

DaoCloud Enterprise

DaoCloud Enterprise Product Documentation

Table of contents

1. Best practice	3
1.1 示例应用体验微服务治理	3
2. Intro	13
2.1 产品优势	13
2.2 环境要求	14
2.3 微服务引擎 Release Notes	15
2.4 适用场景	19
2.5 功能总览	20
3. Ms gateway	21
3.1 监控告警	21
3.2 Api	25
3.3 Domain	45
3.4 Gateway	58
3.5 Logs	67
3.6 Service	70
4. Quickstart	84
4.1 安装微服务引擎	84
4.2 离线升级微服务引擎模块	89
4.3 选择工作空间	92
5. Registry	94
5.1 Integrated	94
5.2 Managed	108

1. Best practice

1.1 示例应用体验微服务治理

微服务引擎属于 DCE 5.0 高级版功能，其中包含注册中心、配置中心、微服务治理（传统微服务、云原生微服务）、云原生网关等功能。本文将通过示例应用带您体验其中的微服务治理功能。

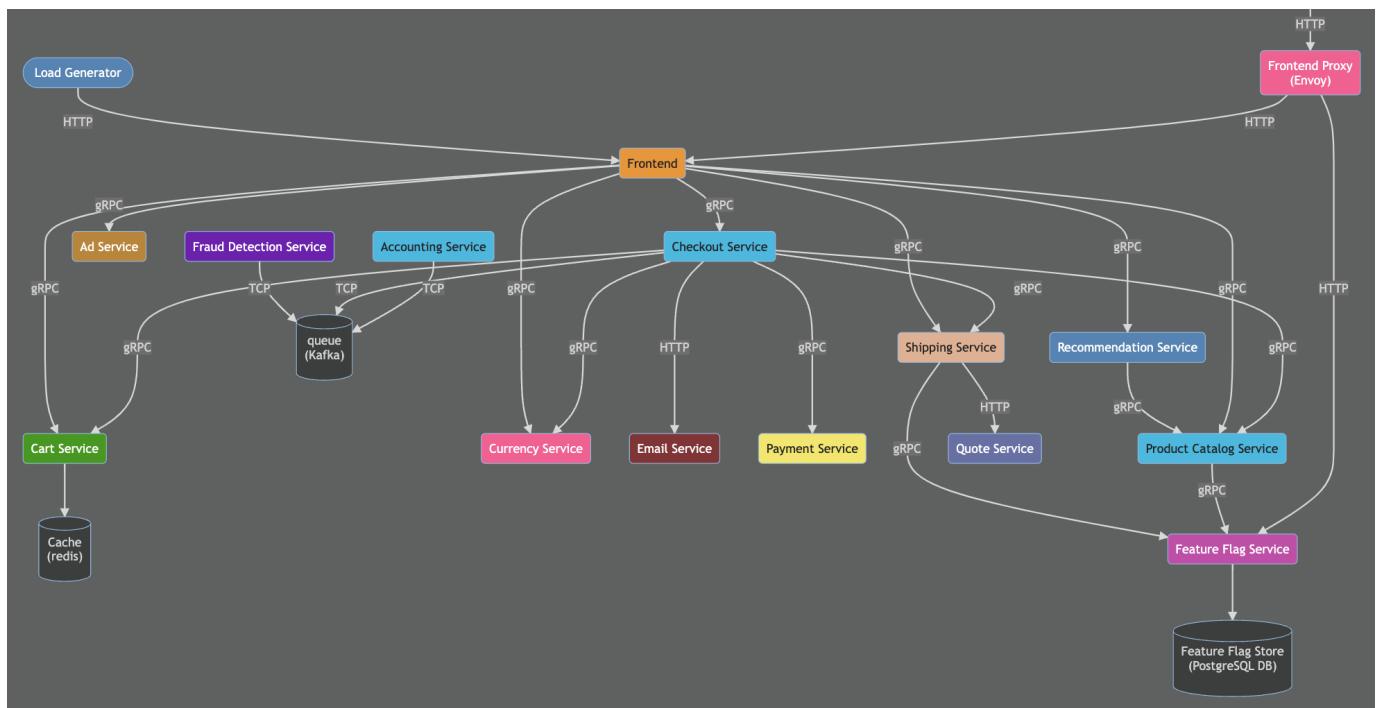
此次最佳实践包含的全部流程如下：

1. 在应用工作台部署示例应用，并启用微服务治理功能
2. 在微服务引擎中启用传统微服务治理插件
3. 在微服务引擎中配置对应的治理规则
4. 在微服务引擎暴露 API 并访问应用

示例应用介绍

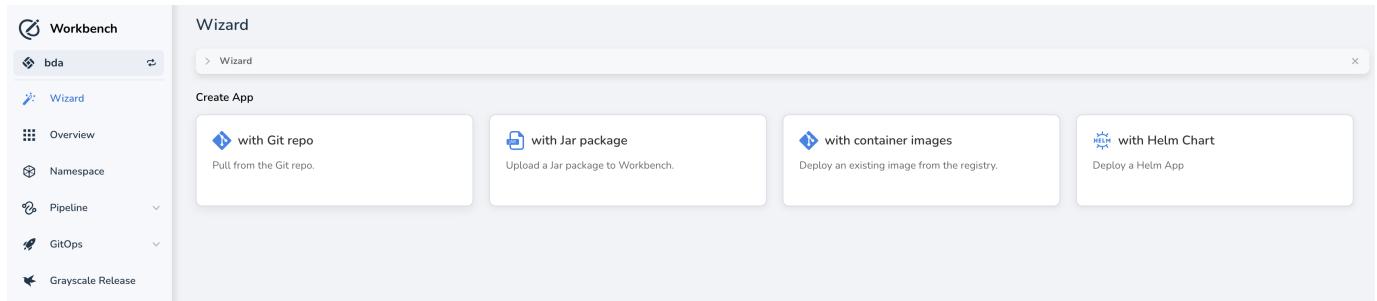
本次实践使用的示例应用基于 OpenTelemetry 的标准演示应用，由 DaoCloud 大微服务团队根据 DCE 5.0 的功能加以优化，以便更好地体现云原生以及可观测能力，呈现微服务治理效果。示例应用已经在 Github 开源，访问该应用的 [Github 仓库地址](#) 可以获取详细信息。

示例应用的架构图如下：



应用部署

应用工作台是 DCE 5.0 的应用管理模块，支持创建/维护多种类型的应用、GitOps 和灰度发布等功能，可以快速将应用部署到任何集群。应用工作台支持基于 Git 仓、Jar 包、容器镜像、Helm 模板部署应用。本次实践基于 Helm 模板 部署示例应用。



部署应用之前需要满足如下的前提条件：

- 在容器管理中添加 Helm 仓库：

- 在微服务引擎中创建 Nacos 注册中心实例

注意记录注册中心的地址信息，后续安装应用时需要用到。

基于 Helm 模板部署

- 在应用工作台 -> 向导 -> 基于 Helm 模板 中，找到 opentelemetry-demo 应用，点击应用卡片进行安装

[← Helm 模板详情](#)

```

helm repo add open-insight https://openinsight-proj.github.io/openinsight-helm-charts
helm install my-otel-demo open-insight/opentelemetry-demo -n webstore-demo --create-namespace

```

demo-UI
we have a NodePort type svc named otel-demo-ui, find it and enjoy the otel-demo

reimplement components

- adservice

support switch redis instance

we use build-in redis by default, if you want to use a redis create by redis-operator(we use `openebs` as PersistentVolume, if you want to use others: `--set redis_resource.storageClassName=your-PersistentVolume`), your need: `--set redis_operator.enabled=true --set redis_resource.enabled=true`

2. 在 Helm 的安装界面，注意确认部署位置是否正确，然后按照下方要求更新 `JAVA_OPTS` 部分的参数配置。

[← 安装 - opentelemetry-demo](#)

参数配置

[YAML](#) [变化](#)

```

86   fieldRef:
87     apiVersion: v1
88     fieldPath: metadata.namespace
89   - name: OTEL_RESOURCE_ATTRIBUTES
90     value: >-
91       k8s.namespace.name=$(K8S_NAMESPACE),k8s.node.name=$(OTEL_RESOURCE_ATTRIBUTES_NODE_NAME),k8s.pod.name=$(OTEL_RESOURCE_ATTRIBUTES_POD_NAME),k8s.pod.uid=$(OTEL_RESOURCE_ATTRIBUTES_POD_UID),skoala.regi...
92   - name: JAVA_OPTS
93     value: >-
94       -javaagent:/jmx_prometheus_javaagent-0.17.0.jar=12345:/prometheus-jmx-config.yaml
95       -Dspring.extraAddLabel=Daocloud -Dspring.randomError=false
96       -Dspring.matrixRow=200 -Dmeter.port=8888
97       -Dspring.cloud.nacos.discovery.enabled=false
98       -Dspring.cloud.nacos.config.enabled=false
99       -Dspring.cloud.Sentinel.enabled=false
100      -Dspring.cloud.nacos.config.server-addr=nacos-test.skoala-test:8848
101      -Dspring.application.name=adservice-springcloud
102      -Dspring.cloud.nacos.discovery.server-addr=nacos-test.skoala-test:8848
103      -Dspring.cloud.nacos.discovery.metadata.k8s_cluster_id=xxx
104      -Dspring.cloud.nacos.discovery.metadata.k8s_cluster_name=skoala-dev
105      -Dspring.cloud.nacos.discovery.metadata.k8s_namespace_name=skoala-test
106      -Dspring.cloud.nacos.discovery.metadata.k8s_workload_type=deployment

```

根据上方记录的注册中心地址，将下方带有注释的参数更新如下：

```
java
-javaagent:./jmx_prometheus_javaagent-0.17.0.jar=12345:./prometheus-jmx-config.yaml
-Dspring.extraAdLabel=Daocloud -Dspring.randomError=false
-Dspring.matrixRow=200 -Dmeter.port=8888
-Dspring.cloud.nacos.discovery.enabled=false      # 修改，以启用 Nacos 服务注册发现
-Dspring.cloud.nacos.config.enabled=false        # 修改，以启用 Nacos 配置管理能力
-Dspring.cloud.sentinel.enabled=false
-Dspring.cloud.nacos.config.server-addr=nacos-test.skoala-test:8848      # 修改，以配置 Nacos 注册中心地址
-Dspring.application.name=adservice-springcloud
-Dspring.cloud.nacos.discovery.server-addr=nacos-test.skoala-test:8848      # 修改，以配置 Nacos 注册中心地址
-Dspring.cloud.nacos.discovery.metadata.k8s_cluster_id=xxx                  # 修改，以配置 Nacos 注册中心所在集群 ID
-Dspring.cloud.nacos.discovery.metadata.k8s_cluster_name=skoala-dev          # 修改，以配置 Nacos 注册中心所在集群名称
-Dspring.cloud.nacos.discovery.metadata.k8s_namespace_name=skoala-test        # 修改，以配置 Nacos 注册中心所在命名空间
-Dspring.cloud.nacos.discovery.metadata.k8s_workload_type=deployment
-Dspring.cloud.nacos.discovery.metadata.k8s_workload_name=adservice-springcloud
-Dspring.cloud.nacos.discovery.metadata.k8s_service_name=adservice-springcloud
-Dspring.cloud.nacos.discovery.metadata.k8s_pod_name=${HOSTNAME}
-Dspring.cloud.sentinel.enabled=false
-Dspring.cloud.sentinel.transport.dashboard=nacos-test-sentinel.skoala-test:8080
```

3. 应用创建成功后，会显示在应用工作台的 Helm 应用列表。

名称	状态	命名空间	Chart	仓库	创建时间
open-insight	已部署	skoala-gateway-lb	open-insightv0.5.0	openinsight	2022-12-27 10:47

Java 项目自行开发调试

如果采用其他部署方式，配置注册中心地址的方法可能有所不同。Java 项目在开发时需要集成 Nacos 的 SDK，而 DCE 5.0 提供的注册中心完全兼容开源 Nacos，所以可以直接使用开源 Nacos 的 SDK。具体操作步骤可参考[基于 Jar 包部署 Java 应用](#)。

使用 `java -jar` 启动项目时，添加对应的环境变量配置即可

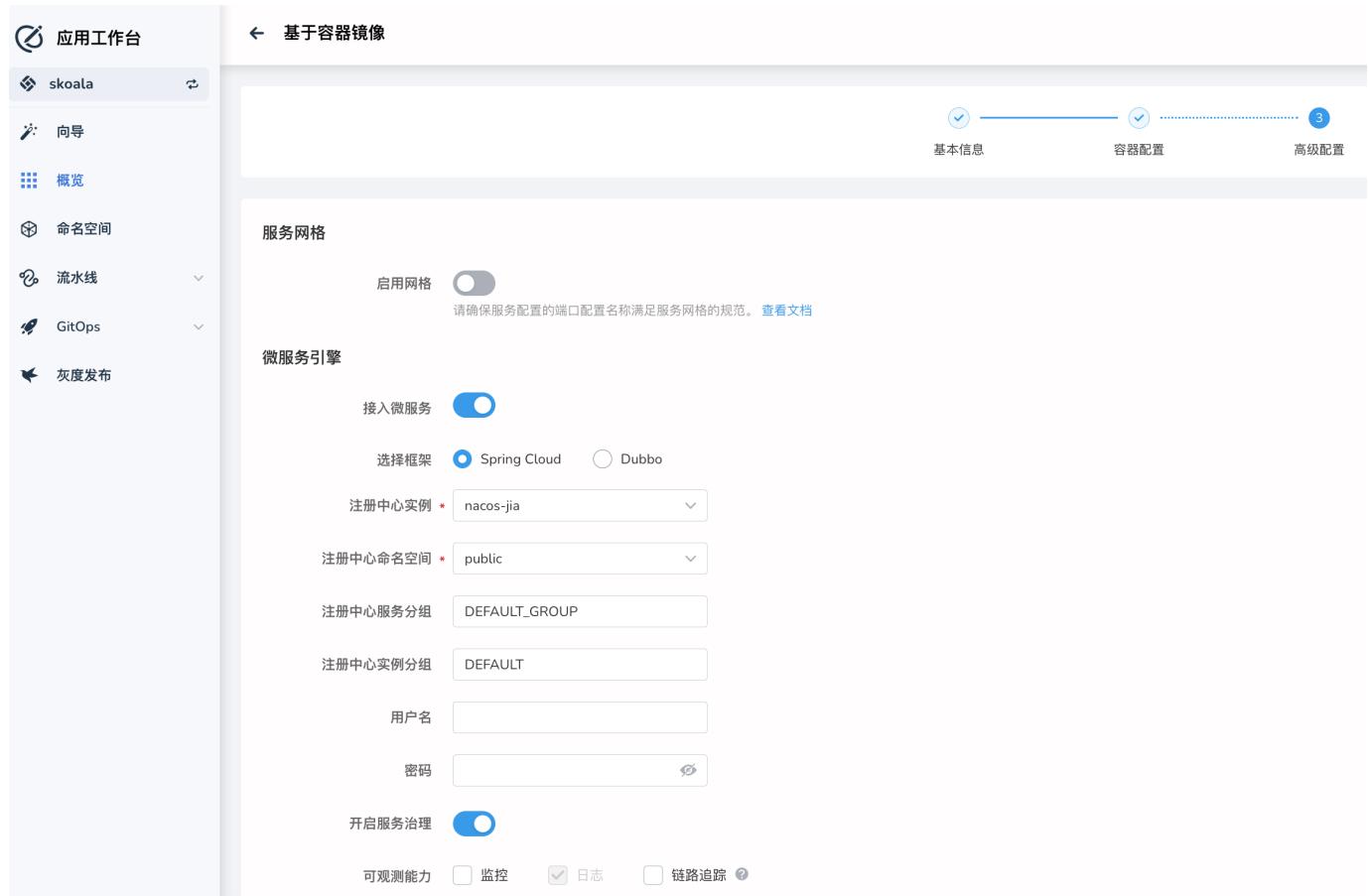
```
-Dspring.cloud.nacos.discovery.enabled=false      # 启用 Nacos 服务注册发现
-Dspring.cloud.nacos.config.enabled=false        # 启用 Nacos 配置管理能力
-Dspring.cloud.sentinel.enabled=false
-Dspring.cloud.nacos.config.server-addr=nacos-test.skoala-test:8848      # 配置 Nacos 注册中心地址
-Dspring.application.name=adservice-springcloud
-Dspring.cloud.nacos.discovery.server-addr=nacos-test.skoala-test:8848      # 配置 Nacos 注册中心地址
-Dspring.cloud.nacos.discovery.metadata.k8s_cluster_id=xxx                  # 配置 Nacos 注册中心所在集群 ID
-Dspring.cloud.nacos.discovery.metadata.k8s_cluster_name=skoala-dev          # 配置 Nacos 注册中心所在集群名称
-Dspring.cloud.nacos.discovery.metadata.k8s_namespace_name=skoala-test        # 配置 Nacos 注册中心所在命名空间
-Dspring.cloud.nacos.discovery.metadata.k8s_workload_type=deployment
-Dspring.cloud.nacos.discovery.metadata.k8s_workload_name=adservice-springcloud
-Dspring.cloud.nacos.discovery.metadata.k8s_service_name=adservice-springcloud
-Dspring.cloud.nacos.discovery.metadata.k8s_pod_name=${HOSTNAME}
```

!!! note

上面的 `metadata` 信息不能缺失，否则注册中心中呈现的服务会缺失这部分信息。

使用容器镜像部署

如果选择基于容器镜像部署应用，可以直接在用户界面配置中开启微服务治理并选取对应的注册中心模块，操作更简便。具体步骤可参考[基于 Git 仓库构建微服务应用](#)。



启用传统微服务治理

开始使用微服务治理功能之前，需要在对应注册中心下的插件中心开启对应的治理插件。插件中心提供 Sentinel 治理和 Mesh 治理两种插件，支持通过用户界面实现可视化配置。安装插件后可以扩展微服务治理能力，满足不同场景下的业务诉求。

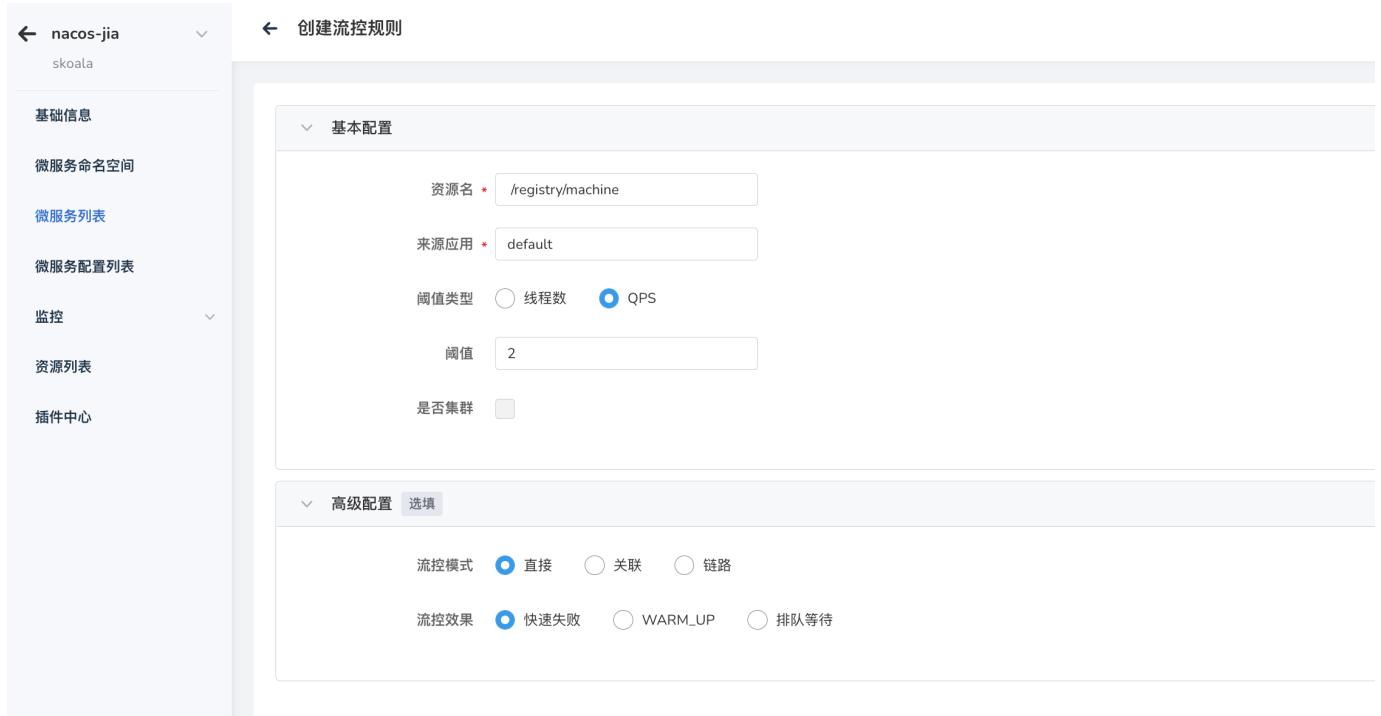
本次实践采用传统微服务治理，即开启 Sentinel 治理插件。如需了解详细步骤，可参考[启用 Sentinel 治理插件](#)。

配置对应的治理规则

应用部署成功后，可以在之前准备注册中心下的 微服务列表 中查看对应的服务。微服务列表提供流控规则、熔断降级、热点规则、系统规则、授权规则等流量治理规则。本次实践以流控规则为例进行演示。

配置流控策略

这里限流策略示例，我们通过简单的配置即可为服务增加对应的限流策略。



测试流控策略

通过访问服务地址，我们可以看到在 1 分钟内请求次数大于 2 次之后，后续请求都会被拦截；在超过 1 分钟后自动恢复。

暴露 API 并访问应用

微服务应用部署完成后，需要通过 API 网关将应用入口开放给外部访问，完成这一步才是完成的服务使用体验。为了暴露服务 API，需要创建云原生网关，将服务接入该网关，并创建对应的 API 路由。

创建云原生网关

首先需要创建一个云原生网关，具体操作步骤可以参考：[创建云原生网关](#)

!!! note

创建网关时，应该将网关部署在示例应用所在的集群，并且该网关需要管辖示例应用所在的命名空间。

部署集群	命名空间	最近更新时间	控制节点实例数	工作节点实例数
skoala-dev	skoala-ruby	2022-12-28 11:42	1 / 1	1 / 1

接入服务

基于 DCE 5.0 的特性，云原生网关可以自动发现所管辖命名空间内的服务，所以无需手动接入服务。

本次演示采用 Nacos 注册中心的服务，很大程度上扩宽了网关可接入的服务数量，可以在 服务接入 中选择接入 Nacos 注册中心的服务。

The screenshot shows the 'Service List' section of the Dev-Gateway NP interface. It displays a table of services with columns: Service Name, Connection Information, Source, Type, Namespace, Associated APIs, and Creation Time. The table includes entries for 'my-otel-demo-adservice', 'nacos-client-springcloud', 'binding-svc', 'amamba-argocd-server', 'vpa-webhook', and 'amamba'. A search bar and a 'Add Service' button are also visible.

!!! info

当服务不在网关所管辖的命名空间内，或者想要接入注册中心或者其他外部服务时（使用域名/IP），可以采用手工接入服务的方式。

创建 API 路由

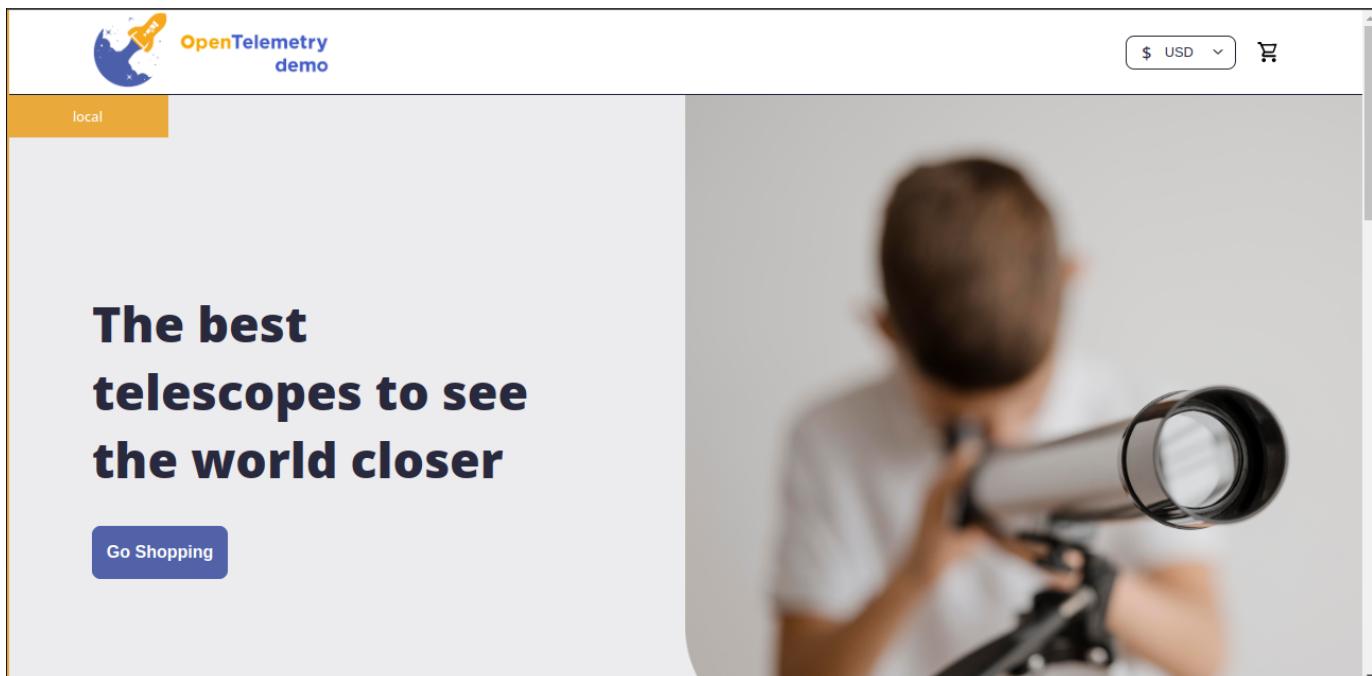
参考文档 [添加 API](#) 创建对应的 API 路由。

The screenshot shows the 'Create API Route' form. It includes fields for 'API Name' (ad-api), 'API Group' (default), 'Associated Domain' (choose from dropdown), 'Matching Rules' (path prefix /, request methods: GET, POST, DELETE, PUT, PATCH, OPTIONS, HEAD), 'Target Service' (selected as Backend Service), and a 'Service List' table with one entry: 'my-otel-demo-adservice' with weight 100.

