

Flomesh 软负载均衡管理员维护手册

1. 引言

1.1. 背景

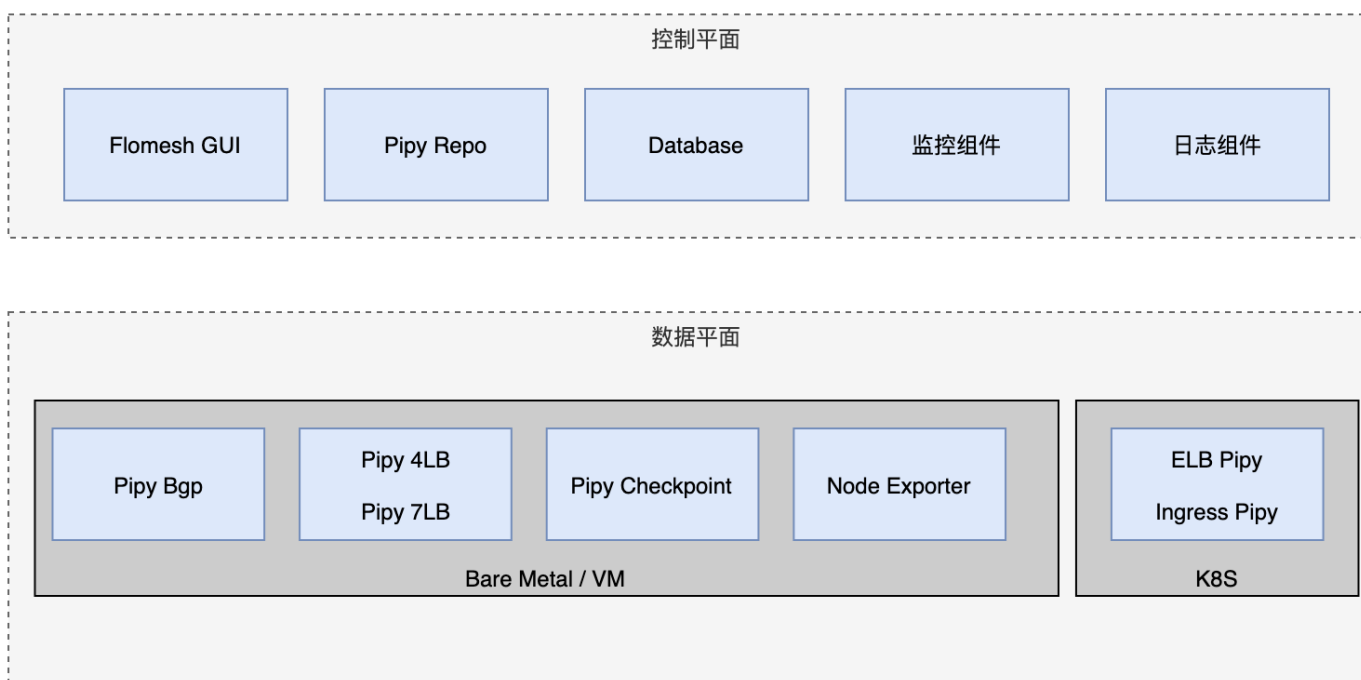
Flomesh 负载均衡软件 (Flomesh Loadbalancer , 以下简称 FLB) 是一站式的应用流量管理平台, 支持裸金属负载均衡, Kubernetes Loadbalancer , 以及 Kubernetes Ingress。本文主要介绍 Flomesh 软负载均衡管理员维护手册, 包括系统初始化、系统权限管理、系统组件维护等内容。

1.2. 定义

Flomesh 图形用户界面简称: Flomesh GUI。

2. 系统概述

2.1. 系统组件架构



FLB 包含控制平面和数据平面。

- 控制平面

- 单独部署，支持容器和 K8S 部署
- 用于集中管理数据平面各 LB 节点，提供 Flomesh 图形用户界面：Flomesh GUI
- 控制平面包含如下组件
 - Flomesh GUI：提供图形化管理界面，包括负载均衡、地址池管理、健康检查、运维中心等功能模块，同时提供了基础的系统管理功能
 - Pipy Repo：后端组件，实现所有 LB 集群的配置集中管理功能
 - Database：后端组件，提供平台数据持久化存储，支持 MySQL/MariaDB, PostgreSQL 等数据库
 - 监控组件：后端组件，提供监控指标收集展示能力，支持标准的 Prometheus[[Data model](#) | [Prometheus](#)]接口
 - 日志组件：后端组件，提供日志收集展示能力，支持 Clickhouse、ELK 等组件
- 数据平面
 - 部署在 LB 节点，支持裸金属、虚拟机、容器和 K8S 集群内部署
 - 用于实现业务流量负载均衡和代理转发
 - LB 节点包括如下进程
 - Pipy Bgp 进程，用来声明 LB 节点 VIP
 - Pipy proxy 进程，用来实现负载均衡，根据需求可分为 4LB 进程和 7LB 进程
 - Pipy checkpoint 进程，用来实现 7 层健康检查拨测
 - Node Exporter 进程，用来收集 LB 节点系统监控信息

3. 管理员 Flomesh GUI 操作指导

3.1. 系统初始化配置

3.1.1. 初始化用户创建

3.1.1.1. 创建初始化用户

部署完成后，第一步需要创建初始化用户。登录 `http://[HOST]:[PORT]/flomesh-gui/`，输入管理员的账户名称、邮箱与密码，完成注册。

3.1.1.2. 初始化用户登录

登录地址：`http://[HOST]:[PORT]/flomesh-gui/#/login`



输入用户名和登录密码进行登录。

3.1.2.初始化组件配置

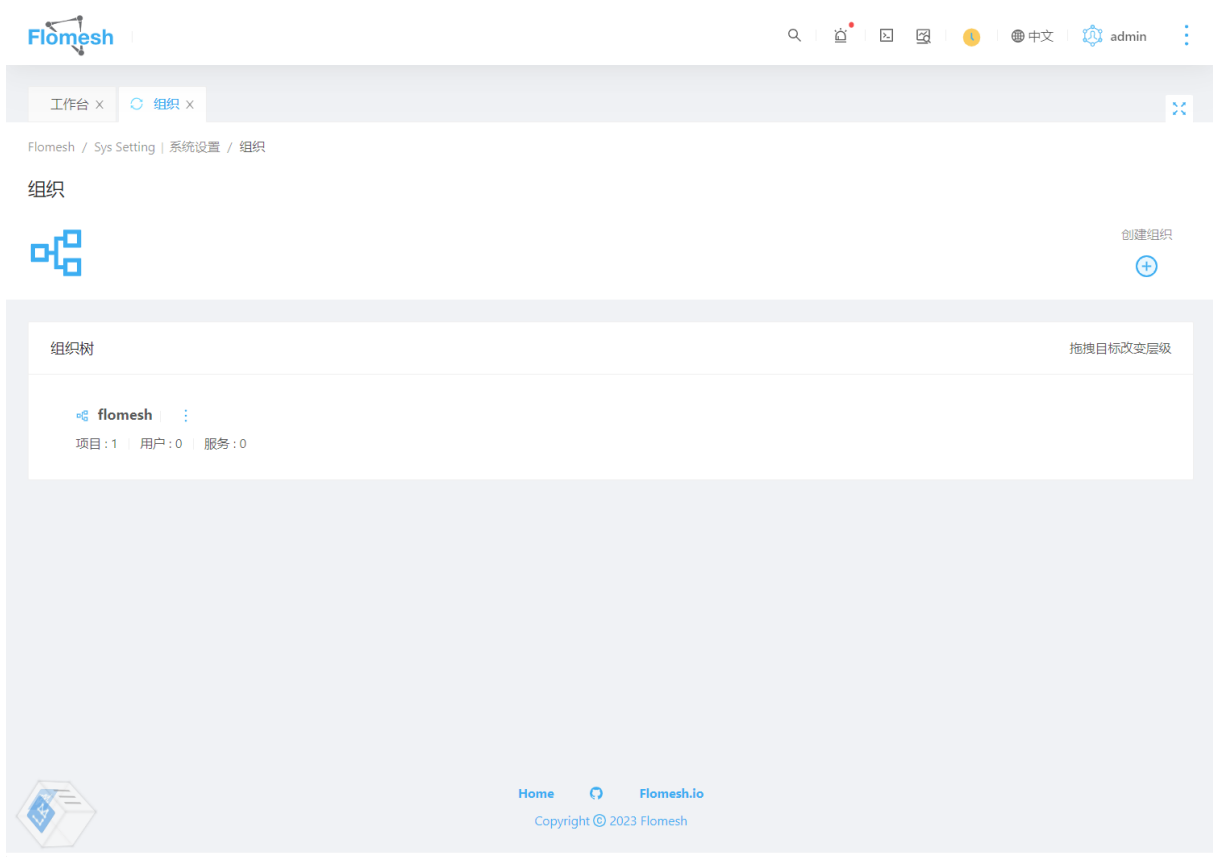
部署完成后，平台需要配置依赖的监控、日志等组件，使用初始化用户登录 [http://\[HOST\]:\[PORT\]/flomesh-gui/#/login](http://[HOST]:[PORT]/flomesh-gui/#/login)，在”组件管理“一栏下，添加监控、日志组件。详见 3.3.1.3 日志组件、3.3.1.4 监控组件。

3.2. 平台权限管理

3.2.1.组织管理

组织是一个管理单元，可以是公司、部门、团队。组织具有上下级关系，形成树状视图。FLB 组织管理模块支持组织成员的增减、组织内成员权限分配，组织的关系维护等。

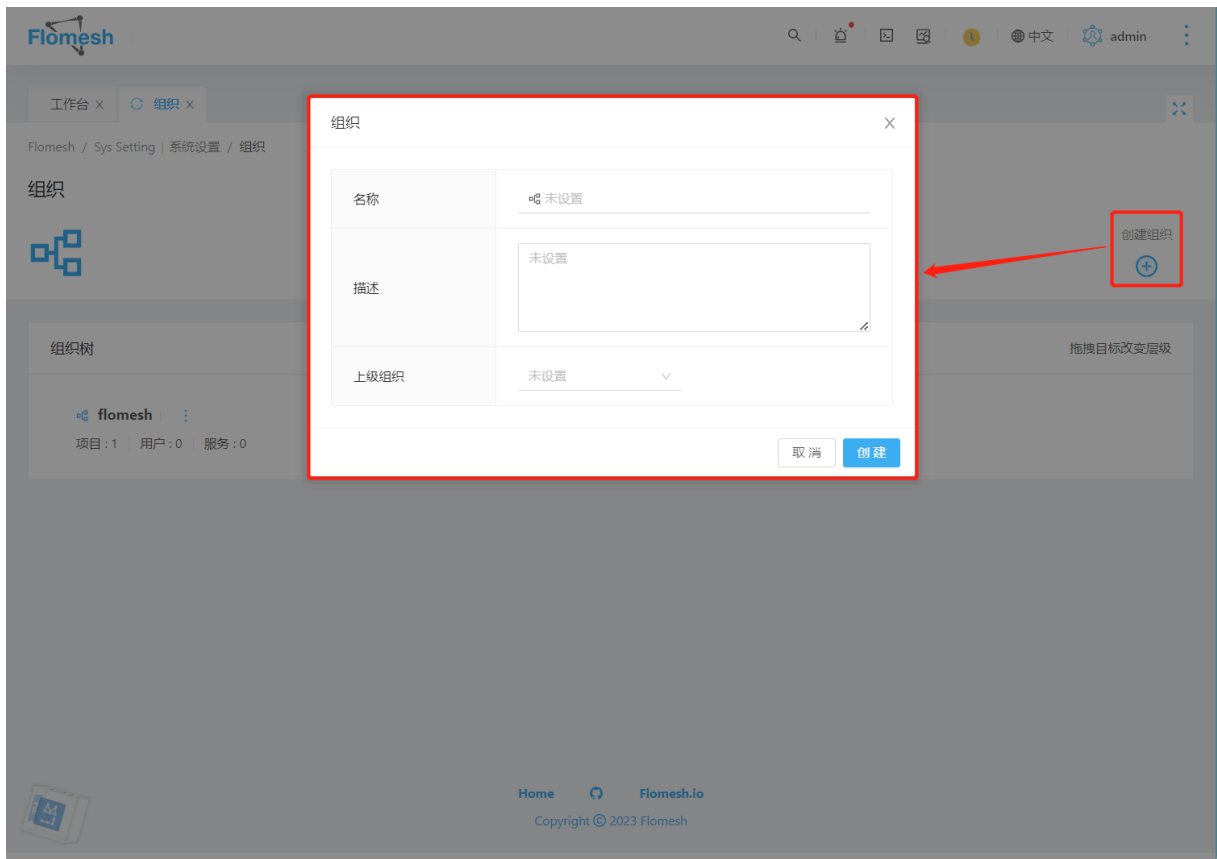
左下角菜单单击“系统设置” > “组织”，进入组织管理页面，该页面展示组织的树状关系视图。通过此页面可以进入组织的创建、编辑、组织详情页面。



