

运维平台

运维管理平台，是用于维护AIRIOT平台的服务或者模板的板块，可进行版本升级等系列操作。

首页

进入运维管理平台，展示的首页为信息统计页面，会显示服务总数、模块总数，服务运行状态统计，服务更新日志和模块更新日志

The screenshot shows the homepage of the Operations Management Platform. On the left is a dark sidebar with a logo and navigation links: 服务管理, 模块管理, 资源统计, 服务诊断, 日志管理, and admin. The main content area has four sections: 1. 统计: Shows Service Total (121) and Module Total (47). 2. 服务状态: A pie chart indicating 99% of services are running and 1% are stopped. 3. 服务更新日志: A table listing updates for various services. 4. 模块更新日志: A table listing updates for various modules. Buttons for '一键离线升级' (One-click Offline Upgrade) and '一键在线升级' (One-click Online Upgrade) are located at the top right.

服务名称	更新时间	更新信息	更新版本
	2024-09-26 10:09:50	安装成功	
https://d.airiot.link/modbus_rtu/v4.1.16/modbus_rtu-linux-x86_64.tar.gz	2024-09-25 16:38:15	安装成功	v4.1.16
	2024-09-23 16:52:59	安装成功	
	2024-09-23 13:55:16	安装成功	
	2024-09-23 13:42:23	安装成功	
	2024-09-23 13:41:43	安装成功	
	2024-09-23 13:39:22	安装成功	

模块名称	更新时间	更新信息	更新版本
@airiot/gis	2024-09-25 15:27:41	安装成功	4.11.0
@airiot/gis	2024-09-25 15:01:23	安装成功	4.11.0
@airiot/gis	2024-09-25 14:08:27	安装成功	4.11.0
@airiot/device	2024-09-25 11:53:06	安装成功	4.1.25
@airiot/components	2024-09-25 11:27:49	安装成功	4.2.8
@airiot/3d	2024-09-25 11:06:07	安装成功	4.1.3
@airiot/gis	2024-09-25 10:53:26	安装成功	4.11.0
@airiot/settings	2024-09-25 10:16:14	安装成功	4.2.8

1. **一键在线升级：**点击可进行全部服务和模块的在线升级，升级过程中只能顺序更新，升级时间较长，且升级过程中不允许其他的服务、模块进行升级，驱动下载也无法进行
2. **一键离线升级：**点击可进行服务和模块的离线升级，即在无网络的情况下进行升级。

服务管理模块主要用来添加服务和管理已添加的服务，可对服务进行升级和回退版本

The screenshot shows the 'Service Management' section of the 'Operation and Maintenance Platform'. It displays a table of services with columns for service name, state, current version, dependency version, image, port, and creation time. At the top, there are buttons for 'Start', 'Restart', 'Stop', 'Delete', and 'Batch Upgrade'. A red arrow points to the '针对选中的服务' (Target Selected Services) button. Another red arrow points to the '针对全部服务' (Target All Services) button, which is highlighted with a red border.

服务名称	状态	当前版本	依赖最低版本	最新版本	镜像	发布端口	创建时间	操作
6257e3362a2e4835d2bc8c6f-66bef79e80f1ba0839dd4b04-driver-mqtt-client	运行	v4.0.22		v4.0.31 升	airiot/driver-mqtt-client:v4.0.22	9000/tcp	2024-09-23 09:32:34	历史版本 离线升级 修改 ...
6257e3362a2e4835d2bc8c6f-66c43e3699208ae6deda2f49-tcp-server-driver	运行	v4.0.47		v4.0.47 升	airiot/tcp-server-driver:v4.0.47	10032/tcp.0.0.0.32961->9404/tcp	2024-09-23 09:32:34	历史版本 离线升级 修改 ...
6257e3362a2e4835d2bc8c6f-66cb9141e493aa19bd7e89-opcda-datalayer	运行	v4.0.5			airiot/opcda-datalayer:v4.0.5		2024-09-23 09:32:34	历史版本 离线升级 修改 ...
6257e3362a2e4835d2bc8c6f-66cd2e3f5ff801059ca85d3-driver-media-server	运行	v4.1.7		v4.1.11 升	airiot/driver-media-server:v4.1.7	0.0.0.0:32883->5060/udp.0.0.0.33026->58200/tcp.0.0.0.33027->8080/tcp	2024-09-23 09:32:34	历史版本 离线升级 修改 ...
6257e3362a2e4835d2bc8c6f-66d04913f54cf2d6ea591157-http-server-driver	运行	v4.1.6		v4.1.7 升	airiot/http-server-driver:v4.1.6	10032/tcp.0.0.0.32950->9505/tcp	2024-09-23 09:32:34	历史版本 离线升级 修改 ...
6257e3362a2e4835d2bc8c6f-66df9e1511e2ff4b79b42b9-opcda-datalayer	运行	v4.0.5			airiot/opcda-datalayer:v4.0.5		2024-09-23 09:32:34	历史版本 离线升级 修改 ...
625f6dbf5433487131f0ff7-64c0e794df68df550ef105cf-g340t	运行	v4.0.17		v4.0.17 升	airiot/g340tv4.0.17	0.0.0.0:33299->58588/tcp.9000/tcp	2024-09-23 09:32:34	历史版本 离线升级 修改 ...

已创建的服务可通过输入服务名称进行搜索，

服务管理

搜索 服务名称



- 启动：启动全部服务；
- 重启：重启全部服务；
- 停止：全部服务停止运行；
- 创建启动：创建并启动全部服务；
- 离线上传中转服务协议：可在无网络的情况下上传数据中转协议；
- 离线上传驱动：可在无网络的情况下上传驱动服务；
- 编辑部署文件：直接编辑部署文件，通过编辑部署文件来控制所有服务，比如控制服务的运行状态；
- 上传镜像：镜像即把一个程序运行所需要的配置和环境打成包进行上传；
- 添加服务：点击添加新的服务；可在线添加直接安装服务即可，也可离线添加，直接上传服务即可，还可输入相应信息进行离线服务的添加。

The screenshot shows a service management interface with a sidebar on the left containing navigation links like '运维管理平台', '服务管理', '模块管理', '资源统计', '服务诊断', '日志管理', and 'admin'. The main area is titled '添加服务管理' and has tabs for '在线添加', '离线添加', and '高级添加'. A search bar labeled '服务名称' is present. Below it is a grid of service cards:

- abplc**: AB PLC驱动, 最新版: v4.0.11, 最新更新时间: 2024-01-16 21:26:07, 安装
- algorithm**: 算法服务, 最新版: v4.0.12, 最新更新时间: 2024-08-22 15:25:02, 安装
- auth**: 角色权限过滤服务, 最新版: v4.2.30, 最新更新时间: 2024-09-10 16:21:49, 安装
- check**: 网络检查服务, 最新版: v4.2.10, 最新更新时间: 2024-06-14 16:20:35, 安装
- compute-record**: 查找引用和计算, 最新版: v4.0.9, 最新更新时间: 2024-05-23 17:17:58, 安装
- computed**: 计算节点服务, 最新版: v4.3.8, 最新更新时间: 2024-07-10 18:10:09, 安装
- core**: 核心服务, 最新版: v4.3.144, 最新更新时间: 2024-09-09 18:01:31, 安装
- data-relay**: 数据中转服务, 最新版: v4.1.1, 最新更新时间: 2024-08-22 15:29:56, 安装
- data-service-driver**: 数据接口驱动, 可用于http client及数据库操作, 最新版: v4.0.22, 最新更新时间: 2024-03-04 18:26:08, 安装
- data-source**: 数据接口服务, 最新版: v4.7.19, 最新更新时间: 2024-08-22 15:21:12, 安装
- db-backup**: 数据库备份还原, 最新版: v4.0.17, 最新更新时间: 2024-08-21 15:17:49, 安装
- db-driver**: db驱动是一款专门用来采集常见关系数据库数据的驱动, 最新版: v4.0.17, 最新更新时间: 2024-03-20 10:18:44, 安装

10. 开始、停止、重启、删除和批量升级：开始、停止、重启、删除和批量升级选中服务（==注： ==不能删除的服务和模块）；
11. 历史版本：历史版本记录的是该服务历史升级版本；
12. 离线升级：点击升级，上传升级服务，可在无网络的情况下进行升级；

The screenshot shows a service history interface with a table of upgrade versions:

创建时间	操作
2023-06-29	<input checked="" type="checkbox"/> 历史版本 <input type="button" value="离线升级"/> <input type="button" value="修改"/> ...
2023-06-29	<input checked="" type="checkbox"/> 历史版本 <input type="button" value="离线升级"/> <input type="button" value="修改"/>
2023-06-29	<input checked="" type="checkbox"/> 历史版本 <input type="button" value="离线升级"/> <input type="button" value="修改"/>

A red box highlights the '历史版本' column. A red box also highlights a dropdown menu that appears when hovering over the '...' button of the first row, listing:

- 日志
- 图表
- 检查
- 控制台

13. 日志：服务运行输出的日志信息；
14. 图表：记录的是该服务器内存、cpu和网络使用情况，可设置刷新时间；
15. 检查：当前运行容器的详情信息；

16. 控制台：用来控制当前服务运行的状态；

← 控制台

The screenshot shows a control panel interface. At the top, there is a field labeled "命令：" (Command) with a placeholder "* 命令:". Below it is a toggle switch labeled "使用固定命令" (Use fixed command). Further down are fields for "用户名" (Username) and "密码" (Password), both with placeholder text. At the bottom are two buttons: "保存" (Save) and "取消" (Cancel).

17. 使用固定命令：选中后，点击文本框，命令直接显示，选择即可；

18. 用户：只能用root用户

模块管理把平台中的功能分模块管理，所有的模块都展示在模块管理平台中，可添加或更新新的模块；

The screenshot shows the module management interface. On the left is a sidebar with navigation links: 运维管理平台, 服务管理, 模块管理 (selected), 资源统计, 服务诊断, 日志管理, and admin. The main area is titled "模块管理" with a search bar and a "批量升级" button. A "添加/更新模块" button is also present. The table lists various modules with columns for 模块名称 (Module Name), 创建时间 (Creation Time), 描述 (Description), 版本 (Version), 依赖最低版本 (Dependent Minimum Version), 最新版本 (Latest Version), and 历史版本 (Historical Version). Each row includes a checkbox and a "历史版本" (Historical Version) button.

模块名称	创建时间	描述	版本	依赖最低版本	最新版本	历史版本
@airiot/3d	2024-09-25 11:06:07	AIRIOT三维组件	v4.1.3		v4.1.3 ↗	[历史版本]
@airiot/ai	2024-09-20 11:42:01	AIRIOT AI辅助模块	v4.0.0		↗	[历史版本]
@airiot/algorithm	2024-08-14 09:53:12	我的算法模块	v4.0.0		v4.0.1 ↗↗	[历史版本]
@airiot/apps	2024-07-02 11:14:22	AIRIOT平台二次开发	v4.0.1		v4.0.2 ↗↗	[历史版本]
@airiot/auth	2024-09-20 14:40:07	AIRIOT平台前端权限管理	v4.5.3		v4.5.4 ↗↗	[历史版本]
@airiot/command-plan	2024-09-23 15:52:51	AIRIOT指令计划	v3.0.0		↗	[历史版本]
@airiot/components	2024-09-25 11:27:49	AIRIOT公共组件	v4.2.8		v4.2.8 ↗	[历史版本]
@airiot/computed	2024-09-23 10:31:00	AIRIOT平台前端计算节点	v4.1.3		v4.1.6 ↗↗	[历史版本]
@airiot/cordova	2024-09-06 17:09:10	AIRIOT平台安卓App	v4.0.0		v4.0.2 ↗↗	[历史版本]
@airiot/core	2024-09-23 14:22:45	AIRIOT平台前端核心	v4.5.14		v4.5.17 ↗↗	[历史版本]
@airiot/dashboard	2024-09-24 16:17:02	云组态平台信息板编辑及管理模块	v4.10.3	依赖 @airiot/echarts v4.0.8 ↗	v4.10.3 ↗	[历史版本]
@airiot/dashboard-connect	2024-09-23 10:13:53	画面中连线	v4.1.2		v4.1.1 ↗↗	[历史版本]
@airiot/data-api	2024-08-29 17:11:13	数据接口模块	v4.0.12		v4.0.14 ↗↗	[历史版本]
@airiot/data-relay	2024-09-24 13:57:46	AIRIOT平台数据中转	v4.0.0		v4.0.0 ↗	[历史版本]

当该模块中的功能新增、更新或修复bug时，可更新该模块版本。

资源统计

资源统计页面统计的是cpu、内存和磁盘的使用情况，超负荷时显示红色



服务诊断

出现错误的服务会在服务诊断页面显示



日志管理

日志管理包括操作日志、运维日志和服务管理日志

操作日志

操作日志记录的是运维平台的操作记录，可通过操作时间和操作类型查询操作日志

The screenshot shows the 'Operation Log' page of the 'Operations Management Platform'. On the left sidebar, there are links for Service Management, Resource Statistics, Service Diagnosis, and Log Management, with 'Log Management' being the active tab. The main area has a search bar with 'Operation Time' set to '2024-06-05 00:00 ~ 2024-06-05 14:30' and 'Operation Type' dropdown. Below is a table with columns: Operation Time, Operation Type, Operation IP, Operation Content, Original Data, Latest Data, and a 'View Details' button. The table contains six log entries from June 5, 2024.

操作时间	操作类型	操作IP	操作内容	原始数据	最新数据	
2024-06-05 14:33:11	安装数据中转服务	172.20.0.19	于2024-06-05 14:33:11在6273677fe5bd0d8ebc7d6a5f项目安装go-example-server数据中转服务: 版本=4.0.0.实例=666006a6401c992b3eab3337	空	空	<button>查看详情</button>
2024-06-05 14:33:02	删除数据中转服务	172.20.0.19	于2024-06-05 14:33:02删除6273677fe5bd0d8ebc7d6a5f-665fd700401c992b3eab3333-go-example-server服务成功	空	空	<button>查看详情</button>
2024-06-05 14:24:08	安装数据中转服务	172.20.0.19	于2024-06-05 14:24:08在6257e3362a2e4835d2bc8c6f项目安装go-example-server数据中转服务: 版本=4.0.0.实例=66600487401c992b3eab3336	空	空	<button>查看详情</button>
2024-06-05 14:23:10	模块安装	113.224.157.165	admin于2024-06-05 14:23:10离线安装模块@airiot/core版本从4.2.24升级到4.2.24成功	空	@airiot/core	<button>查看详情</button>
2024-06-05 14:23:07	安装数据中转服务	172.20.0.19	于2024-06-05 14:23:07在6257e3362a2e4835d2bc8c6f项目安装go-example-server数据中转服务: 版本=4.0.0.实例=6660044a01c992b3eab3335	空	空	<button>查看详情</button>

默认查询

默认查询当前天的操作日志，点击设置框可自定义默认查询时间，点击重置icon，查询全部的操作日志

A modal dialog for setting the default query time. It features a top row with 'Current Day' and a 'Reset' icon. Below is a date range selector with 'From' and 'To' fields. The 'From' field is set to '1' days ago, and the 'To' field is set to 'Current'. There are 'Cancel' and 'Confirm' buttons at the bottom.

清除日志

清除日志包括一键清除和周期清除

1. 一键清除指将操作日志进行一次性的清除操作

清除日志

X

* 清除方式: 一键清除

* 操作类型: 清除数据

操作类型用来定义清除数据的不同操作方式, 当执行清除数据时, 选择时间范围的数据均被清除, 选择保留数据时, 定义保留数据的时长, 其余数据被清除

* 时间定义: 开始日期 → 结束日期

可选时间范围或只定义一个时间, 定义一个时间默认将定义时间之前日志清除

日志类型:

确定

取消

操作类型-清除数据: 选择清除数据时, 需定义时间范围, 时间范围内的数据将被清除

操作类型-保留数据: 选择保留数据时, 需定义时间天数, 设置时间内的数据将不会被清除, 其余全部被清除

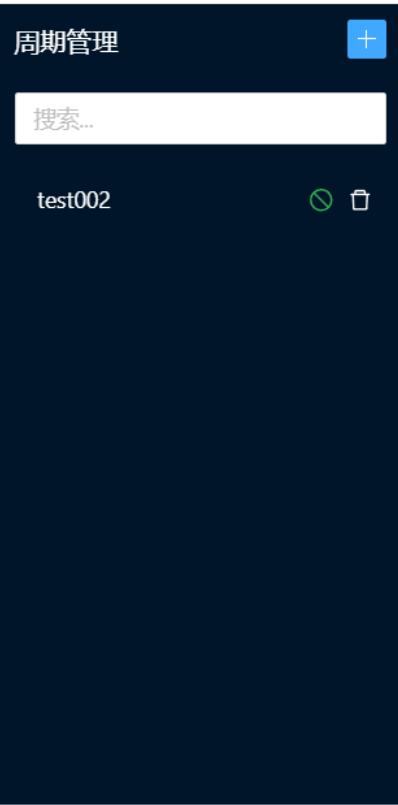
日志类型: 选择类型的日志将会被清除, 不选的情况下默认清除全部日志

2. 周期清除指将操作日志定期进行清除操作

清除日志

X

* 清除方式: 周期清除



* 周期名称:

日志类型:

* 清除周期:

数量:

日期单位:



周期定义后，将会按照周期定时清除周期内的日志信息

清除时间: 请先选择周期

可定义每个周期清除执行时间，不定义默认周期的00:00:00执行

保留数据: 输入数值

天

确定

取消

清除周期: 周期设置后，按照周期清除日志信息

清除时间: 每个周期具体的执行时间，不定义默认周期的00:00:00执行

保留数据: 设置后，定义时间内的数据将会被保留，其余数据将全部被清除

运维日志

记录系统、设备或软件在运行过程中产生的各种操作的信息集合

1. 显示时间戳：选中后，日志中显示时间戳
 2. 时间：可通过时间查询设置时间内的运维日志
 3. 行数：运维日志显示行数
 4. 自动刷新：选中后，运维日志可自动刷新

可通过快捷搜索，搜索日志内容

运维日志

日志查询

查询

显示时间截:

时间:

行数:

自动刷新:

日志详情

```
[user_session=MTcyMTg4NzU5MnxEdi1EQkFFQ180UUFBUKFCRUFBQV9nR0fNFFBQWdemRISnBibWNNQ3dBSmNlSnZhBVZqZEVS_0JuTjbjWx1Wnd3YUFCZzJN] [uZWRNeK5qSmhNbUwT0RNMVpEsmlZemhqTm1ZR2MzUnlhZ -cnE] Referer:[http://121.89.244.23:13030/api/log/operation] User-Agent:[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.0.0 Safari/537.36]" ip=223.70.159.244 method=GET proto=HTTP/1.1 res_length=175044 res_status=200 tag=_request_trace_id=trace_id-1-2024.07.25.14.41.08.527-474 url="/api/docker/containers/operations/logs/info?timestamps_ms=false&tail=100&refresh=true" user_agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.0.0 Safari/537.36" user_id= version=4.0.0 Dstime="2024-07-25T14:41:09+08:00" level=info msg="["[http://api/docker/containers/operations/logs/info?GET-223.70.159.244-200(5ms)] content_length=0 header='{"accept": "application/json", "accept-encoding": "gzip, deflate", "accept-language": "zh-CN,zh;q=0.9", "authorization": "Bearer eyJhbGciOiUzUxMlsInR5cCl6IkpxVCi9eyJleHh0JEq3Mje4OTYwMjQslmhC16MTcyMTg4ODgyNCwibmJmjoxNzlxDg4ODI0LCJzdWlOiJhZG1pbj9DszOl62HG7mKeY5-MiE5ElMfv2fbXjEtThTFGg-K-EVjkMlOsQJxFTU79KYLEVWvZLeY9HMWX9qkWQ Connection:[keep-alive] Content-Type:[application/json] Cookie:"}]
```

```
[user_session=MTcyMTg4NzU5MnxEdi1EQkFFQ180UUFBUKFCRUFBQV9nR0fNFFBQWdemRISnBibWNNQ3dBSmNlSnZhBVZqZEVS_0JuTjbjWx1Wnd3YUFCZzJN] [uZWRNeK5qSmhNbUwT0RNMVpEsmlZemhqTm1ZR2MzUnlhZ -cnE] Referer:[http://121.89.244.23:13030/api/log/operation] User-Agent:[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.0.0 Safari/537.36]" ip=223.70.159.244 method=GET
```

服务管理日志

服务管理日志用来记录服务的安装、卸载和运行状态的记录

The screenshot shows the 'Service Management Log' section of a management platform. On the left sidebar, there are icons for Service Management, Resource Statistics, Service Diagnosis, Log Management, and a user account labeled 'admin'. The main area has a title 'Service Management Log' and a sub-section 'Log Search'. It includes a toggle for 'Display Time Range' (on), a date range selector from '选择日期' to '到', a row limit input set to '100', and an 'Automatic Refresh' toggle (on). Below this is a section titled 'Log Details' which is completely redacted with a large black rectangle.

服务管理

1. 不能删除的服务和模块

服务名称	模块名称
web、traefik、spm、postgres、nmqtt、influx、etcd、core、auth、task、front、redis、loki、promtail	@airiot/core、@airiot/auth

2. 依赖版本

服务管理和模块管理在版本更新过程中，可能存在相互依赖的关系，如服务管理中core升级后，需依赖某一服务或者模块，在会在core升级成功后，提示依赖的其他模块或者服务，须同步升级。

可到表格中【依赖最低版本】字段下升级依赖的模块，也可到对应的服务管理或者模块管理进行升级。

有效解决多服务、模块未同步更新导致功能无法正常使用的问题。

空间管理

空间可以实现灵活的多项目管理，实行租户管理，每个租户可以创建自己的项目。空间的功能分为两个管理模块，一个客户空间管理（客户即整个管理平台的超级管理员，只有一个账号），一个租户空间管理，客户可以创建多个租户。

通过空间的功能，实现用户创建多项目，单用户下有多租户创建多项目，每一个项目之前数据、权限、内容等彼此保密独。

空间管理包括[项目管理](#)、[租户管理](#)、[授权管理](#)、[系统设置](#)和个人设置。

空间管理

- 项目管理
- 租户管理
- 授权管理
- 系统设置

admin

项目统计

总数 70 租户项目 1 平台项目 69

近一周项目新增统计

日期	新增项目数
2023-05-30	3
2023-05-31	3
2023-06-01	0
2023-06-02	0
2023-06-03	3
2023-06-04	0
2023-06-05	4

近一周租户新增统计

日期	新增租户数
2023-05-30	0
2023-05-31	2
2023-06-01	0
2023-06-02	0
2023-06-03	0
2023-06-04	0
2023-06-05	0

最新消息

暂无数据

中文 英文

项目管理

项目管理中，已添加的项目在项目界面以列表形式或卡片形式进行展示。

The screenshot displays a grid of project cards. Each card contains a project name, a summary section with 'admin' and '项目剩余授权' (remaining authorization), and two detailed sections: '项目设备统计' (Equipment Statistics) and '其他信息' (Other Information). The cards are color-coded: red, green, blue, purple, and grey. A sidebar on the left shows navigation links for '空间管理' (Space Management), '项目管理', '租户管理', '授权管理', '系统设置', '帮助文档', and 'admin'. Top navigation includes a search bar, filter buttons, and a '添加项目' (Add Project) button.

已添加的项目可通过项目名称直接搜索，也可点击更多过滤按钮，通过设置过滤条件进行精确搜索，过滤字段包括项目状态、项目范围和扩展字段中选中过滤查询中显示的字段

The screenshot shows the same project management interface as above, but with a '项目过滤' (Project Filter) modal open in the center. The modal contains fields for '项目状态' (Project Status) with '禁用' (Disabled) selected, '所属租户' (Owner Tenant) set to '全部项目' (All Projects), '数字' (Number) with '无限制' (Unlimited) selected, and '文本' (Text) with a search input field. Below the modal, the project cards are visible, and the sidebar and top navigation are present.

通过过滤器搜索到结果后，项目的左上方会提示当前所选择的过滤条件，在其最右边有个清除按钮可以清空过滤条件。

过滤条件：所属租户:admin

< 1 2 3 >



当使用搜索条件搜索时，会显示搜索的高亮标识，用户能够知道当前列表是经过搜索后的结果

添加项目

点击“添加项目”按钮，填写项目信息，添加项目，填写完信息后点击保存按钮，返回项目管理界面。

R
空间管理

← 添加项目

项目ID①:	<input type="text"/>
* 项目名称:	<input type="text"/>
* 项目行业:	<input type="text"/>
所属租户:	<input type="text"/>
* 授权②:	<input type="button" value="添加授权"/>
描述:	<input type="text"/>
背景色③:	<input type="button" value="选择颜色"/>
域名④:	<input type="text"/>
<input type="checkbox"/> 支持Https格式 <input type="checkbox"/> 使用域名访问⑤	
端口⑥:	<input type="text"/>
<input type="button" value="保存"/> <input type="button" value="取消"/>	

项目ID:项目ID仅支持小写字母、数字，不可为纯数字、default，必须以字母开头。不填时，保存时将自动生成。项目ID会在中台访问时作为浏览器路径的组成部分。

项目名称:自定义填写项目名称；

项目行业:点击选择项目行业，选项为系统内置行业；

所属租户:点击选择所属租户，创建项目时可将项目分配给所属租户进行管理，同一个项目只能属于一个租户，选择所属租户时支持模糊搜索；

授权: 设置项目可用的授权信息，包括基础授权、数据采集与控制引擎授权、数据分析引擎授权、可视化引擎授权、业务流引擎授权、二次开发引擎授权和高级授权；

项目信息中的必填项需全部填写，其中基础授权信息的必填项包括：授权时长、起始时间、用户数量、数据表数量，左侧是客户可用的授权信息，右侧是需要填入的项目授权信息，填入的数值不能超过左侧数值；

授权定义 X

基础授权 数据采集与控制引擎授权 可视化引擎授权 数据分析引擎授权 业务流引擎授权 二次开发引擎授权 高级授权

客户可用授权 项目授权

时间

时间	36124天	* 时长:	11111
	2123-06-06	* 起始时间②:	2024-07-08

用户

用户数量	99939752	* 用户数量:	11111
------	----------	---------	-------

数据表

数据表数量	99906676	* 数据表数量:	111
设备记录数量	-	设备记录数量②:	111

确定

• 基础授权

时长: 项目的有效时长;

起始时间: 项目的有效起始时间, 项目有效时才可以登录平台;

用户数: 项目可添加的用户数量;

数据表数量: 该项目中可创建的数据表数量;

设备记录数量: 该项目中可创建的设备数量, 不填写时默认不限制;

• 数据采集与控制引擎授权

数据点-点数: 项目可创建的数据点数量;

• 数据分析引擎授权

BI (数据集+数据视图) : 项目是否用BI功能;

• 可视化引擎授权

画面: 项目可创建画面的数量;

GIS组件: 项目是否可使用GIS组件;

三维组件: 项目是否可使用三维组件;

ISC视频组件: 项目是否可使用ISC视频组件;

海康无插件视频: 项目是否可用海康无插件视频;

画面回放（画面事件画面回放动作）：项目是否可使用画面回放功能；

手机前台打包app：项目是否可用手机前台打包app；

- **业务流引擎授权**

流程数量：项目可创建流程的数量； 通知：项目中是否可使用发送微信、发送企业微信、发送钉钉和发送短信功能

网关：项目中是否可使用mqtt网关、opcua网关、opcda网关和db网关功能

执行脚本：项目中是否可使用执行脚本功能

- **二次开发引擎** 二次开发：项目中是否可使用OauthApp和扩展应用功能

- **高级授权** 数据接口：项目中是否可使用数据接口功能；

AI：项目中是否可以使用AI功能；

视频管理：项目中是否可以使用视频管理模块，视频管理模块可对平台中接入国标GB2818协议的摄像头设备进行统一管理，可实现视频的监控，设备使用情况分析等功能；

系统设置-催款模块功能：项目中是否可以使用催款模块，催款模块用来提醒非admin用户授权到期提醒配置；

kafka服务：项目中是否使用kafka服务功能；通过kafka服务，实现将项目A（生产者）实时数据发布到kafka的topic中，项目B(消费者)可通过订阅kafka服务中的topic获得生产者的实时数据。

中转服务：项目中是否可以使用中转服务功能

是否免密登录：选中后登录项目时无需输入密码可直接登录；

描述：描述项目信息；

背景色：用于设定项目管理卡片展示时，卡片的背景色，如图：



域名：域名不可重复，填写后并开启【使用域名访问】时，访问项目后台和前台可通过域名访问项目；注意域名须为备案后的域名。

使用域名访问：选中后进入前后台时，访问地址为域名地址

支持https格式：选中后需上传https格式证书，上传后访问地址为https格式

端口：每个项目可以定义自己的端口号，目前端口号限制在5001-65535范围内，其他不可用

媒体库公共资源：公共资源文件夹用于存储空间项目的资源共享。启用后，项目媒体库中将创建公共资源文件夹，所有启用的项目均可查看和使用其中的文件，该文件夹不可删除和修改移动等。



公共资源权限设置：启用媒体库公共资源后，可进行公共资源权限的设置。

- 创建子文件夹：选中后，项目用户可在公共资源中创建子文件夹。
- 上传文件：选中后，项目用户可在公共资源中上传文件。
- 修改子文件夹、文件：选中后，项目用户可修改公共资源下的子文件、文件。
- 删除子文件夹、文件：选中后，项目用户可删除公共资源下的子文件、文件。

批量删除

删除一个或多个项目，被删除的项目不能恢复。点击批量删除按钮，选中要删除的项目或点击全选开关选中所有项目，点击批量删除按钮即可，未选中项目前，批量按钮置灰。

项目 搜索项目名称 更多过滤

卡片形式 列表形式 + 添加项目

共 52 条 - 每页 24 条 全选 批量删除 取消

项目	点数	用户数	其他信息
fwer...	4	4	查看详情
水务...	900	900	查看详情
功能...	999+	999+	查看详情
还原...	111	11	查看详情
媒体...	999+	999+	查看详情
新项目...	100	10	查看详情
hp...	333	33	查看详情
数据...	0	0	查看详情
数据...	0	0	查看详情
测试...	15	9	查看详情
hp...	0	0	查看详情