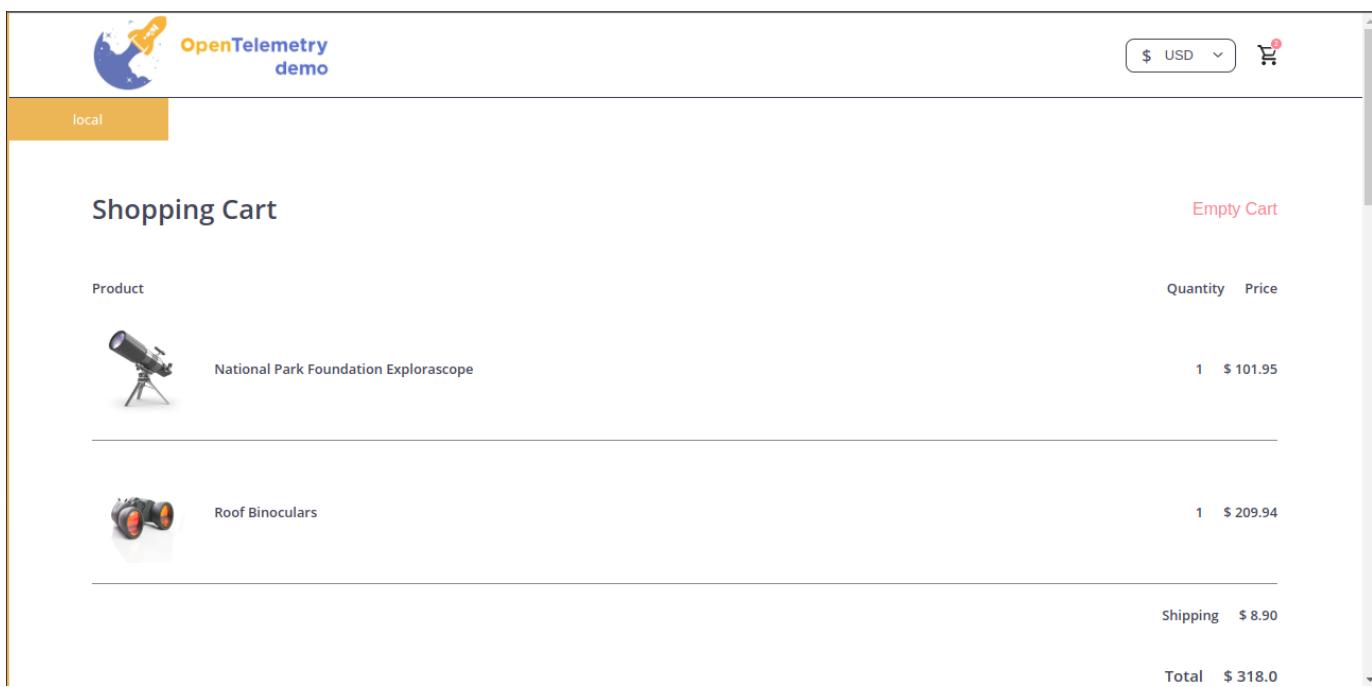
访问应用

当网关 API 创建完成后，使用创建 API 时配置的 域名 与 外部 API 路径 即可成功访问到应用页面，如下图所示。

示例应用的首页：



示例应用的订单确认页面：



结语

以上就是整个微服务引擎模块的体验之旅。在整个 DCE 5.0 的能力支持下我们顺利完成了应用部署、启用微服务治理、配置并测试微服务治理策略、通过云原生网关开放 API、实际访问应用等操作。

更多能力

当我们的应用部署成功之后，对于后续的应用维护过程中，实际非常依赖 DCE 5.0 提供的可观测能力。接下来，我们会补充对应可观测能力实践。

- 查看应用部署后的拓扑结构
- 查看应用的日志内容
- 查看云网关网关 API 的访问日志

2. Intro

2.1 产品优势

「DaoCloud 道客」提供的微服务引擎包含微服务治理中心和微服务网关两个模块，提供微服务的注册发现、流量治理、配置管理、网关 API 等一组简单实用且高性能的微服务治理能力，帮助从传统微服务架构稳定升级到云原生微服务架构。

相比其他同类产品，「DaoCloud 道客」提供的微服务引擎具有以下优势：

==== “一站式治理”

提供服务注册发现、配置管理、流量治理、链路追踪、指标监控、网关管理、API 管理、域名管理、监控告警、网关策略等功能，满足微服务生命周期中各个阶段的管理需求，实现一站式治理。

==== “无感迁移”

完全兼容 Nacos 开源注册中心和 Envoy、contour 开源网关，支持在零代码改造的前提下完成从自建服务到「DaoCloud 道客」微服务引擎的迁移。传统微服务无需作任何改动，即可通过注册中心接入到微服务引擎，进而实现流量治理、配置管理、链路追踪、指标监控等功能。

==== “平稳过渡”

统一管理传统微服务和云原生微服务，支持接入传统注册中心（Zookeeper、Eureka、Nacos）和云原生注册中心（Kubernetes、Service Mesh）类型的注册中心，实现从传统微服务生态向云原生微服务生态的平稳过渡，助力企业走向云原生化。

==== “开放兼容”

微服务引擎支持传统和云音乐的注册中心，也支持 Spring Cloud、Dubbo 等主流的开源微服务框架，以及 Envoy、Sentinel、Contour 等开源网关组件。此外，还可以和 [DCE 5.0] (../../../../dce/what.md) 的多云编排、数据服务中间件、服务网格、应用工作台等模块自由组合，实现更多定制化、精细化的功能。

==== “可视化界面与数据”

通过功能封装对外暴露简单易用的交互页面，支持通过简洁直观的 UI 界面进行所有操作，降低操作难度，真正实现点点鼠标就可以完成微服务整个生命周期中的各类管理操作。

2.2 环境要求

微服务引擎中集成了多款组件。各个组件必须达到特定的版本才能确保微服务引擎的高效运行。具体的组件版本要求如下：

!!! note

稳定支持版本：微服务引擎团队经过验证后认为可稳定支持的组件版本，建议参照下表使用特定版本的组件。

组件类型	开源组件	稳定支持版本	备注
必备	Nacos	2.1.0	
必备	Sentinel	v1.8.4	
必备	contour	1.21.1-debian-11-r0	Helm contour Chart
必备	envoy	1.22.2-debian-11-r1	
必备	nginx	1.21.6-debian-11-r5	
必备	Helm	v3.8.2	
必备	Grafana	v8.4.6	
可选	Zookeeper	release-3.5.10	
可选	Eureka	v1.10.17	
可选	Istio	1.14.1	
可选	Kubernetes	1.23, 1.22, 1.21	
可选	prometheus	v2.36.0	

2.3 微服务引擎 Release Notes

本页列出微服务引擎的 Release Notes，便于您了解各版本的演进路径和特性变化。

2023-02-25

v0.18.0

新功能

- 新增 添加注册中心配置中心分离 API
- 新增 添加概览相关逻辑及 API

修复

- 修复 gateway-api 镜像版本问题
- 修复 负载均衡模式网关的 IP 池加载问题
- 修复 健康检查相关问题

2023-02-22

v0.17.1

修复

- 修复 发布过程 Chart 更新问题

2023-02-22

v0.17.0

新功能

新增 网关 NodePort 支持

- 新增 网关 LoadBalancer 支持
- 新增 Sentinel 规则统计 API
- 新增 Sentinel 治理的服务列表 API
- 新增 网关 API 的 Cookie 重写策略
- 新增 概览数据定时任务
- 新增 定时收集异常 Sentinel 任务
- 新增 Sentinel 集群流控详情 API
- 新增 网关接入服务列表端口选择
- 新增 网关服务健康检查策略
- 新增 网关 API 中对健康策略的支持
- 新增 Sentinel 统计相关 API
- 新增 支持 chart 离线化的 CI 流程
- 新增 在每日构建中增加外部镜像安全扫描能力
- 新增 发布自动更新 chart 中镜像版本

修复

- 修复 Nacos Namespace 创建异常的问题
- 修复 Nacos 持久化存储修改异常的问题

- 修复 Nacos 生命周期管理资源校验问题
- 修复 网关监控面板数据展示问题
- 修复 Ghippo 链接 GRPC 地址缺失问题
- 修复 Sentinel 获取集群流控 API 的问题
- 修复 托管 Nacos 资源状态不更新的问题
- 修复 Sentinel 适配 Nacos public 字符串问题
- 修复 Sentinel 获取资源 API 没有聚合不同实例问题
- 修复 Sentinel 系统规则不生效问题
- 修复 网关服务注册中心类型分页错误的问题
- 修复 创建服务端口错误的问题
- 修复 数据库初始化组建的问题
- 修复 使用 Helm 命令替代 Argocd 部署 Alpha 环境
- 修复 基础镜像 CVE 问题并升级至 3.17.2

优化

- 优化 升级 gateway-api 到 v0.6.0
- 优化 待更新资源获取由 clusterpedia 改为 client-go
- 优化 Sentinel 应用监控模版
- 优化 将离线 chart 构建 CI 步骤独立
- 优化 Contour 升级到 v1.24.1
- 优化 envoy 升级到 v1.25.1
- 优化 通过 Chart 能力使 Skoala Init 安装时固定命名空间

2022-12-30

v0.16.1

修复

- 修复 构建镜像时重复创建 builder 的问题

优化

- 优化 Sentinel 应用监控面板细节

2022-12-29

v0.16.0

修复

- 修复 Sentinel 调用具备认证开启 Nacos 接口的问题
- 修复 nacos-operator 频繁修改服务资源的问题

优化

- 优化 添加 Sentinel 服务的 Grafana 监控面板
- 优化 升级 Insight 为最新版本支持通过集群名查询监控数据

2022-12-28

v0.15.2

新功能

- 新增 网关 API 对于认证服务器的支持
- 新增 托管注册中心服务接入 API
- 新增 Sentinel 集群流控相关 API

修复

- 修复 Sentinel 规则拼接错误的问题
- 修复 Sentinel 仪表盘名称的问题
- 修复 管理组件 Chart 对于生产换环境的 Service IP 问题
- 修复 Nacos 控制器处理逻辑的问题
- 修复 与集群管理集成的 egress 地址问题

优化

- 优化 托管 Nacos 监控仪表盘问题
- 优化 nacos-operator 数据库初始化的文件获取地址
- 优化 更新 Sentinel 镜像至 v0.6.0

2022-12-22

v0.14.0

新功能

- 新增 Init Chart 所需镜像的离线支持
- 新增 获取托管 Nacos 的令牌

修复

- 修复 Skoala Chart 中 Values 命名问题
- 修复 CI 流程中的镜像问题

优化

- 优化 设置默认日志输出到控制台
- 优化 升级 nacos-operator 到社区版本
- 优化 更新 Nacos 自定义资源的认证开启支持
- 优化 设置默认组件日志级别

2022-12-21

v0.13.0

新功能

- 新增 对接中间件 MySQL 和 Redis 的相关 API
- 新增 网关 JWT 验证支持的 API
- 新增 网关域名校验逻辑
- 新增 Sentinel 资源列表 API
- 新增 网关查询注册中心服务的接口
- 新增 版本发布后推送 Init Chart 至 addon 仓库

- 新增 版本发布时完成 gitlab release 操作
- 新增 动态更改日志级别

修复

- 修复 全局限流规则更新时不生效的问题
- 修复 Envoy Log Level 未设置问题
- 修复 更新网关时异常未判断的问题
- 修复 托管 Nacos 的数据库初始化问题

优化

- 优化 注册中心列表按更新时间降序排列
- 优化 统一网关 JWT 相关的字段名称
- 优化 网关域名列表增加是否开启 JWT 的字段
- 优化 Sentinel 服务名连接符的逻辑
- 优化 升级 Contour 到 1.23 版
- 优化 升级 Envoy 到 1.24 版
- 优化 升级 k8s.io/相关组件到 0.25 版
- 优化 将 go-replayers 组件回归社区版本
- 优化 将 go-helm-client 组件回归社区版本
- 优化 升级 Contour 到 1.23.1 版
- 优化 修改 Agent 组件为强制不注入网格边车
- 优化 将 Nacos 镜像默认配置回归社区版本
- 优化 移除 Nacos 镜像相关 CI 流程

2022-12-13

v0.12.2

新功能

- 新增 添加 Sentinel 自身监控的 Grafana 模板支持
- 新增 添加自定义配置网关索引的配置信息

修复

- 修复 微服务集成可观测组件的状态问题
- 修复 注册中心开启网格插件能力的治理状态问题
- 修复 网关日志索引问题
- 修复 前置依赖检查接口的问题
- 修复 Sentinel 与 Nacos 默认命名空间匹配的逻辑问题
- 修复 连接容器管理模块的端口异常情况的逻辑

2.4 适用场景

「DaoCloud 道客」微服务引擎是面向业界主流微服务生态的一站式微服务管理平台，主要提供微服务治理中心和微服务网关两个维度的功能，具体包括服务注册发现、配置管理、流量治理、服务级别的链路追踪、API 管理、域名管理、监控告警等，覆盖了微服务生命周期中的各种管理场景。

基于所提供的各项功能，微服务引擎可用于微服务的注册与发现、配置管理、微服务流量治理，以及微服务网关管理等场景。典型的应用场景如下：

一站式管理大量异构微服务

随着应用服务的爆发式增长，微服务的数量越来越多，使用的架构也可能各不相同，传统的微服务和云原生微服务并存。各个微服务之间互相调用，互相依赖，牵一发而动全身，容易出现级联效应，造成系统雪崩。随着微服务系统的运维难度和成本越来越高，用户需要一个能够同时管理传统微服务和云原生微服务、监控服务信息、追踪服务级别的链路调用情况、统一管理微服务配置、提供微服务网关的一站式产品。微服务引擎完全覆盖了这些场景需求，能够很好地满足用户需要。

从传统到云原生的转变和过渡

云原生在弹性扩展、屏蔽底层差异、故障处理等方面具有显著优势，一些企业在云原生浪潮的影响下，希望采取稳态模式逐渐从传统的微服务架构转向云原生微服务架构，有些企业希望采取敏态模式实现快速转变。无论采取稳态还是敏态模式，微服务引擎都是很好的选择，因为其支持传统微服务和云原生微服务的统一纳管，支持 Sentinel 和 Service Mesh 两种流量治理模式，分别适用于传统微服务和云原生微服务的东西向流量治理。

微服务网关的可视化与高效能

很多开源的网关产品仅支持命令行操作，使用门槛高、难度大。某些情况下，多个微服务共用一个网关，网关资源开销大，而且可能逐渐成为整个系统的资源瓶颈。使用「DaoCloud 道客」微服务引擎在 Contour 的基础上增加界面操作能力，大大降低了使用门槛和维护成本，可以实现多网关管理，在不同集群和不同命名空间下轻松创建多个网关实例。这些网关实例之间互相隔离，因而具有更高的可用性和稳定性。此外，还可以使用网关路由的级联功能，实现微服务的蓝绿部署动态切换。

2.5 功能总览

「DaoCloud 道客」微服务引擎是面向业界主流微服务生态的一站式微服务管理平台，主要提供微服务治理中心和微服务网关两个维度的功能，具体包括服务注册发现、配置管理、流量治理、服务级别的链路追踪、API 管理、域名管理、监控告警等，覆盖了微服务生命周期中的各种管理场景。微服务引擎具有很强的兼容性，不仅可以无缝对接 DCE 5.0 的其他组件，也可以完美兼容 Spring Cloud、Dubbo 等开源生态，帮助您更便捷地使用开源微服务技术构建自己的微服务体系。

微服务注册与发现

统一纳管传统微服务和云原生微服务，实现从传统微服务生态向云原生微服务生态的平稳过渡，助力企业走向云原生化。

- 支持创建 Nacos 托管中心，进而管理微服务命名空间、治理微服务流量、管理微服务配置、链路追踪与监控等。
- 支持接入 Eureka、Zookeeper、Nacos 三类传统的微服务注册中心。
- 支持接入 Kubernetes 和 Service Mesh 两类云原生微服务注册中心。

微服务流量治理

在流量治理层面，采用线上流量治理方案，可以快速与主流开源微服务框架集成，用 Sentinel 和 Mesh 解决不同生产情况下的痛点。

- 支持通过 Sentinel 使用流控、熔断降级、热点、系统、授权、集群流控等规则治理传统微服务的东西向流量。
- 支持通过 Service Mesh 通过虚拟服务、目标规则、网关规则在网格中治理微服务流量。

微服务配置中心

Nacos 托管注册中心可作为微服务的配置管理器，可以从不同项目中抽取通用配置事先统一管理，也可以为同一项目应用多个不同配置，实现差异化管理。

- 基于微服务命名空间和分组（Group）对配置文件进行隔离。
- 结合 `@RefreshScope` 注解动态更新配置项。
- 管理配置文件的历史版本，支持版本差异对比并一键回滚到特定版本。
- 支持查询当前配置的监听者以及 MD5 校验值。
- 提供示例代码，便于新手快速使用客户端编程消费该配置，降低新手使用门槛。

微服务网关

微服务网关肩负管理微服务南北流量管控的重要作用，提供 API 管理、接口限流、多种策略安全认证、黑白名单、路由转发、MockAPI 等能力，同时提供企业级高性能和高扩展的云服务能力。

- 多网关管理：原生支持对容器管理模块中的多集群、多命名空间的网关实例进行管理，支持网关实例的全生命周期管理。
- API 策略管理：通过图形化界面进行 API 的增删改查，配置 API 策略，例如负载均衡、路径改写、超时配置、重试机制、请求头/响应头重写、WebSocket、本地限流、健康检查等，同时保障原生 API 的能力不受影响。
- 插件管理：提供丰富的插件功能，支持安全、流量管控、缓存等插件，支持一键开启/停用插件。
- 监控告警：微服务网关在部署时会自动配置监控、告警等功能。每个网关都自带完善的资源监控和网关业务监控。

3. Ms gateway

3.1 监控告警

微服务网关通过内置 Grafana 看板监控网关的资源使用情况、Envoy 的监控详情、服务监控详情、APIServe、HTTPProxy 等资源的运行信息。

查看监控告警信息的步骤如下所示：

- 进入微服务引擎模块，在左侧导航栏点击微服务网关，点击目标网关的名称。

微服务网关	部署集群	命名空间	创建时间	最近更新时间	控制节点实例数	工作节点实例数
dev-ruby	skoala-dev	skoala-ruby	2022-12-22 15:34	2022-12-22 15:34	0 / 0	0 / 0
dev-gateway-lb	skoala-dev	skoala-gateway-lb	2022-12-22 11:22	2022-12-22 11:22	0 / 0	0 / 0
dev-gateway-np	skoala-dev	skoala-gateway-np	2022-12-07 11:46	2022-12-08 10:19	0 / 0	0 / 0

- 在左侧导航栏点击 监控告警，点击 业务监控看版 页签可查看业务监控信息。

在页面右上角可选择监控时间窗口和刷新周期。

业务监控看板

Last 30 minutes

资源 (3 panels)

Envoy详情 (10 panels)

服务详情

端点(健康百分比) 端点(健康) 端点(不健康) 端点(总计)

上行应答(2xx) 上行应答(3xx) 上行应答(4xx) 上行应答(5xx)

3. 点击 资源看版 页签可查看资源监控信息。

资源看板

Last 7 days

资源

CPU利用率 内存利用率 数据接收

APIServer统计 (2 panels)

HTTPProxy (8 panels)

3.2 Api

添加 API

微服务网关支持对网关实例的 API 进行全生命周期管理，包括 API 的添加、更新和删除。本页介绍如何添加 API。

前提条件

- 有可选的域名，可参考[域名管理](#)创建域名。
- 如果 API 的目标服务为后端服务，则需要确保有可选的后端服务，可参考通过[手动](#)或[自动](#)方式接入服务。

创建 API 的步骤如下：

1. 点击网关名称进入网关概览页面，然后在左侧导航栏点击 API 管理，在页面右上角点击 添加 API，进入 API 的创建页面。

The screenshot shows the DaoCloud service gateway management interface. On the left, there's a sidebar with various options like 'Default', 'API Management' (which is selected and highlighted with a red box), 'Service Access', 'Monitoring & Alerts', 'Log View', 'Plugin Center', and 'Domain Management'. The main area is titled 'Microservices Gateway michelle.video / API Management'. It has a search bar and a table with columns: API Name, Associated Domain, Request Method, Request Path, Target Service Type, Status, Last Update Time, and Operations. A blue 'Add API' button is located in the top right corner of the main area, also highlighted with a red box.

2. 填写配置信息。

配置分为基本配置和策略配置两部分。填写基本配置信息时需要注意：

- API 名称：包含小写字母、数字和以及特殊字符(- .)，不能以特殊字符开头和结尾。
- API 分组：选择 API 所属的分组组名。如果输入不存在的分组名称，则自动创建一个新的分组。
- 关联域名：填写关联域名后，可以通过 域名 + 端口号 的方式访问 API。找不到域名时可以添加新域名，可参考[添加域名](#)。
- 匹配规则：只允许符合该规则的请求通过。如果设置了多条规则，需要同时满足所有规则才能放行。如果添加了请求头，则需要在访问该 API 时添加相应的请求头。
- 请求方法：选择 HTTP 协议的请求方式。有关各种请求方式的详细说明，可参考 W3C 的官方文档[方式定义](#)。
- 目标服务：选择将请求直接发送到后端服务，或者重定向到其他服务，或者直接返回 HTTP 状态码。
- 如果选择后端服务，则需要配置权重。权重越大，网关向其分发的流量就越多。

The screenshot shows the 'Create Microservices Gateway API' dialog. On the left, there's a sidebar with 'Default', 'API Management' (selected), 'Service Access', 'Monitoring & Alerts', 'Log View', 'Plugin Center', and 'Domain Management'. The main dialog has tabs for 'Basic Configuration' (必填) and 'Policy Configuration'. In the 'Basic Configuration' tab, there are fields for 'API Name' (video), 'API Group' (video), 'Associated Domain' (ad.service.virtualhost), and 'Matching Rules' (Path: 前缀匹配 /, Request Methods: GET checked, POST, DELETE, PUT, PATCH, OPTIONS, HEAD). Below that is the 'Target Service' section, which includes a table with columns: Service List, Service Name, and Weight. A service named 'dce-system-dxinsight' is listed with a weight of 56. The 'Policy Configuration' tab is also visible at the bottom. At the very bottom right, there are 'Cancel' and 'Confirm' buttons.

支持九种 API 策略：负载均衡、路径改写、超时配置、重试机制、请求头重写、响应头重写、WebSocket、本地限流、健康检查。有关各项策略的配置说明，可参考[API 策略配置](#)。

3. 在页面右下角点击 确定。

点击确定后，如果所有配置都正常，右上角会弹出创建网关 API 成功的提示信息。可以在 API 管理页面查看新建的 API。

4. API 上线

API 创建成功后，默认处于下线状态，此时无法访问。需要将 API 调整为上线，才能正常访问。API 上线有两种方式。

- 在 API 在 API 管理 页面找到需要更新的 API，在该 API 的右侧点击 选择 API 上线。

API 名称	关联域名	API 分组	请求方法	请求路径	目标服务类型	状态	最近更新时间
video	ad.service.vir...	video	GET	/	集群服务	已下线	2022-11-01 10:5
ad-service	ad.service.vir...	video	GET	/	集群服务	在线	

- 点击 API 名称进入 API 详情页，在页面右上角点击 并选择 API 上线。

Header 关键字	判断条件	对应值
暂无数据		

服务名称	权重
dce-system-dxinsight	56

!!! info

点击 API 名称进入 API 详情，可查看 API 的详细配置信息，例如上下线状态、域名、匹配规则、目标服务、策略配置等。

配置 API 策略

DCE 5.0 微服务网关支持九种 API 策略：负载均衡、路径改写、超时配置、重试机制、请求头重写、响应头重写、WebSocket、本地限流、健康检查。可以单独使用某一种策略，也可以组合使用多种策略达到最佳实践。

有两种方式可以配置 API 策略：

- 在创建 API 的过程中设置策略，参考[添加 API](#)。
- 在 API 创建完成后通过[更新 API 策略配置](#)进行调整。

每一项策略的配置说明如下：

视频教程：[API 策略的高级配置](#)

负载均衡

当 API 的目标后端服务为多实例服务时，可以通过负载均衡策略控制流量分发，根据业务场景调整不同服务的实例接收到的流量。

- 随机

默认的负载均衡策略。选择随机规则时，网关会将请求随机分发给后端服务的任意实例。在流量较小时，部分后端服务可能会负载较多。效果参考下图：

▲ **pythonProject** python3 lb_count.py

指向 10.244.1.135 的 流量为： 89

指向 10.244.2.207 的 流量为： 107

指向 10.244.3.193 的 流量为： 104

- 轮询

向后端服务的所有实例轮流分发请求，各个服务实例接收到的请求数基本相近。此规则可以在流量较小时保障流量的平均分配。效果参考下图：

▲ **pythonProject** python3 lb_count.py

指向 10.244.1.135 的 流量为： 101

指向 10.244.2.207 的 流量为： 99

指向 10.244.3.193 的 流量为： 100

- 权重

根据 API 目标后端服务的权重分发流量，权重数值越大，优先级越高，承担的流量也相对较多。服务权重的配置入口见下图：

创建微服务网关 API

基本配置 必填

API 名称 *

包含小写字母、数字和以及特殊字符(-_)，且不能以特殊字符开头和结尾，长度 63，创建后不可更改

API 分组 *

由名称检索下拉选取，分组名称不存在时可创建

关联域名 *

请选择域名

找不到域名, 去添加

匹配规则 *

路径 * 前缀匹配 Path 匹配值, 如: /user, /user/add

请求方法 *

全选
 GET POST DELETE PUT PATCH
 OPTIONS HEAD

请求头 [+ 添加](#)

目标服务 *

后端服务 重定向 直接返回

服务列表	服务名称	权重
contour-gateway-video-dev-gtw	34	<input checked="" type="checkbox"/>
dce-system-dxinsight	89	<input checked="" type="checkbox"/>

[+ 添加](#)

取消 确定

- Cookie

将来源请求头中属于相同 Cookie 的流量分发到固定的后端服务实例，前提是后端服务能够根据 Cookie 做出不同的响应处理。

- 请求 Hash

选择请求 Hash 时，可以通过一些高级策略来进负载均衡分配。当前支持的 Hash 策略为：IP、请求参数。

gateway-vid... Default

创建微服务网关 API

概览 API 管理 服务接入 监控告警 日志查看 插件中心 域名管理

基本配置 必填

策略配置 选填

负载均衡

负载均衡策略 随机 轮询 权重 Cookie 请求 Hash

* 选择使用请求哈希作负载均衡时，以下哈希策略必须至少启用一个

哈希请求头或请求参数	哈希策略	关键字	匹配终止
请求头	hash	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> 启用
+ 添加			

哈希来源IP 启用

路径改写

超时配置

重试机制

请求头重写

响应头重写

取消 确定

路径改写

如果对外暴露的 API 路径与后端服务提供的路径不一致，可以改写 API 路径使其与后端服务的路径一致，确保服务的正常访问。启用路径改写后，网关会将外部请求流量转发到重写后的路径。

注意：需要确保重写的路径是真实存在的，并且路径正确，以“/”开头。

The screenshot shows the DaoCloud API Gateway configuration interface. On the left sidebar, there are several navigation items: 概览 (Overview), API 管理 (API Management), 服务接入 (Service Access), 监控告警 (Monitoring & Alerting), 日志查看 (Log View), 插件中心 (Plugin Center), and 域名管理 (Domain Management). The main content area has a header '创建微服务网关 API' (Create Microservices Gateway API) and a sub-header '基础配置' (Basic Configuration). Below this, there is a section titled '策略配置' (Policy Configuration) with the note '必填' (Required). Under '策略配置', there are several policy options: 负载均衡 (Load Balancing) (disabled), 路径改写 (Path Rewrite) (enabled, with '原路径' set to '/' and '重写路径' set to '/test1'), 超时配置 (Timeout Configuration) (disabled), 重试机制 (Retry Mechanism) (disabled), 请求头重写 (Request Header Rewrite) (disabled), 响应头重写 (Response Header Rewrite) (disabled), WEBSOCKET (disabled), and 本地限流 (Local Flow Control) (disabled). At the bottom right of the configuration panel are '取消' (Cancel) and '确定' (Confirm) buttons.

超时配置

设置请求响应的最大时长，如果超出所设置的最大时长，则直接请求失败。超时时长支持数值类型为 ≥ 1 的整数值，时间单位为“秒 (s)”。

超时配置默认处于关闭状态，开启后必须配置超时时长。开启超时配置有助于减少异常处理导致的阻塞问题。

The screenshot shows the configuration interface for a microservices gateway API. On the left sidebar, there are navigation links: 概览 (Overview), API 管理 (API Management), 服务接入 (Service Access), 监控告警 (Monitoring & Alerting), 日志查看 (Log View), 插件中心 (Plugin Center), 域名管理 (Domain Management), and a Default dropdown. The main content area has a header '创建微服务网关 API' (Create Microservices Gateway API) with a back arrow. Below it, the '基本配置' (Basic Configuration) tab is active, indicated by a red '必填' (Required) badge. Other tabs include '策略配置' (Policy Configuration) and '进阶配置' (Advanced Configuration). Under '策略配置', several options are listed with toggle switches: 负载均衡 (Load Balancing), 路径改写 (Path Rewriting), 超时配置 (Timeout Configuration) (which is selected and shows a timeout of 5 seconds), 重试机制 (Retry Mechanism), 请求头重写 (Rewrite Request Headers), 响应头重写 (Rewrite Response Headers), WEBSOCKET, 本地限流 (Local Flow Control), and 健康检查 (Health Check). At the bottom right of the configuration panel are '取消' (Cancel) and '确定' (Confirm) buttons.

重试机制

微服务网关的 API 支持配置非常丰富的重试机制。启用重试机制后，网关会在请求失败时自动重新尝试访问。达到重试超时时间之后自动触发再次重试，当重试次数达到配置的最大重试次数时停止重试。重试机制默认处于关闭状态，开启后必须配置重试次数和重试超时时长。

支持通过自定义配置选择不同的重试条件，自定义重试状态码等。

HTTP 重试

- 5XX 响应错误：后端服务响应 HTTP status_code 大于 500 时进行重试。
- 网关错误：当响应结果为网关错误提示时，自动进行重试。
- 请求重置：当响应结果为请求重置消息时，自动进行重试。
- 连接失败：当响应结果为网络连接失败的返回时，自动进行重试。
- 拒绝流：当响应结果为后端服务将请求标记为拒绝处理时，自动进行重试。
- 指定状态码：当后端服务响应 HTTP status_code 为特定状态码时自动进行重试，支持配置特定的状态码。

GRPC 重试

- 请求被取消：当响应结果为 GRPC 请求被后端服务取消时自动进行重试。
- 响应超时：当后端服务响应超时，自动进行重试。
- 服务内部错误：当响应结果为服务内部错误时，自动进行重试。
- 资源不足：当响应结果为资源不足时，自动进行重试。

- 服务不可用时：当响应结果为后端不可用时，自动进行重试。

The screenshot shows the configuration interface for a microservices gateway API. The left sidebar includes links for Overview, API Management, Service Access, Monitoring & Alerting, Log View, Plugin Center, and Domain Management. The main panel has a breadcrumb path: gateway-video... > 创建微服务网关 API. The configuration screen is titled 'Basic Configuration' (必填) and includes sections for Load Balancing, Path Rewriting, Timeout Configuration, and the current 'Retry Mechanism'. Under 'Retry Mechanism', the 'Retry Times' is set to 3, and the 'Retry Timeout Duration' is 3 seconds. There is a 'More Custom Configuration' dropdown. Under 'HTTP Retry Timeout Conditions', '5XX' and '指定状态码' (Custom Status Code) are selected. A 'Custom Status Code' input field contains '404' with a '+' button for adding more. Under 'GRPC Timeout Conditions', '请求被取消' (Request Canceled), '资源不足' (Insufficient Resources), and '服务不可用' (Service Unavailable) are checked. At the bottom right are 'Cancel' and 'Confirm' buttons.

