%
- Requests per second can increase by 2x to 3x. This means your site can be viewed by 200% to 300% more users with the same performance.
- Latency *can* be reduced by an order of magnitude. More realistically, a site that loads in 2-3 seconds without Redis might load in 1-2 seconds with it.

A brief look at containerization, and why we can enable Redis safely

At Wellbuiltnetworks, we are using a server management system that uses containerization to isolate every web site from every other web site. This offers us some amazing advantages:

- Vulnerabilities in one web site don't affect neighboring web sites. If a web site gets compromised, the worst the hacker can do is take over the user's container/website, but other web sites are unaffected. With most hosting compromising one host can allow an attacker to access the other sites on the host and spread the damage.
- We can control resources on a web site by web site level, so of one web site starts to take off we can identify it, and...

- ...move it to a server with more resources. The ability to migrate sites between servers with minimal down-time offers great improvements for our customer experience over a typical CPanel or DirectAdmin site where each web site is housed on a single server. In the system we use, web sites are housed on a cluster (called an Estate) and can be freely moved for maintenance, performance, or growth purposes.
- Programs like Redis can be installed inside each containerized web site, so it can only be accessed by the site it's assigned to, and the lack of security features in Redis doesn't matter, even though we are technically using shared hosting.

Enabling Redis *does* consume more memory than leaving it off (which is why our hosting platform leaves it off by default and we need to manually toggle it), but we are selling web sites that solve your problems, and well-cached web sites solve problems better than those that aren't cached.

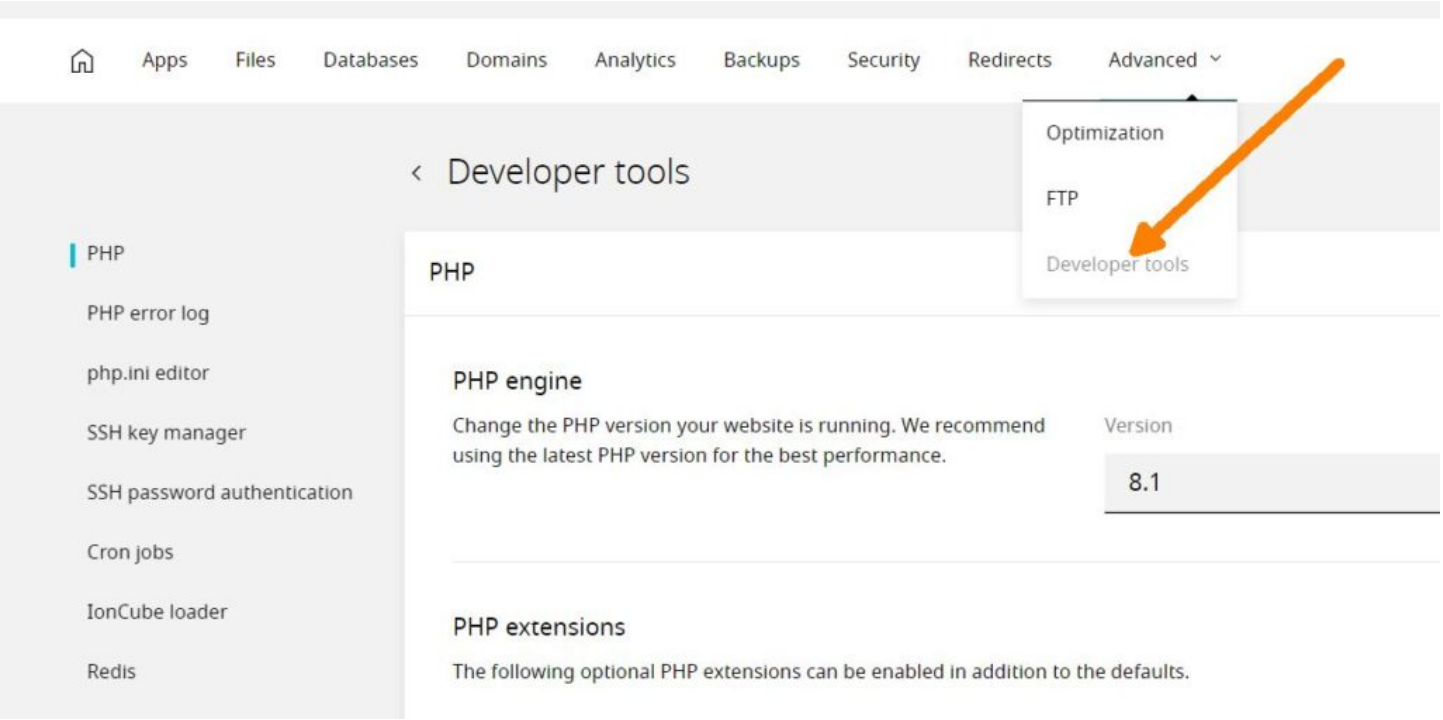
Enabling Redis in the Enhance Control Panel