查看后台

点击进入后台按钮，进入后台查看。

点击查看后台

水务项目

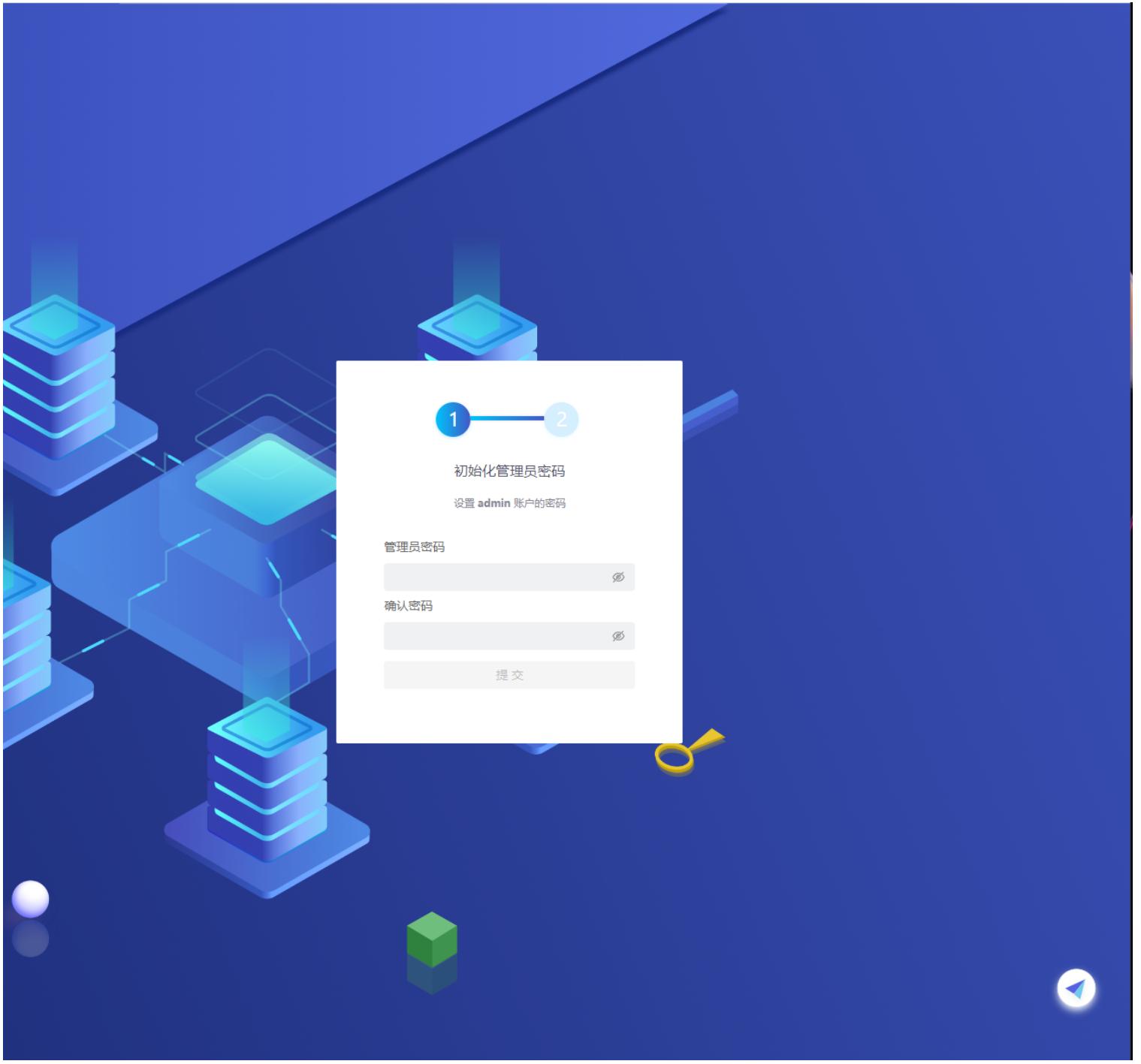
丽娜的空间 项目剩余授权

点数	用户数	其他信息
900	900	查看详情

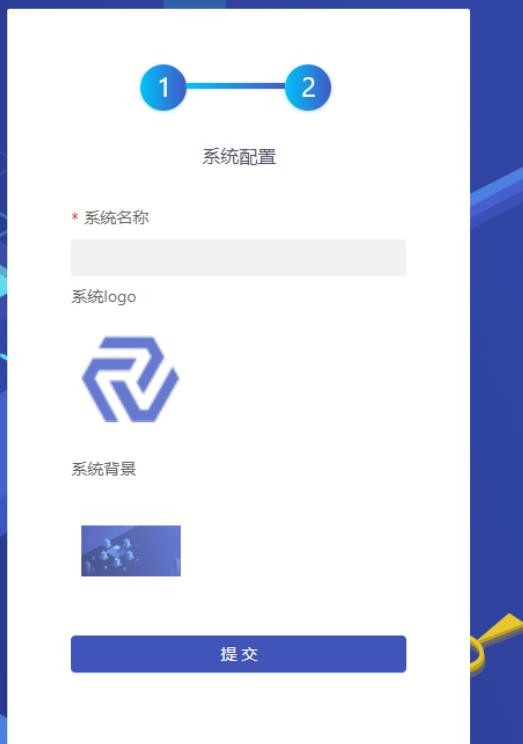
项目设备统计

总数	在线数
2	0

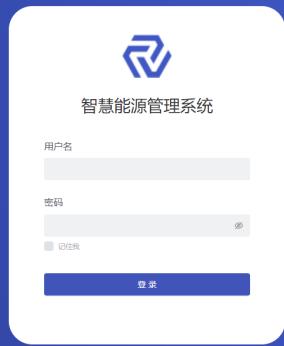
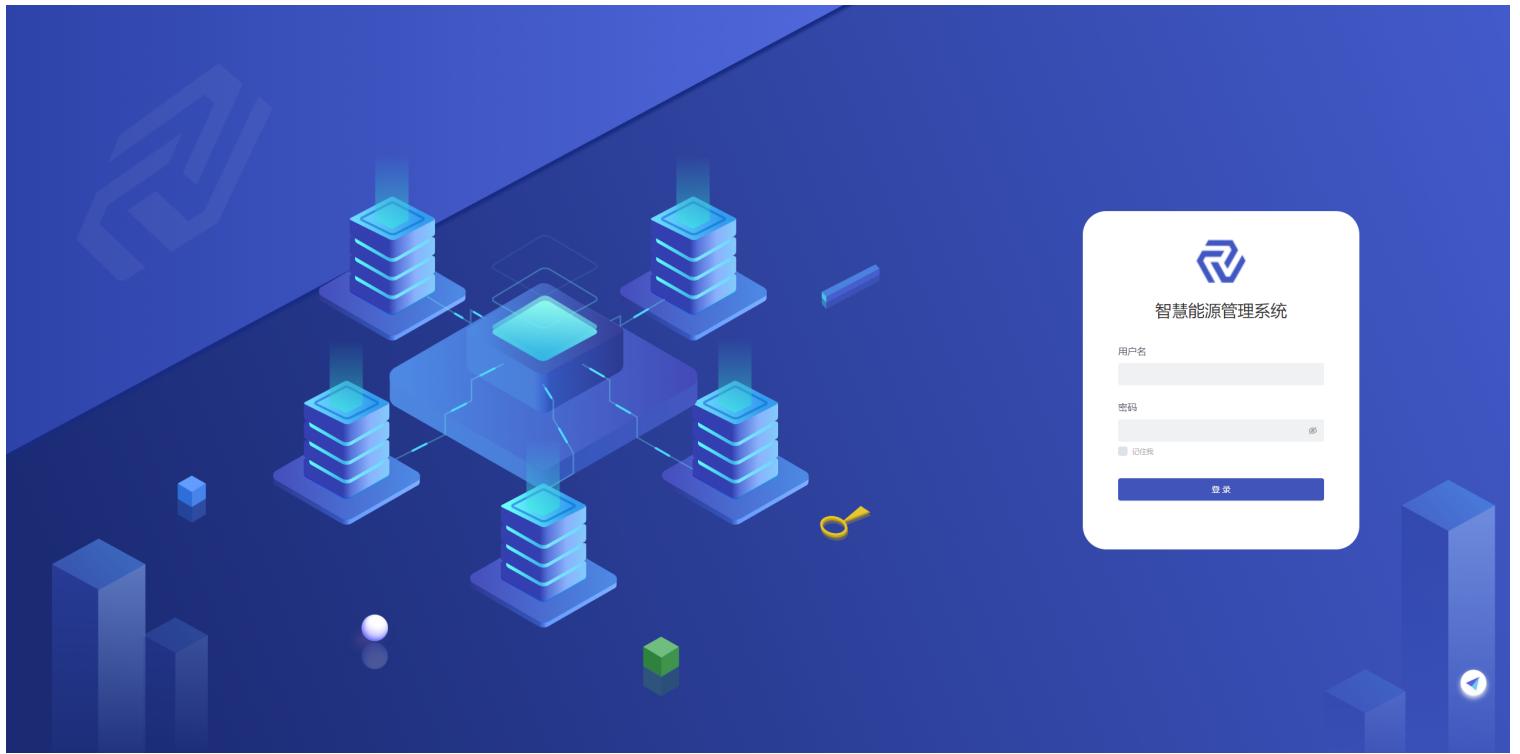
新创建的项目需初始化管理员密码；



密码配置完毕后，进入下一步骤系统配置；



配置完毕后，登录项目，输入用户名和密码



登录后进入系统后台设置界面，如图：

智慧能源管理系统

欢迎使用 智慧能源管理系统 全局搜索 搜索

跳转至前台

数据概览

设备管理 > 设备监控 > 报表管理 > 数据分析 > 可视化 > 任务管理 > 业务流程 > 二次开发 > 通用功能 > 权限管理 > 系统设置 > 设备台帐 > 我的任务 > 接口报错日志 > 系统主题 > 视频管理 >

最近编辑

编辑对象 编辑者 编辑时间

暂无数据

查看前台

点击前台图标可进入前台界面查看，前台系统展示内容可在后台中设置：

点击查看前台



项目操作

卡片展示

项目创建完毕后，支持修改项目，禁用项目，项目备份，和删除项目；



修改项目：修改项目的信息；

禁用项目：项目禁用后则该项目无法登录，数据不会丢失，被禁用的项目还可重启；

备份项目：点击备份项目数据，选择备份内容，点击备份即可，备份的文件存储在项目中台的通用功能>备份管

理菜单下



1. 文件名称：文件名称自动生成，也可自定义。文件名称不可重复。

2. 备份内容：备份内容有媒体文件、前端文件和基础数据。

- 设备免备份文件：设置媒体库备份免备份文件，选中的文件为免备份文件，不会进行备份。
- 基础数据：基础数据选中时，显示具体数据内容，选中的数据才会进行备份

上传备份：通过该入口上传项目备份文件，上传后该备份文件显示在项目【备份与还原】列表中，可进行还原

还原项目：还原项目中的备份文件。点击还原项目，弹出备份列表弹窗，选中项目中已存在的备份文件还原即可



数据库备份项目：对项目数据进行数据库备份，选择备份内容，点击备份即可



数据库上传备份: 通过该入口上传数据库备份文件，上传后显示在【数据库备份】列表中，可进行还原。

数据库还原项目: 还原项目的数据库备份文件。点击数据库还原项目弹出数据库还原项目弹窗，选择列表中已存在的备份文件还原即可。

The screenshot shows a list of database backup projects. At the top, there is a search bar and filter options. Below the header, the list displays the following information:

文件名	备份文件	备份时间	操作
AIRIOT 物联网平台_2024-07-03 16_22_39 (1).zip	file server (媒体文件)	2024-07-05 11:31:13	[还原] [下载] [删除]
AIRIOT 物联网平台_2024-07-03 16_56_26缺少共享自定义.zip	file server (媒体文件)	2024-07-03 16:57:49	[还原] [下载] [删除]
AIRIOT 物联网平台_2024-07-03 16_22_39.zip	file server (媒体文件)	2024-07-03 16:22:59	[还原] [下载] [删除]
AIRIOT 物联网平台_2024-07-03 16_21_52.zip	file server (媒体文件)	2024-07-03 16:22:59	[还原] [下载] [删除]
AIRIOT 物联网平台_2024-07-03 16_08_30.zip	file server (媒体文件)、postgres (基础数据)	2024-07-03 16:15:58	[还原] [下载] [删除]

At the bottom left, there is a button labeled "未选择任何数据". On the right side, there are navigation buttons for the list.

删除项目: 数据会被永久删除，无法恢复。

列表展示

空间项目除了卡片形式展示外，还可以以列表形式展示，操作同卡片展示中的操作

The screenshot shows a software interface for managing database backup projects. At the top, there's a navigation bar with tabs for '项目' (Project), a search bar, and a filter button labeled '更多过滤' (More Filtering). On the right of the bar are buttons for '卡片形式' (Card Style) and '列表形式' (List Style). Below the bar, a header bar displays the title '数据库还原项目' (Database Restore Project) and a close button 'X'. Underneath is a search and filter section with fields for '文件名' (File Name), '备份时间' (Backup Time) with '开始日期' (Start Date) and '结束日期' (End Date) inputs, and a date range selector. There are also '搜索' (Search) and '重置' (Reset) buttons. To the right of this section, it says '共 5 条 - 每页 15 条' (5 items - 15 per page). The main content area is a table with the following data:

文件名	备份文件	备份时间	操作
AIRIOT 物联网平台_2024-07-03 16_22_39 (1).zip	file server (媒体文件)	2024-07-05 11:31:13	[还原] [下载] [删除]
AIRIOT 物联网平台_2024-07-03 16_56_26缺少共享自定义.zip	file server (媒体文件)	2024-07-03 16:57:49	[还原] [下载] [删除]
AIRIOT 物联网平台_2024-07-03 16_22_39.zip	file server (媒体文件)	2024-07-03 16:22:59	[还原] [下载] [删除]
AIRIOT 物联网平台_2024-07-03 16_21_52.zip	file server (媒体文件)	2024-07-03 16:22:59	[还原] [下载] [删除]
AIRIOT 物联网平台_2024-07-03 16_08_30.zip	file server (媒体文件)、postgres (基础数据)	2024-07-03 16:15:58	[还原] [下载] [删除]

At the bottom left, there's a button '未选择任何数据' (No data selected) with a dropdown arrow. At the bottom right, there are navigation buttons for page 1 of 1.

租户管理

管理平台中的租户，已添加的租户可通过租户名称、租户类型、租户行业和租户状态进行搜索。

The screenshot shows a tenant management interface with the following details:

租户名称	租户行业	租户类型	创建时间	租户状态	操作
小测试3	其他	个人	2023-05-31 10:46:36	已启用	[项目] [详情] [禁用] [删除]
小测试2	其他	个人	2023-05-31 09:32:25	已启用	[项目] [详情] [禁用] [删除]
testpass	畜牧业	个人	2023-05-09 14:16:10	已启用	[项目] [详情] [禁用] [删除]
test测试	畜牧业	个人	2023-05-06 17:55:52	已启用	[项目] [详情] [禁用] [删除]
AAA	畜牧业	企业	2023-05-04 14:35:15	已启用	[项目] [详情] [禁用] [删除]
test	畜牧业	个人	2023-04-26 18:26:54	已启用	[项目] [详情] [禁用] [删除]
小测试22	畜牧业	个人	2023-04-26 12:01:50	已启用	[项目] [详情] [禁用] [删除]
testz	其他	个人	2023-04-07 13:34:10	已启用	[项目] [详情] [禁用] [删除]
aaa	油田	企业	2023-01-28 16:34:06	已启用	[项目] [详情] [禁用] [删除]
小测试1	其他	个人	2022-05-20 15:53:48	已启用	[项目] [详情] [禁用] [删除]
jj	畜牧业	个人	2022-04-14 13:56:45	已启用	[项目] [详情] [禁用] [删除]
test11	畜牧业	个人	2022-04-07 16:45:37	已启用	[项目] [详情] [禁用] [删除]
小测试	其他	个人	2022-04-07 16:44:59	已启用	[项目] [详情] [禁用] [删除]

项目：查看租户空间所创建的项目；

详情：查看租户详情；

禁用：禁用后租户无法登录，但租户的数据不会丢失，禁用后的租户还可重启；

修改：修改租户信息；

删除：删除租户；

添加租户

点击“添加租户”按钮，填写租户信息，添加租户，填写完信息后点击保存按钮，返回租户管理界面。

< 添加租户

The screenshot shows a 'Add Tenant' form with the following fields:

- * 租户类型: 个人 企业
- * 租户名称:
- * 租户密码: (眼睛图标)
- * 租户行业: (highlighted with a blue border)
- * 租户项目权限: 请选择租户项目权限
- * 授权:
- 描述:

At the bottom are two buttons: 保存 (Save) and 取消 (Cancel).

租户类型: 租户类型选择个人或企业；

租户名称: 自定义租户名称，登录系统时输入租户名称和租户密码，便可进入该租户空间；

租户密码: 自定义租户密码；

租户行业: 点击选择租户行业；

租户项目权限: 租户项目权限包括不限制、允许选择我创建的模板、允许选择市场中的模板和允许创建空白项目；

授权: 详见 [项目管理](#)授权；

描述: 描述租户信息；

租户空间

登录租户名称和租户密码，进入租户空间，首页记录了该租户空间的基本信息，如项目信息和授权信息。

The screenshot shows the tenant's homepage with the following sections:

- 项目统计**: Displays three metrics: 项目总数 (0), 模板总数 (0), and 禁用总数 (0).
- 授权**: Shows basic authorization categories: 数据采集与控制 (0), 数据分析引擎 (0), 可视化引擎 (0), 业务流引擎 (0), and 二次开发引擎 (0). A detailed view for '时间' (Time) shows creation time (2023-06-05), deadline (2023-06-17), remaining time (12 days), and progress (2.88%).
- 表**: Shows 0 tables with up to 12 rows.
- 最新消息**: Shows a message icon with '暂无数据' (No data available).

点击我的项目，进入项目管理界面，项目管理界面点击添加项目按钮可添加项目，添加方式同项目管理中项目添加方式。

The project management interface includes:

- 搜索**: Includes a search input field, a '禁用' (Disable) checkbox, a '搜索' (Search) button, and a '重置' (Reset) button.
- 显示形式**: Options for '卡片形式' (Card style) and '列表形式' (List style).
- 操作**: Buttons for '添加项目' (Add project), '0条数据' (0 items), '每页24条' (24 items per page), and a refresh icon.
- 消息**: A placeholder message '暂无数据' (No data available) with a document icon.



注意：这里项目授权必须在租户可用授权范围之内。

授权管理

展示空间项目可用功能，以及升级授权

The screenshot shows the 'Authorization Management' interface. On the left sidebar, under 'Project Management', 'Customer Management', and 'Authorization Management' (which is highlighted in blue), there are links for 'System Settings' and 'Help Documents'. The main area is titled 'Authorization Information' and contains the 'Upgrade Authorization' section. It includes three steps: 1. Copy your machine code (604be7220fa124c50793b29621e8666ba9b963). 2. Apply for trial authorization (Enterprise Edition). 3. Upload files. Below these steps is a 'Value-added Services' section with options like 'GIS Components', '3D Components', 'ISC Video Components', 'Hikvision Plugin Video', 'Video Playback (Event Playback)', and 'Mobile Front-end Bundling App', all marked as 'Available'. At the bottom, there's a 'Data Analysis Engine Authorization' section.

升级授权

- 第一步：复制当前空间机器码；
- 第二步：请将机器码及升级内容告知给相关销售人员，由销售人员将升级授权文件发送给您；可通过填写表单或添加客服微信联系到销售人员
- 第三步：将我们提供的升级授权文件上传，即可完成升级授权。

系统设置

基础设置

系统设置包括系统名称设置、系统版权信息设置、系统图片设置和系统背景。

基础设置

* 系统名称: 空间管理
系统版权信息: v1
系统语言: 简体中文
系统图片:
系统背景:
项目图片:
项目背景:
项目背景:
安全策略: 错误次数: 1
冻结时长: 1 分
保存 取消

系统名称: 设置空间系统名称；

系统语言: 平台展示语言设置，默认为简体中文

系统图片: 上传空间的logo图片；

系统背景: 登录系统的背景图片；

项目图片: 上传该空间中建立的项目中的logo，配置后该空间的每一个项目初始化之后都默认使用此logo；

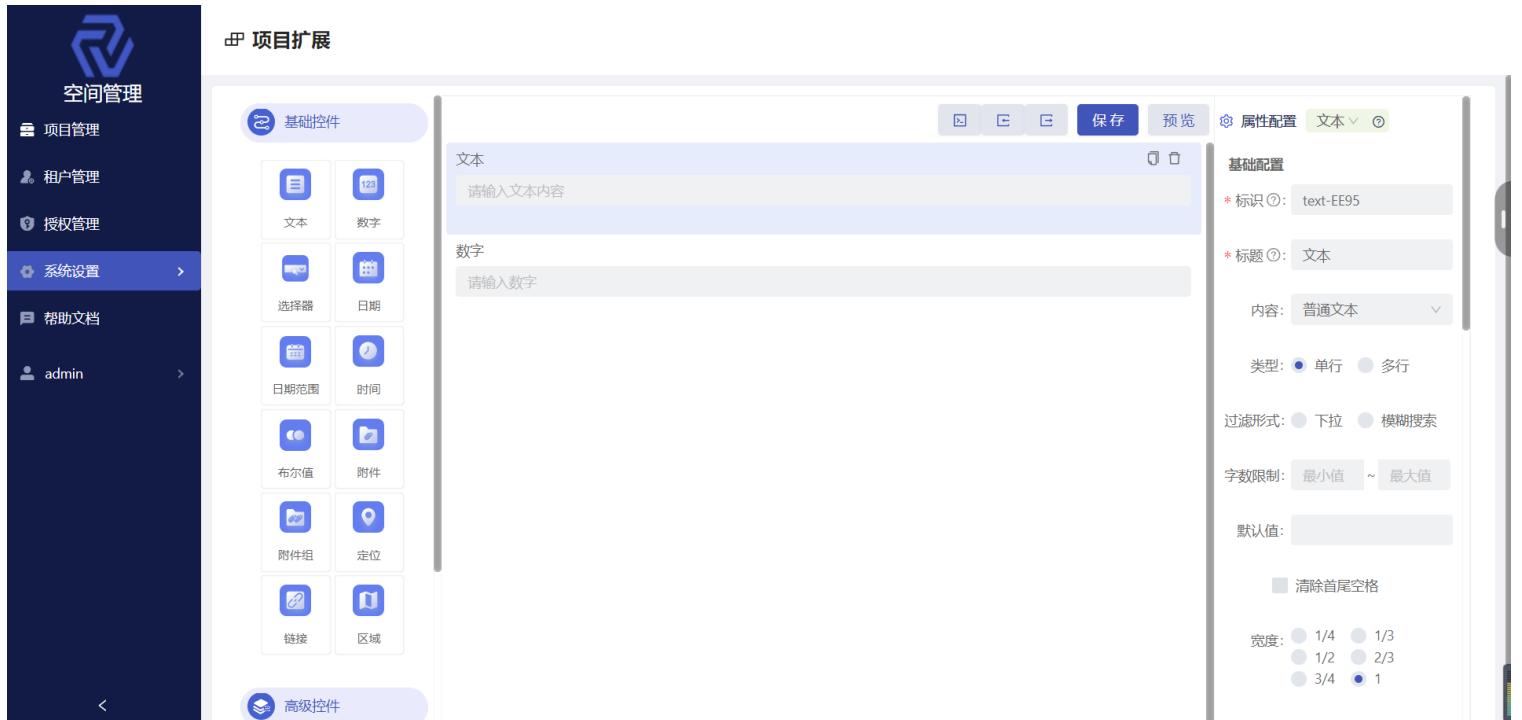
项目背景: 空间中所建立的项目的登录背景，配置后该空间的每一个项目初始化之后都默认使用此背景；

错误次数: 默认不限制，定义次数后，输错限制次数，账号将被冻结，默认冻结1分钟；

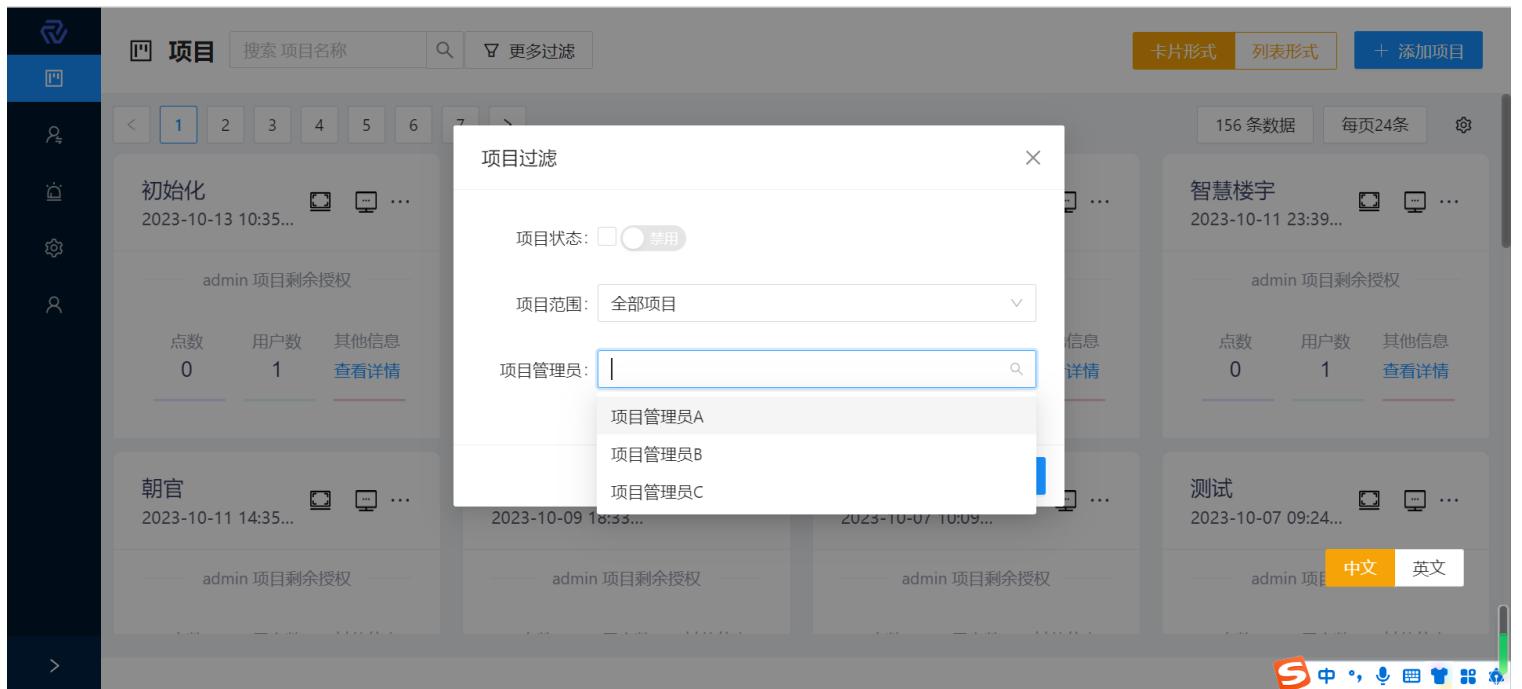
冻结时长: 默认1分钟，用户可定义输错限制次数后的冻结时长；

项目扩展

设置项目信息的扩展字段，当项目表单中的字段信息不满足需求时，可通过项目扩展中新增字段信息，新增的字段信息将在项目表单中展示。如下图，添加项目管理员字段信息，方便知道该项目由谁负责管理



另外，可以在字段的权限中勾选过滤查询中显示，则其会在项目搜索处的更多过滤中作为可选的过滤条件，如下图所示：



控件：包括基础控件、高级控件，鼠标点击控件拖入到编辑区域，添加控件；

属性配置：设置控件的属性；

保存：点击保存后，扩展的字段将同步到项目管理中的新增项目表单和修改项目的表单中；

预览：预览控件的效果；

安装问题

安装过程卡住

情况1：pm2安装或运行失败

打开任务管理器->服务查看pm2.exe服务的状态，若状态为已停止，右键点击启动后观察是否可以正常运行。若pm2.exe服务不存在或者不能正常运行，请联系开发人员。

情况2：etcd初始化失败

1. 打开任务管理器，详细信息，查看etcd.exe进程的PID
2. 打开命令行，输入'netstat -ano | findstr etcd进程的PID'，查看etcd进程监听的端口
3. 文件资源管理器进入安装目录\lib\base\etcd-c3.5.9
4. 打开etcd.conf文件查看listen-client-urls端口是否与第2步etcd所监听的端口一致，若不一致，则
 - 1) 任务管理器右键点击etcd.exe，结束进程
 - 2) 打开命令行，进入安装目录\lib\base\etcd-c3.5.9
 - 3) 输入etcd.exe --config.file=etcd.conf
5. 若不成功，则联系开发人员

情况3：运维服务AIR-OPERATION安装或启动失败

AIR-OPERATION服务安装失败

1. 以管理员身份运行cmd 2. 输入

```
SC CREATE AIR-OPERATION binpath= ""安装目录\lib\operation\operation.exe" -runEnv pm2 -c  
"..\configs/config.toml"" start= auto
```

```
SC START AIR-OPERATION
```

注：将安装目录替换为实际安装平台的目录

AIR-OPERATION无法启动

查看运维服务的日志 位置在安装目录\lib\operation\log\operation.log 1) 日志中出现

```
加载 programs.yml 文件失败, yaml: unmarshal errors
```

需要检测安装目录\lib\operation\programs.yml文件的格式是否正常 2) 日志中出现

```
panic: Fatal error config file: While parsing config: toml: invalid character in comment
```

需要检测安装目录\lib\operation\configs\config.toml文件的格式及编码是否正常

卸载问题

平台卸载完成后安装目录仍有airiot、lib文件文件夹

原因

pm2进程管理程序关闭进程只有3秒的延时，之后便不在处理该进程，若电脑性能不足，可能会有未被关闭的进程残留，导致文件无法删除。

方案1

重启电脑即可手动删除

方案2

打开任务管理器手动结束进程

操作步骤如下：

1.鼠标移动到任务栏点击右键，选择任务管理器



2. 打开详细信息，鼠标移动到标头右键选择列，添加命令行

The screenshot shows the Windows Task Manager window with the '详细信息' (Details) tab selected. A context menu is open over the column headers, specifically over the '命令行' (Command Line) header. The menu items shown are '隐藏列(D)' (Hide Column) and '选择列(S)' (Select Column), with '选择列(S)' highlighted by a red arrow. The main table lists various system processes with their details like PID, status, user, CPU usage, memory usage, and command line arguments.

