

Made in the Shade – Kinsta Audit Report

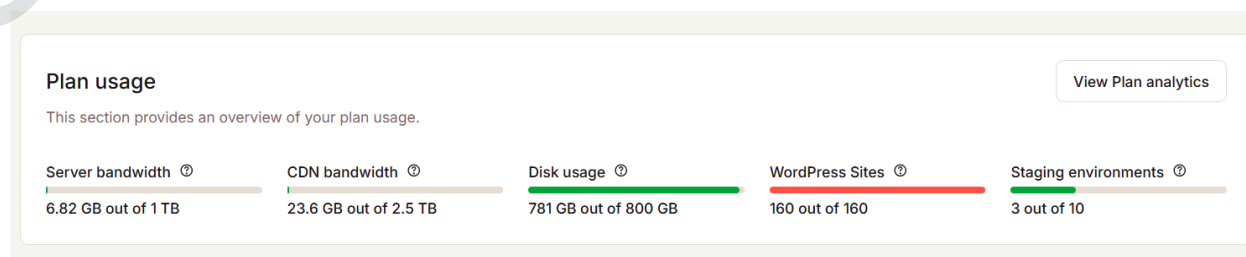
Objective:

This audit covers a thorough analysis of the Kinsta hosting environment, along with a performance comparison between **mitsaustin.com (Divi)** and **mitsabilene.com**. It outlines why the Kadence site performs significantly faster than Divi and outlines recommendations to improve the Divi site's loading speed and overall performance.

1. Kinsta Environment Review

- All Kinsta settings are properly configured and optimized for performance, except the cache expiration time.
- Caching, CDN, and Edge Caching are functioning as expected.
- Security configurations are solid and follow best practices.
- There are no error logs in the Kinsta dashboard, confirming that there are no PHP-related issues.
- The site's traffic patterns appear normal, with no unusual spikes or server-side bottlenecks, and most traffic is from the USA, which is expected behavior.
- Disk usage is currently **781 GB out of 800 GB**, and the **WordPress site limit (160/160)** has been fully reached. While the current setup remains stable, further expansion may be restricted.

It is recommended to **upgrade the hosting plan** or **review existing site allocations** to ensure smooth future operations.



2. Comparison Between MitsAustin (Divi) and MitsAbilene (Kadence)

These are the primary differences between Divi and Kadence site that explain their performance difference.

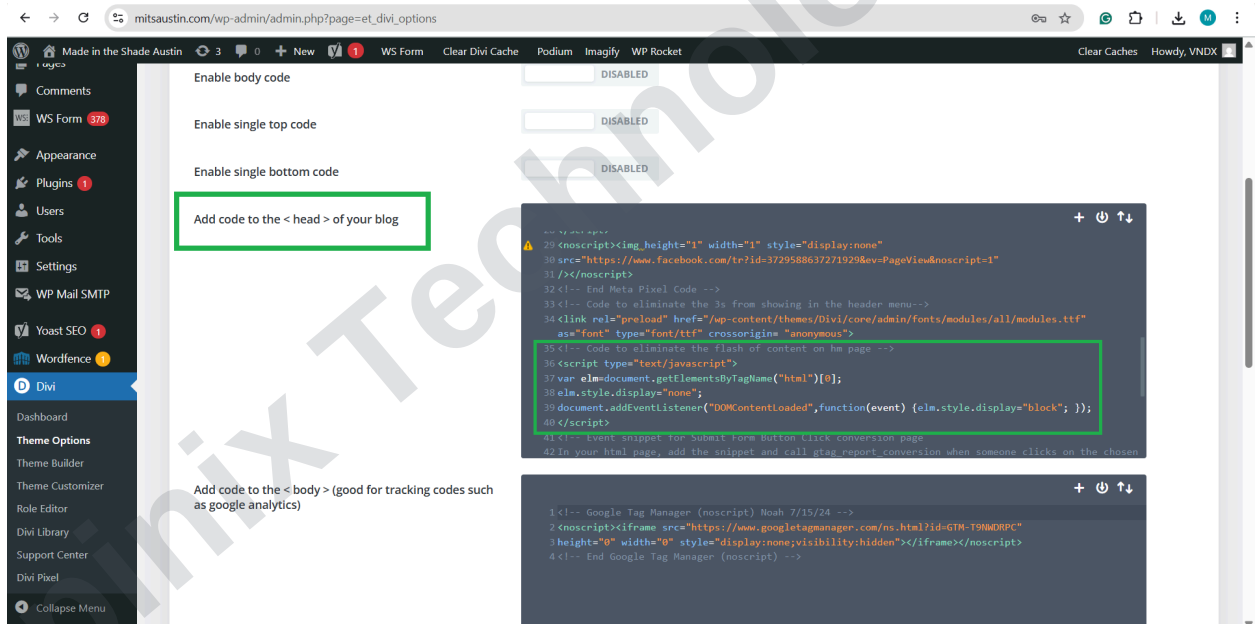
- **Hero Section Media** – The Divi site uses a background **video** in the hero section, which alone causes a **6.17 MB** network payload and significantly slows down the initial load. In contrast, the Kadence site loads a lightweight **image** of just **153 KB**, resulting in much faster rendering.
- **Google Tag Manager (GTM) Load** – On the Divi site, GTM is configured to load several third-party scripts such as **AdRoll**, **Google Ads**, **Facebook Pixel**, and other tracking tags. Each of these adds multiple network requests and increases the total page weight. The Kadence site uses far fewer GTM tags, keeping its load time much lower.
- **Font Requests** – The Divi site loads **13 font files**, while the Kadence site requests only **6**. Each additional font adds to the overall network activity and render-blocking resources.
- **Custom Header Code Impacting LCP** – The Divi site includes a custom JavaScript snippet in the header that hides the HTML until the DOM content is fully loaded. This increases the **Largest Contentful Paint (LCP)** time, making the initial render slower. The Kadence site also has this, and it should be removed from both sites.
- **Total HTTP Requests** – The Divi site makes around **162 total HTTP requests** (HTML, CSS, JS, video, fonts, etc), whereas the Kadence site makes only **73 requests**. The higher number of requests on Divi directly contributes to longer load times and lower performance scores.