请求头/响应头改写

支持添加、修改、删除请求头和响应头及其对应的值。

- 增加请求头/响应头：使用 **设置** 动作，填写新的关键字和新值。
- 修改请求头/响应头：使用 **设置** 动作，填写已有的关键字并赋予新值。
- 移除请求头/响应头，使用 **移除** 动作，只填写需要移除的关键字即可，无需填写对应的值。

The screenshot shows the configuration interface for a microservices gateway API. On the left, there's a sidebar with navigation links: 概览 (Overview), API 管理 (API Management), 服务接入 (Service Access), 监控告警 (Monitoring & Alerts), 日志查看 (Log View), 插件中心 (Plugin Center), and 域名管理 (Domain Management). The main area is titled '创建微服务网关 API' (Create Microservices Gateway API) and shows the 'Basic Configuration' step. Under '策略配置' (Policy Configuration), there are sections for '负载均衡' (Load Balancing), '路径改写' (Path Rewriting), '超时配置' (Timeout Configuration), and '重试机制' (Retry Mechanism). The '请求头重写' (Request Header Rewrite) section is expanded, showing two rows of actions:

动作	关键字	值
移除	User-Agent	
设置	Host	demo.virtualhost

A '+ 添加' (Add) button is available to add more rules. Below this is the '响应头重写' (Response Header Rewrite) section, which currently has one row of settings:

动作	关键字	值
设置	date	1027

A '+ 添加' (Add) button is also present here. At the bottom right of the configuration panel are '取消' (Cancel) and '确定' (Confirm) buttons.

Websocket

WebSocket 是一种在单个 TCP 连接上进行全双工通信的协议。WebSocket 使得客户端和服务器之间的数据交换变得更加简单，允许服务端主动向客户端推送数据。在 WebSocket API 中，浏览器和服务器只需要完成一次握手，两者之间就直接可以创建持久性的连接，并进行双向数据传输。

启用 Websocket 之后支持通过 Websocket 协议访问 API 的后端服务。

The screenshot shows the DaoCloud API Gateway configuration interface. The left sidebar includes sections for Overview, API Management, Service Access, Monitoring, Log View, Plugins, and Domain Management. The main area is titled 'Create Microservices Gateway API' under 'Default'. It features a 'Basic Configuration' step with a red '必填' (Required) label, followed by a 'Policy Configuration' step with a 'Select' label. Under 'Policy Configuration', there are several toggle switches: 负载均衡 (Load Balancing), 路径改写 (Path Rewriting), 超时配置 (Timeout Configuration), 重试机制 (Retry Mechanism), 请求头重写 (Request Header Rewrite), 响应头重写 (Response Header Rewrite), WEBSOCKET (selected), 本地限流 (Local Throttling), and 健康检查 (Health Check). At the bottom right are 'Cancel' and 'Confirm' buttons.

本地限流

微服务网关支持丰富的限流能力，支持在 API 层级启用本地限流能力。

- 请求速率：时间窗口（秒/分/时）内允许的最大请求速率，例如每分钟最多允许 3 次请求。支持输入 $>=1$ 的整数。
- 允许溢出速率：达到预设的请求速率时，仍旧允许额外处理一部分请求，适用于业务突增的流量高峰时段。支持输入 $>=1$ 的整数。
- 限制返回码：默认返回码为 429，表示请求次数过多。可参考 envoy 官方文档了解[本地限流支持的状态码](#)。
- Header 关键字：默认为空，可根据需求自行设置。

下图中的配置表示：每分钟最多允许请求 8 次 (3+5)，第 9 次访问时会返回 429 状态码，提示访问次数过多。每次请求成功后返回的响应内容都会带上 `ratelimit: 8` 响应头。

The screenshot shows the configuration interface for a specific API gateway endpoint named 'gateway-video'. The left sidebar includes sections for Overview, API Management, Service Access (with Service List and Source Management), Monitoring & Alerting, Log View, Plugin Center, and Domain Management. The main panel is titled 'Create Microservices Gateway API' and displays various configuration options under the 'Local Rate Limiting' tab. Key settings include:

- Request Rate: Set to 3 requests per minute.
- Overflow Rate: Set to 5 requests per minute.
- Restriction Return Code: Set to 429.
- Header Configuration:

Header	键	值
	ratelimit	8

A '+' button is available to add more entries.
- Health Check: An option to enable health checks.

At the bottom right are 'Cancel' and 'Confirm' buttons.

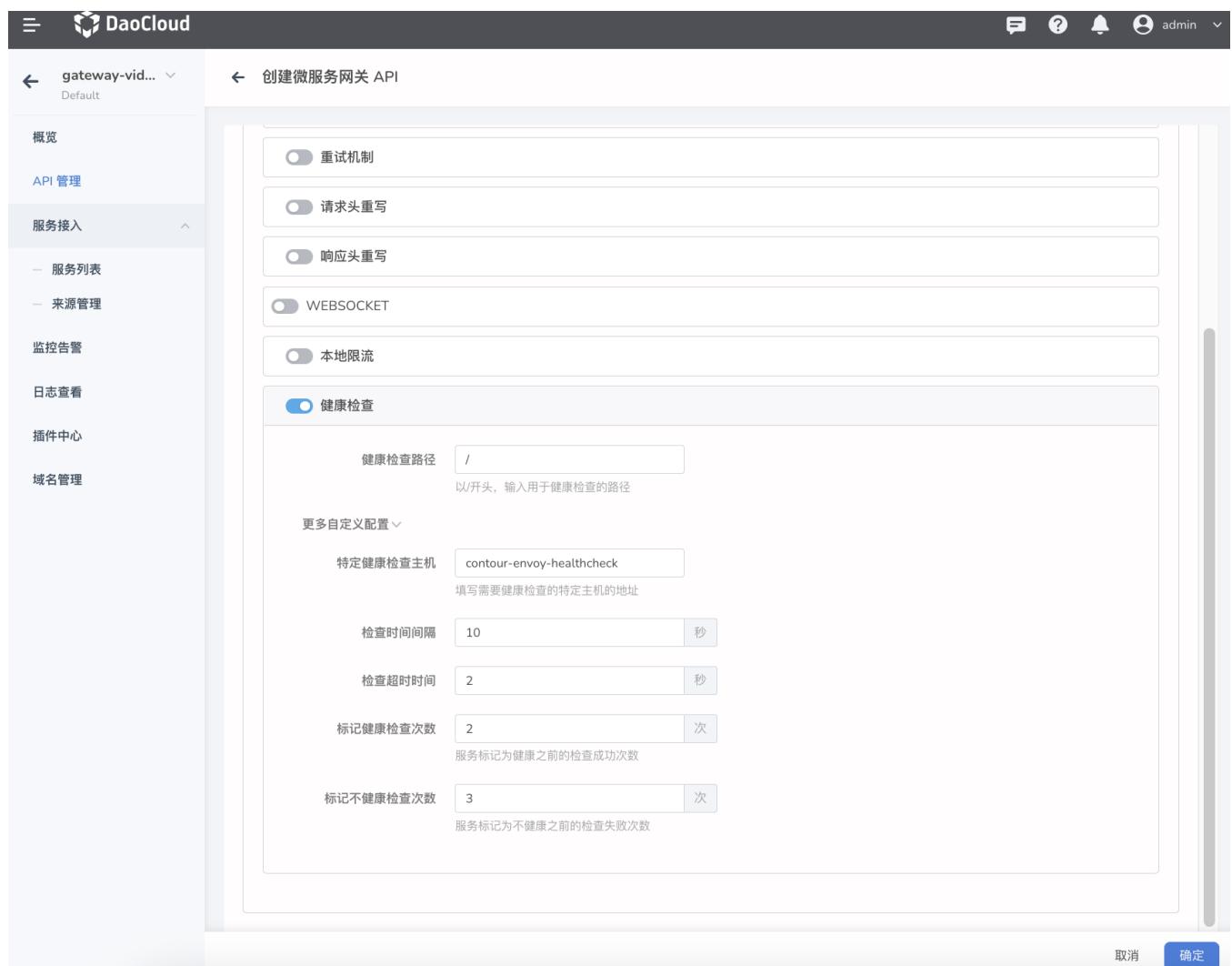
!!! info

除了在 API 层级的本地限流能力之外，还可以通过[配置域名策略] (./domain/domain-policy.md)针对整个域名进行限流处理。当 API 与域名同时配置限流策略时，以 API 层级的限流策略为准。

健康检查

通过设置健康检查地址，可以有效保证当后端服务异常时，网关自动进行负载均衡调整。对不健康的后端服务进行标记，停止向该服务分发流量。当后端服务恢复并通过设定的健康检查条件后，自动恢复流量分发。

- 健康检查路径：以“/”开头，并且全部后端服务的所有实例都应提供相同的健康检查接口。
- 特定健康检查主机：配置主机地址后，仅对该主机进行健康检查。
- 检查时间间隔：每次健康检查的时间间隔，时间单位为“秒”，例如每隔 10 秒进行一次健康检查。
- 检查超时时间：健康检查的最大超时时长，当健康检查超过配置的时长时，直接标记健康检查失败。
- 标记健康检查次数：连续检查 N 次并且每次结果都是健康时，才将服务实例标记为健康状态；当服务实例被标记为健康状态后，请求流量将会自动分发到该服务实例。
- 标记不健康检查次数：连续检查 N 次并且每次结果都是不健康时，就将服务实例标记为不健康状态，当服务实例被标记为不健康时，停止向该实例分发请求流量。



The screenshot shows the configuration page for a service named "gateway-vid...". On the left sidebar, under "API Management", "Service Access", and "Health Monitoring", the "Health Check" section is selected. The main configuration area contains the following settings:

- 重试机制 (Retry Mechanism):** Off.
- 请求头重写 (Request Header Rewrite):** Off.
- 响应头重写 (Response Header Rewrite):** Off.
- WEBSOCKET:** Off.
- 本地限流 (Local Flow Control):** Off.
- 健康检查 (Health Check):** On.

健康检查路径 (Health Check Path): /

更多自定义配置 (More Custom Configuration):

- 特定健康检查主机 (Specific Health Check Host):** contour-envoy-healthcheck
- 检查时间间隔 (Check Interval):** 10 秒
- 检查超时时间 (Check Timeout):** 2 秒
- 标记健康检查次数 (Mark Healthy Check Times):** 2 次
- 标记不健康检查次数 (Mark Unhealthy Check Times):** 3 次

At the bottom right are "取消 (Cancel)" and "确定 (Confirm)" buttons.

移除 API

微服务网关支持对网关实例的 API 进行全生命周期管理，包括 API 的添加、更新和移除。本页介绍如何移除 API。

可以通过两种方式移除 API。

!!!danger

注意：删除操作是不可逆的。无论 API 是否处于在线状态，删除后均立即失效并且不可恢复。

- 在 API 管理 页面找到需要删除的 API，在该 API 的右侧点击⋮并选择移除。

- 点击 API 名称进入 API 详情页，在页面右上角点击⋮操并选择移除。

更新 API

微服务网关支持对网关实例的 API 进行全生命周期管理，包括 API 的添加、更新和删除。本页介绍如何更新 API。

可以通过两种方式更新 API 的基础配置和策略配置。

- 在 API 管理 页面找到需要更新的 API，在该 API 的右侧点击 选择 修改基础配置 或 修改策略。

- 点击 API 名称进入 API 详情页，在页面右上角点击 编辑 更新基本信息，点击 策略管理 更新策略。

3.3 Domain

添加域名

微服务网关支持对统一托管的域名进行全生命周期管理，包括添加、更新、删除域名域名。通过域名管理，可以将一个域名应用到网关内的多个 API，并且可以配置域名层级的网关策略。本页介绍如何添加域名。

添加域名的步骤如下：

1. 点击目标网关的名称进入网关概览页面，然后在左侧导航栏点击 域名管理，在页面右上角点击 添加域名。

The screenshot shows the DaoCloud gateway management interface. The top navigation bar includes the DaoCloud logo, a message icon, a question mark icon, a bell icon, and a user account section for 'admin'. On the far right, there is a link to '添加域名' (Add Domain). The left sidebar has a tree view with 'gateway-video-dev' expanded, showing 'Default' and several sub-options: 概览 (Overview), API 管理 (API Management), 服务接入 (Service Access), 监控告警 (Monitoring & Alarming), 日志查看 (Log Viewing), 插件中心 (Plugin Center), and 域名管理 (Domain Management), which is currently selected and highlighted in blue. The main content area is titled '微服务网关: gateway-video-dev / 域名管理' (Microservices Gateway: gateway-video-dev / Domain Management). It features a search bar and a table with the following data:

域名	协议	证书	命名空间	最近更新时间	操作
ad.service.virtualhost	HTTP		skoala-video	2022-10-30 11:24	⋮

Below the table, it says '共 1 项' (1 item) and shows page navigation '1 / 1' and a dropdown for '10 项' (10 items).

2. 填写配置信息

域名的配置信息分为 [基础信息](#)（必填）和 [策略配置](#)（可选）两部分。

- 域名：域名创建后不可以修改。
- 协议：默认选择 HTTP。如果选择 HTTPS，需要提供对应的 HTTPS 证书。

目前仅支持选取已经存在的证书，自动签发证书和手动上传证书功能正在开发。

基础信息

域名 * daocloud.io

协议 * HTTP HTTPS

HTTPS 证书 * 选取已存在证书 自动签发 手工上传

选取命名空间 * skoala-test

选取证书 * [dropdown menu]

仅支持选取 默认 (Opaque) 和 TLS 类型，且加密位数不少于 2048 位

证书版本 * v1.2 v1.3

取消 确定

- 本地限流：参考[本地限流](#)
- 跨域：参考[跨域](#)

The screenshot shows the 'Create Domain' page in the DaoCloud interface. On the left, a sidebar lists 'gateway-video' (Default), '概览', 'API 管理', '服务接入', '监控告警', '日志查看', '插件中心', and '域名管理'. The main area has two tabs: '基础信息' and '策略配置'. Under '基础信息', the domain is set to 'daocloud.io' and the protocol is 'HTTP'. Under '策略配置', there are two sections: '本地限流' (Local Flow Control) and '跨域' (Cross-Origin). In '本地限流', settings include '请求速率' (Request Rate) at 2, '时间窗口' (Time Window) in seconds, and '允许溢出速率' (Allow Overflow Rate) at 0. In '跨域', '启用凭证' (Enable Credential) is turned on, and '允许请求方法' (Allow Request Methods) includes 'GET'. At the bottom right are '取消' (Cancel) and '确定' (Confirm) buttons.

3. 在页面右下角点击 确定

点击 确定 后，将自动跳转到 域名管理 页面，可以在域名列表中看到刚才新建的域名。

The screenshot shows the 'Domain Management' page in the DaoCloud interface. The sidebar includes 'michelle.video' (Default), '概览', 'API 管理', '服务接入', '监控告警', '日志查看', '插件中心', and '域名管理'. The main content area displays a table of domains:

域名	协议	证书	命名空间	最近更新时间	操作
daocloud.io	HTTP		skoala-michelle	2022-09-15 14:20	⋮
michelle.virtualhost	HTTP		skoala-michelle	2022-09-15 11:39	⋮

At the bottom, it says '共 2 项' (2 items total) and shows page navigation. The 'daocloud.io' row is highlighted with a red box.

删除域名

微服务网关支持对统一托管的域名进行全生命周期管理，包括添加、更新、删除域名。通过域名管理，可以将一个域名应用到网关内的多个 API，并且可以配置域名层级的网关策略。

!!! danger

- 正在被 API 使用的域名无法删除，需要先删除相关的 API 然后才能删除域名。
- 域名删除后无法恢复。

可以通过两种方式删除域名。

- 在 域名管理 页面找到需要删除的域名，在点击 并选择 **删除**。

The screenshot shows the DaoCloud domain management interface. On the left is a sidebar with navigation items: 概览, API 管理, 服务接入, 监控告警, 日志查看, 插件中心, and 域名管理. The main area displays a table of domains. One row is selected, showing details: 域名 ad.service.virtualhost, 协议 HTTP, 命名空间 skoala-video, 最近更新时间 2022-10-30 11:24. Below the table are two buttons: 编辑基础信息 and 编辑策略配置. In the bottom right corner of the card, there is a red-bordered 'Delete' button.

- 点击域名名称进入域名的详情页，在页面右上角点击 操并选择 **删除**。

The screenshot shows the detailed view of the domain 'ad.service.virtualhost'. The top right corner features three buttons: 修改基础配置, 修改策略, and a red-bordered 'Delete' button. The page is divided into sections: 基础信息 (Domain Information) which lists the domain name, protocol, last update time, and creation time; 域名信息 (Domain Information); 证书信息 (Certificate Information); and 策略配置 (Policy Configuration) which includes local traffic control settings like 请求速率 (Request Rate), 时间窗口 (Time Window), 允许溢出速率 (Overflow Rate), and 限制返回码 (Return Code Limit). At the bottom is a table for Header settings.

如果域名正在被某个 API 使用，需要页面提示点击 **查看服务详情** 去删除对应的 API。

确定删除资源

X

当前域名： ad.service.virtualhost 正在被API 使用中，请先在对应的 API 内解绑服务。

取消

查看服务详情

配置域名策略

微服务网关提供了域级别的策略配置能力。配置域级别的策略之后，无需对同一域名下的多个 API 重复配置策略。目前支持两种域级别的策略：跨域和本地限流。

有两种方式可以配置域名策略：

- 在创建域名的过程中设置策略，参考[添加域名](#)。
- 在域名创建完成后通过[修改域名](#)进行调整。

对跨域和本地限流策略的详细说明如下：

本地限流

为域名配置本地限流之后，该配置会自动应用到使用此域名的所有 API。

有关本地限流的详细配置说明，可参考[本地限流](#)。

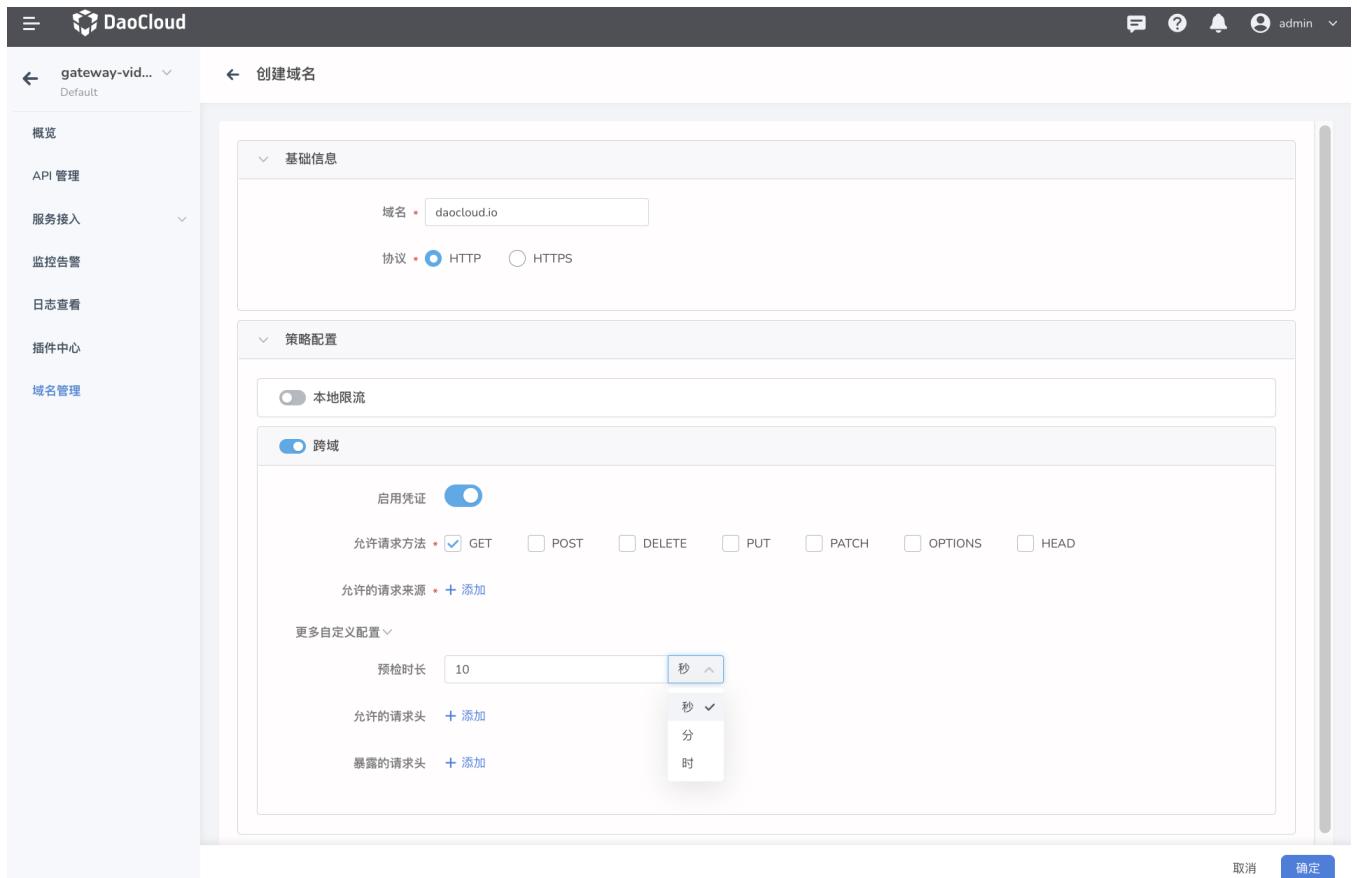
!!! note

如果 API 层级与域名层级的限流策略冲突时，以 API 层级的限流策略为准。

跨域

填写配置时需要注意：

- 启用凭证：开启后，需要对跨域请求进行凭证检查。检查通过之后，才能开始处理跨域请求。
- 允许的请求方法：选择 HTTP 协议的请求方式。有关各种请求方式的详细说明，可参考 W3C 的官方文档[方式定义](#)。
- 允许的请求来源：限定多个特定的请求来源，通常使用 IP。
- 预检时长：在处理跨域请求之前检查凭证、请求方法等事项所用的时间，时间单位为秒、分、时。
- 允许的请求头：限定特定的 HTTP 请求头关键字。添加关键字后，需在请求头中加上对应的关键字才能正常访问目标服务。
- 暴露的请求头：控制暴露的请求头关键字，可以配置多项。



The screenshot shows the 'Create Domain' configuration interface in the DaoCloud API Gateway. The main configuration pane is titled '基础信息' (Basic Information) and contains fields for 'Domain' (daocloud.io) and 'Protocol' (HTTP). Below this is the '策略配置' (Policy Configuration) section, which has a 'Cross-Origin' toggle switch turned on. Under this, the 'Enable Credential' toggle is also turned on. A list of allowed request methods is shown, with 'GET' checked and others like 'POST', 'DELETE', etc., unselected. There are sections for 'Allowing Request Sources' (with a '+ Add' button), 'More Custom Configuration' (with a 'Precheck Duration' field set to 10 seconds), and 'Exposed Request Headers' (with a '+ Add' button). At the bottom right of the configuration pane are 'Cancel' and 'Confirm' buttons.

更新域名

微服务网关支持对统一托管的域名进行全生命周期管理，包括添加、更新、删除域名。通过域名管理，可以将一个域名应用到网关内的多个 API，并且可以配置域名层级的网关策略。

可以通过两种方式修改域名的基础信息和策略配置。

- 在 域名管理 页面找到需要更新的域名，在右侧点击 选择 编辑基础信息 或 编辑策略配置。

域名	协议	证书	命名空间	最近更新时间
ad.service.virtualhost	HTTP		skoala-video	2022-10-30 11:24

- 点击域名名称进入域名详情页，在页面右上角点击 修改基础配置 更新基本信息，点击 修改策略 更新策略。

键	值
暂无数据	

3.4 Gateway

查看网关详情

您可以在概览页面查看网关的详细信息，包括名称、部署位置、网关运行状态、服务入口方式、控制节点/工作节点的健康状态、API、插件等信息。

进入网关详情页面

在网关列表页面，选择要查看的网关实例，点击…在下拉列表中选择网关详情。

网关详情

网关详情分为：基本信息、网络信息、网关数据、资源情况、资源负载、插件信息等部分。

部分数据的说明如下：

- 手工服务接入数：指在服务列表页面通过添加服务手工添加到当前网关的服务数量。
- 自动服务接入数：指在服务列表页面通过纳管服务自动添加到当前网关的服务数量。
- 域名管理：指当前网关下的域名数量。
- API 数量：指当前网关中使用的 API 数量。
- 控制/工作节点实例数：显示节点实例的总数和健康状况。/左侧的数字表示当前在线的实例数，右侧的数字表示节点实例总数。
 - 如果在线节点/左侧数字等于全部应在线节点/右侧数字，则显示为绿色。
 - 如果在线节点/左侧数字小于全部应在线节点/右侧数字，则显示为红色。
- Top 10 APIs：按照 API 响应代码 2xx、4xx、5xx 的数量由高到低排列的前 10 个 API，默认按照响应代码 200 的数量降序排列。可查看过去 15 分钟、过去 1 小时、过去 24 小时的数据。
- 资源负载：以折线图的形式显示网关控制节点和工作节点在过去 1 小时内或过去 3 小时内的 CPU 使用情况和 Memory 使用情况。
- 插件信息：显示当前网关的插件启停情况以及其他可用插件。

相关操作

除了查看网关详情，您还可以通过网关详情页面进行更新网关配置、删除网关、诊断网关等操作。

- 更新网关：在页面顶部点击编辑可以跳转到更新网关配置的页面，具体步骤参见[更新网关配置](#)。
- 删除网关：在页面顶部点击…后选择删除可以跳转到删除网关的页面，具体步骤参见[删除网关](#)。
- 诊断网关：在页面顶部点击诊断模式可以进入网关调试模式，具体步骤参见[诊断网关](#)。
- 接入服务：在“网关数据”部分点击“手工/自动服务接入数”可以进入服务接入页面，具体步骤参见[手动接入服务或自动纳管服务](#)。
- 管理 API：在“网关数据”部分点击“API 数量”可以进入 API 列表，进行增删改查等操作。
- 管理域名：在“网关数据”部分点击“域名管理”可以进入域名列表，进行增删改查等操作。

创建微服务网关

微服务网关支持多租户实例的高可用架构，兼容多种模式微服务的统一网关接入能力。本页介绍如何创建微服务网关实例。

创建微服务网关的步骤如下：

- 在左侧导航栏点击 微服务网关，在 微服务网关列表 页的右上角点击 创建网关，进入微服务网关的创建页面。

The screenshot shows the DaoCloud platform's Microservices Gateway List page. The left sidebar has a navigation bar with 'Default' selected, '注册中心' (Registry Center), and '微服务网关' (Microservices Gateway) highlighted. The main area is titled '微服务网关列表' (Microservices Gateway List). It lists three gateways:

- test** (运行中): 部署集群 kpanda-global-cluster, 命名空间 default. Created at 2022-09-05 20:10, last updated at 2022-09-05 20:10. Access address: http://hello.baidu.com:30700. Control nodes: 3/3. Work nodes: 3/3.
- vide** (运行中): 部署集群 th1, 命名空间 th-demo. Created at 2022-09-07 11:23, last updated at 2022-09-07 11:23. Access address: -. Control nodes: 3/3. Work nodes: 3/3.
- gateway** (运行中): 部署集群 skoala-stage, 命名空间 skoala-gateway. Created at 2022-09-09 10:01, last updated at 2022-09-13 10:05. Access address: http://gateway.virtualhost:62810. Control nodes: 1/1. Work nodes: 2/2.

