

# Spring Cloud Alibaba 开源项目揭秘

肖京

# 个人简介

- 姓名 肖京 阿里巴巴中间件 高级开发工程师 花名 亦盏
- 主要负责 Spring Cloud Alibaba 开源项目，以及阿里云企业级分布式应用服务（Enterprise Distributed Application Service, 简称 EDAS）中 Dubbo 和 Spring Cloud 商业化部分的工作。

# 目录

为什么要做 Spring Cloud Alibaba

Spring Cloud Alibaba 路线图

Spring Cloud Alibaba 组件简介

Demo 演示

# 为什么要做 Spring Cloud Alibaba

# spring cloud 简介

Spring: the source for modern java



## Spring Boot

### BUILD ANYTHING

Spring Boot is designed to get you up and running as quickly as possible, with minimal upfront configuration of Spring. Spring Boot takes an opinionated view of building production ready applications.

## Spring Cloud

### COORDINATE ANYTHING

Built directly on Spring Boot's innovative approach to enterprise Java, Spring Cloud simplifies distributed, microservice-style architecture by implementing proven patterns to bring resilience, reliability, and coordination to your microservices.

## Spring Cloud Data Flow

### CONNECT ANYTHING

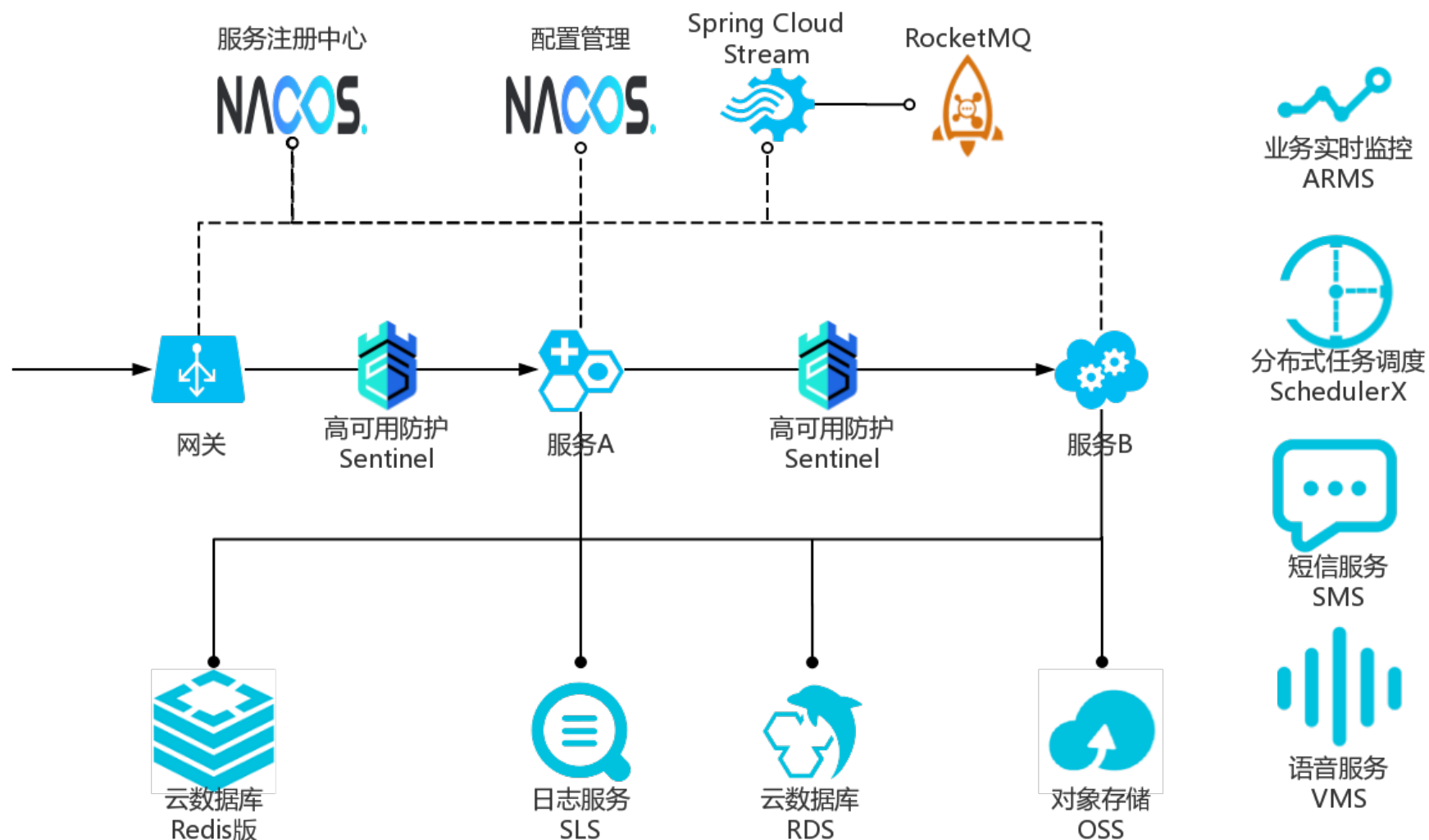
Connect the Enterprise to the Internet of Anything-mobile devices, sensors, wearables, automobiles, and more. Spring Cloud Data Flow provides a unified service for creating composable data microservices that address streaming and ETL-based data processing patterns.