There are two steps to enable Redis:

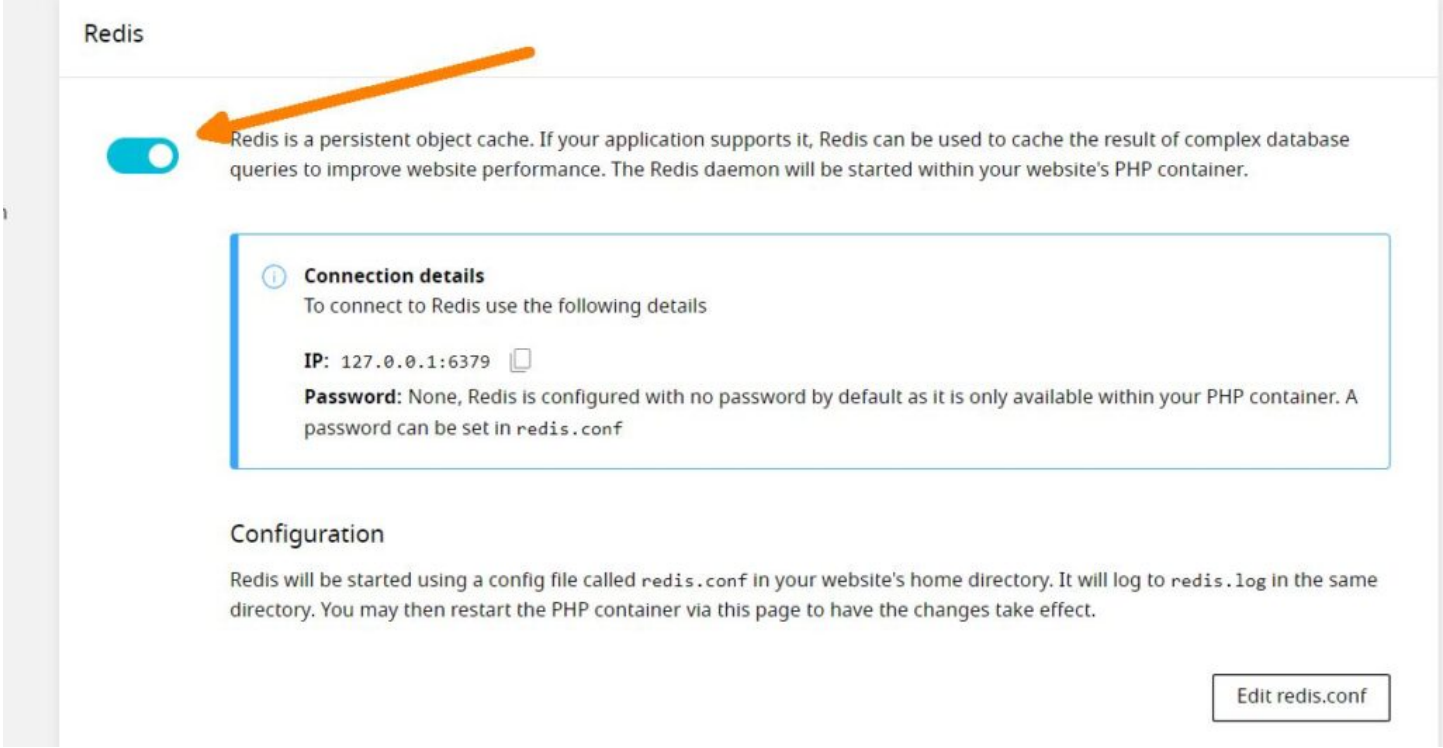
Step One: Turn it on in the control panel

Again, we do this for all of our concierge clients, but for other Enhance users here are the steps:

First, go to Developer Tools in your panel:



Then, scroll to the bottom and slide the toggle for Redis "on."