3.2.1.1. 创建新的组织

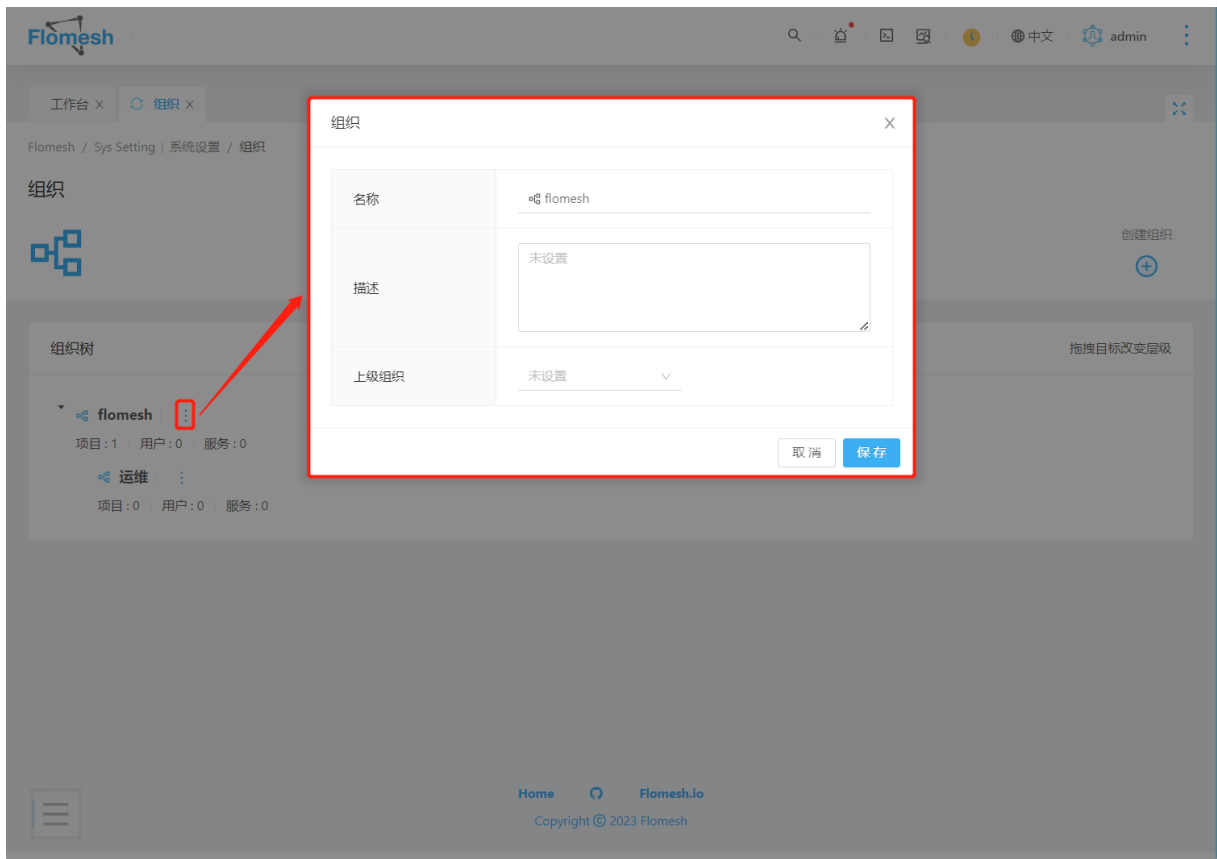
组织管理页面右上角单击“+”，进入创建组织页面，配置组织信息，组织信息参数如下表所示。

组织基本信息	
参数名称	说明
名称	必填项，自定义组织名称。
描述	选填项，对组织的描述信息。
上级组织	必填项，该组织所属的上级组织。



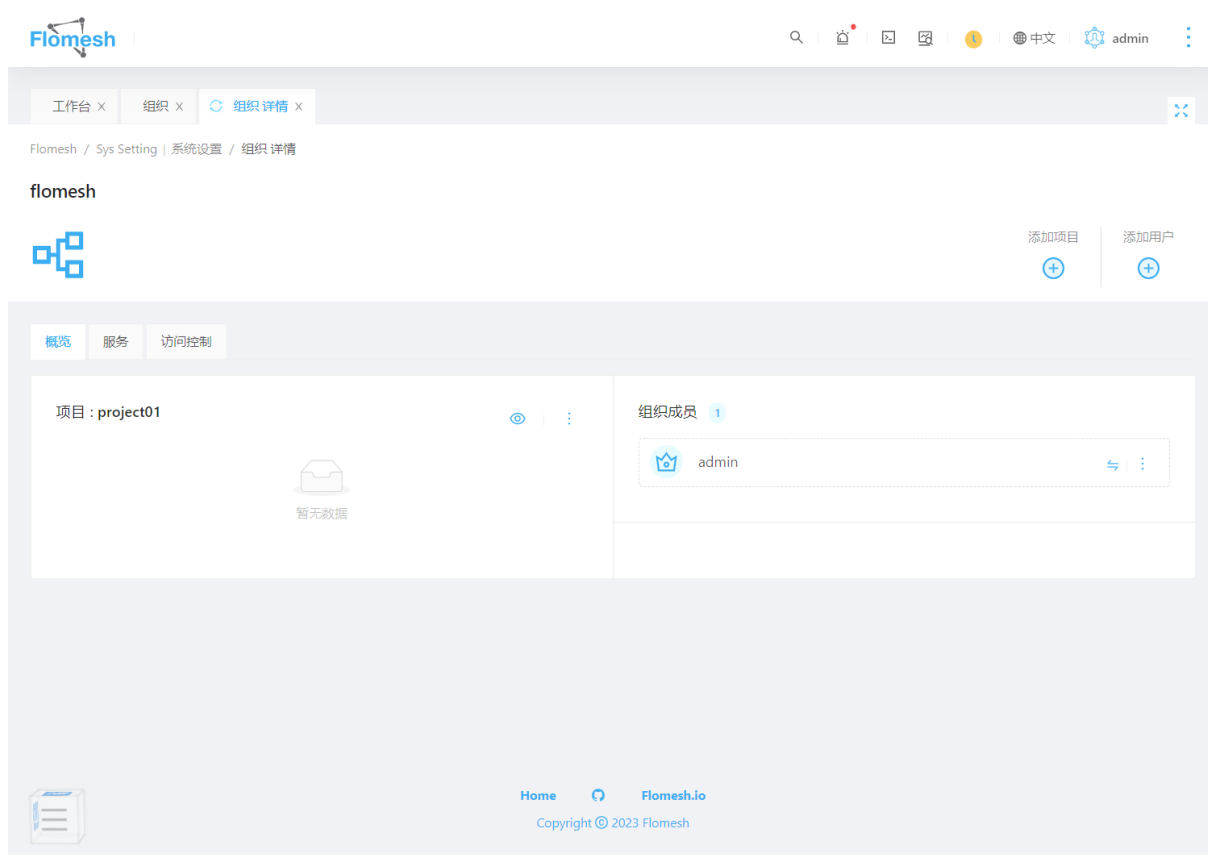
3.2.1.2. 修改组织信息

在组织管理页面鼠标移到需要编辑的组织右侧下拉菜单，弹出菜单选择“编辑”，进入编辑组织页面，可以对组织名称、描述和上级组织进行修改。



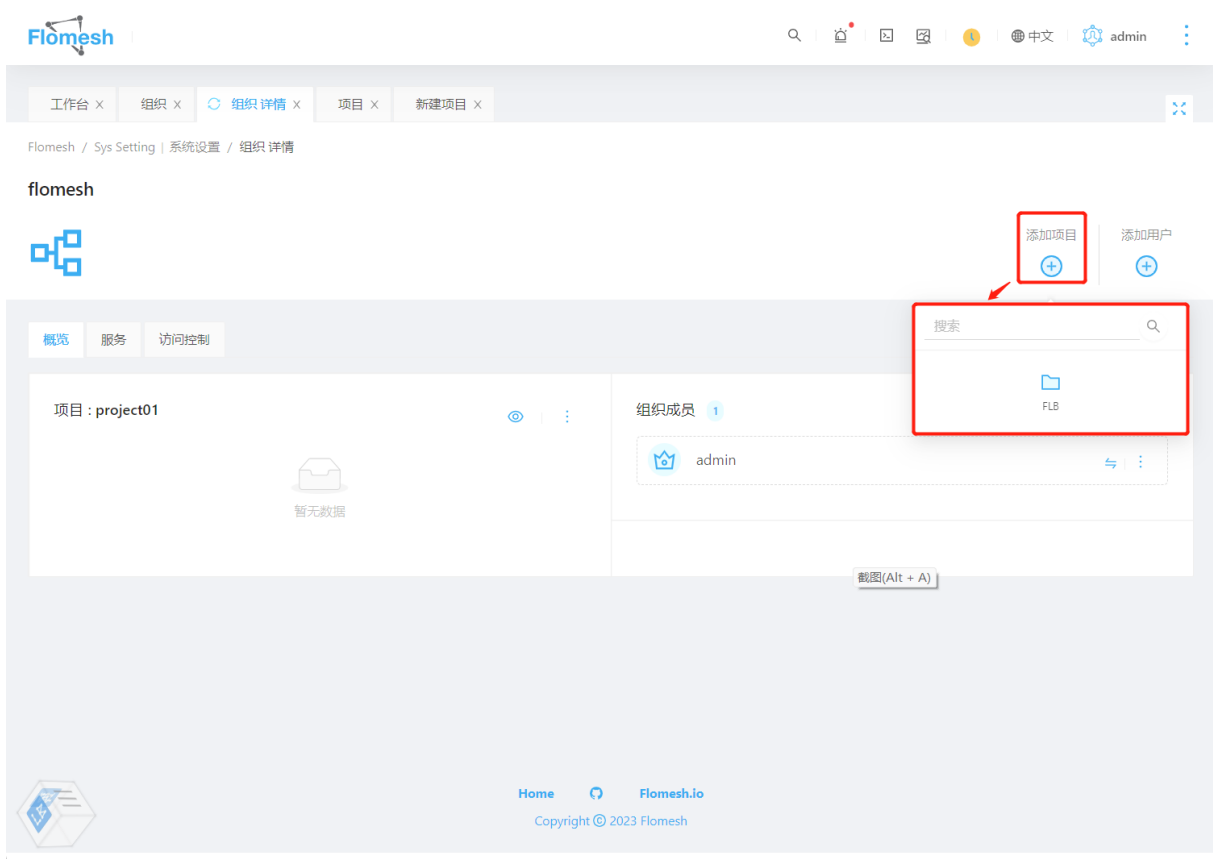
3.2.1.3. 组织详情

在组织管理页面鼠标移到需要管理的组织右侧下拉菜单，弹出菜单选择“管理”，进入组织详情页面，支持组织内的项目和用户的增减、用户组织权限管理、组织的服服务管理、访问控制白名单管理。