# 为什么要做 Spring Cloud Alibaba

- 阿里全面拥抱开源，将阿里多年技术经验回馈给社区。
  - RocketMQ、Dubbo、Nacos、Sentinel、Pouch、Dragonfly
- Spring Cloud 在微服务集成领域影响力很大。
- 让 Spring Cloud 用户更方便地使用阿里云。

# Spring Cloud Alibaba 路线图

# spring cloud alibaba 生态图





# spring cloud alibaba roadmap

- |                   |          |   |
|-------------------|----------|---|
| ➤ Sentinel        | 限流降级     | ✓ |
| ➤ Nacos Discovery | 服务发现     | ✓ |
| ➤ Nacos Config    | 服务配置     | ✓ |
| ➤ 阿里云 OSS         | 对象存储     | ✓ |
| ➤ RocketMQ        | 消息       |   |
| ➤ 阿里云 SLS         | 日志管理     |   |
| ➤ 阿里云 SchedulerX  | 分布式任务调度  |   |
| ➤ 阿里云 ARMS        | 业务实时监控诊断 |   |
| ➤ .....           |          |   |

# Spring Cloud Alibaba 组件简介

# Spring Cloud Nacos Config 分布式配置管理

- 分布式配置管理让你在修改配置时，无需再修改代码或配置文件。
- 标准的 Spring Cloud 实现，开箱即用
- 一站式解决方案，无需 git、spring cloud bus、kafka
- 支持多种数据格式，完美支持中文
- 实时推送，推送状态可查
- 支持历史配置查询，支持回滚
- 基于阿里内部配置管理组件开源，经历过多次双十一检验，支持超大规模集群

# Spring Cloud Nacos Discovery 服务注册与发现

- 服务注册与发现让你在服务扩缩容时无需再关心服务调用。
- 标准的 Spring Cloud 实现，开箱即用。
- 支持实时推送，秒级服务发现。
- 多层容灾机制，尽量保证服务发现中心宕机不影响应用调用。
- 基于阿里内部服务发现组件开源，经历过多次双十一检验，支持超大规模集群。

# Spring Cloud Sentinel 限流降级保护服务

- 在流量洪峰到达或下游服务较差时主动限流降级保护自己。
- 默认集成 servlet、dubbo、RestTemplate、feign(doin) ，支持自定义限流降级
- 支持控制台动态配置限流降级规则，支持多种外部配置源
- 完备的监控功能，限流降级监控可以实时查看
- 轻量级，核心模块不依赖其他 jar 包；
- 性能损耗小，单机 4 万 QPS 时性能损耗 2.36%
- 基于阿里内部服务发现组件开源，经历过多次双十一检验，每天上亿次调用

# Demo 演示

# Nacos Config 获取配置

## 新建配置

\* Data ID:

service-provider.yaml

更多高级选项

描述:

配置格式:

☐ TEXT ☐ JSON ☐ XML ☒ YAML ☐ HTML ☐ Properties

\* 配置内容: ? :

```
1 user:
2   name: 小明
3   age: 15
```

← → ↺

127.0.0.1:18082/user

Hello Nacos Config!Hello 小明 15!

