Mini-Guide to Web Performance

Version 1.0 — June 2025

Author: Alex Oliveira

Table of Contents

1. Overview & Objectives
2. Key Metrics
3. Front-End Loading Optimizations
4. Server & Network Optimizations
5. Third-Party Orchestration
6. Continuous Monitoring & Testing
7. Quick Checklist
8. Further Resources
9. Future Updates

1. Overview & Objectives

Why does performance matter? — it impacts user experience, SEO, conversion rates and infrastructure costs.
Scope — this guide covers front-end, back-end and continuous monitoring, always driven by real metrics.

2. Key Metrics (Core Web Vitals and more)

Metric	Recommended Threshold	What it measures
Largest Contentful Paint (LCP)	≤ 2.5 s	Loading speed of the main content
Interaction to Next Paint (INP)	≤ 200 ms	Real interaction latency (replaced FID in 2024)
Cumulative Layout Shift (CLS)	≤ 0.1	Visual stability
Time to First Byte (TTFB)	≤ 0.8 s	Server/network latency
First Contentful Paint (FCP)	≤ 1.8 s	First element rendered on screen

Tip: start the guide with this summary and quick links to PageSpeed Insights, Lighthouse, and WebPageTest.

3. Front-End Loading Optimizations

Critical resource order

• Inline critical CSS; use <link rel="preload"> / rel="prefetch" for fonts and above-the-fold assets.

Minification & compression

• Minify CSS/JS; enable Brotli or Gzip on the server.

JavaScript strategies

• type="module" + async/defer; split bundles (code-splitting) and leverage dynamic import().

Fonts

• font-display: swap; sub-set; avoid multiple variants; preconnect to font CDNs.

Images & media

• Modern formats (AVIF, WebP); srcset/sizes; lazy-loading (loading="lazy"); streaming via HLS/DASH.

HTTP/2 or HTTP/3

• Multiplexing, header compression and lower latency over HTTP/1.1.

4. Server & Network Optimizations (Back-End/CDN)

• Cache-Control & ETags — immutable policy vs. short cache.
• Geographically close CDN — lowers TTFB; use automated cache purging.
• Edge Functions/WAF — processing or validation closer to users.

• Database & API tuning — indexing, pagination, payload compression (JSON/GraphQL).
• Serve static assets efficiently — object storage + ‘origin shield’ on the CDN.

5. Third-Party Orchestration

• Audit external scripts (tag managers, chat, ads).
• Load them async/deferred; apply Subresource Integrity (SRI).
• Set a performance budget for external weight & request count.

6. Continuous Monitoring & Testing

• Lab vs. Field — Lighthouse local ≠ real RUM (e.g., Google CrUX, Elastic RUM).
• Alerts — set thresholds for LCP/INP/CLS in Grafana, Datadog, etc.
• CI/CD automation — reject builds below a Lighthouse score of 90 or exceeding bundle budgets.

7. Quick Checklist (one page)

☐ Critical CSS inline + font preloading
☐ AVIF/WebP images with lazy-loading
☐ Minified & compressed assets (Brotli/Gzip)
☐ font-display: swap & font sub-set
☐ JS async/defer & code-splitting
☐ Proper Cache-Control headers
☐ CDN configured and tested
☐ Third-party scripts audited
☐ Lighthouse CI integrated
☐ RUM monitoring in production

8. Further Resources

• web.dev/fast, MDN Web Performance docs, official Core Web Vitals criteria.
• Tools: Lighthouse, PageSpeed Insights, WebPageTest, Chrome DevTools, Sentry Performance.
• Communities & newsletters: PerfPlanet, Chrome DevRel, Calibre Perfweekly.

9. Future Updates

Review this guide every 6 months—metrics and best practices evolve (e.g., FID was replaced by INP in March 2024).

```
<link rel="preload" href="/fonts/Inter.woff2" as="font" type="font/woff2"
crossorigin>
```