数智天枢

安装部署与运维手册

版本：V1.1.0

日期：2026-11-13

西安国能数智科技有限公司

www.ai-energy.cn

西安国能数智科技有限公司

目　录

1 交付边界 …… 3

2 服务器要求 …… 3

3 架构与端口 …… 3

4 首次安装 …… 4

4.1 交付物 …… 4

4.2 安装步骤 …… 4

4.3 配置说明 …… 5

5 授权许可 …… 5

5.1 生成请求 …… 5

5.2 授权方签发 …… 5

5.3 安装证书 …… 6

6 日常运维 …… 6

6.1 上云 API 2.0 SDK 专用日志 …… 6

7 备份与恢复 …… 7

8 升级与回滚 …… 7

9 常见故障 …… 8

10 验收清单 …… 8

西安国能数智科技有限公司

> 适用环境：Ubuntu Server 22.04 LTS / x86_64，单机 Docker Compose 交付形态：离线签名发布、机器绑定授权 更新日期：2026-11-23

本文是客户侧安装、授权、升级和运维的操作依据。所有命令在目标服务器上以 `sudo` 执行， 统一入口为 `/opt/dehub/bin/dehubctl` 。

1 交付边界

- 支持：Ubuntu Server 22.04 LTS x86_64 单机 Docker Compose；纯内网或公网； 数智天枢与上云 API 2.0 底座只要求网络互通，不要求同机部署。
- 不支持：ARM、Kubernetes、Docker Swarm、高可用、多节点数据库。
- 默认提供 HTTP。公网部署由客户自行配置 HTTPS、VPN/零信任和访问控制。
- 责任划分：客户负责主机、网络、账号、日常维护和备份保管；授权方只负责签发 License （含续期、降级和换机 `REBIND` ）。客户不能自行签发或绕过机器绑定。

2 服务器要求

配置项	要求	说明
CPU / 内存	8 Core / 64 GB	约支持 20 台机场同时在线
数据盘	2000 GB SSD	必须使用 SSD，建议独立挂载 `/data`
网络	不低于 100 Mbps	每台机场两路直播约 5 Mbps，按比例扩容
操作系统	Ubuntu Server 22.04 LTS x86_64	时间同步（NTP）必须正常
Docker / Compose	24.0.7 / 2.21.0	使用 `docker compose` v2

约 50 台机场建议 16 Core / 64 GB，并扩容 SSD 与带宽。数智天枢与上云 API 2.0 底座同机时， 不能把上述规格视为两套系统共享后的充足规格，必须重新做容量评估。

3 架构与端口

```
浏览器/外部系统 -> Web :19080 -> API :19081 -> PostGIS :15432
                                          |
                                          +-> 上云 API 2.0 底座 :30812 / :8152 / :30802
```

西安国能数智科技有限公司

端口	用途
19080/TCP（可改）	数智天枢页面和 `/api`，对外开放
19081/TCP	API 维护端口，默认只绑定 `127.0.0.1`
15432/TCP	PostgreSQL，默认只绑定 `127.0.0.1`
30812/TCP	上云 API 2.0 底座 OpenAPI / SDK / WebSocket
8152/TCP	上云 API 2.0 底座 PaaS 与设备管理资源
30802/TCP	上云 API 2.0 底座 MinIO（航线和文件）

常驻容器严格为 `dehub-web`、`dehub-api`、`postgis`。Web/API 使用宿主机网络， 安装前确认上述端口未被占用。

4 首次安装

4.1 交付物

首次部署只接收一个通过受控渠道取得的安装包，不需要执行独立验签：

```
dehub-offline-installer_VERSION_amd64.deb
```

包内自带后续升级的信任根，安装时自动校验内容并把发布公钥写入 `/opt/dehub/trust`。 该包只能用于首次安装，检测到已有版本时会拒绝覆盖，后续升级必须使用升级包。

