

Manual Técnico - OPHION

1. Visão Geral

OPHION é uma plataforma open source de observabilidade que combina métricas, logs e traces com inteligência artificial para monitoramento proativo de infraestrutura.

2. Stack Tecnológico

| Camada | Tecnologia | Versão | |-----|-----|-----| | Backend | Go | 1.22+ | | Frontend | Next.js | 14.1.0 | | UI Components | Radix UI | Latest | | Gráficos | Recharts, Chart.js, D3 | Latest | | Estilização | TailwindCSS | Latest | | Observabilidade | OpenTelemetry | Latest | | IA | OpenAI API | GPT-4 | | Container | Docker Compose | Latest |

3. Estrutura do Projeto

```
`` ophion/ ├── cmd/ # Entry points | ├── server/main.go # API Server principal | └── agent/main.go # Agente de coleta ├── internal/ # Código interno | ├── ai/ # Módulos de IA | | ├── engine.go # Motor de IA | | ├── insights.go # Geração de insights | | ├── autohealing.go # Auto-recuperação | | ├── smart_alerts.go # Alertas inteligentes | | └── copilot.go # Assistente IA | ├── api/ # Handlers da API | | ├── ai_handlers.go # Endpoints de IA | | └── ratelimit.go # Rate limiting | ├── auth/ # Autenticação | | ├── handlers.go | | └── middleware.go | ├── otel/ # OpenTelemetry | | ├── otlp_receiver.go # Receptor OTLP | └── security/ # Segurança | └── security.go ├── dashboard/ # Frontend Next.js | ├── app/ # App Router | ├── components/ # Componentes React | └── package.json ├── deploy/ # Configurações de deploy | ├── docker/ | ├── remote-agent/ | ├── examples/ # Exemplos de instrumentação | ├── otel-nodejs/ | ├── otel-python/ | ├── configs/ # Configurações | ├── web/ # Assets web | ├── docker-compose.yml # Orquestração | └── instrument.sh # Script de instrumentação |
```

```
install.sh # Script de instalação |— go.mod / go.sum # Dependências Go |—  
README.md `|
```

4. Componentes do Sistema

4.1 API Server (Go)

- **Porta:** 8080 - **Função:** Backend principal, API REST, processamento de dados -
Módulos: Auth, AI, OTLP Receiver, Rate Limiting

4.2 Dashboard (Next.js)

- **Porta:** 3000 - **Função:** Interface web, visualização de métricas, configuração -
Tecnologias: React 18, TanStack Query, Recharts, D3

4.3 OTLP Receiver

- **Portas:** 4317 (gRPC), 4318 (HTTP) - **Função:** Recepção de traces, métricas e logs via
OpenTelemetry

4.4 Agente

- **Função:** Coleta de dados em hosts remotos - **Deploy:** Container ou binário standalone

5. Módulos de IA

5.1 AI Engine

- Motor central de processamento de IA - Integração com OpenAI GPT-4 - Cache de
respostas

5.2 Insights

- Análise automática de padrões - Detecção de anomalias - Sugestões de otimização

5.3 Auto-Healing

- Detecção de problemas - Execução automática de ações corretivas - Playbooks
configuráveis

5.4 Smart Alerts

- Correlação inteligente de alertas - Redução de ruído - Priorização automática

5.5 Copilot

- Assistente interativo via chat - Consultas em linguagem natural - Geração de queries e dashboards

6. API Endpoints

6.1 Autenticação

```
` POST /api/auth/login POST /api/auth/logout GET /api/auth/me `
```

6.2 Métricas

```
` GET /api/metrics GET /api/metrics/:name POST /api/metrics/query `
```

6.3 Traces

```
` GET /api/traces GET /api/traces/:traceId POST /api/traces/search `
```

6.4 Logs

```
` GET /api/logs POST /api/logs/query `
```

6.5 IA

```
` POST /api/ai/insights POST /api/ai/chat POST /api/ai/analyze POST  
/api/ai/autohealing/trigger `
```

6.6 Alertas

```
` GET /api/alerts POST /api/alerts PUT /api/alerts/:id DELETE /api/alerts/:id `
```

7. Requisitos de Sistema

7.1 Servidor

- CPU: 4 cores - RAM: 8GB - Disco: 100GB SSD - SO: Linux (Ubuntu 22.04+, Debian 12+)

7.2 Dependências

- Docker 24.0+ - Docker Compose 2.0+ - Go 1.22+ (para desenvolvimento) - Node.js 20+ (para desenvolvimento)

7.3 Rede

- Porta 3000: Dashboard - Porta 8080: API - Porta 4317: OTLP gRPC - Porta 4318: OTLP HTTP

8. Instalação e Deploy

8.1 Quick Start (Docker)

```
` bash git clone https://github.com/bigdux/ophion.git cd ophion docker compose up -d `
```

8.2 Instalação Completa

```
` bash
```

Download e instalação

```
curl -fsSL https://ophion.io/install.sh | bash
```

Configuração

```
cp .env.example .env vim .env
```

Iniciar

```
docker compose up -d `
```

8.3 Variáveis de Ambiente

```
` env
```

API

```
API_PORT=8080 API_SECRET=your-secret-key
```

OpenAI

```
OPENAI_API_KEY=sk-xxx
```

Database

```
DB_HOST=localhost DB_PORT=5432 DB_NAME=ophion
```

OTLP

```
OTLP_GRPC_PORT=4317 OTLP_HTTP_PORT=4318 `
```

9. Instrumentação de Aplicações

9.1 Script Universal

```
` bash
```

Auto-detecta linguagem

```
./instrument.sh my-container
```

Linguagem específica

```
./instrument.sh my-container nodejs ./instrument.sh my-container python  
./instrument.sh my-container java ./instrument.sh my-container dotnet ``
```

9.2 Linguagens Suportadas

- Node.js (auto-instrumentation) - Python (auto-instrumentation) - Java (agent) - .NET (auto-instrumentation) - Go (manual/SDK)

10. Segurança

10.1 Autenticação

- JWT tokens - Session management - Role-based access control

10.2 Comunicação

- HTTPS obrigatório em produção - TLS para OTLP - API rate limiting

10.3 Dados

- Encryption at rest (opcional) - Audit logging - Data retention policies

11. Monitoramento da Própria Plataforma

- Health checks em todos os componentes - Self-monitoring dashboard - Alertas de sistema --- *Documento gerado automaticamente em 2026-02-09*