# **Best Practices for Hosting WordPress Site on Azure App Service**

## **Overview:**

One of the most frequently asked questions by Azure App Service customers is how to speed up the WordPress site hosted on App Service. Slow page load time can impact viewer experience and discourage them from visiting your website again. Many search engines consider page load time as well when ranking results. This can impact the visibility of your website in search engine results. First, we will look at the components in App Service that may contribute to the latency, then we will look at optimizations that can improve performance from within the WordPress App itself.

## **Azure App Service optimizations**

### **Choosing the right Azure WordPress Marketplace App Service**

There are several versions of WordPress available on the Azure App Service Marketplace, but we recommend WordPress on Linux from the Azure Marketplace as your starting point. This is optimized for App Service. Today, thousands of customers are running their WordPress sites on our Marketplace offerings.

While the Marketplace image comes with a fresh install of WordPress, customers can replace the code on the Web App and bring their own WordPress codebase. For cases in which additional PHP extensions are needed for plugins or themes, we recommend customers to customize the Marketplace Docker image.

### **Sizing Considerations**

It is important to anticipate the expected hardware requirements for your workload. The app service plan, and MySQL database need to be an appropriate SKU to support the workload. See the below link for information on how to assess the resource requirements for your app.

[https://azure.microsoft.com/en-us/migration/get-started/#assess](https://azure.microsoft.com/en-us/migration/get-started/)

### **Co-locate WordPress site and database**

Make sure your database and website are in the same Azure region. Network latency can increase the page load time for your website (esp. admin pages). Keeping the site and database components in the same region will reduce the network latency and improve the page load time for your website.

### **MySQL database solution VS in-app (Windows only)**

Compare the database options for your WordPress site. In general Azure Database for MySQL is recommended for optimal performance and ability to scale. For smaller/development workload scenarios the MySQL in-app may be an option.

* Azure Database for MySQL (default option). Azure MySQL is the managed service for MySQL server on Azure, it supports multiple SKU options to meet various performance, scale and price needs. Please choose the proper SKU base on your business needs.
* MySQL in-app (Windows only) provides a MySQL instance running on the same instance with web app and shares resources from the same App Service plan. Note that web apps using MySQL in-app are not intended for production purposes, and they will not scale beyond a single instance.

### **Always-on**

For low traffic web site, you can enable Always On to keep your web app always loaded, as by default, apps are unloaded if they are idle for some period of time.

### **Use Azure CDN**

If the slowness can be diagnosed as a result of static content taking a long time to load, then use [Azure CDN](https://azure.microsoft.com/en-us/services/cdn/).

### **Redis cache:**

[Azure Redis cache](https://azure.microsoft.com/en-us/services/cache/) can also be integrated with WordPress with the help of WP Redis plugin to get better performance.

## **WordPress Application optimizations**

### **Optimize your database**

WordPress auto-saves everything but the disadvantage with this is that your database gets filled with a lot of post revisions, trackbacks, pingback, comments and trash items quickly. You may hit the database size and I/O limit if the database is not cleaned up regularly. [WP-Optimize](http://wordpress.org/plugins/wp-optimize/) is a great plugin which allows you to clean up your database making it more efficient.

## **Caching**

If you start seeing a surge of user traffic hitting your site, you should consider adding some form of caching for your site to handle the spike of traffic.

* **WordPress on Windows**
  + [IIS output caching](https://www.iis.net/learn/manage/managing-performance-settings/walkthrough-iis-output-caching): It’s easy to configure and setup. It significantly improves site throughput but may prevent confirmation messages from showing up when submitting comments. To learn more about IIS output caching, click here.
  + [WP Super cache](https://wordpress.org/plugins/wp-super-cache/): It significantly improves site throughput and correctly handles comments submissions and other visitors’ actions. It slightly more complex to setup and configure than compared to IIS output caching. To learn more about Super cache configuration on IIS click here.
  + [App Service Dynamic Cache](https://github.com/projectkudu/kudu/wiki/Configurable-settings#turning-on-the-dynamic-cache-feature): turn on the Dynamic cache to improve performance.
  + [Wincache](https://www.php.net/manual/en/book.wincache.php): Create a .user.ini under wwwroot of your web application to add additional PHP settings. Enable wincache settings in a .user.ini [wincache.fcenabled](http://php.net/manual/en/wincache.configuration.php) = 1 and [wincache.reroute\_enabled](http://php.net/manual/en/wincache.configuration.php) = 1
* **WordPress on Linux**
  + [Redis Object Cache](https://wordpress.org/plugins/redis-cache/): the Redis cache plugin is installed by default, you can activate the plugin and connect it to database to improve the site performance.