- 填写配置

网关配置分为基本配置（必填）和高级配置（选填）两部分。填写配置信息时需要注意以下事项：

- 网关名称：长度不超过 63 个字符，支持字母、数字、下划线，在网关创建完成后不可更改。
- 部署位置：一个命名空间中只能部署一个网关。
- 管辖命名空间：默认管辖网关所在的命名空间，支持同时管辖多个命名空间，同一个命名空间不能被两个网关同时管辖。
- 控制节点数：单副本存在不稳定性，需谨慎选择。

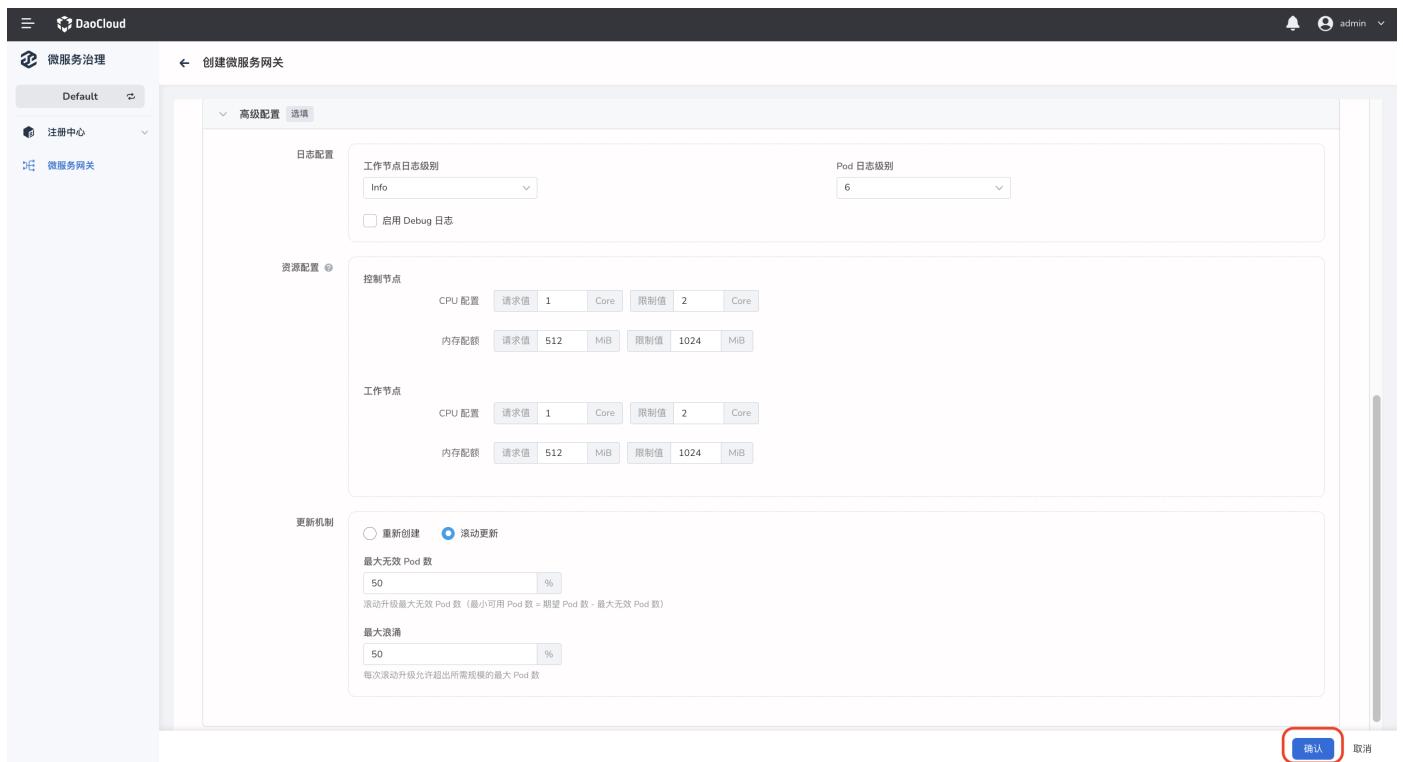
Basic Configuration (必填)

- 网关名称: michelle
- 部署位置: skoala-stage
- 管辖命名空间: skoala-demo
- 服务入口方式:
 - 节点访问
 - 负载均衡 启用 HTTPS
- 资源配置:
 - 控制节点数: 3 副本
 - 工作节点数: 3
- 组件版本依赖:
 - 控制组件: docker.io/bitnami/contour:1.22.0-debian-11-r2
 - 网关组件: docker.io/bitnami/envoy:1.23.0-debian-11-r2

Advanced Configuration (选填)

- 日志配置:
 - 工作节点日志级别: Info
 - Pod 日志级别: 6
 - 启用 Debug 日志
- 资源配置:
 - 控制节点:
 - CPU 配置: 请求值 1 Core, 限制值 2 Core
 - 内存配置: 请求值 512 MiB, 限制值 1024 MiB
 - 工作节点:
 - CPU 配置: 请求值 1 Core, 限制值 2 Core
 - 内存配置: 请求值 512 MiB, 限制值 1024 MiB
- 更新机制:
 - 重新创建
 - 滚动更新
 - 最大无效 Pod 数: 50 %
 - 最大浪涌: 50 %
 - 每次滚动升级允许超出所需规模的最大 Pod 数

3. 确认所填信息，在页面右下角点击 **确认**。



点击 [确认](#) 后，右上角弹出网关创建成功的消息。系统会自动返回 [微服务网关列表](#) 页，可以在列表中看到新创建的网关实例。

删除微服务网关

微服务网关支持多租户实例的高可用架构，兼容多种模式微服务的统一网关接入能力。本页介绍如何删除微服务网关实例。

删除网关同样也有两种方式。为保证服务不受影响，删除网关之前需要释放到网关中全部路由的 API。

!!!danger

网关移除后不可恢复，请谨慎操作。

- 在 [微服务网关列表](#) 页选择需要移除的网关实例，在实例右侧点击 并选择 [删除](#)。

The screenshot shows the DaoCloud service gateway list interface. On the left is a sidebar with navigation links: 微服务引擎 (Default), 微服务治理中心, 微服务网关 (selected), and 告警管理. The main area has a title '微服务网关列表'. A search bar at the top includes filters for 部署集群 (All) and 命名空间. Below the search is a table with one row for 'gateway-video'. The row details are: 部署集群: kpanda-global-cluster, 命名空间: skoala-video, 创建时间: 2022-10-26 11:06, 最近更新时间: 2022-10-26 13:57, 接入地址: http://video1.virtualhost:30040. To the right of the table are buttons for 编辑 and 删除, with the 'Delete' button highlighted by a red box.

- 点击网关名称进入进入概览页面后，在右上角 并选择 [删除](#)。

The screenshot shows the DaoCloud service gateway detail interface for 'gateway-video'. The left sidebar has links: 概览 (selected), API 管理, 服务接入, 监控告警, 日志查看, 插件中心, and 域名管理. The main area has a title '微服务网关: gateway-video / 概览'. It contains several sections: '基本信息' (Gateway Name: gateway-video, Deployment Location: kpanda-global-cluster/skoala-video, Namespace: skoala-video, Status: Running), '网络信息' (NodePort: http://video1.virtualhost:30040, Access Address: http://video1.virtualhost:30040), '资源情况' (Control Node Instance Count: 2/2, Work Node Instance Count: 1/2, Status: Abnormal), and '资源负载' (Resource Load: No Data). The top right corner of the main area has buttons for '诊断模式', '编辑', and a red-highlighted '删除' (Delete) button.

诊断网关

使用诊断模式可以根据所填参数对网关进行调试，便于您测试网关配置的可用性以及查找故障所在。

诊断网关的步骤如下：

进入诊断模式

在【网关列表】页面选择需要诊断的网关，点击【…】在下拉列表中选择【网关详情】，进入网关详情页面后点击右上角的【诊断模式】。

输入诊断参数并执行

根据不同的场景输入相关参数，点击【执行】，即可看到当前配置的调试结果。

- 如需修改参数重新调试，关闭【诊断模式】开关按钮，重新输入参数并点击执行即可。
- 点击底部的【下载日志】可以下载当前配置的诊断结果。

退出诊断模式

点击【】即可退出诊断模式。

更新微服务网关

微服务网关支持多租户实例的高可用架构，兼容多种模式微服务的统一网关接入能力。本页介绍如何更新微服务网关实例。

更新网关配置

有两种方式可以更新网关配置。

- 在 [微服务网关列表](#) 页选择需要更新的网关实例，在实例右侧点击 [...](#) 并选择 [编辑](#)。

网关名称	部署集群	命名空间	最近更新时间	控制节点实例数	工作节点实例数
test	kpanda-global-cluster	default	2022-09-05 20:10	3 / 3	3 / 3
vide	th1	th-demo	2022-09-07 11:23	3 / 3	3 / 3
gateway	skoala-stage	skoala-gateway	2022-09-09 10:01	1 / 1	2 / 2

- 点击网关名称进入进入概览页面后，在右上角点击 [编辑](#)。

基本信息	网络信息
网关名称: gateway-video 部署位置: kpanda-global-cluster/skoala-video 管辖命名空间: skoala-video 服务状态: 运行中 最近更新时间: 2022-10-26 13:57 配置更新方式: 重新创建 创建时间: 2022-10-26 11:06 Envoy 版本: Contour 版本 Envoy 日志级别: STATUS_UNSPECIFIED	服务入口方式: NodePort 访问地址: http://video1.virtualhost:30040 控制节点实例数: 2 / 2 工作节点实例数: 1 / 2 资源情况: 2 / 2 资源负载: 1 / 2

3.5 Logs

实例日志

微服务网关支持查看请求日志和实例日志。本页介绍如何查看实例日志以及查看日志时的相关操作。

查看方式

点击目标网关的名称，进入网关概览页面，然后在左侧导航栏点击 日志查看 -> 实例日志。

The screenshot shows the DaoCloud web interface for managing microservices. On the left, there's a sidebar with various navigation options like 'Overview', 'API Management', 'Service Access', 'Monitoring & Alerts', 'Log View' (which is currently selected and highlighted in blue), and 'Plugin Center'. The main content area is titled 'Microservice Gateway: gateway-video-dev / Log View'. Below the title, there are two tabs: 'Request Log' and 'Instance Log', with 'Instance Log' being the active one. The main pane displays a message '暂无数据' (No data available) with a small envelope icon.

相关操作

- 筛选日志：支持筛选实例仅查看某个容器组的日志，也可以参考 [KQL 语法](#)直接输入关键字查找特定内容。
- 限定时间范围：日志时间范围可以选择：近 5 分钟的日志、近 1 小时、近 12 小时、近 7 天的日志，或者自定义时间范围。
- 自动刷新：在查看日志时，开启自动刷新可以查看实时日志。

!!! info

如需查看更多日志或下载日志，可前往[\[可观测性模块\]](#)(../../../../insight/intro/what.md)通过[\[日志查询\]](#)(../../../../insight/user-guide/data-query/log.md)功能查询或下载特定集群、命名空间、工作负载、容器组的日志。

请求日志

微服务网关支持查看请求日志和实例日志。本页介绍如何查看实例日志以及查看日志时的相关操作。

查看方式

点击目标网关的名称，进入网关概览页面，然后在左侧导航栏点击 [日志查看](#) -> [请求日志](#)。

相关操作

- 筛选日志：支持通过 Request ID、请求路径、域名、请求方法、HTTP、GRPC 等条件筛选日志，支持按照请求开始时间、请求耗时、请求服务耗时对日志进行排序。

- 限定时间范围：可选择近 5 分钟、10 分钟、15 分钟的日志，或者自定义时间范围。

请求日志

实例日志

搜索

2022-11-02 14:02:07 - 2022-11-02 14:07:07

导出日志

请求开始时间 Request ID

近5分钟 近10分钟 近15分钟

2022年11月 2022年12月

日	一	二	三	四	五	六	日	一	二	三	四	五	六
30	31	1	2	3	4	5	27	28	29	30	1	2	3
6	7	8	9	10	11	12	4	5	6	7	8	9	10
13	14	15	16	17	18	19	11	12	13	14	15	16	17
20	21	22	23	24	25	26	18	19	20	21	22	23	24
27	28	29	30	1	2	3	25	26	27	28	29	30	31
4	5	6	7	8	9	10	1	2	3	4	5	6	7

清空 确定

- 导出日志：支持将日志文件导出到本地。

请求日志

实例日志

搜索

导出日志

请求开始时间 Request ID 域名 请求路径 请求方法 响应状态 请求耗时 请求服务耗时

暂无数据

3.6 Service

自动纳管服务

添加成功的服务会出现在服务列表页面，添加 API 时也可以选择列表中的服务作为目标后端服务。微服务网关支持通过手动接入和自动纳管两种方式添加服务。本页介绍如何自动纳管服务。

网关实例创建成功之后，服务来源中的服务会被自动添加到该网关实例的服务列表中，无需手动添加。

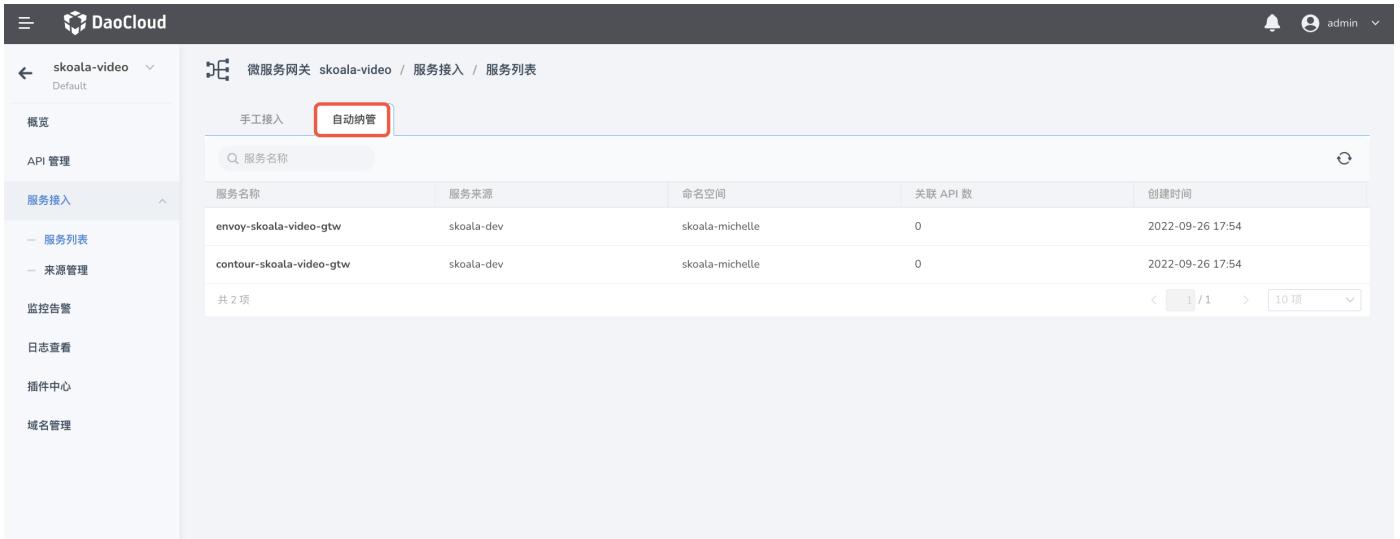
查看自动纳管的服务

- 在微服务网关列表页面点击目标网关的名称，进入网关概览页后，在左侧导航栏点击 服务接入 → 服务列表。



The screenshot shows the DaoCloud service gateway interface. On the left, there's a sidebar with navigation links: 'Default', 'Overview', 'API Management', 'Service Access' (with 'Service List' highlighted by a red box), 'Source Management', 'Monitoring & Alerts', 'Log View', 'Plugin Center', and 'Domain Management'. The main content area is titled 'Microservices Gateway test / Service Access / Service List'. It has tabs for 'Manual Connection' and 'Automatic Management' (which is selected). Below the tabs is a search bar with placeholder 'Service Name'. A table header includes columns for 'Service Name', 'Connection Information', 'Service Source', 'Namespace', 'Associated API Count', and 'Creation Time'. The table body is empty, showing 'No data found'.

- 在服务列表页面点击 自动纳管。



This screenshot shows the 'skoala-video' gateway service list page. The left sidebar is identical to the previous one. The main content area is titled 'Microservices Gateway skoala-video / Service Access / Service List'. The 'Automatic Management' tab is selected. The table lists two services: 'envoy-skoala-video-gtw' and 'contour-skoala-video-gtw', both from 'skoala-dev' with 'skoala-michelle' namespace, 0 associated APIs, and creation times of 2022-09-26 17:54. Navigation controls at the bottom show page 1 of 1 and 10 items per page.

配置服务策略

- 在服务列表 → 自动纳管 页面找到目标服务，在右侧点击  选择 策略配置。

The screenshot shows the DaoCloud service mesh interface. The top navigation bar includes the DaoCloud logo, user information (admin), and a message center. The main title is "微服务网关: gateway-video-dev / 服务接入 / 服务列表". On the left sidebar, under the "服务接入" section, the "服务列表" item is highlighted with a red box. The top navigation bar has two tabs: "手工接入" and "自动纳管", with "自动纳管" also highlighted with a red box. The main content area displays a table of services with the following columns: 服务名称 (Service Name), 服务来源 (Service Source), 命名空间 (Namespace), 关联 API 数 (Associated API Count), and 创建时间 (Creation Time). The table contains five entries:

服务名称	服务来源	命名空间	关联 API 数	创建时间
contour-gateway-vid...	skoala-dev	-	0	2022-10-27 18:45
envoy-gateway-video...	skoala-dev	-	0	策略配置
contour-gateway-vid...	skoala-dev	-	0	2022-10-27 18:45
envoy-gateway-video...	skoala-dev	-	0	2022-10-27 18:45
grafana	skoala-dev	-	0	2022-10-27 18:45

At the bottom, it says "共 5 项" (5 items) and shows page navigation: < 1 / 1 >. A dropdown menu indicates "10 项" (10 items).

2. 按需调整服务策略配置，在弹框右下角点击 确定。

- HTTPS 证书验证：开启后，必须通过证书校验才能成功访问该服务。
- 服务熔断：当最大连接数、最大处理连接数、最大并行请求数、最大并行重试数 任何一个 指标达到设定的阈值时，自动切断对该服务的调用，保护系统整体的可用性。当指标降到设定的阈值之后，自动恢复对该服务的调用。

编辑contour-gateway-video-dev-gtw



服务来源类型 集群服务

服务名称 contour-gateway-video-dev-gtw

连接信息 http://contour-gateway-video-dev-gtw.skoala-video:8001

HTTPS 证书验证 启用

HTTPS 端口 8001

访问证书选择 contour-gateway-video-dev-gt...

主题备用名称 (SAN) kubernetes

服务熔断 启用

最大连接数 0

最大处理连接数 0

最大并行请求数 0

最大并行重试数 0

取消

确定

查看服务详情

1. 在 服务列表 -> 自动纳管 页面找到目标服务，点击服务名称。

微服务网关：gateway-video-dev / 服务接入 / 服务列表

服务名称	服务来源	命名空间	关联 API 数	创建时间
contour-gateway-video-gtw	skoala-dev	-	0	2022-10-27 18:45
contour-gateway-video-vid...	skoala-dev	-	0	2022-10-27 18:45
envoy-gateway-video...	skoala-dev	-	0	2022-10-27 18:45
grafana	skoala-dev	-	0	2022-10-27 18:45

2. 查看服务名称、来源、关联的 API 等信息。支持安装 最近更新时间 进行排序。

微服务网关：gateway-video-dev / 服务接入 / 服务详情

基本信息

contour-gateway-video-dev-gtw	skoala-dev	2022-10-27 18:45
服务名称	服务来源	创建时间

关联 API 列表

API 名称	关联域名	请求方法	请求路径	目标服务类型	状态	最近更新时间
暂无数据						

!!! info

对于自动纳管的服务，仅支持上述操作，不支持更新、删除服务的操作。

手动接入服务

添加成功的服务会出现在服务列表页面，添加 API 时也可以选择列表中的服务作为目标后端服务。微服务网关支持通过手动接入和自动发现两种方式添加服务。本页介绍如何手动接入服务。

前提条件

需要事先在来源管理中添加对应的服务来源，才能在手动接入服务时选择对应的服务来源类型。

接入服务

- 在 微服务网关列表 页面点击目标网关的名称，进入网关概览页后，在左侧导航栏点击 服务接入 --> 服务列表。



The screenshot shows the DaoCloud service gateway interface. At the top, there's a header with the DaoCloud logo and a user account. Below it is a navigation bar with tabs for 'test' and 'Default'. The main content area has a breadcrumb path: '微服务网关 test / 服务接入 / 服务列表'. There are two tabs at the top of the list table: '手工接入' (selected) and '自动纳管'. Below the tabs is a search bar with placeholder '服务名称'. The table has columns: '服务名称', '连接信息', '服务来源', '命名空间', '关联 API 数', and '创建时间'. A blue button labeled '添加服务' is located at the bottom right of the table. On the left side, there's a sidebar with sections: '概览', 'API 管理', '服务接入' (with '服务列表' highlighted by a red box), '来源管理', '监控告警', '日志查看', '插件中心', and '域名管理'.

- 在 服务列表 页面点击 手工接入 --> 添加服务。



This screenshot is identical to the one above, showing the 'Service List' page under 'Service Access'. The 'Manual Access' tab is selected. A red box highlights the 'Add Service' button in the top right corner of the table. The rest of the interface, including the sidebar and table structure, is the same as the previous screenshot.

3. 选择服务来源，配置服务连接信息，点击确定。

- 集群服务：选择目标服务所在的集群和命名空间，填写访问协议、地址以及端口。

选择服务 *

服务名称	命名空间	标签
myapp-service	hyh-system	argocd.argoproj.io/instance=888
test-svc	hyh-system	app=test-svc
test-dao-2048-replicas	hyh-test	app=test-dao-2048-replicas
test-rollout	hyh-test	app=test-rollout

连接信息 *

协议	HTTP
地址	10.108.251.209
端口	8080

取消 确定

对于集群服务的访问方式，可在 容器管理 -> 容器网络 -> 服务 中点击服务名称进行查看：

The screenshot shows the DaoCloud platform interface for managing services. The top navigation bar includes the DaoCloud logo, user information (admin), and a '手动接入服务' (Manually Integrate Service) button. The main content area displays a service named 'test-svc' within a cluster ('amamba-dev') and namespace ('hyh-system'). Key details shown include:

- 基本信息 (Basic Information):** Shows the service name 'test-svc', cluster IP '10.108.251.209/-', port '80', and creation time '2022-10-27 01:05'.
- 服务端口 (Service Port):** Lists the protocol as 'TCP', port as '80', and the service port as '8080'.
- 关联容器组 (Associated Container Groups):** A table lists a single container group named 'test-svc-7d56c858...', which is currently in a '等待中' (Waiting) state.
- 事件 (Events):** A table indicates '暂无数据' (No data available).

The left sidebar navigation includes sections like '集群概览' (Cluster Overview), '节点管理' (Node Management), '工作负载' (Workload), '无状态负载' (Stateless Load), '有状态负载' (Stateful Load), '守护进程' (Daemon), '任务' (Task), '定时任务' (Scheduled Task), '容器组' (Container Group), 'Helm 应用' (Helm Application), '容器网络' (Container Network), and '配置与密钥' (Config & Keys). The '容器网络' section is expanded, and the '服务' (Service) sub-section is selected and highlighted with a red box.

- 网格服务：
接入网格服务的功能正在开发中，敬请期待。
- 注册中心服务：选择目标服务所在的注册中心，填写访问协议、地址和端口。

服务名称 * adservice

连接信息 *

协议	HTTP
地址	10.108.251.209
端口	8080

取消 确定

- 外部服务：填写服务名称、访问协议、地址、端口。

服务名称 * adservice

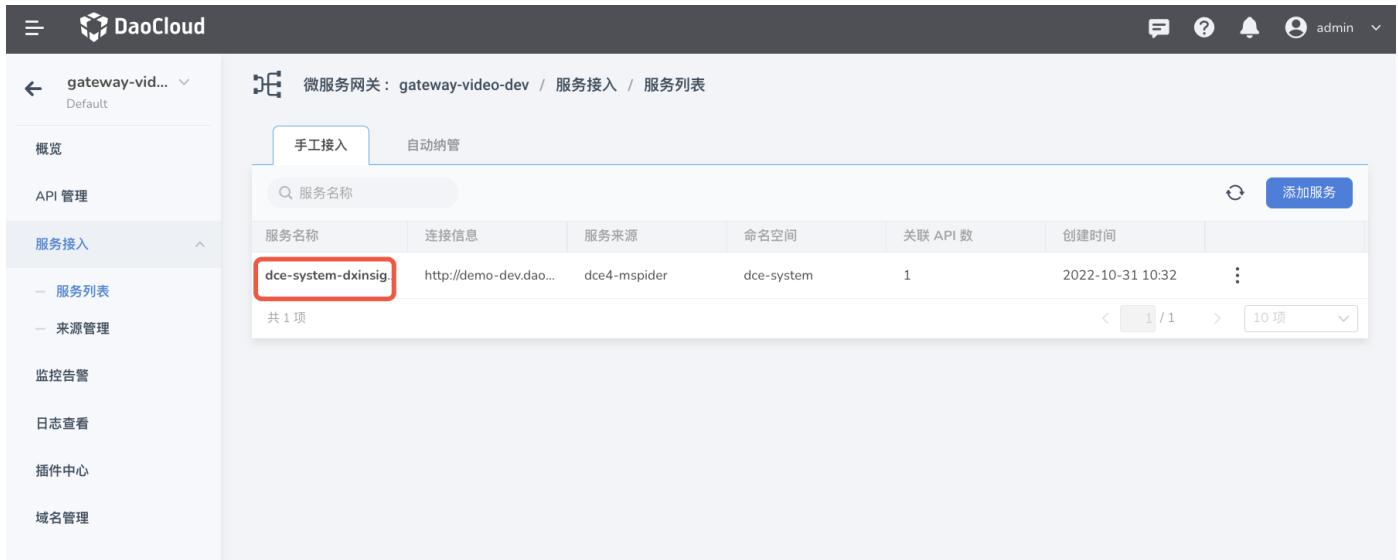
连接信息 *

协议	HTTP
地址	10.6.222.21
端口	30904

取消 确定

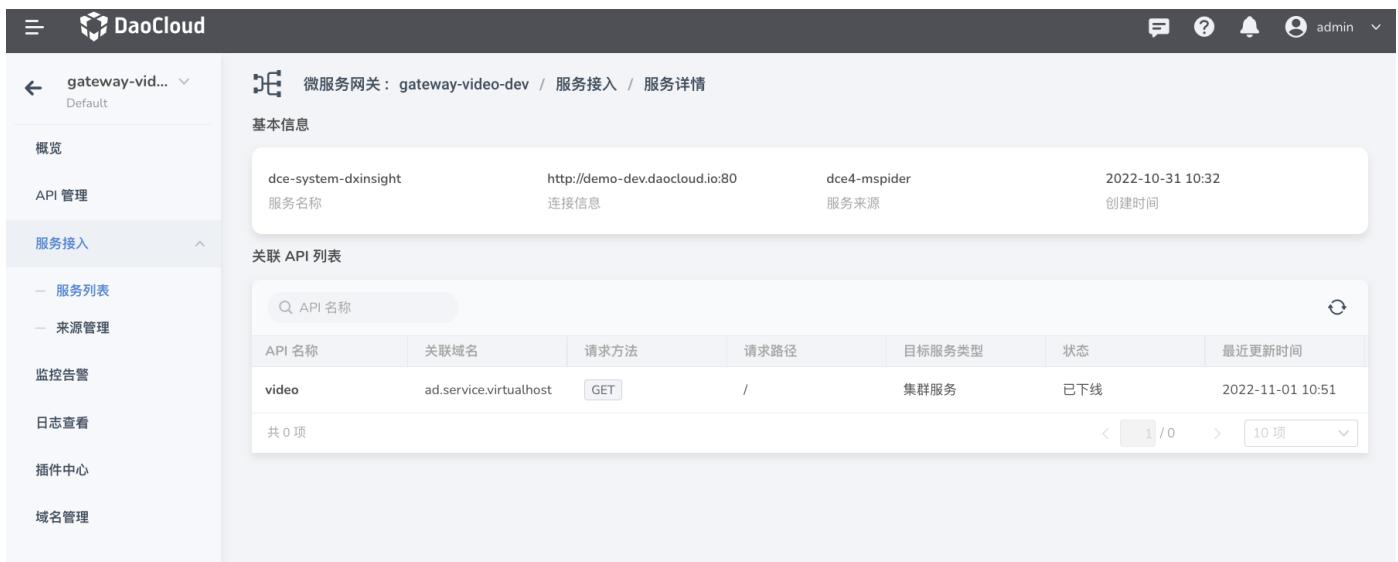
查看服务详情

- 在服务列表页面点击目标服务的名称，进入服务详情页面。



The screenshot shows the DaoCloud interface for managing services. On the left, there's a sidebar with options like 'Default', '概览', 'API 管理', '服务接入' (selected), '服务列表' (selected), '来源管理', '监控告警', '日志查看', '插件中心', and '域名管理'. The main content area is titled '微服务网关: gateway-video-dev / 服务接入 / 服务列表'. It has tabs for '手工接入' (selected) and '自动纳管'. Below is a search bar and a table with columns: 服务名称, 连接信息, 服务来源, 命名空间, 关联 API 数, 创建时间. A single row is shown: dce-system-dxinsig, http://demo-dev.dao..., dce4-mspider, dce-system, 1, 2022-10-31 10:32. The entire row is highlighted with a red box.

