基于 Spring Cloud 完整的微服务架构实战

2017-12-24 鹏磊 开源中国

作者：鹏磊

首发：搜库云

本项目是一个基于 Spring Boot、Spring Cloud、Spring Oauth2 和 Spring Cloud Netflix 等框架构建的微服务项目。

技术栈

- Spring boot - 微服务的入门级微框架，用来简化 Spring 应用的初始搭建以及开发过程。
- Eureka - 云端服务发现，一个基于 REST 的服务，用于定位服务，以实现云端中间层服务发现和故障转移。
- Spring Cloud Config - 配置管理工具包，让你可以把配置放到远程服务器，集中化管理集群配置，目前支持本地存储、Git 以及 Subversion。
- Hystrix - 熔断器，容错管理工具，旨在通过熔断机制控制服务和第三方库的节点,从而对延迟和故障提供更强大的容错能力。
- Zuul - Zuul 是在云平台上提供动态路由，监控，弹性，安全等边缘服务的框架。Zuul 相当于是设备和 Netflix 流应用的 Web 网站后端所有请求的前门。
- Spring Cloud Bus - 事件、消息总线，用于在集群（例如，配置变化事件）中传播状态变化，可与 Spring Cloud Config 联合实现热部署。
- Spring Cloud Sleuth - 日志收集工具包，封装了 Dapper 和 log-based 追踪以及 Zipkin 和 HTrace 操作，为 SpringCloud 应用实现了一种分布式追踪解决方案。
- Ribbon - 提供云端负载均衡，有多种负载均衡策略可供选择，可配合服务发现和断路器使用。
- Turbine - Turbine 是聚合服务器发送事件流数据的一个工具，用来监控集群下 hystrix 的 metrics 情况。

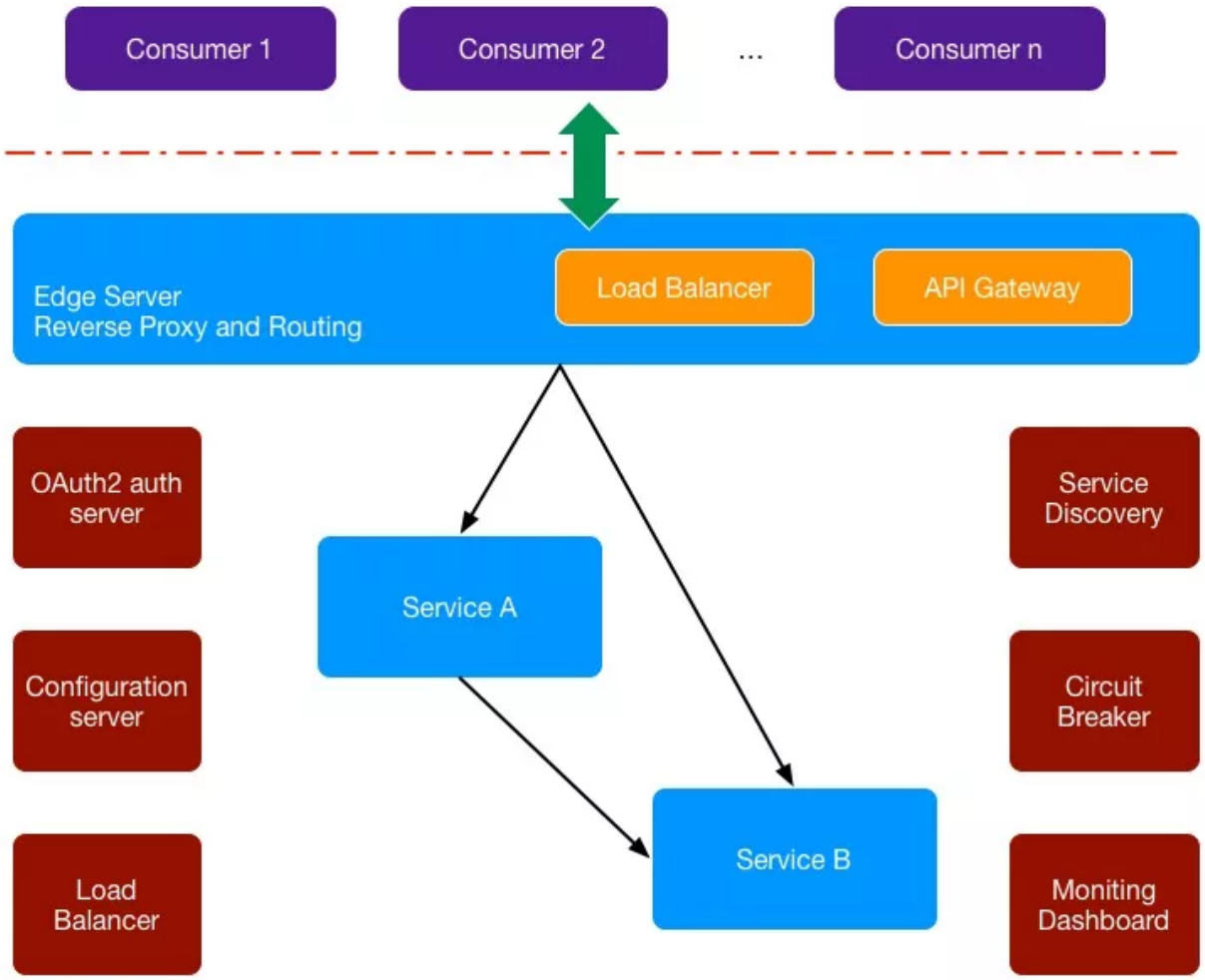
- Spring Cloud Stream - Spring 数据流操作开发包，封装了与 Redis、Rabbit、Kafka 等发送接收消息。
- Feign - Feign 是一种声明式、模板化的 HTTP 客户端。
- Spring Cloud OAuth2 - 基于 Spring Security 和 OAuth2 的安全工具包，为你的应用程序添加安全控制。