4.2 安装步骤

```
# 1. 安装
sudo apt install ./dehub-offline-installer_VERSION_amd64.deb

# 2. 生成配置模板并填写
sudo /opt/dehub/bin/dehubctl init-config
sudo vi /opt/dehub/config/dehub.env

# 3. 预检
sudo /opt/dehub/bin/dehubctl precheck

# 4. 安装并启动（首次必须带 --skip-license-check）
sudo /opt/dehub/bin/dehubctl install --skip-license-check

# 5. 初始化管理员
sudo install -m 0600 -o root -g root /dev/null /run/dehub-admin-password
sudo vi /run/dehub-admin-password        # 12-64 位，含大写、小写、数字和特殊字符
sudo test -s /run/dehub-admin-password
sudo /opt/dehub/bin/dehubctl admin-init admin /run/dehub-admin-password
sudo rm -f /run/dehub-admin-password
```

- 干净主机上没有 License 文件，`install` 必须带 `--skip-license-check`，否则返回 `LICENSE_MISSING`（退出码 21）。此时业务接口全部拒绝访问，只保留登录、授权、健康检查和诊断入口。

- 管理员已存在时 `admin-init` 幂等返回，不覆盖密码。

4.3 配置说明

只编辑 `/opt/dehub/config/dehub.env`。`<...>` 是必须替换的占位符，未替换时 `precheck` 和 `install` 返回 `CONFIG_PLACEHOLDER_UNFILLED`（退出码 11），不会带着占位符上线。数据库口令、 加密密钥、内部端口由工具写入 `/opt/dehub/state/runtime.env`，该文件禁止手工修改。

键	说明
`DEHUB_PUBLIC_BASE_URL`	数智天枢对外访问地址，浏览器、WebSocket 和系统回调都以此为准
`DEPLOY_BIND_IP`	Web 监听网卡，`0.0.0.0` 允许外部访问，`127.0.0.1` 仅本机
`KOC3_HOST`	上云 API 2.0 底座域名或 IP，只填 host，不带协议、端口和路径
`DEHUB_ACCESS_TOKEN`	统一访问令牌（HTTP `accesstoken` / WS `accessToken`），必填、人工填写、不自动生成。升级默认沿用现有系统令牌；未配置底座 webhook、由底座自行校验令牌内容的环境改取服务令牌
`DEHUB_LINK_VERIFY_MINIO_ACCESS_KEY_ID` / `_SECRET`	底座 MinIO 共用账号和密码（`dehub.object-storage`），用于航线上传、飞行回放、告警证据和 Terra 预览

配置必须为 `0600 root:root`，每行 `KEY=value`，不能有未知或重复的键。改完配置用 `dehubctl restart` 重建容器生效，不要用 `docker compose restart`，它不会重新读取配置。

5 授权许可

流程固定三步：**客户生成请求** → **授权方签发证书** → **客户安装证书**。Web 与 CLI 两条通道等价， 同一台机器的请求和证书互通。

5.1 生成请求

浏览器打开 `http://<DEHUB_IP>:19080`，用管理员账号登录 →「授权状态」页 →「生成请求」→ 下载 `license-request.json` 交授权方。安装时已自动生成过请求，页面可直接下载或刷新。

```
# CLI 等价命令
sudo /opt/dehub/bin/dehubctl license request
```

请求含机器指纹， `machine-binding.json` 随之落盘，请勿手工修改。

5.2 授权方签发

授权方持 Ed25519 私钥离线签发 `license.lic`。首次签发为 `INITIAL`，续期 `RENEWAL`， 降级 `DOWNGRADE`，换机 `REBIND`，同一证书序列的 `licenseRevision` 必须递增。 私钥不进入任何交付物，客户不能自行签发。

5.3 安装证书

「授权状态」页 →「导入证书」→ 上传 `license.lic`。校验通过后状态变为可用，业务页面恢复访问。

```
sudo /opt/dehub/bin/dehubctl license validate /path/to/license.lic
sudo /opt/dehub/bin/dehubctl license install /path/to/license.lic
sudo /opt/dehub/bin/dehubctl license status
```

续期或升级授权时服务已在运行，**不需要** `--skip-license-check`，重复上述三步即可。

安装失败会返回稳定的 `reasonCode`，常见的有：签名校验失败 `LICENSE_SIGNATURE_INVALID`、 机器指纹不符 `LICENSE_MACHINE_MISMATCH`（换机须走 `REBIND`）、版本不在授权范围 `LICENSE_VERSION_MISMATCH`、revision 回滚 `LICENSE_REVISION_ROLLBACK`。 完整对照见包内 `contracts/ops/reason-codes.json`。