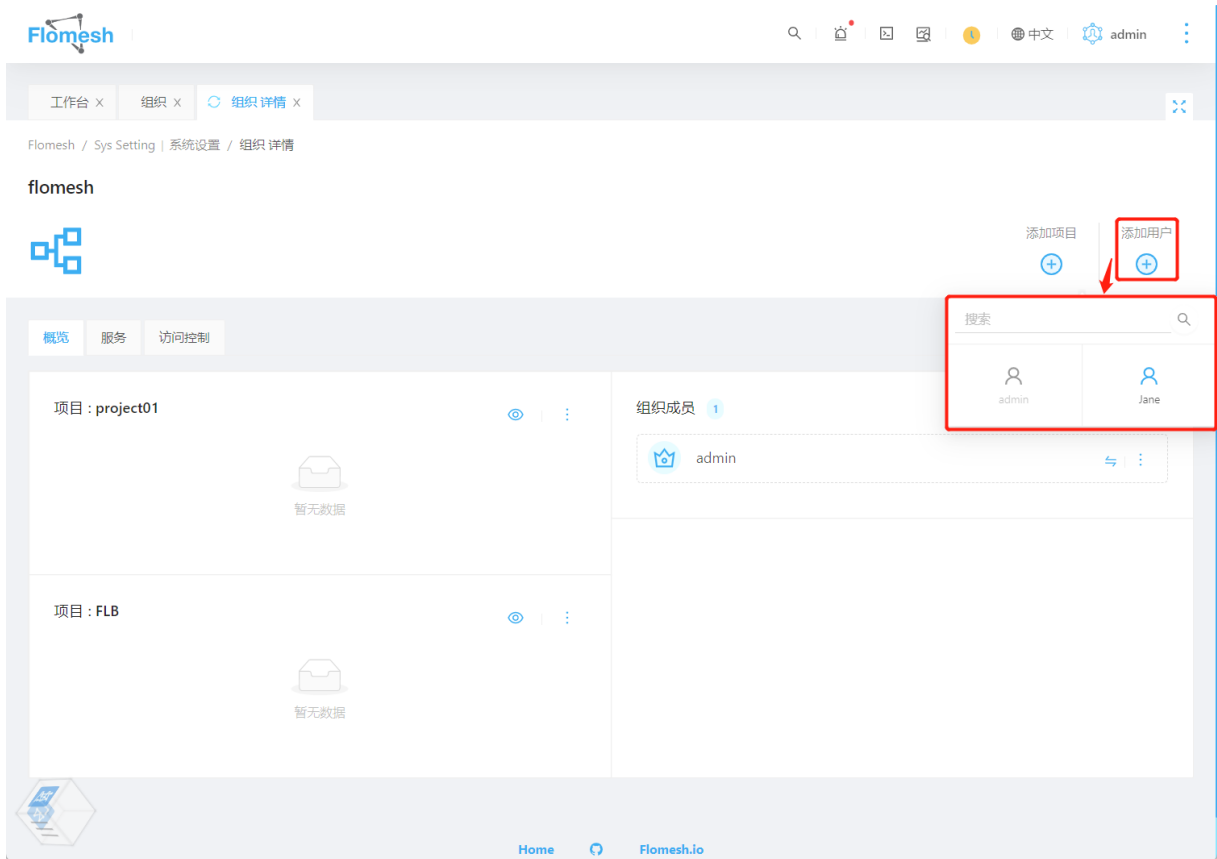
3.2.1.4. 为组织添加所管理的项目

组织详情页面右上角单击添加项目“+”，弹出项目列表，鼠标单击项目即被添加到组织中，项目列表支持按项目名称进行搜索。



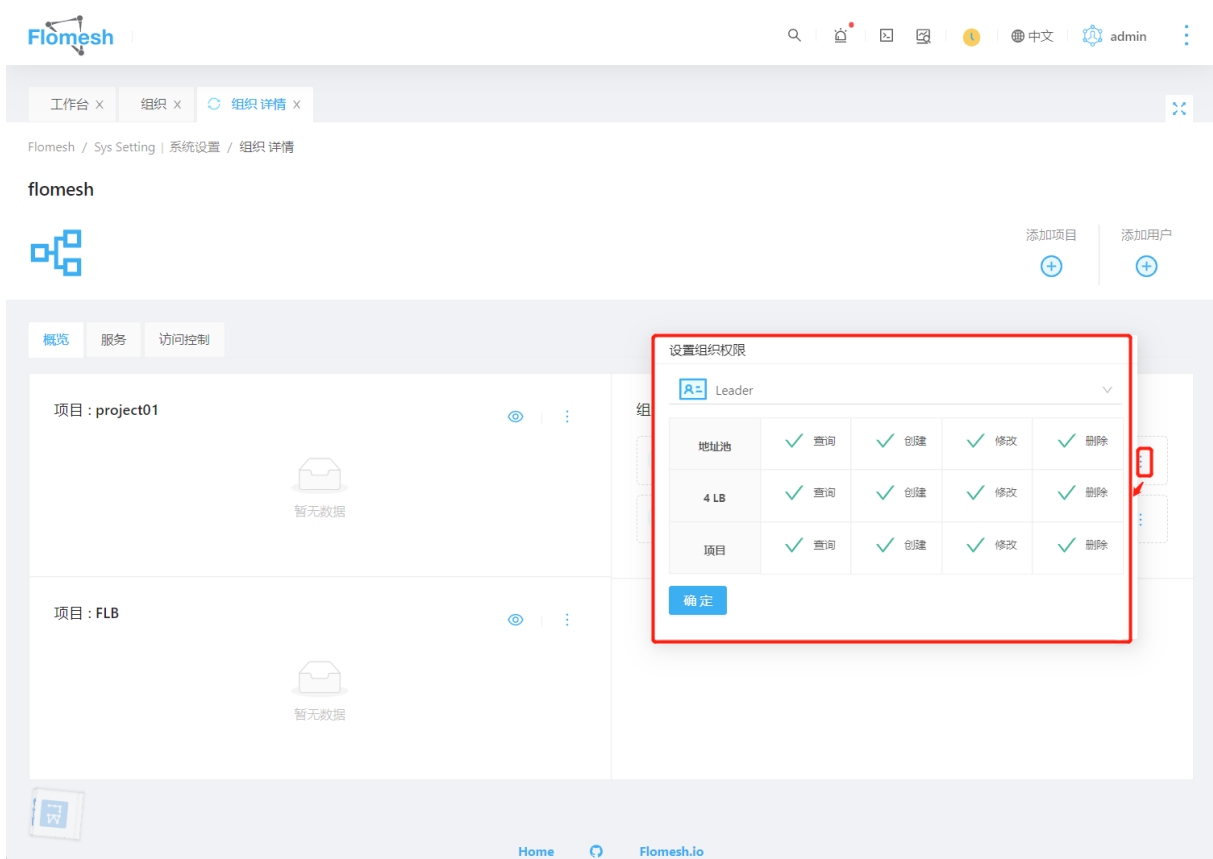
3.2.1.5. 为组织添加用户

组织详情页面右上角单击添加用户“+”，弹出用户列表，鼠标单击用户即被添加到组织中，用户列表支持按用户名进行搜索。



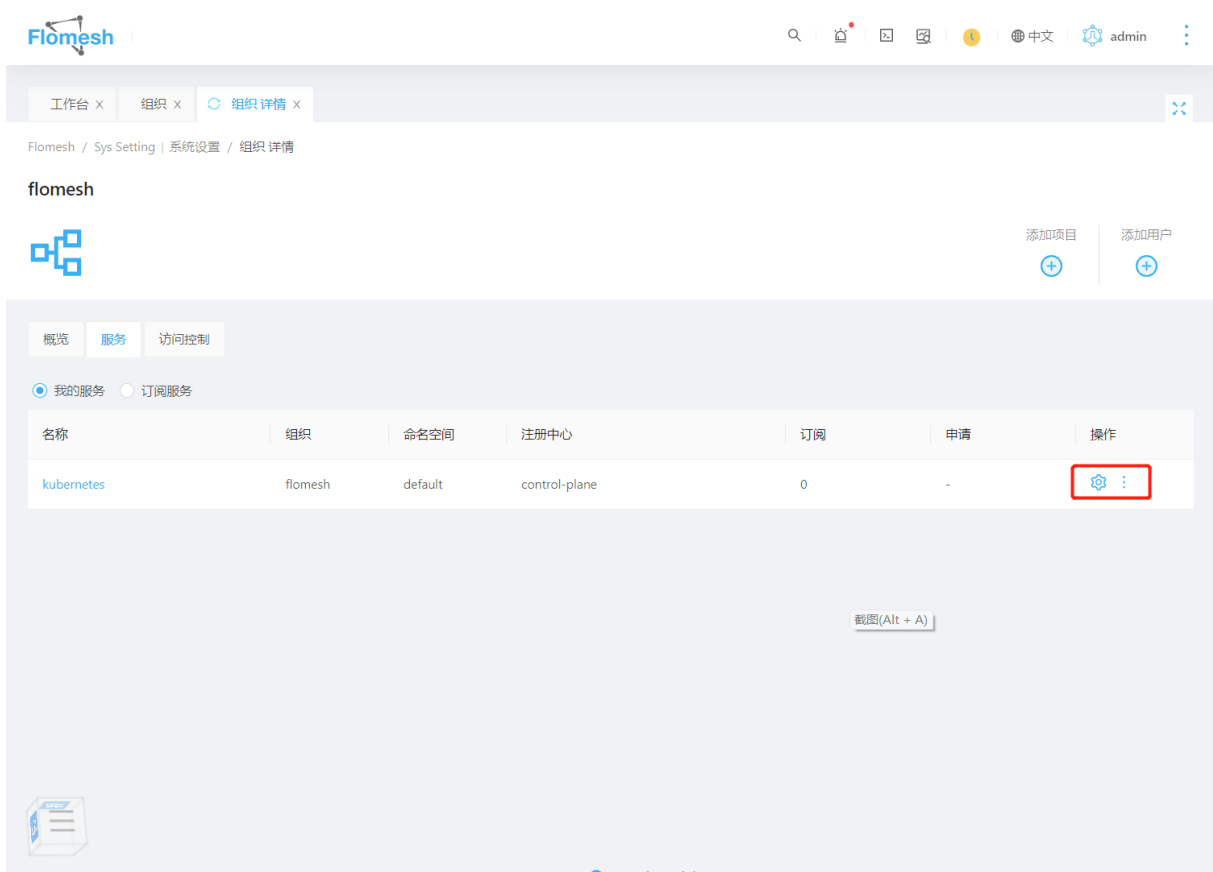
3.2.1.6. 设置成员的组织权限

组织详情页面>概览页卡组织成员列表，鼠标移到需要编辑的成员右侧下拉菜单，弹出菜单选择“设置组织权限”，弹出组织角色选择下拉列表，选择成员在组织中对应的角色权限。



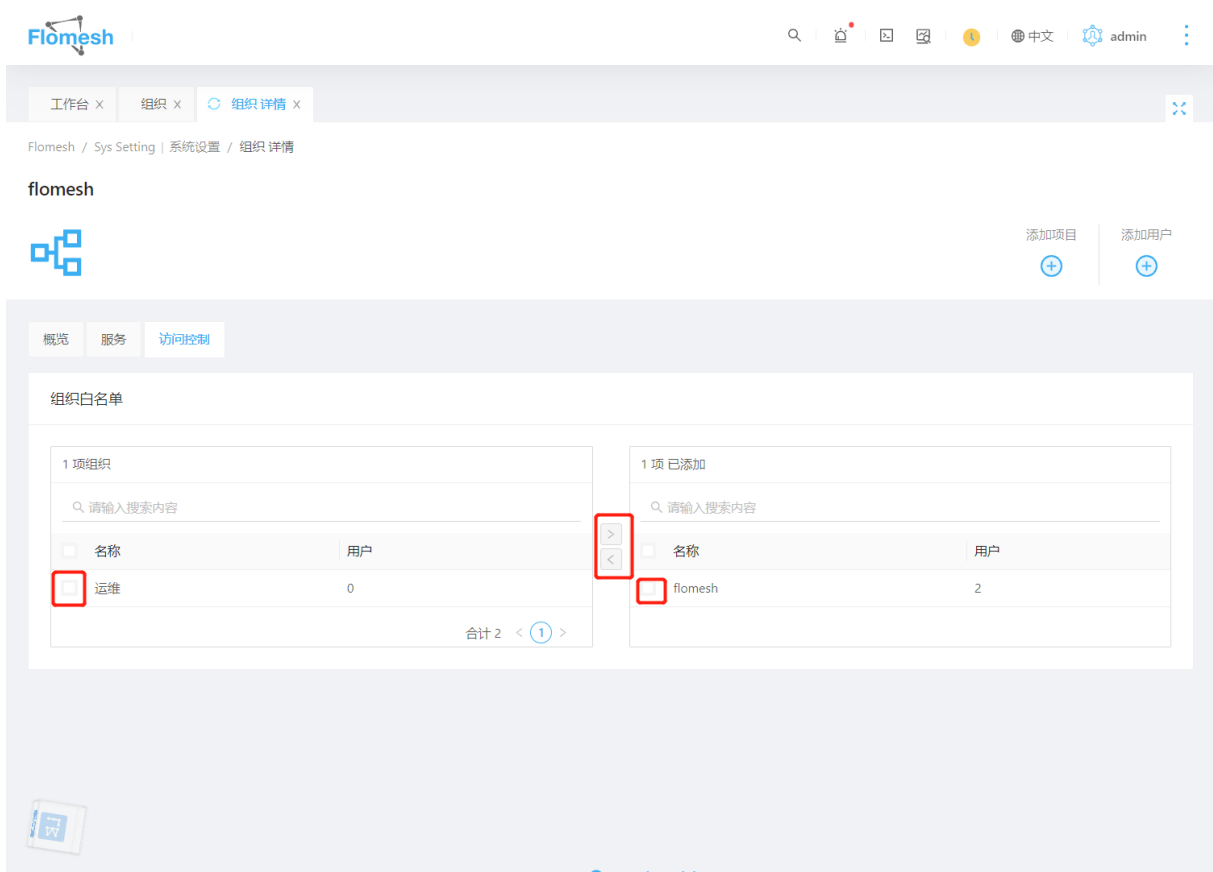
