

repl-sentinel 安装部署文档

面向部署运维人员的一步一步操作手册（运维实战导向）。事实源为 09-■■■■■■■■■■，已改写为可照做流程。
产品版本口径：VERSION = 0.721.3。

1. 交付物说明

开发者侧产出 releases/repl-sentinel-v<■■■>.tar.gz，解压后结构：

```
repl-sentinel-v<版本>/
├── repl-sentinel-image-v<版本>-linux-amd64.tar  后端镜像（API，静态二进制）
├── repl-sentinel-web-image-v<版本>-linux-amd64.tar 前端镜像（Nginx + 静态站点 + 反代）
├── start.sh                                启动脚本（幂等）
└── readme.txt                             部署说明
```

同级附 repl-sentinel-v<■■■>.tar.gz.sha256 与 repl-sentinel-verify.sh（校验并解压）。

运行时两个容器：

- repl-sentinel-backend：容器内：6081，不对宿主机发布端口（仅容器网络内可达）。
- repl-sentinel-web：Nginx，对外 PORT（容器内 80），反代 /api、/healthz、/readyz 到别名 backend。

start.sh 幂等：重复执行只替换同名容器，./data 保留，升级/重启配置与管理账号不丢。

2. 环境要求

- 目标机已安装 Docker（无需任何配置文件；无需在被监控库侧装 agent）。
- 操作系统：Linux（Alpine 基础镜像路线；宿主机不限，能跑 Docker 即可）。
- 端口约定：前端对外默认 6080（可经 PORT 改），后端容器内 6081 不外露；新增默认端口须落在 6000-6100 且避开已用值。
- 被监控 MySQL：8.0.22+（SHOW REPLICA STATUS）；5.7/8.0.22- 自动回退 SHOW SLAVE STATUS；GTID 需 5.6+。
- 资源参考：单实例建议 cpus：'0.2'、memory：64M。

3. 快速部署

```
./repl-sentinel-verify.sh # 校验 SHA-256，通过自动解压（失败不解压、退出 1）
cd repl-sentinel-v<版本>
./start.sh                # 可选 PORT=6090 / DATA_DIR=/srv/rs-data / NETWORK=rs-net
# 浏览器打开 http://<本机IP>:PORT/
# 首次访问：创建管理员账号并登录
# 单页内：选架构 → 填各节点连接（首次须测试连接全通过）→ 采集与告警设置 → 保存并开始监控
```

start.sh 行为：docker load 两镜像 → 创建网络 → 启后端容器（别名 backend，挂 DATA_DIR:/data）→ 启前端容器（映射 PORT:80）→ 打印引导。改完配置无需重启容器。

4. 运行参数与运行时变量

容器对外端口：PORT（容器内 80，前端映射）。

运行时变量（非业务，经 .env + compose / dev-start 注入；无旧环境变量兼容导入）：

变量	默认值	说明
LISTEN_ADDR	:6081	后端监听地址
DATA_DIR	/data	持久化目录（容器内）
TZ	Asia/Shanghai	时区
WEB_PORT	:6080	仅 dev-start 使用，前端本地端口

配置入口唯一：业务配置全部在 Web 页面完成；运行时变量仅上述几项，严禁旧环境变量兼容导入。

5. 构建（开发者侧）

```
cd scripts
./build-web.sh    # 前端产物 → frontend/dist
./build.sh release # 先 test.sh，再 build-web.sh + docker-build.sh，打包到 ../releases/
```

等价单步（上下文为项目根）：

- 后端：docker build -f deploy/Dockerfile -t repl-sentinel:latest .（基础 alpine:3.20，含 ca-certificates、tzdata，声明 VOLUME /data）。
- 前端：docker build -f deploy/frontend.Dockerfile -t repl-sentinel-web:latest .（构建阶段 node:18-alpine；运行 nginx:1.27-alpine）。

铁则：任何代码修改必须 ./test.sh 通过；涉及页面/接口行为的改动必须 scripts/dev-start.sh 测试环境验证通过后，才可 ./build.sh release 打包交付；改 UI 的改动须目视/无头实测。