### **Optimize Storage**

If your WordPress site is heavy with images, then images can take up most of bandwidth for your site. A couple of things you can do in this case: \* Store all your media content in an Azure Storage Blob: Use [Azure Storage Plugin](https://wordpress.org/plugins/windows-azure-storage/) which allows your website to store any new content to Azure storage instead of uploads directly of your WordPress site. If your site is media content heavy, it can significantly help to use an Azure CDN with the Azure Storage to reduce bandwidth.  
 \* For Linux site, you can also mount [additional storage accounts](https://blogs.msdn.microsoft.com/appserviceteam/2018/09/24/announcing-bring-your-own-storage-to-app-service/) to the container which runs your WordPress site, you can use the additional storage for content, image and media files. \*

### **Compress images**

[WP Smush.it](http://wordpress.org/plugins/wp-smushit/) is a great plugin that automatically compresses images as you upload them to the media library. There is no data loss during compression, and you won’t see any difference in the quality of images. If you have thousands of images are saved in your media library, you can run them all through the plugin, compressing them to a manageable size.

### **Reduce HTTP requests**

Every website makes multiple dependent requests either to JS, CSS files or third-party services or libraries. Putting all JavaScript into one JavaScript file and all CSS in one CSS file is considerably more efficient and reducing the dependent requests wherever possible can help. For CSS/JS scripts you can run a minify plugin like [Better WordPress Minify](https://wordpress.org/plugins/bwp-minify/) which will combine all your style sheets and JavaScript files into one hence reducing the number of requests that the browser needs to make. If your site uses third party libraries, it’s better to copy them locally them to make a call to the service that offers the library to reduce bandwidth.

### **Diagnose if your theme is slowing down your site**

Some themes just like plugins can be poorly written and can slow down your site. Test if your theme is responsible for the long page load times. To do this revert to the default WordPress template Twenty Fourteen and check if the page load time is much better. If yes, then the Theme could be the culprit. To resolve this, pick another theme. If there is no difference in the page load time after reverting the Theme to an older version, the try some of the tips listed in this blog.

### **Diagnose if any plugin is slowing down your site**

Some plugins can be the culprit in slowing down your site especially if they are poorly written or configured. Use a plugin like [P3](https://wordpress.org/plugins/p3-profiler/) to understand how much all the plugins used by the application do impact your page load time in order to take necessary actions to speed up your site If there are unused plugins, this can add to you page load time as WordPress tries to load all the activated plugins. Clean up or remove all the plugins that are not being used. Used only the plugins that can improve your website and not drag out the performance of your site. We have seen at least two plugins being slow on the App Service platform including Captcha and Visual composer. Find appropriate replacements of these plugins.

### **Turn off Pingbacks and Trackbacks if you don't use it**

WordPress uses Pingbacks and trackbacks methods to alert other blogs that your posts link to. \* A pingback is a type of comment that’s created when you link to another blog post where pingbacks are enabled. \* Trackbacks are a way to notify legacy blog systems that you’ve linked to them. They can be a drain the page speed and are usually better turned off if you don’t really need them. To learn more about Pingbacks and Trackbacks, [click here](https://make.wordpress.org/support/user-manual/building-your-wordpress-community/trackbacks-and-pingbacks/).

### **Specify image dimension**

Before the content is displayed to the end user, the browser had to identify the layout of the content around the images. Not knowing the size of the images, the browser has more work to do figure this out and take longer. It a best practice to specify image dimensions which saves the browser from having to go through this step, speeding things up.

### **Other Optimizations**

For a more exhaustive list of WordPress optimizations see documentation at [WordPress Optimization](https://codex.wordpress.org/WordPress_Optimization).

**Site Performance (Projected)**

We performed testing with P1V2 (Base Production App Service Plan) with 1vCore Azure Database for MySQL (Base MySQL instance). A max simultaneous user count of 170 was established. We used this to calculate a max monthly load below. You will need to adjust the numbers based on your business case.

|  |  |
| --- | --- |
| **Test Configuration** | **Value** |
| User Duration | 10 Minutes 6 sessions per hour (6 X 170) = 1,020 |
| Business Hours | 12 Hours per day (12 X 1020) = 12,240 |
| Peak Usage Adjustment | Assuming a margin of 3x over baseline for peak burst. 12,240 / 3 = 4,080 |
| Monthly Projection | (30 x 4,080) = 122,400 average monthly users |