应用架构

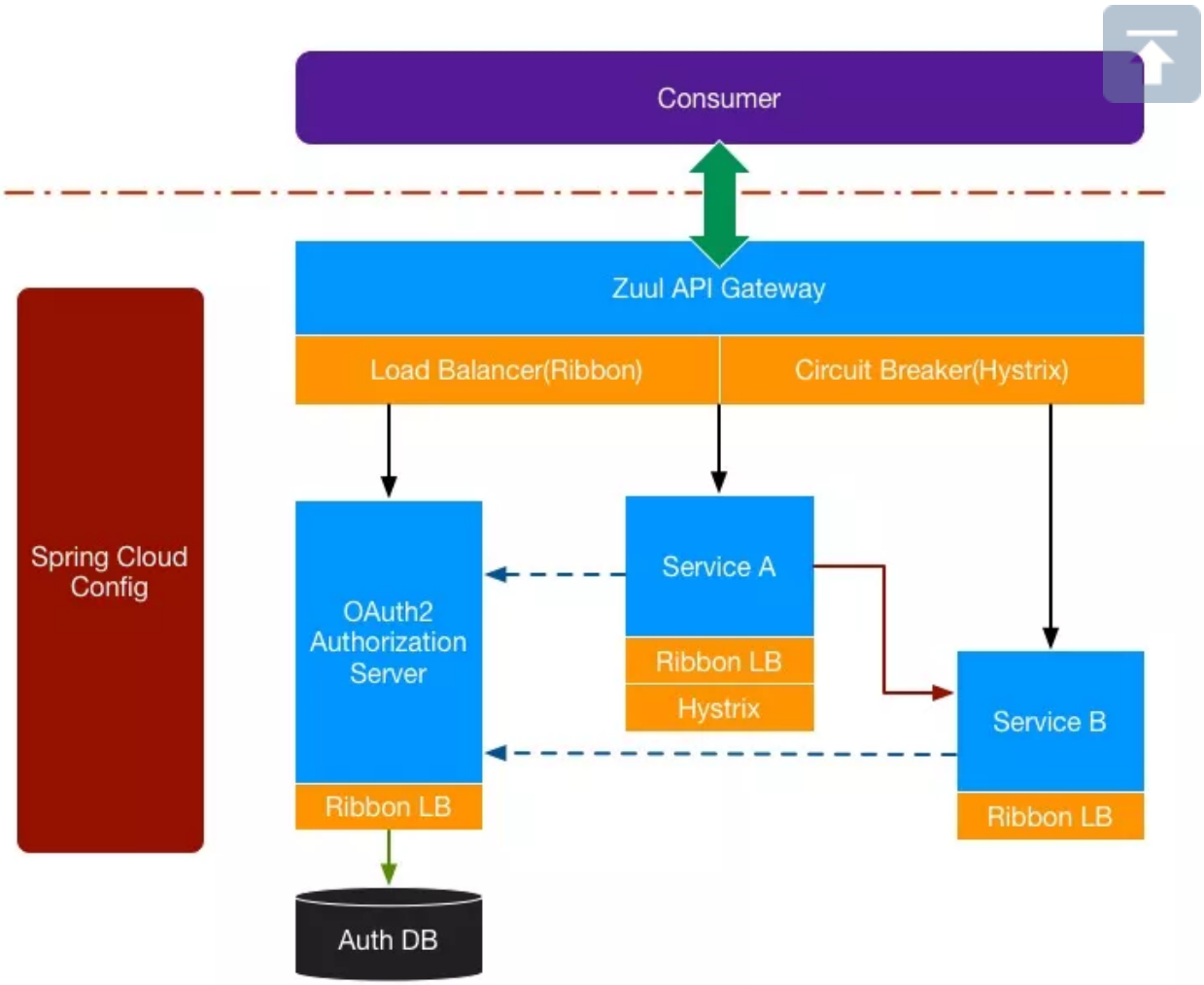
该项目包含 8 个服务

- registry - 服务注册与发现
- config - 外部配置
- monitor - 监控
- zipkin - 分布式跟踪
- gateway - 代理所有微服务的接口网关
- auth-service - OAuth2 认证服务
- svca-service - 业务服务A
- svcb-service - 业务服务B

体系架构


Consumer 1
Consumer 2
...
Consumer n
Edge Server
Reverse Proxy and Routing
Load Balancer
API Gateway
OAuth2 auth server
Service Discovery
Service A
