



大连易衡科技有限公司

# Flomesh

## 软负载用户手册

### 修订记录

版本	修订人	审核人	修订内容	说明	修改时间
V1.0					

# 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。

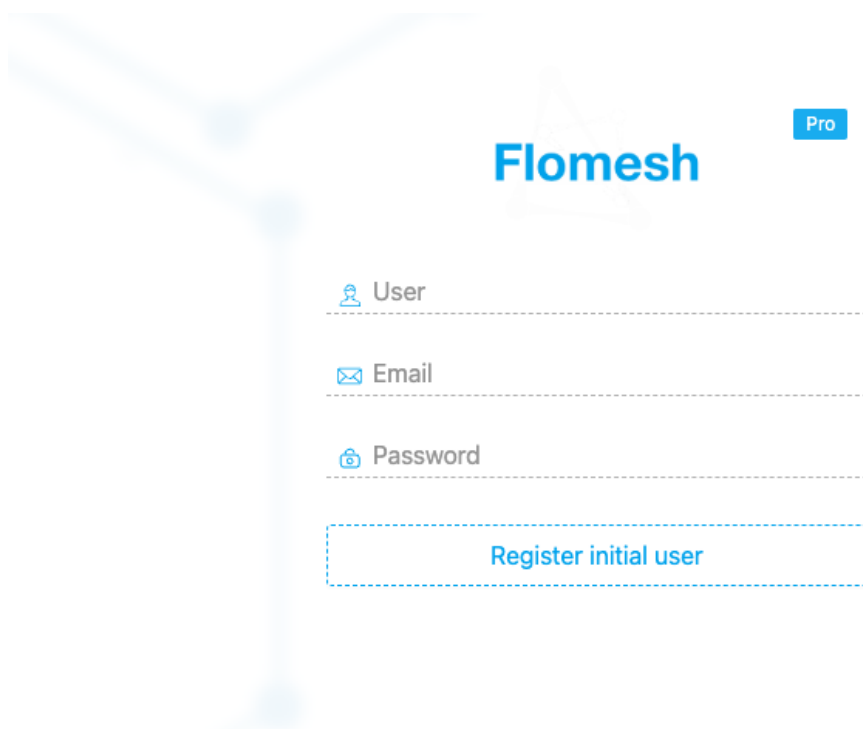
## 1.2 定义

Flomesh 图形用户界面简称：Flomesh GUI。

# 2. 用户登录

平台登录地址：[http://\[HOST\]:\[PORT\]/flomesh-gui/#/login](http://[HOST]:[PORT]/flomesh-gui/#/login)

## 2.1 新用户注册



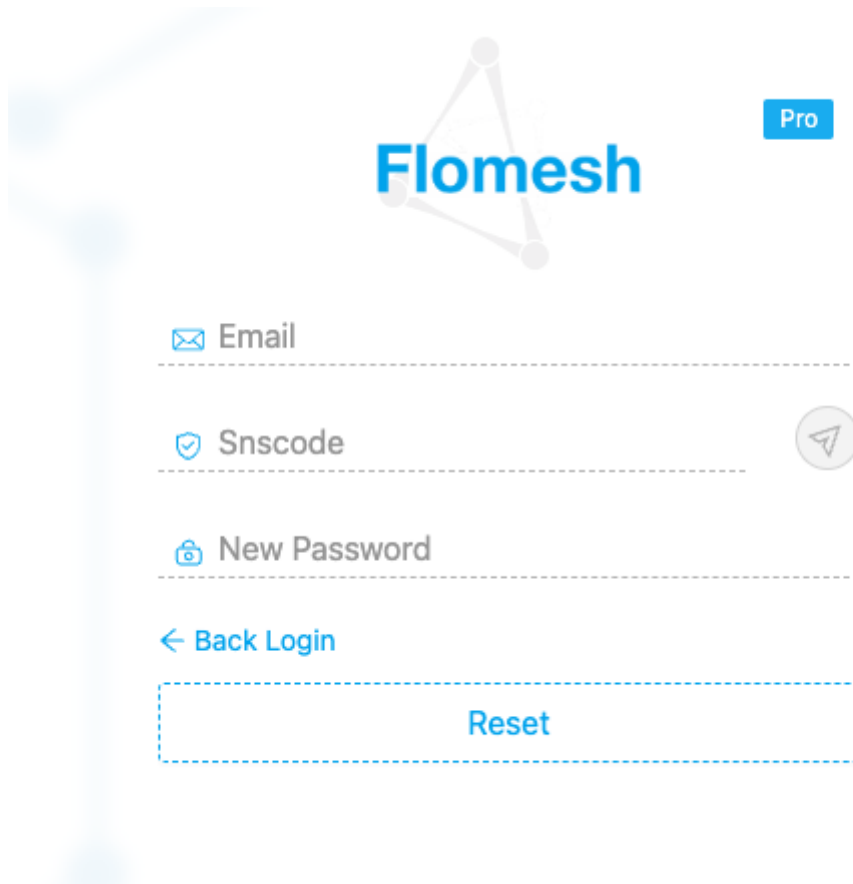
具体操作：打开平台登录地址，首次打开会自动跳转到注册页面，进行新用户的注册。

## 2.2 用户登录



具体操作：打开平台登录地址，输入用户名和密码进行登录。

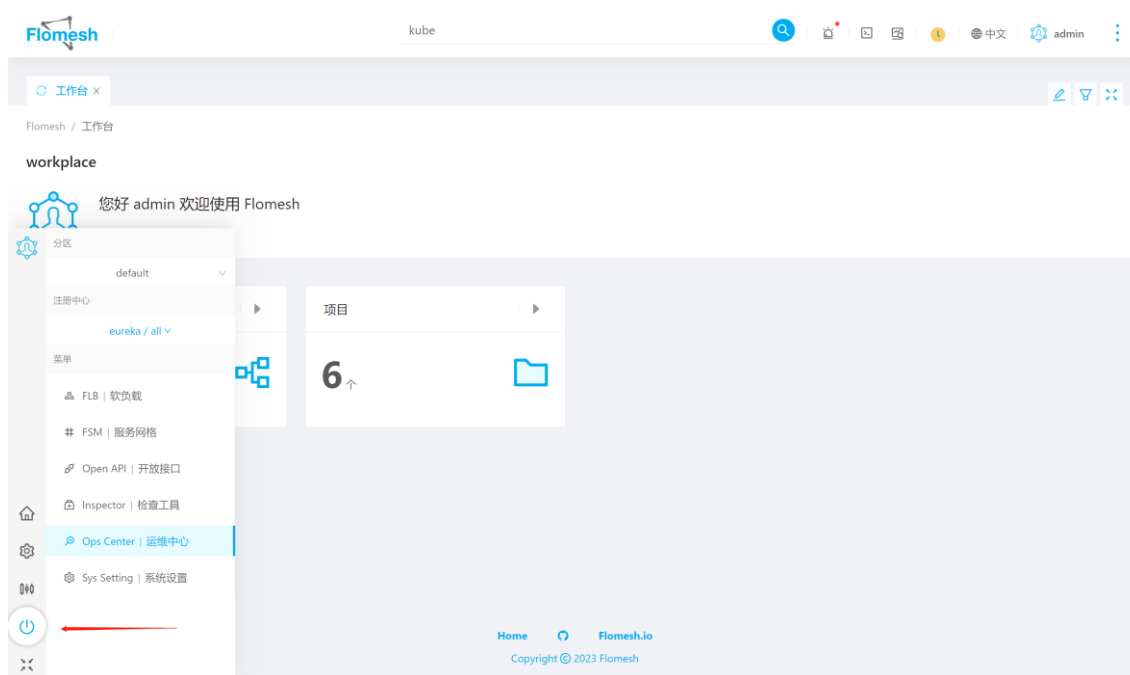
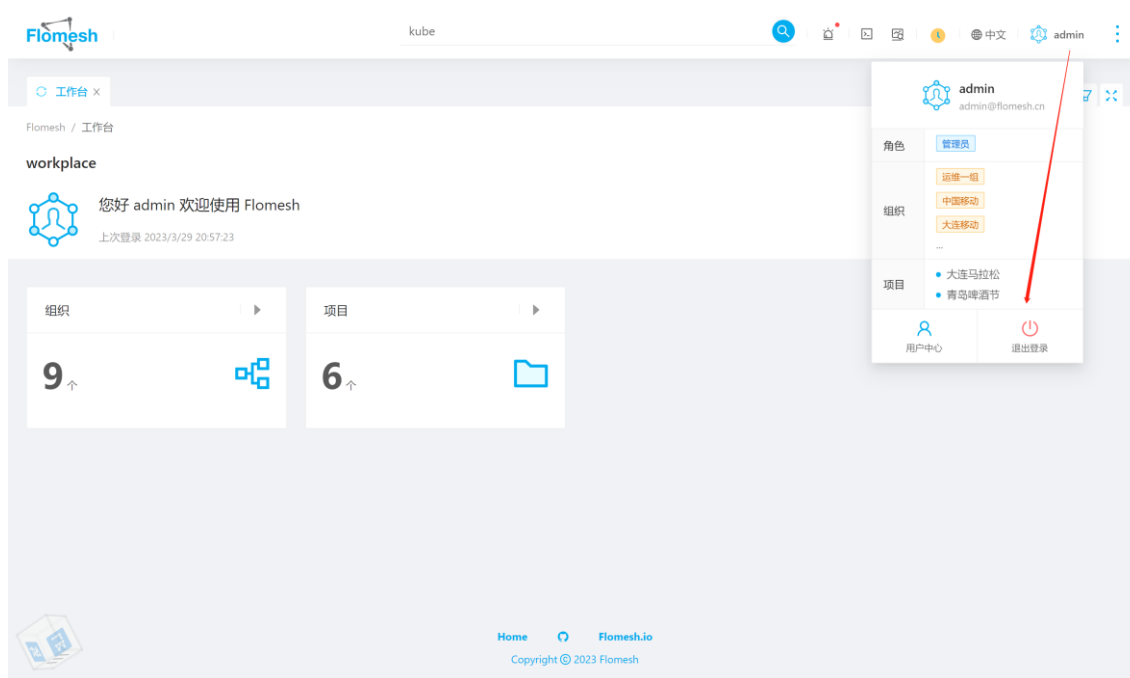
## 2.3 密码找回



The image shows a web interface for password recovery. At the top center is the 'Flomesh' logo, which consists of a blue triangle with a white dot at each vertex and a white circle in the center. To the right of the logo is a blue 'Pro' badge. Below the logo are three input fields, each with a dashed line underneath. The first field is labeled 'Email' with an envelope icon. The second field is labeled 'Snscode' with a shield icon and a circular arrow icon to its right. The third field is labeled 'New Password' with a padlock icon. Below these fields is a blue link that says '< Back Login'. At the bottom of the form is a blue dashed rectangular button labeled 'Reset'.

具体操作：单击忘记密码，输入邮箱，单击发送验证码，输入从邮箱中获得的验证码，输入新密码，单击重置。

## 2.4 退出登录



具体操作：点击右上角退出登录按钮，或点击左下角退出登录按钮。

## 2.5 平台权限管理说明

FLB 提供基于组织、项目、角色、用户的权限管理机制。在登录后，用户可查看自己所属角色、组织和项目。

组织树概念：

## 组织树



组织是 FLB 中最大的权限划分范围。组织下可创建多个项目，组织下可创建下级组织。

用户需要绑定组织和项目，并分配不同的组织角色和项目角色。

用户只能在绑定的组织角色、项目角色下进行相关资源及配置的查看、配置删除操作。

**admin**  
yowang@flomesh.cn

角色	管理员
组织	flomesh
项目	• project01

 用户中心       退出登录

## 3. 基础功能

### 3.1 服务搜索

服务搜索可以快速搜索系统中的服务，并从搜索结果中快捷访问服务信息。

工作台 x

🔍 ⏴ ⏵

Flomesh / 工作台

### workplace



您好 admin 欢迎使用 Flomesh

上次登录 2023/3/29 20:57:23

组织

9个



项目

6个



[Home](#) [Flomesh.io](#)

