URL Shortener

Production Architecture with Docker & Nginx

User Browser

Web Access Port: 80/443 HTTPS/HTTP



Nginx Reverse Proxy

Port: 80

- Static File Serving
- API Request Routing
 - Rate Limiting
 - SSL Termination
- Gzip Compression
- Security Headers



Frontend Container

Next.js + TypeScrip

- Static Assets
- React Components
 - Tailwind CSS
- Client-side Routing
- Redux State Management

Backend Container

Node.js + Expres

- REST API Endpoints
- URL Shortening Logic
- **OR Code Generation**
- Visitor Analytics
- Database Operations



MongoDB Container

Port: 27017

- urlshorteners collection
 - grcodes collection
 - visitors collection
 - Persistent Storage
 - Data Analytics

Request Flow

User → Nginx (Port 80) → Frontend (Static) / Backend (API) → MongoDB (Data)

Performance Optimization

Nginx serves static files 10x faster than Node.js, with 70% bandwidth reduction through gzip compression and intelligent caching strategies.

Security & Protection

Multi-layered security with rate limiting, security headers, container isolation, and proper access controls for production environments.

Horizontal Scalability

Each service can be scaled independently based on demand, with load balancing capabilities and container orchestration support.

Maintainable Architecture

Clear separation of concerns with microservices architecture, containerized deployment, and environment-based configuration management.

Single Port Access

 $\label{thm:correction} \mbox{Everything accessible via port 80, eliminating CORS issues, providing clean URLs, and simplifying deployment and SSL management.}$

DevOps Ready

Production-ready with health checks, comprehensive logging, monitoring capabilities, and CI/CD pipeline integration support.

Technology Stack