- 查看服务来源、连接信息、关联 API 等信息。



The screenshot shows the 'Service Details' page for the 'dce-system-dxinsig' service. The sidebar is identical to the previous screenshot. The main content area is titled '微服务网关: gateway-video-dev / 服务接入 / 服务详情'. It has a '基本信息' section with fields: 服务名称 (dce-system-dxinsig), 连接信息 (http://demo-dev.daocloud.io:80), 服务来源 (dce4-mspider), and 创建时间 (2022-10-31 10:32). Below is a '关联 API 列表' section with a table. The table has columns: API 名称, 关联域名, 请求方法, 请求路径, 目标服务类型, 状态, 最近更新时间. One entry is listed: video, ad.service.virtualhost, GET, /, 集群服务, 已下线, 2022-11-01 10:51. The entire table is highlighted with a red box.

更新服务

更新基本信息

- 在服务列表 页面找到需要更新的服务，在服务右侧点击 ，选择 基本信息。

The screenshot shows the DaoCloud service catalog interface. On the left, there's a sidebar with navigation items like 'Default', '概览', 'API 管理', '服务接入' (selected), '服务列表' (under '服务接入'), '来源管理', '监控告警', '日志查看', '插件中心', and '域名管理'. The main area displays a table titled '微服务网关: gateway-video-dev / 服务接入 / 服务列表'. The table has columns: 服务名称 (dce-system-dxinsig...), 连接信息 (http://demo-dev.dao...), 服务来源 (dce4-mspider), 命名空间 (dce-system), 关联 API 数 (1), 创建时间 (2022-10-31 10:32). A modal window is open over the table, with its top-right corner highlighted by a red box. The modal contains three buttons: '基本信息' (highlighted), '策略配置', and '删除'.

2. 更新基本信息，点击确定。

This screenshot is identical to the one above it, showing the same service catalog interface and the same modal window with the '基本信息' button highlighted by a red box.

!!! danger

如果更新基本信息时选择了其他的服务，那么原来的服务会被删除，相当于添加了一个新的服务。但原服务关联的 API 会被自动关联到新的服务。

编辑dce-system-dxinsight

服务名称	命名空间	标签
dce-system-dxinsight	dce-system	dce.daocloud.io/app=dce-
dce-system-daocloudservicemesh	dce-system	dce.daocloud.io/app=dce-
dce-system-loadbalancermanager	dce-system	dce.daocloud.io/app=dce-
dce-system-dnsservice	dce-system	dce.daocloud.io/app=dce-
dce-system-uds	dce-system	app.kubernetes.io/manage

服务名称 dce-system-daocloudservicemesh

连接信息 *

协议	HTTP
地址	demo-dev.daocloud.io
端口	80
端口范围: 1-65535	

取消 确定

更新策略配置

- 在服务列表页面找到需要更新的服务，在服务右侧点击 ，选择策略配置。

微服务网关：gateway-video-dev / 服务接入 / 服务列表

服务名称	连接信息	服务来源	命名空间	关联 API 数	创建时间	操作
dce-system-dxinsig...	http://demo-dev.dao...	dce4-mspider	dce-system	1	2022-10-31 10:32	策略配置

- 更新策略配置，点击确定。

编辑dce-system-dxinsight

X

服务来源类型 集群服务

服务名称 dce-system-dxinsight

连接信息 http://demo-dev.daocloud.io:80

HTTPS 证书验证 关闭服务熔断 启用

最大连接数 34

最大处理连接数 0

最大并行请求数 0

最大并行重试数 0

取消

确定

删除服务

在 服务列表 页面找到需要删除的服务，在服务右侧点击 ，选择 **删除**。

微服务网关: gateway-video-dev / 服务接入 / 服务列表

服务名称 连接信息 服务来源 命名空间 关联 API 数 创建时间

dce-system-dxinsig...	http://demo-dev.dao...	dce4-mspider	dce-system	1	2022-10-31 10:32
-----------------------	------------------------	--------------	------------	---	------------------

共 1 项

基本信息 策略配置

删除

删除服务之前，需要确保没有 API 正在使用该服务。如果该服务正在被某个 API 使用，需要先根据页面提示，点击 API 管理 删除关联的 API 之后才能删除该服务。

服务删除提示

当前 dce-system-dxinsight 正在被API 使用中，请先在对应的 API 内解绑服务。

取消

API 管理

4. Quickstart

4.1 安装微服务引擎

前言

本教程旨在补充需要手工安装和升级的方式。

优先通过安装器进行安装 <https://docs.daocloud.io/install/intro/>，请以官方教程文档为准。

安装器 v0.3.28 及之前

默认安装不支持；在安装规划时，可以修改 `manifest.yaml` 开启 Skoala 自动安装

```
./dce5-installer install-app -m /sample/manifest.yaml
```

安装器 v0.3.29 及之后

安装时，注意查看当前最新版本号：[去下载页查看最新版本](#)

支持默认安装 Skoala；但仍旧建议检查 `manifest.yaml`，确保 Skoala 会被安装器安装。

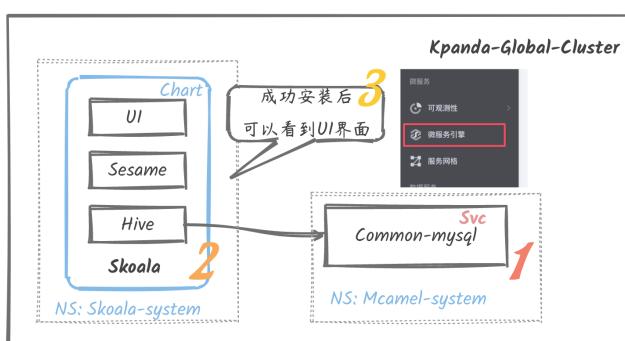
`enable` 需要为 `true`，需要指定对应的 `helmVersion`:

```
...
components:
  skoala:
    enable: true
    helmVersion: v0.12.2
  variables:
...
...
```

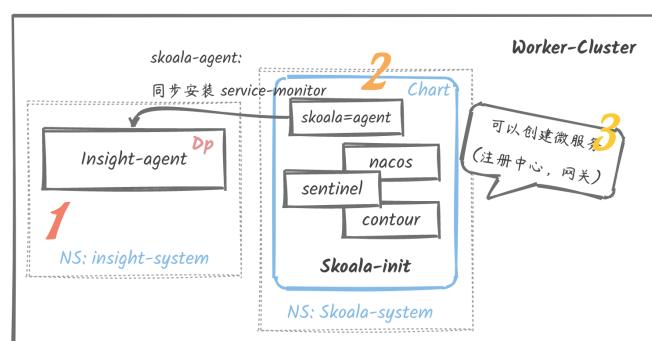
重要的话：默认安装器版本携带的是当时最新经过测试的版本；如无特殊情况，不建议更新默认的 `helm` 安装版本。

安装前检测

Skoala 部署结构



Skoala 系统模块，依赖 common-mysql



Skoala 应用模块，部署在工作集群，依赖 Insight-agent

检测 skoala 安装情况

查看 命名空间为 `skoala-system` 的之中是否有以下对应的资源，如果没有任何资源，说明 Skoala 的确没有安装。

```
~ kubectl -n skoala-system get pods
NAME          READY   STATUS    RESTARTS   AGE
hive-8548cd9b59-948j2   2/2     Running   2 (3h48m ago)   3h48m
sesame-5955c878c6-jz8cd  2/2     Running   0          3h48m
ui-7c9f5b7b67-9rpzc    2/2     Running   0          3h48m
```

NAME	NAMESPACE	REVISION	UPDATED	STATUS	CHART	APP VERSION
skoala	skoala-system	3	2022-12-16 11:17:35.187799553 +0800 CST	deployed	skoala-0.13.0	0.13.0

依赖 common-mysql 的安装情况

skoala 在安装时需要用到 mysql 来存储配置，所以必须要保证数据库存在；另外查看下 common-mysql 是否有 skoala 这个数据库。

~ kubectl -n mcamel-system get statefulset	READY	AGE
mcamel-common-mysql-cluster-mysql	2/2	7d23h

建议给到 skoala 用到的数据库信息如下：

- host: mcamel-common-mysql-cluster-mysql-master.mcamel-system.svc.cluster.local
- port: 3306
- database : skoala
- user: skoala
- password:

关于 insight-agent

Skoala 所有的监控的信息，需要依赖 Insight 的能力，则需要在集群中安装对应的 insight-agent；

集群别名	集群角色	网络模式	CPU 使用率	内存使用率	正常/节点总数
simon-cluster-longhorn	接入集群	Calico	35.8%	2.2%	3 / 3
network-sriov	工作集群	Calico +1	35.8%	2.2%	1 / 1

对 Skoala 的影响：

- 如果 skoala-init 安装时未先安装 insight-agent，不会安装 service-monitor
- 如果需要安装 service-monitor，请先安装 insight-agent，再安装 skoala-init。

安装过程

~~初始化数据库表~~

在 skoala-release/skoala 版本 v0.17.1 之后，不需要进行下方表格初始化，现已支持自动进行数据库初始化。

~~如果在 common-mysql 内的 skoala 数据库为空，请登录到 skoala 数据库后，执行以下 SQL：~~

如果提示初始化失败，请检查在 Skoala 数据库内有 3 张表，注意检测对应 SQL 是否全部是生效。

mysql> desc api;					
Field	Type	Null	Key	Default	Extra
id	bigint unsigned	NO	PRI	NULL	auto_increment

```
| is_hosted | tinyint | YES | 0 | |
| registry | varchar(50) | NO | MUL | NULL |
| service_name | varchar(200) | NO | NULL |
| nacos_namespace | varchar(200) | NO | NULL |
| nacos_group_name | varchar(200) | NO | NULL |
| data_type | varchar(100) | NO | NULL |
| detail | mediumtext | NO | NULL |
| deleted_at | timestamp | YES | NULL |
| created_at | timestamp | NO | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| updated_at | timestamp | NO | CURRENT_TIMESTAMP | DEFAULT_GENERATED on update CURRENT_TIMESTAMP |
+-----+-----+-----+-----+-----+
mysql> desc book;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id | bigint unsigned | NO | PRI | NULL | auto_increment |
| uid | varchar(32) | YES | UNI | NULL |
| name | varchar(50) | NO | UNI | NULL |
| author | varchar(32) | NO | NULL |
| status | int | YES | 1 |
| isPublished | tinyint unsigned | NO | 1 |
| publishedAt | timestamp | YES | NULL |
| deleted_at | timestamp | YES | NULL |
| createdAt | timestamp | NO | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| updatedAt | timestamp | NO | CURRENT_TIMESTAMP | DEFAULT_GENERATED on update CURRENT_TIMESTAMP |
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> desc registry;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id | bigint unsigned | NO | PRI | NULL | auto_increment |
| uid | varchar(32) | YES | UNI | NULL |
| workspace_id | varchar(50) | NO | default |
| ext_id | varchar(50) | YES | NULL |
| name | varchar(50) | NO | MUL | NULL |
| type | varchar(50) | NO | NULL |
| addresses | varchar(1000) | NO | NULL |
| namespaces | varchar(2000) | NO | NULL |
| is_hosted | tinyint | NO | 0 |
| deleted_at | timestamp | YES | NULL |
| created_at | timestamp | NO | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| updated_at | timestamp | NO | CURRENT_TIMESTAMP | DEFAULT_GENERATED on update CURRENT_TIMESTAMP |
+-----+-----+-----+-----+-----+
12 rows in set (0.00 sec)
```

配置 skoala helm repo

配置好 skoala 仓库，即可查看和获取到 skoala 的应用 chart

```
~ helm repo add skoala-release https://release.daocloud.io/chartrepo/skoala
~ helm repo update
```

需要事先安装 Helm

重点内容：增加完成后 Skoala-release 之后，常用需要关注的有 2 个 Chart：

- Skoala 是 Skoala 的控制端的服务，
- 安装完成后，可以网页看到微服务引擎的入口
- 包含 3 个组件 ui、hive、sesame
- 需要安装在全局管理集群
- Skoala-init 是 Skoala 所有的组件 Operator
- 仅安装到指定工作集群即可
- 包含组件有：skoala-agent, nacos, contour, sentinel
- 未安装时，创建注册中心和网关时会提示缺少组件

默认情况下，安装完成 skoala 到 kpanda-global-cluster(全局管理集群)，就可以在侧边栏看到对应的微服务引擎的入口了。

查看 skoala 组件 最新版本

升级部署脚本，一键部署全部组件。

在全局管理集群，查看 Skoala 的最新版本，直接通过 helm repo 来更新获取最新的；

```
~ helm repo update skoala-release
~ helm search repo skoala-release/skoala --versions
NAME          CHART VERSION  APP VERSION DESCRIPTION
skoala-release/skoala  0.13.0      0.13.0      The helm chart for Skoala
skoala-release/skoala  0.12.2      0.12.2      The helm chart for Skoala
skoala-release/skoala  0.12.1      0.12.1      The helm chart for Skoala
skoala-release/skoala  0.12.0      0.12.0      The helm chart for Skoala
....
```

在部署 skoala 时，会携带当时最新的前端版本，如果想要指定前端 ui 的版本，可以去看前端代码仓库获取对应的版本号：

在工作集群，查看 Skoala-init 的最新版本，直接通过 helm repo 来更新获取最新的

```
~ helm repo update skoala-release
~ helm search repo skoala-release/skoala-init --versions
NAME          CHART VERSION  APP VERSION DESCRIPTION
skoala-release/skoala-init  0.13.0      0.13.0      A Helm Chart for Skoala init, it includes Skoal...
skoala-release/skoala-init  0.12.2      0.12.2      A Helm Chart for Skoala init, it includes Skoal...
skoala-release/skoala-init  0.12.1      0.12.1      A Helm Chart for Skoala init, it includes Skoal...
skoala-release/skoala-init  0.12.0      0.12.0      A Helm Chart for Skoala init, it includes Skoal...
....
```

执行部署（同样适用于升级）

直接执行命令即可，注意对应的版本号

```
~ helm upgrade --install skoala --create-namespace -n skoala-system --cleanup-on-fail \
--set ui.image.tag=v0.9.0 \
--set sweet.enable=true \
--set hive.configMap.data.database.host=mcamel-common-mysql-cluster-mysql-master.mcamel-system.svc.cluster.local \
--set hive.configMap.data.database.port=3306 \
--set hive.configMap.data.database.user=root \
--set hive.configMap.data.database.password=xxxxxxx \
--set hive.configMap.data.database.database=skoala \
skoala-release/skoala \
--version 0.13.0
```

自定义并初始化数据库参数；需要将数据库信息做配置添加进去 --set sweet.enable=true \ --set hive.configMap.data.database.host= \ --set hive.configMap.data.database.port= \ --set hive.configMap.data.database.user= \ --set hive.configMap.data.database.password= \ --set hive.configMap.data.database.database= \

自定义前端 ui 版本 ui.image.tag=v0.9.0

查看部署的 pod 是否启动成功

```
~ kubectl -n skoala-system get pods
NAME          READY   STATUS    RESTARTS   AGE
hive-8548cd9b59-948j2  2/2     Running   2 (3h48m ago)  3h48m
sesame-5955c878c6-jz8cd 2/2     Running   0          3h48m
ui-7c9f5b7b67-9rpzc  2/2     Running   0          3h48m
```

卸载 skoala

这一步骤卸载，会把 skoala 相关的资源删除。

```
~ helm uninstall skoala -n skoala-system
```

更新 skoala

更新操作同 3.4 部署，使用 helm upgrade 指定新版本即可

安装 skoala-init 到工作集群

由于 Skoala 涉及的组件较多，我们将这些组件打包到同一个 Chart 内，也就是 skoala-init，所以我们应该在用到微服务引擎的工作集群安装好 skoala-init

```
~ helm search repo skoala-release/skoala-init --versions
NAME          CHART VERSION  APP VERSION DESCRIPTION
skoala-release/skoala-init  0.13.0      0.13.0      A Helm Chart for Skoala init, it includes Skoal...
```

```
skoala-release/skoala-init 0.12.2      0.12.2      A Helm Chart for Skoala init, it includes Skoal...
skoala-release/skoala-init 0.12.1      0.12.1      A Helm Chart for Skoala init, it includes Skoal...
skoala-release/skoala-init 0.12.0      0.12.0      A Helm Chart for Skoala init, it includes Skoal...
....
```

安装命令，同更新；确认需要安装到指定的命名空间，确认看到全部 Pod 启动成功。

```
~ helm upgrade --install skoala-init --create-namespace -n skoala-system --cleanup-on-fail \
skoala-release/skoala-init \
--version 0.13.0
```

除了通过终端安装，UI 的方式可以在 Kpanda 集群管理中 Helm 应用内找到 Skoala-init 进行安装。

The screenshot shows the Kpanda Helm chart management interface. At the top, there is a search bar with the query 'skoala-init'. Below the search bar, there are two dropdown menus: '仓库' (Repository) set to '全部' (All) and '类型' (Type) set to '全部' (All). A gear icon and the text 'skoala-init' are displayed next to the search results. The results section contains one item: 'skoala-init' with a gear icon, labeled '其他' (Other). Below the item, the description reads: 'A Helm Chart for Skoala init, it includes Skoala Agent, Gateway Operator, Sentinel Operator...'. A blue link icon followed by the URL 'https://release.daocloud.io/chartrepo/addon' is also present. At the bottom left of the results section, it says '共 1 项' (1 item).

卸载命令

```
~ helm uninstall skoala-init -n skoala-system
```

4.2 离线升级微服务引擎模块

本页说明从[下载中心](#)下载微服务引擎模块后，应该如何安装或升级。

!!! info

下述命令或脚本内出现的 `skoala` 字样是微服务引擎模块的内部开发代号。

从安装包中加载镜像

您可以根据下面两种方式之一加载镜像，当环境中存在镜像仓库时，建议选择 chart-syncer 同步镜像到镜像仓库，该方法更加高效便捷。

chart-syncer 同步镜像到镜像仓库

1. 创建 load-image.yaml

!!! note

该 YAML 文件中的各项参数均为必填项。您需要一个私有的镜像仓库，并修改相关配置。

== "已安装 chart repo"

若当前环境已安装 chart repo，chart-syncer 也支持将 chart 导出为 tgz 文件。

```
```yaml
source:
 intermediateBundlesPath: skoala-offline # (1)
target:
 containerRegistry: 10.16.23.145 # (2)
 containerRepository: release.daocloud.io/skoala # (3)
repo:
 kind: HARBOR # (4)
 url: http://10.16.23.145/chartrepo/release.daocloud.io # (5)
 auth:
 username: "admin" # (6)
 password: "Harbor12345" # (7)
containers:
 auth:
 username: "admin" # (6)
 password: "Harbor12345" # (7)
```

```

1. 到执行 charts-syncer 命令的相对路径，而不是此 YAML 文件和离线包之间的相对路径
2. 需更改为你的镜像仓库 url
3. 需更改为你的镜像仓库
4. 也可以是任何其他支持的 Helm Chart 仓库类别
5. 需更改为 chart repo url
6. 你的镜像仓库用户名
7. 你的镜像仓库密码

== "未安装 chart repo"

若当前环境未安装 chart repo，chart-syncer 也支持将 chart 导出为 tgz 文件，并存放在指定路径。

```
```yaml
source:
 intermediateBundlesPath: skoala-offline # (1)
target:
 containerRegistry: 10.16.23.145 # (2)
 containerRepository: release.daocloud.io/skoala # (3)
repo:
 kind: LOCAL
 path: ./local-repo # (4)
containers:
 auth:
 username: "admin" # (5)
 password: "Harbor12345" # (6)
```

```

1. 到执行 charts-syncer 命令的相对路径，而不是此 YAML 文件和离线包之间的相对路径
2. 需更改为你的镜像仓库 url
3. 需更改为你的镜像仓库
4. chart 本地路径
5. 你的镜像仓库用户名
6. 你的镜像仓库密码

2. 执行同步镜像命令。

```
shell
charts-syncer sync --config load-image.yaml
```

Docker 或 containerd 直接加载

解压并加载镜像文件。

1. 解压 tar 压缩包。

```
shell
tar xvf skoala.bundle.tar
```

解压成功后会得到 3 个文件:

- hints.yaml
- images.tar
- original-chart

2. 从本地加载镜像到 Docker 或 containerd。

==== "Docker"

```
```shell
docker load -i images.tar
```

```

==== "containerd"

```
```shell
ctr image import images.tar
```

```

!!! note 每个 node 都需要做 Docker 或 containerd 加载镜像操作，加载完成后需要 tag 镜像，保持 Registry、Repository 与安装时一致。

升级

有两种升级方式。您可以根据前置操作，选择对应的升级方案：

==== "通过 helm repo 升级"

1. 检查微服务引擎 helm 仓库是否存在。

```
```shell
helm repo list | grep skoala
```

```

若返回结果为空或如下提示，则进行下一步；反之则跳过下一步。

```
```none
Error: no repositories to show
```

```

2. 添加微服务引擎的 helm 仓库。

```
```shell
helm repo add skoala-release http://(harbor url)/chartrepo/{project}
```

```

3. 更新微服务引擎的 helm 仓库。

```
```shell
helm repo update skoala-release # (1)
```

```

1. helm 版本过低会导致失败，若失败，请尝试执行 helm update repo

4. 选择您想安装的微服务引擎版本（建议安装最新版本）。

```
```shell
helm search repo skoala-release/skoala --versions
```
```none
[root@master ~]# helm search repo skoala-release/skoala --versions
```

```

```
NAME          CHART VERSION APP VERSION DESCRIPTION
skoala-release/skoala  0.14.0      v0.14.0      A Helm chart for Skoala
...
```

5. 备份 `--set` 参数。

在升级微服务引擎版本之前，建议您执行如下命令，备份老版本的 `--set` 参数。

```
```shell
helm get values skoala -n skoala-system -o yaml > bak.yaml
```

```

6. 执行 `helm upgrade`。

升级前建议您覆盖 bak.yaml 中的 `global.imageRegistry` 字段为当前使用的镜像仓库地址。

```
```shell
export imageRegistry={你的镜像仓库}
```

```shell
helm upgrade skoala skoala-release/skoala \
-n skoala-system \
-f ./bak.yaml \
--set global.imageRegistry=$imageRegistry
--version 0.14.0
```

```

==== “通过 chart 包升级”

1. 备份 `--set` 参数。

在升级微服务引擎版本之前，建议您执行如下命令，备份老版本的 `--set` 参数。

```
```shell
helm get values skoala -n skoala-system -o yaml > bak.yaml
```

```

2. 执行 `helm upgrade`。

升级前建议您覆盖 bak.yaml 中的 `global.imageRegistry` 为当前使用的镜像仓库地址。

```
```shell
export imageRegistry={你的镜像仓库}
```

```shell
helm upgrade skoala . \
-n skoala-system \
-f ./bak.yaml \
--set global.imageRegistry=$imageRegistry
```

```

4.3 选择工作空间

首次进入微服务治理模块时，首先必须选择一个工作空间。

工作空间选择

×

请输入工作空间

| | 工作空间 | ID |
|----------------------------------|-------------------|----|
| <input checked="" type="radio"/> | Default | 2 |
| <input type="radio"/> | kpanda-ws-test001 | 4 |
| <input type="radio"/> | mcamel-ws | 5 |
| <input type="radio"/> | frank-ws | 6 |
| <input type="radio"/> | dsadasd | 8 |
| <input type="radio"/> | pokemon | 9 |
| <input type="radio"/> | skoala | 10 |
| <input type="radio"/> | test-folder-01 | 11 |
| <input type="radio"/> | mspider-ubuntu-52 | 15 |
| <input type="radio"/> | sadasdas | 24 |

共 20 项 < 1 / 2 > 10 项 ▾

确认

如需更改当前所在的工作空间，可以在左侧边栏点击更换图标重新选择工作空间。

| 接入中心名称 | 状态 | 类型 | 接入时间 | 版本 | 微服务总数 | 正常/实例总数 |
|-----------------------|-----|----|---------------------|---------|-------|---------|
| kpanda-global-cluster | 运行中 | CO | 2022-09-15 11:00:00 | v1.24.0 | 0 | 0 / 0 |
| test-nacos | 运行中 | CO | 2022-09-15 17:00:00 | 2.0.4 | 10 | 10 / 10 |
| test | 运行中 | CO | 2022-09-15 17:00:00 | N/A | 1 | 1 / 1 |

!!! note

如果当前没有可选的工作空间，需要先联系超级管理员创建一个工作空间。

[下载 DCE 5.0](#) { .md-button .md-button--primary } 安装 DCE 5.0 { .md-button .md-button--primary } 申请社区免费体验 { .md-button .md-button--primary }

5. Registry

5.1 Integrated

接入注册中心

注册中心支持接入 [Nacos 注册中心](#)、[Eureka 注册中心](#)、[Zookeeper 注册中心](#)、[Kubernetes 注册中心](#)、[Mesh 注册中心](#)。

相对于托管型注册中心而言，接入型注册中心只支持一些基础操作，例如查看基本信息、监控信息等。如需体验更高级更全面的管理操作，需要创建[托管注册中心](#)。

接入注册中心的步骤如下：

- 在左侧导航栏点击 微服务治理中心 --> 接入注册中心，然后在页面右上角点击 接入注册中心。

接入注册中心列表

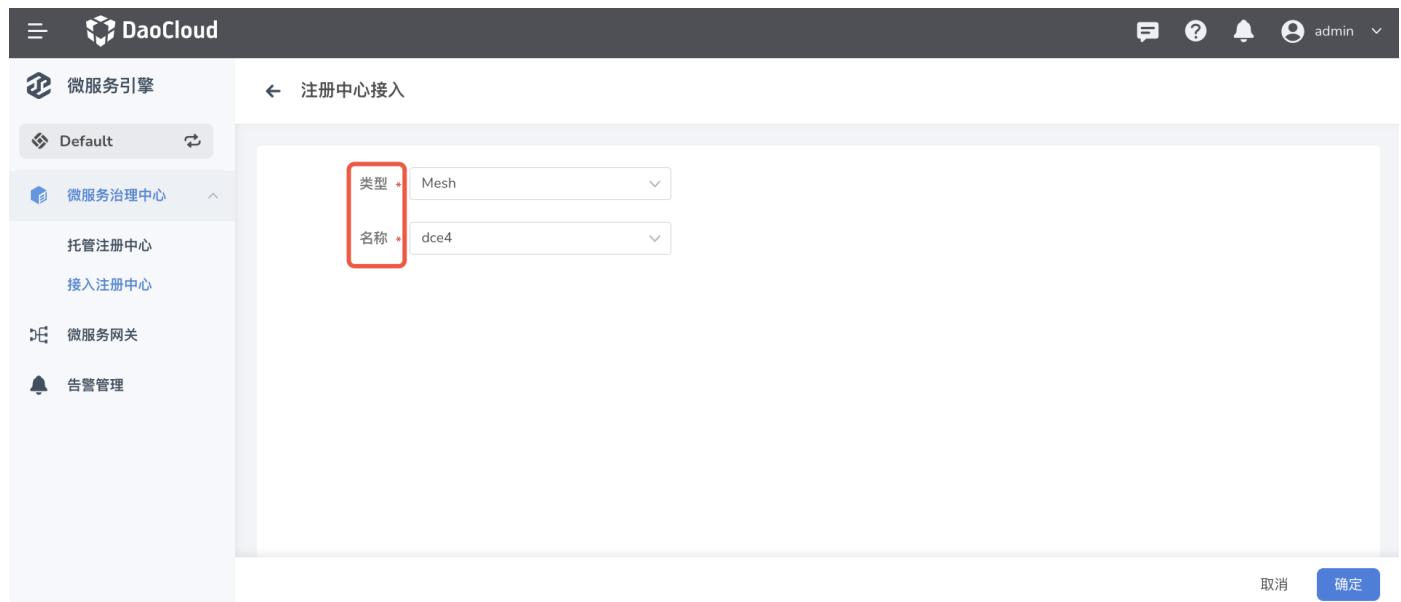
| 名称 | 状态 | 服务总数 | 正常 / 实例总数 |
|-----------------------|-----|------|-----------|
| kpanda-global-cluster | 运行中 | 26 | 39 / 41 |
| nacos-new | 异常 | 0 | 0 / 0 |
| nacos-test | 运行中 | 12 | 14 / 14 |