3.2.1.7. 组织的 service 管理

组织详情页面>服务页卡服务列表，鼠标移到需要编辑的服务右侧操作菜单，可以对服务进行编辑、配置和删除操作。



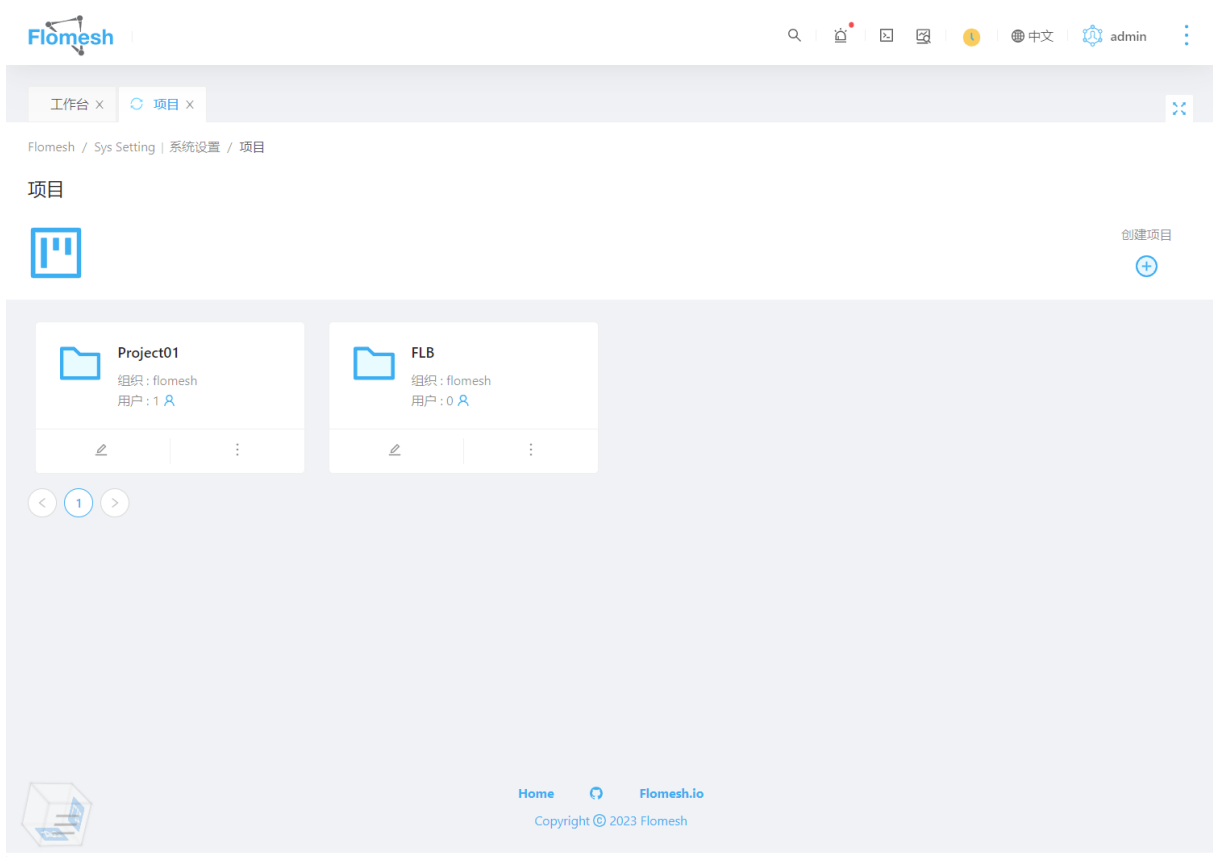
3.2.1.8. 组织的访问控制管理

组织详情页面>访问控制页卡白名单列表，可以将组织加入或移出白名单。



3.2.2.项目管理

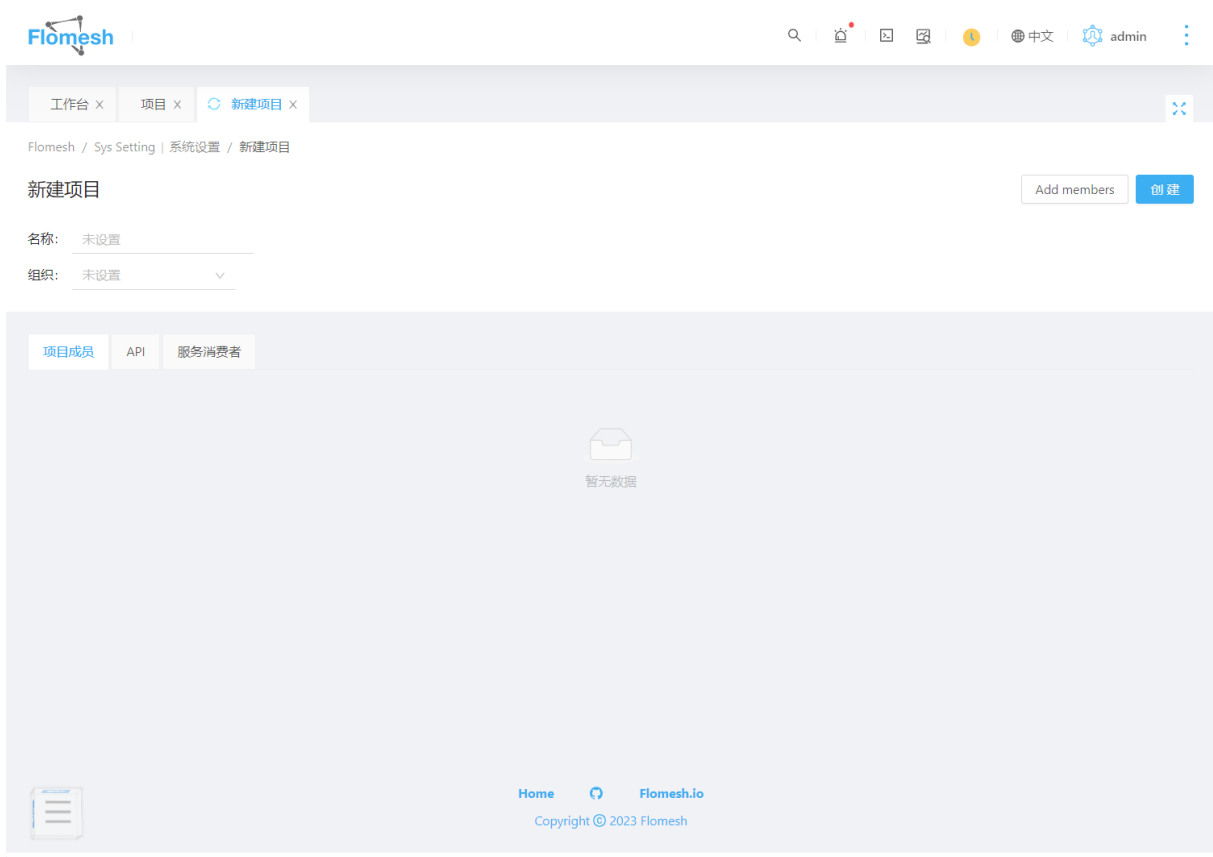
左下角菜单单击“系统设置”>“项目”，进入项目管理页面，该页面展示项目列表。通过此页面可以进入项目的创建、编辑、项目详情页面。



3.2.2.1. 创建新的项目

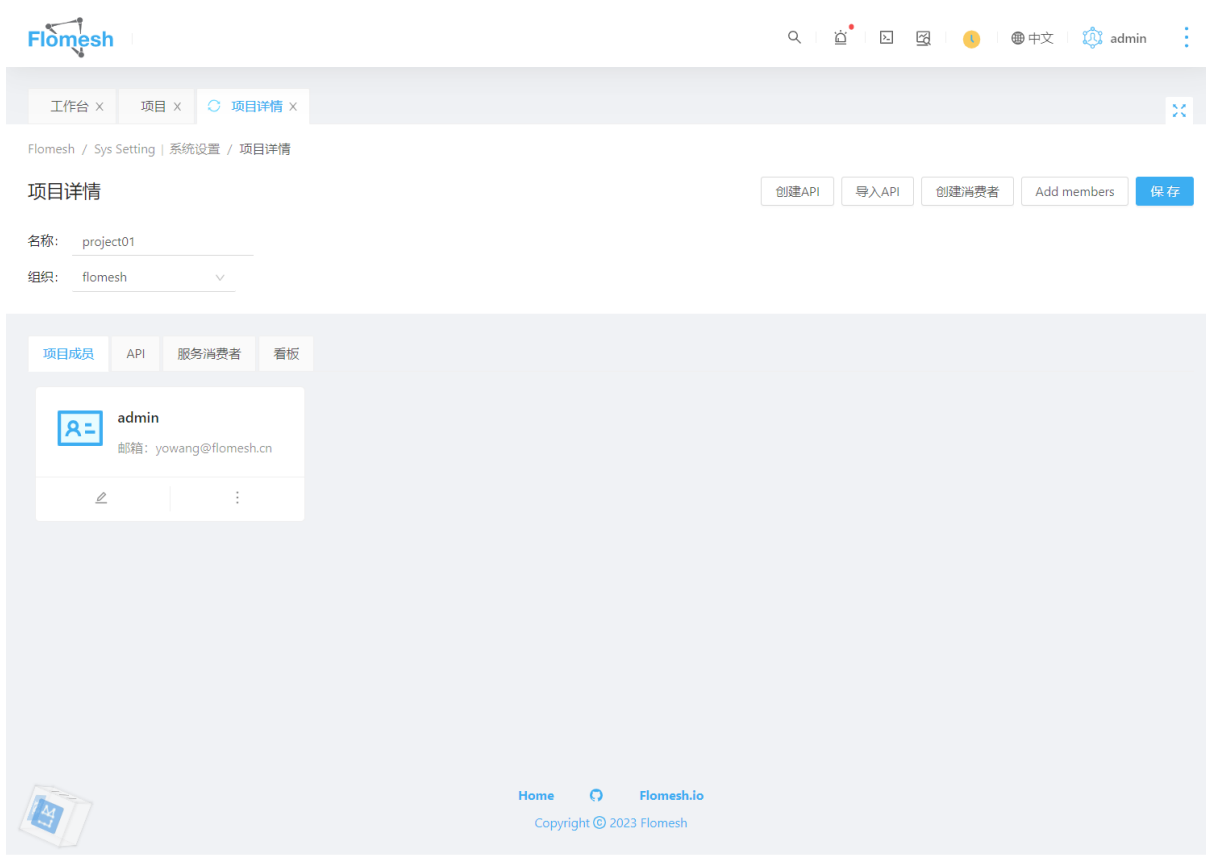
项目管理页面右上角单击“+”，进入新建项目页面，配置项目信息，项目信息参数如下表所示。