机器绑定基于目标机系统 UUID 与 `/etc/machine-id`。签发后不要修改或替换这些标识； 更换主板、虚拟机克隆、裸机恢复都必须重新申请 `REBIND` 证书。

6 日常运维

```
sudo /opt/dehub/bin/dehubctl help                       # 全部命令
sudo /opt/dehub/bin/dehubctl start | stop | restart     # restart 重建 Web/API 并重读配置
sudo /opt/dehub/bin/dehubctl restart --all              # 含 PostGIS
sudo /opt/dehub/bin/dehubctl status
sudo /opt/dehub/bin/dehubctl logs dehub-api --tail 200
sudo /opt/dehub/bin/dehubctl verify | smoke | diagnose
```

变更配置的固定顺序：备份配置 → 编辑 → 保持 `0600 root:root` → `precheck` → `restart`。若 `status` 返回 `RECOVERY_REQUIRED`，按输出 JSON 处理：给出 `nextCommand` 时只执行该命令； 输出只有 `writeGate` 和 `message`（表示无法确认业务写入已冻结）时不要执行 `recover`， 保持服务与数据现状并联系交付方。两种情况都不要删除 journal 或并发启动其他维护命令。

6.1 上云 API 2.0 SDK 专用日志

SDK 原生日志单独写入宿主机数据盘，不包含在 `dehubctl logs dehub-api` 的容器标准输出日志中。默认路径如下；如果安装配置中的 `DATA_ROOT` 已修改，以实际值为准：

```
${DATA_ROOT}/logs/sdk/flight_hub_sdk.log*
默认：/data/dehub/logs/sdk/flight_hub_sdk.log*
```

实时查看当前日志或列出轮转文件：

```
sudo ls -lh /data/dehub/logs/sdk/flight_hub_sdk.log*
sudo tail -f /data/dehub/logs/sdk/flight_hub_sdk.log
```

提交问题时优先使用标准诊断包，自动收集最近 4 个 SDK 日志文件，每个文件最多保留最后 2 MiB，并对配置和日志中的密钥类内容做脱敏：

```
sudo /opt/dehub/bin/dehubctl diagnose
```

诊断包输出在 `/opt/dehub/diagnostics/dehub-diagnostics-<时间>.tar.gz`。需要保留完整轮转日志时，使用受控介质手工打包；原始日志可能包含设备序列号、地址等运行信息，只能通过受控渠道传输：

```
DATA_ROOT=/data/dehub
OUTPUT="/tmp/dehub-sdk-logs-$(date -u +%Y%m%dT%H%M%SZ).tar.gz"
sudo tar -C "$DATA_ROOT/logs" -czf "$OUTPUT" sdk
sudo chmod 0600 "$OUTPUT"
```

SDK 日志轮转上限为单文件 64 MiB、最多 64 个文件。修改日志配置后必须执行 `sudo /opt/dehub/bin/dehubctl restart` 使 API 容器重新读取配置。

7 备份与恢复

```
sudo /opt/dehub/bin/dehubctl backup                     # 默认输出到 /data/dehub-backups
sudo /opt/dehub/bin/dehubctl backup --output /mnt/backup
sudo /opt/dehub/bin/dehubctl backup --with-storage      # 连媒体目录一起打包
sudo /opt/dehub/bin/dehubctl backup --verify BACKUP.tar.gz
```

备份包含数据库、配置、License 和机器绑定，含密钥，目录 `0700`、归档 `0600`。 默认**不含媒体目录** `storage`：它只随设备上报增长、不随版本变化，升级与回滚都不改写它， 把它打进归档会让每次升级多花十几分钟（实测 11 GB 媒体让升级总耗时从 ~4 分钟涨到 ~20 分钟）。 需要整机快照（例如换机器冷备）时加 `--with-storage`；归档清单和备份结果中的 `includesStorage` 表示该归档是否实际携带媒体目录，媒体目录不存在时即使请求了 `--with-storage` 也返回 `false`。恢复仅在字段显式为 `true` 时切换媒体目录； `false` 或字段缺失时保留线上媒体目录不动，即使包内意外存在 `storage/`。 必须使用受控的加密介质异机保存并设定保留期。建议每日备份、每月做一次真实恢复演练， 未演练的备份不能视为可恢复。

```
# 同机原地恢复
INSTALL_ID="$(sudo cat /opt/dehub/state/install-id)"
sudo /opt/dehub/bin/dehubctl restore BACKUP.tar.gz \
  --mode in-place --confirm-current-install "$INSTALL_ID"

# 裸机恢复（新机器，需授权方签发的 REBIND 证书）
sudo /opt/dehub/bin/dehubctl restore BACKUP.tar.gz \
  --mode bare-metal --rebind-license /path/rebind-license.lic
```

