DEPLOYMENT.md — Sentinel

Table of Contents

TL;DR

```
git clone <repository-url> sentinel && cd sentinel
npm install

cp .env.example .env  # Edit DATABASE_PATH and SSH credentials

node src/server.js
```

Navigate to http://localhost:3000

Prerequisites

Requirement	Version	Installation
-------------	---------	--------------
Node.js	18.x or higher	https://nodejs.org/en/download/
npm	8.x or higher	Comes with Node.js
SQLite3	3.x	Pre-installed on most systems
SSH Access	-	SSH keys for monitored hosts
Linux	Ubuntu 20.04+ recommended	For production deployment

System Requirements:

- 512MB RAM minimum (1GB+ recommended)
- 100MB disk space minimum
- Port 3000 available (or configure alternative)
- SSH access to monitored VMs/hosts

Step-by-Step Setup

1. Clone and Install

```
git clone <repository-url> sentinel
cd sentinel


npm install
```

2. Configure Environment

```
cp .env.example .env
nano .env  # or use your preferred editor
```

Required Configuration:

```
DATABASE_PATH=/path/to/sentinel.db
PORT=3000


NODE_ENV=production
```

3. Configure SSH Access

Create SSH credentials file:

```
mkdir -p config
nano config/hosts.json
```

Example `config/hosts.json`:

```
{
  "hosts": [
    {
      "id": "vm1",
      "name": "Claudinator",
      "host": "192.168.1.100",
      "port": 22,
      "username": "forge",
      "privateKeyPath": "/home/forge/.ssh/id_rsa"
    }
  ]
}
```

4. Initialize Database

The database auto-initializes on first run, but you can verify:

```
node -e "require('./src/services/storage').init()"
```

5. Start the Server

Development:

```
npm run dev
```

Production:

```
npm start
```

6. Verify Installation

```
curl http://localhost:3000/health
```

Expected: {"status":"ok","timestamp'

Open browser: `http://localhost:3000`

Environment Variables

Variable	Description	Example	Required
----------	-------------	---------	----------
`DATABASE_PATH`	SQLite database file location	`/var/lib/sentinel/sentinel.db`	✅ Yes
`PORT`	HTTP server port	`3000`	❌ No (default: 3000)
`NODE_ENV`	Environment mode	`production`	❌ No (default: development)
`LOG_LEVEL`	Logging verbosity	`info` , `debug` , `error`	❌ No (default: info)

`CRON_SCHEDULE`	Monitoring interval	`/5` *(every 5 min)*	❌ No (default: /5)
`SSH_TIMEOUT`	SSH connection timeout (ms)	`30000`	❌ No (default: 30000)
`MAX_CONCURRENT_CHECKS`	Parallel host checks	`5`	❌ No (default: 3)

Optional Configuration Files:

- `config/hosts.json` — Monitored host definitions (required for monitoring)
- `config/alerts.json` — Alert rule configuration (optional)
- `config/backup-targets.json` — Backup monitoring targets (optional)

Running in Production

Option 1: systemd Service (Recommended for Ubuntu)

Create service file:

```
sudo nano /etc/systemd/system/sentinel.service
```

```
[Unit]
Description=Sentinel Infrastructure Monitor

After=network.target


[Service]

Type=simple

User=forge

WorkingDirectory=/opt/sentinel

ExecStart=/usr/bin/node /opt/sentinel/src/server.js

Restart=always

RestartSec=10

StandardOutput=append:/var/log/sentinel/access.log
```

```
StandardError=append:/var/log/sentinel/error.log

Environment=NODE_ENV=production

Environment=DATABASE_PATH=/var/lib/sentinel/sentinel.db


[Install]

WantedBy=multi-user.target
```

Setup and start:

```
sudo mkdir -p /var/log/sentinel /var/lib/sentinel
sudo chown forge:forge /var/log/sentinel /var/lib/sentinel

sudo systemctl daemon-reload

sudo systemctl enable sentinel

sudo systemctl start sentinel

sudo systemctl status sentinel
```

View logs:

```
sudo journalctl -u sentinel -f
```

Option 2: PM2 (Alternative)

```
npm install -g pm2
pm2 start src/server.js --name sentinel

pm2 save

pm2 startup  # Follow instructions to enable on boot
```

PM2 Commands:

```
pm2 status
pm2 logs sentinel
```

```
pm2 restart sentinel

pm2 stop sentinel
```

Option 3: Screen/tmux (Quick & Dirty)

```
screen -S sentinel
npm start

Press Ctrl+A, then D to detach

screen -r sentinel  # To reattach
```

Docker

Dockerfile

```
FROM node:18-alpine

Install build dependencies for bett

RUN apk add --no-cache python3 make g++ sqlite

WORKDIR /app

Copy package files

COPY package*.json ./

Install dependencies
```

```
RUN npm ci --only=production
```