名称	PID	状态	用户名	CPU	内存使用	命令行
系统中断	-	正在运行	SYSTEM			
系统空闲进程	0	正在运行	SYSTEM			
System	4	正在运行	SYSTEM	03	20 K	
Registry	124	正在运行	SYSTEM	00	16,224 K	
SvpnAgent.exe	268	正在运行	SYSTEM	00	2,192 K	"C:\SSLVPN CLIENT\SvpnAgen"
smss.exe	504	正在运行	SYSTEM	00	340 K	
RuntimeBroker.exe	548	正在运行	Administrat...	00	2,124 K	C:\Windows\System32\Runtime
WUDFHost.exe	588	正在运行	LOCAL SE...	00	1,232 K	"C:\Windows\System32\WUDF
svchost.exe	612	正在运行	SYSTEM	00	10,556 K	C:\WINDOWS\system32\svcho
csrss.exe	692	正在运行	SYSTEM	00	1,100 K	
wininit.exe	800	正在运行	SYSTEM	00	756 K	
csrss.exe	808	正在运行	SYSTEM	00	728 K	
WUDFHost.exe	828	正在运行	LOCAL SE...	00	17,956 K	"C:\Windows\System32\WUDF
services.exe	872	正在运行	SYSTEM	00	5,692 K	
lsass.exe	884	正在运行	SYSTEM	01	5,820 K	C:\WINDOWS\system32\lsass.
fontdrvhost.exe	908	正在运行	UMFD-0	00	1,220 K	"fontdrvhost.exe"
winlogon.exe	960	正在运行	SYSTEM	00	1,292 K	winlogon.exe
fontdrvhost.exe	1016	正在运行	UMFD-1	00	1,596 K	"fontdrvhost.exe"
svchost.exe	1156	正在运行	NETWOR...	00	7,372 K	C:\WINDOWS\system32\svcho
WUDFHost.exe	1172	正在运行	LOCAL SE...	00	1,532 K	"C:\Windows\System32\WUDF"

3.按照命令行排序，找到AIRIOT平台相关服务，如下图，右键结束进程

进程	性能	应用历史记录	启动	用户	详细信息	服务
conhost.exe	6208	正在运行	SYSTEM	00	380 K	\?\C:\WINDOWS\system32\conhost.exe 0x4
conhost.exe	6276	正在运行	SYSTEM	00	64 K	\?\C:\WINDOWS\system32\conhost.exe 0x4
conhost.exe	6376	正在运行	SYSTEM	00	64 K	\?\C:\WINDOWS\system32\conhost.exe 0x4
conhost.exe	7364	正在运行	SYSTEM	00	212 K	\?\C:\WINDOWS\system32\conhost.exe 0x4
conhost.exe	9444	正在运行	SYSTEM	00	244 K	\?\C:\WINDOWS\system32\conhost.exe 0x4
conhost.exe	10944	正在运行	SYSTEM	00	220 K	\?\C:\WINDOWS\system32\conhost.exe 0x4
AggregatorHost.exe	8084	正在运行	SYSTEM	00	1,908 K	AggregatorHost.exe
etcdkeeper.exe	9540	正在运行	SYSTEM	00	156 K	C:\AIRIOT\lib\base\etcdkeeper\etcdkeeper.exe -auth -p 58080
etcd.exe	10284	正在运行	SYSTEM	00	18,088 K	C:\AIRIOT\lib\base\etcd-v3.5.9\etcd.exe --config-file=etcd.conf
influxd.exe	1736	正在运行	SYSTEM	00	82,892 K	C:\AIRIOT\lib\base\influxdb-1.8.3-1\influxd.exe -config influxdb.conf
loki-windows-amd... loki-windows-amd...	11008 10716	正在运行 正在运行	SYSTEM SYSTEM	00 00	37,556 K 33,672 K	C:\AIRIOT\lib\base\loki\loki-windows-amd64.exe -config ./syslogloki/local-config.yaml -server.grpc-max-receive-size=1000000000 C:\AIRIOT\lib\base\loki\loki-windows-amd64.exe -config ./local-config.yaml -server.grpc-max-receive-msg-size=1000000000
mosquitto.exe	9324	正在运行	SYSTEM	00	564 K	C:\AIRIOT\lib\base\mosquitto\mosquitto.exe -c mosquitto.conf
nginx.exe	2064	正在运行	SYSTEM	00	588 K	C:\AIRIOT\lib\base\nginx-1.18.0\nginx.exe -c conf/nginx.conf
nginx.exe	10688	正在运行	SYSTEM	00	84 K	C:\AIRIOT\lib\base\nginx-1.18.0\nginx.exe -c conf/nginx.conf
pg_ctl.exe	4820	正在运行	SYSTEM	00	296 K	C:\AIRIOT\lib\base\pgsql\bin\pg_ctl.exe runservice -D C:\AIRIOT\airiot\db\postgres\pgdata -w
promtail-windows-... promtail-windows-...	9696 8900	正在运行 正在运行	SYSTEM SYSTEM	00 00	75,728 K 11,932 K	C:\AIRIOT\lib\base\promtail\promtail-windows-amd64.exe -config ./syslogpromtail/promtail-config.yaml -log.level=debug C:\AIRIOT\lib\base\promtail\promtail-windows-amd64.exe -config ./promtail-config.yaml -log.level=debug
traefik.exe	9564	正在运行	SYSTEM	00	10,848 K	C:\AIRIOT\lib\base\traefik\traefik.exe
modbus.exe	7408	正在运行	SYSTEM	00	8,084 K	C:\AIRIOT\lib\driver\default-6612500b5e9b0212067135a8-modbus\modbus.exe --serviceId=6612500b5e9b0212067135a8
win-exporter.exe	10976	正在运行	SYSTEM	00	19,380 K	C:\AIRIOT\lib\exporter\win-exporter.exe
node.exe	7752	正在运行	SYSTEM	00	24,072 K	C:\AIRIOT\lib\node\index.js --harmony C:\ProgramData\pm2\service\index.js
operation.exe	7612	正在运行	SYSTEM	00	38,968 K	C:\AIRIOT\lib\operation\operation.exe -runEnv pm2 -c ./configs/config.toml
prometheus.exe	5060	正在运行	SYSTEM	00	50,568 K	C:\AIRIOT\lib\prometheus\prometheus.exe --config-file=prometheus.yaml
algorithm.exe	10632	正在运行	SYSTEM	00	30,280 K	C:\AIRIOT\lib\service\algorithm\algorithm.exe -c ./configs/config.yaml
auth.exe	3360	正在运行	SYSTEM	00	21,780 K	C:\AIRIOT\lib\service\auth\auth.exe -c ./configs/config.yaml
check.exe	10668	正在运行	SYSTEM	00	76,312 K	C:\AIRIOT\lib\service\check\check.exe -c ./configs/config.yaml
computed.exe	10220	正在运行	SYSTEM	00	25,508 K	C:\AIRIOT\lib\service\computed\computed.exe -c ./configs/config.yaml
compute-record.exe	10332	正在运行	SYSTEM	00	31,264 K	C:\AIRIOT\lib\service\compute-record\compute-record.exe -c ./configs/config.yaml
core.exe	23172	正在运行	SYSTEM	00	189,168 K	C:\AIRIOT\lib\service\core\core.exe -c ./configs/config.yaml
data-source.exe	5256	正在运行	SYSTEM	00	32,504 K	C:\AIRIOT\lib\service\data-source\data-source.exe -c ./configs/config.yaml
driver.exe	3636	正在运行	SYSTEM	00	30,468 K	C:\AIRIOT\lib\service\driver\driver.exe -c ./configs/config.yaml
emulator.exe	10232	正在运行	SYSTEM	00	17,244 K	C:\AIRIOT\lib\service\emulator\emulator.exe -c ./configs/config.yaml
flow.exe	1516	正在运行	SYSTEM	00	39,396 K	C:\AIRIOT\lib\service\flow\flow.exe -c ./configs/config.yaml
flow-engine.exe	7508	正在运行	SYSTEM	00	33,036 K	C:\AIRIOT\lib\service\flow-engine\flow-engine.exe -c ./configs/config.yaml
python.exe	10780	正在运行	SYSTEM	00	9,432 K	C:\AIRIOT\lib\service\flow-python-script\python\python.exe ./main.py --config ./configs/config.yaml
flow-schedule.exe	9440	正在运行	SYSTEM	00	25,512 K	C:\AIRIOT\lib\service\flow-schedule\flow-schedule.exe -c ./configs/config.yaml
front.exe	10328	正在运行	SYSTEM	00	15,484 K	C:\AIRIOT\lib\service\front\front.exe -c ./configs/config.yaml
history-computed....	11760	正在运行	SYSTEM	00	21,936 K	C:\AIRIOT\lib\service\history-computed\history-computed.exe -c ./configs/config.yaml
i18n-errcode.exe	3948	正在运行	SYSTEM	00	204 K	C:\AIRIOT\lib\service\i18n-errcode\i18n-errcode.exe -c ./configs/config.yaml
mapping.exe	9432	正在运行	SYSTEM	00	16,024 K	C:\AIRIOT\lib\service\mapping\mapping.exe -c ./configs/config.yaml
online.exe	2164	正在运行	SYSTEM	00	14,608 K	C:\AIRIOT\lib\service\online\online.exe -c ./configs/config.yaml
rule.exe	10488	正在运行	SYSTEM	00	33,372 K	C:\AIRIOT\lib\service\rule\rule.exe -c ./configs/config.yaml
stats.exe	8760	正在运行	SYSTEM	00	15,408 K	C:\AIRIOT\lib\service\stats\stats.exe -c ./configs/config.yaml
storage.exe	11904	正在运行	SYSTEM	00	17,124 K	C:\AIRIOT\lib\service\storage\storage.exe -c ./configs/config.yaml
syslog.exe	10560	正在运行	SYSTEM	00	16,596 K	C:\AIRIOT\lib\service\syslog\syslog.exe -c ./configs/config.yaml
task.exe	8568	正在运行	SYSTEM	00	28,624 K	C:\AIRIOT\lib\service\task\task.exe -c ./configs/config.yaml
task-schedule.exe	18740	正在运行	SYSTEM	00	67,280 K	C:\AIRIOT\lib\service\task-schedule\task-schedule.exe -c ./configs/config.yaml
transfer.exe	1540	正在运行	SYSTEM	00	13,616 K	C:\AIRIOT\lib\service\transfer\transfer.exe -c ./configs/config.yaml
warning.exe	9336	正在运行	SYSTEM	00	43,392 K	C:\AIRIOT\lib\service\warning\warning.exe -c ./configs/config.yaml
ws-data.exe	10392	正在运行	SYSTEM	00	35,408 K	C:\AIRIOT\lib\service(ws-data\ws-data.exe -c ./configs/config.yaml
chrome.exe	23432	正在运行	Administrator	00	1,040 K	C:\Users\Administrator\AppData\Local\Google\Chrome\Application\chrome.exe --type=crashpad-handler "--use
WeChatAppEx.exe	15444	正在运行	Administrator	00	1,412 K	C:\Users\Administrator\AppData\Roaming\Tencent\WeChat\XPlugin\Plugins\RadiusWMF\9129\extracted\runtin
explorer.exe	9140	正在运行	Administrator	00	33,864 K	C:\WINDOWS\Explorer.EXE
TrustedInstaller.exe	17300	正在运行	SYSTEM	00	1,248 K	C:\WINDOWS\servicing\TrustedInstaller.exe
dllhost.exe	7696	正在运行	NETWOR...	00	180 K	C:\WINDOWS\system32\DllHost.exe /Processid:{48DA6741-1BF0-4A44-8325-293086C79077}
dllhost.exe	14960	正在运行	Administrator	00	788 K	C:\WINDOWS\system32\DllHost.exe /Processid:{973D20D7-562D-44B9-B70B-5A0F49CCDF3F}
igfxCUIService.exe	2648	正在运行	SYSTEM	00	904 K	C:\WINDOWS\System32\DriverStore\FileRepository\igd1h64.inf_amd64_2681fc9dedd95cd9\igfxCUIService.exe

端口占用

查看服务日志

打开运维页面<http://127.0.0.1:13030>

查看异常状态的服务日志(鼠标放在异常服务最右侧...处，然后点击日志按钮)

The screenshot shows a list of services in the Operations Management Platform. The services listed include: exporter, flow, flow-engine, flow-opt, flow-schedule, front, i18n-errcode, influx, live, log-handle-node, loki, mapping, mqtt, online, prometheus, promtail, report, rule, stats, storage, sync, syslog, syslogloki, syslogpromtail, task, and task-schedule. The 'influx' service is highlighted with a yellow background and has a red arrow pointing to its log icon. Other services like 'flow', 'flow-engine', and 'flow-schedule' also have red arrows pointing to their log icons. The 'influx' service is currently in an '异常' (Abnormal) state.

接口服务端口被占用

```
bind: Only one usage of each socket address (protocol/network address/port) is normally permitted.
```

日志中包含上面信息的为服务端口被占用

例如core服务端口被占用:

中间件端口被占用

1.web服务

nginx: [emerg] bind() to 0.0.0.0:3030 failed (10013: An attempt was made to access a socket in a way forbidden by its access permissions)

nginx: [emerg] bind() to 0.0.0.0:3030 failed (10013: An attempt was made to access a socket in a way forbidden by its access permissions)

nginx: [emerg] bind() to 0.0.0.0:3030 failed (10013: An attempt was made to access a socket in a way forbidden by its access permissions)

nginx: [emerg] bind() to 0.0.0.0:3030 failed (10013: An attempt was made to access a socket in a way forbidden by its access permissions)

nginx: [emerg] bind() to 0.0.0.0:3030 failed (10013: An attempt was made to access a socket in a way forbidden by its access permissions)

2.etcd服务

```
[{"level": "info", "ts": "2024-05-09T15:10:51.414048+0800", "caller": "etcdmain/config.go:350", "msg": "loaded server configuration, other configuration command line flags and environment variables will be ignored if provided", "path": "etcd.conf"}]
[{"level": "info", "ts": "2024-05-09T15:10:51.414048+0800", "caller": "etcdmain/etcdgo:73", "msg": "Running", "args": "[\"D:\\etcd\\v3.5.9\\etcd.exe\", \"--config-file=etcd.conf\"]"}
[{"level": "info", "ts": "2024-05-09T15:10:51.414048+0800", "caller": "etcdmain/etcd.go:100", "msg": "failed to detect default host", "error": "default host not supported on windows, amd64"}
[{"level": "info", "ts": "2024-05-09T15:10:51.414048+0800", "caller": "etcdmain/etcd.go:116", "msg": "server has been already initialized", "data-dir": "./_ainot/db/etcd", "dir-type": "member"}
[{"level": "warn", "ts": "2024-05-09T15:10:51.414048+0800", "caller": "embeded/etcd.go:673", "msg": "Running http and grpc server on single port. This is not recommended for production."}
[{"level": "info", "ts": "2024-05-09T15:10:51.414048+0800", "caller": "embed/etcd.go:127", "msg": "configuring peer listeners \"listen-peer-urls\" [\"http://127.0.0.1:2380\"]"}
[{"level": "info", "ts": "2024-05-09T15:10:51.414048+0800", "caller": "embed/etcd.go:135", "msg": "configuring client listeners \"listen-client-urls\" [\"http://127.0.0.1:2379\"]"}
[{"level": "info", "ts": "2024-05-09T15:10:51.41456+0800", "caller": "embed/etcd.go:76", "msg": "closing etcd server", "name": "etcd1", "data-dir": "./_ainot/db/etcd", "advertise-peer-urls": ["http://localhost:2380"], "advertise-client-urls": ["http://localhost:2379"]}
[{"level": "info", "ts": "2024-05-09T15:10:51.415073+0800", "caller": "embed/etcd.go:78", "msg": "closing etcd server", "name": "etcd2", "data-dir": "./_ainot/db/etcd", "advertise-peer-urls": ["http://localhost:2380"], "advertise-client-urls": ["http://localhost:2379"]}
[{"level": "fatal", "ts": "2024-05-09T15:10:51.415073+0800", "caller": "etcdmain/etcd.go:204", "msg": "discovery failed", "error": "listen tcp 127.0.0.1:2379 bind: Only one usage of each socket address (protocol/network address/port) is normally permitted.", "stacktrace": "go.etcd.io/etcd/server/v3/etcdmain.startEtcdOrProxyV2\nntgo.etcd.io/etcd/server/v3/etcdmain.main\nMain\nntgo.etcd.io/etcd/server/v3/etcdmain.main.go:40\nnmain.main\nntgo.etcd.io/etcd/server/v3/main.go:31\nruntime.main\nn\\truntime.proc.go:250"}]
```

3.influx服务

```
ts=2024-05-09T07:40:54.705429Z lvl=info msg="Registered diagnostics client" log_id=0p34bgc0000 service=monitor name=runtime
ts=2024-05-09T07:40:54.705429Z lvl=info msg="Registered diagnostics client" log_id=0p34bgc0000 service=monitor name=network
ts=2024-05-09T07:40:54.705429Z lvl=info msg="Registered diagnostics client" log_id=0p34bgc0000 service=monitor name=system
ts=2024-05-09T07:40:54.705429Z lvl=info msg="Starting precreation service" log_id=0p34bgc0000 service=shard-precreation check_interval=10m advance_period=30m
ts=2024-05-09T07:40:54.705429Z lvl=info msg="Starting snapshot service" log_id=0p34bgc0000 service=snapshot
ts=2024-05-09T07:40:54.705429Z lvl=info msg="Starting continuous query service" log_id=0p34bgc0000 service=continuous_querier
ts=2024-05-09T07:40:54.705429Z lvl=info msg="Storing statistics" log_id=0p34bgc0000 service=monitor db_instance=_internal db_rp=monitor interval=10s
ts=2024-05-09T07:40:54.705429Z lvl=info msg="Starting HTTP service" log_id=0p34bgc0000 service=httpd authentication=true
ts=2024-05-09T07:40:54.705429Z lvl=info msg="opened HTTP access log" log_id=0p34bgc0000 service=httpd path=stderr
ts=2024-05-09T07:40:54.705429Z lvl=info msg="Auth is enabled but shared_secret is blank. RearrangeAuthentication is disabled." log_id=0p34bgc0000 service=httpd
run: open server: open service: listen tcp 127.0.0.1:8086: bind: Only one usage of each socket address (protocol/network address/port) is normally permitted.
```

解决方法

1.关闭占用端口的进程

- 1) 键入'win'+'r', 输入cmd, 回车, 打开命令行
- 2) 查看占用服务端口的进程PID

```
netstat -ano | findstr 被占用的端口号
```

- 3) 关闭进程

```
kill -9 步骤2查到的PID
```

2.修改平台服务的端口

注：若占用平台服务端口的进程正在使用，不可关闭，则需要修改平台服务的端口

修改接口服务的端口

以core服务为例

- 1) 修改接口服务配置文件，位置在 安装目录\lib\service\core\configs\config.toml
- 2) core服务占用了两个端口，分别为GRPC端口9223, httpserver端口59015，按照实际情况修改即可

```
config.yaml
1  App:
2    Etcd:
3      DialTimeout: 10
4      Endpoints:
5        - 127.0.0.1:2379
6      Password: ENC(ZGVsbDEyMw==)
7      Username: root
8      EtcdConfig: /airiot/config/pro.json
9    Server:
10      GRPC:
11        Port: 9223
12      HTTP:
13        Port: 59015
14      Metadata:
15        env: local
16      ServerName: core
17    Backup: ./backup
18    CryptoType: ''
19    FileServer: ./fileServer
20    FileServerType: local
21    Html: ../front/html
22    LicenseDir: ./config
23    MaxFreeDiskSpaceForBackup: 10
24    MediaLibraryPath: mediaLibrary
25    SVG: ./svg
26    SystemDefaultMediaLibrary: systemDefaultMediaLibrary
27    SystemDefaultMediaLibraryMod: false
28    UseLicenseRefreshInterval: 60
```

3) 在运维页面重启core服务

The screenshot shows a service management interface with the following details:

- Service Management tab is active.
- Search bar contains "core".
- Action buttons: 开始 (Start), 停止 (Stop), 重启 (Restart, highlighted with a red arrow), 删除 (Delete), 批量升级 (Batch Upgrade).
- Table view:

服务名称	状态	当前版本
core	运行	v4.3.59

修改etcd服务的端口

几乎所有的接口服务都依赖etcd服务，所以如果etcd端口被占用，建议将数据备份然后将平台卸载重装。

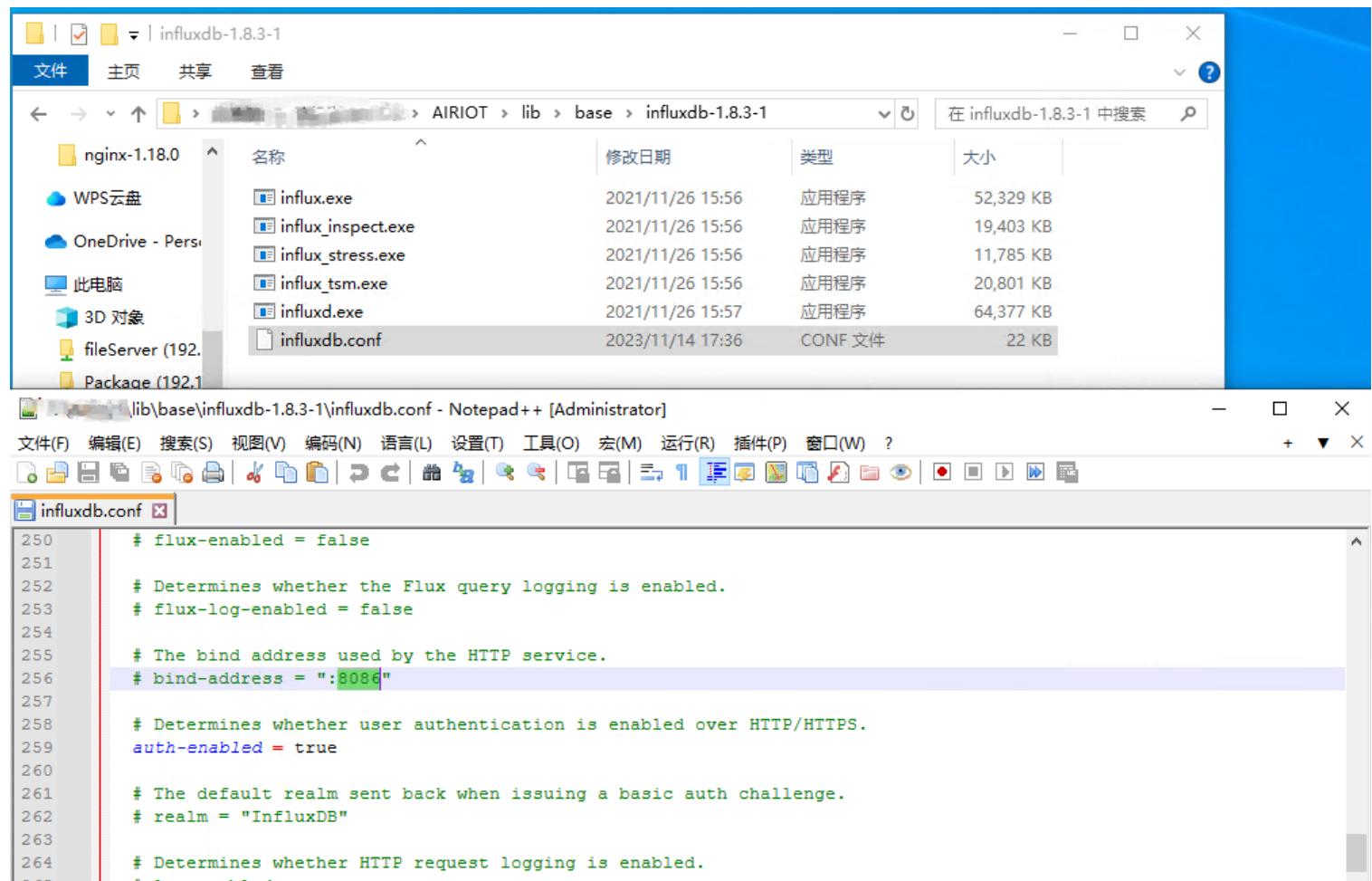
操作步骤：

注：如果平台是刚刚安装或者不需要备份，可以跳过第2、7步

- 1) 打开文件资源管理器，进入安装目录；
- 2) 复制airiot文件夹到其他目录；
- 3) 在安装目录下双击uninst.exe卸载平台；
- 4) 双击安装包重新安装平台；
- 5) 管理员身份打开命令行，输入'pm2 kill'；
- 6) 打开任务管理器，找到air-redis、AIR-PGSQL两个服务，右键停止；
- 7) 将备份的airiot文件夹还原到安装目录，直接覆盖即可；
- 8) 打开任务管理器，分别启动air-redis、AIR-PGSQL、AIR-OPERATION三个服务。

修改influx服务的端口

1. 打开influx配置文件(安装目录/lib/base/influx-1.8.3-1/influx.conf)，如图：



2. 修改8086端口为其他空闲端口，并保存配置文件

3. 浏览器打开<http://127.0.0.1:58080>, 点击etcdkeeper, 输入密码(咨询客服)

4.点击左侧airiot/config/pro.json项，修改influx的配置，如图，将8086修改为第2步相同的空闲端口：

```
1  {
2      "App": {
3          "RunMode": "release",
4          "PrintConfig": true,
5          "Swagger": true,
6          "Platform": "Multiple",
7          "Pprof": {
8              "Enable": true,
9              "Prefix": "",
10             "Port": 9000
11         },
12         "Gorm": {
13             "Debug": false,
14             "LogLevel": 3,
15             "DBType": "postgres",
16             "MaxLifetime": 3600,
17             "MaxOpenConns": 20,
18             "MaxIdleConns": 10,
19             "TablePrefix": "",
20             "EnableAutoMigrate": true,
21             "Postgres": {
22                 "Host": "127.0.0.1",
23                 "Port": 5432,
24                 "User": "root",
25                 "Password": "ENC(ZGVsbDEyMw==)",
26                 "DBName": "postgres",
27                 "SSLMODE": "disable"
28             }
29         },
30         "TSDB": {
31             "DBType": "influx",
32             "DBName": "tsdb",
33             "Timeout": 5,
34             "MaxConn": 50,
35             "Influx": {
36                 "Protocol": "HTTP",
37                 "Addr": "http://127.0.0.1:8086",
38                 "Username": "admin",
39                 "Password": "ENC(ZGVsbDEyMw==)",
40                 "Sharding": true,
41                 "CreateDatabaseSql": "create database \"%$\""
42             }
43         },
44         "Redis": {
45             "Type": "",
46             "Addrs": [
47                 "127.0.0.1:6379"
48             ],
49             "Password": "ENC(ZGVsbDEyMw==)",
50             "PoolSize": 10
51         }
52     }
53 }
```

5.打开运维的programs.yml文件(安装目录/lib/operation/programs.yml)，修改influx服务的端口为第2步相同的空闲端口

文件夹操作界面显示了文件夹结构和文件列表。

	名称	修改日期	类型	大小
nginx-1.18.0	build	2024/4/7 9:59	文件夹	
WPS 云盘	configs	2024/4/7 9:59	文件夹	
OneDrive - Perso	data	2024/4/7 10:02	文件夹	
此电脑	driverRepo	2024/4/7 15:49	文件夹	
3D 对象	log	2024/4/7 10:02	文件夹	
fileServer (192.	operation.exe	2024/3/11 10:14	应用程序	26,701 KB
Package (192.1	programs.yml	2024/4/15 17:57	Yaml 源文件	11 KB
视频	programs.yml.bak	2024/5/15 17:37	BAK 文件	11 KB
图片				

打开 Notepad++ 显示了 programs.yml 内容，高亮了与本题相关的部分。

```

246     repo: log-handle-node
247     pm2: {}
248     - namespace: base
249       name: influx
250       command: influxd.exe -config influxdb.conf
251       environ: []
252       directory: ./lib/base/influxdb-1.8.3-1
253       ports:
254         - 8086
255         - 8089
256       start_auto: true
257       start_retries: -1

```

6.重启AIR-OPERATION服务

任务管理器的服务视图显示了正在运行的 AIR-OPERATION 服务。

名称	PID	描述	状态	组
AarSvc		Agent Activation...	已停止	AarSvcGr...
AarSvc_201b49b2f		Agent Activation...	已停止	AarSvcGr...
AIR-OPERATION	7612	AIR-OPERATION	正在运行	