# Nacos Config 修改配置

内容比较



当前值		原始值	
1	user:	1	user:
2	name: 小明	2	name: 小明
3	age: 16	3	age: 15



127.0.0.1:18082/user

Hello Nacos Config!Hello 小明 16!

确认发布



# Nacos Config 配置回滚

## 配置回滚

\* Data ID:

service-provider.yaml

更多高级选项

\* 操作类型:

更新

\* MD5:

4857add51fef39ff05aa0febbabda4f2

\* 配置内容:

user: name: 小明 age: 15

回滚

返回

?

回滚配置

确定要 以下配置吗?

Data ID: service-provider.yaml

Group: DEFAULT\_GROUP

确认

取消

←

→

↺

i

127.0.0.1:18082/user

Hello Nacos Config!Hello 小明 15!

# Nacos Config Endpoint 查看记录

← → ↺ ⓘ 127.0.0.1:18082/actuator/nacos-config

```
{
  + NacosConfigProperties: {...},
  - RefreshHistory: [
    - {
      timestamp: "2018-10-11 17:21:32",
      dataId: "service-provider.yaml",
      md5: "4857add51fef39ff05aa0febbabda4f2"
    },
    - {
      timestamp: "2018-10-11 17:21:21",
      dataId: "service-provider.yaml",
      md5: "da15035991cbbbf1289e34e9d38c2c71"
    },
    - {
      timestamp: "2018-10-11 17:21:07",
      dataId: "service-provider.yaml",
      md5: "4857add51fef39ff05aa0febbabda4f2"
    }
  ],
  - Sources: [
    - {
      lastSynced: "2018-10-11 17:21:32",
      dataId: "service-provider.yaml"
    }
  ]
}
```

# Nacos Config 控制台查询推送状态

## 配置详情

\* Data ID:

service-provider.yaml

更多高级选项

描述:

\* MD5:

4857add51fef39ff05aa0febbabda4f2

\* 配置内容:

user: name: 小明 age: 15

返回

## 监听查询

Public

查询维度:

配置

Data ID:

service-provider.yaml

Group:

DEFAULT\_GROUP

查询

查询结果：共查询到 条满足要求的配置。

IP	MD5
127.0.0.1	4857add51fef39ff05aa0febbabda4f2

# Nacos Discovery 发现服务

```
@Bean
@LoadBalanced
public RestTemplate restTemplate() {
    return new RestTemplate();
}

@FeignClient(name = "service-provider")
public interface EchoService {
    @RequestMapping(value = "/user", method = RequestMethod.GET)
    String user();
}
```

```
@RequestMapping(value = "/user/rest", method = RequestMethod.GET)
public String rest() {
    return restTemplate.getForObject("http://service-provider/user/", String.class);
}

@RequestMapping(value = "/user/feign", method = RequestMethod.GET)
public String feign() {
    return echoService.user();
}
```

← → ↻ ⓘ 127.0.0.1:18081/user/rest

Hello Nacos Config!Hello 小明 15!

← → ↻ ⓘ 127.0.0.1:18081/user/feign

Hello Nacos Config!Hello 小明 15!