共 3 项

- 填写配置信息，然后在页面底部点击 确定。

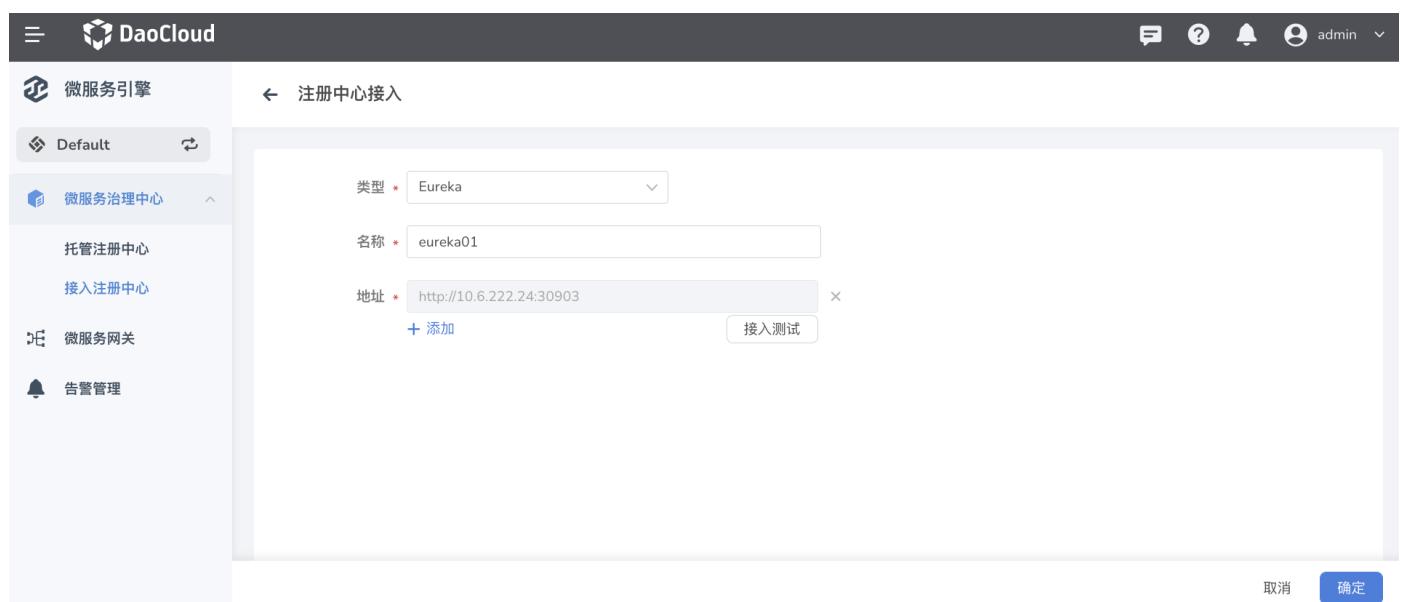
接入不同类型的注册中心需要填写不同的配置信息。

- Kubernetes/Mesh 注册中心：直接选择想要接入的集群或网格服务。
- 如果找不到想要添加的 Kubernetes 集群，可以去容器管理模块[接入集群](#)或[创建集群](#)。
- 如果找不到想要添加的网格服务，可以去网格服务模块[创建网格](#)。



- Nacos/Zookeeper/Eureka 注册中心：填写注册中心的名称和地址，点击 接入测试。

地址栏变灰则说明接入测试成功。对于分布式的高可用注册中心，还可以点击 + 添加可以输入多个地址。



微服务管理

接入[注册中心](#)之后，可以通过注册中心对其中的微服务进行管理。微服务管理主要指查看注册中心下的微服务，

!!! note

接入型注册中心仅支持基础的管理操作。对于更复杂的管理场景，建议创建[托管注册中心](../../../../managed/registry-lcm/create-registry.md)以便执行更多高级操作。

- 在接入注册中心列表 页面点击目标注册中心的名称。

- 在左侧导航栏点击 微服务管理， 查看微服务列表和基本信息。

在当前页面，可以复制微服务的名称，可以查看当前注册中心下的所有微服务，以及各个微服务的所属命名空间、实例情况、请求统计数据等。

- 点击微服务的名称，查看微服务的实例列表、接口列表、监控信息等。

全球服务端口

注册中心名称: global-service / 微服务管理 / abc

服务信息

| abc | 2 | 2 | - | - | - |
|-------|------|-------|--------|----------|------------|
| 微服务名称 | 实例总数 | 健康实例数 | 请求量(次) | 请求成功率(%) | 请求平均耗时(ms) |
| abc | 2 | 2 | - | - | - |

链路追踪

实例列表

| 实例 ID/名称 | 实例状态 | 健康状态 | 服务端口 | IP 地址 | 更新时间 |
|-------------------|------|------|------|-----------------|---------------------|
| dao-2048-75486... | 在线 | 运行中 | - | 192.168.201.168 | 2022-10-31 15:40:03 |

共 1 项

1 / 1 10 项

- 实例列表：查看实例状态、IP 地址、服务端口等。

点击实例名称，还可以进一步查看该实例的监控信息和元数据。

The screenshot shows the DaoCloud service instance management interface. At the top, the navigation bar includes the DaoCloud logo, user information (admin), and a search bar with the text "注册中心名称: global-service / 微服务管理 / abc / dao-2048-7548646669-z9rhr". A red box highlights this search bar. Below the search bar is a section titled "实例信息" (Instance Information) containing fields for "实例 ID/名称" (Instance ID/Name), "所属服务" (Service), "实例状态" (Instance Status - Unknown), "服务端口" (Service Port), and "IP 地址" (IP Address). Below this is a timeline from "2022-11-14 14:48:52" to "2022-11-14 14:53:52". Under the timeline is a placeholder message: "暂无数据,请参考操作手册进行配置." (No data available, please refer to the operation manual for configuration.). At the bottom right of the main content area is a blue "刷新" (Refresh) button and a red-outlined "创建" (Create) button.

- 接口列表：查看微服务已经具有的接口，或者创建新的接口。

The screenshot shows the DaoCloud service interface. The top navigation bar is identical to the previous screenshot. The main content area is titled "服务信息" (Service Information) and displays a table with columns: "微服务名称" (Microservice Name: abc), "实例总数" (Total Instances: 2), "健康实例数" (Healthy Instances: 2), "请求量(次)" (Request Count: -), "请求成功率(%)" (Success Rate (%): -), and "请求平均耗时(ms)" (Average Response Time (ms): -). Below the table are three tabs: "实例列表" (Instance List), "接口列表" (API List, which is selected and highlighted with a blue border), and "监控" (Monitoring). At the bottom right of the main content area is a blue "刷新" (Refresh) button and a red-outlined "创建" (Create) button. A placeholder message "No API definition provided." is visible in the center of the content area.

- 监控信息：查看微服务的监控信息，包括请求数、错误率、响应耗时、请求率等。

支持自定义时间范围。

注册中心名称: global-service / 微服务管理 / abc

服务信息

| abc | 2 | 2 | - | - | - |
|-------|------|-------|--------|----------|------------|
| 微服务名称 | 实例总数 | 健康实例数 | 请求量(次) | 请求成功率(%) | 请求平均耗时(ms) |
| abc | 2 | 2 | - | - | - |

实例列表 接口列表 监控

2022-11-01 00:00:00 - 2022-11-14 00:00:00

请求概览

请求数 (次) 错误率 (%)

■ 请求成功 ■ 请求失败

■ 错误率 (%)

暂无数据

14:49 14:50 14:51 14:52 14:53 14:54 14:55 14:56 14:57

响应耗时 (毫秒) 请求率 (req/s)

■ P50 ■ P75 ■ P90 ■ P95 ■ P99 ■ avg

■ 请求率 (req/s)

暂无数据

1 0.8 0.6 0.4 0.2 0

1 0.8 0.6 0.4 0.2 0

移除注册中心

- 在接入注册中心列表页选择需要移除的注册中心，在右侧点击 **...** 并选择 移除。

The screenshot shows the '接入注册中心列表' (Access Registry Center List) page. It lists three registry centers:

- kpanda-global-cluster**: 运行中 (Running). Type: CO, 接入时间: 2022-08-30 14:49, 接入类型: 插入 (Insert), 版本: v1.20.7. 显示 152 微服务总数, 207 / 223 正常/实例总数.
- nacos-test**: 运行中 (Running). Type: CO, 接入时间: 2022-09-02 17:48, 接入类型: 插入 (Insert), 版本: 2.0.4. 显示 11 微服务总数, 15 / 15 正常/实例总数. 这一行的右侧有一个红色方框包围了右侧的三个操作按钮：编辑、移除（被选中）、和更多。
- nacos-test11**: 运行中 (Running). Type: CO, 接入时间: 2022-09-09 10:24, 接入类型: 插入 (Insert), 版本: 2.0.4. 显示 9 微服务总数, 9 / 9 正常/实例总数.

- 输入注册中心的名称，点击 移除。

移除接入托管注册中心

X

确定要移除 eureka-benchmark-test 吗？

请输入eureka-benchmark-test

移除
取消

!!! note

接入型的注册中心仅支持‘移除’操作，而托管型的注册中心仅支持‘删除’操作。二者的区别在于：

- 移除：只是将注册中心从 DCE 5.0 的微服务引擎中移除，不会删除原有的注册中心和数据，后续还可以再次接入该注册中心。
- 删除：删除注册中心及其中的所有数据，后续无法再次使用该注册中心，需要重新创建新的注册中心。

链路追踪

DCE 5.0 支持服务级别的链路追踪与查询，查看各个微服务的链路时延分布情况，在出现故障时可以快速定位故障。

更多详情，可参考[可观测性—>链路查询](#)。

微服务治理

微服务治理主要指基于 Istio 的网格服务流量治理。微服务引擎中集成了 DCE 5.0 的[服务网格](#)模块，支持通过虚拟服务、目标规则、网关规则三种方式治理网格服务。

!!! note

- 仅 Mesh 类型的注册中心支持此功能。
- Kubernetes/Zookeeper/Nacos/Eureka 类型的注册中心不会在左侧导航栏显示、微服务治理、操作。

有关微服务治理的更多内容，可参考[服务网格-->流量治理](#)。

更新注册中心配置

微服务引擎目前仅支持更新 Nacos/Zookeeper/Eureka 注册中心的配置。

- 在接入注册中心列表页选择需要更新的注册中心，在右侧点击 **...** 并选择 **编辑**。

| 注册中心 | 微服务总数 | 正常 / 实例总数 |
|-----------------------|-------|-----------|
| kpanda-global-cluster | 152 | 207 / 223 |
| nacos-test | 11 | 15 / 移除 |
| nacos-test11 | 9 | 9 / 9 |

- 增加或删除注册中心的地址，然后在页面底部点击 **确定**。

编辑
X

类型 * NACOS
X

名称 * nacos-new
X

地址 * http://10.6.222.21:30698
X

X

+ 添加
接入测试

取消
确定

!!! note

如需更新 Kubernetes/Mesh 注册中心：

- 可以先 [移除已经接入的注册中心] ([remove-registry.md](#))，然后再重新接入其他的注册中心。
- 也可以去容器管理模块 [更新对应的集群] ([../../../../kpanda/user-guide/clusters/upgrade-cluster.md](#))，或者去服务网格模块 [更新对应的网格服务] ([../../../../mspider/user-guide/service-mesh/create-mesh.md](#))。

5.2 Managed

微服务配置列表

微服务引擎支持基于微服务命名空间对微服务进行分组管理，以便灵活地根据自己的需要按照环境或者应用、模块等分组管理微服务以及 Spring 的大量配置。配置列表主要提供了微服务配置的增删改查、查看历史版本、回滚、订阅者查询等核心管理能力。

新建配置

1. 进入 微服务引擎 -> 微服务治理中心 -> 托管注册中心模块，点击目标注册中心的名称。

| 容器集群 | 命名空间 | 微服务总数 | 配置总数 | 正常 / 节点总数 |
|--------------|-------------|-------|------|-----------|
| skoala-dev | nacos-yxt | 0 | 1 | 1 / 1 |
| skoala-stage | nacos-video | 0 | 0 | 1 / 1 |

2. 在左侧导航栏点击 微服务配置列表，然后在右上角点击 创建。

| Data ID | Group | 所属应用 |
|--------------------------------|----------------|------|
| my-otel-demo-adservice | DEFAULT_GROUP | ... |
| nacos-client-dubbo3x-yang-c... | dubbo | ... |
| sentinel-demo-degrade-rules | SENTINEL_GROUP | ... |

3. 填写各项配置信息

DaoCloud

nacos-video skoala

基础信息

微服务命名空间

微服务列表

微服务配置列表

监控

资源列表

插件中心

创建微服务配置

基本配置

命名空间: public

Data ID: adservice

Group: DEFAULT_GROUP

配置格式: TEXT JSON XML YAML HTML PROPERTIES (selected)

配置内容:

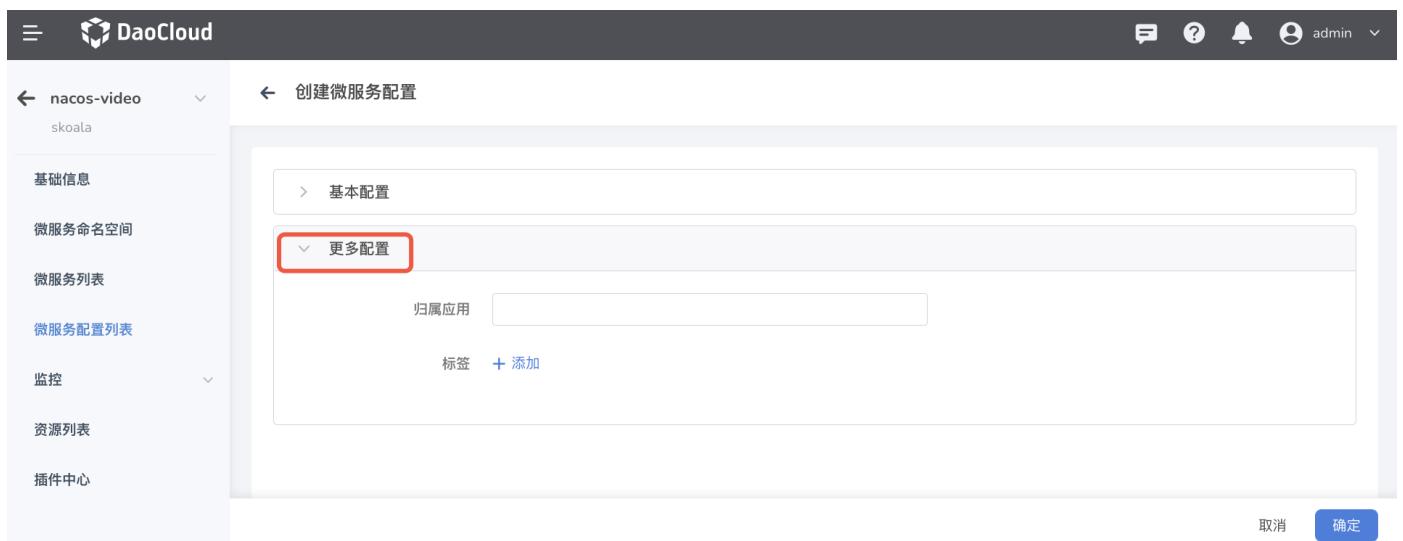
```
1 test.config=dynamic
```

配置描述

取消 确定

This screenshot shows the 'Create Microservices Configuration' dialog in the DaoCloud Nacos interface. The left sidebar lists various service components like 'nacos-video' and 'skoala'. The main dialog is titled 'Basic Configuration' and includes fields for Namespace (public), Data ID (adservice), Group (DEFAULT_GROUP), and Configuration Format (PROPERTIES). The configuration content is a single line: 'test.config=dynamic'. At the bottom are 'Cancel' and 'Confirm' buttons.

- 命名空间：选择当前配置所属的微服务命名空间，默认为 `public`
不同命名空间下的服务和配置严格隔离，不能互相引用。例如，A 命名空间下的服务不能引用 B 命名空间下的配置。
- Data ID：相当于当前配置文件的名称。同一分组下的 Data ID 不能重复。
完整格式为： `${prefix}-${spring.profiles.active}.${file-extension}`。
- `prefix` 默认为 `spring.application.name` 的值，也可以通过配置项 `spring.cloud.nacos.config.prefix` 来配置。
- `spring.profiles.active` 即为当前环境对应的 profile，详情可以参考 Spring Boot 文档。注意：当 `spring.profiles.active` 为空时，对应的连接符 `-` 也将不存在，Data ID 的拼接格式变成 `${prefix}.${file-extension}`。
- `file-extension` 为配置内容的数据格式，可以通过配置项 `spring.cloud.nacos.config.file-extension` 来配置。
- Group：选择当前配置所属的分组，默认为 `DEFAULT_GROUP`。
通常用命名空间隔离环境，用分组区分项目。
- 配置格式：设置当前配置文件的格式，
配置内容：输入服务源代码中具有 `@Value` 注解的配置项。
支持格式校验，如果配置内容存在格式错误，系统会自动提示错误。
服务源代码中具有 `@RefreshScope` 注解的配置项支持动态更新。
更多配置->归属应用（选填）：选择当前配置所属的应用。



4. 在页面右下角点击 确定。

查看配置

1. 进入 微服务引擎 -> 微服务治理中心 -> 托管注册中心 模块，点击目标注册中心的名称。

托管注册中心列表

| 实例名称 | 容器集群 | 命名空间 | 微服务总数 | 配置总数 | 正常 / 节点总数 |
|-------------|--------------|-------------|-------|------|-----------|
| nacos-yxt | skoala-dev | skoala-yxt | 0 | 1 | 1 / 1 |
| nacos-video | skoala-stage | skoala-demo | 0 | 0 | 1 / 1 |

2. 在左侧导航栏点击 微服务配置列表，点击目标配置的 Data ID。

托管注册中心: nacos-video / 微服务配置列表

| Data ID | Group | 所属应用 |
|----------------------------------|----------------|------|
| my-otel-demo-adservice | DEFAULT_GROUP | |
| nacos-client-dubbo3x-yang-con... | dubbo | |
| sentinel-demo-degrade-rules | SENTINEL_GROUP | |

3. 查看该配置的基本信息、配置内容、历史版本、监听者、示例代码等信息。

- 新手可以借助示例代码快速使用客户端编程消费该配置，降低新手使用门槛。
- 支持查询当前配置的监听者以及 MD5 校验值，便于了解正在使用此配置的微服务，以及配置变更是否成功推送到客户端。

The screenshot shows the DaoCloud Nacos configuration management interface. The left sidebar shows a tree structure with 'nacos-video' selected, containing 'skoala', '基础信息', '微服务命名空间', '微服务列表', '微服务配置列表' (which is currently selected), '监控', '资源列表', and '插件中心'. The main content area displays the configuration details for 'my-otel-demo-adservice'. It includes a '基本信息' card with fields like Data ID (my-otel-demo-adservice), Group (DEFAULT_GROUP), 所属应用 (public), and 命名空间 (命名空间). Below this is a table with rows for '标签' (text) and '配置格式' (MD5). At the bottom, there are tabs for '配置内容' (selected), '历史版本', '监听查询', and '示例代码'. The '配置内容' tab shows a single line of code: 1 test.config=dynamic.

历史版本回滚

微服务配置列表会记录配置文件的历史版本，支持一键回滚到特定的历史版本，便于用户在改错配置时快速恢复，降低微服务系统中的配置可用性风险。回滚时支持版本差异对比，便于对比当前版本和目标回滚版本的差异，校验修改内容，降低改错带来的风险。

1. 进入 微服务引擎 -> 微服务治理中心 -> 托管注册中心模块，点击目标注册中心的名称。

| 容器集群 | 命名空间 | 微服务总数 | 配置总数 | 状态 |
|--------------|-------------|-------|------|-----------------|
| skoala-dev | skoala-yxt | 0 | 1 | 1 / 1 正常 / 节点总数 |
| skoala-stage | skoala-demo | 0 | 0 | 1 / 1 正常 / 节点总数 |

2. 在左侧导航栏点击 微服务配置列表，点击目标配置的 Data ID。

| Data ID | Group | 所属应用 |
|----------------------------------|----------------|------|
| my-otel-demo-adservice | DEFAULT_GROUP | ... |
| nacos-client-dubbo3x-yang-con... | dubbo | ... |
| sentinel-demo-degrade-rules | SENTINEL_GROUP | ... |

3. 点击 历史版本 页签，找到对应的记录，在记录右侧点击 : 并选择 回滚。

基础信息
微服务命名空间
微服务列表
微服务配置列表
监控
资源列表
插件中心

托管注册中心: nacos-video / 微服务配置列表

| Data ID | Group | 所属应用 |
|--------------------------------|----------------|------|
| my-otel-demo-adservice | DEFAULT_GROUP | |
| nacos-client-dubbo3x-yang-c... | dubbo | |
| sentinel-demo-degrade-rules | SENTINEL_GROUP | |

共 3 项

4. 对比版本差异并确认，然后在右下角点击回滚。

基础信息
微服务命名空间
微服务列表
微服务配置列表
监控
资源列表
插件中心

托管注册中心: nacos-video / 微服务配置列表 / my-otel-demo-adservice

基本信息

| | | |
|------------------------|---------------|----------------------------------|
| my-otel-demo-adservice | DEFAULT_GROUP | public |
| Data ID | Group | 命名空间 |
| - | text | fa4d0443c8196a5fac60761c6fa03... |
| 标签 | 配置格式 | MD5 |

配置内容 历史版本 监听查询 示例代码

| 来源 IP | 操作人 | 更新时间 | 操作类型 | 更多 |
|------------|-----|------------------|------|-----|
| 10.244.2.6 | | 2022-11-25 11:28 | 更新 | ... |
| 10.244.2.6 | | 2022-11-25 11:28 | 更新 | 回滚 |
| 10.244.2.6 | | 2022-11-25 11:27 | 创建 | ... |

共 3 项

更新/删除配置

1. 进入 微服务引擎 -> 微服务治理中心 -> 托管注册中心 模块，点击目标注册中心的名称。

微服务引擎

微服务治理中心

托管注册中心

接入注册中心

微服务网关

告警管理

部署位置 全部 实例名称

nacos-yxt 运行中

容器集群 skoala-dev 命名空间 skoala-yxt
版本 2.0.4 创建时间 2022-11-29 17:00:00
IP地址 -

0 1 1 / 1

微服务总数 配置总数 正常 / 节点总数

nacos-video 运行中

容器集群 skoala-stage 命名空间 skoala-demo
版本 2.0.4 创建时间 2022-11-25 10:00:00
IP地址 -

0 0 1 / 1

微服务总数 配置总数 正常 / 节点总数

共 2 项 1 / 1 10 项

2. 在左侧导航栏点击 微服务配置列表，在目标配置右侧点击 : 按钮根据需要选择更新或删除。

基础信息

微服务命名空间

微服务列表

微服务配置列表

监控

资源列表

插件中心

微服务命名空间 public 搜索

创建

| Data ID | Group | 所属应用 | 操作 |
|--------------------------------|----------------|------|--------------------------------|
| my-otel-demo-adservice | DEFAULT_GROUP | | ...
编辑
删除 |
| nacos-client-dubbo3x-yang-c... | dubbo | | ...
编辑
删除 |
| sentinel-demo-degrade-rules | SENTINEL_GROUP | | ...
编辑
删除 |

共 3 项 1 / 3 1 项

微服务命名空间

微服务命名空间可用于隔离生产、开发、测试等不同环境下的服务、配置等资源。微服务引擎模块中的命名空间指的是微服务命名空间，即 Nacos 语境中的命名空间，并非 Kubernetes 场景下的命名空间。

!!! note

- 创建托管型注册中心实例时，系统会自动创建一个名为 `**public**` 的默认命名空间。该命名空间不可编辑、不可删除，属于系统自带的保留命名空间。
- 不同命名空间下的服务和配置严格隔离，不能互相引用。例如，A 命名空间下的服务不能引用 B 命名空间下的配置。

创建微服务命名空间

1. 进入 微服务引擎 → 微服务治理中心 → 托管注册中心模块，点击目标注册中心的名称。

| 容器集群 | 命名空间 | 微服务总数 | 配置总数 |
|------------|-------------|-------|-------|
| skoala-dev | skoala-test | 0 | 1 / 1 |
| skoala-dev | lfj-demo22 | 0 | 1 / 1 |

2. 在左侧导航栏点击 微服务命名空间，然后在右上角点击 创建。

| 微服务命名空间名称 | 命名空间 ID | 配置数 | 操作 |
|---------------|--------------------------------------|-----|----|
| public (保留空间) | public | 1 | ⋮ |
| ruby | 268ab4a5-58c2-4651-8c52-124e938e0d2d | 2 | ⋮ |

3. 填写命名空间的 ID，名称和描述信息，在页面右下角点击 确定。

如果不填写 ID，系统会自动生成一个 ID。命名空间 ID 在创建之后不可更改。

The screenshot shows the DaoCloud web interface for managing microservices. The top navigation bar includes the DaoCloud logo, user information (admin), and a message center icon. The left sidebar lists several sections: 基础信息, 微服务命名空间 (selected), 微服务列表, 微服务治理策略, 微服务配置列表, 监控告警, 日志查看, and 插件中心. The main content area is titled '创建' (Create) and shows two input fields: '命名空间 ID' (Namespace ID) with value '1' and '命名空间名称' (Namespace Name) with value 'ns'. Below these fields are two buttons: '取消' (Cancel) and '确定' (Confirm).

更新微服务命名空间

- 在对应命名空间的右侧操作栏下点击 编辑，进入更新页面。

The screenshot shows the DaoCloud interface for managing service namespaces. On the left, there's a sidebar with options like '基础信息', '微服务命名空间' (which is selected and highlighted in blue), '微服务列表', '微服务治理策略', '微服务配置列表', '监控告警', '日志查看', and '插件中心'. The main content area is titled '托管注册中心: nacos-web / 微服务列表'. It displays a table with two rows:

| 微服务命名空间名称 | 命名空间 ID | 配置数 | 操作 |
|---------------|--------------------------------------|-----|---|
| public (保留空间) | public | 1 | 更多 |
| ruby | 268ab4a5-58c2-4651-8c52-124e938e0d2d | 2 | 编辑 更多 |

A red box highlights the 'Edit' button for the 'ruby' namespace.

- 修改命名空间的名称、描述信息，点击 确定。

The screenshot shows the 'Edit' dialog box for a service namespace. At the top, it says '命名空间 ID 268ab4a5-58c2-4651-8c52-124e938e0d2d'. Below that is a field labeled '命名空间名称 *' containing 'ruby'. At the bottom right are '取消' (Cancel) and '确定' (Confirm) buttons.

删除微服务命名空间

在对应命名空间的右侧操作栏下点击 删除，然后在弹框中点击 立即删除。

!!! note

删除命名空间之前，必须清理该命名空间下的所有资源，否则无法删除命名空间。

The screenshot shows the DaoCloud interface for managing service namespaces. The sidebar and main content area are identical to the previous screenshot, but the 'ruby' namespace row in the table has a different '操作' (Operation) column:

| 微服务命名空间名称 | 命名空间 ID | 配置数 | 操作 |
|---------------|--------------------------------------|-----|---|
| public (保留空间) | public | 1 | 更多 |
| ruby | 268ab4a5-58c2-4651-8c52-124e938e0d2d | 2 | 删除 更多 |

A red box highlights the 'Delete' button for the 'ruby' namespace.