右侧菜单栏显示了服务的操作按钮：开始(S)、停止(T)、重新启动(R)、打开服务(V)、在线搜索(O)、转到详细信息(D)。

7.管理员身份打开cmd，输入pm2 kill

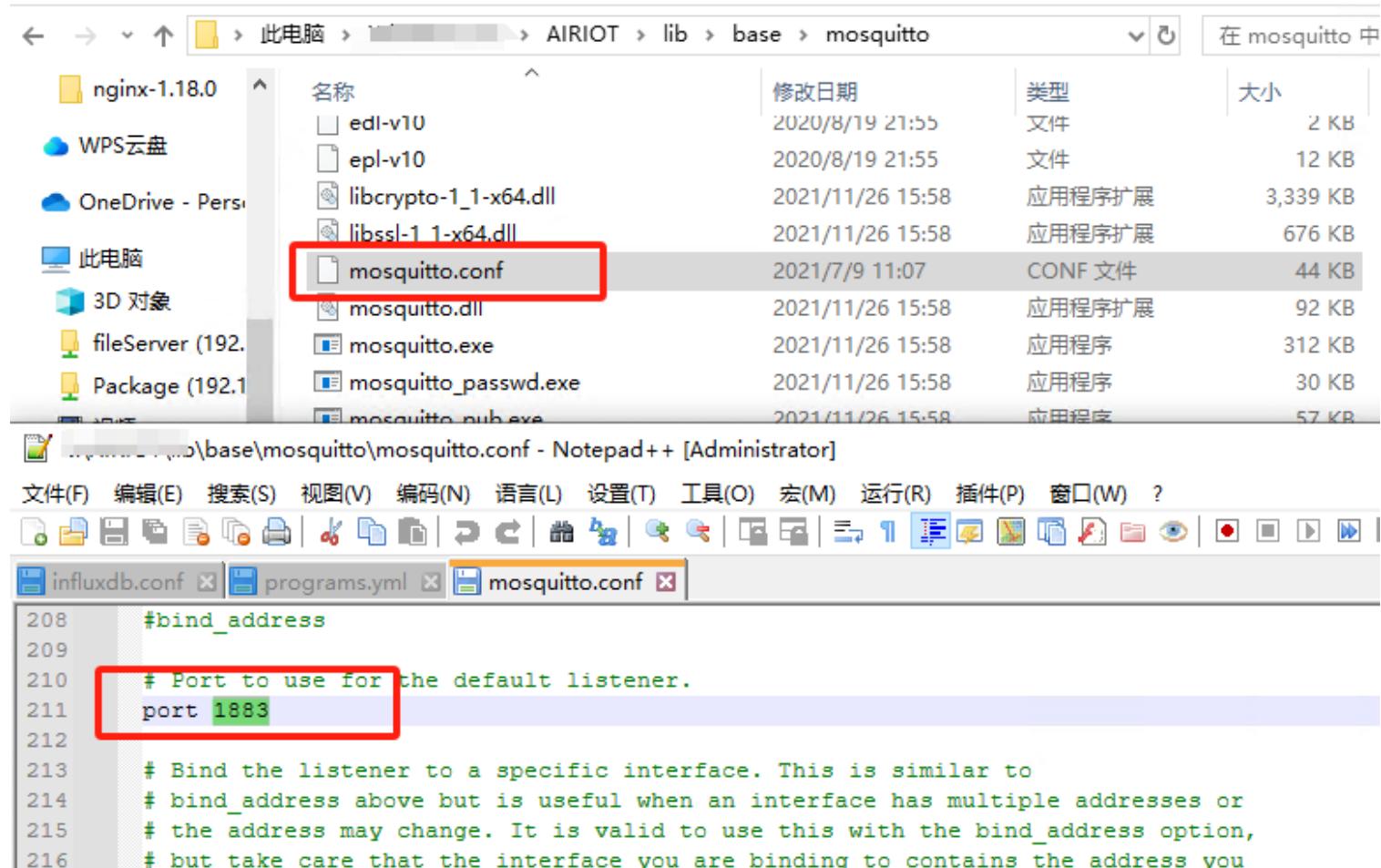
```
Administrator: C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.19045.4412]
(c) Microsoft Corporation。保留所有权利。
C:\Users\Administrator>pm2 kill
[PM2] Applying action deleteProcessId on app [all](ids: [
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
  10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
  20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
  30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
  40, 41, 42
])
[PM2] [@jessety/pm2-logrotate:pm2-logrotate](0) ~
[PM2] [core](1) ~
[PM2] [warning](2) ~
[PM2] [task](3) ~
[PM2] [flow-python-script](4) ~
[PM2] [flow-opc](5) ~
[PM2] [syslogpromtail](7) ~
[PM2] [ws-data](6) ~
[PM2] [syslog](8) ~
[PM2] [driver](9) ~
[PM2] [mqtt](10) ~
[PM2] [etcdkeeper](11) ~
[PM2] [auth](12) ~
[PM2] [log-handle-node](22) ~
[PM2] [algorithm](13) ~
[PM2] [transfer](14) ~
[PM2] [task-schedule](15) ~
[PM2] [computed](16) ~
[PM2] [check](17) ~
[PM2] [promtail](18) ~
[PM2] [mapping](19) ~
[PM2] [emulator](20) ~
[PM2] [stats](21) ~
[PM2] [history-computed](23) ~
[PM2] [traefik](25) ~
[PM2] [flow-schedule](24) ~
[PM2] [compute-record](26) ~
[PM2] [rule](27) ~
[PM2] [loki](28) ~
[PM2] [exporter](29) ~
[PM2] [etcd](30) ~
[PM2] [storage](31) ~
[PM2] [online](32) ~
[PM2] [i18n-errcode](33) ~
[PM2] [flow](34) ~
[PM2] [syslogloki](35) ~
[PM2] [front](36) ~
[PM2] [flow-engine](37) ~
[PM2] [web](38) ~
[PM2] [default-6612500b5e9b0212067135a8-modbus](39) ~
[PM2] [data-source](40) ~
[PM2] [prometheus](41) ~
[PM2] [influx](42) ~
[PM2] [v] All Applications Stopped
[PM2] [v] PM2 Daemon Stopped
```

8. 打开任务管理器，选择服务，重启AIR-OPERATION服务



修改mqtt服务的端口

1. 打开mqtt配置文件(安装目录/lib/base/mosquitto/mosquitto.conf)，如图：



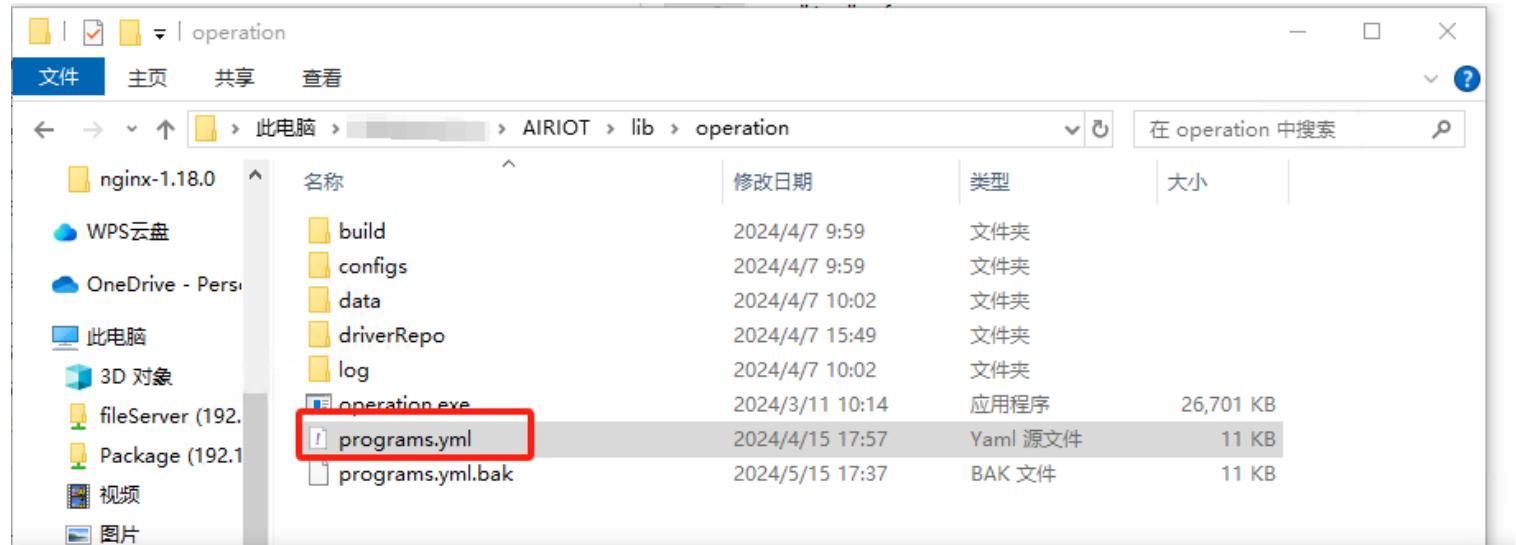
2. 修改1883端口为其他空闲端口，并保存配置文件

3. 浏览器打开<http://127.0.0.1:58080>, 点击etcdkeeper, 输入密码(咨询客服)

4.点击左侧airiot/config/pro.json项，修改mqtt的配置，如图，将1883修改为第2步相同的空闲端口：

```
33     "Timeout": 5,
34     "MaxConn": 50,
35     "Influx": {
36         "Protocol": "HTTP",
37         "Addr": "http://127.0.0.1:8086",
38         "Username": "admin",
39         "Password": "ENC(ZGVsbDEyMw==)",
40         "Sharding": true,
41         "CreateDatabaseSql": "create database \"%s\""
42     },
43     "Redis": {
44         "Type": "",
45         "Addrs": [
46             "127.0.0.1:6379"
47         ],
48         "Password": "ENC(ZGVsbDEyMw==)",
49         "PoolSize": 10
50     },
51     "MQ": {
52         "MQType": "mqtt",
53         "TopicPrefix": "data/",
54         "MQTT": {
55             "Host": "127.0.0.1",
56             "Port": 1883,
57             "Username": "admin",
58             "Password": "ENC(cHViBGlj)",
59             "Order": true
60         },
61         "Flow": {
62             "Host": "127.0.0.1",
63             "Port": 2333,
64             "WriteTimeout": 60
65         },
66         "CORS": {
67             "Enable": false,
68             "AllowOrigins": [
69                 "*"
70             ],
71             "AllowMethods": [
72                 "GET", "POST", "PUT", "DELETE", "OPTIONS"
73             ]
74         }
75     }
76 }
```

5.打开运维的programs.yml文件(安装目录/lib/operation/programs.yml)，修改mqtt服务的端口为第2步相同的空闲端口

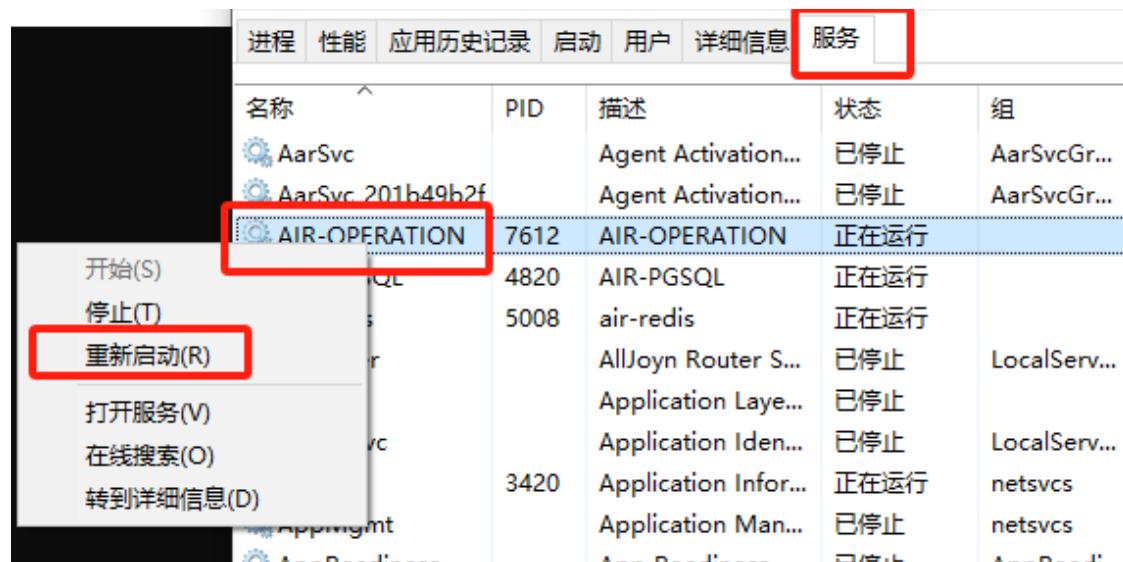


```

196     repo: online
197     pm2: {}
198   - namespace: base
199     name: mqtt
200     command: mosquitto.exe -c mosquitto.conf
201     environ: []
202     directory: ./lib/base/mosquitto
203     ports:
204       - 1883
205     start_auto: true
206     start_retries: -1
207     start_seconds: 3
208     stop_timeout: 3
209     version: 1.6.12
210     repo: mqtt
211     pm2: {}

```

6.重启AIR-OPERATION服务



7.管理员身份打开cmd，输入pm2 kill

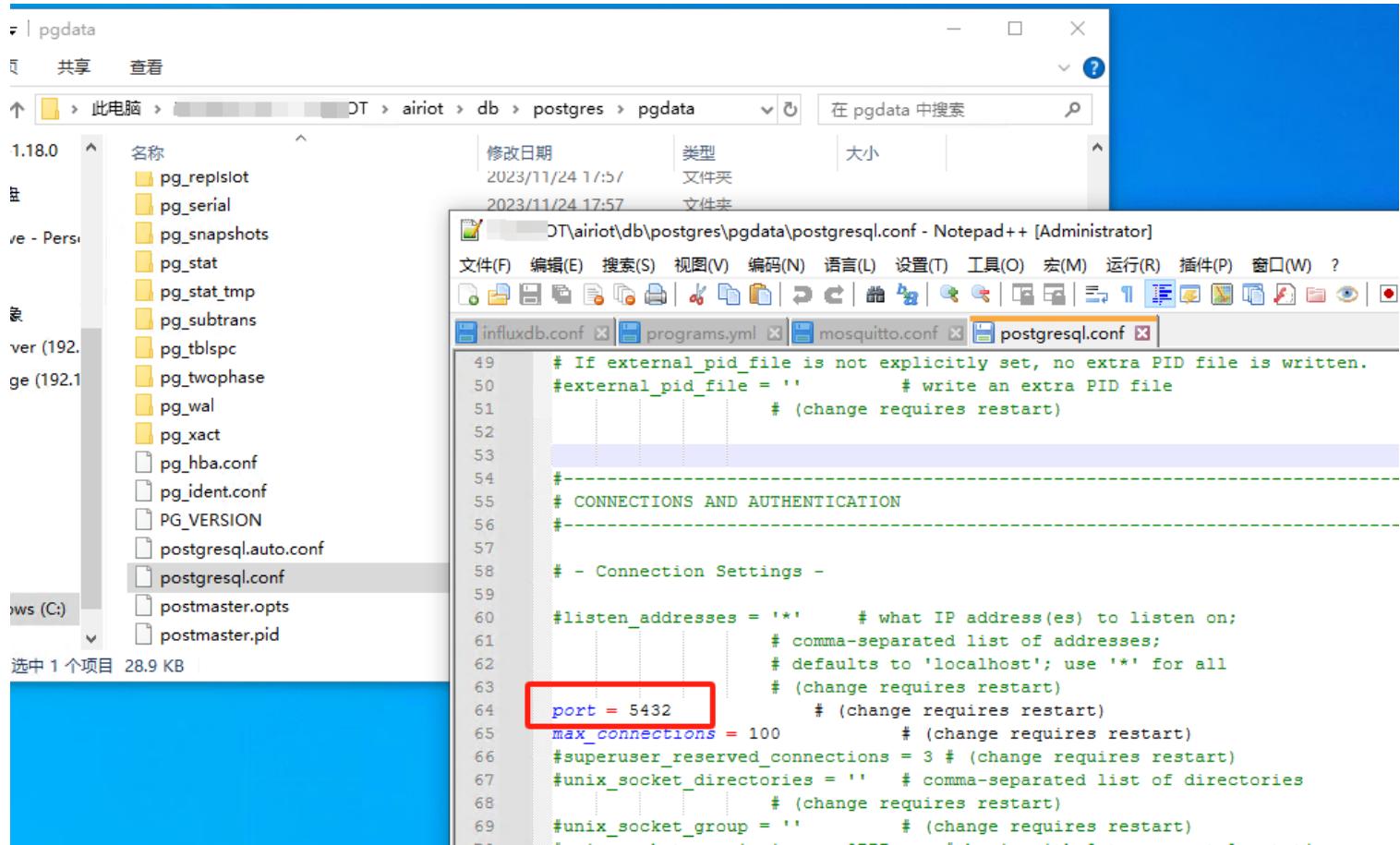
```
Administrator: C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.19045.4412]
(c) Microsoft Corporation。保留所有权利。
C:\Users\Administrator>pm2 kill
[PM2] Applying action deleteProcessId on app [all](ids: [
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
  10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
  20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
  30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
  40, 41, 42
])
[PM2] [@jessety/pm2-logrotate:pm2-logrotate](0) ~
[PM2] [core](1) ~
[PM2] [warning](2) ~
[PM2] [task](3) ~
[PM2] [flow-python-script](4) ~
[PM2] [flow-opc](5) ~
[PM2] [syslogpromtail](7) ~
[PM2] [ws-data](6) ~
[PM2] [syslog](8) ~
[PM2] [driver](9) ~
[PM2] [mqtt](10) ~
[PM2] [etcdkeeper](11) ~
[PM2] [auth](12) ~
[PM2] [log-handle-node](22) ~
[PM2] [algorithm](13) ~
[PM2] [transfer](14) ~
[PM2] [task-schedule](15) ~
[PM2] [computed](16) ~
[PM2] [check](17) ~
[PM2] [promtail](18) ~
[PM2] [mapping](19) ~
[PM2] [emulator](20) ~
[PM2] [stats](21) ~
[PM2] [history-computed](23) ~
[PM2] [traefik](25) ~
[PM2] [flow-schedule](24) ~
[PM2] [compute-record](26) ~
[PM2] [rule](27) ~
[PM2] [loki](28) ~
[PM2] [exporter](29) ~
[PM2] [etcd](30) ~
[PM2] [storage](31) ~
[PM2] [online](32) ~
[PM2] [i18n-errcode](33) ~
[PM2] [flow](34) ~
[PM2] [syslogloki](35) ~
[PM2] [front](36) ~
[PM2] [flow-engine](37) ~
[PM2] [web](38) ~
[PM2] [default-6612500b5e9b0212067135a8-modbus](39) ~
[PM2] [data-source](40) ~
[PM2] [prometheus](41) ~
[PM2] [influx](42) ~
[PM2] [v] All Applications Stopped
[PM2] [v] PM2 Daemon Stopped
```

8. 打开任务管理器，选择服务，重启AIR-OPERATION服务



修改pgsql服务的端口

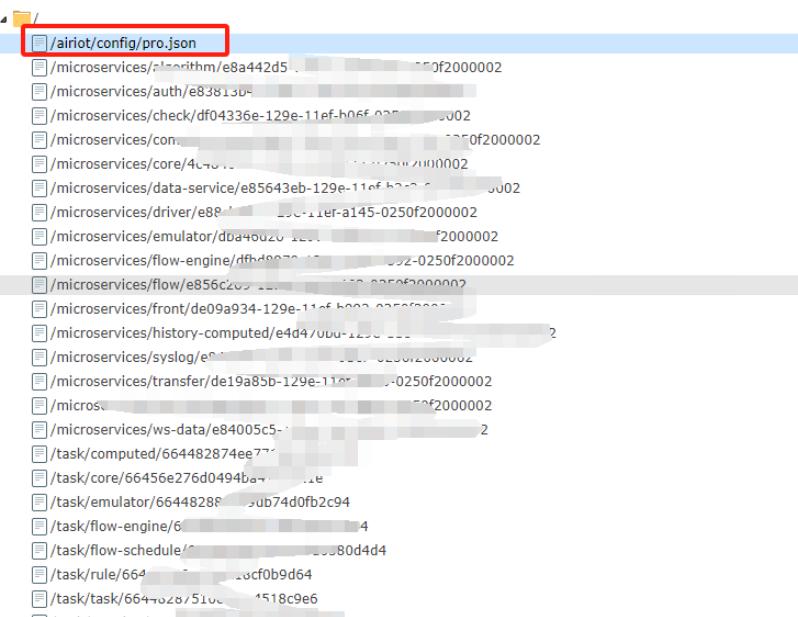
1. 打开pgsql配置文件(安装目录/lib/airiot/db/postgres/pgdata/postgres.conf)，如图：



2. 修改5432端口为其他空闲端口，并保存配置文件

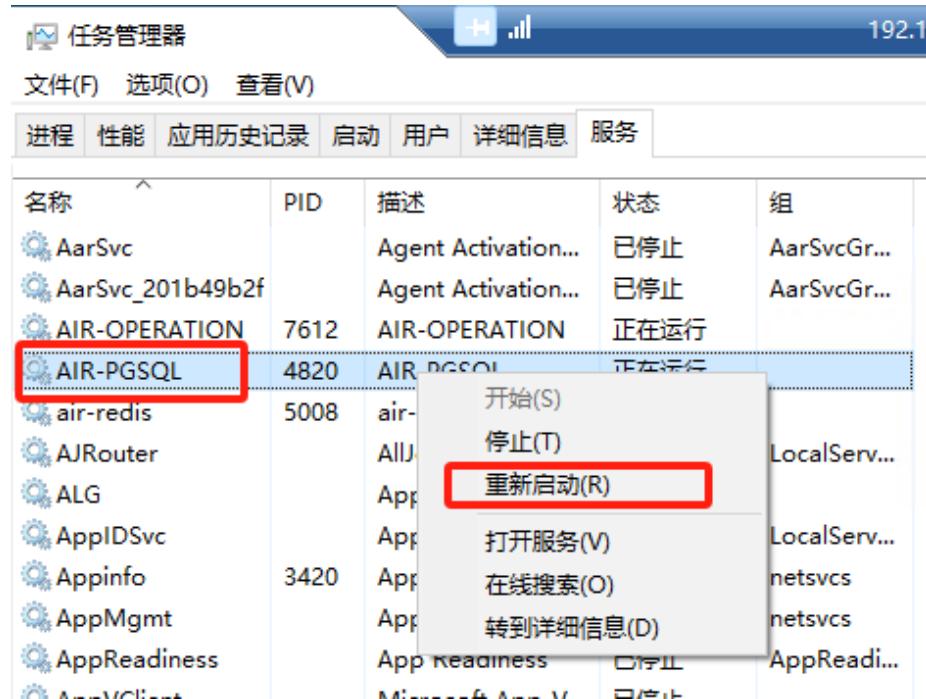
3. 浏览器打开<http://127.0.0.1:58080>, 点击etcdkeeper, 输入密码(咨询客服)

4.点击左侧airiot/config/pro.json项，修改pgsql的配置，如图，将5432修改为第2步相同的空闲端口：



```
1  {
2      "App": {
3          "RunMode": "release",
4          "PrintConfig": true,
5          "Swagger": true,
6          "Platform": "Multiple",
7          "Pprof": {
8              "Enable": true,
9              "Prefix": "",
10             "Port": 9000
11         },
12         "Gorm": {
13             "Debug": false,
14             "LogLevel": 3,
15             "DBType": "postgres",
16             "MaxLifetime": 3600,
17             "MaxOpenConns": 20,
18             "MaxIdleConns": 10,
19             "TablePrefix": "",
20             "EnableAutoMigrate": true,
21             "Postgres": {
22                 "Host": "127.0.0.1",
23                 "Port": 5432, // This line is highlighted
24                 "User": "root",
25                 "Password": "ENC(ZGvbDEyMw==)",
26                 "DBName": "postgres",
27                 "SSLMode": "disable"
28             }
29         },
30         "TSDB": {
31             "DBType": "influx",
32             "DBName": "tsdb",
33             "Timeout": 5,
34             "MaxConn": 50,
35             "Influx": {
36                 "Protocol": "HTTP",
37                 "Addr": "http://127.0.0.1:8086",
38                 "Username": "admin",
39                 "Password": "ENC(ZGvbDEyMw==)",
40                 "Sharding": true,
41                 "CreateDatabaseSql": "create database \"%s\""
42             }
43         },
44         "Redis": {
45             "Tunne...": ...
46         }
47     }
48 }
```

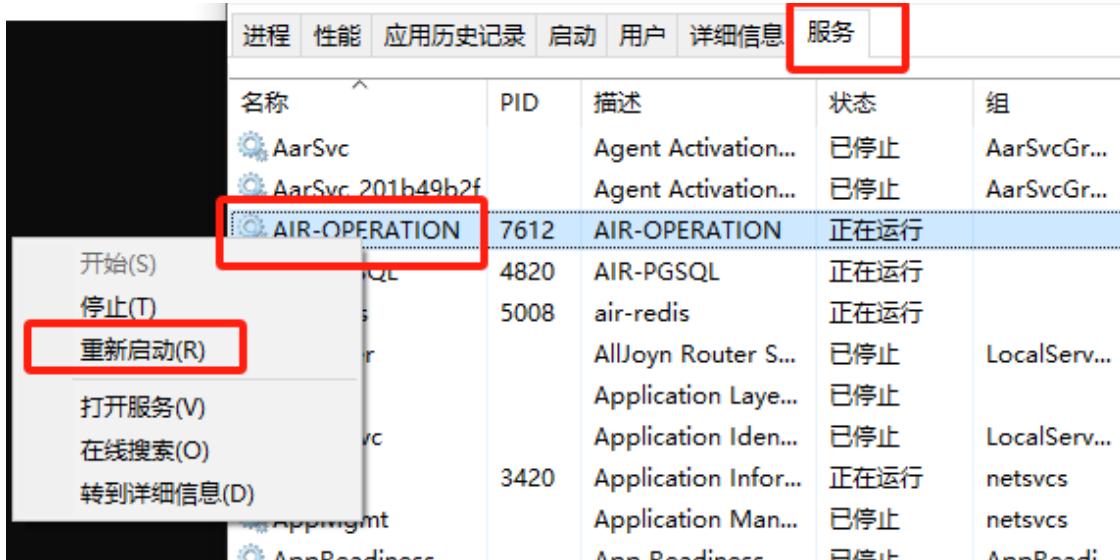
5.打开任务管理器，选择服务，重启AIR-PGSQL服务



6.管理员身份打开cmd，输入pm2 kill

```
Administrator: C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.19045.4412]
(c) Microsoft Corporation。保留所有权利。
C:\Users\Administrator>pm2 kill
[PM2] Applying action deleteProcessId on app [all](ids: [
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
  10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
  20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
  30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
  40, 41, 42
])
[PM2] [@jessety/pm2-logrotate:pm2-logrotate](0) ~
[PM2] [core](1) ~
[PM2] [warning](2) ~
[PM2] [task](3) ~
[PM2] [flow-python-script](4) ~
[PM2] [flow-opc](5) ~
[PM2] [syslogpromtail](7) ~
[PM2] [ws-data](6) ~
[PM2] [syslog](8) ~
[PM2] [driver](9) ~
[PM2] [mqtt](10) ~
[PM2] [etcdkeeper](11) ~
[PM2] [auth](12) ~
[PM2] [log-handle-node](22) ~
[PM2] [algorithm](13) ~
[PM2] [transfer](14) ~
[PM2] [task-schedule](15) ~
[PM2] [computed](16) ~
[PM2] [check](17) ~
[PM2] [promtail](18) ~
[PM2] [mapping](19) ~
[PM2] [emulator](20) ~
[PM2] [stats](21) ~
[PM2] [history-computed](23) ~
[PM2] [traefik](25) ~
[PM2] [flow-schedule](24) ~
[PM2] [compute-record](26) ~
[PM2] [rule](27) ~
[PM2] [loki](28) ~
[PM2] [exporter](29) ~
[PM2] [etcd](30) ~
[PM2] [storage](31) ~
[PM2] [online](32) ~
[PM2] [i18n-errcode](33) ~
[PM2] [flow](34) ~
[PM2] [syslogloki](35) ~
[PM2] [front](36) ~
[PM2] [flow-engine](37) ~
[PM2] [web](38) ~
[PM2] [default-6612500b5e9b0212067135a8-modbus](39) ~
[PM2] [data-source](40) ~
[PM2] [prometheus](41) ~
[PM2] [influx](42) ~
[PM2] [v] All Applications Stopped
[PM2] [v] PM2 Daemon Stopped
```

7. 打开任务管理器，选择服务，重启AIR-OPERATION服务



修改redis服务的端口

1. 管理员身份打开cmd，停止并删除air-redis服务

```
SC STOP air-redis  
SC DELETE air-redis
```

2. 创建新的服务，启动指令修改redis端口6379为空闲端口

```
# 将安装目录替换为平台的安装目录，例如安装在D:/AIRIOT，则将‘安装目录’替换为D:/AIRIOT  
# 将新的空闲端口替换  
# 将${NEW_PASSWORD}替换为redis的实际密码，在安装目录下install.bat脚本中可以查到  
"安装目录\lib\base\redis-latest\redis-server" --service-install --service-name AIR-REDIS --  
loglevel verbose --logfile "安装目录\airiot\db\redisdb\redis.log" --bind 127.0.0.1 --port 新的  
空闲端口 --requirepass ${NEW_PASSWORD} --appendonly yes  
  
SC DESCRIPTION AIR-REDIS AIRIOT平台Redis数据库  
  
SC START AIR-REDIS
```

3. 浏览器打开<http://127.0.0.1:58080>，点击etcdkeeper，输入密码(咨询客服)

4.点击左侧airiot/config/pro.json项，修改redis的配置，如图，将2379修改为第2步相同的空闲端口：

ETCD Keeper v3

127.0.0.1:2379

```
» /airiot/config/pro.json

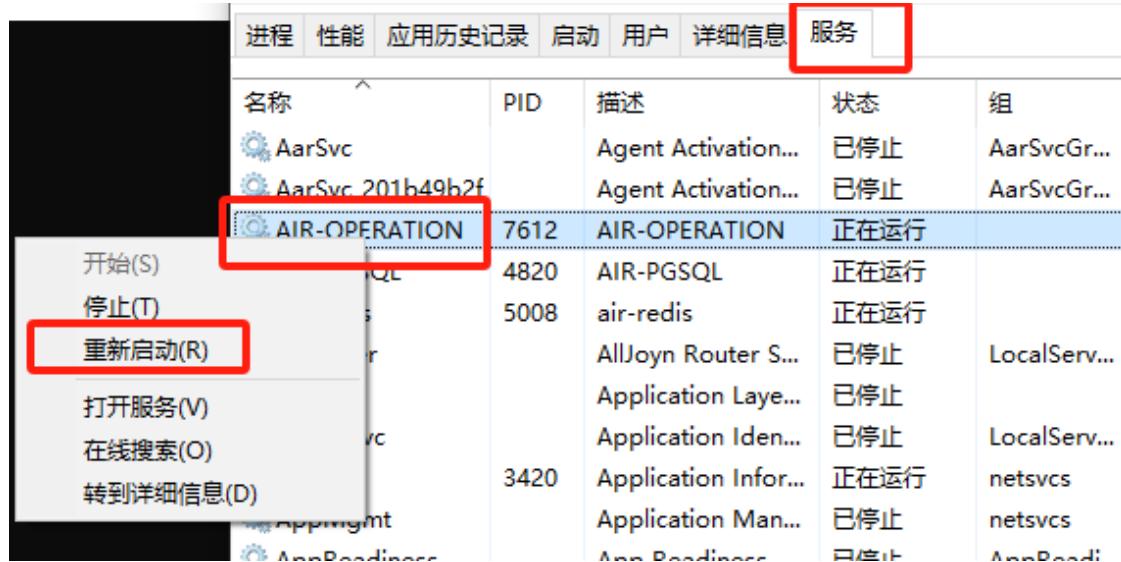
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

    "Timeout": 5,
    "MaxConn": 50,
    "Influx": {
        "Protocol": "HTTP",
        "Addr": "http://127.0.0.1:8086",
        "Username": "admin",
        "Password": "ENC(ZGVsbDEyMw==)",
        "Sharding": true,
        "CreateDatabaseSql": "create database \"%s\""
    },
    "Redis": {
        "Type": "",
        "Addrs": [
            "127.0.0.1:6379"
        ],
        "Password": "ENC(ZGVsbDEyMw==)",
        "PoolSize": 10
    },
    "MQ": {
        "MQType": "mqtt",
        "TopicPrefix": "data/",
        "MQTT": {
            "Host": "127.0.0.1",
            "Port": 1883
        }
    }
}
```