Configuration server
Circuit Breaker
Service B
Load Balancer
Moniting Dashboard

应用组件



启动项目

- 使用 Docker 快速启动
    1. 配置 Docker 环境
    2. mvn clean package 打包项目及 Docker 镜像
    3. 在项目根目录下执行 docker-compose up -d 启动所有项目
- 本地手动启动
    1. 配置 rabbitmq
    2. 修改 hosts 将主机名指向到本地

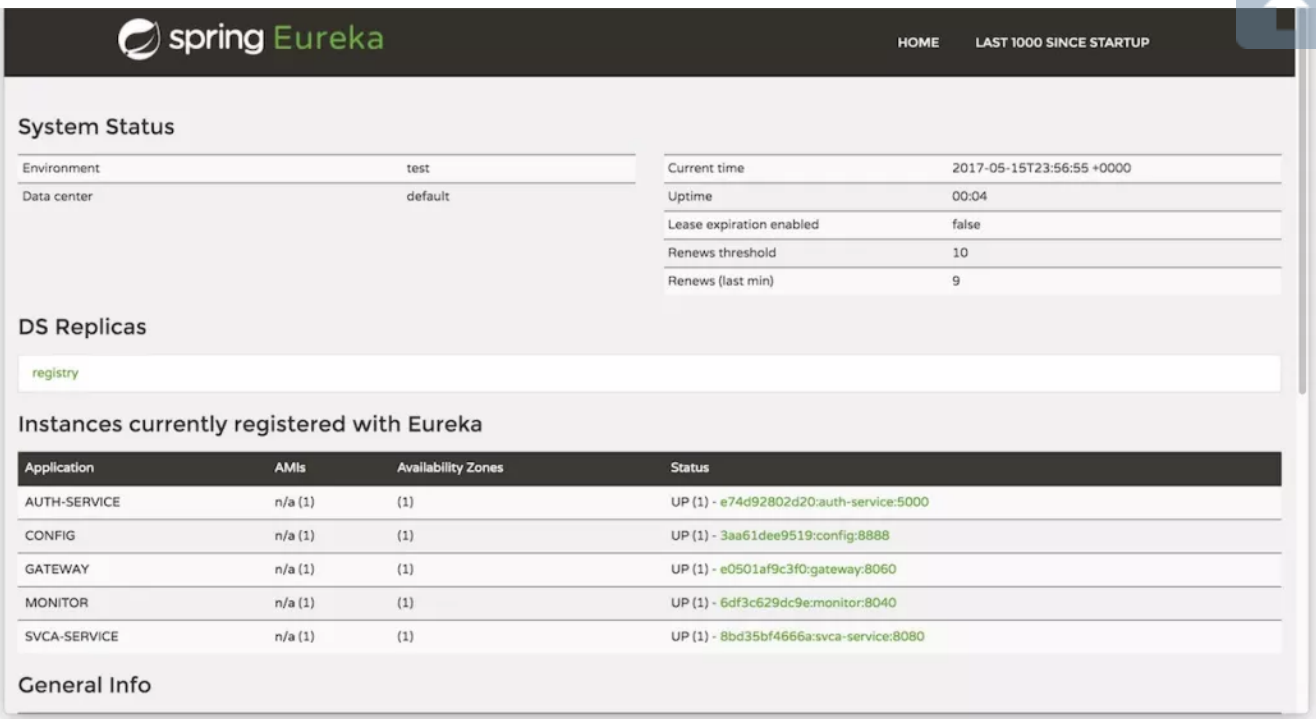
       127.0.0.1 registry config monitor rabbitmq auth-service