Copyright © 2023 Flomesh



名称: kube-dns  
ID: 28



Invalid Date | 命名空间: kube-system | UID: 7193def6-298c-4a07-ab4c-0b4a9fa05385



名称: kube-dns  
ID: 2

Invalid Date | 命名空间: kube-system | UID: aa97d251-153f-4100-87fc-b3710b1711e2



名称: kube-dns  
ID: 63

Invalid Date | 命名空间: kube-system | UID: 762facae-e0bd-4c71-a522-a88c113e4183



名称: kubernetes  
ID: 35

Invalid Date | 命名空间: default | UID: b6353754-f848-4d1c-9e2a-c4dc8e87edf6



名称: kubernetes  
ID: 1

Invalid Date | 命名空间: default | UID: ec5b1d82-6d3c-4b43-973f-3bb641e71c51

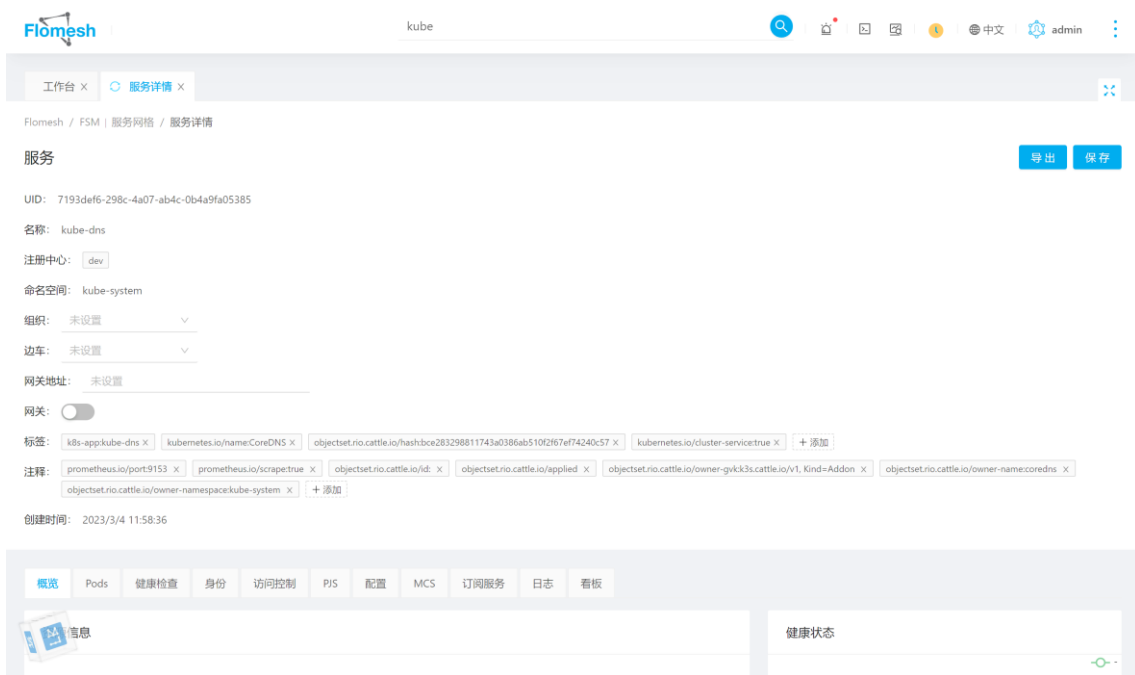


名称: kubernetes  
ID: 66

Invalid Date | 命名空间: default | UID: b8203ea8-83ce-4228-8c97-e009c282c958



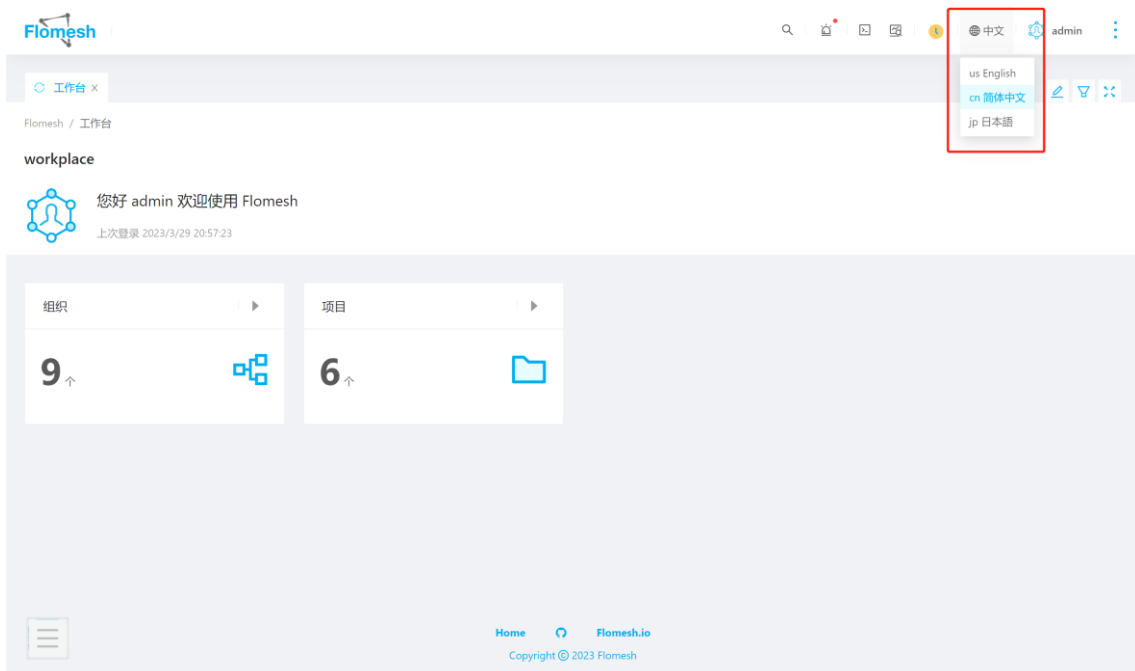
名称: kubernetes-dashboard  
ID: 16



具体操作：单击右上角搜索按钮，输入关键字，单击搜索按钮，显示包含关键字的服务，单击服务名称可以查看该服务详情。

## 3.2 语言切换

目前支持中文、英语和日语。



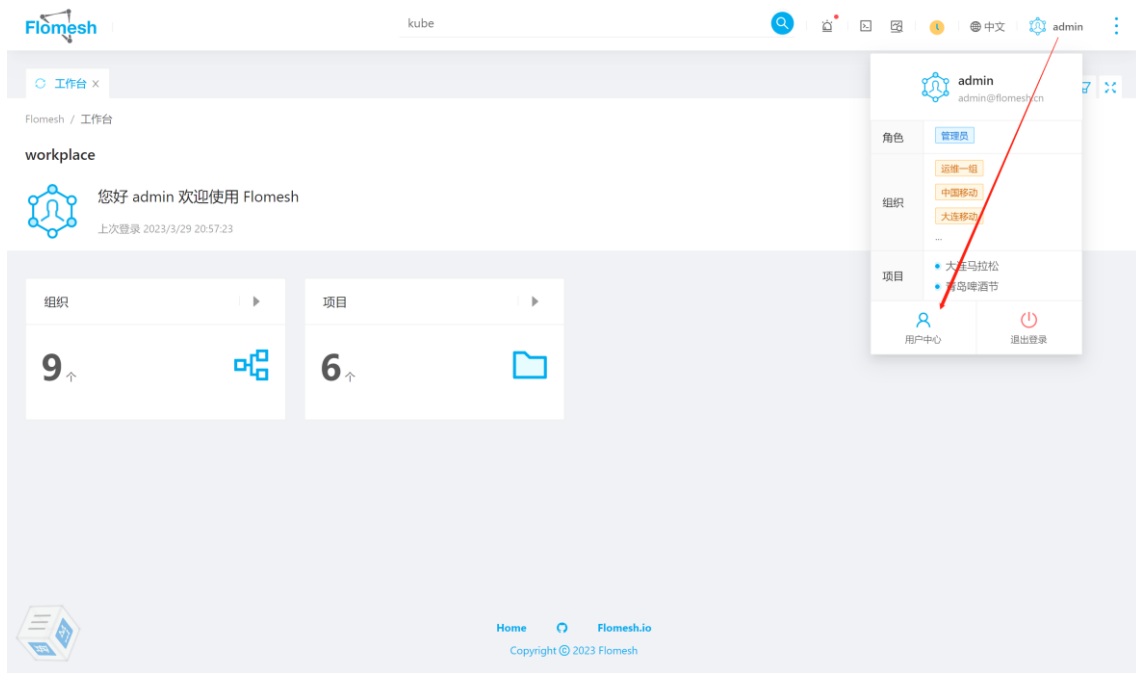
操作步骤：选择右上角语言下拉菜单，单击选择一个语言，系统页面语言显示为所选择语种。

