LARAVEL CHEAT SHEET

Nginx for Laravel

Public `public/`, TLS, PHP-FPM, a reverse proxy for WebSockets

Rules

- Nginx exposes **only** `public/`.
- App paths → `index.php`; hidden files and arbitrary PHP → `deny`.
- nginx ≥ 1.25.1: `http2 on;` (not `listen … http2`).
- Always `nginx -t` before `reload`.
- TLS higher up (LB/proxy) → configure trusted proxies in Laravel.

Minimal safe host

```
server {
    listen 443 ssl;
    http2 on;
    server_name example.com;
    root /var/www/current/public;      # public/ only
    index index.php;

    location / { try_files $uri $uri/ /index.php?$query_string; }

    location ~ \.php$ {
        try_files $uri =404;           # don't execute non-existent scripts
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;
        fastcgi_pass unix:/run/php/php8.5-fpm.sock;
    }

    location ~ /\.(?!well-known).* { deny all; }
}
```

`$realpath_root` resolves the release symlink before handing the script to PHP-FPM — essential for atomic deploys.

Port 80 → redirect only

```
server { listen 80; server_name example.com;
         return 301 https://$host$request_uri; }

server {
    listen 443 ssl;
    http2 on;
    server_name example.com;
    root /var/www/current/public;
    client_max_body_size 20m;
    add_header X-Content-Type-Options nosniff always;

    location / { try_files $uri $uri/ /index.php?$query_string; }
    location ~ \.php$ { try_files $uri =404; include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;
        fastcgi_pass unix:/run/php/php8.5-fpm.sock; }
    location ~ /\.(?!well-known).* { deny all; }
}
```

Keep TLS, redirects and FastCGI values in **reviewed** `include` files.

Reverse proxy (Reverb)

```
location /app/ {
    proxy_pass http://reverb:8080;
    proxy_http_version 1.1;                 # WebSocket needs HTTP/1.1
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host $host;
    proxy_read_timeout 60s;
}
```

Cache and rate limiting

```
limit_req_zone $binary_remote_addr zone=login:10m rate=10r/m;

location = /login {
    limit_req zone=login burst=10 nodelay;
    try_files $uri $uri/ /index.php?$query_string;
}
```

Cache aggressively **only** assets with a hash in the URL. Never logged-in HTML or authorization responses without an explicit key and invalidation.

`limit_req` on login does **not** replace authorization or webhook signature verification.

Verify

```
$ sudo nginx -t
$ sudo systemctl reload nginx
$ curl -I https://example.com/health
```

When NOT to use this

- Shared hosting or a platform that manages Nginx → its supported integration, not this config.
- Building Docker images, local caching and the deploy mechanism are separate topics — different failure modes and rollback.

RECAP

The sheet in short

- Serve `public/` only, `http2 on;`
- `try_files $uri =404` in the PHP block
- `$realpath_root` for atomic deploys
- Port 80 is just `return 301`
- WebSocket: `proxy_http_version 1.1` + upgrade
- Cache hashed assets only; `nginx -t` before reload

Found this useful?

Save this cheat sheet for later.

Full article: dominik-dev.pl

Share • Comment