       或者修改各服务配置文件中的相应主机名为本地 ip
    3. 启动 registry、config、monitor、zipkin
    4. 启动 gateway、auth-service、svca-service、svcb-service

项目预览

注册中心

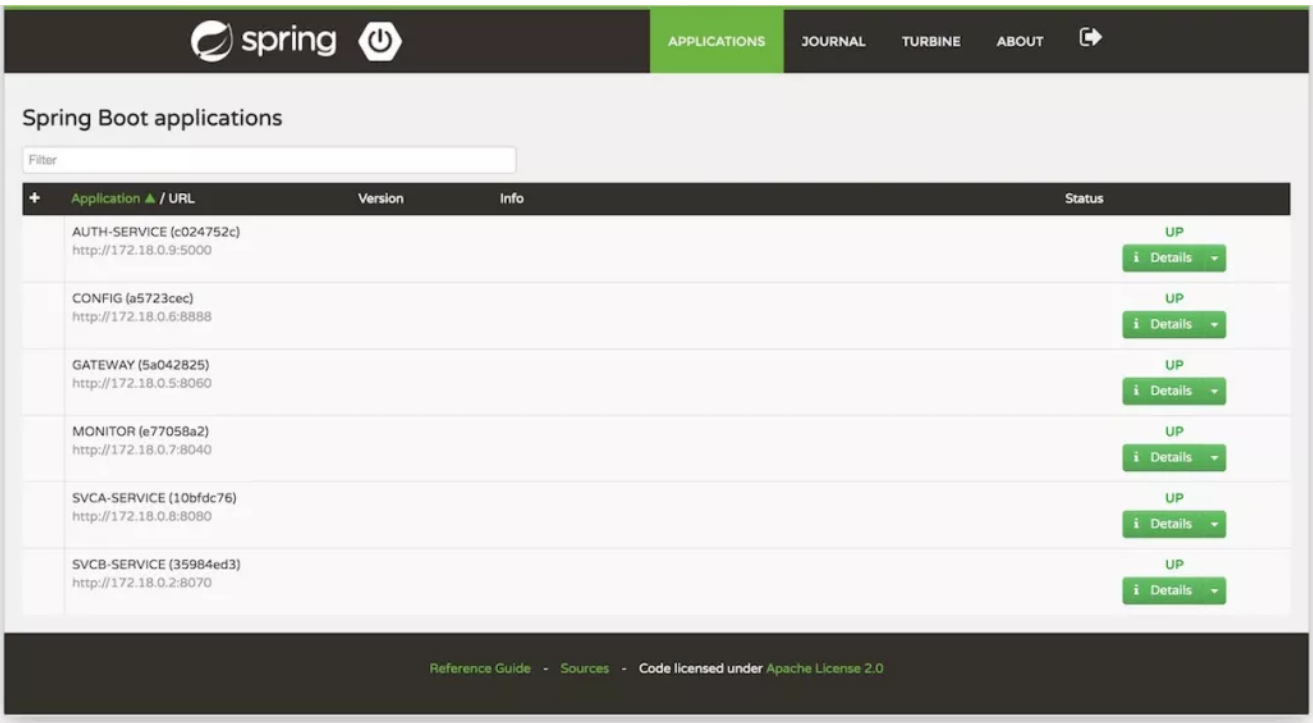
访问 http://localhost:8761/ 默认账号 user，密码 password