## 3.3 用户中心

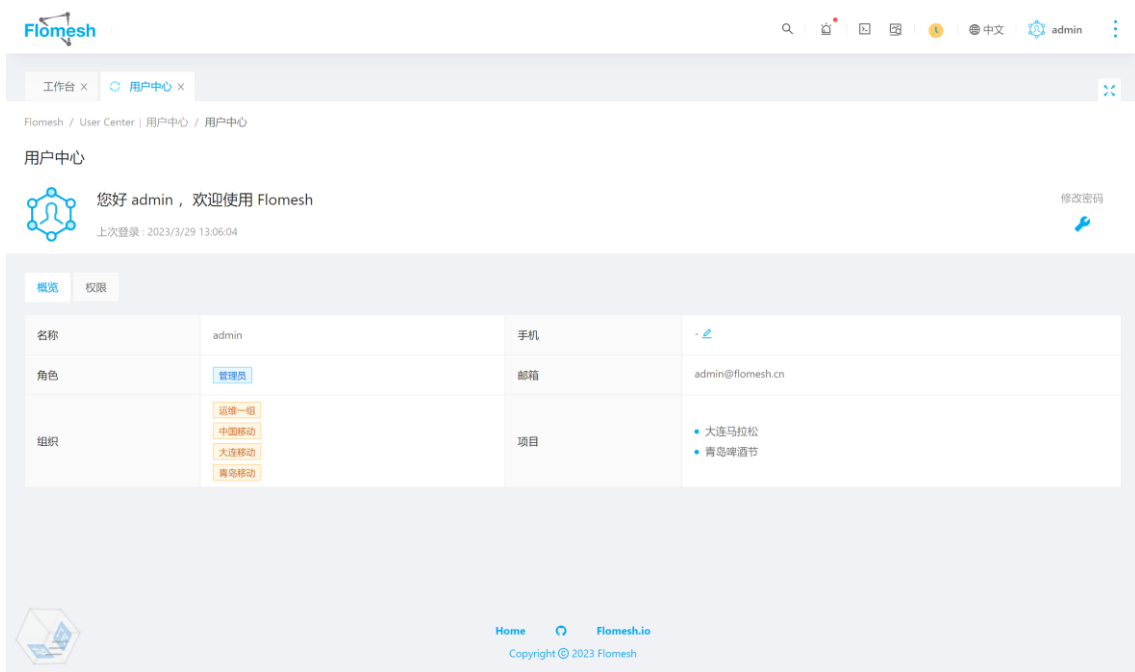
用户中心为当前登录用户提供用户相关信息的管理，如用户详情和密码相关的操作。



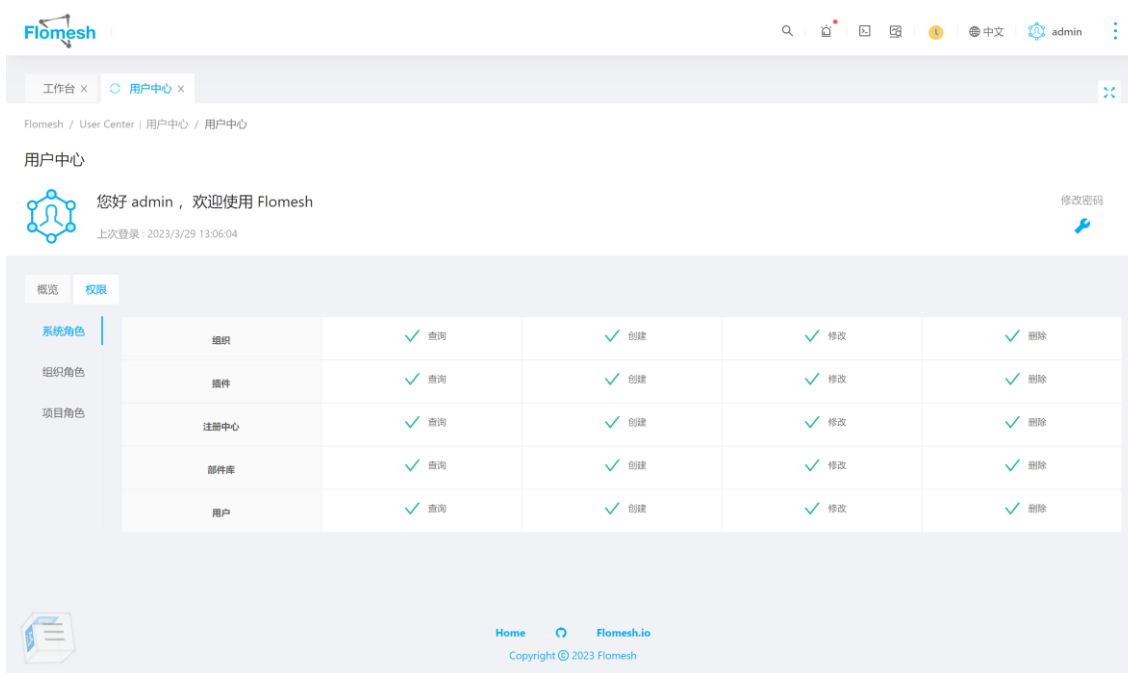
### 3.3.1 用户详情展示



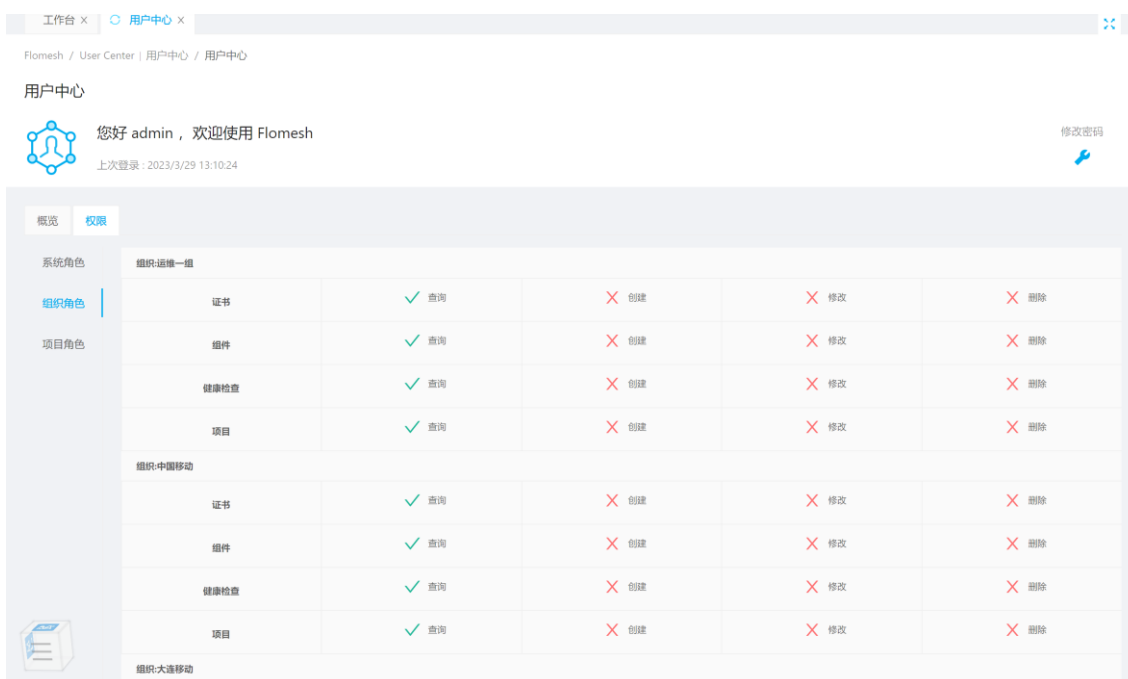
用户中心入口



用户中心工作台



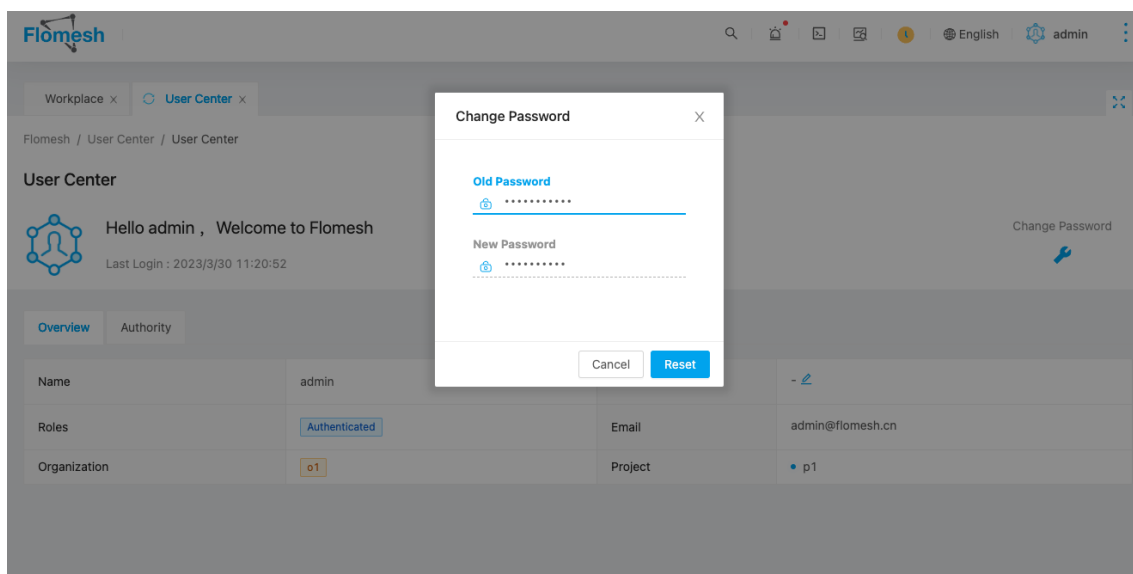
用户系统角色权限详情



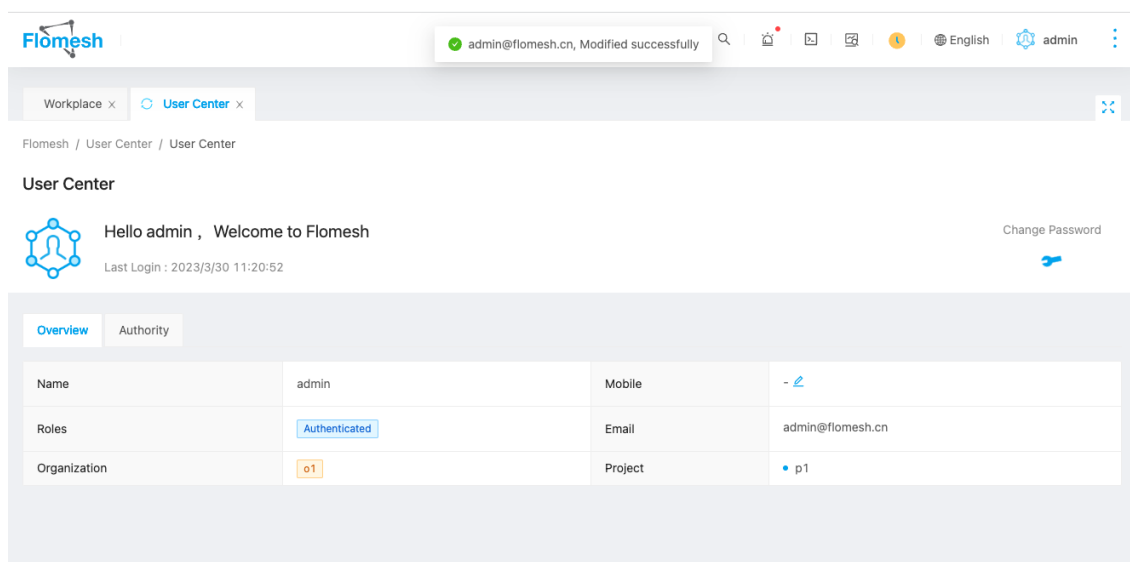
用户组织角色权限详情

具体操作：单击用户中心选项，进入到用户中心工作台，显示用户概览和正确权限详情，以及与用户匹配的角色和组织。

### 3.3.2 用户密码修改



1.



具体操作：在用户中心工作台，单击修改密码，输入新密码，单击重置，提示密码修改成功，退出当前登录，使用新密码登录。

### 3.3.3 修改用户手机号

Flomesh / User Center | 用户中心 / 用户中心

用户中心

您好 admin，欢迎使用 Flomesh  
上次登录：2023/3/29 13:22:51

修改密码

概览 权限

名称	admin	手机	13011011011
角色	管理员	邮箱	admin@flomesh.cn
组织	<ul style="list-style-type: none"><li>运维一组</li><li>中国移动</li><li>大连移动</li><li>青岛移动</li></ul>	项目	<ul style="list-style-type: none"><li>大连马拉松</li><li>青岛啤酒节</li></ul>

Home Flomesh.io  
Copyright © 2023 Flomesh

Flomesh / User Center | 用户中心 / 用户中心

用户中心

您好 admin，欢迎使用 Flomesh  
上次登录：2023/3/29 13:22:51

修改密码

概览 权限

名称	admin	手机	13011011011
角色	管理员	邮箱	admin@flomesh.cn
组织	<ul style="list-style-type: none"><li>运维一组</li><li>中国移动</li><li>大连移动</li><li>青岛移动</li></ul>	项目	<ul style="list-style-type: none"><li>大连马拉松</li><li>青岛啤酒节</li></ul>

Home Flomesh.io  
Copyright © 2023 Flomesh

具体操作：在用户中心工作台，单击编辑手机，输入新手机号，退出编辑模式，新手机号保存成功。

## 4. 四层负载均衡操作指导

FLB 提供 TCP/UDP 四层负载均衡能力。在 “FLB | 软负载” 功能菜单下，支持配置地址池、4LB 规则，提供看版和操作记录展示功能。



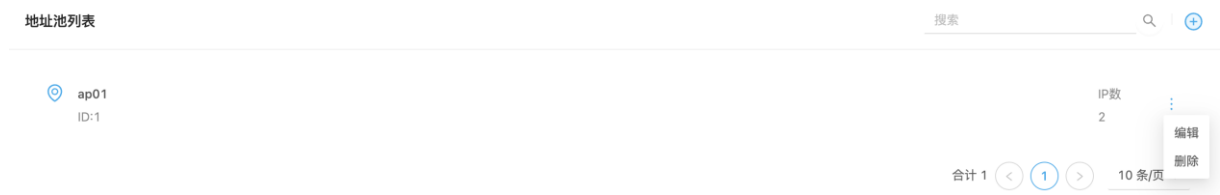
### 4.1 地址池配置

#### 4.1.1 什么是地址池配置

地址池是 FLB 对应用服务的 IP 地址进行管理的功能。地址池代表一组提供相同应用服务的 IP 地址。一个地址池配置中可包含多个 IP 池，便于实现不同地区的用户访问不同的 IP 池，并达到就近接入的效果。地址池配置包括：地址池名称、地址池所属组织、地址池类别、IP 池。

## 4.1.2 功能说明

### 4.1.2.1 地址池列表查看



#### 1. 地址池列表

地址池列表会在地址池列表展示所有地址池名称/ID 及 IP 数。

#### 2. 地址池名称/ID

创建地址池时，FLB 会校验地址池名称，确保系统中不会出现两个同样的地址池名称

创建地址池后，FLB 会自动创建一个地址池 ID，作为地址池唯一标识，用于系统识别管理

#### 3. IP 数

IP 数表示该地址池包含 IP 数量

#### 4. 新建地址池

创建新地址池的入口。单击 “+” 按钮，会进入地址池创建界面。

#### 5. 编辑地址池

创建地址池后，可继续编辑该地址池配置。单击 “编辑” 按钮，会进入地址池详情界面，可对地址池进行编辑。

### 4.1.2.2 新建地址池/地址池详情



#### 1. 名称

地址池名称，用于区分不同地址池，在创建 4LB 规则时需要根据名称选择使用的地址池名称。

#### 2. 组织

地址池所属组织，用于区分资源权限。组织用户在创建 4LB 规则时，仅能选择所在组织拥

有的地址池。

### 3. 类别/IP 池

FLB 支持 IPv4 和 IPv6，在地址池类别中，可按照 IP 段或者 CIDR 方式来配置 IP 池范围。

## 4.2 4LB 规则配置

### 4.2.1 什么是 4LB 规则配置

4LB 规则包括监听端口、协议配置、TLS 配置、负载均衡算法配置、负载均衡后端服务器组配置，提供高级策略配置（限速）、健康检查配置和日志功能。

### 4.2.2 功能说明



The screenshot shows a web interface for managing 4LB configurations. At the top, there are tabs for '工作台' and '4LB'. Below the tabs is a table titled '4LB列表'. The table has columns for '名称', '全局ID', '组织', '地址池', '上游状态', '端点', '更新时间', and '操作'. Two entries are visible: 'flomesh/fsm-ingress-pipy-controller:80' and 'lb01'. At the bottom right, there are pagination controls showing '合计 2', '1', and '10 条/页'.