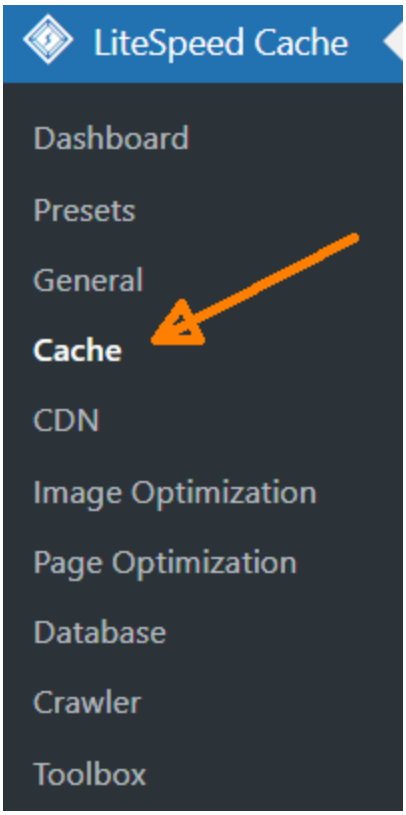


There is no need to edit the configuration file for most users. Now that you've done this, Redis is available to your WordPress web site.

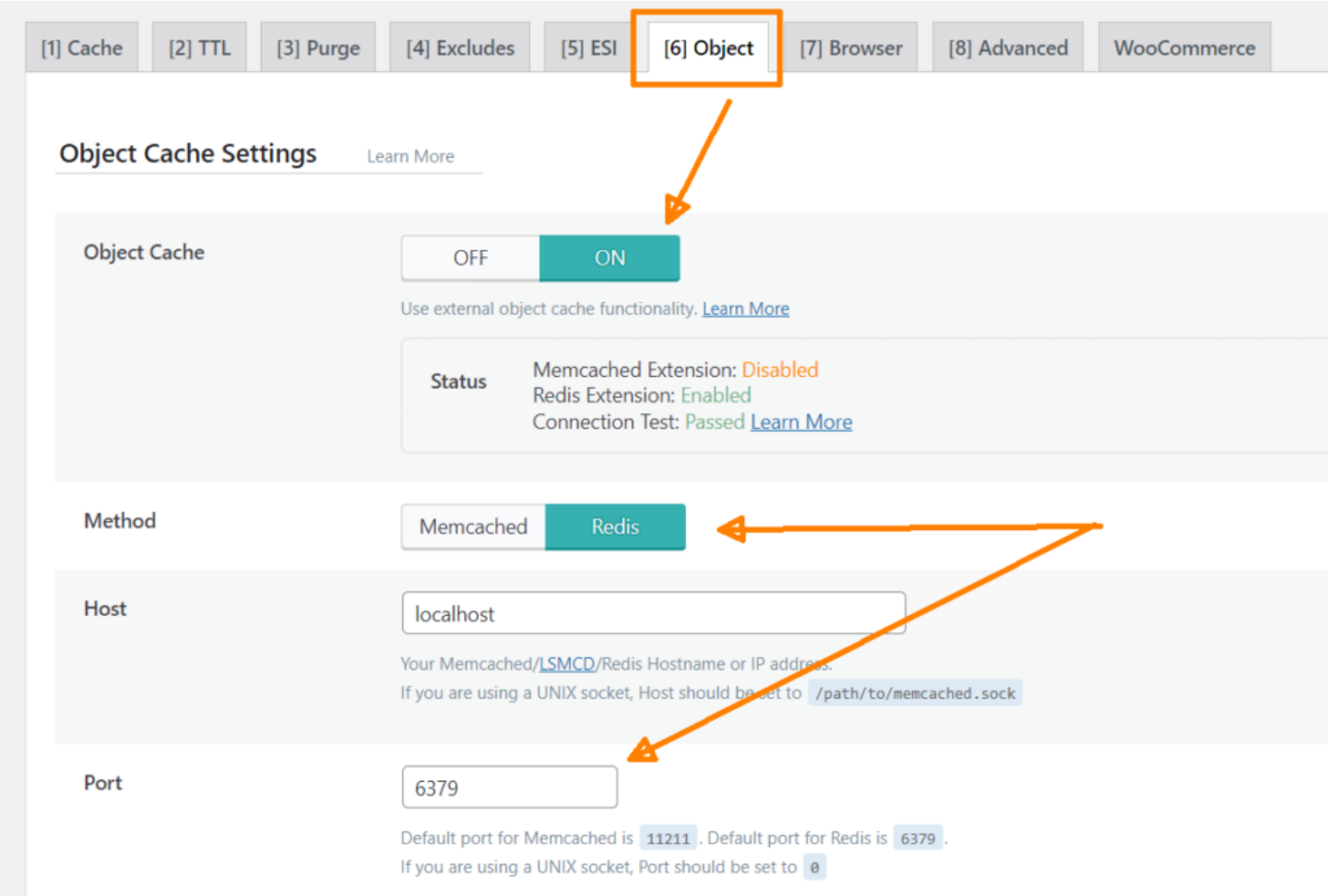
Enable Redis inside Wordpress

Most of our web servers run OpenLiteSpeed or LiteSpeed, and on these servers the preferred cache is lsCache. Enabling Redis in lsCache is pretty straightforward:

First, choose the Cache settings in lsCache



Then, choose the Cache tab (6), make sure all the settings are configured for Redis (you will need to enter the port correctly, but the rest are toggles), and hit save changes. Done.



If you are on an Apache server instead, you can install the **Redis Object Cache** plugin from the WordPress repository - just search for Redis and you'll find it. There's a Pro version available, but if you're reading this you don't need it - the tweaks it enables are useful for the largest WordPress sites but not for 99.99% of us.

(The Redis Object Cache plugin will work fine on LiteSpeed servers too, but since lsCache is already installed and it supports Redis there's no need for a separate plugin).

Opcode Caching