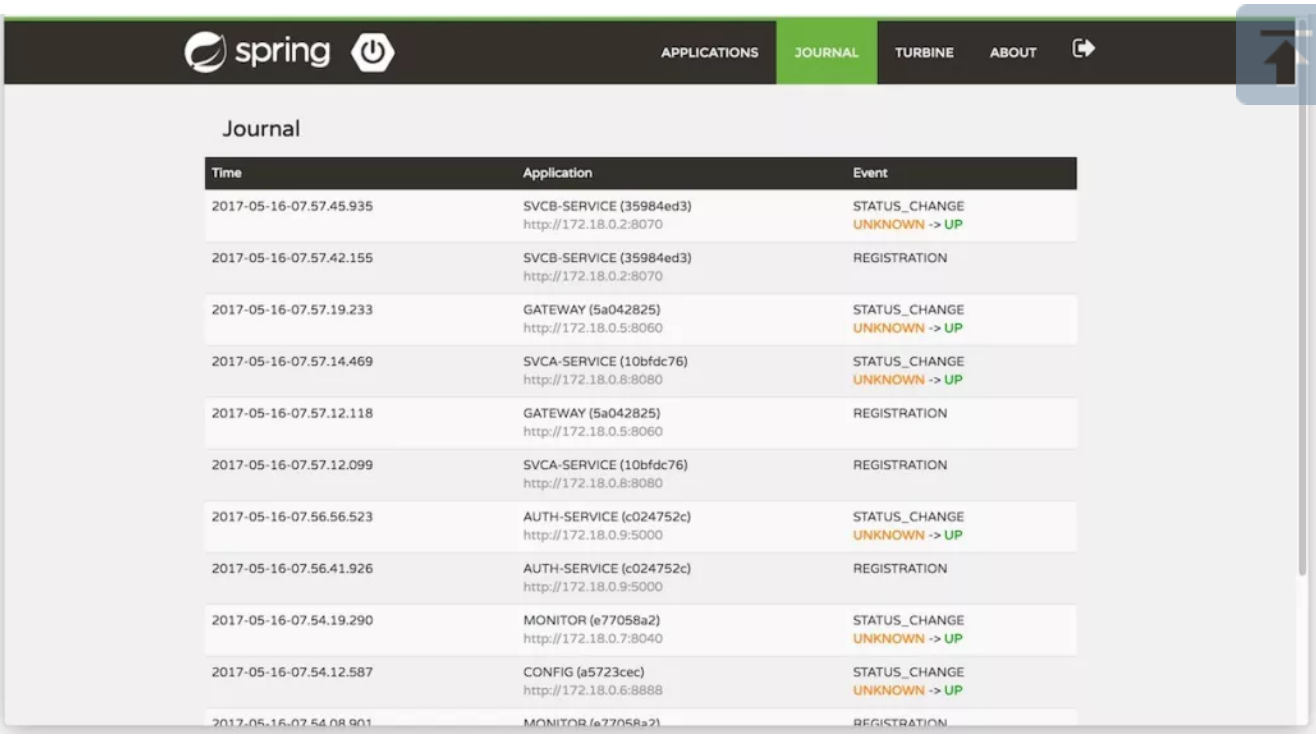
监控

访问 http://localhost:8040/ 默认账号 admin，密码 admin