名称	全局ID	组织	地址池	上游状态	端点	更新时间	操作
flomesh/fsm-ingress-pipy-controller:80	cd43927f-a12c-4c0c-bd85-574d753e3a49	flomesh	ap01		100.100.100.2:80	2023/3/25 08:54:12	⋮
lb01	050dead9-d1bb-444e-b4aa-e8afac7dfa42	flomesh	ap01		100.100.100.1:80	2023/3/23 14:23:07	⋮

#### 4.2.2.1 4LB 列表查看

##### 1. 4LB 列表

4LB 列表会展示所有 4LB 配置，包含 4LB 名称、全局 ID、组织、地址池、上游状态、端点和更新时间。

##### 2. 名称

创建 4LB 时，FLB 会校验 4LB 名称，确保系统中不会出现两个同样的 4LB 名称。

创建 4LB 后，FLB 会自动创建一个 4LB 全局 ID，作为该 4LB 唯一标识，用于系统识别管理。

##### 3. 组织

4LB 所属组织，用于区分资源权限。

##### 4. 地址池

4LB 关联的地址池，用户在创建 4LB 规则时，仅能选择所在组织拥有的地址池。

##### 5. 端点

4LB 对外暴露的 IP+端口。

##### 6. 上游状态

FLB 针对 4LB 服务提供 TCP/UDP 主动健康检查，此处展示 4LB 上游服务运行状态。

## 7. 新建 4LB 规则

创建新 4LB 规则的入口。单击 “+” 按钮，会进入地址池创建界面。

## 8. 编辑 4LB 规则

创建 4LB 后，可继续编辑该 4LB 配置。单击 “编辑” 按钮，会进入 4LB 详情界面，可对 4LB 进行编辑。

### 4.2.2.2 新建 4LB

#### 4.2.2.2.1 4LB 基础规则配置

工作台 x 4LB x 4LB 详情 x 新建 4LB x

Flomesh / FLB | 软负载 / 新建 4LB

### 4lb-01 创建

名称: 4lb-01 端口: 1883

组织: flomesh 唯一IP:

地址池: ap01 协议: TCP

集群: ft01 源地址池: 192 . 168 . 10 . 0 / 24

TLS可用:  TLS端口: 443

证书: 未设置

#### 1. 名称

4LB 显示名称，创建 4LB 时，会校验 4LB 名称，确保系统中不会出现两个同样的 4LB 名称。

#### 2. 组织

4LB 规则所属组织，选择对应组织后，仅该系统管理员和组织成员能够修改该 4LB 规则。

#### 3. 地址池

4LB 关联的地址池，用户在创建 4LB 规则时，仅能选择所在组织拥有的地址池。

#### 4. 集群

4LB 规则所在 4LB 集群，选择集群后，该 4LB 规则将在对应 4LB 集群的 4LB 节点上生效。

注：4LB 集群创建与配置参考《Flomesh 管理员手册》

#### 5. 端口

4LB 规则对外监听的端口

#### 6. 唯一 IP

当选择唯一 IP 时，该 4LB 规则将会独自占用所配置地址池内的 IP 地址，该 4LB 规则所在集群内，其他 4LB 规则无法使用相同的地址池，否则相反。

#### 7. 协议

FLB 支持配置 TCP/UDP 协议负载均衡

#### 8. 源地址池



源地址池指的是从 4LB 节点访问上游服务的源 IP。

默认为 0.0.0.0/0，表示不指定源 IP，使用 4LB 本地 IP 地址访问上游服务（注：此模式下，从单个 4LB 节点到单个上游服务的最大并发连接数受 4LB 节点最大可用端口数限制）。

可以 CIDR 模式配置 4LB 节点源地址池。该模式下，4LB 访问上游将从源地址池中均匀选择 IP 访问上游服务，从单个 4LB 节点到单个上游服务的最大并发连接数受 4LB 节点最大可用端口数限制 x 源地址池 IP 数量)

## 9. TLS 可用

FLB 支持在 4LB 节点完成 TLS 卸载

### a. TLS 端口

TLS 监听端口

### b. 证书

选择 TLS 卸载需要绑定的证书，证书配置详情见——。

## 4.2.2.2.2 4LB 后端服务器组配置

The screenshot shows a configuration page for 4LB with three tabs: '配置' (Configuration), '策略' (Policy), and '健康检查' (Health Check). The '配置' tab is active. It is divided into three sections: '目标' (Target), '负载均衡' (Load Balancing), and '配置' (Configuration).  
1. '目标' (Target): A text input field for '后端服务器组' (Backend Server Group) contains the text '2.2.2.2:9000' and '3.3.3.3:9000'. Below the input is a placeholder text: 'List of <ip>:<port>[@<weight>]'.  
2. '负载均衡' (Load Balancing): A dropdown menu for '算法' (Algorithm) is set to 'RoundRobinLoadBalancer'.  
3. '配置' (Configuration): This section contains several settings:  
- '会话保持' (Session Persistence): A toggle switch is turned off.  
- '最大失败次数' (Max Failure Count): '未设置' (Not set).  
- '最大连接数' (Max Connections): '未设置' (Not set).  
- '读取超时时间' (Read Timeout): '未设置' (Not set).  
- '写入超时时间' (Write Timeout): '未设置' (Not set).  
- '闲置超时时间' (Idle Timeout): '未设置' (Not set).

### 1. 后端服务器组

4LB 规则对应的后端服务地址，配置例子：

2.2.2.2:9000@100

3.3.3.3:9000@100

test.com:9000@200

注：@后的数值表示权重，后端服务器组可配置 IP+Port 形式或域名+Port 形式

### 2. 算法

FLB 支持以下几种算法：

- 轮询算法 (RoundRobinLoadBalancer, 支持按权重轮询, 配置例子如上)
- 最小连接数算法 (LeastWorkLoadBalancer)
- 哈希算法 (HashingLoadBalancer)

#### 4.2.2.2.3 4LB 高级规则配置

##### 1. 会话保持

FLB 四层负载均衡支持开启会话保持, 开启后, 单位时间内 (默认 60s) 同一个源 IP 的连接将会固定转发至同一个后端服务器。

##### 2. 最大失败次数

最大失败次数表示在一个健康检查间隔内如果某个后端服务器异常次数累计达到当前设置的最大失败次数, 那么在后续的健康检查间隔内, 4LB 就认为这个节点宕机了, 即在这段时间内不会再将请求转发给它, 直到某次健康检查间隔内该节点的异常次数小于最大失败次数, 4LB 才会将请求继续转发给它。

##### 3. 最大连接数

最大连接数量表示该 4LB 端点支持建立的最大并发连接数。由于 FLB 具有集群管理, 分布式、可扩展的特点, 此处配置的最大连接数作用于此 4LB 规则所在的每个 4LB 节点上, 实际业务最大并发连接数为此处配置的最大连接数 X 4LB 节点数量。

##### 4. 读取超时时间

读取超时时间表示该 4LB 端点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, 4LB 会认为连接已经超时, 自动关闭连接。

##### 5. 写入超时时间

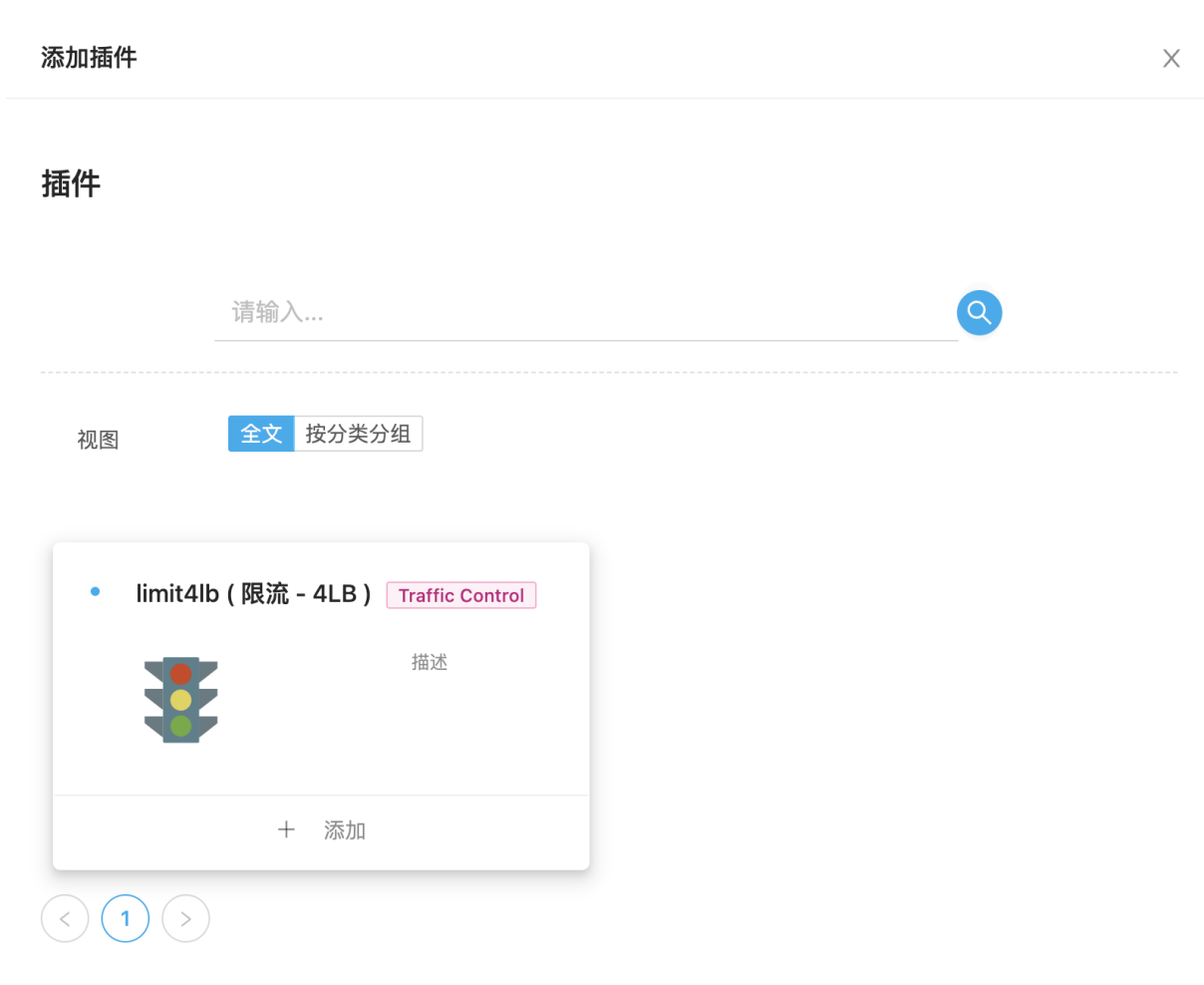
写入超时时间表示该 4LB 端点等待向客户端发送数据的最长时间。如果在规定时间内没有完成数据发送, 4LB 也会认为连接已经超时, 自动关闭连接。

##### 6. 闲置超时时间

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, 4LB 会主动断开连接, 以释放资源。



#### 4.2.2.2.4 4LB 策略配置



1. limit4lb (限流 - 4LB)

FLB 支持对 4LB 添加额外策略，目前支持 4LB 限流策略，实现针对四层流量进行限流

## 2. Limit size

最大请求流量，超过该峰值的部分的请求流量将被限制

## 3. Sync Rate

同步速率，

## 4. 插件开关

插件配置完成后，可选择开启或关闭。

### 4.2.2.2.5 4LB 健康检查配置

The screenshot displays the '健康检查' (Health Check) configuration page. It features a table with the following data:

时间	内容
2023-03-24T02:10:00.030Z	详情
2023-03-24T02:05:00.024Z	详情
2023-03-24T02:00:00.036Z	详情
2023-03-24T01:55:00.029Z	详情
2023-03-24T01:50:00.030Z	详情
2023-03-24T01:45:00.021Z	详情

Below the table, a code block shows the following JSON response:

```
1 {
2   "unhealthy": [
3     "192.168.12.4:49090"
4   ]
5 }
```

## 1. 健康检查

4LB 默认开启后端服务器的四层 TCP/UDP 连通性健康检查，若单位时间内，健康检查失败次数超过最大失败次数，4LB 将会判断该后端服务器状态异常，自动将该服务踢出可用后端服务器组。后续时间内健康检查判断该后端服务器恢复正常连通性后，会将该服务重新添加回后端服务器组。

## 2. 详情

点击“详情”按钮，可以查看异常 4LB 后端服务器地址。

### 4.2.2.3 4LB 详情页

Flomesh / FLB | 软负载 / 4LB 详情

lb01

保存

名称: lb01

端口: 80

组织: flomesh

唯一IP:

地址池: ap01

协议: TCP

集群: ft01

源地址池: 0 . 0 . 0 . 0 / 0

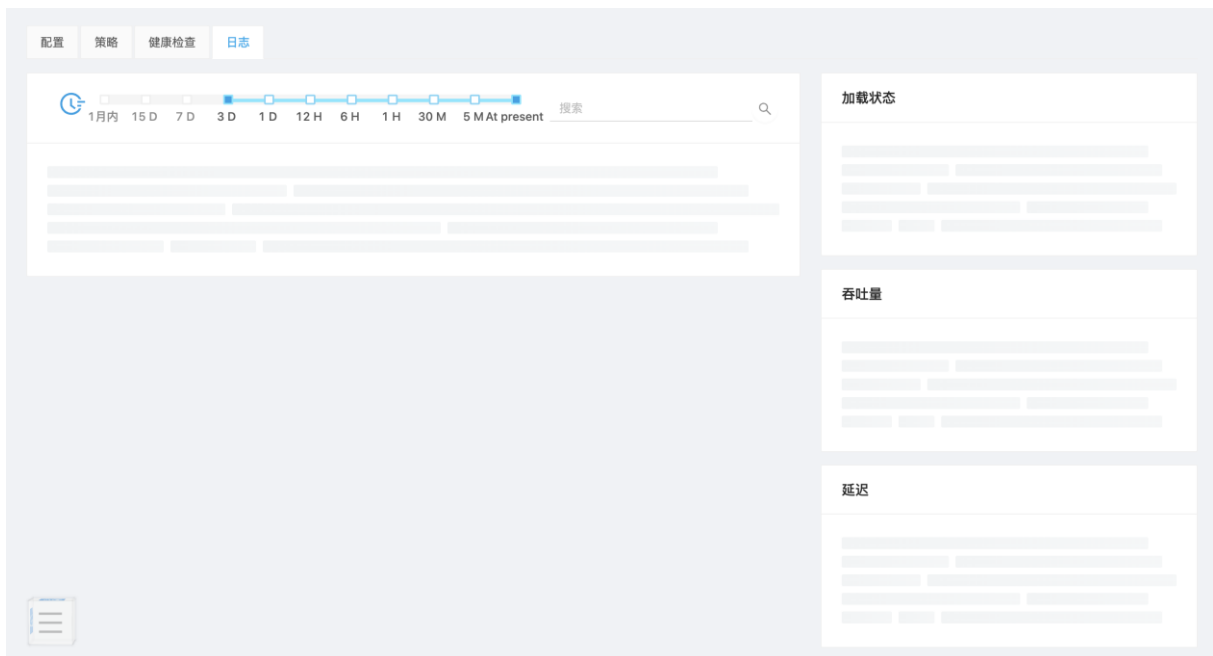
TLS可用:

IP: 100.100.100.1

全局ID: 050dead9-d1bb-444e-b4aa-e8afac7dfa42

在 4LB 详情页，可查看或编辑 4LB 规则，但是组织、地址池、集群、端口、唯一 IP 不可修改。

#### 4.2.2.3.1 4LB 详情页日志查看错误!未定义书签。这部分暂时没有信息



##### 1. 日志

展示 4LB 相关日志信息。

##### 2. 加载状态?? ! 错误!未定义书签。，4lb 这里应该没有加载状态的说法吧

##### 3. 吞吐量

##### 4. 延迟

## 4.3 4LB 看板

### 4.3.1 什么是 4LB 看版

4LB 看版收集了 4LB 集群的运行状态，展示 4LB 各类监控指标，用户可自定义看版界面。

## 4.3.2 功能说明



### 1. 自定义看板界面



点击“”按钮可自定义看板界面。

### 2. 新的仪表盘

点击“新的仪表盘”可创建新的看板界面

### 3. 添加 UI 组件

点击“添加 UI 组件”可创建新的 UI 面板

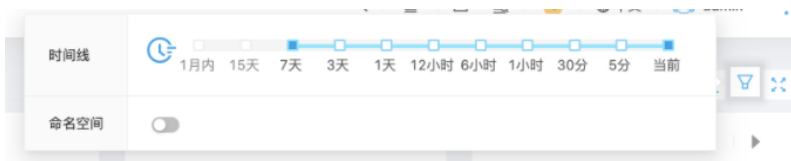
### 4. 删除 UI 组件

点击 UI 面板右上方的“”按钮可移除该 UI 面板

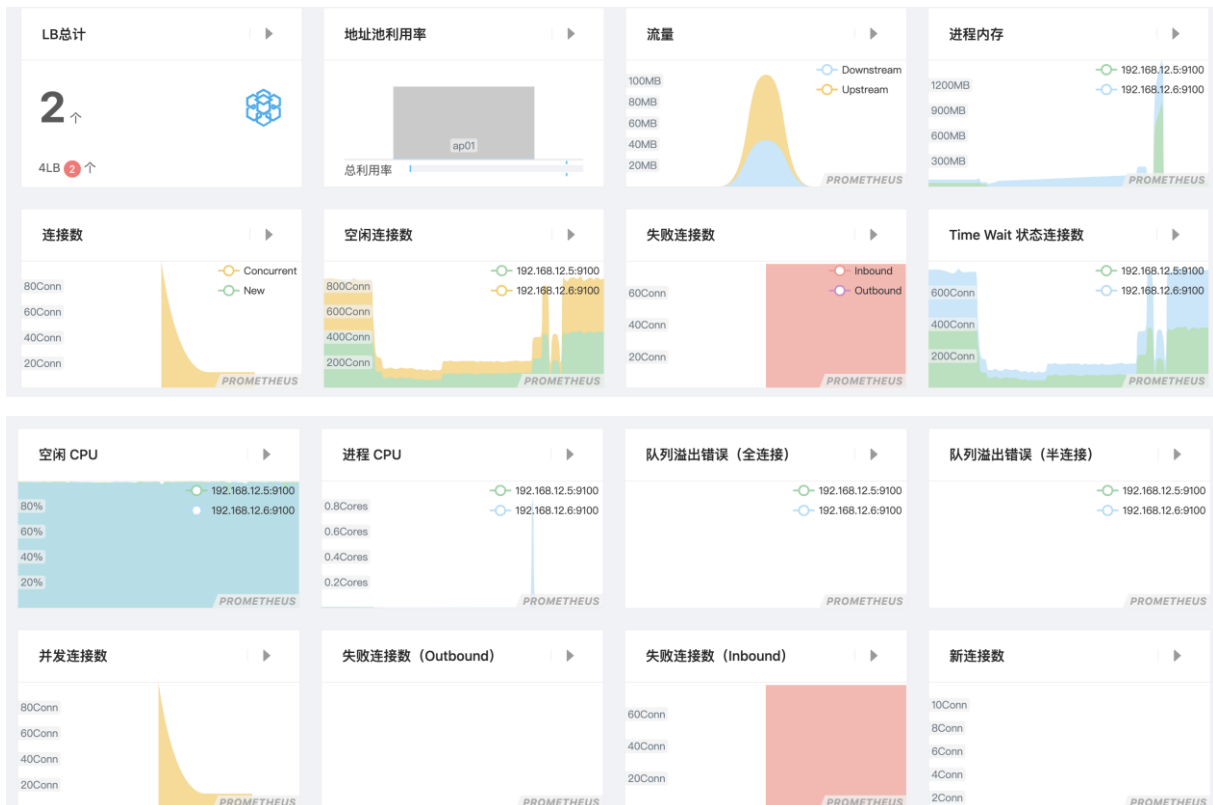
### 5. 时间线



点击“”按钮，可筛选各项指标显示时间线。



### 4.3.3 4LB 指标项说明



#### 1. LB 总计

4LB 规则数量总计

#### 2. 地址池利用率

4LB 地址池利用率

#### 3. 流量

4LB 进出流量总计，Downstream 表示从客户端发出的请求流量，Upstream 表示从服务端发出的请求流量

#### 4. 进程内存

表示 4LB 节点的 Pipy 进程消耗内存大小

#### 5. 连接数

连接数表示 4LB 节点当前建立连接数情况，current 连接数表示当前连接数总和，new 连接数表示当前新建连接数

#### 6. 空闲连接数

空闲连接数是指当前没有被使用的连接数。在负载均衡过程中，服务器会维护一个连接池，用于存储已经建立的连接。当客户端请求到来时，4LB 会从连接池中选择一个空闲连接，将请求转发给服务器。

#### 7. 失败连接数

失败连接数是指由于某种原因，无法建立连接的数量。这些原因可能包括服务器故障、网络中断、连接超时等。当 4LB 无法将请求转发给任何一个服务器时，就会增加失败连接数。

失败连接数 Outbound，表示从 4LB 节点到后端服务器的失败连接数。

失败连接数 Inbound，表示从客户端到 4LB 节点的失败连接数。

#### 8. Time Wait 状态连接数

TimeWait 连接数是指处于 TimeWait 状态的连接数量。在 TCP 协议中，当一条连接被关闭时，会进入 TimeWait 状态。在这个状态下，连接仍然存在，但是不能被使用。这是为了保证所有的数据都能够被传输完毕，并且避免重复的连接。因此，TimeWait 连接数是一个重要的指标，可以反映出 4LB 的性能和 TCP 连接管理的效率。

#### 9. 空闲 CPU

空闲 CPU 是指 4LB 节点空闲 CPU 比例，空闲 CPU 越高，意味着 Pipy 进程开销越小，软负载处于低负载运行状态。

#### 10. 进程 CPU

进程 CPU 是指 4LB 节点 Pipy 进程占用 CPU 比例，进程 CPU 越高，意味着 Pipy 进程开销越大，软负载处于高负载运行状态。

#### 11. 队列溢出错误

队列溢出错误是指当队列中的连接数量超过了系统预设的最大值时，新的连接请求无法被处理而导致的错误。在负载均衡系统中，队列溢出错误通常分为两种类型：全连接队列溢出错误和半连接队列溢出错误。

全连接队列溢出错误通常发生在传输层（TCP）中，当负载均衡器将请求转发给后端服务器时，如果此时服务器的连接数已经达到了最大值，那么负载均衡器就会将请求放入全连接队列中，等待服务器处理。如果全连接队列也已经满了，新的连接请求就无法被处理，从而导致全连接队列溢出错误。

半连接队列溢出错误通常发生在网络层（IP）中，当负载均衡器将请求转发给后端服务器时，如果此时服务器的 SYN 接收队列已经达到了最大值，那么负载均衡器就会将请求放入半连接队列中，等待服务器回复 SYN-ACK 报文。如果半连接队列也已经满了，新的连接请求就无法被处理，从而导致半连接队列溢出错误。

## 5. 七层负载均衡操作指导

FLB 提供 API（Application Programming Interface）级别的七层负载均衡能力。在“Open API | 开放接口”功能菜单下，支持配置 API、策略、服务提供者、服务消费者，提供看版和拓扑展示功能。