5.管理员身份打开cmd，输入pm2 kill

```
Administrator: C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.19045.4412]
(c) Microsoft Corporation。保留所有权利。
C:\Users\Administrator>pm2 kill
[PM2] Applying action deleteProcessId on app [all](ids: [
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
  10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
  20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
  30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
  40, 41, 42
])
[PM2] [@jessety/pm2-logrotate:pm2-logrotate](0) ~
[PM2] [core](1) ~
[PM2] [warning](2) ~
[PM2] [task](3) ~
[PM2] [flow-python-script](4) ~
[PM2] [flow-opc](5) ~
[PM2] [syslogpromtail](7) ~
[PM2] [ws-data](6) ~
[PM2] [syslog](8) ~
[PM2] [driver](9) ~
[PM2] [mqtt](10) ~
[PM2] [etcdkeeper](11) ~
[PM2] [auth](12) ~
[PM2] [log-handle-node](22) ~
[PM2] [algorithm](13) ~
[PM2] [transfer](14) ~
[PM2] [task-schedule](15) ~
[PM2] [computed](16) ~
[PM2] [check](17) ~
[PM2] [promtail](18) ~
[PM2] [mapping](19) ~
[PM2] [emulator](20) ~
[PM2] [stats](21) ~
[PM2] [history-computed](23) ~
[PM2] [traefik](25) ~
[PM2] [flow-schedule](24) ~
[PM2] [compute-record](26) ~
[PM2] [rule](27) ~
[PM2] [loki](28) ~
[PM2] [exporter](29) ~
[PM2] [etcd](30) ~
[PM2] [storage](31) ~
[PM2] [online](32) ~
[PM2] [i18n-errcode](33) ~
[PM2] [flow](34) ~
[PM2] [syslogloki](35) ~
[PM2] [front](36) ~
[PM2] [flow-engine](37) ~
[PM2] [web](38) ~
[PM2] [default-6612500b5e9b0212067135a8-modbus](39) ~
[PM2] [data-source](40) ~
[PM2] [prometheus](41) ~
[PM2] [influx](42) ~
[PM2] [v] All Applications Stopped
[PM2] [v] PM2 Daemon Stopped
```

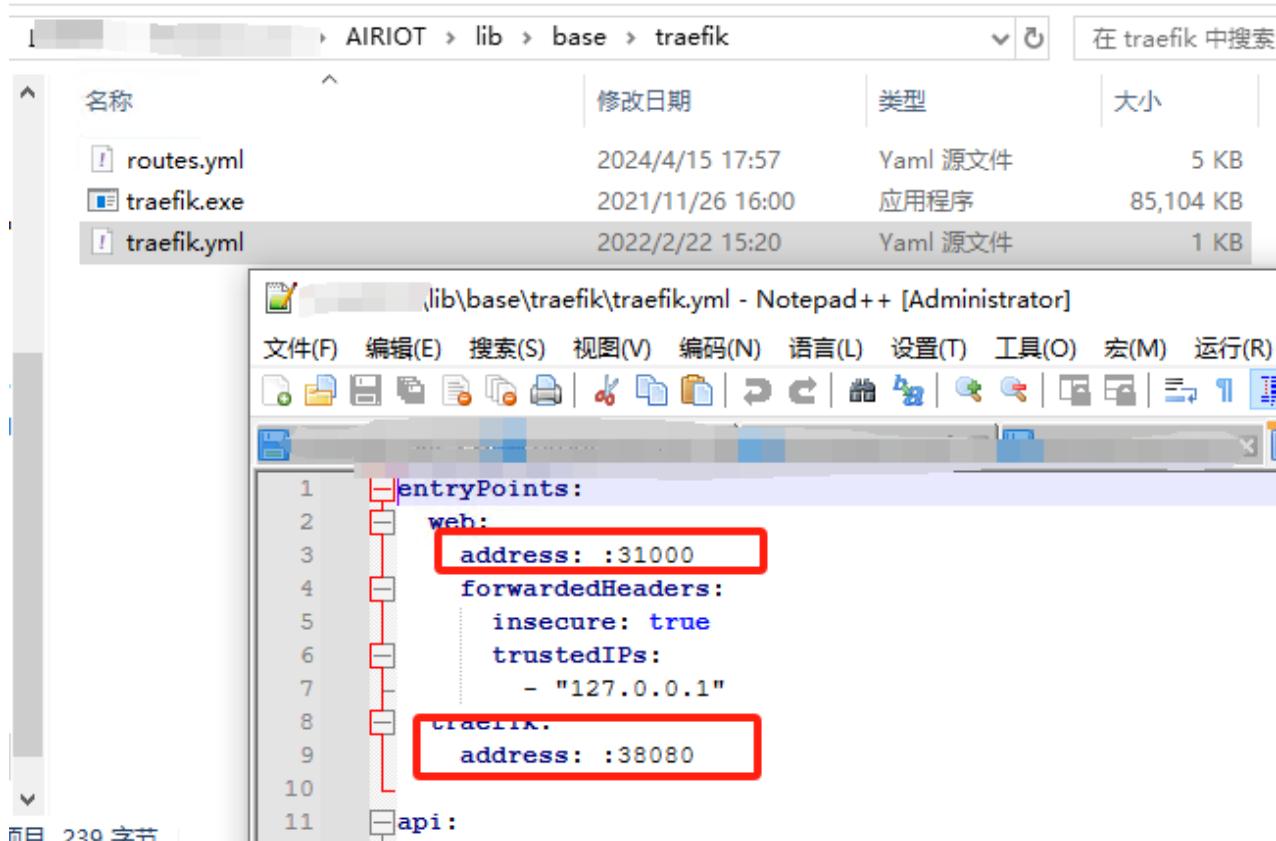
6. 打开任务管理器，选择服务，重启AIR-OPERATION服务



修改traefik服务的端口

traefik监听了两个端口，分别为31000和38080

1. 打开traefik配置文件(安装目录/lib/base/traefik/traefik.yml)，如图：



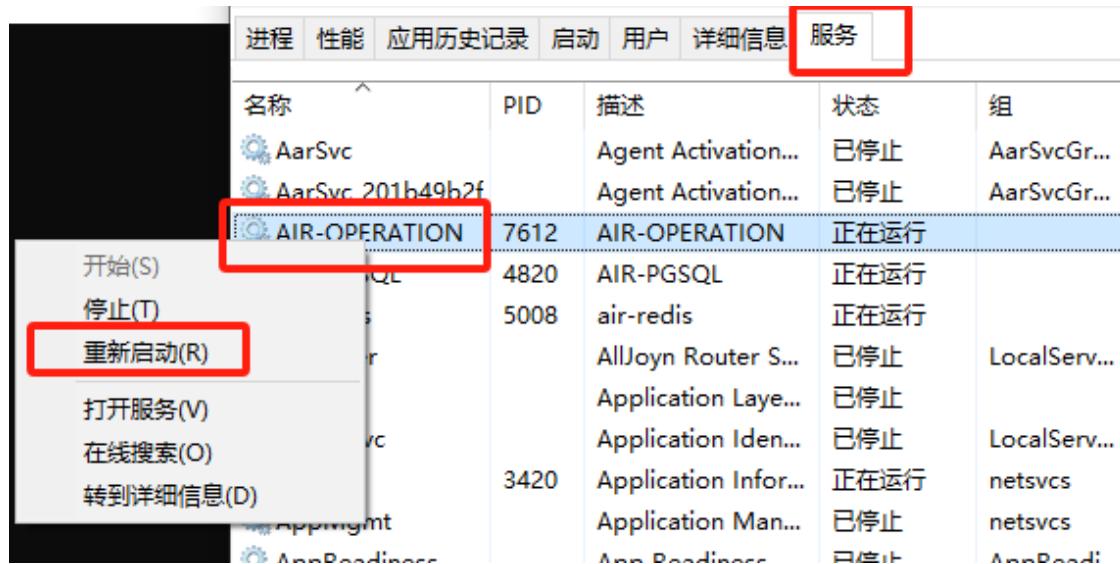
2. 修改被占用的端口为其他空闲端口，并保存配置文件

3. 打开运维的programs.yml文件(安装目录/lib/operation/programs.yml)，修改traefik服务的端口为第2步

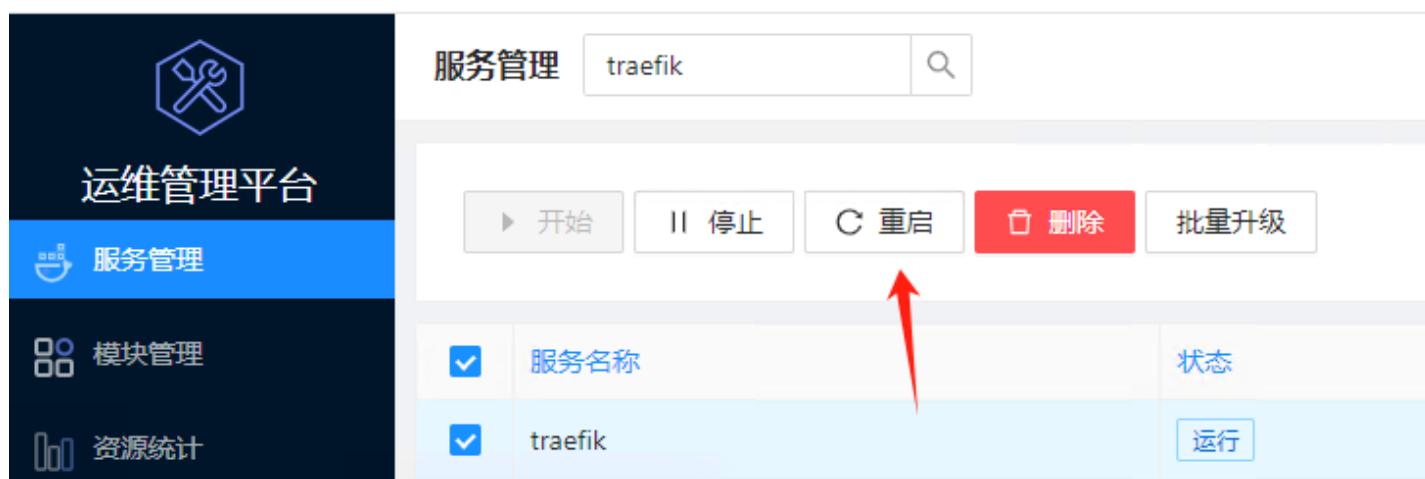
```
pm2: {}  
- namespace: base  
  name: traefik  
  command: traefik.exe  
  environ: []  
  directory: /lib/base/traefik  
  ports:  
    - 31000  
    - 38080  
  start_auto: true  
  start_retries: -1  
  version: 2.3.2  
  pm2: {}  
- namespace: service
```

相同的空闲端口

4. 重启AIR-OPERATION服务



5. 运维页面重启traefik服务



修改web服务的端口

1. 打开 web 配置文件(安装目录/lib/base/nginx-1.18.0/conf/nginx.conf), 如图:

A screenshot of a Windows File Explorer window. The address bar shows the path: AIRIOT > lib > base > nginx-1.18.0 > conf. A red box highlights this path. The main area displays a list of files in the 'nginx-1.18.0' folder. The 'nginx.conf' file is selected and highlighted with a red box. Other files listed include fastcgi.conf, fastcgi_params, koi-utf, koi-win, mime.types, scgi_params, uwsgi_params, and win-utf.

名称	修改日期	类型	大小
fastcgi.conf	2020/4/21 22:12	CONF 文件	2 KB
fastcgi_params	2020/4/21 22:12	文件	2 KB
koi-utf	2020/4/21 22:12	文件	3 KB
koi-win	2020/4/21 22:12	文件	3 KB
mime.types	2020/4/21 22:12	TYPES 文件	6 KB
nginx.conf	2024/3/18 15:18	CONF 文件	5 KB
scgi_params	2020/4/21 22:12	文件	1 KB
uwsgi_params	2020/4/21 22:12	文件	1 KB
win-utf	2020/4/21 22:12	文件	4 KB

A screenshot of Notepad++ showing the contents of the 'nginx.conf' file. The file path in the title bar is '\lib\base\nginx-1.18.0\conf\nginx.conf - Notepad++ [Administrator]'. The code in the editor is as follows:

```
22         keepalive_timeout  65;
23
24         gzip on;
25         gzip_min_length 1k;
26         gzip_buffers 4 16k;
27         gzip_comp_level 8;
28         gzip_types text/plain application/javascript application/x-javascript text/css
29         gzip_vary off;
30         gzip_disable "MSIE [1-6]\.";
31
32
33
34         fastcgi_intercept_errors on;
35         proxy_intercept_errors on;
36
37         server {
38             listen 3030; 3030;
39             server_name localhost;
40
41             client_max_body_size 10240M;
42             client_header_buffer_size 512k;
```

2. 修改被占用的端口为其他空闲端口，并保存配置文件

3. 打开运维的programs.yml文件(安装目录/lib/operation/programs.yml)，修改web服务的端口为第2步相同的空闲端口

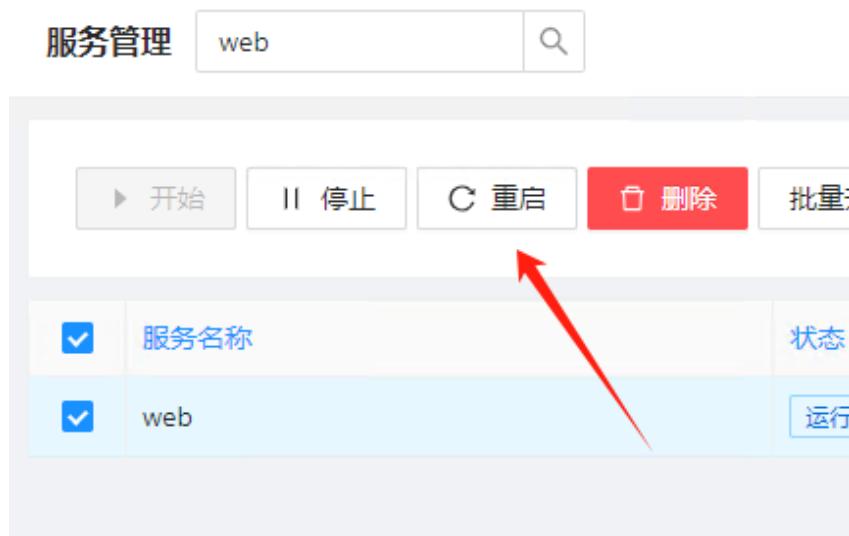
```
  - namespace: base
    name: web
    command: nginx.exe -c conf/nginx.conf
    environ: []
    directory: ./lib/base/nginx-1.18.0
    ports:
      - 3030
    start_auto: true
    start_retries: -1
    version: 1.18.0
    pm2: {}
```

同的空闲端口

4. 重启AIR-OPERATION服务

服务					
名称	PID	描述	状态	组	
AarSvc		Agent Activation...	已停止	AarSvcGr...	
AarSvc_201b49b2f		Agent Activation...	已停止	AarSvcGr...	
AIR-OPERATION	7612	AIR-OPERATION	正在运行		
开始(S)					
停止(T)					
重新启动(R)					
打开服务(V)					
在线搜索(O)					
转到详细信息(D)					

5. 运维页面重启web服务

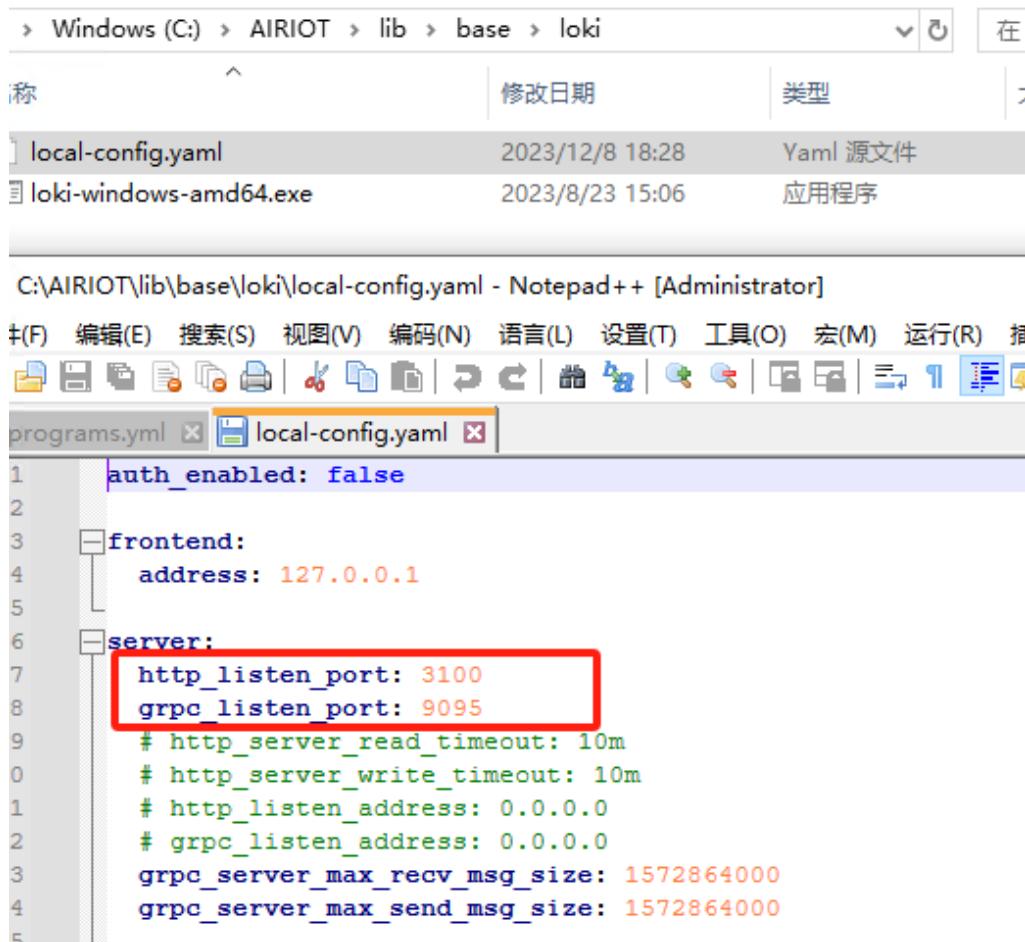


修改loki服务的端口

loki服务监听两个端口，分别为3100,9095

注：若被占用的端口为9095，可以跳过第6、7、8步 1. 打开loki配置文件(安装目录/lib/base/loki/local-

config.yaml), 如图:



2.修改被占用的端口为其他空闲端口，并保存配置文件

3.打开运维的programs.yml文件(安装目录/lib/operation/programs.yml)，修改loki服务的端口为第2步相同的空闲端口

```
- namespace: base
  name: loki
  command: loki-windows-amd64.exe
    -querier.max-concurrent=50
  environ: []
  directory: ./lib/base/loki
  ports:
    - 3100
    - 9095
  start_auto: true
  start_retries: -1
  version: 2.8.0
  repo: loki
  pm2: {}
- namespace: service
```

4.重启AIR-OPERATION服务

The screenshot shows a service management interface with a table of services. The '服务' (Service) tab is selected. A red box highlights the 'AIR-OPERATION' service row, which has a PID of 7612 and a status of '正在运行' (Running). A context menu is open over this row, with the '重新启动(R)' (Restart) option highlighted by a red box. Other options in the menu include '开始(S)', '停止(T)', '打开服务(V)', '在线搜索(O)', and '转到详细信息(D)'.

名称	PID	描述	状态	组
AarSvc		Agent Activation...	已停止	AarSvcGr...
AarSvc_201b49b2f		Agent Activation...	已停止	AarSvcGr...
AIR-OPERATION	7612	AIR-OPERATION	正在运行	
开始(S)	4820	AIR-PGSQL	正在运行	
停止(T)	5008	air-redis	正在运行	
重新启动(R)		AllJoyn Router	已停止	LocalServ...
打开服务(V)		Application Layer	已停止	
在线搜索(O)		Application Ident	已停止	LocalServ...
转到详细信息(D)	3420	Application Infor...	正在运行	netsvcs
		Application Man...	已停止	netsvcs
		App Readiness...	已停止	App Readiness

5.运维页面重启loki服务

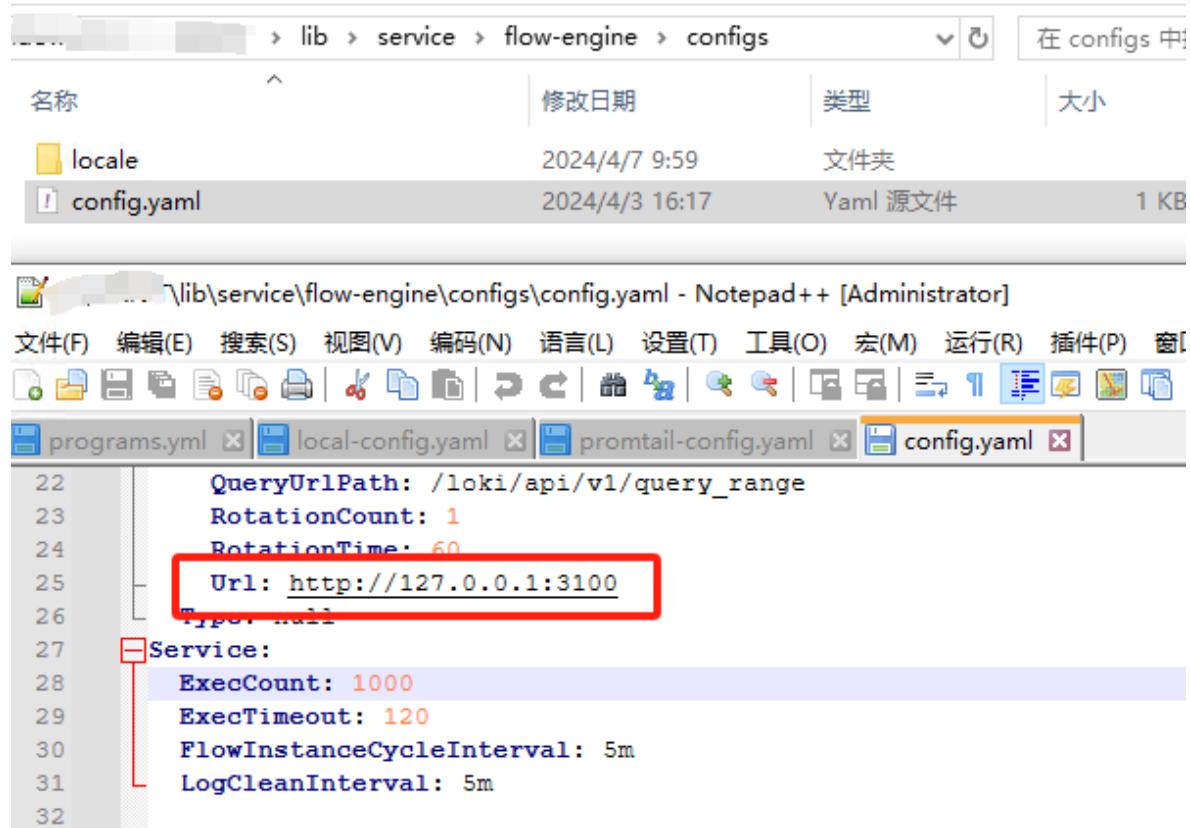
The screenshot shows a service management interface with a table of services. A red box highlights the 'loki' service row, which is selected (indicated by a checked checkbox in the first column). The service is currently '运行' (Running). A red arrow points to the '重启' (Restart) button, which is highlighted with a red box. Other buttons in the row include '开始' (Start), '停止' (Stop), and '删除' (Delete). A '批量升级' (Batch Upgrade) button is also present.

<input type="checkbox"/>	服务名称	状态
<input checked="" type="checkbox"/>	loki	运行
<input type="checkbox"/>	syslogloki	运行

6.修改promtail配置文件(安装目录/lib/base/promtail/promtail-config.yaml)中的loki端口

```
1 server:
2   http_listen_port: 9080
3   grpc_listen_port: 0
4   grpc_server_max_recv_msg_size: 1572864000
5   grpc_server_max_send_msg_size: 1572864000
6
7   positions:
8     filename: ./positions.yaml
9     # 是否忽略并覆盖被破坏的 positions 文件
10    ignore_invalid_yaml: true
11
12   clients:
13     - url: http://127.0.0.1:3100/loki/api/v1/push
14       # batchsize: 1000
15
16   scrape_configs:
17     - job_name: flog_scrape
18       static_configs:
```

7.修改flow-engine配置文件(安装目录/lib/service/flow-engine/configs/config.yaml)中的loki端口

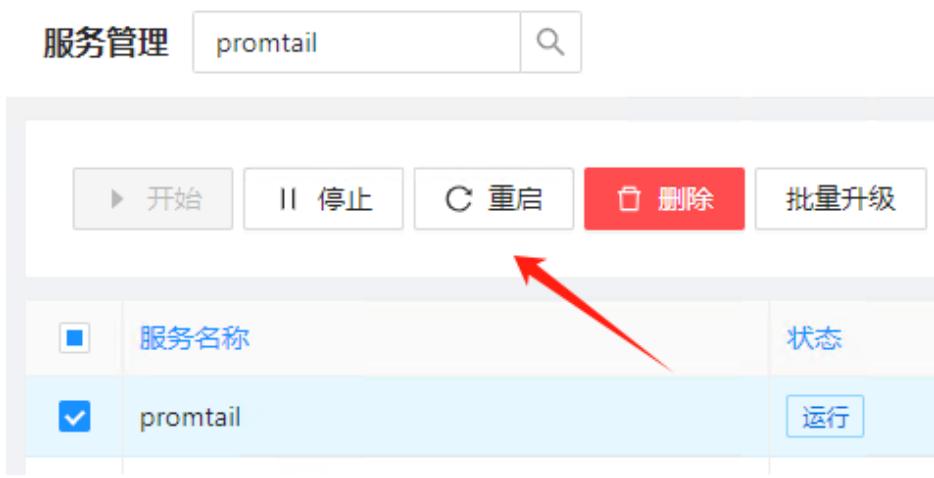


The screenshot shows a Windows file explorer window with the path: lib > service > flow-engine > configs. Inside the configs folder, there are two items: 'locale' (a folder) and 'config.yaml'. The 'config.yaml' file is selected and its details are shown: 修改日期 (Modified Date) is 2024/4/7 9:59, 类型 (Type) is 文件夹 (Folder), and 大小 (Size) is 1 KB. Below the file explorer is a Notepad++ window displaying the 'config.yaml' file content. The URL configuration is highlighted with a red box.

```
22     QueryUrlPath: /loki/api/v1/query_range
23     RotationCount: 1
24     RotationTime: 60
25     Url: http://127.0.0.1:3100
26     Type: null
27     Service:
28       ExecCount: 1000
29       ExecTimeout: 120
30       FlowInstanceCycleInterval: 5m
31       LogCleanInterval: 5m
32
```

8.运维页面重启flow-engine,promtail服务





修改syslogloki服务的端口

syslogloki服务监听两个端口，分别为3101,49095

注：若被占用的端口为49095，可以跳过第6、7、8步
1.打开loki配置文件(安装目录/lib/base/syslogloki/local-config.yaml)，如图：

```
1 auth_enabled: false
2
3 - frontend:
4   address: 127.0.0.1
5
6 - server:
7   http_listen_port: 3101
8   grpc_listen_port: 49095
9   # http_server_read_timeout: 10m
10  # http_server_write_timeout: 10m
11  # http_listen_address: 0.0.0.0
12  # ::1 listen_address: 0.0.0.0
```

2.修改被占用的端口为其他空闲端口，并保存配置文件

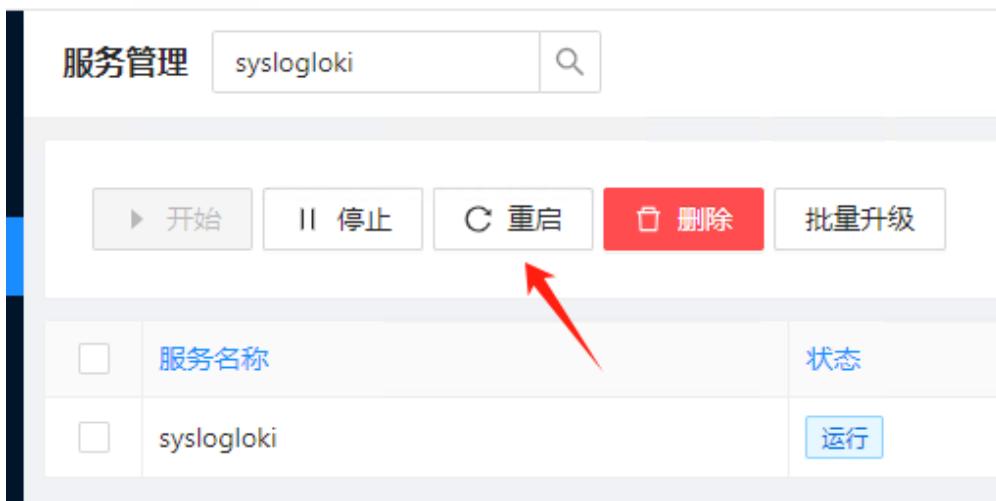
3.打开运维的programs.yml文件(安装目录/lib/operation/programs.yml)，修改syslogloki服务的端口为第2步相同的空闲端口

```
namespace: base
name: syslogloki
command: loki-windows-amd64.exe -config.
environ: []
directory: ./lib/base/loki
ports:
- 3101
- 49095
start_auto: true
start_retries: -1
version: 2.8.0
repo: loki
pm2: {}
namespace: service
```

4.重启AIR-OPERATION服务



5.运维页面重启syslogloki服务



6.修改syslogpromtail配置文件(安装目录/lib/base/syslogpromtail/promtail-config.yaml)中的syslogloki端口

名称	修改日期	类型	大小
positions.yaml	2024/5/17 11:48	Yaml 源文件	49
promtail-config.yaml	2024/1/18 13:40	Yaml 源文件	2

C:\AIRIOT\lib\base\syslogpromtail\promtail-config.yaml - Notepad++ [Administrator]

文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P)



promtail-config.yaml

```
1  server:
2    http_listen_port: 49080
3    grpc_listen_port: 0
4    grpc_server_max_recv_msg_size: 1572864000
5    grpc_server_max_send_msg_size: 1572864000
6
7  positions:
8    filename: ../syslogpromtail/positions.yaml
9    # 是否忽略并覆盖被破坏的 positions 文件
10   ignore_invalid_yaml: true
11
12 clients:
13   - url: http://127.0.0.1:3101/loki/api/v1/push
14     # batchsize: 1000000
15
16 scrape_configs:
17   - job_name: file_scrape
```

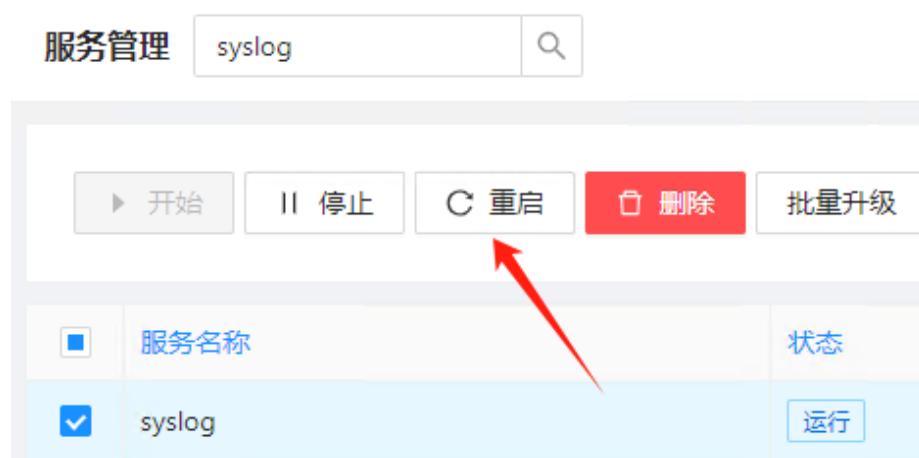
7.修改syslog配置文件(安装目录/lib/service/syslog/configs/config.yaml)中的syslogloki端口

The screenshot shows a Windows File Explorer window with the path: 脑 > Windows (C:) > AIRIOT > lib > service > syslog > configs. A file named config.yaml is selected. Below the file list is a Notepad++ window titled 'config.yaml' containing the following YAML code:

```
1  App:
2    Etcd:
3      DialTimeout: 10
4      Endpoints:
5        - 127.0.0.1:2379
6      Password: ENC(ZGVsbDEyMw==)
7      Username: root
8      EtcdConfig: /airiot/config/pro.json
9      Server:
10        HTTP:
11          Port: 59004
12        Metadata:
13          env: local
14        ServerName: syslog
15      Syslog:
16        Loki:
17          ClearUrlPath: /loki/api/v1/delete
18          QueryUrlPath: /loki/api/v1/query_range
19          Url: http://127.0.0.1:3101
20          Type: null
21
```

The 'Url' field under the 'Loki' section is highlighted with a red rectangle.