Governance

应用接入 Sentinel 规范

为了正常使用 DCE 5.0 微服务引擎提供的 [Sentinel 流量治理](#)和查看 [Sentinel 数据监控](#)，需要将应用接入 Sentinel 控制台，并且传递应用参数时需要满足一定规范。

PROJECT_NAME 参数

project.name 参数的格式应为: `{{nacos_namespace_id}}@@{{nacos_group}}@@{{appName}}`。

需要注意的是：

- 符合此规范时，Sentinel 的治理规则会被推送到对应命名空间下，对应配置分组下的配置中心。
- 如果不符合此规范，比如只有 `appName` 或者 `{{nacos_group}}@@{{appName}}`，则所有治理规则都会被推送到 `public` 命名空间下的 `SENTINEL_GROUP` 配置中心。
- 第一部分 `{{nacos_namespace_id}}` 指的是 Nacos 命名空间的 ID，而非命名空间的名称。
- Nacos 的 `public` 命名空间对应的 ID 是空字符 “”。
- 如果想把应用接入 `public` 命名空间，必须使用空字符串，例如 `@@@appA`。

!!! note

为了正常显示 Sentinel 监控数据，应使用 Sentinel 官方 SDK v1.8.6 以上的版本。

??? note "如何获取 Sentinel 控制台"

- 从 [\[release\]](#)(<https://github.com/alibaba/Sentinel/releases>)页面下载最新版本的控制台 jar 包，或者
- 从最新版本的源码自行构建 Sentinel 控制台
 - 1. 下载[\[控制台\]](#)(<https://github.com/alibaba/Sentinel/tree/master/sentinel-dashboard>)工程
 - 2. 使用以下命令将代码打包成一个 fat jar: `mvn clean package`

具体可参考官方文档: [\[启动控制台\]](#)(<https://sentinelguard.io/zh-cn/docs/dashboard.html>)

创建流控规则

流控规则的原理是监控应用或服务流量的 QPS 指标，当指标达到阈值时根据预先设定的规则对请求流量进行控制，防止应用因短时间内无法处理过多流量而崩溃。使用流控规则后，系统可以在接下来的空闲期间逐渐处理堆积的请求，当指标重新恢复到阈值以下后，恢复正常流量请求控制。

创建流控规则的方式如下：

1. 点击目标托管注册中心的名称，然后在左侧导航栏点击 微服务列表，在最右侧点击更多按钮选择 治理。

注意需要治理的微服务在 是否可以治理一栏应该显示为 是，才能进行后续步骤。

| 微服务名称 | 分组 | 健康/实例总数 | 实例组数 | 触发保护阈值 | 请求量(次) | 请求成功率(%) | 请求平均耗时(ms) | 链路追踪 | 是否可以治理 |
|------------|--------------|---------|------|--------|--------|----------|------------|------|--------|
| lf-0214... | DEFAULT_G... | 2/2 | 1 | 否 | - | - | - | ● - | 是 |
| my-otel... | DEFAULT_G... | 3/3 | 1 | 否 | 0 | 100.00% | 0.00 | ● 入接 | 是 |
| zxw-tes... | DEFAULT_G... | 1/1 | 1 | 否 | - | - | - | ● - | 治理 |

2. 选择 流控规则，然后在右侧点击 创建流控规则。

| 微服务名称 | 分组 | 实例组数 | 健康实例数 | 总实例数 |
|------------------------|---------------|------|-------|------|
| my-otel-demo-adservice | DEFAULT_GROUP | 1 | 3 | 3 |

触发保护阈值: 0, 100.00%, 0.00

请求量(次): -, 请求成功率(%): -, 请求平均耗时(ms): -

服务治理

流控规则 (highlighted)

创建流控规则

3. 参考下列说明填写规则配置，并在右下角点击 确定。

创建流控规则

基础信息

微服务命名空间

微服务列表

微服务配置列表

监控

资源列表

插件中心

取消 确定

- 资源名：资源的名称，资源是指规则的作用领域，即创建此规则是为了治理哪个服务的流量。
- 来源应用：Sentinel 可以针对调用者进行限流，默认为 default，表示不区分来源，适用于所有的调用者。
- 直接模式：当前的资源达到阈值时，直接限流。
- 关联模式：当关联的资源达到阈值时，限流关联资源。
- 链路模式：只记录指定链路上的流量（指定资源从入口资源进来的流量，如果达到阈值，就可以限流）。

4. 创建完成后可以在流控规则列表中查看新建的规则。在右侧点击更多按钮可以编辑更新规则或者删除该规则。

托管注册中心: nacos-gy / 微服务列表 / sentinel-demo

基础信息

微服务命名空间

微服务列表

微服务配置列表

监控

资源列表

插件中心

服务治理

流控规则

触发保护阈值

请求量(次)

请求成功率(%)

请求平均耗时(ms)

实例列表

实例组数

健康实例数

总实例数

服务治理

编辑

删除

创建流控规则

规则介绍

微服务引擎支持通过服务网格或 Sentinel 治理东西向流量。

SENTINEL 治理规则

- [流控规则](#)

流控规则的原理是监控应用或服务流量的 QPS 指标，当指标达到阈值时根据预先设定的规则对请求流量进行控制，防止应用因短时间内无法处理过多流量而崩溃。使用流控规则后，系统可以在接下来的空闲期间逐渐处理堆积的请求，当指标重新恢复到阈值以下后，恢复正常流量请求控制。

- [熔断降级](#)

在分布式系统中，各个服务通常需要调用其他的内部或外部服务才能正常运行，如果被调用的服务不够稳定，那么级联效应会导致调用者自身的响应时间也变长，产生线程堆积甚至导致服务不可用。为了避免这种情况出现，需要通过熔断机制根据预设的规则切断不稳定的调用链路，或者对下游服务进行降级，保护系统的整体可用性。

- [热点规则](#)

热点指经常被访问的数据。设置热点规则时，需要配置热点参数（即需要统计访问量的目标参数），然后系统会统计对该热点参数的请求量，当达到一定的阈值后，包含该热点参数的资源就是被限制调用。热点规则适用于统计被频繁访问的资源，达到一定的阈值后限制对该资源的访问。

- [系统规则](#)

系统规则是指，由 Sentinel 综合系统容量、CPU 使用率、平均响应时间、入口 QPS 等数据，从整体维度出发自动选择流控规则控制请求流量。和其他规则不同的是，系统规则针对的是应用级别的入口流量，即仅对进入应用的流量生效，而其他规则通常是针对资源维度进行控制。

- [授权规则](#)

授权规则允许基于请求来源进行流量治理，例如仅放行白名单中的调用方发起的请求，不放行黑名单中的调用方发起的请求。

[流控规则](#) { .md-button .md-button--primary }

MESH 治理规则

流量治理为用户提供了三种资源配置，虚拟服务、目标规则、网关规则。通过配置相应规则可以实现路由、重定向、熔断、分流等多项流量治理功能。用户可以通过向导或YAML形式创建、编辑治理策略。

- 虚拟服务主要用于对请求流量的路由定制规则，并可以对数据流做出分流、重定向、超时返回等处理。
- 目标规则则更关注流量本身的治理，为请求流量提供更强大的负载均衡、连接存活探寻、熔断等功能。
- 网关规则为 Istio 网关提供服务在网关的暴露方式。

有关这三项治理规则的详细说明，可参考文档[流量治理](#)。

[虚拟服务](#) { .md-button .md-button--primary } [目标规则](#) { .md-button .md-button--primary } [网关规则](#) { .md-button .md-button--primary }

Monitor

组件监控

!!! note

集群中需要[安装 insight-agent 组件](../../../../insight/user-guide/quickstart/install-agent.md)后才能使用监控功能。

微服务引擎通过内置 Grafana 提供全方位的监控功能，覆盖各个微服务、系统组件、以及服务调用链路等监控对象。其中，组件监控功能可以提供对 Nacos 和 Sentinel 的监控信息，包括节点数、服务数、CPU/内存用量、JVM 线程数、http 请求总耗时等多个维度的信息。

查看组件监控的步骤如下：

1. 进入微服务引擎模块，点击目标注册中心的名称。

| 实例名称 | 状态 | 容器集群 | 命名空间 | 版本 | 创建时间 | 微服务总数 | 配置总数 | 正常 / 节点总数 |
|------------|-----|------------|-------------|-------|-----------------|-------|------|-----------|
| nacos-gy | 运行中 | skoala-dev | skoala-test | 2.0.4 | 2022-12-28 1... | 0 | 0 | 1 / 1 |
| nacos-auth | 运行中 | skoala-dev | skoala-test | 2.0.4 | 2022-12-22 2... | 0 | 0 | 1 / 1 |
| nacos-test | 运行中 | skoala-dev | skoala-test | 2.0.4 | 2022-12-19 1... | 0 | 0 | 1 / 1 |

2. 在左侧导航栏点击 监控 -> 组件监控。

| 节点数 | 服务数 | IP个数 | 配置数 | 配置长连... | 读配置统... |
|-----|-----|------|-----|---------|---------|
| 1 | 2 | 2 | 0 | 0 | 0 |

cpu

30.1

内存

38.0

JVM线程数

165

系统负载

6.71

配置水平通知耗时

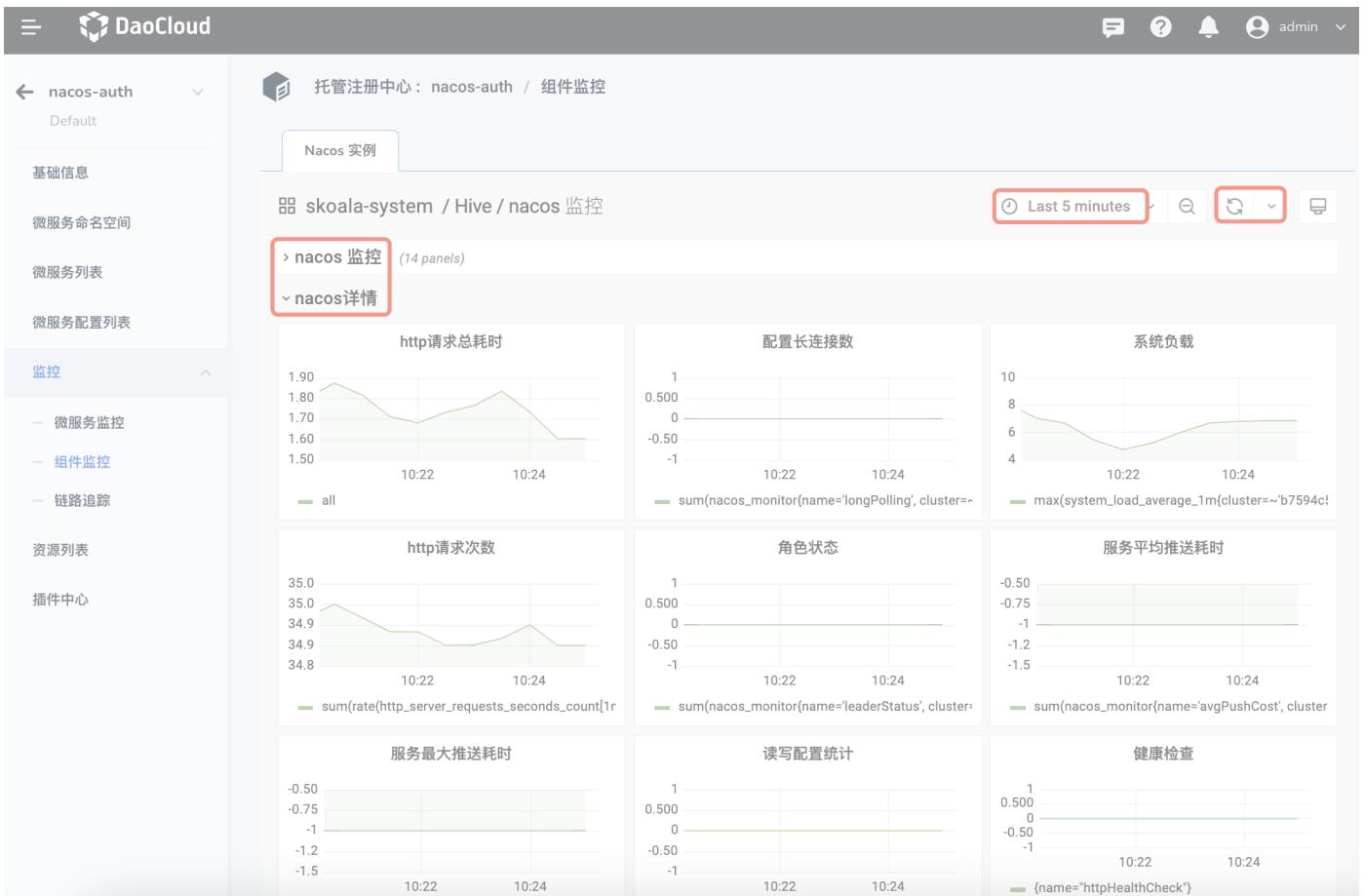
N/A

http请求总耗时

1.83

3. 点击 Nacos 实例页签即可查看 Nacos 实例的监控信息。

下拉页面可查看请求耗时、请求次数等统计详情。在页面右上角可切换统计时间窗口和刷新周期。



微服务监控

微服务引擎通过内置 Grafana 提供全方位的监控功能，覆盖各个微服务、系统组件、以及服务调用链路等监控对象。其中，微服务监控功能可以提供对托管注册中心下各个微服务的监控信息，包括服务响应时间、并发量、异常 QPS、阻塞 QPS、通过 QPS、成功 QPS 等多个维度的信息。

!!! note

集群中需要[安装 insight-agent 组件] (../../../../insight/user-guide/quickstart/install-agent.md)后才能使用监控功能。

查看微服务监控的步骤如下：

1. 进入微服务引擎模块，点击目标注册中心的名称。

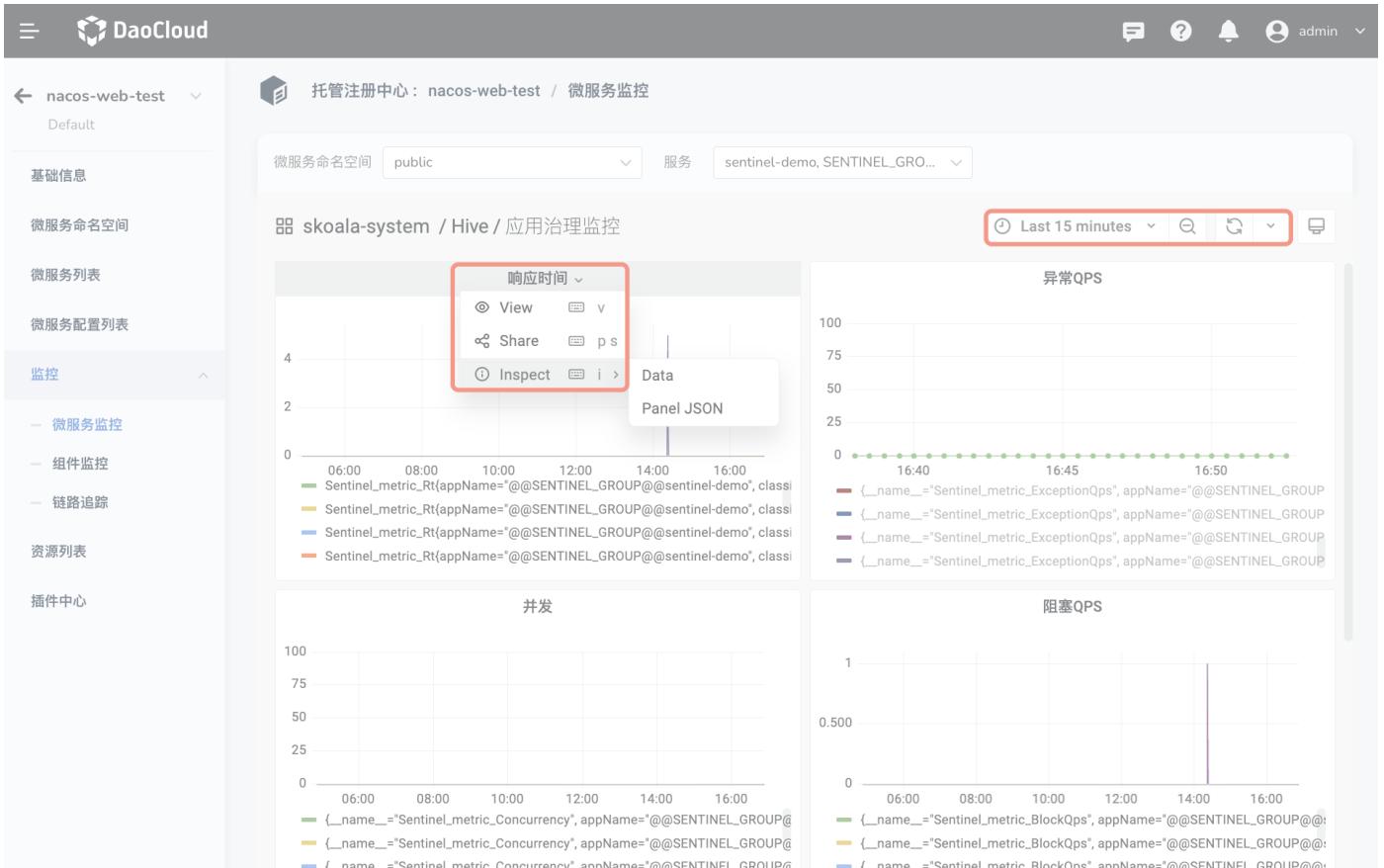
The screenshot shows the DaoCloud interface with the 'Microservices Engine' module selected. The left sidebar lists 'Default', 'Microservice Governance Center', 'Managed Registration Center' (which is expanded), 'Access Registration Center', 'Microservice Gateway', and 'Alert Management'. The main area displays three registered centers: 'nacos-gy' (running, 0 services, 0 configurations, 1 node), 'nacos-auth' (running, 0 services, 0 configurations, 1 node), and 'nacos-test' (running, 0 services, 0 configurations, 1 node). The 'nacos-auth' entry is highlighted with a red box.

2. 在左侧导航栏点击 监控 -> 微服务监控。

The screenshot shows the DaoCloud interface with the 'Microservices Monitoring' section selected under the 'nacos-web-test' registration center. The left sidebar includes 'Basic Information', 'Microservice Namespace', 'Microservice List', 'Microservice Configuration List', 'Monitoring' (which is expanded), 'Component Monitoring', 'Link Tracing', and 'Resource List'. The right side shows four monitoring charts: 'Response Time' (a sharp spike at 14:00), 'Exception QPS' (multiple series with spikes around 16:40, 16:45, and 16:50), 'Concurrent Requests' (mostly flat with minor fluctuations), and 'Block QPS' (mostly flat with minor fluctuations). The 'Monitoring' section is highlighted with a red box.

3. 选择对应的微服务命名空间即可查看该命名空间下的微服务监控数据。

点击仪表盘名称可以进一步查看或分享该监控信息。在页面右上角可切换统计时间窗口和刷新周期。



链路追踪

DCE 5.0 支持服务级别的链路追踪与查询，查看各个微服务的链路时延分布情况，在出现故障时可以快速定位故障。

!!! note

集群中需要[安装 insight-agent 组件](../../../../insight/user-guide/quickstart/install-agent.md)后才能使用监控功能。

更多详情，可参考[可观测性-->链路查询](#)。

Plugins

启用 Mesh 治理插件

操作步骤

- 在托管注册中心列表页面点击目标注册中心的名称，然后。

The screenshot shows the DaoCloud management interface. On the left, there's a sidebar with links like 'Default', '微服务治理中心' (Microservices Governance Center), and '托管注册中心'. The main area is titled '托管注册中心列表' (Managed Registry Center List). It displays two entries: 'test-nacos-2' and 'test-nacos-1'. Each entry has details such as container cluster ('skoala-dev'), namespace ('skoala-test' and 'lfj-demo22'), version ('2.0.4'), creation time ('2022-11-14 17:02...'), and service counts ('0', '0', '1 / 1'). At the bottom, it says '共 2 项' (2 items total).

- 在左侧导航栏点击 插件中心，在 Mesh 治理 卡片上点击 立即开启。

This screenshot shows the 'Plugin Center' page for the 'test-nacos-2' registry center. The left sidebar has a 'Plugin Center' link highlighted with a red box. The main content area shows two plugin cards: 'Sentinel 治理' (Sentinel Governance) and 'Mesh 治理' (Mesh Governance). Both cards are currently '已停用' (Disabled). Each card has a '立即开启' (Enable Now) button, which is also highlighted with a red box.

- 选择想要绑定的服务网格，然后在弹框底部点击 确定。

如果找不到想要的服务网格，可以去服务网格模块创建一个网格。

Mesh 治理配置项

X

服务网格绑定

dce4



取消

确定

4. 如果满足前提条件并且配置正确，页面右上角会弹出“启用 Mesh 插件成功”的消息。

The screenshot shows the DaoCloud interface for managing a service mesh. On the left, there's a sidebar with navigation links like 'test-nacos-2', 'Default', '基础信息', '微服务命名空间', '微服务列表', '微服务配置列表', '监控', '资源列表', and '插件中心'. The main area is titled '托管注册中心: test-nacos-2 / 插件中心' and lists two plugins: 'Sentinel 治理' and 'Mesh 治理'. Both plugins have a status of '已停用' (Disabled) and a '立即开启' (Enable Now) button. A red box highlights a success message at the top right: '启用了 mesh 插件成功' (Mesh plugin enabled successfully), accompanied by a green checkmark icon.

插件中心

插件中心提供 Sentinel 治理和 Mesh 治理两种插件，支持通过用户界面实现可视化配置。安装插件后可以扩展微服务治理能力，满足不同场景下的业务诉求。

!!! info

同一个注册中心实例不能同时开启这两种插件，但可以根据不同的场景进行切换。

The screenshot shows the DaoCloud plugin center interface. At the top, there is a navigation bar with the DaoCloud logo, user information (admin), and a message icon. Below the navigation bar, the title is "托管注册中心: test-nacos-2 / 插件中心". On the left, there is a sidebar with the following menu items: Default, 基础信息, 微服务命名空间, 微服务列表, 微服务配置列表, 监控, 资源列表, and 插件中心 (which is highlighted in blue). The main content area contains two cards. The first card is for "Sentinel 治理", featuring a gear icon, a "立即开启" (Start Now) button, and a "已停用" (Disabled) status indicator. The second card is for "Mesh 治理", featuring a mesh icon, a "立即开启" (Start Now) button, and a "已停用" (Disabled) status indicator. Both cards have descriptive text below them: "这是插件介绍的文本 这是插件介绍的文本 这是插件介绍的文本 这是插件介绍的文本 这是插件介绍的文本 这是插件介绍的文本 这是插件介绍的文本...".

SENTINEL 治理

Sentinel 插件主要适用于传统微服务的治理场景，支持流控规则、熔断规则、热点规则、系统规则、授权规则等多种治理规则。

开启 Sentinel 治理插件后会创建一个 Sentinel 实例，需要为其设置资源配额、选择部署模式（单节点/高可用）和访问方式。

MESH 治理

Mesh 插件主要适用于云原生微服务的治理场景，提供虚拟服务、目标规则、网关规则、对等认证、请求身份认证、授权策略等治理规则。

开启 Mesh 治理插件需绑定一个网格实例，将微服务加入到网格中，并根据服务网格的要求配置边车等资源。

启用 Sentinel 治理插件

Sentinel 是面向分布式、多语言异构化服务架构的流量治理组件，主要以流量为切入点，从流量路由、流量控制、流量整形、熔断降级、系统自适应过载保护、热点流量防护等多个维度来帮助开发者保障微服务的稳定性。

操作步骤

- 在托管注册中心列表页面点击目标注册中心的名称，然后。

The screenshot shows the '托管注册中心列表' (Managed Registry Center List) page. On the left sidebar, '托管注册中心' (Managed Registry Center) is selected. In the main area, there are two entries: 'test-nacos-2' and 'test-nacos-1'. Each entry has a summary card with details like container group, namespace, version, creation time, and service counts. Below each card is a '立即开启' (Enable Now) button. A red box highlights the '立即开启' button for 'test-nacos-2'.

- 在左侧导航栏点击 插件中心，在 Sentinel 治理 卡片上点击 立即开启。

The screenshot shows the '托管注册中心: test-nacos-2 / 插件中心' (Managed Registry Center: test-nacos-2 / Plugins Center) page. The left sidebar shows a tree structure with 'test-nacos-2' expanded, and '插件中心' (Plugins Center) is highlighted with a red box. The main content area displays two plugin cards: 'Sentinel 治理' and 'Mesh 治理'. Both cards show a '已停用' (Disabled) status and a '立即开启' (Enable Now) button, which is also highlighted with a red box.

- 填写各项配置信息，然后在弹框底部点击 确定。

Sentinel 治理配置项

X

资源配置 * 自定义配置资源

CPU 配置

请求值 1 Core 限制值 1 Core

内存配额

请求值 1024 MiB 限制值 1024 MiB

《实例能力评估》

部署模式 * 单节点 高可用

节点数量

1

访问方式 * 节点访问 内部访问

服务端口 8080

目标端口 默认

随机范围: 30000-32767

取消

确定

4. 如果满足前提条件并且配置正确，页面右上角会弹出“启用 Sentinel 插件成功”的消息。

DaoCloud

托管注册中心: test-nacos-2 / 插件中心

启用 sentinel 插件成功

基础信息

微服务命名空间

微服务列表

微服务配置列表

监控

资源列表

插件中心

Sentinel 治理

立即开启

已停用

这是插件介绍的文本 这是插件介绍的文本 这是插件介绍的文本 这是插件介绍的文本 这是插件介绍的文本 这是插件介绍的文本 这是插件介绍的文本 这是插件介绍的文本 这是插件介绍的文本...

Mesh 治理

立即开启

已停用

这是插件介绍的文本 这是插件介绍的文本 这是插件介绍的文本 这是插件介绍的文本 这是插件介绍的文本 这是插件介绍的文本 这是插件介绍的文本 这是插件介绍的文本 这是插件介绍的文本...

Registry lcm

查看注册中心详情

在托管注册中心列表页找到需要查看详情的注册中心实例，点击实例名称进入基础信息页面。

在基础信息页面可以查看注册中心的基础信息、服务治理信息（需要开启服务治理功能）、节点列表、数据持久化信息等。“运行状态”是注册中心实例信息的一部分，用于反映注册中心实例的状态。

相关操作：

- 重启注册中心实例：在页面右上角点击 重启 可以对整个托管注册中心实例进行重启。

The screenshot shows the 'nacos-web' instance details in the DaoCloud interface. The '基础信息' (Basic Information) section displays the service name (nacos-web), version (2.0.3), status (运行中 - Running), and resource usage (1024 MiB / 1024 MiB memory, 1 Core / 1 Core CPU). The '服务治理' (Service Governance) section shows a single Sentinel node (IP: 10.6.222.24:49830, status: 运行中 - Running). The '节点列表' (Node List) section shows one node (nacos-web-0) in the '在线' (Online) state. A prominent blue '重启' (Restart) button is located in the top right corner of the main content area.

- 节点上下线：在节点列表中选择目标节点，修改节点状态。

• 下线

点击 操作 列的 下线 ，修改节点实例状态为“下线”

• 上线

点击 操作 列的 上线 ，修改节点实例状态为“上线”

This screenshot shows the same 'nacos-web' instance details as the previous one, but with a change in the '操作' (Operation) column for the node 'nacos-web-0'. The status has been changed from '在线' (Online) to '下线' (Offline), indicated by a red box. The rest of the interface remains identical to the first screenshot.

- 查看节点元数据

DaoCloud

nacos-web Default

托管注册中心：nacos-web / 基础信息

基础信息

| | | | |
|----------------|-------|---------------------|-----------------|
| nacos-web | 2.0.3 | ● 运行中 | 节点访问 |
| 实例名称 | 开源版本 | 运行状态 | 访问方式 |
| 单节点 | 1 | 1024 MiB / 1024 MiB | 1 Core / 1 Core |
| 部署模式 | 节点数量 | 内存申请值/限制值 | CPU申请值/限制值 |
| 8848/undefined | - | 2022-10-27 18:21 | - |
| 服务端口/目标端口 | IP地址 | 创建时间 | 控制台访问地址 |

数据持久化
不使用外置存储化
方式

服务治理

| | | | |
|------------|------------|-------------------|---------------------|
| Sentinel | 1 | 256 MiB / 512 MiB | 0.1 Core / 0.5 Core |
| 治理方式 | 节点数量 | 内存申请值/限制值 | CPU申请值/限制值 |
| 8848/49830 | ● 运行中 | 10.6.222.24:49830 | - |
| 服务端口/目标端口 | Sentinel状态 | 控制台访问地址 | - |

节点列表

| 节点 IP | 节点状态 | 角色 | 操作时间 | 操作 |
|-------------|------|----------|------|----------------|
| nacos-web-0 | ● 在线 | FOLLOWER | - | ⋮
元数据
下线 |

共 1 项

重启

创建托管注册中心

微服务引擎支持托管 Nacos 注册中心，即在微服务治理中心从零创建一个 Nacos 类型的全新注册中心，并且可以通过微服务治理中心全面管理该注册中心。相对于[接入型注册中心](#)而言，托管型注册中心支持更多操作，包括查看注册中心实例的基础信息、微服务命名空间管理、微服务列表、微服务配置列表、监控告警、日志查看、插件中心等。

!!! note

- 需要事先在目标集群的 `**skoala-system**` 命名空间中安装 `skoala-init` 组件，具体步骤可参考[\[管理 Helm 应用\]](#)(../../../../kpanda/user-guide/helm/helm-app.md)。
- 如果所选集群中没有 `**skoala-system**` 命名空间，可参考[\[创建命名空间\]](#)(../../../../kpanda/user-guide/namespaces/createns.md) 创建一个名为 `**skoala-system**` 的命名空间。

创建托管注册中心的步骤如下。

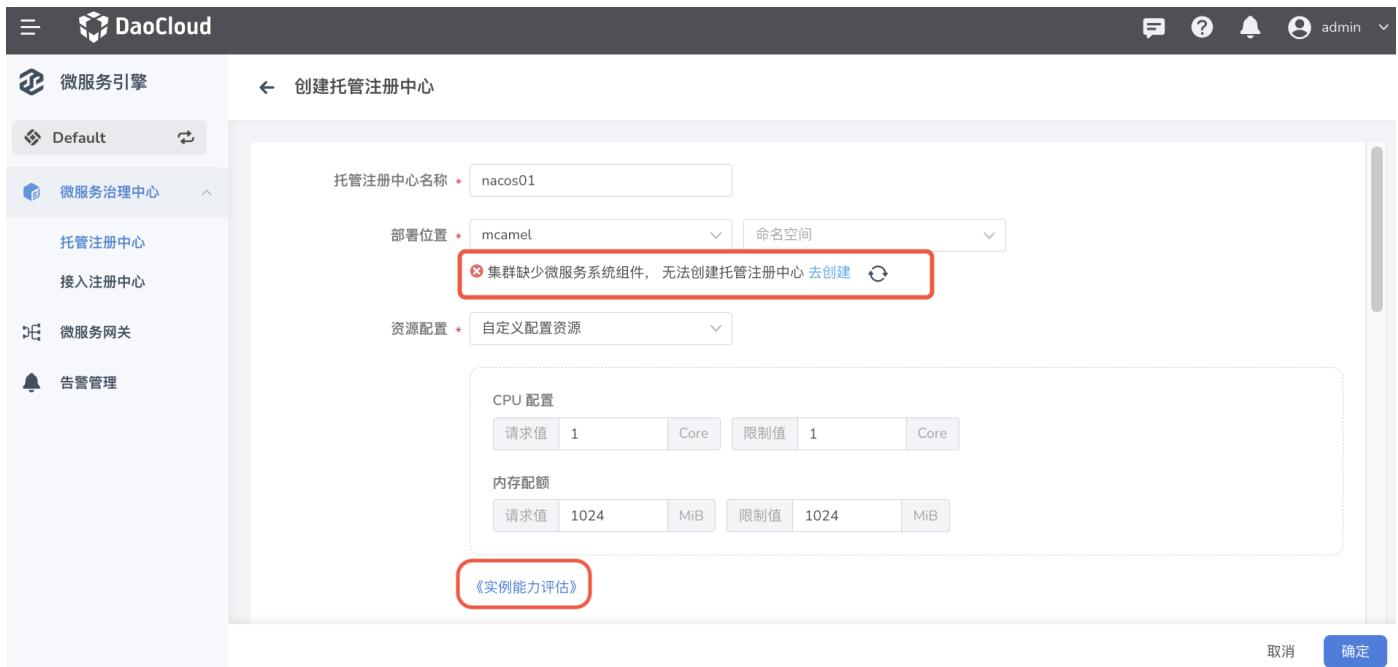
- 在左侧导航栏点击 微服务治理中心 -> 托管注册中心，然后在页面右上角点击 创建托管注册中心，进入托管注册中心实例的创建页面。

The screenshot shows the DaoCloud platform's 'Microservices Engine' section. On the left, there is a navigation sidebar with several items: 'Default' (selected), '微服务治理中心' (highlighted with a red box), '接入注册中心', '微服务网关', and '告警管理'. The main content area is titled '托管注册中心列表' (Managed Registry Center List). It features a search bar with dropdowns for '部署位置' (Deployment Location) and '实例名称' (Instance Name), and a '搜索' (Search) button. A large red box highlights the '创建托管注册中心' (Create Managed Registry Center) button in the top right corner. Below the search bar, there is a message '暂无数据' (No data available) next to a folder icon. At the bottom, it says '共 0 项' (0 items) and includes a pagination section with '1 / 0' and a dropdown for '10 项'.

