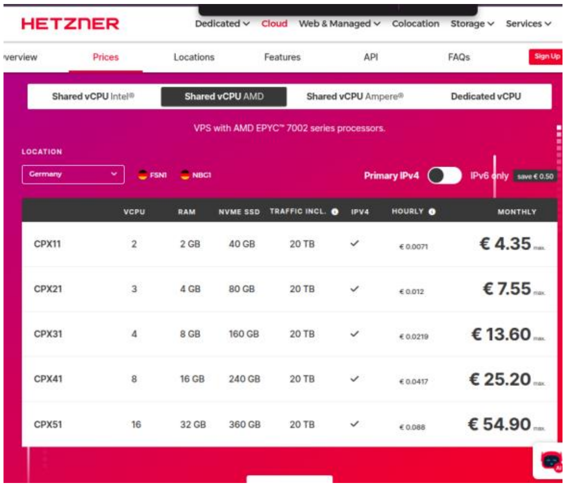
Tools used:

- MS WORD
- Quilt Bot Writing Extension
- Google Chromo
- CLoudflare , Hetzer, DigitalOccean Laravrl documentation pages

Question1: Hetzer or DigitalOcean?

I would choose **Hetzner Cloud** due to its strong price-to-performance ratio.

- **CPX21 (€7.55/month)** is ideal for hosting Laravel and Node.js services (3 vCPU, 4 GB RAM)
- **CPX11 (€4.35/month)** is enough for the WordPress + MariaDB stack via Docker Compose



VPS with AMD EPYC™ 7002 series processors.

LOCATION: Germany

Primary IPv4 ● IPv6 only

	VCPU	RAM	NVME SSD	TRAFFIC INCL.	IPV4	HOURLY	MONTHLY
CPX11	2	2 GB	40 GB	20 TB	✓	€ 0.0071	€ 4.35 max.
CPX21	3	4 GB	80 GB	20 TB	✓	€ 0.012	€ 7.55 max.
CPX31	4	8 GB	160 GB	20 TB	✓	€ 0.0219	€ 13.60 max.
CPX41	8	16 GB	240 GB	20 TB	✓	€ 0.0417	€ 25.20 max.
CPX51	16	32 GB	360 GB	20 TB	✓	€ 0.088	€ 54.90 max.

Architecture is **single-site**, not multisite, to keep each service isolated.
Laravel and Node.js are deployed via **PM2**; WordPress uses Docker.

Redundancy and SSL are handled by Cloudflare.
Each service is restarted independently, and backups can be done via volume or cron jobs.
Links: Hetzner Pricing , CLoudflare

Question2: WordPress on Apache2

Security: Block .env, wp-cnfig.php, and .htaccess; add security headers; hide Apache version

Performance: Enable gzip (**mod_deflate**) and caching (**mod_expires**) for static files

Backup: schedule cron or remote sync
More detail into:

- wordpress-conf/wordpress.conf
- docs/design.md

Question3: Laravel on Nginx + PHP-FPM:

Security: Block access to .env, .git, and storage/; add headers to prevent XSS and MIMES sniffing

Performance: Enable gzip, use try_files for routing, cache static assets with expires headers

Backup: Backup code via Git and database via mysqldump or scheduled cron.
More detail into:

- laravel-conf/laravel.conf
- docs/design.md

Question4: Node.js services:

I'd use PM2 for easy deployment, logging, and auto-restart. For bare-metal, I'd use a systemd unit to keep the service running after reboot. Both approaches support NODE_ENV=production and help with crash recovery.
More detail into:

- docs/design.md
- PM2: nodejs-conf/ecosystem.config.js

Question5: Cloudflare

I'd use Cloudflare for DNS, HTTPS (Full Strict), CDN, caching, and basic WAF protection.
For caching, I'd cache static files (CSS, JS, images) and exclude dynamic paths like /wp-admin or /login using page rules.

Question: CI/CD:

WordPress: No automated pipeline — updates handled via admin panel; backups scheduled with mysqldump and volume sync

Laravel: GitHub Actions pipeline runs on push; installs Composer packages, sets up .env, runs php artisan test, and performs a dummy deploy.

LINK

Node.js: Deployment managed manually using PM2 or systemd; future CI/CD can trigger a restart via SSH or webhook.