8.运维页面重启syslog,syslogpromtail服务



服务管理

syslogprom



开始

停止

C 重启

删除

批量升级



服务名称

状态

syslogpromtail

运行

修改promtail服务的端口

1. 打开promtail配置文件(安装目录/lib/base/promtail/promtail-config.yaml), 如图:

```
1 server:
2   http_listen_port: 9080
3   grpc_listen_port: 0
4   grpc_server_max_recv_msg_size: 1572864000
5   grpc_server_max_send_msg_size: 1572864000
6
7 positions:
8   filename: ./positions.yaml
9   # 是否忽略并覆盖被破坏的 positions 文件
10  ignore_invalid_yaml: true
```

2. 修改被占用的端口为其他空闲端口，并保存配置文件

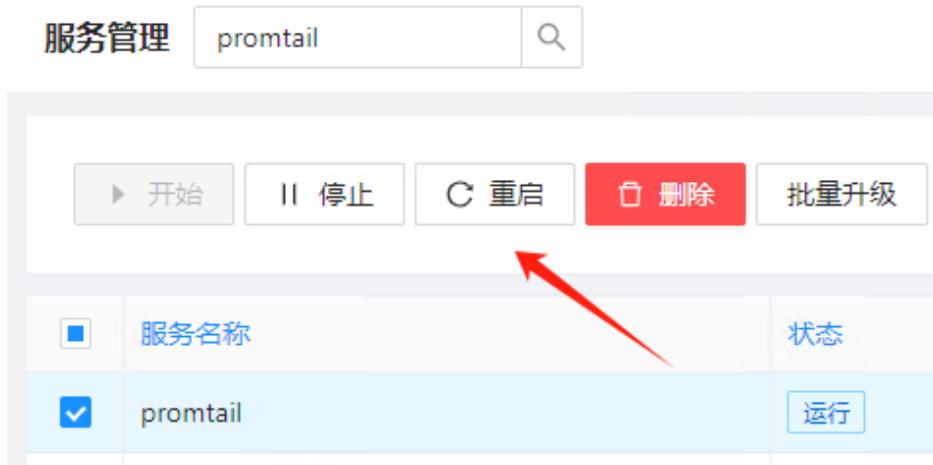
3. 打开运维的programs.yml文件(安装目录/lib/operation/programs.yml), 修改promtail服务的端口为第2步相同的空闲端口

```
- namespace: base
  name: promtail
  command: promtail-windows-amd64.exe -config.file=./promtail-config.yaml --log.level=debug
  environ: []
  directory: ./lib/base/promtail
  ports:
    - 9080
  start_auto: true
  start_retries: -1
  version: 2.8.0
  repo: promtail
  pm2: {}
```

4.重启AIR-OPERATION服务

名称	PID	描述	状态	组
AarSvc		Agent Activation...	已停止	AarSvcGr...
AarSvc_201b49b2f		Agent Activation...	已停止	AarSvcGr...
AIR-OPERATION	7612	AIR-OPERATION	正在运行	
开始(S)	4820	AIR-PGSQL	正在运行	
停止(T)	5008	air-redis	正在运行	
重新启动(R)		AllJoyn Router S...	已停止	LocalServ...
打开服务(V)		Application Laye...	已停止	
在线搜索(O)		Application Iden...	已停止	LocalServ...
转到详细信息(D)	3420	Application Infor...	正在运行	netsvcs
		Application Man...	已停止	netsvcs
		App Readiness	已停止	AppReadi...

5.运维页面重启promtail服务



修改syslogpromtail服务的端口

1. 打开syslogpromtail配置文件(安装目录/lib/base/syslogpromtail/promtail-config.yaml), 如图:

AIRIOT > lib > base > promtail

名称	修改日期	类型	大小
positions.yaml	2024/5/17 15:01	Yaml 源文件	1 KB
promtail-config.yaml	2023/11/14 10:25	Yaml 源文件	2 KB
promtail-windows-amd64.exe	2023/8/23 15:07	应用程序	89,300 KB

T\lib\base\promtail\promtail-config.yaml - Notepad++ [Administrator]

文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P) 窗口(W)

promtail-config.yaml config.yaml programs.yml promtail-config.yaml

```
1 server:
2   http_listen_port: 9080
3   grpc_listen_port: 0
4   grpc_server_max_recv_msg_size: 1572864000
5   grpc_server_max_send_msg_size: 1572864000
6
7 positions:
8   filename: ./positions.yaml
9   # 是否忽略并覆盖被破坏的 positions 文件
10  ignore_invalid_yaml: true
```

2. 修改被占用的端口为其他空闲端口，并保存配置文件

3. 打开运维的programs.yml文件(安装目录/lib/operation/programs.yml), 修改syslogpromtail服务的端口为第2步相同的空闲端口

```
- namespace: base
  name: syslogpromtail
  command: promtail-windows-amd64.exe -config.file=../syslogpromtail/promtail-config.yaml
    --log.level=info
  environ: []
  directory: ./lib/base/promtail
  ports:
    - 49080
  start_auto: true
  start_retries: -1
  version: 2.8.0
  repo: promtail
  pm2: {}
```

4.重启AIR-OPERATION服务

The screenshot shows the Windows Task Manager's Services tab. A red box highlights the '服务' (Services) tab at the top. In the list, the 'AIR-OPERATION' service is selected, and a red box highlights its name. A context menu is open over the service, with the '重新启动(R)' (Restart) option highlighted by a red box.

名称	PID	描述	状态	组
AarSvc		Agent Activation...	已停止	AarSvcGr...
AarSvc_201b49b2f		Agent Activation...	已停止	AarSvcGr...
AIR-OPERATION	7612	AIR-OPERATION	正在运行	
开始(S)	4820	AIR-PGSQL	正在运行	
停止(T)	5008	air-redis	正在运行	
重新启动(R)		AllJoyn Router S...	已停止	LocalServ...
打开服务(V)		Application Laye...	已停止	
在线搜索(O)		Application Iden...	已停止	LocalServ...
转到详细信息(D)	3420	Application Infor...	正在运行	netsvcs
		Application Man...	已停止	netsvcs
		App Readiness...	已停止	App Readiness...

5.运维页面重启syslogpromtail服务

The screenshot shows a 'Service Management' interface. At the top, there is a search bar with 'syslogprom' and a magnifying glass icon. Below the search bar are several buttons: '开始' (Start), '停止' (Stop), '重启' (Restart) which is highlighted with a red box and has a red arrow pointing to it, '删除' (Delete), and '批量升级' (Batch Upgrade). The main area is a table with columns '服务名称' (Service Name) and '状态' (Status). Two services are listed: 'syslogpromtail' (status: 运行 - Running) and another service whose name is partially visible. Both services have a checked checkbox in the first column.

	服务名称	状态
<input checked="" type="checkbox"/>	syslogpromtail	运行
<input checked="" type="checkbox"/>		

平台无法访问

问题描述

在离线的Windows服务器上平台运行一段时间后突然无法访问，只有重启电脑可以解决

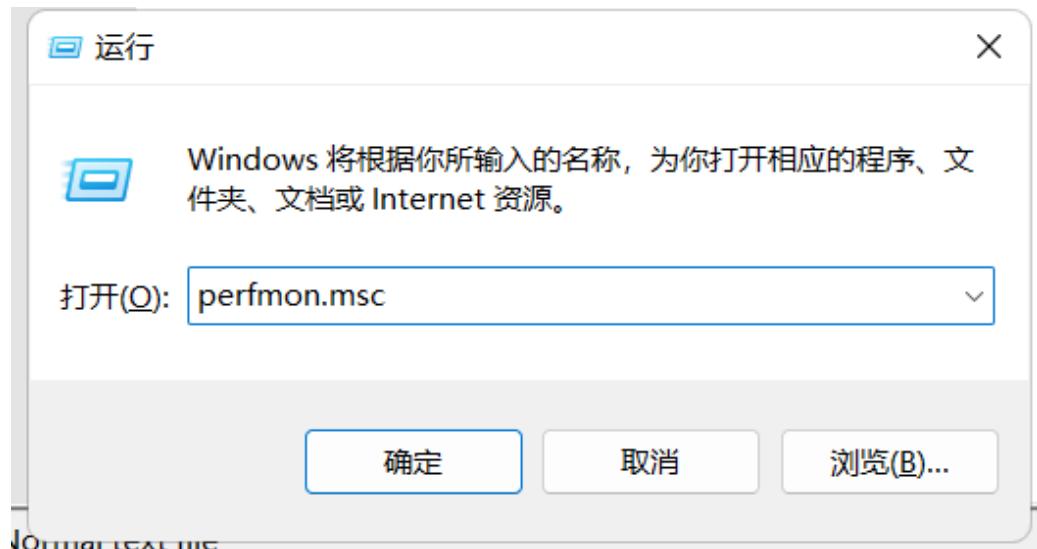
思路

大致是因为电脑tcp连接数过多

解决办法

排查tcp连接数

1.'win+R'打开运行，输入'perfmon.msc'，回车



2.点击'性能监视器'

3.点击上方'+'号，添加TCPv4

添加计数器

可用计数器

从计算机选择计数器(M): <本地计算机> 浏览(B)...

System
TCPIP Performance Diagnostics
TCPIP Performance Diagnostics (Per-CPU)
TCPv4 (highlighted with a red box)
TCPv6
Telephony
Teredo Client
Teredo Relay
Teredo Server

选定对象的实例(I):

1.选择TCPv4

2.点击添加

添加(D) >>

添加的计数器(C)

计数器	父系	实例	计算机
TCPv4	---	---	---

3.点击确定

显示描述(H) 确定 取消

4.查看'Connections Active'的数量，超过50万基本确定思路正确。



排查占用tcp连接数多的进程

1.打开命令行

2.输入'**netstat -noqt**'，查看占用连接数最多的进程PID；也可以打开**powershell**并输入以下指令排查：

```
Get-NetTCPConnection | Group-Object -Property State, OwningProcess | Select -Property Count, Name, @{Name="ProcessName";Expression={(Get-Process -PID ($_.Name.Split(',')[-1].Trim(''))).Name}}, Group | Sort Count -Descending
```

3.若该进程与平台无关且客户允许关闭，则关闭该进程

4.查看平台是否可以正常访问

平台迁移

Windows系统(非pm2)

例如：从C:\AIRIOT 迁移至 D:\AIRIOT,下文中的路径按照实际情况来执行

1.停止平台

以管理员身份打开cmd sc stop AIRIOT

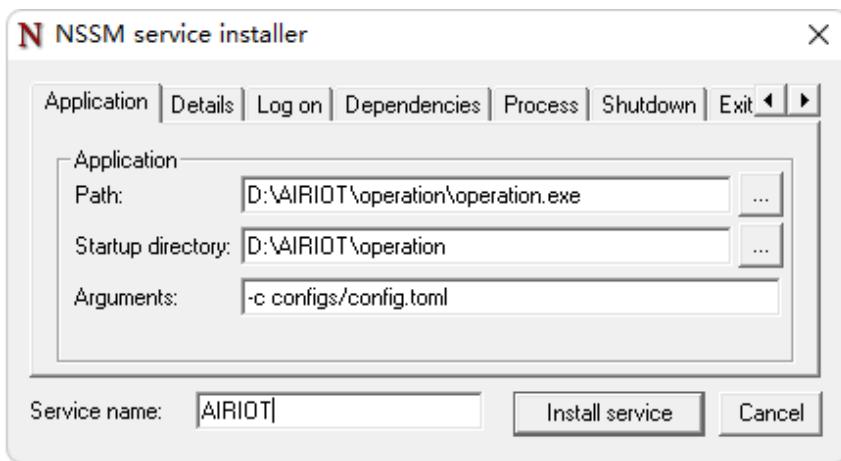
sc delete AIRIOT

2.迁移平台

将C盘下的AIRIOT文件复制到D盘

3.创建服务

打开cmd 进入D:\AIRIOT\nssm-2.24\win64 nssm.exe install



输入内容如

上图所示，点击‘Install service’

4.启动平台

以管理员身份打开cmd sc start AIRIOT

Windows系统(pm2版本)

例如：从C:\AIRIOT 迁移至 D:\AIRIOT,下文中的路径按照实际情况来执行

1.停止平台

打开任务管理器->服务，分别停止AIR-PGSQL、air-redis、AIR-OPERATION、pm2.exe

The screenshot shows the Windows Task Manager with the 'Services' tab selected. A red box highlights the service names: AIR-OPERATION, AIR-PGSQL, air-redis, and pm2.exe. These four services are listed as '正在运行' (Running). Other services like AarSvc, ALG, and PlugPlay are also visible but not highlighted.

名称	PID	描述	状态	组
AarSvc		Agent Activation...	已停止	AarSvcGr...
AarSvc_3964a4		Agent Activation	已停止	AarSvcGr...
AIR-OPERATION	13724	AIR-OPERATION	正在运行	
AIR-PGSQL	4780	AIR-PGSQL	正在运行	
air-redis	4672	air-redis	正在运行	
AJRouter		AllJoyn Router S...	已停止	LocalServi...
ALG		Application Laye...	已停止	
AppIDSvc		Application Iden...	已停止	LocalServi...
Anninfo	14960	Annlication Infor...	正在运行	netsvcs
PlugPlay	668	Plug and Play	正在运行	DcomLau...
pm2.exe	16360	PM2	正在运行	
PNRPAutoReg		PNRP Machine N...	已停止	LocalServi...
PNRPsvc		Peer Name Resol...	已停止	LocalServi...

2.将平台迁移到指定位置

复制C:\AIRIOT文件夹到D盘

3.修改新目录中文件的绝对路径

1) 进入D:\AIRIOT\airiot\loki

修改local-config.yaml文件中的路径，将C:\AIRIOT替换为D:\AIRIOT

2) 进入D:\AIRIOT\airiot\syslogloki

修改local-config.yaml文件中的路径，将C:\AIRIOT替换为D:\AIRIOT

4.删除pm2文件

1) 进入C:\ProgramData目录，如果未找到该目录，再文件查看中选择查看隐藏的文件，如下图

文件夹选项



常规 查看 搜索

文件夹视图



你可以将此视图(如详细信息或图标)应用于这种类型的所有文件夹。

[应用到文件夹\(L\)](#)

[重置文件夹\(R\)](#)

高级设置:

- 鼠标指向文件夹和桌面项时显示提示信息
- 显示驱动器号
- 显示同步提供程序通知
- 显示状态栏
- 隐藏空的驱动器
- 隐藏受保护的操作系统文件(推荐)

隐藏文件和文件夹

- 不显示隐藏的文件、文件夹或驱动器
- 显示隐藏的文件、文件夹和驱动器

- 隐藏文件夹并合并
- 隐藏已知文件类型的扩展名
- 用彩色显示加密或压缩的 NTFS 文件
- 在标题栏中显示完整路径
- 在地址栏中显示完整路径

[还原为默认值\(D\)](#)

[确定](#)

[取消](#)

[应用\(A\)](#)

2) 删除pm2文件夹

> Windows (C:) > ProgramData >

名称	修改日期
Oray	2020/8/11 14:02
OrayClient	2022/7/9 9:26
Package Cache	2023/12/14 18:00
Packages	2024/5/24 21:01
pm2	2024/5/28 10:12
PreEmptive Solutions	2020/12/18 11:01
Pulse Secure	2023/6/15 9:41

5.重新创建服务

管理员模式打开cmd，执行

```
sc delete AIR-OPERATION
sc delete AIR-PGSQL
sc delete air-redis
sc delete pm2.exe

// 进入新的安装目录
cd D:\AIRIOT

// 执行安装脚本
install.bat

// 启动air-operation服务
sc start air-operation
```

管理员: C:\WINDOWS\system32\cmd.exe

```
D:\AIRIOT1>install.bat
Script directory: D:\AIRIOT1\
pg.exe path: "D:\AIRIOT1\lib\base\pgsql\bin\pg_ctl.exe"
pgdata path: "D:\AIRIOT1\airiot\db\postgres\pgdata"
[SC] CreateService 失败 1073:
```

指定的服务已存在。

```
[SC] ChangeServiceConfig2 成功
[SC] StartService 失败 1056:
```

服务的实例已在运行中。

"安装 redis 服务"

```
[SC] ChangeServiceConfig2 成功
[SC] StartService 失败 1056:
```

服务的实例已在运行中。

```
[SC] CreateService 失败 1073:
```

指定的服务已存在。

```
[SC] ChangeServiceConfig2 成功
```

```
WARNING: Bundle was created on win32-10.0.22631 but this is win32-10.0.22000.
```

```
npm notice
```

```
npm notice New major version of npm available! 9.6.7 -> 10.5.2
```

```
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.5.2
```

```
npm notice Run npm install -g npm@10.5.2 to update!
```

```
npm notice
```

```
[PM2] Setting changed
```

```
Module: pm2-logrotate
```

```
$ pm2 set pm2-logrotate:retain 3
```

"安装完成"

```
D:\AIRIOT1>
```

运维忘记密码

Windows系统

停止平台

删除运维数据文件

进入安装目录下/lib/operation/data 删除data目录下的文件

启动平台

安装和授权部分

1. Windows下的平台安装后，运维平台中的etcd-set服务处于启动失败



问题描述：

第一次注册和登录运维平台后，查看服务管理：运维平台中的etcd-set服务处于启动失败的状态。

解决方案：

etcd-set服务为初始化服务，安装时只启动一次，安装成功后该服务的使命结束，可以将其删除掉。

2. Windows单机版平台-运维平台中服务全部启动，但是中台进不去

问题描述

运维平台中服务全部启动，但是中台进不去。

解决方案

datascore服务升级失败，删除安装失败的datascore服务后，重新安装该服务，问题解决。

3. Windows单机版平台-授权文件上传后，授权内容显示不全

问题描述

授权文件上传后，显示不全。

解决方案

授权文件不必解压，直接上传压缩包，问题解决。

4. 空间版服务的i18n的服务一直处于重启状态

问题描述

空间版服务的i18n的服务一直处于重启状态，日志没有打印信息，如图所示：



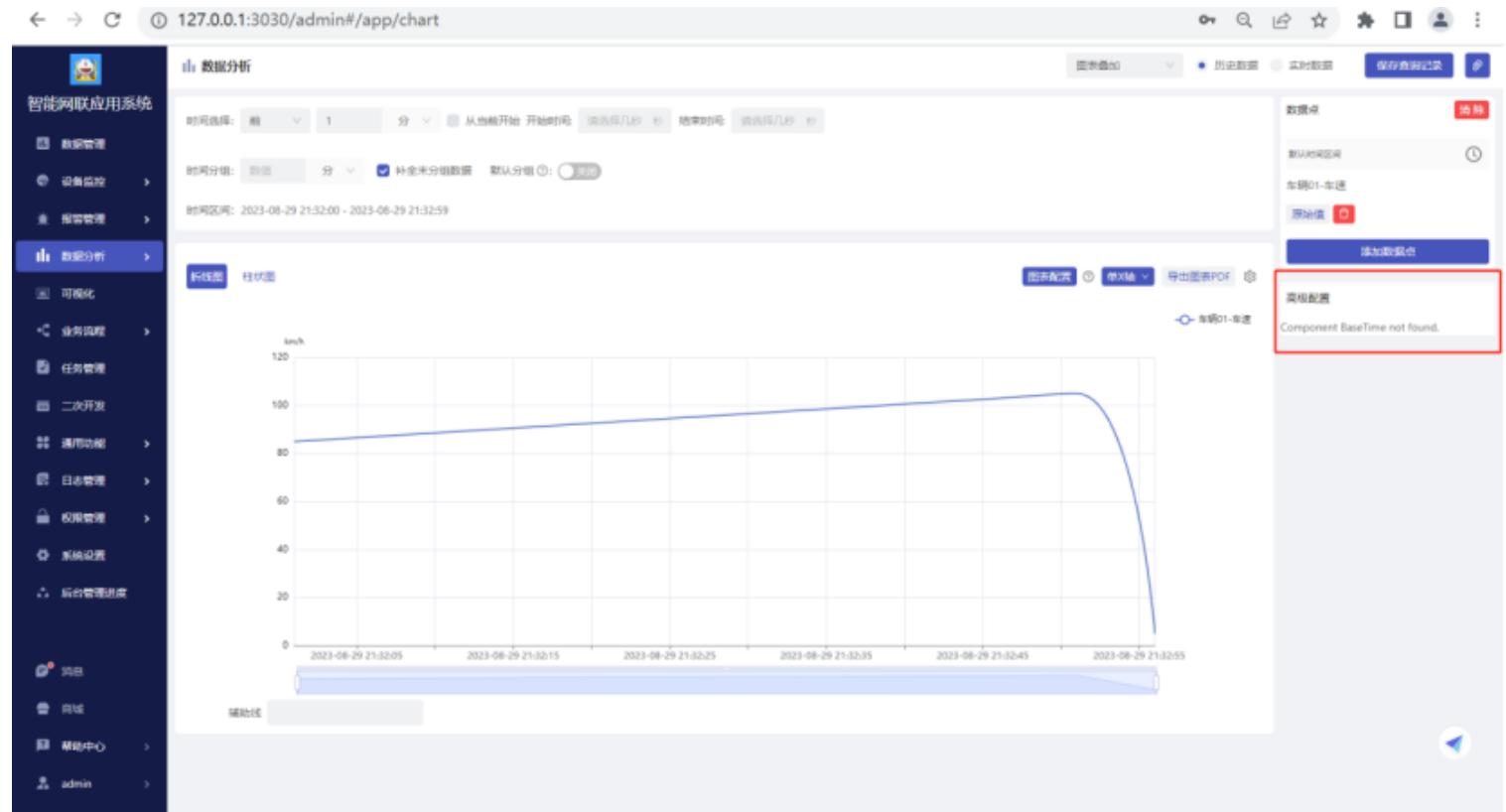
解决方案

该服务是国际化的服务，这个服务只运行一次即可，平台安装成功后，可以将其删除。

5. Windows单机版平台-数据分析功能-高级配置显示问题

问题描述

4.0平台-windows单机版：数据分析显示如图所示：



解决方案

在运维平台中添加，@airiot/gg-project模块，如图所示，问题得以解决。

The screenshot shows the Operations Management Platform (运维管理平台) interface. The left sidebar includes Service Management (服务管理) and Module Management (模块管理, highlighted). The main area is titled 'Module Management' (模块管理) and shows a table for the '@airiot/gg-project' module. The table columns are: 模块名称 (Module Name), 创建时间 (Creation Time), 描述 (Description), 版本 (Version), and 最新版本 (Latest Version). The module details are: @airiot/gg-project, 2023-08-30 14:45:49, 国工数据分析工作表兼容, v4.0.1, and v4.0.1 (latest). A blue button '+ 添加/更新模块' (+ Add/Update Module) is located at the top right of the table.

模块名称	创建时间	描述	版本	最新版本
@airiot/gg-project	2023-08-30 14:45:49	国工数据分析工作表兼容	v4.0.1	v4.0.1

6. Windows单机版平台-用户管理-显示问题

问题描述

单机版平台安装后，打开用户管理功能，显示如图：

The screenshot shows the AIRIOT platform's user management interface. On the left sidebar, under the '权限管理' (Permission Management) section, the '用户管理' (User Management) item is highlighted with a red box. The main content area is titled '添加用户' (Add User). It includes fields for '用户角色' (User Role), '密码' (Password) with a strength meter, '重复密码' (Repeat Password), and '备注' (Remarks). A red box highlights an error message: '表管理: Component TableListSelecttMultiple not found.' Below this, there are sections for '管理表记录' (Manage Table Records) and '报警分组' (Alarm Grouping), each with a '选择表记录' (Select Table Record) button. At the bottom, there are configuration options for '安全配置' (Security Configuration) including '有效时长' (Effective Duration) and '强制修改密码' (Force Password Change).

解决方案

在运维平台中添加，@airiot/components模块，如图所示，问题解决。

The screenshot shows the operations and maintenance platform's module management interface. On the left sidebar, the '模块管理' (Module Management) item is highlighted with a red box. The main content area shows a table of installed modules. One row for '@airiot/components' is selected, showing details: '模块名称' (@airiot/components), '创建时间' (2023-09-02 15:47:03), '描述' (AIRIOT公共组件), '版本' (v4.1.3), and '最新版本' (v4.1.3). A red box highlights the '历史版本' (History Version) link.

使用AIRIOT平台过程中要注意的内容

1.建议使用的浏览器：谷歌和火狐

2.禁用Windows Update

按Win+R输入services.msc并按Enter键打开服务页面，操作如图1-1所示：

运行

X



Windows 将根据你所输入的名称，为你打开相应的程序、文件夹、文档或 Internet 资源。

打开(O):

services.msc



使用管理权限创建此任务。

确定

取消

浏览(B)...

在右侧列表中找到“Windows Update”选项，将该服务停止，然后双击进入详细属性页面，将其启动类型配置为“禁用”，然后单击“应用”>“确定”即可关闭Windows自动更新，操作页面如图1-2和1-3所示：

The screenshot shows the Windows Services console. The title bar says "服务(本地)". In the center, there is a table listing various Windows services. The "Windows Update" service is highlighted with a red box. To its left, there are two buttons: "停止此服务" (Stop this service) and "重启动此服务" (Restart this service), also both highlighted with red boxes. A large red box covers the entire left side of the window, from the title bar down to the service list. On the right side of the service list, there is a column labeled "启动类型" (Type) which contains the word "禁用" (Disabled) next to the Windows Update service. Red text overlaid on the image reads "1停止该服务" (1 Stop this service) and "2将该服务改成禁用" (2 Change this service to Disabled).