# Nacos Discovery Endpoint 查看

<http://127.0.0.1:18081/actuator/nacos-discovery>

```
- subscribe: [
  - {
    jsonFromServer: "{\"dom\":\"service-provider\",\"cacheMillis\":10000,\"provider\",\"port\":18082,\"ip\":\"30.5.125.0\",\"weight\":1.0}],[\"checksum\": \"dc5ff4415e7a559b8ec7b2f69de26a331539250981434\", \"allIps\": false, \"key\": \"service-provider\", \"valid\": true\", \"name\": \"service-provider\", \"clusters\": \"\", \"cacheMillis\": 10000,
    - hosts: [
      - {
        instanceId: \"30.5.125.0#18082#DEFAULT#service-provider\",
        ip: \"30.5.125.0\",
        port: 18082,
        weight: 1,
        healthy: true,
        - cluster: {
          serviceName: null,
          name: \"\",
          - healthChecker: {
            type: \"TCP\"
          },
          defaultPort: 80,
          defaultCheckPort: 80,
          useIPPort4Check: true,
          metadata: { }
        },
        service: null,
        metadata: { }
      },
    ],
    lastRefTime: 1539250981434,
    checksum: \"dc5ff4415e7a559b8ec7b2f69de26a331539250981434\",
    allIps: false,
    key: \"service-provider\",
    valid: true
  },
],
```

```
- NacosDiscoveryProperties: {
  serverAddr: \"127.0.0.1:8848\",
  endpoint: \"\",
  namespace: \"\",
  logName: \"\",
  service: \"service-consumer\",
  weight: 1,
  clusterName: \"DEFAULT\",
  metadata: { },
  registerEnabled: true,
  ip: \"30.5.125.0\",
  networkInterface: \"\",
  port: 18081,
  secure: false,
  accessKey: \"\",
  secretKey: \"\"
}
```

[illegible]

时间	p_qps	b_qps	rt(ms)
18:45:31	8.0	0.0	0.4
18:45:30	8.0	0.0	0.1
18:45:29	8.0	0.0	0.1
18:45:28	8.0	0.0	0.1
18:45:27	7.0	0.0	0.1
18:45:26	7.0	0.0	0.6

# Sentinel 添加限流规则

## 新增流控规则



资源名

/user/

流控应用

default

阈值类型

☒ QPS ☐ 线程数

单机阈值

3

高级选项

新增并继续添加

新增

取消

18:48:36

Blocked by Sentinel (flow limiting)

18:48:36

Blocked by Sentinel (flow limiting)

18:48:36

Blocked by Sentinel (flow limiting)

18:48:36

Blocked by Sentinel (flow limiting)

18:48:36

Hello Nacos Config!Hello 小明 15!

18:48:36

Hello Nacos Config!Hello 小明 15!

18:48:36

Hello Nacos Config!Hello 小明 15!

18:48:36

Blocked by Sentinel (flow limiting)

# Sentinel 添加限流规则

时间	p_qps	b_qps	rt(ms)
18:50:11	3.0	5.0	0.3
18:50:10	3.0	5.0	0.3
18:50:09	3.0	5.0	0.3
18:50:08	3.0	5.0	0.3
18:50:07	3.0	5.0	0.3
18:50:06	3.0	5.0	0.3



# Sentinel Endpoint 查看

← → ↻ ⓘ 127.0.0.1:18082/actuator/sentinel

```
{
  DegradeRules: [ ],
  SystemRules: [ ],
  - FlowRules: [
    - {
      resource: "/user/",
      limitApp: "default",
      grade: 1,
      count: 3,
      strategy: 0,
      refResource: null,
      controlBehavior: 0,
      warmUpPeriodSec: 10,
      maxQueueingTimeMs: 500
    }
  ],
}
```

```
- properties: {
  eager: false,
  enabled: true,
  charset: "UTF-8",
  - transport: {
    port: "8721",
    dashboard: "127.0.0.1:9090",
    heartbeatIntervalMs: null
  },
  - metric: {
    fileSingleSize: null,
    fileTotalCount: null
  },
  - servlet: {
    blockPage: null
  },
  - filter: {
    order: -2147483648,
    - urlPatterns: [
      "/"
    ]
  },
  - flow: {
    coldFactor: "3"
  }
}
}
```

# 谢谢



扫码添加微信  
加入Spring Cloud Alibaba 交流群



关注阿里巴巴中间件微信公众号  
加入开发者技术交流群

项目地址 <https://github.com/spring-cloud-incubator/spring-cloud-alibaba>

Spring Cloud Alibaba QQ群: 819908813

Nacos QQ 群: 807092827

Sentinel 钉钉群: 21977771

钉钉群: 21914947

钉钉群: 21708933