项目基本信息	
参数名称	说明
名称	必填项，自定义项目名称。
组织	必填项，下拉选择菜单，选择项目所属的组织。



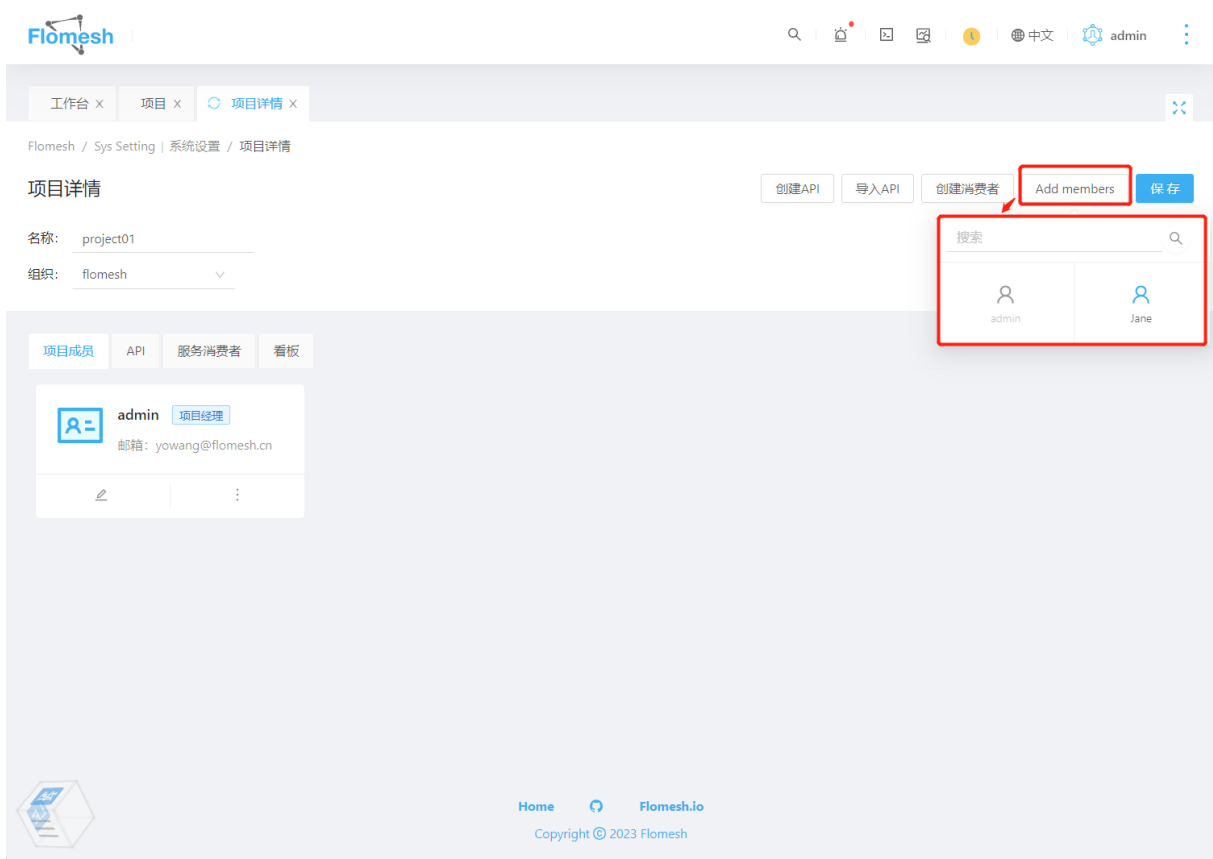
3.2.2.2. 项目详情

在项目管理页面，项目菜单鼠标单击“编辑”，进入项目详情页面，支持项目成员的增减、用户项目权限管理、项目 API 管理、服务消费者管理、看板管理。



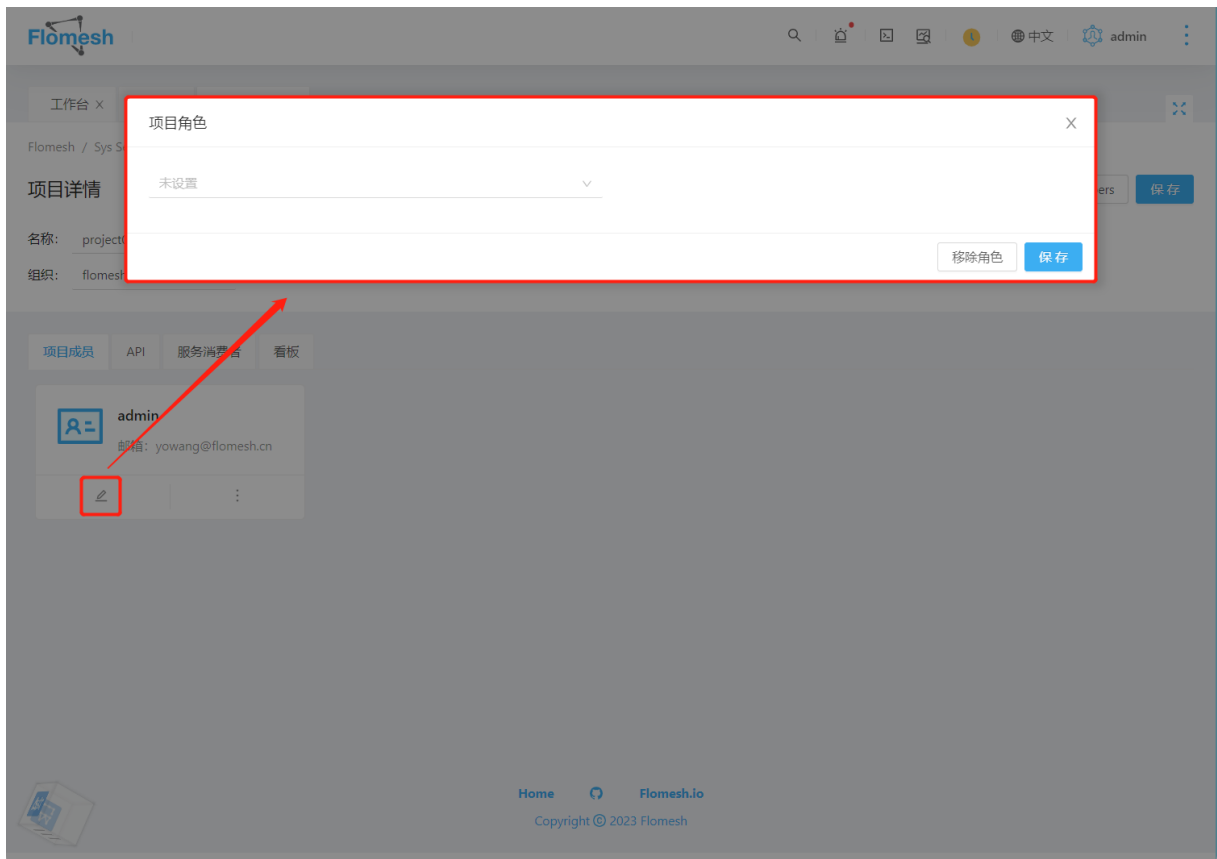
3.2.2.3. 为项目添加用户

项目详情页面右上角单击“添加用户”，弹出用户列表，鼠标单击用户即被添加到项目中，用户列表支持按用户名进行搜索。



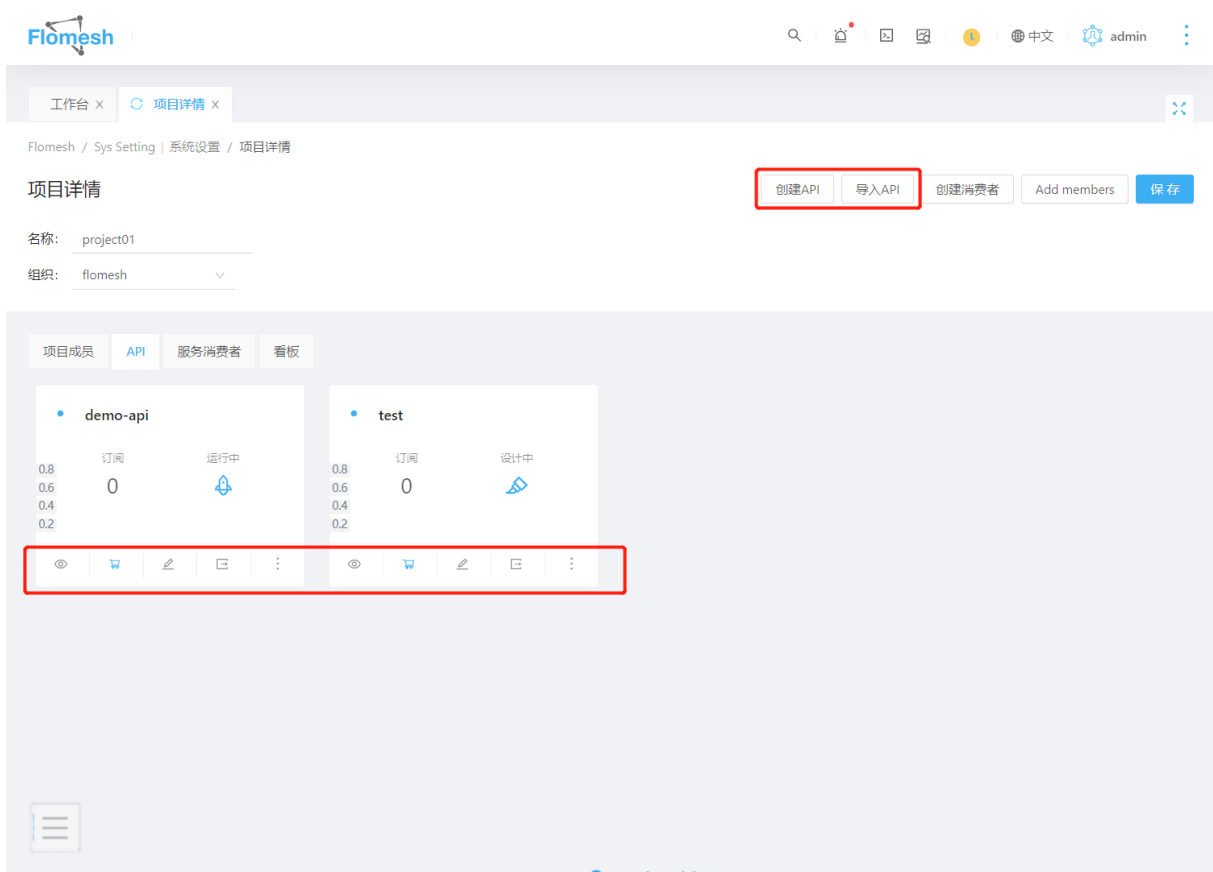
3.2.2.4. 设置成员的项目权限

项目详情页面项目成员页卡，鼠标移到需要编辑的成员菜单，单击“编辑”，弹出项目角色选择下拉列表，选择成员在项目中对应的角色权限。



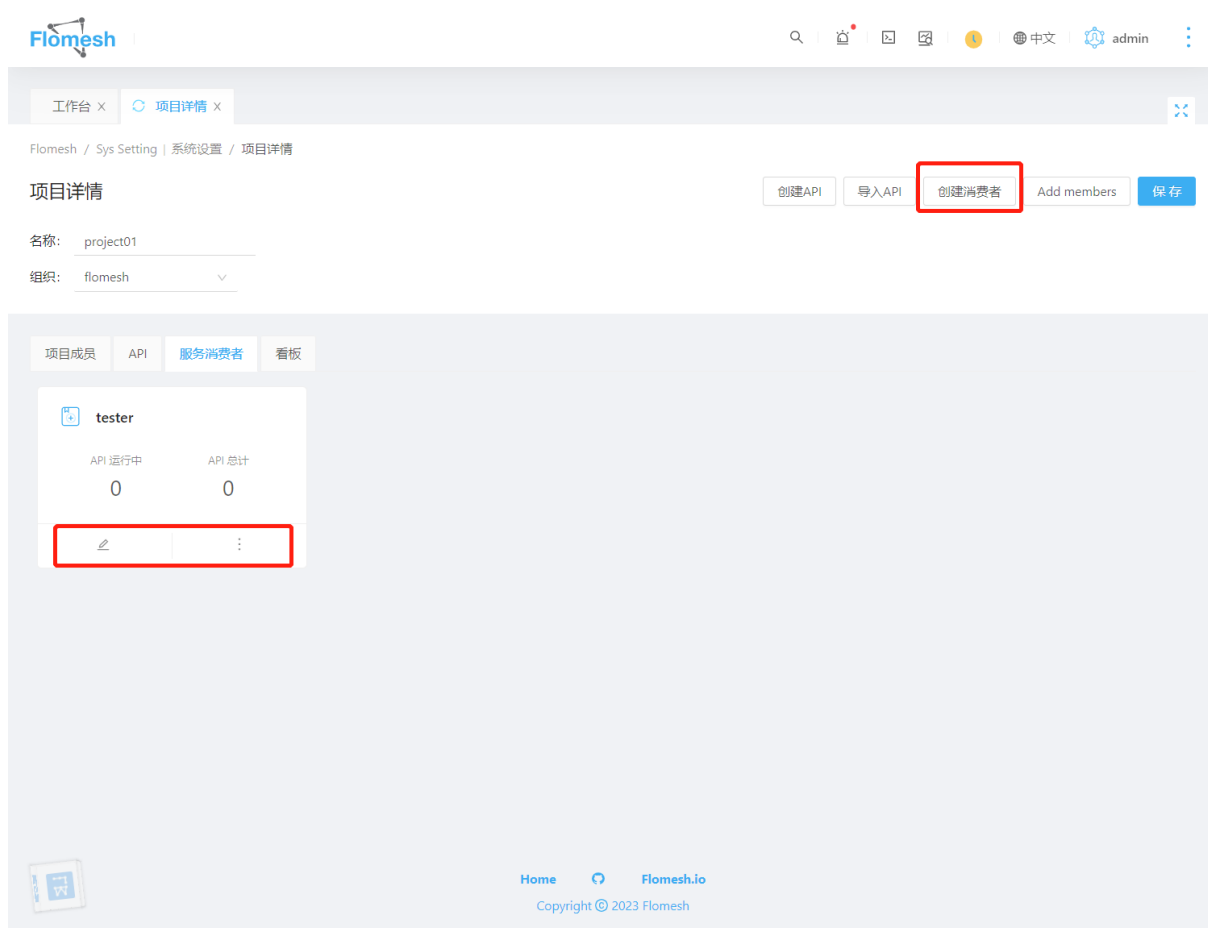
3.2.2.5. 项目 API 管理

项目详情页面, 通过右上角“创建 API”和“导入 API”, API 页卡 API 菜单“预览”、“加入订阅车”、“编辑”、“导出”、“删除”进行 API 管理操作。