Opcode caching removes overhead that exists on every page of your WordPress site and makes your site much more efficient, meaning your pages require fewer resources so they load faster and you can handle more simultaneous users. Both are good.

What you need to understand is that WordPress is built in PHP, and humans can read PHP code, but computers programs (like the web server that serves your site) can't. So what happens is that PHP is

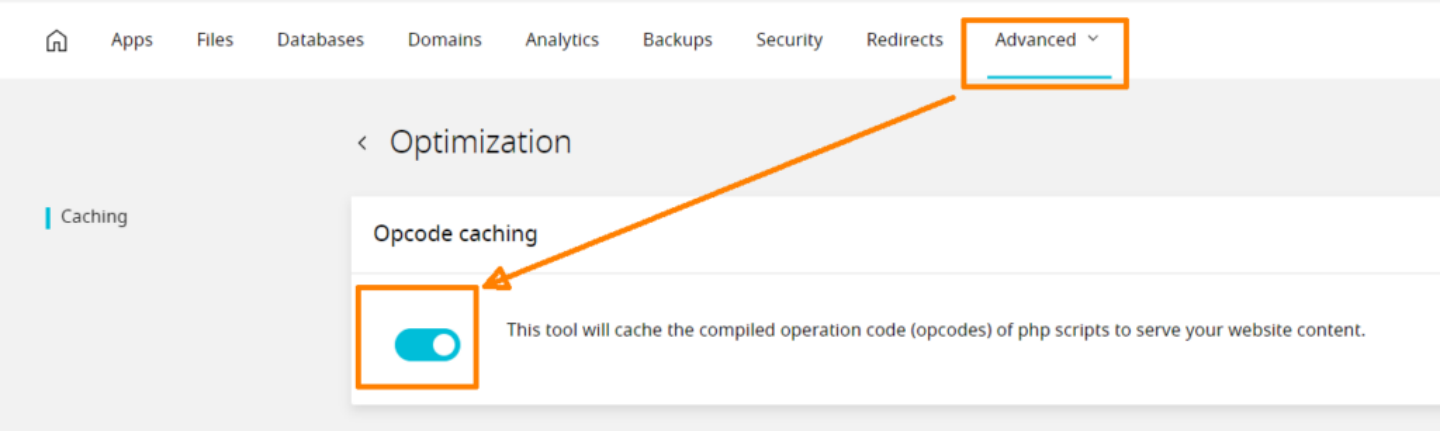
"compiled" into something that the web server can read, then that compiled code is run, then it's discarded when the page has been served.

And this happens *every time* a page is loaded. Wouldn't it make more sense to simply compile the code once, and run that until the source code changes and then compile it one more time? That's what an opcode cache does. Once enabled:

- Page load times can be 3x faster
- Response times can be reduced by 50%-70%
- Your site can handle 2x to 3x more requests per second

It's a free performance boost. Yeah, it costs a little memory, but it's worth turning it on.

In Enhance, that's simple: choose Advanced, Optimization in your control panel and turn on the toggle.



That's it. This works at the PHP level so you don't need to do anything in WordPress to see the improvements.