3. Recommendations to Improve MitsAustin (Divi) Performance

- **Replace the Hero Video with an Optimized Image** – Use a high-quality but lightweight static image (like on the Kadence site) to reduce the hero section payload from 6 MB to under 200 KB.

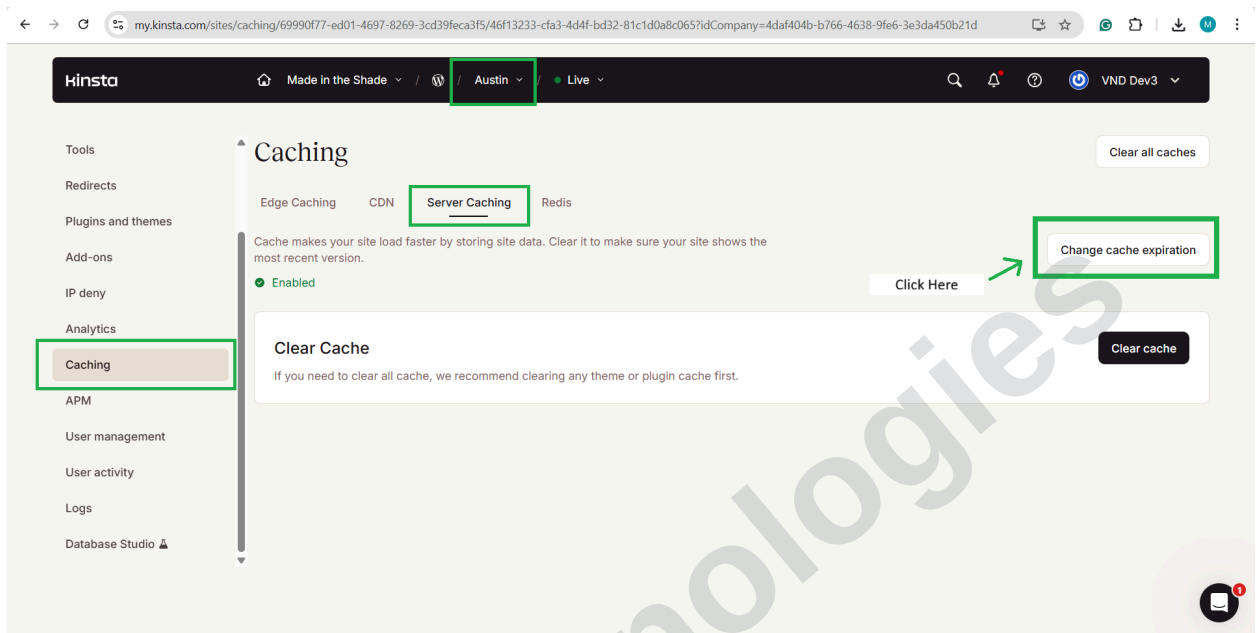
- **Optimize Google Tag Manager Setup** – Review and streamline GTM configuration to remove redundant or inactive tags. Reducing unnecessary tracking scripts will decrease the number of network requests and improve load time.
- **Remove Unnecessary Header Code** – Eliminate the following snippet from the site header to improve LCP and reduce render delay:

```
<script>
var elm=document.getElementsByTagName('html')[0];
elm.style.display='none';
document.addEventListener('DOMContentLoaded',function(){
elm.style.display='block'; });
</script>
```



- **Configure WP Rocket Properly** – Utilize features such as removing unused CSS, optimizing JS delivery, preloading key fonts, image lazy loading, and implementing preload settings, etc. We'll share a WP Rocket configuration file that you simply need to upload to each site, and WP Rocket will be automatically configured with the custom optimal settings.
- **Extend Server Cache Expiration in Kinsta** – Increase the cache expiry duration from 24 hours to 7 days, as the site's content is mostly static. This will

reduce redundant cache rebuilding and improve repeat load performance. **This step must be performed for each site individually.**



4. Conclusion

The Kinsta hosting environment is fully optimized and functioning as expected. The key performance issues on mitsaustin.com are front-end related—mainly the heavy hero video, excessive GTM scripts, high font requests, and inefficient front-end optimizations.

By implementing the above recommendations, mitsaustin.com can achieve load times comparable to—or even faster than—mitsabilene.com, while maintaining design quality, functionality, and user experience.