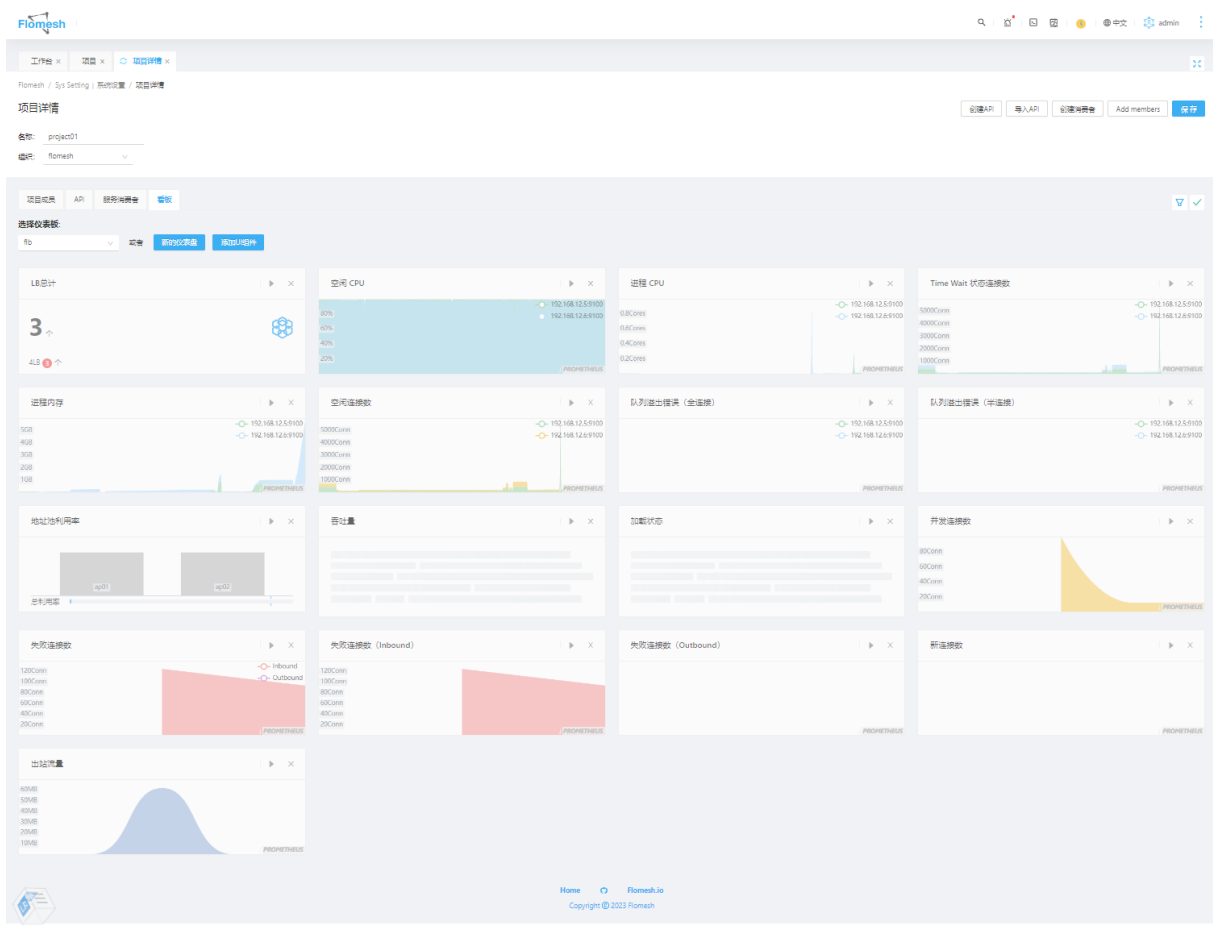
3.2.2.6. 项目服务消费者管理

项目详情页面，通过右上角“创建消费者”，服务消费者页卡消费者菜单“编辑”和“删除”进行服务消费者管理操作。



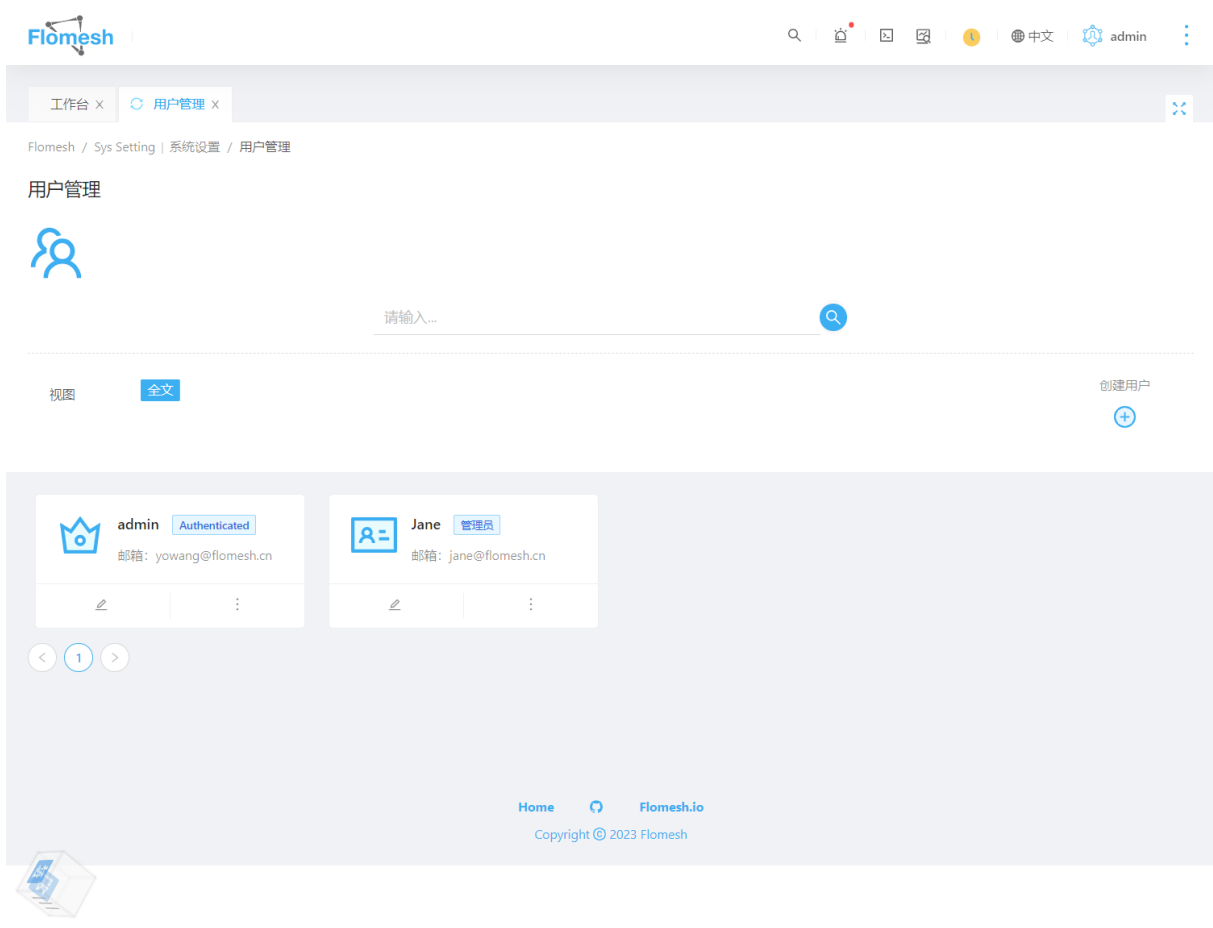
3.2.2.7. 项目看板管理

项目详情页面，单击“看板”页，支持看板的创建、选择、编辑。



3.2.3.用户管理

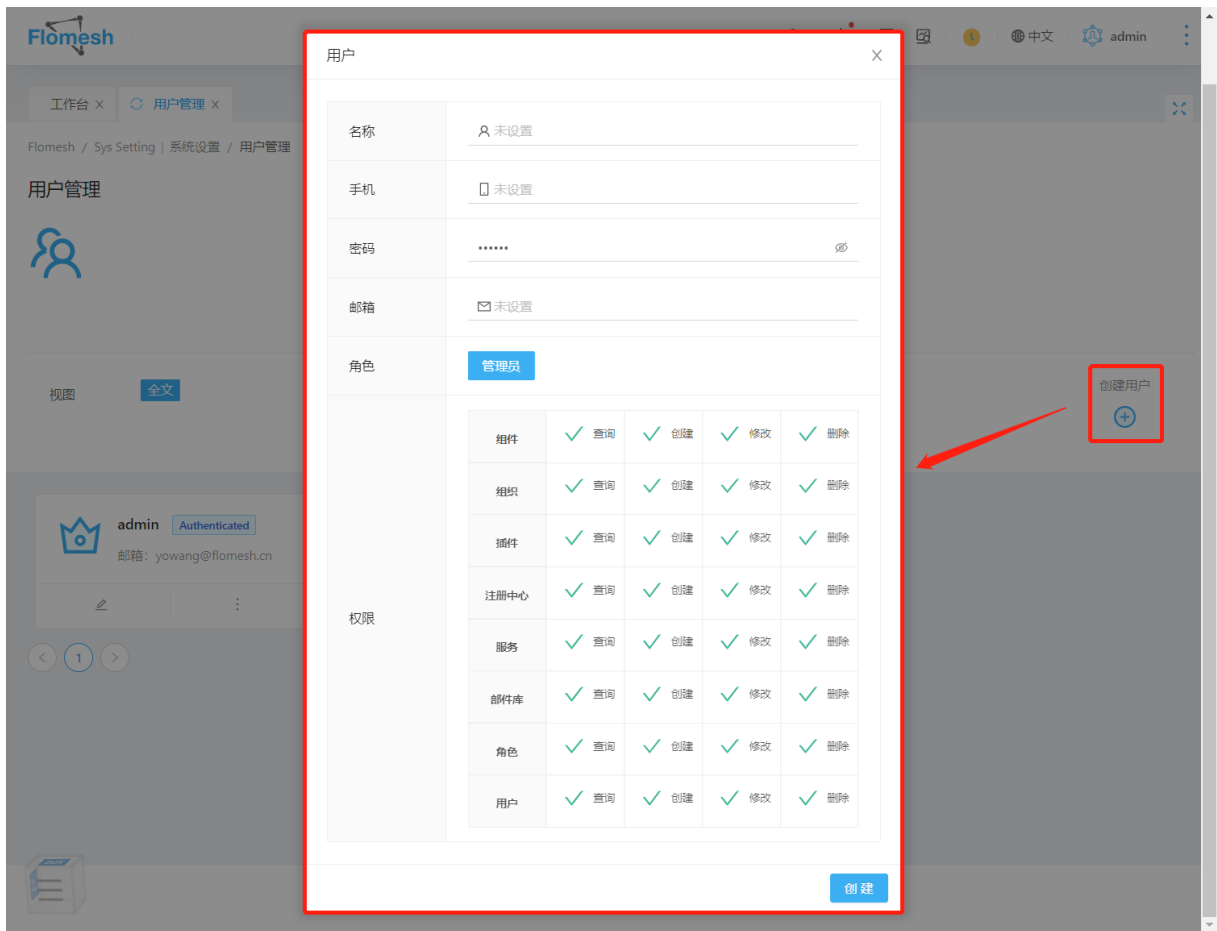
左下角菜单单击“系统设置” > “用户管理”，进入用户管理页面，该页面展示用户列表。通过此页面可以进入用户的创建、编辑页面；支持按用户名搜索。



3.2.3.1. 创建新用户

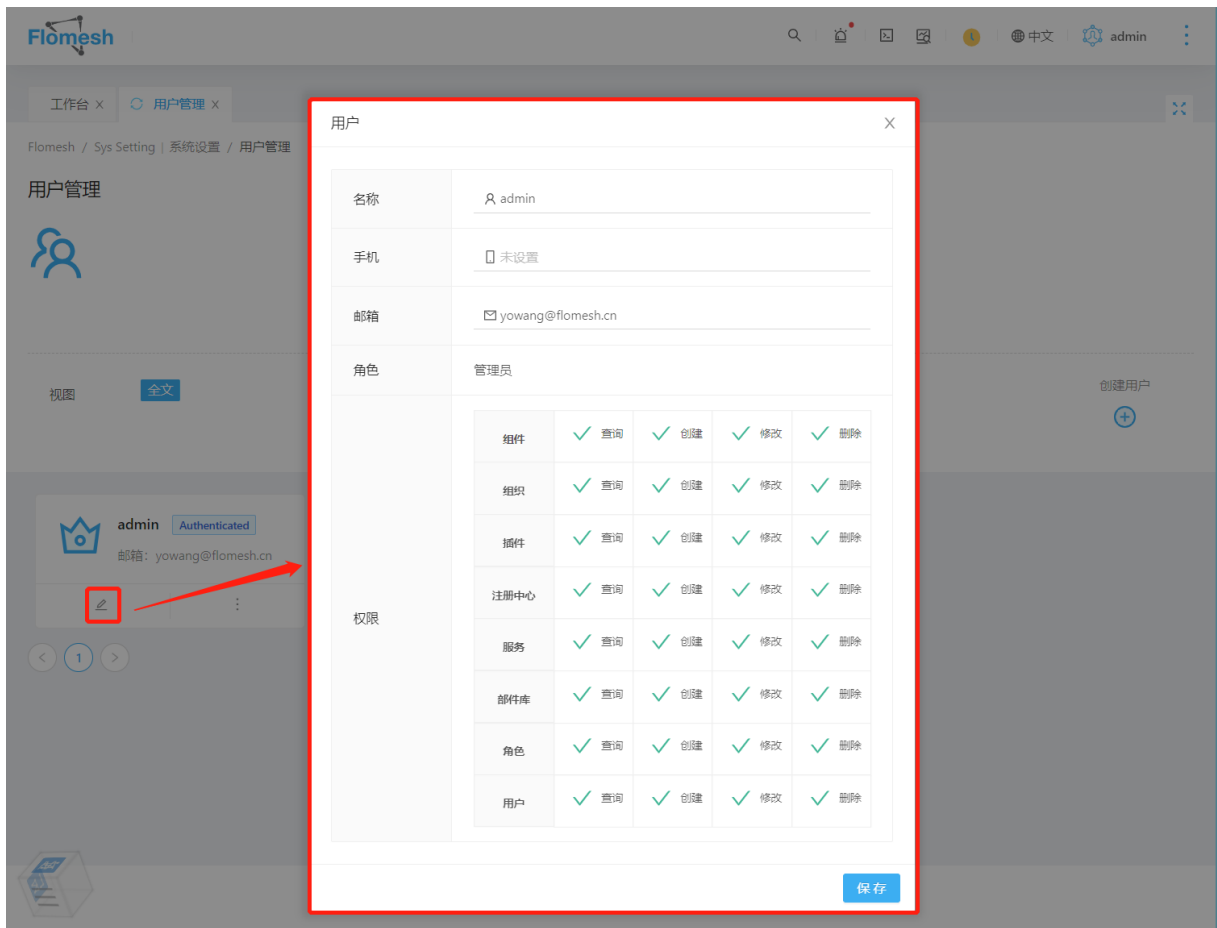
用户管理页面右上角单击“+”，进入新建用户页面，配置用户信息，用户信息参数如下表所示。

用户基本信息	
参数名称	说明
名称	必填项，自定义用户名称。
手机	选填项，该用户手机号。
密码	必填项，该用户登录密码，至少 8 位，同时包含数字，大、小写字母，特殊符号。
邮箱	必填项，该用户邮箱，用于忘记密码时发送验证邮件。
角色	必填项，选择对应权限的系统角色。