- 填写配置信息。

需要注意的是：

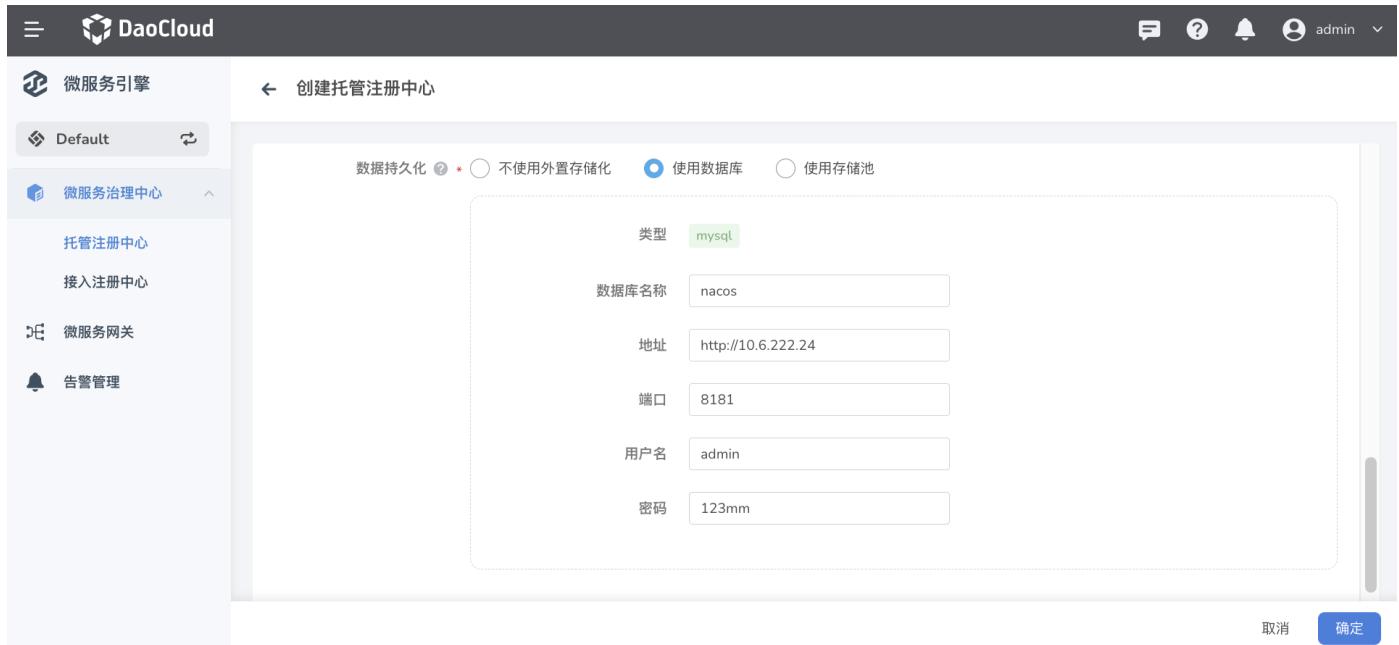
- 注册中心名称：支持输入字母、数字和分隔符（-），注册中心创建之后不可更改名称。
- 部署位置：系统会自动校验所选集群下的 skoala-system 命名空间中是否安装了 skoala-init 组件。
- 如未安装，则无法创建注册中心。可以根据页面提示去安装该组件。
- 注册中心创建之后不可更改部署位置。
- 资源配置：可直接选择 1核2G、2核4G 等配置，也可以自定义配置资源限额。
- 1核2G 指 CPU 的请求值和限制值分别为 2 核，内存的请求值和限制值分别为 2 G，以此类推。
- 点击《实例能力评估》可以查看在 2 Core 4 GiB、4 Core 8 GiB 和 8 Core 16 GiB 等主流规格下的吞吐量 (TPS)。



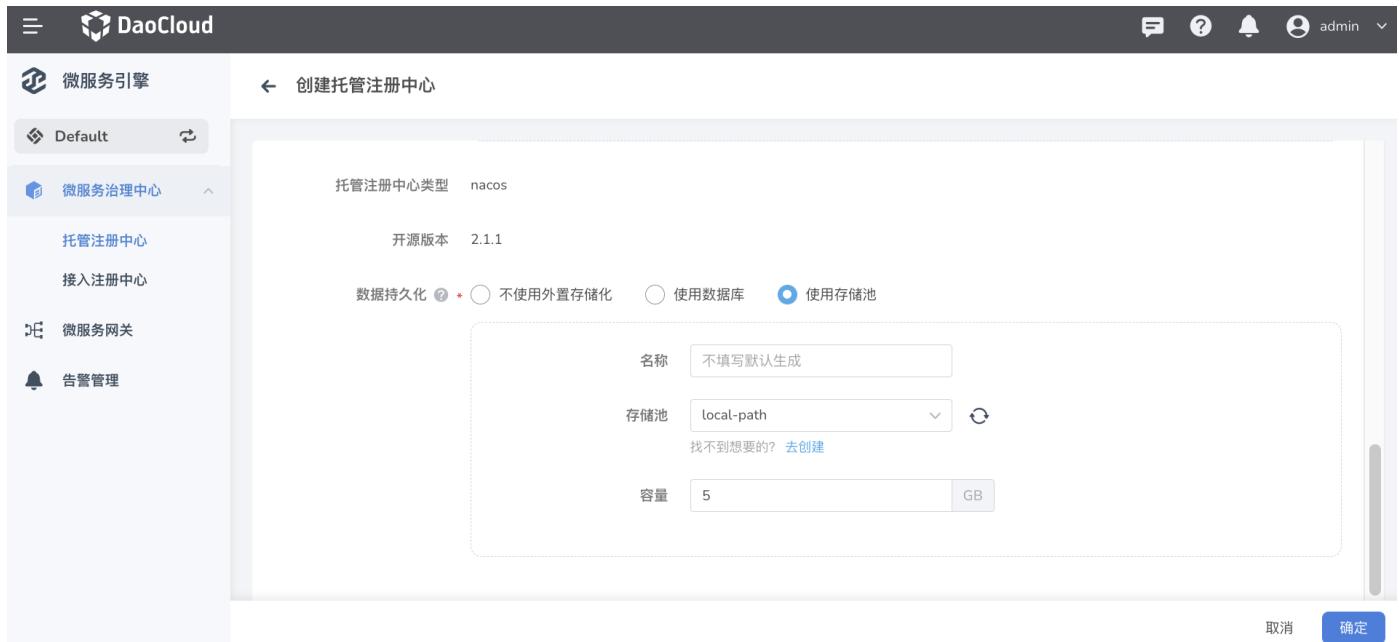
- 访问方式：选择 节点访问 可通过 服务端口+目标端口 的方式从外部访问注册中心，选择 内部访问 则只能在所在的集群范围内通过服务端口访问注册中心。默认的服务端口为 8848。
- 部署模式：选择高可用模式时，节点数量不少于 3 个。生产环境下建议使用 高可用 模式。



- 数据持久化：建议使用外置存储化。
- 不使用外置存储化：数据存储在 Nacos 所在的 Pod 的文件系统里。Pod 重启之后数据会丢失，因此建议使用外部存储。
- 使用数据库：填写数据库的名称、地址、端口、用户名和密码



- 使用存储池：选择注册中心所在集群中已有的存储池并配置存储容量。



3. 在页面底部点击 确定。

如果操作正确，页面右上角会弹出创建成功的消息，托管注册中心列表页会展示新建的注册中心实例。

The screenshot shows the DaoCloud platform interface. In the top right corner, there is a success message: "创建托管注册中心成功" (Create Managed Registry Center Success) with a green checkmark icon. This message is highlighted with a red rectangular box. The main area displays a table titled "托管注册中心列表" (Managed Registry Center List). A single row is present, labeled "nacos01" and marked as "启动中" (Starting). The table includes columns for Container Cluster (容器集群), Namespace (命名空间), Version (版本), IP Address (IP地址), and several summary metrics: 微服务总数 (0), 配置总数 (0), and 正常 / 节点总数 (0 / 0). The bottom of the table shows a pagination indicator "共 1 项" (1 item) and a dropdown for "10 项" (10 items).

!!! info

- 新建注册中心需要一段时间进行初始化，其间处于“启动中”状态。初始化完成后进入“运行中”状态。
- 点击注册中心的名称可以查看所在集群/命名空间、运行状态、资源配额、服务端口、存储配置、节点列表等基础信息。

删除注册中心

- 在托管注册中心列表页选择需要删除的注册中心，在右侧点击 **...** 并选择 **删除**。

- 输入注册中心的名称，点击 **移除**。

确定删除资源

X

确定要删除 nacos01 吗？删除后对应的数据将会全部丢失，请谨慎操作

请输入 nacos01

删除
取消

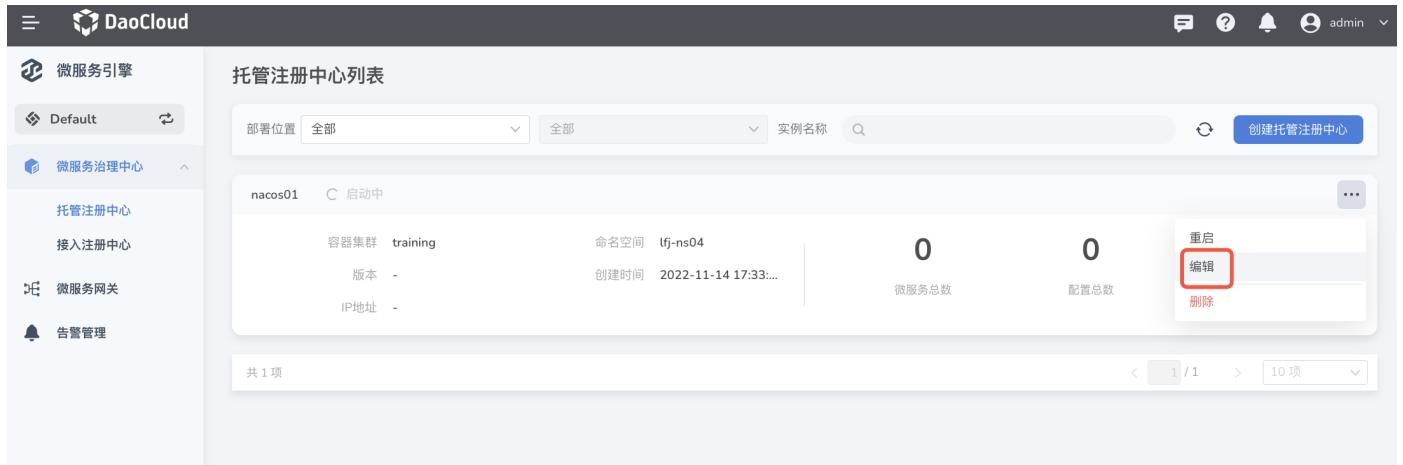
!!! note

接入型的注册中心仅支持‘移除’操作，而托管型的注册中心仅支持‘删除’操作。二者的区别在于：

- 移除：只是将注册中心从 DCE 5.0 的微服务引擎中移除，不会删除原有的注册中心和数据，后续还可以再次接入该注册中心。
- 删除：删除注册中心及其中的所有数据，后续无法再次使用该注册中心，需要重新创建新的注册中心。

更新托管注册中心

- 在接入注册中心列表页选择需要更新的注册中心，在右侧点击 ... 并选择 编辑。



The screenshot shows the '托管注册中心列表' (Managed Registry Center List) page. On the left sidebar, under '微服务引擎', 'Default' is selected. Under '微服务治理中心', '托管注册中心' is selected. The main area displays a single entry: 'nacos01' (启动中). Below it, detailed information is shown: 容器集群 'training', 命名空间 'lfj-ns04', 版本 ' - ', IP地址 ' - ', 创建时间 '2022-11-14 17:33...'. To the right of the entry, two large numbers '0' are displayed for 微服务总数 and 配置总数. At the bottom right of the entry card, there are three buttons: '重启', '编辑' (highlighted with a red box), and '删除'. At the very bottom of the page, there is a footer bar with navigation icons and the text '1 / 1'.

- 更新注册中心的配置，然后在页面底部点击 确定。

!!! warning

- 修改存储池/数据库，变动存储位置后，之前的数据不会随之迁移！
- 托管注册中心名称和部署位置不可编辑。

DaoCloud

微服务引擎

Default

微服务治理中心

托管注册中心

接入注册中心

微服务网关

告警管理

编辑托管注册中心

托管注册中心名称 * nacos01

部署位置 * training / lfj-ns04

资源配置 * 自定义配置资源

CPU 配置

请求值 1 Core 限制值 1 Core

内存配额

请求值 2048 MiB 限制值 2048 MiB

《实例能力评估》

访问方式 * 节点访问 内部访问

服务端口 8848

目标端口 默认 随机范围: 30000-31767

部署模式 * 单节点 高可用

节点数量 1

托管注册中心类型 nacos

开源版本 2.1.1

数据持久化 * 不使用外置存储化 使用数据库 使用存储池

名称 不填写默认生成

存储池 hwameistor-storage-lvm-hdd 去创建

容量 1 GB

取消 确定

Servicelist

查看微服务详情

在服务列表页面点击微服务名称可以查看服务详情，进一步查看实例列表、订阅者、监控、接口列表、元数据、服务治理等信息。

实例列表

首先需要进入目标注册中心，在左侧导航栏点击 [微服务列表](#)，点击目标微服务的名称，进入微服务详情页面后方可执行后续操作。

The screenshot shows the DaoCloud Service Center interface. On the left, there's a sidebar with navigation items: 'nacos-test' (selected), 'Default', '基础信息', '微服务命名空间' (selected), '微服务列表' (highlighted with a red box), '微服务配置列表', '监控' (selected), '资源列表', and '插件中心'. The main content area has a title '托管注册中心 nacos-test / 微服务列表'. It includes a search bar with 'public' selected and a '搜索' button. Below is a table with the following data:

| 微服务名称 | 分组 | 健康/实例数 | 实例组数 | 触发保护阀 | 请求量(次) | 请求成功率 | 请求平均耗 | 链路追踪 | 是否可以治 |
|------------------------|------------|--------|------|-------|--------|---------|-------|------|-------|
| my-otel-demo-adservice | DEFAULT... | 1/1 | 1 | 否 | 0 | 100.00% | 0.00 | ● 接入 | 是 |

At the bottom, it says '共 1 项' and shows page navigation '1 / 1'.

- 服务流量权重的调整

提供流量权重控制的能力，同时开放服务流量的阈值保护，帮助用户更好的地保护服务服务提供者集群不被意外打垮。可以点击实例的编辑按钮，修改实例的权重。如果想增加实例的流量，可以将权重调大，如果不想实例接收流量，则可以将权重设为 0。

| IP | 实例分组 | 端口 | 注册发现... | 上下线状... | 临时实例 | 权重 | 心跳间隔... | 心跳超时... | 服务治理... |
|---------|---------|------|--|---|------|----|---------|---------|---------|
| 10.2... | DEFAULT | 8443 | ● 运行中 | ● 在线 | 是 | 1 | 5000 | 15000 | ● 未知 |

- 实例上下线

提供服务实例的上下线操作，在实例列表的操作列，可以点击实例的 [上线](#) 或者 [下线](#) 按钮。下线的实例不会被包含在健康的实例列表里。

托管注册中心：nacos-test / 微服务列表 / my-otel-demo-adservice

基础信息

微服务命名空间

微服务列表

微服务配置列表

监控

资源列表

插件中心

服务信息

| my-otel-demo-adservice | DEFAULT_GROUP | 1 | 1 | 1 |
|------------------------|---------------|----------|------------|------|
| 微服务名称 | 分组 | 实例组数 | 健康实例数 | 总实例数 |
| - | 0 | 100.00% | 0.00 | |
| 触发保护阈值 | 请求量(次) | 请求成功率(%) | 请求平均耗时(ms) | |

实例列表 订阅者 监控 接口列表 元数据 服务治理

| IP | 实例分组 | 端口 | 注册发现... | 上下线状... | 临时实例 | 权重 | 心跳间隔... | 心跳超时... | 服务治理... |
|---------|---------|------|---------|---------|------|----|---------|---------|---------|
| 10.2... | DEFAULT | 8443 | ● 运行中 | ● 在线 | 是 | 1 | 5000 | 15000 | ● 未知 |

共 1 项

下线

编辑

查看日志

- 实例详情

点击实例名称，可进入实例详情，查看实例的监控和元数据。

The screenshot shows the DaoCloud Service Center interface. On the left, there's a sidebar with navigation items: nacos-test (selected), Default, 基础信息, 微服务命名空间, 微服务列表 (selected), 微服务配置列表, 监控, 资源列表, and 插件中心. The main content area has a title: 托管注册中心: nacos-test / 微服务列表 / my-otel-demo-adservice. It includes sections for 实例分组, 注册发现健康状态, and 服务治理健康状态. Below this is a '服务信息' section for the service 'my-otel-demo-adservice'. The table shows:

| 微服务名称 | DEFAULT_GROUP | 实例组数 | 健康实例数 | 总实例数 |
|--------|---------------|---------|-------|--------------|
| - | 分组 | 1 | 1 | 1 |
| 触发保护阈值 | 0 | 100.00% | 0.00 | 请求平均耗时(ms) ② |

Below the service information is a tab bar with '实例列表' (selected), 订阅者, 监控, 接口列表, 元数据, and 服务治理. The '实例列表' tab shows a table of service instances:

| IP | 实例分组 | 端口 | 注册发现... | 上下线状态 | 临时实例 | 权重 | 心跳间隔... | 心跳超时... | 服务治理... |
|---------------|---------|------|---------|-------|------|----|---------|---------|---------|
| 10.244.184.68 | DEFAULT | 8443 | ● 运行中 | ● 在线 | 是 | 1 | 5000 | 15000 | ● 未知 |

At the bottom of the instance table, it says '共 1 项' (1 item). There are also navigation controls like <, >, 1 / 1, and a dropdown for '10 项'.

- 实例监控

实例监控功能用于监控服务实例的状态，例如服务实例的请求数、错误率、响应耗时、请求率、CPU 指标、内存指标、读写速率、接收发送速率等指标的时序曲线。

响应耗时中 p95 代表线上 95% 的请求耗时都小于某个时间。

The screenshot shows the DaoCloud Service Center interface. On the left, there's a sidebar with navigation items: nacos-test (selected), Default, 基础信息, 微服务命名空间, 微服务列表 (selected), 微服务配置列表, 监控 (selected), 资源列表, and 插件中心. The main content area has a breadcrumb path: 托管注册中心: nacos-test / 微服务列表 / my-otel-demo-adservice. It includes sections for 实例分组, 注册发现健康状态, and 服务治理健康状态. Below these is a '服务信息' section with details for 'my-otel-demo-adservice': 微服务名称 (my-otel-demo-adservice), 分组 (DEFAULT_GROUP), 实例组数 (1), 健康实例数 (1), 总实例数 (1). It also shows metrics: 触发保护阈值 (0), 请求量(次) (0.00), 请求成功率(%) (100.00%), and 请求平均耗时(ms) (0.00). At the bottom, there's a table titled '实例列表' with columns: IP, 实例分组, 端口, 注册发现..., 上下线状态, 临时实例, 权重, 心跳间隔..., 心跳超时..., and 服务治理... . A single row is shown, with the IP '10.244.184.68' highlighted by a red box.

- 实例元数据

The screenshot shows the DaoCloud Servicelist interface. On the left, there's a sidebar with navigation items like 'nacos-test' (selected), 'Default', '基础信息', '微服务命名空间', '微服务列表' (selected), '微服务配置列表', '监控', '资源列表', and '插件中心'. The main content area is titled '托管注册中心: nacos-test / 微服务列表 / my-otel-demo-adservice'. It has sections for '实例分组', '注册发现健康状态', and '服务治理健康状态'. Under '服务信息', it shows a table for 'my-otel-demo-adservice' with columns: 微服务名称 (my-otel-demo-adservice), 分组 (DEFAULT_GROUP), 实例组数 (1), 健康实例数 (1), 总实例数 (1). Below this is a table for '触发保护阈值' with columns: 请求量(次) (0), 请求成功率(%) (100.00%), and 请求平均耗时(ms) (0.00). At the bottom, tabs for '实例列表' (selected), '订阅者', '监控', '接口列表', '元数据', and '服务治理' are shown. A detailed table for '实例列表' shows one row: IP (10.244.184.68), 实例分组 (DEFAULT), 端口 (8443), 注册发现... (运行中), 上下线状态 (在线), 临时实例 (是), 权重 (1), 心跳间隔... (5000), 心跳超时... (15000), 服务治理... (未知). The IP '10.244.184.68' is highlighted with a red box.

订阅者

进入订阅者列表页面，可查看到订阅者的信息，包括 IP 和端口、客户端版本和应用名。

This screenshot shows the '订阅者' (Subscribers) tab from the previous interface. The top navigation and sidebar are identical. The main content area is titled '托管注册中心: nacos-test / 微服务列表 / my-otel-demo-adservice'. It includes sections for '实例分组', '注册发现健康状态', and '服务治理健康状态'. The '服务信息' section is similar to the previous screenshot. The '订阅者' tab is selected at the bottom. A detailed table for '订阅者' shows one row: 地址 (10.244.184.68), 客户端版本 (Nacos-Java-Client:v2.0.3), and 订阅方的服务名 (DEFAULT_GROUP@@my-otel-demo-adservice). The '地址' column is highlighted with a red box.

服务监控

服务监控用于查看特定时间范围内，某个命名空间下微服务的运行状态，并且根据微服务的监控指标（请求数、错误率、响应耗时、请求率）初步判断是否出现异常。

在监控图表的右上角可以更改监控数据的时间范围。

The screenshot shows the service monitoring interface for the 'my-otel-demo-adservice' microservice. The left sidebar includes sections for '基础信息' (Basic Information), '微服务命名空间' (Microservice Namespace), '微服务列表' (Microservice List), '微服务配置列表' (Microservice Configuration List), '监控' (Monitoring), and '资源列表' (Resource List). The main content area displays service information and a monitoring chart. The chart tracks requests per second, error rate, and response time over a specified time range. A red box highlights the time selection bar at the top of the chart area.

接口列表

接口列表显示服务对外暴露的 API 列表。点击右侧的 创建 可以手动录入 API，

The screenshot shows the service interface list for the 'my-otel-demo-adservice' microservice. The left sidebar includes sections for '基础信息' (Basic Information), '微服务命名空间' (Microservice Namespace), '微服务列表' (Microservice List), '微服务配置列表' (Microservice Configuration List), '监控' (Monitoring), and '资源列表' (Resource List). The main content area displays service information and an interface list. A red box highlights the '创建' (Create) button located at the bottom right of the interface list area.

微服务元数据

提供多个维度的服务元数据，帮助用户存储自定义的信息。这些信息都是以 Key-Value 的数据结构存储，在控制台上会以 k1=v1, k2=v2 这样的格式展示。

在右侧点击 编辑 可以修改元数据。

The screenshot shows the DaoCloud service metadata management interface. On the left, there's a sidebar with navigation links: 'nacos-test' (selected), 'Default', '基础信息', '微服务命名空间', '微服务列表' (selected), '微服务配置列表', '监控' (selected), and '资源列表'. The main content area has tabs: '实例分组', '注册发现健康状态', '服务治理健康状态', '服务信息', '实例列表', '订阅者', '监控', '接口列表', '元数据' (selected), and '服务治理'. Below these tabs is a table showing service metadata. At the bottom right of the main content area is a blue 'Edit' button.

微服务治理

开启微服务治理插件后，可以通过 YAML 文件或页面表单为服务创建虚拟服务、目标规则、网关规则等三种治理规则。有关微服务治理的更多说明，可参考[流量治理](#)。

The screenshot shows the DaoCloud service governance management interface. The sidebar and tabs are identical to the previous screenshot. The main content area has tabs: '实例分组', '注册发现健康状态', '服务治理健康状态', '服务信息', '实例列表', '订阅者', '监控', '接口列表', '元数据', and '服务治理' (selected). Below these tabs is a table showing service governance rules. At the bottom right of the main content area is a blue 'Create' button.

微服务列表

微服务列表页面列出了当前注册中心实例下的所有微服务，可以查看微服务的分组、健康状况、保护阈值、请求状况、链路追踪、治理状况等。在列表中点击微服务名称后，还可以进一步查看微服务实例列表、监控信息、接口列表、元数据等。

The screenshot shows the DaoCloud service list interface. On the left, there's a sidebar with navigation links: nacos-web (selected), Default, 基础信息, 微服务命名空间, 微服务列表 (highlighted in blue), 微服务治理策略, 微服务配置列表, 监控告警, 日志查看, and 插件中心. The main area has a title "托管注册中心: nacos-web / 微服务列表". It includes a search bar with dropdowns for "public" and "搜索", and a refresh button. Below is a table with columns: 微服务名称, 分组, 健康/实例..., 实例组数, 触发保护..., 请求量(次...), 请求成功..., 请求平均..., 链路追踪, 是否可以..., 操作. Two rows are listed: "test-weif..." and "sentinel-...". Both rows show "DEFAULT..." as the group, "1/1" as the instance count, and "否" (No) as the health status. The "操作" column contains a green dot icon followed by "接入" and a three-dot menu icon. At the bottom, it says "共 2 项" and shows page 1 of 1 with a "10 项" dropdown.

分组：指微服务的配置文件分组。将配置文件和微服务实例互相分离，便于为不同的服务或组件应用相同的配置。

保护阈值：保护阈值是 0 到 1 之间的一个浮点数，表示集群中健康实例占比的最小值。如果实际健康实例的占比小于或等于该阈值时，就会触发阈值保护，即无论实例（Instance）是否健康，都会将这个实例（Instance）返回给客户端。阈值保护主要是为了防止故障实例过多时所有流量全部流入剩余实例，造成流量压力将剩余实例被压垮形成雪崩效应。

链路追踪：通过[应用工作台](#)创建服务时，可以选择是否接入 [OpenTelemetry](#) 链路追踪组件。如果服务是通过其他途经注册到注册中心的，则可以参考[使用 OTel SDK 为应用程序暴露指标](#)手动接入该组件。建议接入，以便获取完整的可观测信息。

是否可以治理：判断该微服务是否满足治理条件，例如是否开启了治理插件、是否进行了容器化、是否被网格纳管、是否通过[应用工作台](#)创建的一系列要求。