恢复先导入临时数据库和文件并完成校验，全部通过后才切换，失败不会破坏现有生产数据。 不要直接覆盖或删除生产数据库目录。

8 升级与回滚

升级包内嵌签名并由 `dehubctl` 自动验签，部署人员不需要执行单独的验签命令：

```
sudo /opt/dehub/bin/dehubctl upgrade dehub-offline-upgrade-NEW.tar.gz --plan
sudo /opt/dehub/bin/dehubctl upgrade dehub-offline-upgrade-NEW.tar.gz
```

升级自动完成验签、维护模式、强制备份、数据库迁移、smoke 校验和版本切换； smoke 失败会用刚生成的备份自动回滚到旧版本。禁止手工修改 `DEHUB_VERSION`、 `current` 链接或覆盖版本目录。

升级前的强制备份只含数据库、配置与 License（媒体目录原地不动，回滚不需要它）， 因此升级耗时与媒体体量无关。确实需要连媒体一起归档时，在目标机上执行 `dehubctl upgrade dehub-offline-upgrade-`

`NEW.tar.gz --with-storage`。 手工回滚到不识别 `includesStorage` 的旧版本时，使用的备份必须实际携带 `storage/`； 不带媒体目录的新格式备份会在切换版本前被拒绝，不会用空目录覆盖线上媒体。

9 常见故障

现象	处理
大量 API 返回 401	退出后重新用管理员账号登录；不要在前端配置固定系统 token
底座设备管理不可用	从服务器探测底座端口（`timeout 5 bash -c '</dev/tcp/<底座IP>/30812'`），核对地址、令牌和 MinIO 凭据
`CONFIG_INVALID`	检查配置权限 `0600 root:root`、是否有未知键或未替换的 `<...>` 占位符
`LICENSE_MACHINE_MISMATCH`	机器标识已变更，重新生成请求并申请 `REBIND` 证书
容器或数据库启动失败	执行 `dehubctl status`、`logs postgis --tail 300`、`logs dehub-api --tail 300`，检查磁盘、内存、端口和目录权限

不要执行 `docker compose down -v`，不要删除 `/data/dehub` 下的数据库目录，不要清空 `/opt/dehub/state`，也不要直接修改数据库迁移记录，这些操作可能造成不可恢复的数据丢失。

提交支持前先生成诊断包：

```
sudo /opt/dehub/bin/dehubctl diagnose    # 输出在 /opt/dehub/diagnostics，权限 0600
```

诊断包不含 License 正文、业务数据和明文密钥，提交前可再人工检查一次。

10 验收清单

- [ ] 服务器规格、SSD、Docker/Compose、时间同步符合要求，防火墙策略已记录。
- [ ] 发布包来自受控渠道，SHA-256 与交付记录一致。
- [ ] `verify`、`status`、`smoke` 全部成功，`curl -i http://127.0.0.1:19080/healthz` 正常。
- [ ] 管理员可登录，权限与角色正常，无固定系统 token。
- [ ] License 有效，且与当前机器、install ID 和版本匹配。
- [ ] 设备管理、底座 PaaS、通知 WebSocket、MinIO 和实际业务链路验收通过。
- [ ] 已完成一次备份，并在隔离环境完成一次恢复演练。
- [ ] 已用测试升级包验证升级流程和失败自动回滚。
- [ ] 客户已建立备份保留、巡检、账号、证书和变更制度。

验收记录至少保存：客户/项目、服务器 IP、版本、包 SHA-256、License ID 后缀与 revision、 安装时间、实施人、外部集成检查结论、备份恢复与升级回滚演练结论、遗留项和客户签字。 不得记录密码、令牌、私钥或完整 License 内容。