3.2.3.2. 修改用户资料

在用户管理页面鼠标移到需要编辑的用户菜单，鼠标单击“编辑”，进入编辑用户页面，可以对用户名称、手机、邮箱、角色权限进行修改。

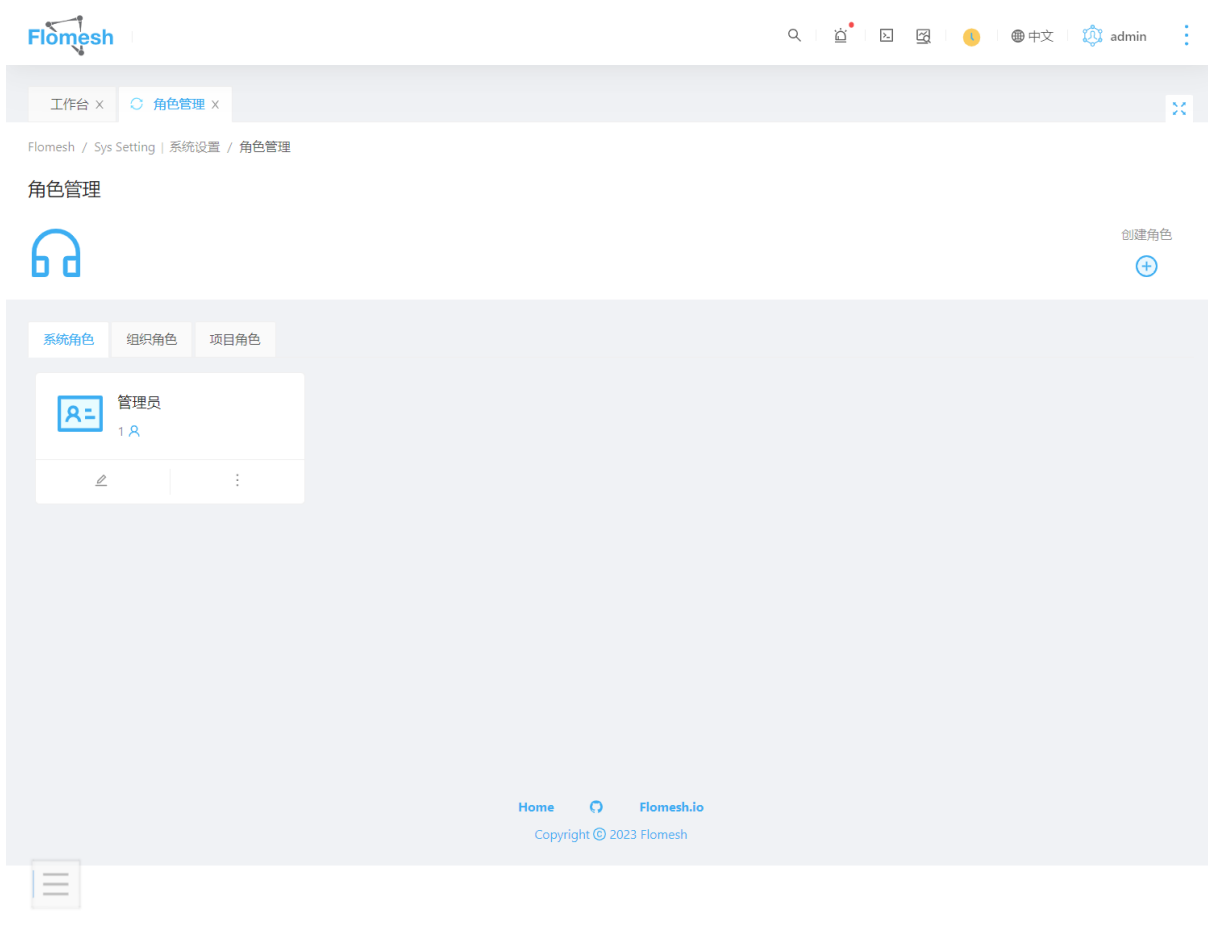


3.2.4.角色管理

左下角菜单单击“系统设置” > “角色管理”，进入角色管理页面，该页面展示角色列表。通过此页面可以进入角色的创建、编辑页面。

目前默认可以创建三种角色：系统角色、组织角色、项目角色。不同角色作用于不同的对象，具体对应关系如下表所示。

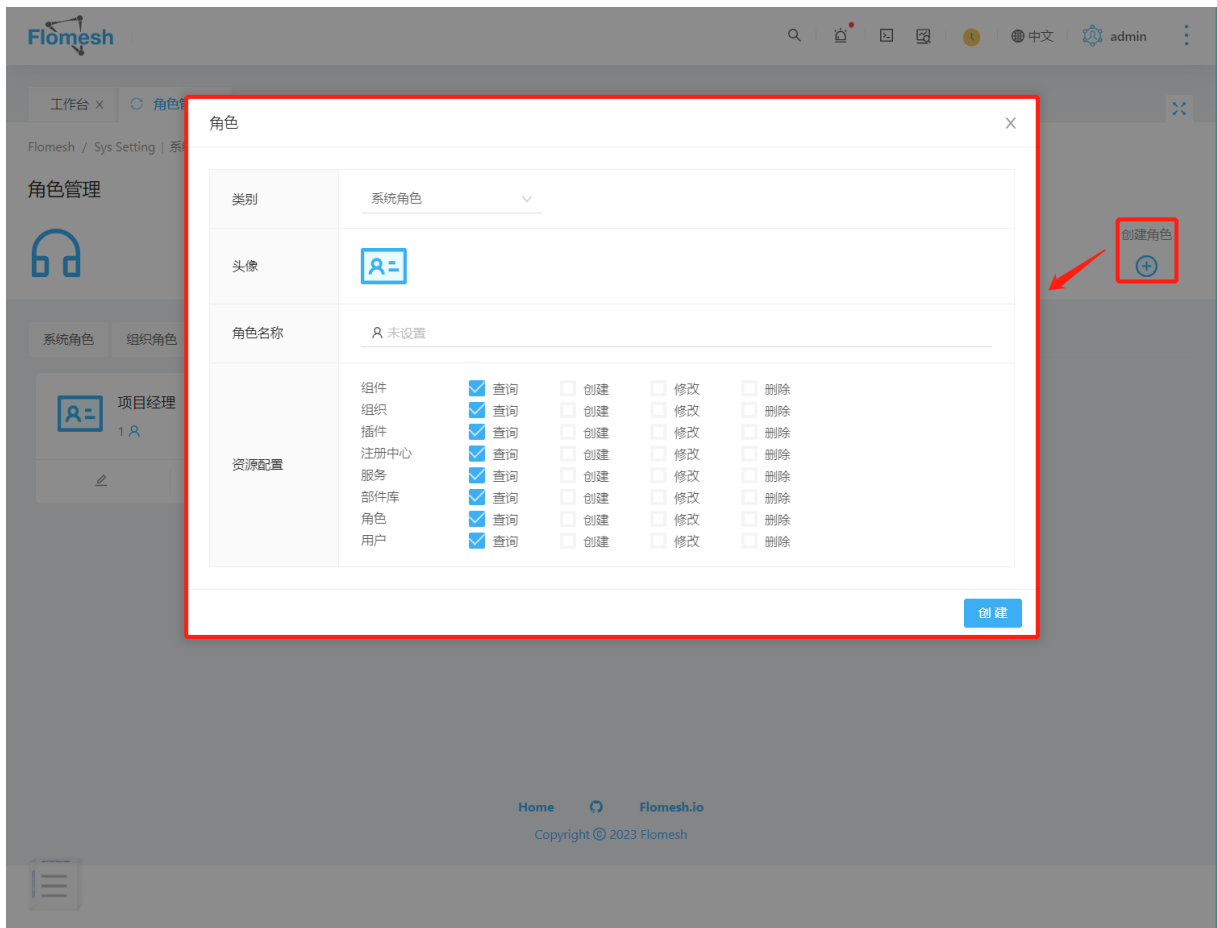
角色资源配置对应关系	
角色	资源配置
系统角色	组件、插件、注册中心、服务、部件库、组织、角色、用户
组织角色	地址池、4LB、项目
项目角色	开放接口、服务消费者、策略、服务提供者



3.2.4.1. 创建角色

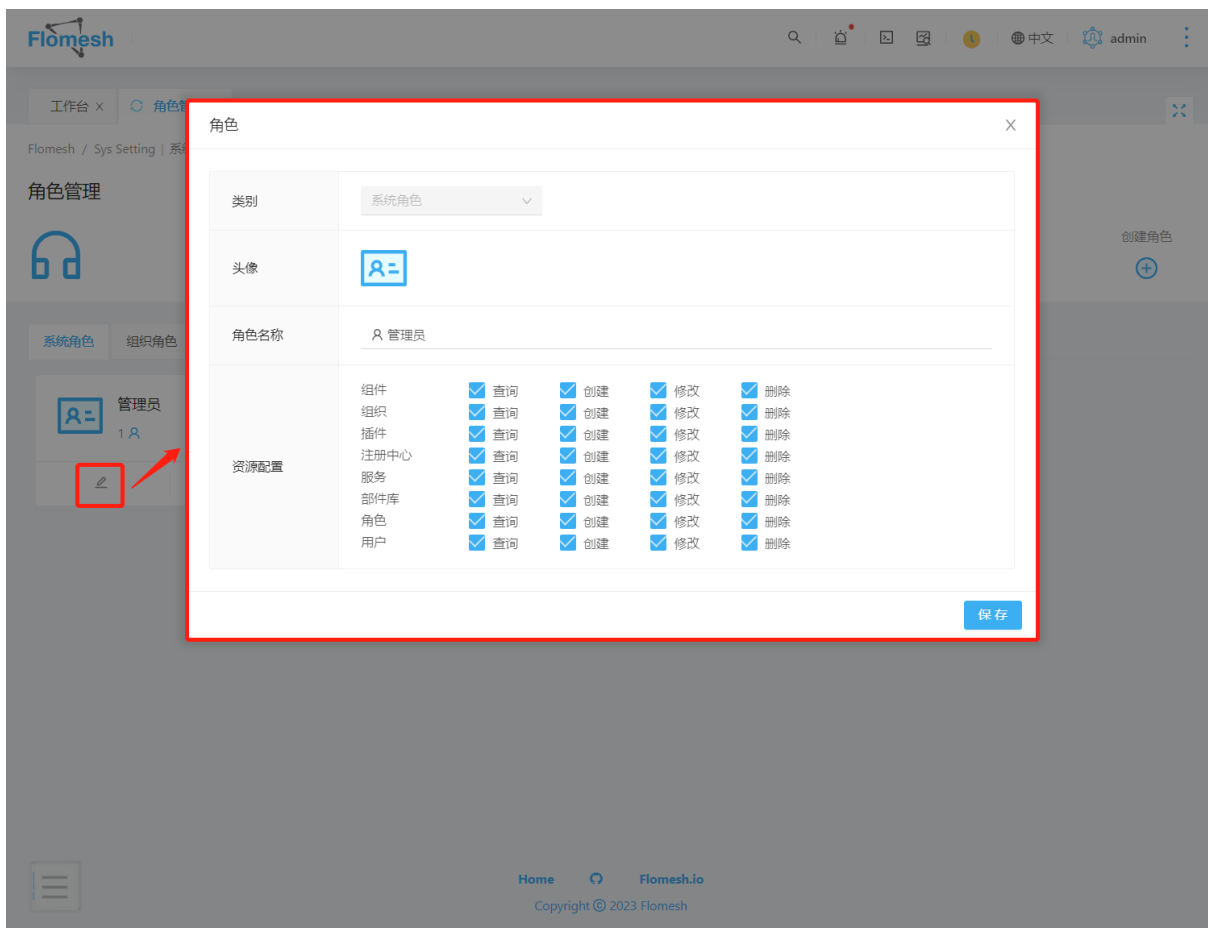
角色管理页面右上角单击“+”，进入新建角色页面，配置角色信息，角色信息参数如下表所示。

角色基本信息	
参数名称	说明
类别	必填项，下拉选择系统角色、组织角色或项目角色。
角色名称	必填项，自定义角色名称。
资源配置	必填项，勾选各资源的查询、创建、修改和删除权限，支持多选。



3.2.4.2. 编辑角色权限

在角色管理页面鼠标移到需要编辑的角色菜单，鼠标单击“编辑”，进入编辑角色页面，可以对类别、角色名称、资源配置进行修改。



3.3. 运维中心管理

3.3.1. 组件管理

组件管理指的是对平台所依赖的各类组件进行维护管理，点击 可创建组件

Flomesh / Ops Center | 运维中心 / 组件管理

组件管理



创建组件



3.3.1.1. Log

Log 组件是 FLB 用于收集 Pipy 进程产生的日志数据，并传递给各类日志数据库的组件。

在 GUI 配置前，需要提前获取 Log4pipy 进程运行地址 [HOST]: [PORT]。

ID	7		
名称	log	类别	Log
配置	日志类别: Clickhouse	日志表名: log	
	日志地址: clickhouse	日志输入路径: http://192.168.12.5:9876/logs	

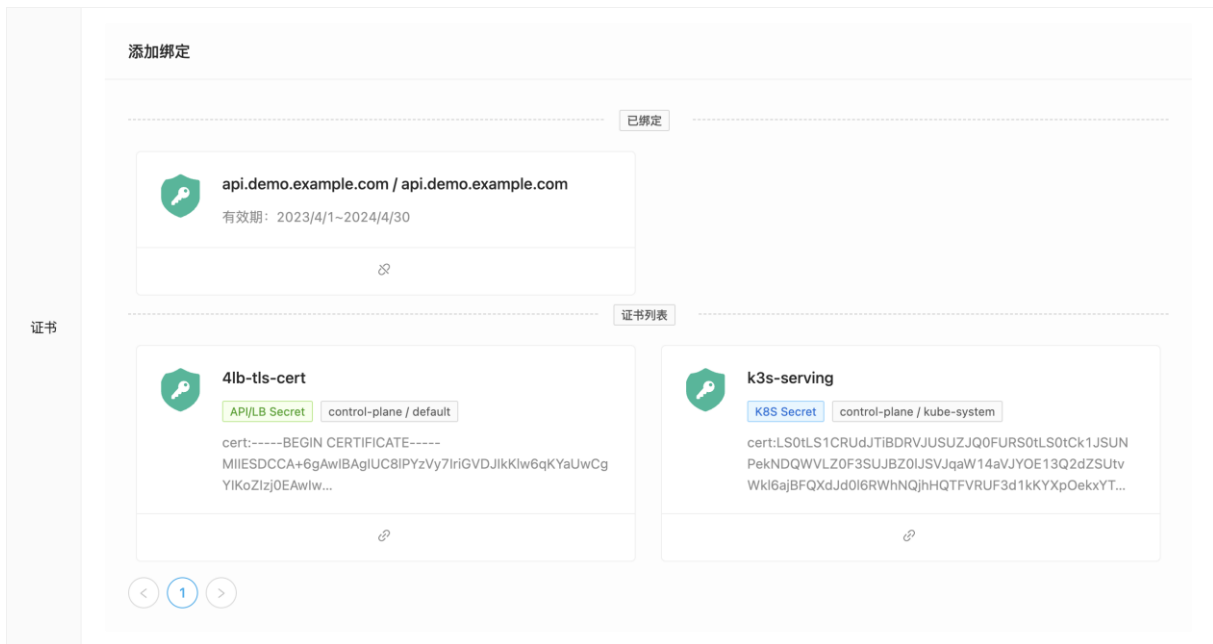
取消 保存



1. 名称：log 组件名称
2. 日志类别：日志服务器类型
3. 日志地址：选择日志服务器地址，日志服务器地址在 Clickhouse 组件内配置
4. 日志表名：日志存放的数据表名
5. 日志输入路径：Log 组件实际运行地址：http://[HOST]:[PORT]/logs

3.3.1.2. Pipy

Pipy 组件是 FLB 的一组 7LB/Open API 集群，在创建 7LB/Open API 策略时，将会选择对应的 Pipy 集群。

ID	9		
名称	demo-7lb	类别	Pipy
端口	Http: 8080 Https: 8443		
配置	日志: log	模板: default	
	连接超时时间: 未设置	读取超时时间: 未设置	
	写入超时时间: 未设置		



1. 名称：Pipy 集群名称
2. 端口：可配置 http 端口和 https 端口
3. 日志：选取 Log 组件
4. 模板：Pipy 的配置文件模板，默认为 default，如有定制需求，请联系 Flomesh 技术服务团队
5. 连接超时时间：表示指在客户端没有新数据传输的情况下，连接保持活动状态的最长时间。如果超过这个时间，Pipy 会主动断开连接，以释放资源
6. 读取超时时间：表示 Pipy 等待从客户端读取数据的最长时间。如果超过这个时间仍未收到请求，Pipy 会认为连接已经超时，自动关闭连接
7. 写入超时时间：表示该 Pipy 等待向客户端发送数据的最长时间。如果在规定时间内没有完成数据发送，Pipy 会认为连接已经超时，自动关闭连接
8. 证书：在“运维中心 | 证书管理”下配置的证书会在此处展示，点击  按钮，可绑定证书，点击  按钮，可解除绑定。
9. 在创建完成后，需要在 7LB/Open API 节点创建配置文件以及 systemd

```
# mkdir /etc/pipy-7lb/
```

```
# chown nobody /etc/pipy-7lb
```

```
创建配置文件 /etc/pipy-7lb/env.defaults
```

```
HOME_DIR=/etc/pipy-7lb

PIPY_INSTANCE_NAME=71b01

REPO="https://192.168.12.5:31060/repo/flomesh/pipy/demo-71b/"

# set extra args in ARGS if you need more arguments for pipy
# ARGS="--tls-trusted=/etc/pipy-7lb/ssl/ca.pem --tls-
cert=/etc/pipy-7lb/ssl/client.pem --tls-key=/etc/pipy-
71b/ssl/client-key.pem"
ARGS=""
```