Copy application

```
COPY . .
```

Create volume mount points

```
VOLUME ["/app/data", "/app/config"]
```

Expose port

```
EXPOSE 3000
```

Set environment

```
ENV NODE_ENV=production

ENV DATABASE_PATH=/app/data/sentinel.db
```

Health check

```
HEALTHCHECK --interval=30s --timeout=10s --start-period=5s --retries=3 \

  CMD node -e "require('http').get('http://localhost:3000/health', (r) => process.e

CMD ["node", "src/server.js"]
```

docker-compose.yml

```
version: '3.8'


services:


  sentinel:
    build: .
    container_name: sentinel
    restart: unless-stopped
    ports:
      - "3000:3000"
    volumes:
      - ./data:/app/data
      - ./config:/app/config
      - ~/.ssh:/root/.ssh:ro  # SSH keys for host monitoring
    environment:
      - NODE_ENV=production
      - DATABASE_PATH=/app/data/sentinel.db
      - PORT=3000
    networks:
      - sentinel-net


networks:


  sentinel-net:
    driver: bridge
```

Build and Run

```
docker build -t sentinel:latest .

docker run -d \


  --name sentinel \
  -p 3000:3000 \
  -v $(pwd)/data:/app/data \
  -v $(pwd)/config:/app/config \
  -v ~/.ssh:/root/.ssh:ro \
  sentinel:latest
```

With docker-compose:

```
docker-compose up -d

docker-compose logs -f sentinel



docker-compose restart sentinel
```

Troubleshooting

Port Already in Use

Error: `EADDRINUSE: address already in use :::3000`

```
# Find what's using port 3000

sudo lsof -i :3000
```

Or

```
sudo netstat -tlnp | grep 3000
```

Kill the process

```
sudo kill -9 <PID>
```

Or use different port

```
PORT=3001 npm start
```

Database Locked Error

Error: `SQLITE_BUSY: database is locked`

```
# Check for zombie processes
ps aux | grep node
```

Kill old instances

```
pkill -f "node.*sentinel"
```

Restart cleanly

```
npm start
```

SSH Connection Failures

Error: `Error: All configured authentication methods failed` **Fixes:**

1. Verify SSH key permissions:

```
chmod 600 ~/.ssh/id_rsa
chmod 700 ~/.ssh
```

2. Test SSH manually:

```
ssh -i ~/.ssh/id_rsa forge@192.168.1.100
```

3. Check `config/hosts.json` paths are absolute
4. Ensure user has SSH access to target hosts

Missing Dependencies

Error: `Cannot find module 'better-sqlite3'`

```
# Rebuild native modules
npm rebuild better-sqlite3

Or clean install

rm -rf node_modules package-lock.json

npm install
```

Permission Denied on Database

Error: `SQLITE_CANTOPEN: unable to open database file`

```
# Check database directory permissions
ls -la /var/lib/sentinel/

Fix ownership

sudo chown -R forge:forge /var/lib/sentinel

Ensure directory exists
```

```
mkdir -p $(dirname $DATABASE_PATH)
```

High Memory Usage

Symptoms: Node process consuming excessive memory **Fixes:**

1. Limit concurrent checks in `.env`:

```
MAX_CONCURRENT_CHECKS=2
```

2. Increase check interval:

```
CRON_SCHEDULE=/10    *  # Every 10 minutes instead of 5
```

3. Set Node memory limit:

```
NODE_OPTIONS="--max-old-space-size=512" npm start
```

Dashboard Not Loading

Checks:

1. Server is running:

```
curl http://localhost:3000/health
```

2. Check logs:

```
# systemd
sudo journalctl -u sentinel -n 50

PM2

pm2 logs sentinel

Direct

tail -f /var/log/sentinel/error.log
```

3. Verify port accessibility:

```
# From remote machine
curl http://<server-ip>:3000/health
```

Check firewall

```
sudo ufw status

sudo ufw allow 3000/tcp
```

Node Version Issues

Error: `SyntaxError: Unexpected token '?.'` or similar

```
# Check Node version
node --version
```

Must be 18.x or higher

Install correct version:

```
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -

sudo apt-get install -y nodejs
```

Cron Jobs Not Running

Symptom: Monitoring data not updating **Debug:**

```
# Check cron service is running
```

Add debug logging to .env:

```
LOG_LEVEL=debug
```

Restart and watch logs

```
npm start
```

Verify cron schedule:

```
// Test in Node console
node -e "console.log(require('node-cron').validate('/5    *'))"

// Should print: true
```

Need Help?

- Check logs first: `sudo journalctl -u sentinel -n 100`
- Verify prerequisites are met
- Test SSH connections manually
- Review `config/hosts.json` syntax
- Ensure database path is writable

Related Documentation

- Readme
- Architecture