## 5.1 服务提供者配置

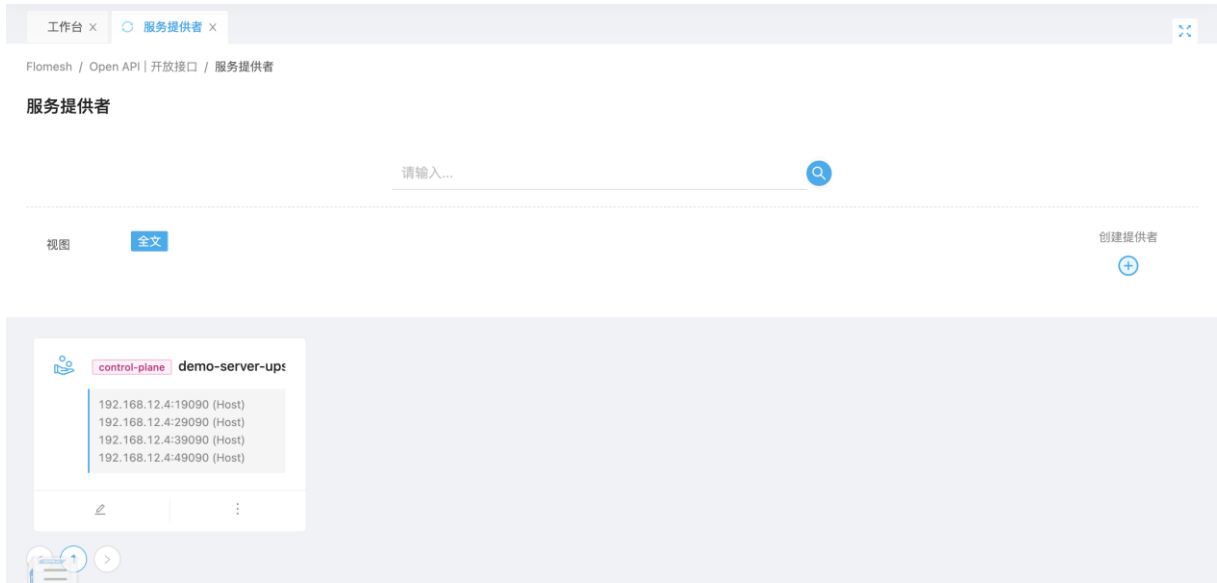
### 5.1.1 什么是服务提供者

服务提供者指的是一个提供服务的应用程序组。

在微服务架构中，服务提供者将自己的服务注册到服务注册中心中，以便其他应用程序可以发现和使用它。

在 FLB 中，支持手动输入服务提供者信息，也支持从注册中心同步服务提供者信息。

## 5.1.2 功能说明




### 1. 服务提供者列表

在 FLB 界面可展示所有服务提供者信息，包含服务提供者名称、提供者所在注册中心、提供者地址信息

### 2. 创建服务提供者

点击  创建服务提供者

### 3. 修改服务提供者配置

点击  修改服务提供者配置

### 4. 删除服务提供者



点击 ，选择  删除，完成删除服务提供者操作

名称	<input type="text" value="demo-server-upstream"/>	显示名	<input type="text" value="demo-server-upstream"/>																												
集群	<input type="text" value="control-plane (l..."/> ▾	健康检查路径	<input type="text" value="/"/>																												
算法	<input type="text" value="RoundRobinLoadBalancer"/> ▾	Header关键字	<input type="text" value="x-forwarded-for"/>																												
会话保持	<input checked="" type="checkbox"/>	过期时间 (秒)	<input type="text" value="60"/>																												
目标	<div><span>主机</span> <span>服务</span></div> <table><tbody><tr><td>主机</td><td><input type="text" value="192.168.12.4"/></td><td>端口</td><td><input type="text" value="19090"/></td><td>权重</td><td><input type="text" value="100"/></td><td><input type="button" value="⊖"/></td></tr><tr><td>主机</td><td><input type="text" value="192.168.12.4"/></td><td>端口</td><td><input type="text" value="29090"/></td><td>权重</td><td><input type="text" value="100"/></td><td><input type="button" value="⊖"/></td></tr><tr><td>主机</td><td><input type="text" value="192.168.12.4"/></td><td>端口</td><td><input type="text" value="39090"/></td><td>权重</td><td><input type="text" value="100"/></td><td><input type="button" value="⊖"/></td></tr><tr><td>主机</td><td><input type="text" value="192.168.12.4"/></td><td>端口</td><td><input type="text" value="49090"/></td><td>权重</td><td><input type="text" value="100"/></td><td><input type="button" value="⊖"/> <input type="button" value="⊕"/></td></tr></tbody></table>			主机	<input type="text" value="192.168.12.4"/>	端口	<input type="text" value="19090"/>	权重	<input type="text" value="100"/>	<input type="button" value="⊖"/>	主机	<input type="text" value="192.168.12.4"/>	端口	<input type="text" value="29090"/>	权重	<input type="text" value="100"/>	<input type="button" value="⊖"/>	主机	<input type="text" value="192.168.12.4"/>	端口	<input type="text" value="39090"/>	权重	<input type="text" value="100"/>	<input type="button" value="⊖"/>	主机	<input type="text" value="192.168.12.4"/>	端口	<input type="text" value="49090"/>	权重	<input type="text" value="100"/>	<input type="button" value="⊖"/> <input type="button" value="⊕"/>
主机	<input type="text" value="192.168.12.4"/>	端口	<input type="text" value="19090"/>	权重	<input type="text" value="100"/>	<input type="button" value="⊖"/>																									
主机	<input type="text" value="192.168.12.4"/>	端口	<input type="text" value="29090"/>	权重	<input type="text" value="100"/>	<input type="button" value="⊖"/>																									
主机	<input type="text" value="192.168.12.4"/>	端口	<input type="text" value="39090"/>	权重	<input type="text" value="100"/>	<input type="button" value="⊖"/>																									
主机	<input type="text" value="192.168.12.4"/>	端口	<input type="text" value="49090"/>	权重	<input type="text" value="100"/>	<input type="button" value="⊖"/> <input type="button" value="⊕"/>																									
描述	<input type="text" value="未设置"/>																														

保存

## 1. 名称

服务提供者名称信息

## 2. 显示名

服务提供者显示名称

## 3. 集群

服务提供者所关联的注册中心名称

## 4. 健康检查路径

服务提供者提供的健康检查路径，FLB 的拨测点将对该路径持续发起拨测，并根据状态返回码判断该服务提供者健康状态。

## 5. 服务提供者地址信息

服务提供者 IP+Port 信息

## 6. 算法

FLB 支持以下几种算法：

- 轮询算法 (RoundRobinLoadBalancer, 支持按权重轮询)
- 最小连接数算法 (LeastWorkLoadBalancer)
- 哈希算法 (HashingLoadBalancer)

## 7. Header 关键字

用作会话保持的请求特征判断

## 8. 会话保持

会话保持可配置开启或关闭，开启后，过期时间内同一请求特征的请求会被转发至同一个后端服务。

a. 过期时间（秒）

会话保持过期时间

## 9. 目标

服务提供者的后端服务目标地址，即真实的上游服务地址。

a. 主机

服务提供者目标类型之一：上游服务的 IP+Port

主机	端口	权重	
192.168.12.4	19090	100	-
192.168.12.4	29090	100	-
192.168.12.4	39090	100	-
192.168.12.4	49090	100	- +

b. 服务

服务提供者目标类型之一：注册中心发现的服务

名称	集群
<input type="checkbox"/> traffic-guru	control-plane
<input type="checkbox"/> kube-dns	control-plane

合计 13 < 1 2 >

## 5.2 服务消费者配置（选配）

### 5.2.1 什么是服务消费者

服务消费者指的是一个使用服务的应用程序组。

在微服务架构中，服务消费者从服务注册中心中查找需要的服务，并调用服务提供者提供的服务。

## 5.2.2 功能说明

## 5.3 API 配置

### 5.3.1 什么是 API

API 是应用程序编程接口的缩写，是一组定义服务提供者和服务消费者之间通信规则的接口。API 定义了服务提供者提供的服务的功能、参数、返回值等信息，让服务消费者可以调用这些服务。

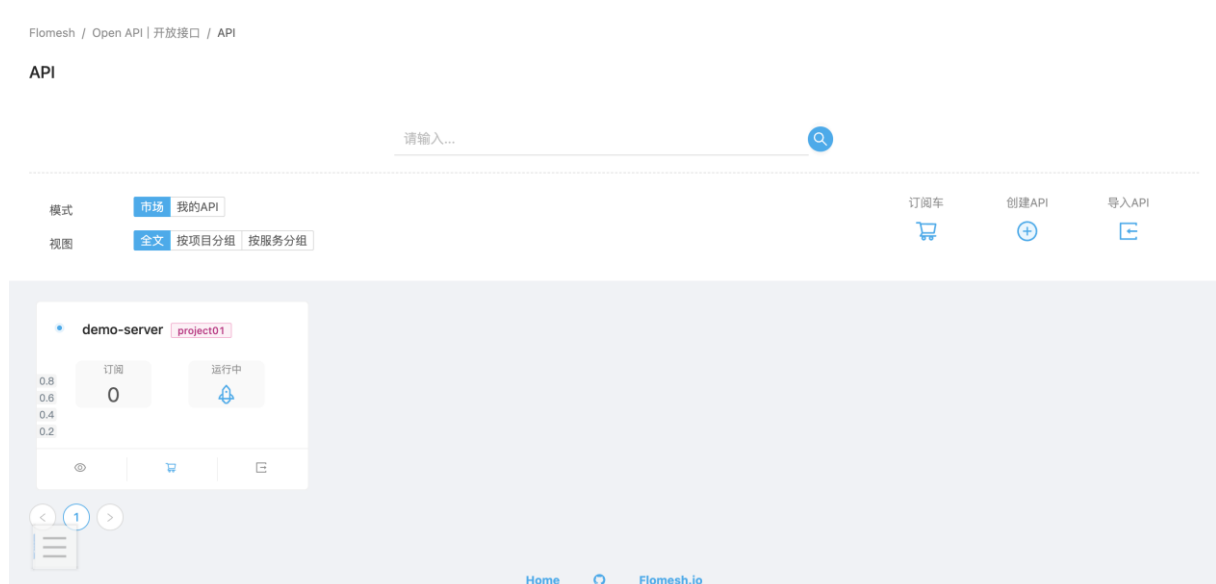
在微服务架构中，API 是服务提供者和服务消费者之间的桥梁，它定义了服务提供者和服务消费者之间的通信协议，确保它们可以彼此交互和通信。API 也可以帮助开发人员更好地管理和维护微服务架构中的服务。

在 FLB 中，API 可作为反向代理、正向代理、7 层负载均衡和 API 管理的统一平台。通过对 API 规则的配置，用户可以实现上述不同应用场景策略的统一配置与管理。