配置参数说明如下：

1. HOME_DIR: 配置文件所在目录
 2. PIPY_INSTANCE_NAME: 当前 pipy 实例的名称，该名称会记录在请求日志以及 metric label 当中
 3. REPO: pipy repo 的地址，应为
`http://[HOST]:[PORT]/repo/flomesh/pipy/[Pipy 集群名称]/`
 4. ARGS: pipy 运行的额外参数，默认为空
10. 创建 systemd 配置 `/etc/systemd/system/pipy-7lb.service`

```

[Unit]
Description = pipy
After = network.target

[Service]
EnvironmentFile=-/etc/pipy-7lb/env.defaults
ExecStartPre=/bin/bash -c 'if [ ! -f ${HOME_DIR}/uuid ]; then
uuidgen > ${HOME_DIR}/uuid; fi'
ExecStart=/bin/bash -c "/usr/local/bin/pipy --reuse-port --
threads=max --instance-uuid=$(cat ${HOME_DIR}/uuid) --instance-
name=${PIPY_INSTANCE_NAME} $ARGS $REPO"
ExecStop=/usr/bin/kill $MAINPID
Restart=on-failure
User=nobody
AmbientCapabilities=CAP_NET_BIND_SERVICE
AmbientCapabilities=CAP_NET_ADMIN
LimitNOFILE=655360

[Install]
WantedBy = multi-user.target

```

保存后 systemd reload 并启动 pipy

```

# systemctl daemon-reload
# systemctl enable pipy-7lb
# systemctl start pipy-7lb

```

3.3.1.3. Pipy4lb

Pipy4lb 组件是 FLB 的一组 4LB 集群，在创建 4LB 策略时，将会选择对应的 Pipy4LB 集群。

ID	4		
名称	ft01	类别	Pipy4lb
组织	flomesh x		
配置	日志: log 全局最大连接数: 3000 默认读取超时时间: 60s 默认闲置超时时间: 60s 全局ID前缀: 未设置	模板: default 默认最大连接数: 1000 默认写入超时时间: 60s 最大LB数: 200	
健康检查	主机: localhost 间隔: 5s	端口: 8888 失败次数: 3	

1. 名称：Pipy4LB 集群名称
2. 组织：该 Pipy4LB 归属的组织
3. 日志：选取 Log 组件
4. 模板：Pipy4LB 的配置文件模板，默认为 default，如有定制需求，请联系 Flomesh 技术服务团队
5. 全局最大连接数：表示 Pipy4LB 节点支持的并发连接数上限
6. 默认最大连接数：表示每个 4LB 端点支持的并发连接数上限。该配置项在每个 4LB 策略中可单独配置，若不配置则安装此处配置生效
7. 默认读取超时时间：表示 Pipy4LB 等待从客户端读取数据的最长时间。如果超过这个时间仍未收到请求，Pipy4LB 会认为连接已经超时，自动关闭连接。该配置项在每个 4LB 策略中可单独配置，若不配置则安装此处配置生效
8. 默认写入超时时间：表示 Pipy4LB 等待向客户端发送数据的最长时间。如果在规定时间内没有完成数据发送，Pipy4LB 会认为连接已经超时，自动关闭连接。该配置项在每个 4LB 策略中可单独配置，若不配置则安装此处配置生效
9. 默认闲置超时时间：表示指在客户端没有新数据传输的情况下，连接保持活动状态的最长时间。如果超过这个时间，Pipy4LB 会主动断开连接，以释放资源。该配置项在每个 4LB 策略中可单独配置，若不配置则安装此处配置生效
10. 健康检查配置：
 - a. 主机：健康检查节点地址
 - b. 端口：健康检查节点端口
 - c. 间隔：健康检查间隔，Pipy4LB 节点对后端服务器组进行 TCP/UDP 端口探活的间隔时间
 - d. 失败次数：健康检查间隔内失败次数，Pipy4LB 节点对后端服务器探活失败超过该失败次数后，将会判断该 4LB 后端服务失效，停止将链接负载均衡到该后端服

务，直到后续间隔时间内探活失败次数小于该失败次数。

11. 在创建完成后，需要在 4LB 节点配置 pipy 以及 systemd

```
# mkdir /etc/pipy-4lb/
```

```
# chown nobody /etc/pipy-4lb
```

创建配置文件 /etc/pipy-4lb/env.defaults

```
# The dir of this configuration file
HOME_DIR=/etc/pipy-4lb

# The name for the pipy
PIPY_INSTANCE_NAME=pipy

# The name for the pipy
LB_ID="pipy"

# IP of the node running pipy
NODE_IP=""

# The repo address
REPO="http://localhost:6060/repo/path/to/pipy-repo"

# Set extra args in ARGS if you need more arguments for pipy
#ARGS="--tls-trusted=/etc/pipy/ssl/ca.pem --tls-
cert=/etc/pipy/ssl/client.pem --tls-key=/etc/pipy/ssl/client-
key.pem"
ARGS=""
```

配置参数说明如下：

1. HOME_DIR: 配置文件所在目录
2. PIPY_INSTANCE_NAME: 当前 pipy 实例的名称，该名称会记录在 metric label 当中
3. LB_ID: 当前 pipy 实例的名称或 ID，该名称会记录在日志当中
4. NODE_IP: 当前 pipy 实例所在节点的 IP 地址，该信息回记录在日志当中
5. REPO: pipy repo 的地址，应为
`http://[HOST]:[PORT]/repo/flomesh/pipy4lb/[Pipy4LB 集群名称]/`
6. ARGS: pipy 运行的额外参数，默认为空

配置 systemd 启动文件 /etc/systemd/system/pipy-4lb.service 如下

```
[Unit]
Description = pipy
After = network.target

[Service]
EnvironmentFile=-/etc/pipy-4lb/env.defaults
ExecStartPre=/bin/bash -c 'if [ ! -f ${HOME_DIR}/uuid ]; then
uuidgen > ${HOME_DIR}/uuid; fi'
ExecStart=/bin/bash -c "/usr/local/bin/pipy --reuse-port --
threads=max --instance-uuid=$(cat ${HOME_DIR}/uuid) --instance-
name=${PIPY_INSTANCE_NAME} $ARGS $REPO"
ExecStop=/usr/bin/kill $MAINPID
Restart=on-failure
User=nobody
AmbientCapabilities=CAP_NET_BIND_SERVICE
AmbientCapabilities=CAP_NET_ADMIN
LimitNOFILE=655360

[Install]
WantedBy = multi-user.target
```

保存后 systemd reload 并启动 pipy

```
# systemctl daemon-reload
# systemctl enable pipy-4lb
# systemctl start pipy-4lb
```

Pipy4lb 组件可配置 Pipy BGP 策略，以实现 IP 高可用和 BGP 宣告。

```
BGP Speaker 配置

1 {
2   "as": 4266620060,
3   "id": "0.0.0.0",
4   "peers": [
5     "192.168.12.2"
6   ],
7   "ipv4": {
8     "nextHop": "0.0.0.0",
9     "reachable": [],
10    "unreachable": []
11  },
12  "ipv6": {
13    "nextHop": "::",
14    "reachable": [],
15    "unreachable": []
16  }
17 }
```

页面上需要配置的参数如下：

1. "as": BGP speaker 的 AS 号
2. "peers": BGP speaker 的邻居节点

"as" 和 "peers" 为集群共同的配置，BGP speaker 的 id, nextHop 会在 speaker 节点上通过环境变量配置。reachable 和 unreachable 为系统依照地址池自动生成的配置，不需要在此处填写。

BGP speaker 节点配置，在 /etc/pipy-speaker/env.defaults 下

```
# Set BGP Speaker ID, in general, it's a IP address of the host
BGP_SPEAKER_ID=

# Set BGP IPv4 nexthop, in general, it's a IP address of the host
BGP_SPEAKER_IPV4_NEXTHOP=

# Set BGP IPv6 nexthop, in general, it's a IP address of the host
BGP_SPEAKER_IPV6_NEXTHOP=

# the speaker repo address
REPO="http://localhost:6060/repo/path/to/speaker-repo"

# Set extra args in ARGS if you need more arguments for pipy
#ARGS="--tls-trusted=/etc/pipy-speaker/ssl/ca.pem --tls-
cert=/etc/pipy-speaker/ssl/client.pem --tls-key=/etc/pipy-
speaker/ssl/client-key.pem"
ARGS=""
```

配置项如下：

1. BGP_SPEAKER_ID 为 BGP speaker 节点 ID，通常可设置为节点 IP
2. BGP_SPEAKER_IPV4_NEXTHOP 为 BGP speaker 宣告 IPv4 网段的下一跳地址，设置为当前 speaker 节点的 IPv4 地址
3. BGP_SPEAKER_IPV6_NEXTHOP 为 BGP speaker 宣告 IPv6 网段的下一跳地址，设置为当前 speaker 节点的 IPv6 地址
4. REPO 设置为 4LB 集群对应的 BGP repo 地址，为
`http://[HOST]:[PORT]/repo/flomesh/pipy4lbbgp/[Pipy4LB 集群名称]/`
5. ARGS 为 speaker 启动所需的额外参数，默认可不填

BGP speaker 的 systemd 配置文件如下

3.3.1.4. Prometheus

FLB 支持对接 Prometheus[[Data model](#) | [Prometheus](#)]接口，用来收集展示监控数据。

ID	8		
名称	<input type="text" value="prometheus"/>	类别	Prometheus
配置	<p>YAML JSON</p> <pre>1 --- 2 host: "http://10.43.151.104:7070" 3</pre>		

1. 名称：组件名称
2. host：Prometheus 的访问地址

3.3.1.5. Clickhouse

FLB 支持对接各类日志数据库，其中包括 Clickhouse，用来展示 FLB 访问日志。

组件属性 ×

名称	<input type="text" value="clickhouse"/>	类别	Clickhouse
配置	<p>YAML JSON</p> <pre>1 --- 2 host: "192.168.68.11" 3 port: 30023 4 user: "flomesh" 5 password: "password" 6 database: "default" 7</pre>		

1. 名称：组件名称
2. host：Clickhouse 地址
3. port：Clickhouse 端口
4. user：数据库用户名

5. password : 数据库密码
6. database : 数据库名称

3.3.1.6. Checkpoint

Checkpoint 用来对 7 层服务进行拨测，实现 http/https 服务的健康状态检测。

名称	checkpoint	类别	Checkpoint
模板	default		
配置	<pre>YAML JSON 1 --- 2 reportURL: "http://192.168.12.5:8888/" 3 listen: "8888" 4 interval: "5s" 5 failures: "3" 6</pre>		

1. 名称 : 组件名称
2. 模板 : 默认为 default, 如需定制, 请联系 flomesh 技术服务团队
3. 健康检查配置 :
4. reportURL : 健康检查节点地址
5. listen : 健康检查节点端口
6. interval : 健康检查间隔, Checkpoint 节点对后端服务器组进行 http/https 端口探活的间隔时间
7. failures : 健康检查间隔内失败次数, Checkpoint 节点对后端服务器探活失败超过该失败次数后, 将会判断该后端服务失效, 停止将链接负载均衡到该后端服务, 直到后续间隔时间内探活失败次数小于该失败次数。

3.3.2. 证书管理

FLB 支持导入 SSL/TLS 证书, 并绑定到 4LB/OpenAPI 节点上, 实现证书卸载。

名称	4lb-tls-cert
类别	API/LB Secret
证书 (cert)	-----BEGIN CERTIFICATE----- MIIESDCCA+6gAwIBAgIUc8IPYzVv7IriGVDJkKlw6qKYaUwCgYIKo ZlZj0EAWlw RzELMAkGA1UEBhMCQ04xETAPBgNVBAGTCENoYW95YW5nMR
证书 (key)	-----BEGIN RSA PRIVATE KEY----- MIIJKAIBAAKCAgEAsdQtXv5vfi8FLULZ3gtFkBqloyLu+t7V9ui6XrG FAV+Tjb3A ZQ8/G2/sf2V/m9lg+lepXl+5Rsq4VeNoSqLNtB77Lr36eH1xJ495U

取消

保存

3.3.3.告警配置

FLB 支持配置告警策略。

工作台 x 告警配置 x 新建告警规则 x

Flomesh / Ops Center | 运维中心 / 新建告警规则

alarm 添加接收者 创建

名称: alarm

规则: ! resSize > 90 限制

渠道: 系统设置 邮箱

作用域 接收者

服务目标 API目标 全局应用

1 项 target

1 项 已添加

合计 1 < 1 >

3.3.4.注册中心

FLB 支持同步多个注册中心信息，完成跨区域的服务注册发现。



3.3.5.操作记录

FLB 支持查看平台操作记录。

用户	IP	操作	名称	更新时间	内容
admin	::ffff:10.42.0.0	LoginSuccess	admin	2023/4/11 09:55:08	接口数据
admin	::ffff:10.42.0.0	LoginSuccess	admin	2023/4/11 09:34:09	接口数据
admin	::ffff:10.42.0.0	LoginSuccess	admin	2023/4/10 20:02:54	接口数据
admin	::ffff:10.42.0.0	LoginSuccess	admin	2023/4/10 17:34:52	接口数据

3.3.6.Pipy 推送日志

FLB 支持查看 Flomesh-GUI 对 Pipy-repo 发起的配置推送记录。仅在配置变动或系统重启情况下才会进行配置推送。

路径	类别	Version	更新时间	内容
/flomesh/pipy4lbbgp/ft01	post	2557	2023/4/11 09:33:46	数据
/flomesh/pipy4lb/ft01	post	2557	2023/4/11 09:33:46	数据
/flomesh/pipy4lbbgp/ft01	post	2552	2023/4/11 09:33:41	数据
/flomesh/pipy4lb/ft01	post	2552	2023/4/11 09:33:41	数据

内容

```
1 {
2   "/config.json": "{\n  \"as\": 4266628868,\n  \"id\": \"0.0.0.0\",\n  \"peers\": [\n    \"192.168.12.2\"\n  ],\n  \"ipv4\": {\n    \"nextHop\": \"0.0.0.0\",\n    \"
3 }
```

3.4. 4LB 管理

详见《Flomesh 软件负载用户手册-v1.0》

3.5. 7LB/OpenApi 管理

详见《Flomesh 软件负载用户手册-v1.0》