

# Deployment

- [1 Deployment - FACIN\\_IA\\_D\\_P\\_N\\_1](#)
  - [1.1 🏠 Ambientes de Deployment](#)
    - [1.1.1 1. Development](#)
    - [1.1.2 2. Staging](#)
    - [1.1.3 3. Production](#)
  - [1.2 🚀 Pipeline de Deployment](#)
  - [1.3 🔒 Secrets & Configurations](#)
    - [1.3.1 GitHub Secrets \(Required\)](#)
    - [1.3.2 Environment Variables](#)
  - [1.4 🐳 Docker Deployment](#)
    - [1.4.1 Dockerfile Example](#)
    - [1.4.2 Build & Run](#)
  - [1.5 🌐 Deployment Platforms](#)
    - [1.5.1 Heroku](#)
    - [1.5.2 AWS/EC2](#)
    - [1.5.3 GCP/App Engine](#)
  - [1.6 🛡 CI/CD Configuration](#)
    - [1.6.1 GitHub Actions Workflow](#)
  - [1.7 📈 Monitoring Post-Deployment](#)
    - [1.7.1 AgenticOps Dashboard](#)
    - [1.7.2 Application Logs](#)
    - [1.7.3 Health Checks](#)
  - [1.8 🗃 Database Migrations](#)
    - [1.8.1 SQLite to PostgreSQL](#)
  - [1.9 📁 Pre-Deployment Checklist](#)
  - [1.10 💡 Rollback Procedures](#)
    - [1.10.1 Heroku Rollback](#)
    - [1.10.2 Docker Rollback](#)
    - [1.10.3 Database Rollback](#)
  - [1.11 📞 Support & Troubleshooting](#)
    - [1.11.1 Common Issues](#)
    - [1.11.2 Escalation](#)
  - [1.12 📄 Related Documentation](#)

## 1 Deployment - FACIN\_IA\_D\_P\_N\_1

Guia de deployment e publicação do projeto.

### 1.1 🏠 Ambientes de Deployment

#### 1.1.1 1. Development

- Local: `localhost:8501`
- Automático: push para branch `develop`
- Validação: Tipo “warning”

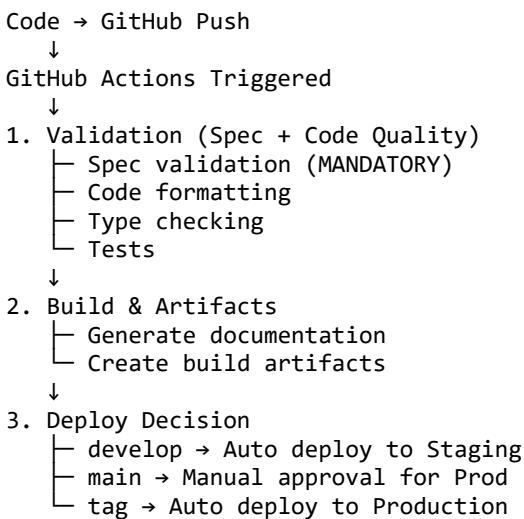
### 1.1.2 2. Staging

- URL: `staging-facin-ia.herokuapp.com` (exemplo)
- Automático: PR aprovada para `main`
- Validação: Tipo “error”, obrigatória

### 1.1.3 3. Production

- URL: `facin-ia.herokuapp.com` (exemplo)
- Manual: Tag de versão (v1.0.0)
- Validação: Todas as checks devem passar

## 1.2 🚀 Pipeline de Deployment



## 1.3 🔒 Secrets & Configurations

### 1.3.1 GitHub Secrets (Required)

```

OPENAI_API_KEY:          # OpenAI API key
GROQ_API_KEY:           # Groq API key
AGENTICOPS_API_KEY:     # AgenticOps key (optional)
HEROKU_API_KEY:          # Heroku deployment key
HEROKU_APP_NAME:         # Heroku app name
DATABASE_URL:            # Production database URL

```

### 1.3.2 Environment Variables

```

# .env.production
DEBUG=false
LOG_LEVEL=INFO
ENVIRONMENT=production
AGENTICOPS_ENABLED=true

```

## 1.4 🏰 Docker Deployment

### 1.4.1 Dockerfile Example

```
FROM python:3.12-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY . .
EXPOSE 8501
CMD ["streamlit", "run", "app.py", \
    "--server.port=8501", \
    "--server.address=0.0.0.0"]
```

### 1.4.2 Build & Run

```
# Build
docker build -t facin-ia:1.0.0 .

# Run Locally
docker run -p 8501:8501 \
    -e OPENAI_API_KEY=sk-... \
    -e GROQ_API_KEY=gsk-... \
    facin-ia:1.0.0

# Push to registry
docker push your-registry/facin-ia:1.0.0
```