### 5.3.2 功能说明

#### 5.3.2.1 API 市场模式与订阅车管理

API 市场指的是 FLB 平台上创建的 API 列表集合，用户可在 API 市场订阅 API，订阅后的 API 会在订阅车内，用于绑定服务消费者。



### 5.3.2.2 我的 API 模式



#### 1. 模式

API 模式，指的是 API 管理模式，可选择市场和我的 API。进行 API 配置管理时需要切换模式至我的 API。

- a. 市场：该模式下只能查看并订阅 API
- b. 我的 API：该模式下可查看、订阅、编辑和删除 API

#### 2. 阶段

FLB 支持 API 全声明周期管理，包含设计、审查、运行和停止 4 个阶段，FLB 支持按阶段展示 API 列表。

#### 3. 视图

FLB 支持按项目或服务分组，方便统一管理各 API。

- a. 按项目分组

视图

全文 按项目分组 按服务分组



模式

市场 我的API

阶段

全文 设计 审查 运行 停止

视图

全文 按项目分组 按服务分组 project01 x



### b. 按服务分组

模式

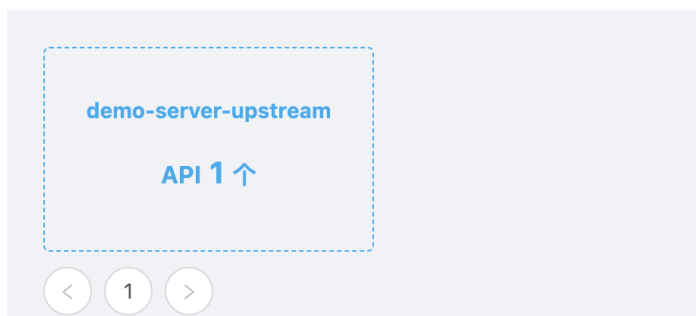
市场 我的API

阶段

全文 设计 审查 运行 停止

视图

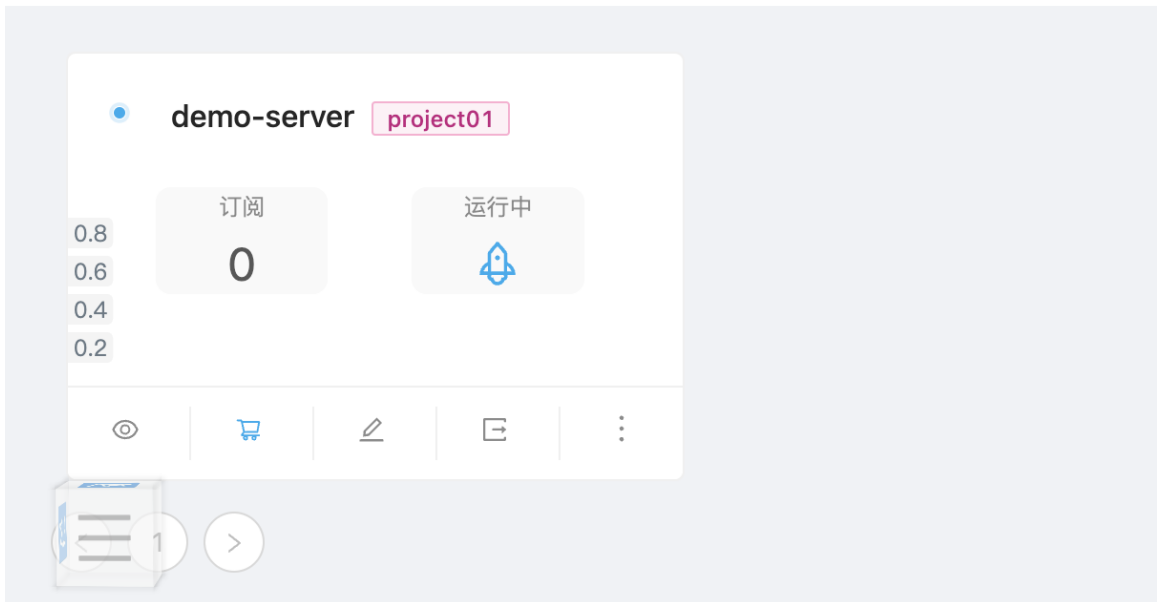
全文 按项目分组 按服务分组



模式 市场 我的API

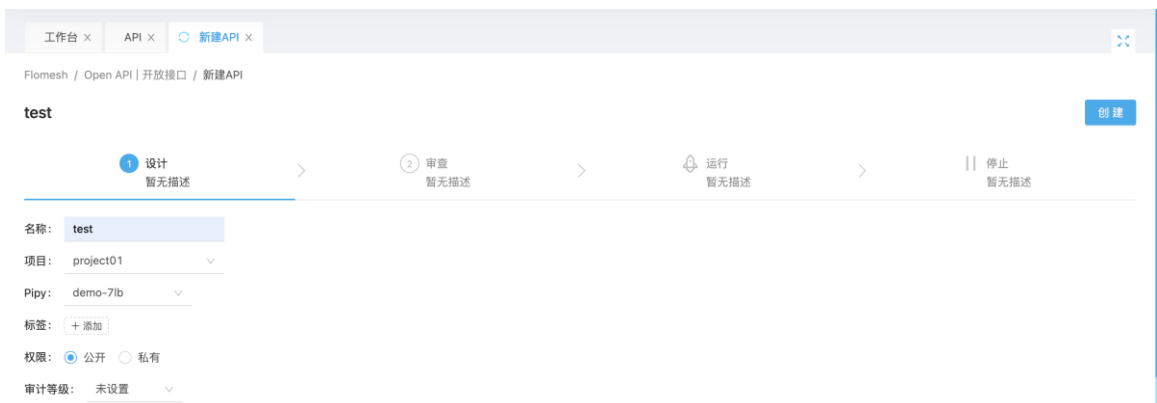
阶段 全文 设计 审查 运行 停止

视图 全文 按项目分组 按服务分组 demo-server-upstream x



### 5.3.2.3 API 全生命周期管理

#### 5.3.2.3.1 创建 API



#### 1. 名称

API 显示名称，创建 API 时，会校验 API 名称，确保系统中不会出现两个同样的 API 名称

#### 2. 项目

API 所属项目，选择对应项目后，仅该系统管理员和项目成员能够配置该 API

#### 3. Pipy



API 所在 Pipy 集群，选择集群后，该 API 将在对应 Pipy 集群的节点上生效

#### 4. 标签

API 标签，便于管理 API

#### 5. 权限

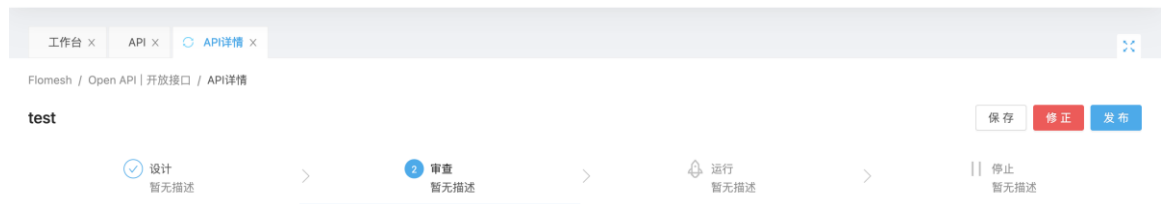
公开表示该 API 对所有人可见，私有表示仅本项目成员可订阅该 API。

#### 6. 审计等级

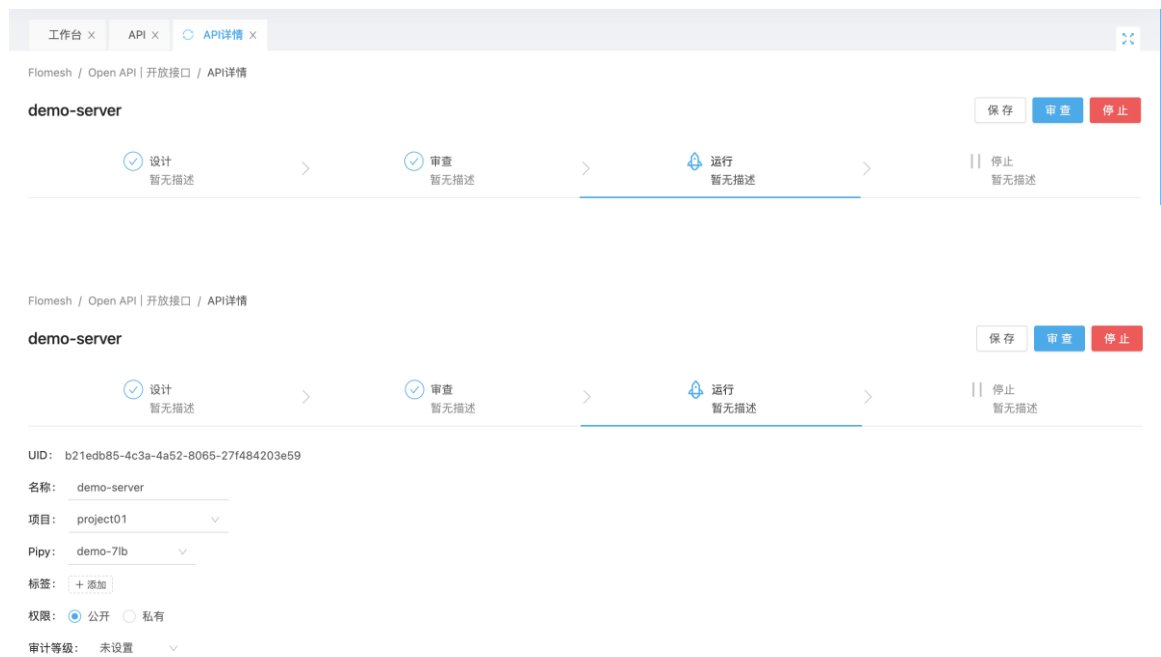
### 5.3.2.3.2 发起 API 审查



### 5.3.2.3.3 API 修正与发布



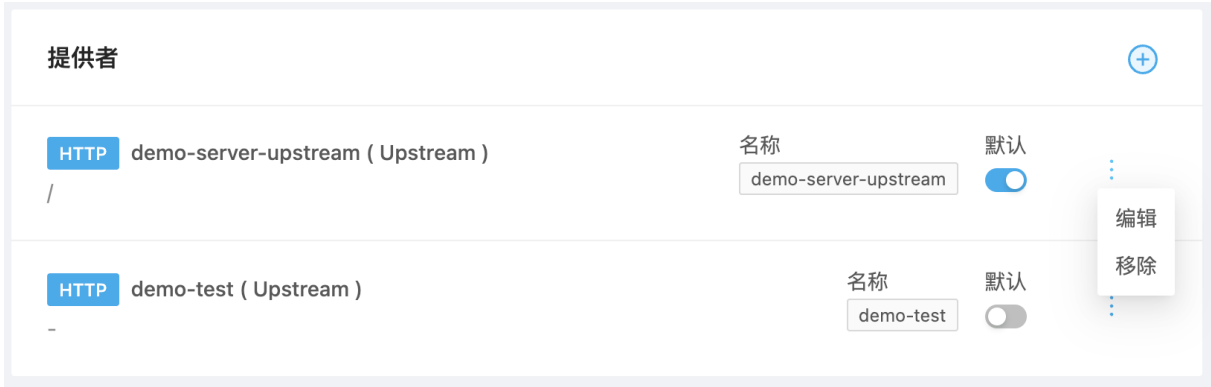
### 5.3.2.3.4 API 停止



## 5.3.2.4 API 规则配置

### 5.3.2.4.1 提供者配置

提供者指的是为该 API 提供服务的上游服务器组。



1. 点击 可添加新的提供者
2. 点击“编辑”可编辑提供者
3. 点击“移除”可移除提供者
4. 默认选项，当有多个提供者时，如没有配置定制路径等功能，默认选择该提供者作为服务提供者

提供者

×

名称	demo-server-upstream		
类别	Upstream	协议	HTTP x
服务提供者	demo-server-upstream x	选择	路径 /
版本	+ 添加		

取消

保存

1. 名称

提供者名称

2. 类别与服务提供者

支持 Host 和 Upstream 两种类别的提供者，一个 API 内最多支持一个 Host 类别的提供者

- a. Host 类别

直接键入 Host 地址信息，支持添加多个地址信息

## 提供者

名称	demo-test
类别	Host <span>▼</span>
主机	<input type="text" value="192.168.12.50:8081"/> <input type="text" value="192.168.12.51:8081"/> <input type="text" value="未设置"/>
版本	<input type="button" value="+ 添加"/>

### b. Upstream 类别

选择创建的服务提供者

#### 提供者

名称	demo-test
类别	Upstream
服务提供者	<input type="text" value="demo-test"/> <input type="button" value="选择"/>
版本	<input type="button" value="+ 添加"/>

搜索

### 3. 协议

Pipy 代理节点和提供者之间的通信协议，支持多选。

### 4. 路径

服务提供者的访问路径。

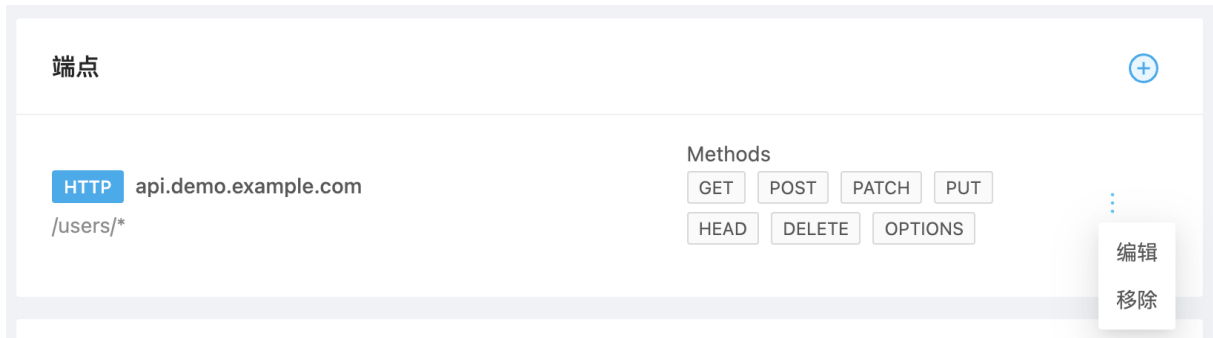
### 5. 版本


服务提供者版本信息。

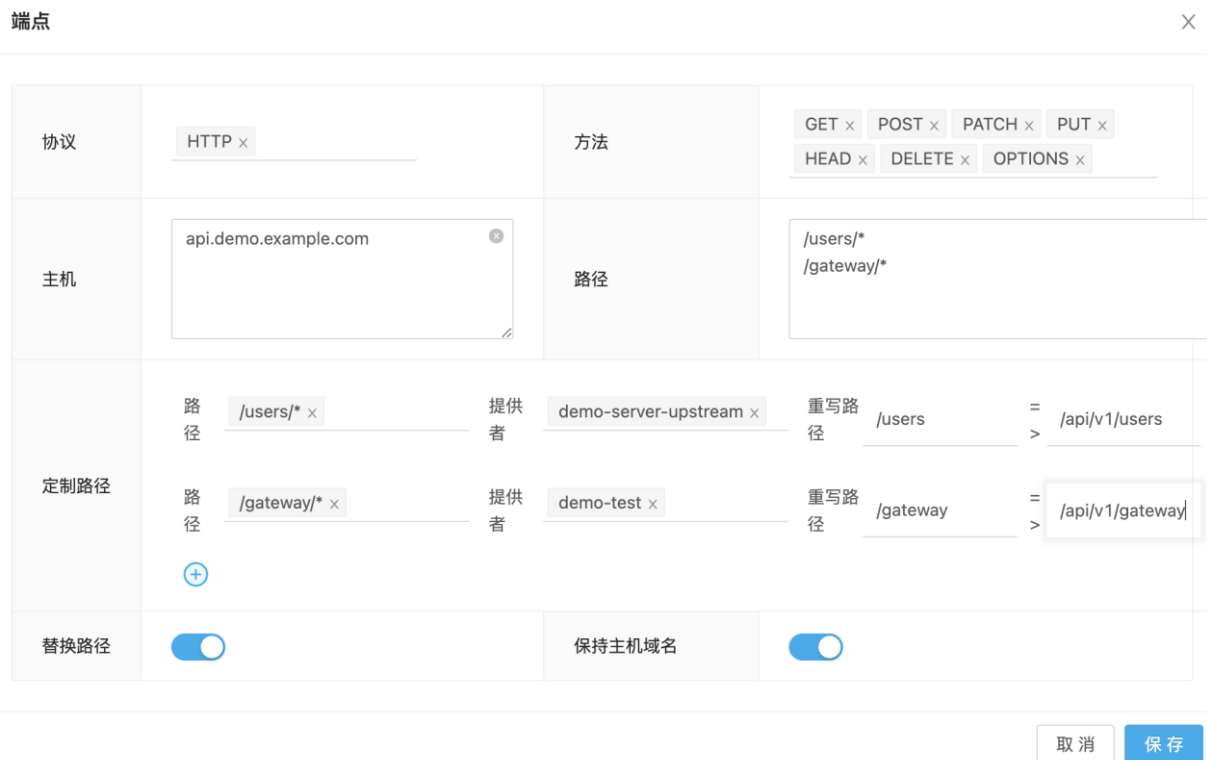
版本	<input type="text" value="version:v2"/> <input type="button" value="+ 添加"/>
----	---

### 5.3.2.4.2 端点配置

端点指的是该 API 对外暴露的接口地址。



1. 点击 , 可添加新的端点
2. 点击“编辑”，可编辑端点
3. 点击“移除”，可移除端点



1. 协议，端点对外暴露的通信协议，支持多选
2. 方法，端点对外暴露的请求方法，支持多选
3. 主机，端点对外暴露的主机信息，支持配置 IP 或域名，支持多行配置

主机	<div style="border: 1px solid #ccc; padding: 5px;">             api.demo.example.com              api.example.com           </div>
----	--

如上述配置，端点将接收 Host 为 api.demo.example.com 和 api.example.com 的请求

#### 4. 路径，端点对外暴露的路径信息，支持多行配置

路径	<div style="border: 1px solid #ccc; padding: 5px;">             /users/              /gateway/*              /app/{path}/demo1/*           </div>
----	---