名称	描述	状态	启动类型	登录为
Windows Event Collector	此服...	手动	网络服务	
Windows Event Log	此服...	正在...	自动	本地服务
Windows Font Cache Servi...	通过...	正在...	自动	本地服务
Windows Image Acquisitio...	为扫...	正在...	自动(触发...	本地服务
Windows Installer	添加...	手动	本地系统	
Windows Management Ins...	提供...	正在...	手动	本地系统
Windows Media Player Net...	使用...	手动	网络服务	
Windows Mixed Reality Op...	Enab...	手动	本地系统	
Windows Modules Installer	启用 ...	手动	本地系统	
Windows Presentation Fou...	通过...	正在...	手动	本地服务
Windows PushToInstall 服务	为 M...	手动(触发...	本地系统	
Windows Remote Manage...	Win...	手动	网络服务	
Windows Search	为文...	正在...	手动	本地系统
Windows Time	维护...	手动(触发...	本地系统	
Windows Update	启用...	正在...	禁用	本地系统
Windows 安全中心服务	Win...	正在...	手动	本地系统
Windows 备份	提供 ...	手动	本地系统	
Windows 感知服务	启用...	手动(触发...	本地服务	
Windows 感知模拟服务	实现...	手动	本地系统	
Windows 更新医生服务	启用 ...	手动	本地系统	

名称	描述	状态
Windows Event Collector	此服...	正在...
Windows Event Log	此服...	正在...
Windows Font Cache Servi...	通过...	正在...
Windows Image Acquisitio...	为扫...	正在...
Windows Installer	添加...	
Windows Management Ins...	提供...	正在...
Windows Media Player Net...	使用...	
Windows Mixed Reality Op...	Enab...	
Windows Modules Installer	启用 ...	
Windows Presentation Fou...	通过...	正在...
Windows PushToInstall 服务	为 M...	
Windows Remote Manage...	Win...	
Windows Search	为文...	正在...
Windows Time	维护...	
Windows Update	启用...	正在...
Windows 安全中心服务	Win...	正在...
Windows 备份	提供 ...	
Windows 感知服务	启用...	
Windows 感知模拟服务	实现...	
Windows 更新医生服务	启用 ...	
Windows 管理服务	执行...	
Windows 推送通知系统服务	此服...	正在...
Windows 许可证管理器服务	为 M...	正在...
Windows 移动热点服务	提供...	
Windows 预览体验成员服务	为 W...	
WinHTTP Web Proxy Auto...	Win...	正在...
Wired AutoConfig	有线...	手动
WLAN AutoConfig	WLA...	正在...



启用HTTPS访问

如果想要给平台添加 `https` 的访问方式, 可以按以下步骤操作:

1. 准备证书

可以使用自签名的证书或向相关机构申请安全证书.

证书包含以下 2 个文件:

- <文件>.key
- <文件>.pem

使用自签名的证书时, 浏览器会有不安全访问的相关提示.

2. 修改平台配置

1. 上传证书

将 [步骤1](#) 中得到的文件上传到平台服务器上.

可以将证书文件放置在 `平台安装根目录/lib/base/nginx-1.18.0/cert` 目录下. 例如:

`D:/AIRIOT/lib/base/nginx-1.18.0/cert`

2. 修改配置

修改 `web` 服务的 `nginx.conf` 配置文件. 该文件的路径通常为 `平台安装根目录/lib/base/nginx-1.18.0/conf/nginx.conf`. 使用文本编辑工具打开该文件, 按照以下步骤进行操作:

1. 将现有端口修改为https

如果要将现有端口修改为 `https` 的访问方式, 需要修改 `nginx.conf` 中的 `3030` 端口对应的 `Server` 内容, 添加 `ssl` 相关配置.

注: 需要注意标有注释的内容.

```
server {  
    # 将端口号改为 443 并启用 ssl
```

```

listen      443 ssl;
# 访问域名，多个域名以空格分开
server_name  localhost;

# 证书相关配置，具体路径为实际证书存放路径
ssl_certificate_key  平台安装根目录/lib/base/nginx-1.18.0/cert/<文件名>.key;
ssl_certificate      平台安装根目录/lib/base/nginx-1.18.0/cert/<文件名>.pem;

# ssl验证相关配置
ssl_session_timeout  5m;      #缓存有效期
ssl_ciphers ECDHE-RSA-AES128-GCM-
SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;      #加密算法
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;      #安全链接可选的加密协议
ssl_prefer_server_ciphers on;      #使用服务器端的首选算法

# 其余配置保持不变即可
}

```

2. 添加新端口为https

如果想同时允许 `http` 和 `https` 两种访问方式时，需要额外添加一套平台配置。步骤如下：

- 将 `3030` 端口的 `Server` 配置复制一份
- 按前面的 [步骤](#) 修改新复制的 `Server`.

```

# 原有的配置
server {
    listen      3030;
    server_name  localhost;
    keepalive_timeout  180;
    client_header_timeout 180;
    send_timeout 180;
    client_body_timeout 180;
    client_max_body_size 300000M;
    proxy_connect_timeout 3600s;
    proxy_send_timeout 3600s;
    proxy_read_timeout 3600s;
    client_header_buffer_size 512k;
    large_client_header_buffers 4 512k;
    charset utf-8;

    # 其它内容
}

# 新复制并修改后的配置
server {
    # 将端口号改为 443 并启用 ssl
}

```

```
listen      443 ssl;
# 访问域名，多个域名以空格分开
server_name  localhost;

# 证书相关配置，具体路径为实际证书存放路径
ssl_certificate_key    平台安装根目录/lib/base/nginx-1.18.0/cert/<文件名>.key;
ssl_certificate        平台安装根目录/lib/base/nginx-1.18.0/cert/<文件名>.pem;

# ssl验证相关配置
ssl_session_timeout  5m;      #缓存有效期
ssl_ciphers ECDHE-RSA-AES128-GCM-
SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;      #加密算法
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;      #安全链接可选的加密协议
ssl_prefer_server_ciphers on;      #使用服务器端的首选算法

# 其余配置保持不变即可
}
```

3. 重启相关服务

登录 **运维管理系统**, 访问地址通常为 **http://ip:13030**, 点击 **服务管理** 菜单, 找到 **web** 服务, 勾选后点击**重启**.

重启后, 查看 **web** 服务的运行状态, 如果为 **运行**, 则表示修改成功, 否则表示修改失败. 此时点击 **web** 服务的 **更多** 菜单中的日志查看日志, 并根据日志中的错误信息做相应的调整, 调整后再次启动 **web**.

修改成功后即可通过 **https** 协议访问平台.

loki服务报错

问题描述

```
error contacting scheduler
```

报错详情：

```
level=error ts=2023-05-06T07:09:48.227728Z caller=scheduler_processor.go:86 msg="error contacting scheduler" err="rpc error: code = Unavailable desc = connection error: desc = \"transport: Error while dialing dial tcp 172.19.83.237:9095: connectex: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.\""  
addr=172.19.83.237:9095  
level=error ts=2023-05-06T07:09:48.321964Z caller=scheduler_processor.go:86 msg="error contacting scheduler" err="rpc error: code = Unavailable desc = connection error: desc = \"transport: Error while dialing dial tcp 172.19.83.237:9095: connectex: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.\""  
addr=172.19.83.237:9095  
level=error ts=2023-05-06T07:09:48.368463Z caller=scheduler_processor.go:86 msg="error contacting scheduler" err="rpc error: code = Unavailable desc = connection error: desc = \"transport: Error while dialing dial tcp 172.19.83.237:9095: connectex: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.\""  
addr=172.19.83.237:9095  
level=error ts=2023-05-06T07:09:48.414947Z caller=scheduler_processor.go:86 msg="error contacting scheduler" err="rpc error: code = Unavailable desc = connection error: desc = \"transport: Error while dialing dial tcp 172.19.83.237:9095: connectex: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.\""  
addr=172.19.83.237:9095  
level=error ts=2023-05-06T07:09:48.447443Z caller=scheduler_processor.go:86 msg="error contacting scheduler" err="rpc error: code = Unavailable desc = connection error: desc = \"transport: Error while dialing dial tcp 172.19.83.237:9095: connectex: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.\""  
addr=172.19.83.237:9095  
level=error ts=2023-05-06T07:09:48.523981Z caller=scheduler_processor.go:86 msg="error contacting scheduler" err="rpc error: code = Unavailable desc = connection error: desc = \"transport: Error while dialing dial tcp 172.19.83.237:9095: connectex: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.\""  
addr=172.19.83.237:9095  
level=error ts=2023-05-06T07:09:48.554591Z caller=scheduler_processor.go:86 msg="error contacting scheduler" err="rpc error: code = Unavailable desc = connection error: desc = \"transport: Error while dialing dial tcp 172.19.83.237:9095: connectex: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.\""  
addr=172.19.83.237:9095
```

原因：

Windows电脑有多个网卡，而loki服务未指定当前网卡，默认使用了一个未生效的网卡

解决办法：

在安装目录（默认C:/AIRIOT）/loki下修改配置文件local-config.yaml，如下图：

```
common:
  path_prefix: /tmp/data/loki
  storage:
    s3:
      endpoint: minio:9000
      insecure: true
      region: es-patata-1
      bucketnames: test-logs
      access_key_id: root
      secret_access_key: SuperPassword1
      s3forcepathstyle: true
  # storage:
  #   filesystem:
  #     chunks_directory: /tmp/data/loki/chunks
  #     rules_directory: /tmp/data/loki/rules
  replication_factor: 1
  ring:
    instance_addr: 127.0.0.1 ←
    kvstore:
      store: inmemory

schema config:
```

平台迁移

linux系统

例如: 将/opt/app迁移至/home/app

1.停止平台

进入平台安装路径 例如/opt/app

```
cd /opt/app  
./uninstall.sh
```

2.平台迁移

```
cd /home  
mv /opt/app .
```

3.修改docker-compose.yml

将docker-compose.yml和docker-compose-operation.yml中的/opt/app替换为/home/app

4.启动平台

```
./runOperation.sh  
./run.sh
```

运维管理忘记密码

linux系统

删除运维数据文件

进入安装目录下operation/data 删除data目录下的文件

重启operation容器

```
docker restart operation
```

docker路径迁移

前提：迁移docker一般情况是/root所在磁盘空间不足，因为docker的默认文件位置在/var/lib/docker，和root在一个磁盘，所以要对docker进行迁移

停止docker

```
systemctl stop docker
```

迁移docker文件路径

注：假设从/var/lib/docker迁移至/home/docker，/home所在磁盘为一个比较大的磁盘

```
cd /var/lib
cp -r docker /home/docker // 将docker文件迁移至/home
mv docker docker-old // 备份docker文件
ln -s /home/docker /var/lib/docker // 创建链接，/var/lib/docker指向新的docker路径/home/docker
```

启动docker

```
systemctl start docker
```

平台正常启动之后，删除docker-old

```
cd /var/lib
rm -rf docker-old
```

!(INFO)

注：若遇到mqtt，mongo启动报错permission denied，解决办法：

1. 停止容器

```
docker stop mongo mongo-set mqtt
```

2. 删除容器

```
docker rm mongo mongo-set mqtt
```

3. 删除镜像

```
docker rmi mongo:4.2.12
docker rmi airiot/rabbitmq:3.8.3-management-alpine
```

4. 重新导入镜像

```
// 假设平台安装在/home/app,则镜像的位置在/home/app/lib/base
cd /home/app/lib/base
docker load < mongo.tar.gz
docker < load < rabbitmq.tar.gz
```

5.启动平台

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

linux安装docker后无法访问宿主机端口

问题描述： CentOS 8 系统。Docker安装后，Docker容器，在防火墙关闭状态、防火墙端口规则开放状态依然无法访问主机端口。可以ping通宿主机IP，但是curl提示unreachable

解决方法：

1. 开启IP地址转发 默认情况下 `firewalld`会禁止转发流量，可以执行 `firewall-cmd --query-masquerade` 查看状态，应该是no，请执行下面的命令永久开启转发。

```
firewall-cmd --add-masquerade --permanent
```

2. 重启防火墙

```
firewall-cmd --reload
```

3. 重启docker

```
systemctl restart docker
```

docker网段冲突

! INFO

错误信息： ERROR: could not find an available, non-overlapping IPv4 address pool among the defaults to assign to the network

解决方案

修改docker网卡默认网段

docker网卡默认网段一般为172.17.0.1/16

```
cd /etc/docker  
vi daemon.json
```

添加网段配置 {"bip": "172.20.0.1/16"}

```
systemctl daemon-reload  
systemctl restart docker
```

指定容器网桥

修改docker-compose.yml文件及docker-compose-operation.yml文件中的网桥配置 如：

```
web:  
  container_name: web  
  image: nginx:1.21.6-alpine  
  logging:  
    driver: json-file  
    options:  
      max-size: 100m  
      max-file: "1"  
  networks:  
    - backend  
  ports:  
    - 3030:80  
  restart: always  
  ulimits:
```

```
nproc: 4096
nofile:
  soft: 1024
  hard: 3072
volumes:
- /etc/localtime:/etc/localtime:ro
- /opt/dev/airiot/web/html:/usr/share/nginx/html
- /opt/dev/airiot/web/portal:/usr/share/nginx/portal
- /opt/dev/airiot/web/nginx.conf:/etc/nginx/nginx.conf

networks:
backend:
  ipam:
    config:
      - subnet: "10.11.0.0/24"
```

```
version: '3'

services:
  operation:
    container_name: operation
    image: airiot/operation:latest
    restart: always
    environment:
      - PROJECTNAME=dev
      - "CORS.ENABLE=true"
      - INTERNETACCESS=false
  networks:
    - backend
  ports:
    - "13031:10088"
  volumes:
    - /etc/localtime:/etc/localtime:ro
    - /var/run/docker.sock:/var/run/docker.sock:ro
    - /opt/dev/:/app/compose/
    - /opt/dev/airiot/web/html:/app/html
    - /opt/dev/operation/data:/data
    - /opt/dev/operation/log:/log
    - /opt/dev/operation/configs:/app/configs
    - /opt/dev/operation/build:/app/web
    - /opt/dev/operation/driverRepo:/driverRepo
  ulimits:
    nproc: 4096
    nofile:
      soft: 1024
      hard: 3072
  logging:
    driver: "json-file"
    options:
```

```
max-size: 100m
max-file: "1"

networks:
  backend:
    ipam:
      config:
        - subnet: 10.10.0.0/24
```

启动平台

```
./runOperation.sh
./run.sh
```

Rocky Linux 系统安装平台后，服务一直重启

问题描述： 在 Rocky Linux 系统安装平台后，服务报错，运维页面打不开

原因： 容器操作文件无权限

解决方法：

1. 添加安全策略配置 在docker-compose.yml文件中，服务节点下添加SELinux配置"security_opt"，并指定"label=disable"

```
version: '3'
services:
  myapp:
    image: myapp-image
    security_opt:
      - label=disable
```

2. 重启平台

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

too many open files

描述

http: Accept error: accept tcp [::]:9000: accept: too many open files; retrying in 1s

解决

修改docker-compose.yml文件unlimit字段的soft值，改大一点

临时修改 fs.inotify.max_user_instances

```
sysctl -w fs.inotify.max_user_instances=1024
```

若要用就修改，需要添加到系统配置中

```
vi /etc/sysctl.conf  
fs.inotify.max_user_instances=1024
```

Influx查询数据

1. 进入容器

```
docker exec -it influx sh
```

2. 连接influx

```
influx -username admin -password 密码
```

3. 连接tsdb数据库

```
use tsdb
```

4. 查询数据

```
select 数据点标识,id from "项目id_模型id" order by time limit 10
```

```
[root@iZ8vb396j1yl30zflw5x9qZ ~]# docker exec -it influx sh
/ # ^C
/ #
/ #
/ # influx -username admin -password dell123
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
> use tsdb
Using database tsdb
> select cpuUtil,id from "62393c36df61c0d066fa0e21_62eb88e5d2b5f9ddc5bfa9e2" order by time limit 10
name: 62393c36df61c0d066fa0e21_62eb88e5d2b5f9ddc5bfa9e2
time          cpuUtil id
-----
1666174210881000000 13      snmp-node1
1666174220884000000 13      snmp-node1
1666174230927000000 13      snmp-node1
1666174240886000000 13      snmp-node1
1666174250889000000 13      snmp-node1
1666174260895000000 13      snmp-node1
1666174270895000000 13      snmp-node1
16661742808990000000 13      snmp-node1
1666174290900000000 13      snmp-node1
1666174300948000000 13      snmp-node1
> |
```

启用HTTPS访问

如果想要给平台添加 `https` 的访问方式, 可以按以下步骤操作:

1. 准备证书

可以使用自签名的证书或向相关机构申请安全证书.

证书包含以下 2 个文件:

- <文件>.key
- <文件>.pem

使用自签名的证书时, 浏览器会有不安全访问的相关提示.

2. 修改平台配置

1. 上传证书

将 [步骤1](#) 中得到的文件上传到平台服务器上.

可以将证书文件放置在 `/平台安装根目录/airiot/web/ssl` 目录下

2. 修改配置

修改 `web` 服务的 `nginx.conf` 配置文件. 该文件的路径通常为 `/平台安装根目录/airiot/web/nginx.conf`. 使用 `vim` 或其它编辑工具打开该文件, 按照以下步骤进行操作:

1. 将现有端口修改为https

如果要将现有端口修改为 `https` 的访问方式, 需要修改 `nginx.conf` 中的 `80` 端口对应的 `Server` 内容, 添加 `ssl` 相关配置.

注: 需要注意标有注释的内容.

```
server {
    # 将端口号改为 443 并启用 ssl
    listen      443 ssl;
    # 访问域名, 多个域名以空格分开
```

```

server_name localhost;

# 证书相关配置，具体路径为实际挂载到容器中的路径
ssl_certificate      /usr/local/nginx/cert/<文件名>.pem;    # pem文件的路径
ssl_certificate_key  /usr/local/nginx/cert/<文件名>.key;  # key文件的路径

# ssl验证相关配置
ssl_session_timeout  5m;        #缓存有效期
ssl_ciphers ECDHE-RSA-AES128-GCM-
SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;    #加密算法
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;    #安全链接可选的加密协议
ssl_prefer_server_ciphers on;    #使用服务器端的首选算法

# 其余配置保持不变即可
}

```

2. 添加新端口为https

如果想同时允许 `http` 和 `https` 两种访问方式时，需要额外添加一套平台配置。步骤如下：

- 将 `80` 端口的 `Server` 配置复制一份
- 按前面的 [步骤](#) 修改新复制的 `Server`.

```

# 原有的配置
server {
    listen      80;
    server_name localhost;
    keepalive_timeout 180;
    client_header_timeout 180;
    send_timeout 180;
    client_body_timeout 180;
    client_max_body_size 300000M;
    proxy_connect_timeout 3600s;
    proxy_send_timeout 3600s;
    proxy_read_timeout 3600s;
    client_header_buffer_size 512k;
    large_client_header_buffers 4 512k;
    charset utf-8;

    # 其它内容
}

```

```

# 新复制并修改后的配置
server {
    # 将端口号改为 443 并启用 ssl
    listen      443 ssl;
    # 访问域名，多个域名以空格分开
}

```

```

server_name localhost;

# 证书相关配置，具体路径为实际挂载到容器中的路径
ssl_certificate      /usr/local/nginx/cert/<文件名>.pem;    # pem文件的路径
ssl_certificate_key  /usr/local/nginx/cert/<文件名>.key;  # key文件的路径

# ssl验证相关配置
ssl_session_timeout  5m;        #缓存有效期
ssl_ciphers ECDHE-RSA-AES128-GCM-
SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;      #加密算法
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;    #安全链接可选的加密协议
ssl_prefer_server_ciphers on;    #使用服务器端的首选算法

# 其余配置保持不变即可
}

```

3. 挂载证书文件并修改相关端口

修改 `docker-compose.yml` 文件, 作如下修改:

1. 将证书挂载到 `web` 容器中.
2. 修改映射端口.

```

web:
  container_name: web
  image: nginx:alpine
  logging:
    driver: json-file
    options:
      max-file: "1"
      max-size: 100m
  networks:
    - backend
    - operation
  ports:
    # 原访问端口, http 协议
    - 3030:80
    # 新增端口, https 协议. 例如想修改外部访问端口为 8443 时, 可修改为 8443:443
    - 443:443
  restart: always
  ulimits:
    nofile:
      hard: 3072
      soft: 1024
    nproc: 4096
  volumes:
    - /etc/localtime:/etc/localtime:ro

```

```
- /opt/app/airiot/web/html:/usr/share/nginx/html  
# 添加该行内容, 将存放证书的目录挂载到容器中. 需要保证这里挂载到容器内的路径与前面 nginx.conf 文件中的证书路径保持一致.  
# 格式为: 本地存放证书的目录:/usr/local/nginx/cert/  
- /opt/app/airiot/web/ssl:/usr/local/nginx/cert/  
- /opt/app/airiot/web/nginx.conf:/etc/nginx/nginx.conf  
- /opt/app/airiot/web/webCfg:/etc/nginx/webCfg  
- /opt/app/airiot/web/portal:/usr/share/nginx/portal
```

4. 重启相关服务

进入到平台安装目录下(即 `docker-compose.yml` 文件所在目录), 执行以下命令重启相关服务, 重启成功后上述修改后效.

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

5. 检查是否修改成功

执行完上述重启命令后, 使用以下命令确认 `web` 是否正常启动

```
docker ps -a | grep web
```

如果 `web` 的状态为停止状态, 表示配置有误, 可通过以下命令查看日志, 并根据日志内容作相关的调整.

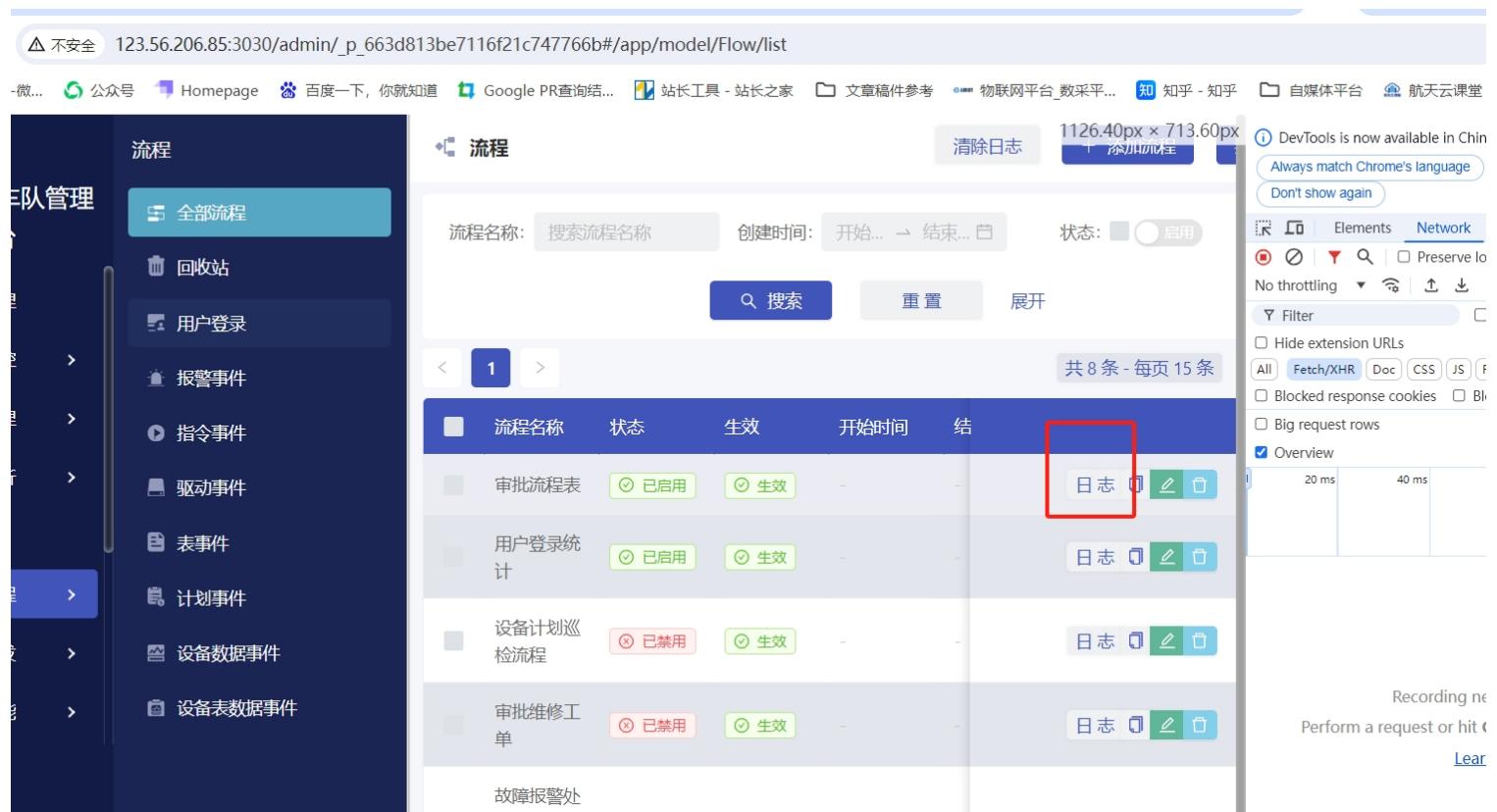
```
docker logs web
```

如果 `web` 正常启动后, 即可通过 `https` 访问平台

流程日志输出与流程变量绑定关系查找

流程输出的值和格式都可以在流程日志里面看到，选择绑定一个字段后，也可以在流程日志里面看到。两者可以对应。

1.点击日志



The screenshot shows a web interface for managing processes. On the left, there's a sidebar with various event categories like User Login, Alarm Events, and Plan Events. The main area is titled '流程' (Processes) and lists several processes with columns for name, status, effectiveness, start time, end time, and a 'Logs' button. The 'Logs' button for the first process, '审批流程表', is highlighted with a red box. The interface includes a search bar, filter options, and a network tab in the developer tools.

流程名称	状态	生效	开始时间	结束时间	操作
审批流程表	已启用	生效	-	-	日志
用户登录统计	已启用	生效	-	-	日志
设备计划巡检流程	已禁用	生效	-	-	日志
审批维修工单	已禁用	生效	-	-	日志
故障报警处					

2.查看配置好流程后触发的一条流程日志

△ 不安全 123.56.206.85:3030/admin/_p_663d813be7116f21c747766b#/app/model/Flow/667d2e52c099c29a1cf960bf/record

路况官网-微... 公众号 Homepage 百度一下，你就知道 Google PR查询结... 站长工具 - 站长之家 文章稿件参考 物联网平台_数采平... 知乎 - 知乎 自媒体平台 航天云

智能车队管理平台

- 数据管理
- 设备监控
- 报警管理
- 数据分析
- 可视化
- 业务流程 >
- 扩展开发
- 通用功能
- 消息

审批流程表流程日志

触发时间 2024-06-27 11:26:28 → 2024-06-28 11:26:28

搜索 重置 13条 每页15条 查询数量

触发流程数据	状态	终止/失败原因	触发时间	查看
表记录增加 - 表...	已完成		2024-06-28 11:0...	查看
表记录增加 - 表...	已完成		2024-06-28 10:5...	查看
表记录增加 - 表...	已完成		2024-06-28 10:5...	查看
表记录增加 - 表...	已完成		2024-06-28 10:4...	查看
表记录增加 - 表...	已完成		2024-06-28 10:4...	查看

DevTools is now available in Chinese!
Always match Chrome's language | Switch DevTools to Chinese | Don't show again

Elements Network
No throttling | Filter
Hide extension URLs | All Fetch/XHR Doc CSS
Blocked response cookies
Big request rows
Overview
20 ms 40 ms

3.之后会出来两个接口，上面的里面是流程配置，下面的是流程本次执行的输出数据

△ 不安全 123.56.206.85:3030/admin/_p_663d813be7116f21c747766b#/app/model/Flow/202e787e-1056-464c-a65d-909bcf21e536/detail/2...
35编辑器官网-微... 公众号 Homepage 百度一下，你就知道 Google PR查询结... 站长工具 - 站长之家 文章稿件参考 物联网平台_数采平... 知乎 - 知乎 自媒体平台 航天云课堂

智能车队管理平台

- 数据管理
- 设备监控
- 报警管理
- 数据分析
- 可视化
- 业务流程 >
- 扩展开发
- 通用功能
- 消息
- 帮助中心
- admin

审批流程表节点日志

941.60px × 713.60px

已完成
数据: 表记录增加 - 表名(工作审批表) 记录id: 667e286b5dd8cca2d56150cd

执行日志 流转详情

触发流程数据	节点类型	状态	进入时间
表记录增加 - 表名(工作审批表) 记录id: 667e286b5dd8cca2d56150cd	表事件	已完成	2024-06-28 11:26:28
admin对工作审批表进行审批，审批通过	审批	已完成	2024-06-28 11:26:28
【工作审批表】更新记录 更新1条记录	更新记录	已完成	2024-06-28 11:26:28

DevTools is now available in Chinese!
Always match Chrome's language | Switch DevTools to Chinese | Don't show again

Elements Network
Preserve log | Disable cache | No throttling | Filter
Invert | Hide data URLs | Hide extension URLs
All Fetch/XHR Doc CSS JS Font Img Media Manifest WS Wasm Other
Blocked response cookies | Blocked requests | 3rd-party requests
Big request rows | Group by frame | Screenshots
Overview
Name Status Type Initiator Size Time
667d2e52c099c29a1cf960bf 200 fetch index.js?version=4.2 3.5 kB 22 ms
job?query=%7B%22filter%2... 200 fetch index.js?version=4.2 2.3 kB 38 ms

2 / 3 requests | 5.8 kB / 13.5 kB transferred | 22.8 kB / 30.3 kB resources

4.publishFlow里是配置的所有流程处理节点，找到绑定了字段的那个流程节点，{}包起来的就是绑定字段，然后例如图中，按Flow_0F5A03E4.select-27B5这个路径去流程输出数据接口里面去找对应的值

13.56.206.85:3030/admin/_p_663d813be7116f21c747766b#/app/model/Flow/667d2e52c099c29a1cf960bf/record/202e787e-1056-464c-a65d-909bcf21e536/detail/2...