6. compose 部署（可选，非交付形态）

运行时变量经 deploy/.env（模板 .env.example）注入：

```
services:
  backend:
    image: repl-sentinel:latest
    container_name: repl-sentinel-backend
    restart: unless-stopped
    environment: { TZ: ${TZ:-Asia/Shanghai}, DATA_DIR: /data, LISTEN_ADDR: :6081 }
    volumes: [ "${HOST_DATA_DIR:-../data}:/data" ]
    expose: [ "6081" ]          # 仅容器网络内可达
  web:
    image: repl-sentinel-web:latest
    container_name: repl-sentinel-web
    restart: unless-stopped
    depends_on: [backend]
    ports: [ "${PORT:-6080}:80" ] # 唯一对外端口
```

7. 测试环境（dev-start，不构建镜像）

直接以两个本地进程运行，标准化启停：

```
cd scripts
./dev-start.sh start # 构建后端 + 启后端 :6081 与前端 :6080，等待就绪
./dev-start.sh status # 两进程/端口状态 + 登录账号 + 是否已配置
./dev-start.sh logs # 跟踪后端日志（前端日志见 data/log/web.log）
./dev-start.sh restart # 重启（配置留 data/dev/，日志 data/log/{dev,web}.log）
./dev-start.sh stop # 按 PID 停止（不影响其他进程）
```

`./dev-start.sh reset` # 停并清空测试配置（回「尚未配置」，含 `auth.json`）

流程约定：测试环境通过后，再执行 `scripts/build.sh release` 打包交付。

8. 监控账号创建（生产库侧，用户手动执行）

主从不同名，防 1396：

-- 从库（每从库一份）

```
CREATE USER 'monitor_slave'@'%' IDENTIFIED BY '...';  
GRANT REPLICATION CLIENT ON *.* TO 'monitor_slave'@'%';
```

-- 主库（额外需 PROCESS）

```
CREATE USER 'monitor_master'@'%' IDENTIFIED BY '...';  
GRANT REPLICATION CLIENT, PROCESS ON *.* TO 'monitor_master'@'%';
```

- 不需要 SELECT、不需要任何写权限。
 - 主库与从库监控账号必须不同名。
 - 详细见 `docs/monitor-account.md`。
-

9. 升级 / 重置 / 重建账号 / 卸载

- 升级：重新 `./start.sh`（幂等替换容器，`./data` 保留）。
 - 重置配置：页面集群卡片「重置」（两段式，仅清空单集群连接信息重配）；`POST /api/config/reset` API（删 `config.json`，`auth.json` 保留）供脚本/运维，页面无入口。
 - 重建账号：删 `auth.json` 后首访重建（旧会话失效）。
 - 卸载：`docker rm -f repl-sentinel-backend repl-sentinel-web`；数据目录 `./data` 按需保留或删除。
 - 资源限制建议：`cpus: '0.2'`、`memory: 64M`。
-

10. 持久化与磁盘格式

- 持久化文件（均在 `$DATA_DIR`，0600）：`config.json`（v5，含密码）、`auth.json`（salt+hash）、`sessions.json`（记住登录会话，无口令）。原子写入（临时文件 + rename）。备份/迁移只需拷贝该目录；`docker rm` 容器不影响配置。
 - 磁盘格式版本：v4 → v5 仅迁移全局 relay 两阈值默认值（v4 及更早的 0 视作未启用改写为 64MB/1GB；集群级显式 0 不改写）；v5 对 v0.700.0 及更早二进制不可读（回退前先备份 `config.json`，v5 与 v4 字段集相同，改回 `"version": 4` 即可被旧版读）。
-

11. 排查

- 存活探针：`GET /healthz` 免登录，未配置也 200。
- 就绪探针：`GET /readyz` 全节点（含主库）健康才 200，否则 503；容器编排用其做 readiness。
- 状态接口：`GET /api/status`（需登录）返回拓扑、`nodes[]`（含 replica/source 与 `checks[]`）；未配置返回 `configured: false`。
- 日志：
 - 容器部署：后端日志走 stdout（符合容器规范）；前端日志见 `data/log/web.log`。
 - `dev-start`：`data/log/{dev,web}.log`，`./dev-start.sh logs` 跟踪后端。
- 常见问题：

- 页面打不开：确认 PORT 映射与防火墙；确认 repl-sentinel-web 容器运行。
- 总览显示「尚未配置」：首次保存未成功，回到集群配置页确认全部节点测试连接通过。
- 邮件收不到：在告警邮件页「发送测试邮件」验证；检查内置/外置 SMTP 配置与收件人是否为空。

反馈与修订统一回流至 docs/prd/；本手册不与 PRD 冲突时以 PRD + docs/architecture.md 现行章节为准。

以上是 EVE 的回答