如上配置，路径支持以下几种写法：

- a. 全匹配：/users/
- b. 正则表达式匹配：/gateway/\*, /app/{path}/demo1/\*

#### 5. 定制路径

定制路径也称路径重写是指将客户端请求的 URL 路径重写为另一个路径。这通常是由于服务器上的应用程序需要不同的 URL 路径来处理不同的请求。

✕

协议	HTTP ✕		方法	GET ✕ POST ✕ PATCH ✕ PUT ✕ HEAD ✕ DELETE ✕ OPTIONS ✕		
主机	<div style="border: 1px solid #ccc; padding: 2px;">             api.demo.example.com              api.example.com           </div>		路径	<div style="border: 1px solid #ccc; padding: 2px;">             /users/              /gateway*              /app/{path}/demo1*           </div>		
定制路径	路径	/users/* ✕	提供者	demo-server-upstream ✕	重写路径	/users = > /api/v1/users
	路径	/gateway/* ✕	提供者	demo-test ✕	重写路径	/gateway = > /api/v1/gateway

+
-

- a. 点击 + 可添加定制路径规则
- b. 点击 - 可删除定制路径规则
- c. 路径：此处输入需要改写的客户端请求路径
- d. 提供者：此处选择该路径对应的提供者
- e. 重写路径：将客户端请求路径改写为提供者能够处理的路径，支持正则表达式

#### 6. 替换路径

替换路径指的是将客户端请求的 URL 路径替换为提供者配置中的路径







## 7. 保持主机域名

保持主机域名指的是保持客户端请求携带的 Host 信息，若不开启，则该 Host 信息将会被替换为提供者的实际 Host 地址。

# 5.4 策略管理

## 5.4.1 什么是策略

策略是指在基础 API 规则外，能够添加的额外策略，以插件形式提供，以满足不同用户的各类复杂需求。FLB 目前支持如下策略 Basic 认证、Jwt 认证、国密 SM2、Hmac 认证、跨域、黑白名单、限速限流、script（请求改写）、灰度发布、缓存（cache）。

<ul style="list-style-type: none"><li><b>basic ( Basic认证 )</b> <span>Authentication</span></li></ul>  <p>描述</p> <p>+ 添加</p>	<ul style="list-style-type: none"><li><b>jwt ( Jwt认证 )</b> <span>Authentication</span></li></ul>  <p>描述</p> <p>+ 添加</p>
<ul style="list-style-type: none"><li><b>sm2 ( 国密SM2 )</b> <span>Authentication</span></li></ul>  <p>描述</p> <p>+ 添加</p>	<ul style="list-style-type: none"><li><b>hmac ( Hmac(认证) )</b> <span>Authentication</span></li></ul>  <p>描述</p> <p>+ 添加</p>
<ul style="list-style-type: none"><li><b>cors ( 跨域 )</b> <span>Security</span></li></ul>  <p>描述</p> <p>+ 添加</p>	<ul style="list-style-type: none"><li><b>ban ( 黑白名单 )</b> <span>Security</span></li></ul>  <p>描述</p> <p>+ 添加</p>

- **limit (限速限流)** Traffic Control


描述

+ 添加

- **limit4lb (限流 - 4LB)** Traffic Control


描述

+ 添加

- **script** Transformations


描述

+ 添加

- **canary (灰度发布)** Transformations


描述

+ 添加

- **cache (缓存)** Traffic Control


描述

+ 添加

## 5.4.2 功能说明

### 5.4.2.1 Basic 认证

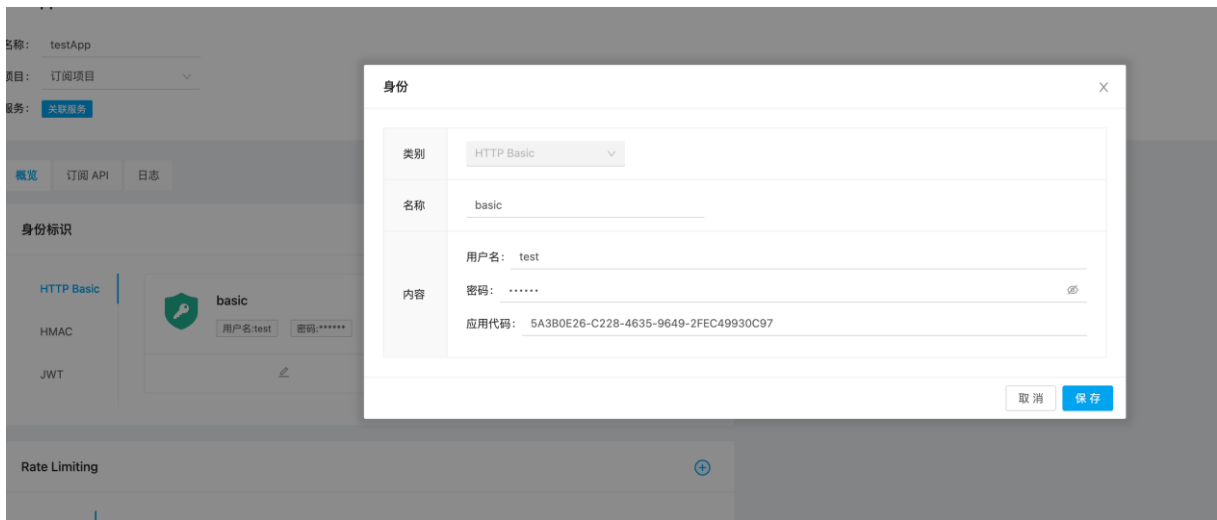
Basic 认证是一种 HTTP 认证方式，基于用户名和密码进行验证，将用户名和密码进行 base64 编码后发送给服务器进行验证。



用户名：访问此 API 的用户名

密码：访问此 API 的密码

如果填写了用户名和密码，则对所有此接口的访问者生效，如果需要针对不同访问者进行 basic 认证，则需要新建**消费者**，如下图：



名称：密钥名称

用户名：消费者使用的用户名

密码：消费者使用的密码

应用代码：标识消费者的应用代码

#### 5.4.2.2 Jwt 认证

Jwt 认证是一种基于 JSON Web Token 的认证方式，通过在请求头中加入 JWT Token 进行身份验证。



Header 名称：从 header 从取得 jwt token 的参数名

验证字段：验证 jwt 中的字段值，如 exp 为验证过期时间，nbf 为验证 token 的起始时间



类别	JWT <input type="button" value="v"/>
名称	test
内容	算法: HS256 <input type="button" value="v"/>
	Header: 未设置
	参数: 未设置
	密钥: 123456
	关键字段: 1234567890

取消

保存

算法：jwt 算法

Header: jwt header

参数: jwt payload

密钥：jwt 密钥

关键字段：标识用户身份的 jwt 关键字段

### 5.4.2.3 国密 SM2

国密 SM2 是中国自主研发的一种非对称加密算法，用于保证数据传输的安全性。

sm2 ( 国密SM2 ) ✓  X

[配置](#) [脚本](#)

---

启用:

#### 5.4.2.4 Hmac 认证

Hmac 认证是一种基于散列消息认证码的认证方式，通过对请求数据进行哈希计算并加上密钥进行身份验证。

hmac ( Hmac(认证) ) ✓  X

[配置](#) [脚本](#)

---

启用:       算法: \_\_\_\_\_      headers: 未设置

#### 5.4.2.5 Cors 跨域

跨域指的是在浏览器中，一个网页的脚本试图访问另一个网页的数据，跨越了域名的限制。在 FLB 中，允许配置在不同域名中透传请求头信息。

cors ( 跨域 )  X

[配置](#) [脚本](#)

---

Origins: \*      方法: GET,HEAD,PUT,PATCH,POST,DELETE,OPTIONS      请求头: accept,x-requested-with,Content-Type,X-Custom-Header

Credentials:

Origins: 设置 Access-Control-Allow-Origin

方法: 设置 Access-Control-Allow-Methods

请求头: 设置 Access-Control-Allow-Headers

Credentials: 设置 Access-Control-Allow-Credentials

#### 5.4.2.6 Ban 黑白名单

黑白名单是一种访问控制机制，黑名单中的服务或源 IP 被禁止访问，而白名单中服务或源 IP 被允许访问。



Allow：白名单 IP 列表，一行一个

Deny：黑名单 IP 列表，一行一个

状态：状态返回码

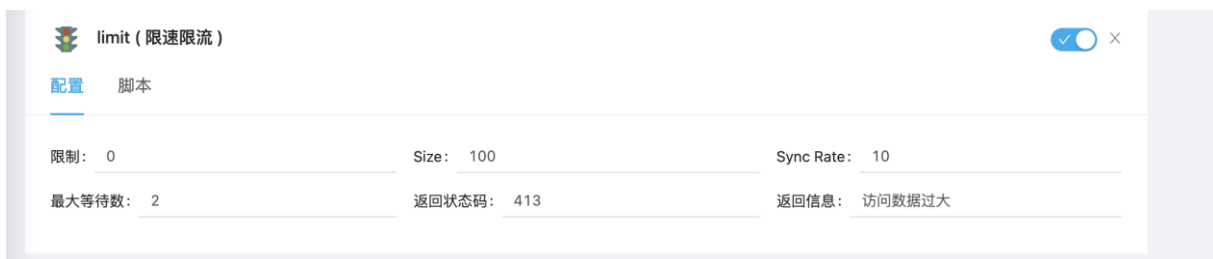
信息：返回信息

允许请求头：当开启请求头验证的时候，这里配置的请求头如果匹配，则请求被拦截

请求头验证：配置是否进行请求头校验拦截

#### 5.4.2.7 Limit 限流限速

限速限流是一种流量控制机制，通过限制请求的并发数量或请求大小来保护服务在极端情况下可用。



限制：设置单位时间请求限制次数

Size：设置单位时间请求大小限制

Sync Rate：设置限制时间的间隔

最大等待数：同时允许等待的请求数量

返回状态：返回状态码

返回信息：返回信息体

#### 5.4.2.8 Script 请求改写

通过自定义 Pipy 配置脚本，实现请求头或者请求 Body 内容改写。

```
1 (msg, _inbound) => (  
2   void(  
3     msg.head.headers['X-Real-IP'] = _inbound.remoteAddress,  
4     msg.head.headers['REMOTE_HOST'] = _inbound.remoteAddress,  
5     msg.head.headers['X-Forwarded-For'] = msg.head.headers['x-forwarded-for'] ?  
6       msg.head.headers['x-forwarded-for'] + "," + _inbound.localAddress :  
7       _inbound.localAddress ,  
8     msg.head.headers['X-Forwarded-Proto'] = msg.head.protocol  
9   )  
10 )
```

### 5.4.2.9 Canary 灰度发布

灰度发布是根据请求特征，将请求分发到不同的服务提供者版本上，以实现灰度发布或蓝绿测试。

canary (灰度发布)

配置 脚本

系统代码:  请求头标识:

### 5.4.2.10 Cache 缓存

针对静态的内容或者变更频率低的响应，可以通过在代理层设置缓存响应的方式，在不影响体验的情况下降低对上游和网络的压力。

cache (缓存)

配置 脚本

超时时间:  文件后缀名:

超时时间：缓存的超时时间

文件后缀名：缓存数据的后缀，例如 html,json

## 5.5 API 看版

### 5.5.1 什么是 API 看版

API 看版收集了 API 集群的运行状态，展示 API 各类监控指标，用户可自定义看版界面。

### 5.5.2 功能说明

## 5.6 API 拓扑

### 5.6.1 什么是 API 拓扑

### 5.6.2 功能说明



#### 1. 自定义看版界面



点击“”按钮可自定义看版界面。

#### 2. 新的仪表盘

点击“新的仪表盘”可创建新的看版界面

#### 3. 添加 UI 组件


点击“添加 UI 组件”可创建新的 UI 面板

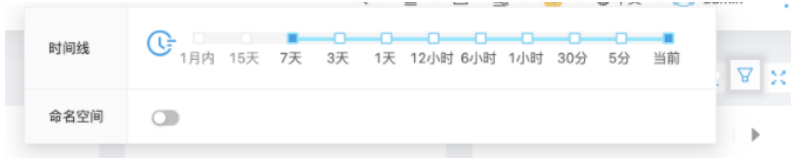
#### 4. 删除 UI 组件

点击 UI 面板右上方的“”按钮可移除该 UI 面板

## 5. 时间线



点击“”按钮，可筛选各项指标显示时间线。



### 5.6.3 API 指标项说明

#### 1. API 状态

API 状态展示 API 健康状态

#### 2. 延迟

延迟表示请求从客户端经过 API 代理到后端服务的响应时间总和

#### 3. 用户代理

用户代理表示请求头中 user agent 信息

#### 4. QPS

QPS 是指每秒查询率 (Query Per Second)，是指 API 在一秒钟内所处理的查询数量

#### 5. 错误率

表示范围码为 4xx, 5xx 的请求比例

#### 6. API 指标

展示最近时间内 API 调用失败和成功次数