Homepage 百度一下，你就知道 Google PR查询结... 站长工具 - 站长之家 文章稿件参考 物联网平台_数采平... 知乎 - 知乎 自媒体平台 航天云课堂 所有书签

审批流程表节点日志

已完成

数据: 表记录增加 - 表名(工作审批表) 记录id: 667e286b5dd8cca2d56150cd

执行日志	流转详情
触发流程数据	节点类型 状态
表记录增加 - 表名(工作审批表) 记录id: 667e286b5dd8cca2d56150cd	表事件 已完成
admin对工作审批表进行审批, 审批通过	审批 已完成
【工作审批表】更新记录 更新1条记录	更新记录 已完成

DevTools is now available in Chinese! Always match Chrome's language Switch DevTools to Chinese Don't show again

Network tab in DevTools showing a request for "publishFlow". The response payload is highlighted with a red box, showing the JSON structure of the flow record.

```

{
  "name": "审批流程表",
  "notSavable": false,
  "publishFlow": [
    {
      "id": "Flow_324124D5",
      "type": "flowEnd"
    }
  ],
  ...
}
  
```

5.去最上面的那个里面找

表名(工作审批表) 记录id: 667e286b5dd8cca2d56150cd

节点类型	状态
表事件	已完成
审批, 审批通过	已完成
录 更新1条记录	已完成

Network tab in DevTools showing a request for "select-27B5". The response payload is highlighted with a red box, showing the JSON structure of the selected record.

```

{
  "id": "007e286b52e44bc4514f5547",
  "flowId": "667d2e52c099c29a1cf960bf",
  ...
}
  
```

6.Flow_OF5A03E4.select-27B5 值对应的就是那个绑定字段的值

DevTools is now available in Chinese! [Always match Chrome's language](#) [Switch DevTools to Chinese](#) [Don't show again](#)

Elements Console Sources Network Performance Memory > 3 323 1

Filter Invert Hide data URLs Hide extension URLs

All Fetch/XHR Doc CSS JS Font Img Media Manifest WSS Wasm Other Blocked response cookies

Blocked requests 3rd-party requests

Big request rows Group by frame

Overview Screenshots

节点类型	状态
表事件	已完成
审批	已完成
更新记录	已完成

Name Headers Payload Preview Response Initiator Timing

```

667d2e52c099c29a1cf960bf
job?query=%7B%22filter%22%3A...
log?query=%22sort%22(%22tim...
log?query=%22sort%22(%22tim...
log?query=%22sort%22(%22tim...
log?query=%22sort%22(%22tim...
log?query=%22sort%22(%22tim...
667d2e52c099c29a1cf960bf: [
  #table: {tableName: "table", id: "工作审批表", title: "工作审批表", createT...
  #table: {tableName: "table", id: "工作审批表", title: "工作审批表", createTime: "2024-06-28T11:05:15.428987018+08:00", creator: "admin", creatorName: "admin", date-813B: "2024-06-28", extFlowType: "表记录增加", extUserMap: {creator: {$user: {tableName: "user", id: "admin", name: "admin"}}, flowTriggerUser: "admin", flowTriggerUserMap: {$user: {tableName: "user", id: "admin", name: "admin"}}, id: "667e286b5dd8cca2d56150cd", relate-table-83CF: {disabled: false, email: "18511086929@163.com", id: "admin", na...
  select-27B5: null, text-C339: "电脑采购审批", _triggerType__: "user", _user__: "admin", _table: "工作审批表"
]

```

10 / 11 requests | 11.9 kB / 19.7 kB transferred

Console What's new

Highlights from the Chrome 126 update

跨域配置

跨域可以通过配置nginx的方案解决，配置后通过 `http://ip:3030/rest` 访问。

跨域配置

在 Nginx 中配置跨域资源共享 (CORS, Cross-Origin Resource Sharing) 允许不同来源的客户端访问你的资源。通过正确配置 HTTP 响应头，Nginx 可以让浏览器允许跨域请求。以下是配置跨域的几种常见方案：

`nginx.conf` 文件所在位置: `app/airiot/web/nginx.conf`。

跨域配置，在一些环境实践，可以只配置 `OPTIONS` 部分：

```
add_header 'Access-Control-Allow-Origin' '*' always;
add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE' always;
add_header 'Access-Control-Allow-Headers' 'Origin, *' always;
add_header 'Access-Control-Allow-Credentials' 'true' always;

if ($request_method = 'OPTIONS') {
    # 添加 CORS 头，确保只设置一次
    add_header 'Access-Control-Allow-Origin' '*';
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE';
    add_header 'Access-Control-Allow-Headers' 'Origin, *';
    add_header 'Access-Control-Allow-Credentials' 'true';
    return 204;
}
```

1. 单项目版本参考

可以在 `nginx.conf` 中添加如下配置：

```
user    nginx;
worker_processes  auto;

error_log  /var/log/nginx/error.log warn;
pid        /var/run/nginx.pid;

events {
    worker_connections  100000;
}
```

```
http {
    include      /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile      on;
    #tcp_nopush   on;

    keepalive_timeout  65;
    gzip on;
    gzip_min_length 1k;
    gzip_buffers 4 16k;
    gzip_comp_level 8;
    gzip_types text/plain application/javascript application/x-javascript text/css
application/xml text/javascript application/x-httdp-php image/jpeg image/gif image/png;
    gzip_vary off;
    gzip_disable "MSIE [1-6]\.";

    fastcgi_intercept_errors on;
    proxy_intercept_errors on;
    server_tokens off;
    server {
        listen      80;
        server_name localhost;

        keepalive_timeout 180;
        client_header_timeout 180;
        client_body_timeout 180;
        client_max_body_size 10240M;
        client_header_buffer_size 512k;
        large_client_header_buffers 4 512k;
        charset utf-8;

        error_page 404 /404.html;

        location /404.html {
            root   /usr/share/nginx/html;
            try_files $uri /404.html;
        }

        location /404 {
            add_header 'Access-Control-Allow-Origin' '*' always;
            add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE'
            always;
        }
    }
}
```

```
add_header 'Access-Control-Allow-Headers' 'Origin, *' always;
add_header 'Access-Control-Allow-Credentials' 'true' always;

if ($request_method = 'OPTIONS') {
    # 添加 CORS 头, 确保只设置一次
    add_header 'Access-Control-Allow-Origin' '*';
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE';
    add_header 'Access-Control-Allow-Headers' 'Origin, *';
    add_header 'Access-Control-Allow-Credentials' 'true';
    return 204;
}
proxy_pass http://traefik:80/front/404;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Real-Port $remote_port;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}

location /rest/front {
    add_header 'Access-Control-Allow-Origin' '*' always;
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE'
always;
    add_header 'Access-Control-Allow-Headers' 'Origin, *' always;
    add_header 'Access-Control-Allow-Credentials' 'true' always;

    if ($request_method = 'OPTIONS') {
        # 添加 CORS 头, 确保只设置一次
        add_header 'Access-Control-Allow-Origin' '*';
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE';
        add_header 'Access-Control-Allow-Headers' 'Origin, *';
        add_header 'Access-Control-Allow-Credentials' 'true';
        return 204;
    }
    proxy_pass http://traefik:80/front;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Real-Port $remote_port;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}

location ^~ /rest/static {
    alias /usr/share/nginx/html/static/;
}

location /rest/ {
    set $http_method $request_method;
    if ($http_x_request_http_method = "DELETE") {
        set $http_method $http_x_request_http_method;
    }
    if ($http_x_request_http_method = "PUT") {
        set $http_method $http_x_request_http_method;
    }
}
```

```

    }

    if ($http_x_request_http_method = "PATCH") {
        set $http_method $http_x_request_http_method;
    }

    add_header 'Access-Control-Allow-Origin' '*' always;
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE'
always;
    add_header 'Access-Control-Allow-Headers' 'Origin, *' always;
    add_header 'Access-Control-Allow-Credentials' 'true' always;

    if ($request_method = 'OPTIONS') {
        # 添加 CORS 头, 确保只设置一次
        add_header 'Access-Control-Allow-Origin' '*';
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE';
        add_header 'Access-Control-Allow-Headers' 'Origin, *';
        add_header 'Access-Control-Allow-Credentials' 'true';
        return 204;
    }

    add_header X-Content-Type-Options nosniff;
    add_header X-XSS-Protection 1;
    add_header Content-Security-Policy "default-src 'self'; style-src * 'unsafe-
inline'; img-src * data:; object-src 'self'; script-src * 'unsafe-eval' 'unsafe-inline';
font-src * data:; worker-src * blob:;";
    add_header Referrer-Policy value;
    add_header X-Permitted-Cross-Domain-Policies value;
    add_header X-Download-Options value;
    add_header X-Frame-Options SAMEORIGIN;
    add_header Strict-Transport-Security value;
    proxy_method $http_method;
    proxy_pass  http://traefik:80/;
    proxy_read_timeout 300;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Real-Port $remote_port;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $http_host;
    error_page 404 /404;
}

location /ws {
    add_header 'Access-Control-Allow-Origin' '*' always;
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE'
always;
    add_header 'Access-Control-Allow-Headers' 'Origin, *' always;
    add_header 'Access-Control-Allow-Credentials' 'true' always;

    if ($request_method = 'OPTIONS') {
        # 添加 CORS 头, 确保只设置一次
        add_header 'Access-Control-Allow-Origin' '*';
    }
}

```

```
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE';
        add_header 'Access-Control-Allow-Headers' 'Origin, *';
        add_header 'Access-Control-Allow-Credentials' 'true';
        return 204;
    }
    proxy_pass    http://traefik:80/core/ws;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
}

location / {
    root    /usr/share/nginx/html;
    index  index.html index.htm;
    #try_files $uri /index.html;
    try_files $uri /rest/front/static/index;
}

location /admin {
    root    /usr/share/nginx/html;
    index  index.html index.htm;
    #try_files $uri /index.html;
    try_files $uri /rest/front/static/admin;
}

location ~ /ISAPI|SDK/ {
    if ($http_cookie ~ "webVideoCtrlProxy=(.+)") {
        proxy_pass http://$cookie_webVideoCtrlProxy;
        break;
    }
}

location ^~ /WebSocketVideoCtrlProxy {
    #web socket
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host $host;

    if ($http_cookie ~ "webVideoCtrlProxyWs=(.+)") {
        proxy_pass
http://$cookie_webVideoCtrlProxyWs/$cookie_webVideoCtrlProxyWsChannel?$args;
        break;
    }
    if ($http_cookie ~ "webVideoCtrlProxyWss=(.+)") {
        proxy_pass
http://$cookie_webVideoCtrlProxyWss/$cookie_webVideoCtrlProxyWsChannel?$args;
        break;
    }
}
```

```
location /video/ {
    add_header 'Access-Control-Allow-Origin' '*' always;
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE'
always;
    add_header 'Access-Control-Allow-Headers' 'Origin, *' always;
    add_header 'Access-Control-Allow-Credentials' 'true' always;

    if ($request_method = 'OPTIONS') {
        # 添加 CORS 头, 确保只设置一次
        add_header 'Access-Control-Allow-Origin' '*';
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE';
        add_header 'Access-Control-Allow-Headers' 'Origin, *';
        add_header 'Access-Control-Allow-Credentials' 'true';
        return 204;
    }

    proxy_pass  http://traefik:80/;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    error_page 404 /404;
}

location /ws/lsp/ {
    add_header 'Access-Control-Allow-Origin' '*' always;
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE'
always;
    add_header 'Access-Control-Allow-Headers' 'Origin, *' always;
    add_header 'Access-Control-Allow-Credentials' 'true' always;

    if ($request_method = 'OPTIONS') {
        # 添加 CORS 头, 确保只设置一次
        add_header 'Access-Control-Allow-Origin' '*';
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE';
        add_header 'Access-Control-Allow-Headers' 'Origin, *';
        add_header 'Access-Control-Allow-Credentials' 'true';
        return 204;
    }

    proxy_pass  http://traefik:80/;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    error_page 404 /404;
}

}
```

2. 空间版本参考

可以在 `nginx.conf` 中添加如下配置：

```
user    nginx;
worker_processes  auto;

error_log  /var/log/nginx/error.log warn;
pid        /var/run/nginx.pid;

events {
    worker_connections  100000;
}

http {
    include      /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format  main  '$remote_addr - $proxy_add_x_forwarded_for- $remote_user [$time_local]
"$request" '
                    '$status $body_bytes_sent "$http_referer" '
                    '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile      on;
    #tcp_nopush   on;

    keepalive_timeout  65;
    gzip on;
    gzip_min_length 1k;
    gzip_buffers 4 16k;
    gzip_comp_level 8;
    gzip_types text/plain application/javascript application/x-javascript text/css
application/xml text/javascript application/x-httpd-php image/jpeg image/gif image/png;
    gzip_vary off;
    gzip_disable "MSIE [1-6]\.";

    fastcgi_intercept_errors on;
    proxy_intercept_errors on;
    server_tokens off;
    server {
        listen      80;
        server_name localhost;
        keepalive_timeout 180;
        client_header_timeout 180;
        client_body_timeout 180;
        client_max_body_size 10240M;
        client_header_buffer_size 512k;
        large_client_header_buffers 4 512k;
    }
}
```

```
charset utf-8;

error_page 404 /404.html;

location /404.html {
    root /usr/share/nginx/html;
    try_files $uri /404.html;
}

location /404 {
    add_header 'Access-Control-Allow-Origin' '*' always;
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE'
always;
    add_header 'Access-Control-Allow-Headers' 'Origin, *' always;
    add_header 'Access-Control-Allow-Credentials' 'true' always;

    if ($request_method = 'OPTIONS') {
        # 添加 CORS 头, 确保只设置一次
        add_header 'Access-Control-Allow-Origin' '*';
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE';
        add_header 'Access-Control-Allow-Headers' 'Origin, *';
        add_header 'Access-Control-Allow-Credentials' 'true';
        return 204;
    }
    proxy_pass http://traefik:80/front/404;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Real-Port $remote_port;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}

location /portal/ {
    alias /usr/share/nginx/portal/;
    index index.html index.htm;
    try_files $uri /index.html;
}

location /rest/front {
    add_header 'Access-Control-Allow-Origin' '*' always;
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE'
always;
    add_header 'Access-Control-Allow-Headers' 'Origin, *' always;
    add_header 'Access-Control-Allow-Credentials' 'true' always;

    if ($request_method = 'OPTIONS') {
        # 添加 CORS 头, 确保只设置一次
        add_header 'Access-Control-Allow-Origin' '*';
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE';
        add_header 'Access-Control-Allow-Headers' 'Origin, *';
        add_header 'Access-Control-Allow-Credentials' 'true';
        return 204;
    }
}
```

```
        }
        proxy_pass    http://traefik:80/front;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Real-Port $remote_port;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
location ^~ /rest/static {
    alias /usr/share/nginx/html/static/;
}
location /portal/rest/ {
    set $http_method $request_method;
    if ($http_x_request_http_method = "DELETE") {
        set $http_method $http_x_request_http_method;
    }
    if ($http_x_request_http_method = "PUT") {
        set $http_method $http_x_request_http_method;
    }
    if ($http_x_request_http_method = "PATCH") {
        set $http_method $http_x_request_http_method;
    }

    add_header 'Access-Control-Allow-Origin' '*' always;
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE'
always;
    add_header 'Access-Control-Allow-Headers' 'Origin, *' always;
    add_header 'Access-Control-Allow-Credentials' 'true' always;

    if ($request_method = 'OPTIONS') {
        # 添加 CORS 头, 确保只设置一次
        add_header 'Access-Control-Allow-Origin' '*';
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE';
        add_header 'Access-Control-Allow-Headers' 'Origin, *';
        add_header 'Access-Control-Allow-Credentials' 'true';
        return 204;
    }
    add_header X-Content-Type-Options nosniff;
    add_header X-XSS-Protection 1;
    add_header Content-Security-Policy "default-src 'self'; style-src * 'unsafe-
inline'; img-src * data:; object-src 'self'; script-src * 'unsafe-eval' 'unsafe-inline';
font-src * data:; worker-src * blob:;";
    add_header Referrer-Policy value;
    add_header X-Permitted-Cross-Domain-Policies value;
    add_header X-Download-Options value;
    add_header X-Frame-Options SAMEORIGIN;
    add_header Strict-Transport-Security value;
    proxy_method $http_method;

    proxy_pass    http://traefik:80/;
    proxy_read_timeout 300;
    proxy_set_header X-Real-IP $remote_addr;
```

```
proxy_set_header X-Real-Port $remote_port;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header Host $http_host;
error_page 404 /404;
}

location /rest/ {
    set $http_method $request_method;
    if ($http_x_request_http_method = "DELETE") {
        set $http_method $http_x_request_http_method;
    }
    if ($http_x_request_http_method = "PUT") {
        set $http_method $http_x_request_http_method;
    }
    if ($http_x_request_http_method = "PATCH") {
        set $http_method $http_x_request_http_method;
    }
    add_header X-Content-Type-Options nosniff;
    add_header X-XSS-Protection 1;
    add_header Content-Security-Policy "default-src 'self'; style-src * 'unsafe-inline'; img-src * data:; object-src 'self'; script-src * 'unsafe-eval' 'unsafe-inline'; font-src * data:; worker-src * blob:";
    add_header Referrer-Policy value;
    add_header X-Permitted-Cross-Domain-Policies value;
    add_header X-Download-Options value;
    add_header X-Frame-Options SAMEORIGIN;
    add_header Strict-Transport-Security value;
    proxy_method $http_method;
    add_header 'Access-Control-Allow-Origin' '*' always;
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE' always;
    add_header 'Access-Control-Allow-Headers' 'Origin, *' always;
    add_header 'Access-Control-Allow-Credentials' 'true' always;

    if ($request_method = 'OPTIONS') {
        # 添加 CORS 头，确保只设置一次
        add_header 'Access-Control-Allow-Origin' '*';
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE';
        add_header 'Access-Control-Allow-Headers' 'Origin, *';
        add_header 'Access-Control-Allow-Credentials' 'true';
        return 204;
    }

    proxy_pass http://traefik:80/;
    proxy_read_timeout 300;
    # proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Real-Port $remote_port;
    # proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Real-IP $remote_addr;
```

```
proxy_set_header Host $http_host;
error_page 404 /404;
}

location /ws {
    add_header 'Access-Control-Allow-Origin' '*' always;
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE'
always;
    add_header 'Access-Control-Allow-Headers' 'Origin, *' always;
    add_header 'Access-Control-Allow-Credentials' 'true' always;

    if ($request_method = 'OPTIONS') {
        # 添加 CORS 头, 确保只设置一次
        add_header 'Access-Control-Allow-Origin' '*';
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE';
        add_header 'Access-Control-Allow-Headers' 'Origin, *';
        add_header 'Access-Control-Allow-Credentials' 'true';
        return 204;
    }
    proxy_pass http://traefik:80/core/ws;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
}

location = / {
    rewrite ^/(.*) portal/index.html redirect;
}

location / {
    root /usr/share/nginx/html;
    index index.html index.htm;
    #try_files $uri /index.html;
    try_files $uri /rest/front/static/index;
}

location = /admin {
    return 404;
}

location /admin {
    root /usr/share/nginx/html;
    index index.html index.htm;
    #try_files $uri /index.html;
    try_files $uri /rest/front/static/admin;
}

location ~ /ISAPI|SDK/ {
    if ($http_cookie ~ "webVideoCtrlProxy=(.+)") {
```

```
        proxy_pass http://$cookie_webVideoCtrlProxy;
        break;
    }
}

location ^~ /webSocketVideoCtrlProxy {
    #web socket
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host $host;

    if ($http_cookie ~ "webVideoCtrlProxyWs=(.+)") {
        proxy_pass
http://$cookie_webVideoCtrlProxyWs/$cookie_webVideoCtrlProxyWsChannel?$args;
        break;
    }
    if ($http_cookie ~ "webVideoCtrlProxyWss=(.+)") {
        proxy_pass
http://$cookie_webVideoCtrlProxyWss/$cookie_webVideoCtrlProxyWsChannel?$args;
        break;
    }
}

location /video/ {
    add_header 'Access-Control-Allow-Origin' '*' always;
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE'
always;
    add_header 'Access-Control-Allow-Headers' 'Origin, *' always;
    add_header 'Access-Control-Allow-Credentials' 'true' always;

    if ($request_method = 'OPTIONS') {
        # 添加 CORS 头, 确保只设置一次
        add_header 'Access-Control-Allow-Origin' '*';
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE';
        add_header 'Access-Control-Allow-Headers' 'Origin, *';
        add_header 'Access-Control-Allow-Credentials' 'true';
        return 204;
    }
    proxy_pass  http://traefik:80/;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    error_page 404 /404;
}

location /ws/lsp/ {
    add_header 'Access-Control-Allow-Origin' '*' always;
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE'
always;
```

```
add_header 'Access-Control-Allow-Headers' 'Origin, *' always;
add_header 'Access-Control-Allow-Credentials' 'true' always;

if ($request_method = 'OPTIONS') {
    # 添加 CORS 头, 确保只设置一次
    add_header 'Access-Control-Allow-Origin' '*';
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE';
    add_header 'Access-Control-Allow-Headers' 'Origin, *';
    add_header 'Access-Control-Allow-Credentials' 'true';
    return 204;
}
proxy_pass    http://traefik:80/;
proxy_http_version 1.1;
proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection "upgrade";
error_page   404 /404;
}
}
include ./webCfg/*.conf;
}
```

通过这些配置，你可以在 Nginx 中灵活地设置跨域访问规则，确保你的应用能够处理不同来源的请求。

关于AIRIOT3.0升级到4.0的说明

感谢您使用AIRIOT平台！AIRIOT 3.0现已升级至AIRIOT 4.0版本。本次升级带来了多项重要改进和新功能，以提升您的使用体验和平台性能。现在对升级步骤进行说明，按此说明即可升级到4.0版本中。

升级步骤

步骤一：下载4.0安装包

访问AIRIOT官方网站或授权渠道，进入4.0版本下载专区，找到AIRIOT 4.0升级包。

下载安装包到您的服务器或本地计算机并进行安装。

The screenshot shows the AIRIOT website's 'Version Download' page. At the top, there is a navigation bar with links for Product, Case Studies, Download, Cooperation, Resource Center, University Program, and About Us, along with a phone number 010-84860985. Below the navigation bar, there is a large button labeled 'Apply for Trial Authorization'. To the right of this button, there is a link 'Historical Versions ->'. The main content area is a table listing download links for various system configurations:

System	Version	Supported System Version	Update Time	Download Link
Linux and Mac amd64 普通v4.0版本	20240719	Linux、Mac	2024-07-19	Download
Linux and Mac amd64 空间v4.0版本	20240719	Linux、Mac	2024-07-19	Download
Linux and Mac aarch64 普通v4.0版本	20240719	Linux、Mac	2024-07-19	Download
Linux and Mac aarch64 空间v4.0版本	20240719	Linux、Mac	2024-07-19	Download
Windows 普通v4.0版本	20240523	Windows	2024-05-23	Download

步骤二：4.0平台中安装 @airiot/v3patch模块

由于3.0版本中的部分组件在4.0版本中已优化了代码结构，直接升级可能导致这些组件无法显示。为确保这些组件在4.0版本平台上正常显示，需要安装@airiot/v3patch。

1. 获取@airiot/v3patch

可复制链接到浏览器中下载@airiot/v3patch，链接：

https://d.airiot.link/front/@airiot/v3patch/@airiot_v3patch-1.0.1.tgz。

2. 安装@airiot/v3patch

进入运维管理平台，在模块管理中，离线上传下载的@airiot/v3patch文件。

步骤三：3.0平台中安装dashboard-upgrade、migration服务

由于3.0版本中的部分组件在4.0版本中已优化了代码结构，直接升级可能导致这些组件无法显示。为确保这些组件在4.0版本平台上正常显示，需要安装dashboard-upgrade、migration服务。

1. 获取dashboard-upgrade、migration服务

可进入官网下载dashboard-upgrade、migration，链接：

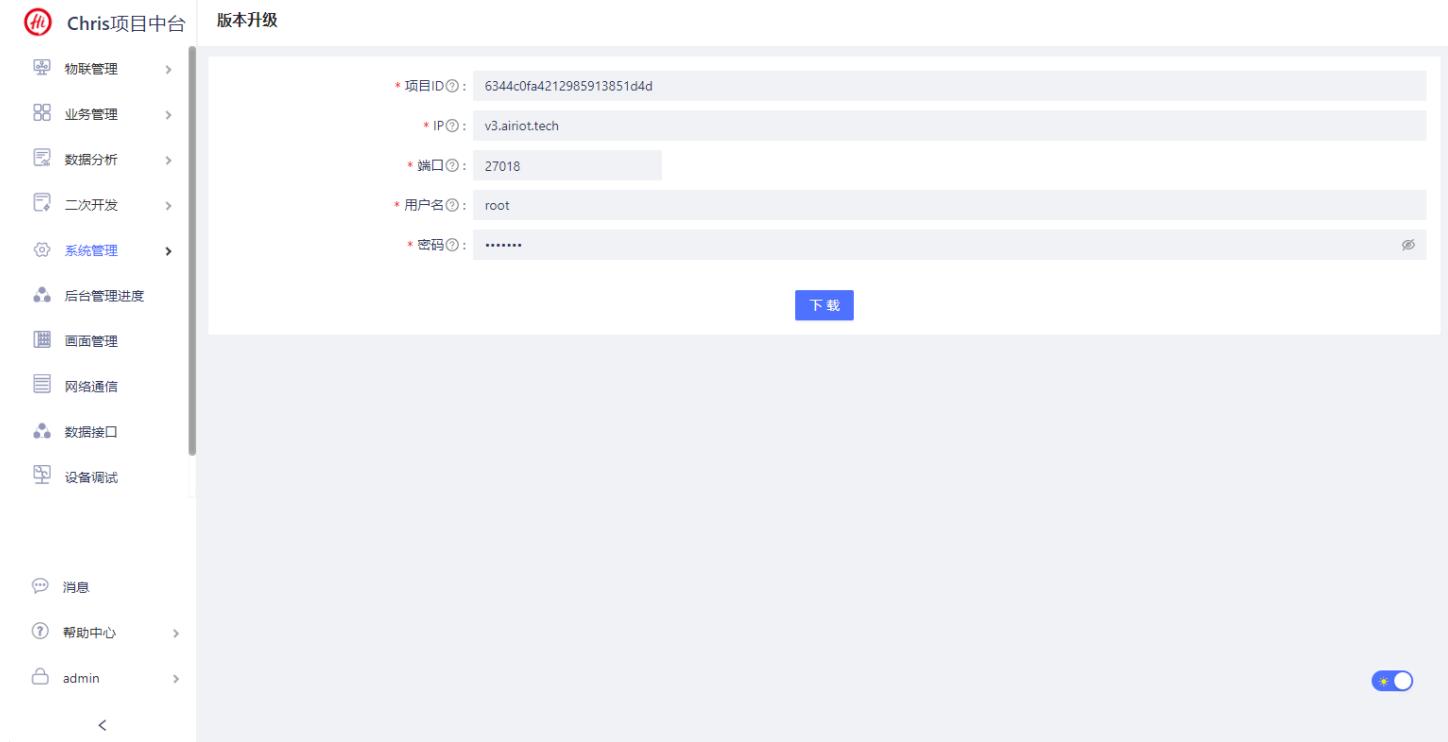
<https://www.airiot.tech/repos/dashboard-upgrade/v3/releases>。

2. 安装dashboard-upgrade、migration服务

进入运维管理平台，在服务管理中，离线上传下载的dashboard-upgrade、migration文件。

步骤四：到3.0平台中下载版本升级文件

1. 路径：系统管理>版本升级。



2. 填写以下信息：

- 项目ID:填写当前项目ID。
- IP:填写当前项目的IP地址或域名（注意:不填写http://）。
- 端口：项目的mongo数据库端口，一般为27017。
- 用户名：填写项目的mongo数据库端口。

- 密码：填写项目的mongo数据库密码。

3. 填写后点击下载，将当前项目版本升级文件打包下载到本地。

步骤五：到4.0平台中上传备份文件并还原

1. 路径:通用功能>备份管理

文件名	备份文件	备份时间	文件大小(M)
AIRIOT智慧系统_2024-02-01 11:28:15	file server (媒体文件)、postgres (基础数据)	2024-02-01 11:28:26	0

2. 将3.0版本升级文件上传到4.0平台中，操作详见[备份管理中上传备份说明](#)；

3. 将上传的备份文件进行还原，还原操作详见[备份管理中还原说明](#)。

4. 还原成功后，即完成将3.0项目升级到4.0项目数据库文件升级。

步骤六：迁移媒体库文件

由于3.0到4.0升级无法统一迁移媒体库文件，需单独处理，具体方式如下（二选其一即可）：

1. 方式一：从3.0平台下载媒体文件，并上传到4.0平台，需保持相同目录结构。

2. 方式二：需进行目录复制，操作如下：

- linux: `airiot/fileServer` 下项目 id 和 `mediaLibrary` 下项目 id 拷贝到 4.0 下 `airiot/fileServer` 对应的新的项目 id 下和 `mediaLibrary` 下新项目 id 下。
- windows: `core/fileServer` 下项目 id 和 `mediaLibrary` 下项目 id 拷贝到 4.0 下 `airiot/fileServer` 对应的新的项目 id 下和 `mediaLibrary` 下新项目 id 下。

历史数据升级步骤

1. 进入 etcd <http://ip:58080/etckeeper/> 输入用户名、密码;
2. 修改 `/airiot/config/pro.json` 配置, 将 Sharding 由 true 改为 false;

```
Nodes /airiot/config/pro.json
16 "MaxLifetime": 3600,
17 "MaxOpenConns": 20,
18 "MaxIdleConns": 10,
19 "TablePrefix": "",
20 "EnableAutoMigrate": true,
21 "Postgres": {
22     "Host": "127.0.0.1",
23     "Port": 5432,
24     "User": "root",
25     "Password": "ENC[REDACTED]!",
26     "DBName": "postgres",
27     "SSLMode": "disable"
28 },
29 },
30 "TSDB": {
31     "DBType": "influx",
32     "DBName": "tsdb",
33     "Timeout": 5,
34     "MaxConn": 50,
35     "Influx": {
36         "Protocol": "HTTP",
37         "Addr": "http://127.0.0.1:8086",
38         "Username": "admin",
39         "Password": "ENC[REDACTED]",
40         "Sharding": true,
41         "CreateDatabaseSql": "CREATE DATABASE \"\$s\""
42     }
43 },
44 "Redis": {
45     "Type": "",
46     "Addrs": [
47         "127.0.0.1:6379"
48     ],
49     "Password": "ENC[REDACTED]",
50     "PoolSize": 10
51 },
52 "MQ": {
```

3. 重启 `core` 和 `storage` 服务。

技术支持

如果您在升级过程中遇到任何问题, 或需要任何技术支持, 请随时联系我们的客户服务团队。我们将竭诚为您提供帮助。