## 1.5 Deployment Platforms

### 1.5.1 Heroku

```
# Login
heroku login

# Create app
heroku create facin-ia

# Set environment variables
heroku config:set OPENAI_API_KEY=sk-...
heroku config:set GROQ_API_KEY=gsk-...

# Deploy
git push heroku main

# View Logs
heroku logs --tail
```

### 1.5.2 AWS/EC2

```
# SSH into instance
ssh -i key.pem ec2-user@instance-ip

# Clone repository
git clone https://github.com/gutpassos/FACIN_IA.git
cd FACIN_IA
```

```

# Setup environment
python -m venv venv
source venv/bin/activate
pip install -r requirements.txt

# Configure supervisor or systemd
sudo cp facin-ia.service /etc/systemd/system/
sudo systemctl enable facin-ia
sudo systemctl start facin-ia

# Nginx reverse proxy
sudo apt install nginx
sudo cp nginx.conf /etc/nginx/sites-available/facin-ia

```

### 1.5.3 GCP/App Engine

```

# Initialize App Engine
gcloud app create

# Deploy
gcloud app deploy app.yaml

# View Logs
gcloud app logs read

```

---

## 1.6 CI/CD Configuration

### 1.6.1 GitHub Actions Workflow

O arquivo `.github/workflows/validate.yml` já está configurado com:

1. Validation Stage
    - Especificação JSON (OBRIGATÓRIO)
    - Code quality checks
    - Type checking
    - Tests with coverage
  2. Build Stage
    - Generate documentation
    - Create artifacts
    - Build Docker image
  3. Deploy Stage
    - Auto-deploy to staging on PR merge
    - Manual approval for production
    - Automated rollback on failure
- 

## 1.7 Monitoring Post-Deployment

### 1.7.1 AgenticOps Dashboard

```

# URL: https://agenticops.io/dashboard
# Project: FACIN_IA
# Monitora:
# - LLM calls
# - Agent performance

```

```
# - Error rates
# - Latency metrics
```

## 1.7.2 Application Logs

```
# Local logs
tail -f logs/app_YYYYMMDD.log
tail -f logs/error_YYYYMMDD.log
tail -f logs/agenticops.log

# Heroku logs
heroku logs --tail

# CloudWatch (AWS)
aws logs tail /aws/lambda/facin-ia --follow
```

## 1.7.3 Health Checks

```
# Basic health endpoint
curl http://localhost:8501

# Health check path (if implemented)
curl http://localhost:8501/health
```

## 1.8 Database Migrations

```
# Export SQLite
sqlite3 folha_pagamento.db ".dump" > backup.sql

# Create PostgreSQL database
createdb facin_ia_prod

# Import (may need adjustments)
psql facin_ia_prod < backup.sql

# Update config
export DATABASE_URL=postgresql://user:pass@host:5432/facin_ia_prod
```

## 1.9 Pre-Deployment Checklist

- All tests passing (70%+ coverage)
- Spec validation successful
- Code quality checks passing
- No security vulnerabilities detected
- Documentation updated
- CHANGELOG.md updated
- Version bumped (if releasing)
- Environment variables configured
- Database backups created
- Monitoring configured
- Rollback plan in place

## 1.10 🚨 Rollback Procedures

### 1.10.1 Heroku Rollback

```
# View releases
heroku releases

# Rollback to previous version
heroku rollback v42

# Deploy specific commit
git push heroku <commit-sha>:main
```

### 1.10.2 Docker Rollback

```
# Use previous image tag
docker run -d \
  --name facin-ia \
  -p 8501:8501 \
  your-registry/facin-ia:1.0.0-previous
```

### 1.10.3 Database Rollback

```
# Restore from backup
sqlite3 folha_pagamento.db < backup.sql

# PostgreSQL
psql facin_ia_prod < backup.sql
```

## 1.11 📞 Support & Troubleshooting

### 1.11.1 Common Issues

#### Port already in use

```
lsof -i :8501
kill -9 <PID>
```

#### Module not found

```
pip install -r requirements.txt
pip install -r requirements-dev.txt
```

#### Database connection error

```
# Check connection string
echo $DATABASE_URL

# Test connection
python -c "import sqlite3; sqlite3.connect('folha_pagamento.db')"
```

### 1.11.2 Escalation

1. Check logs: logs/error\_YYYYMMDD.log
2. Review recent changes: git log --oneline
3. Check status page: GitHub Status

4. Contact: gut.passos@gmail.com

---

## 1.12 Related Documentation

- [README.md](#) - Project overview
  - [QUICKSTART.md](#) - Getting started
  - [CONTRIBUTING.md](#) - Development guide
  - [ARCHITECTURE.md](#) - System design
- 

**Versão:** 1.0.0

**Data:** 27/02/2026

**Status:** Production Ready (Maturity Level 1)