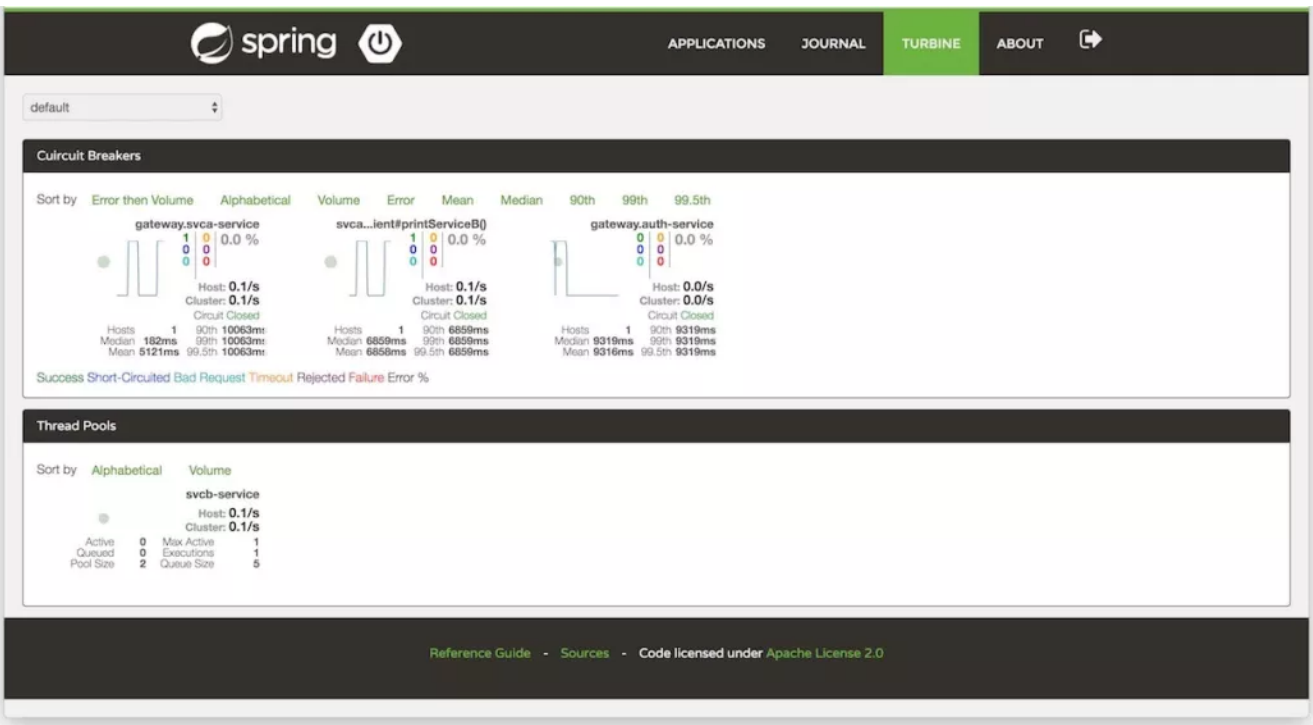
控制面板



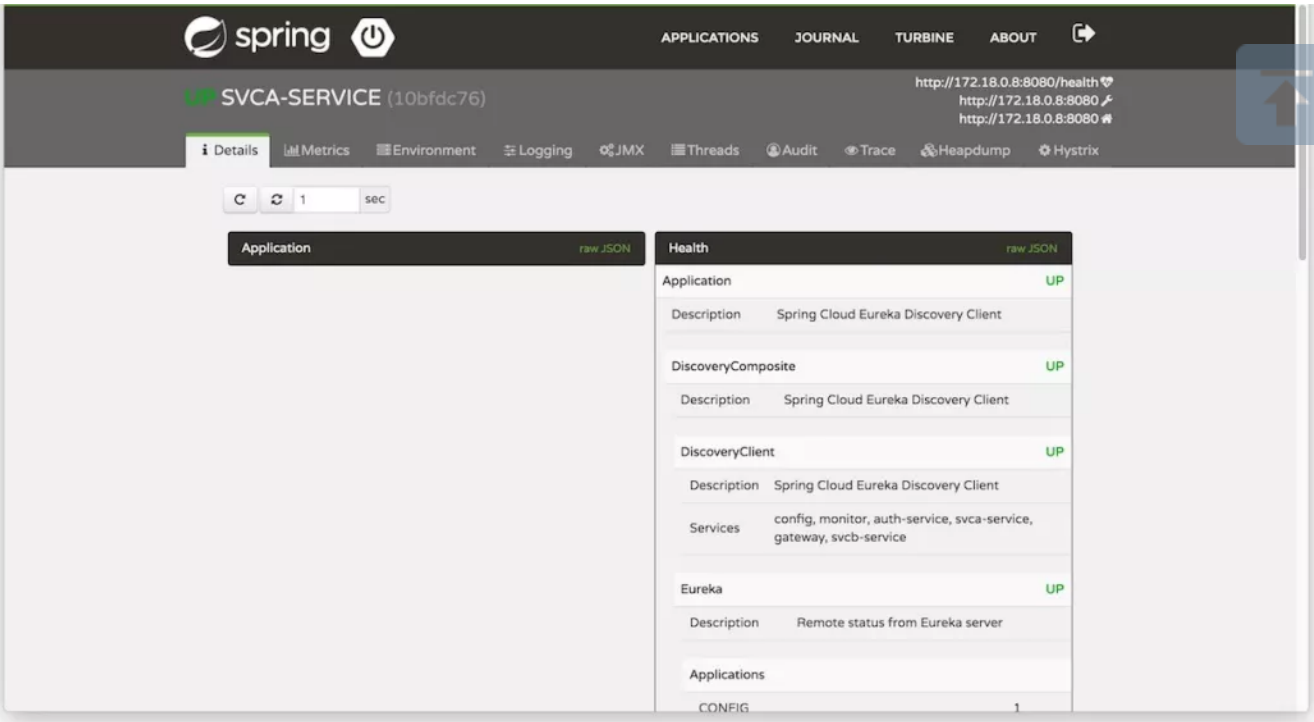
应用注册历史



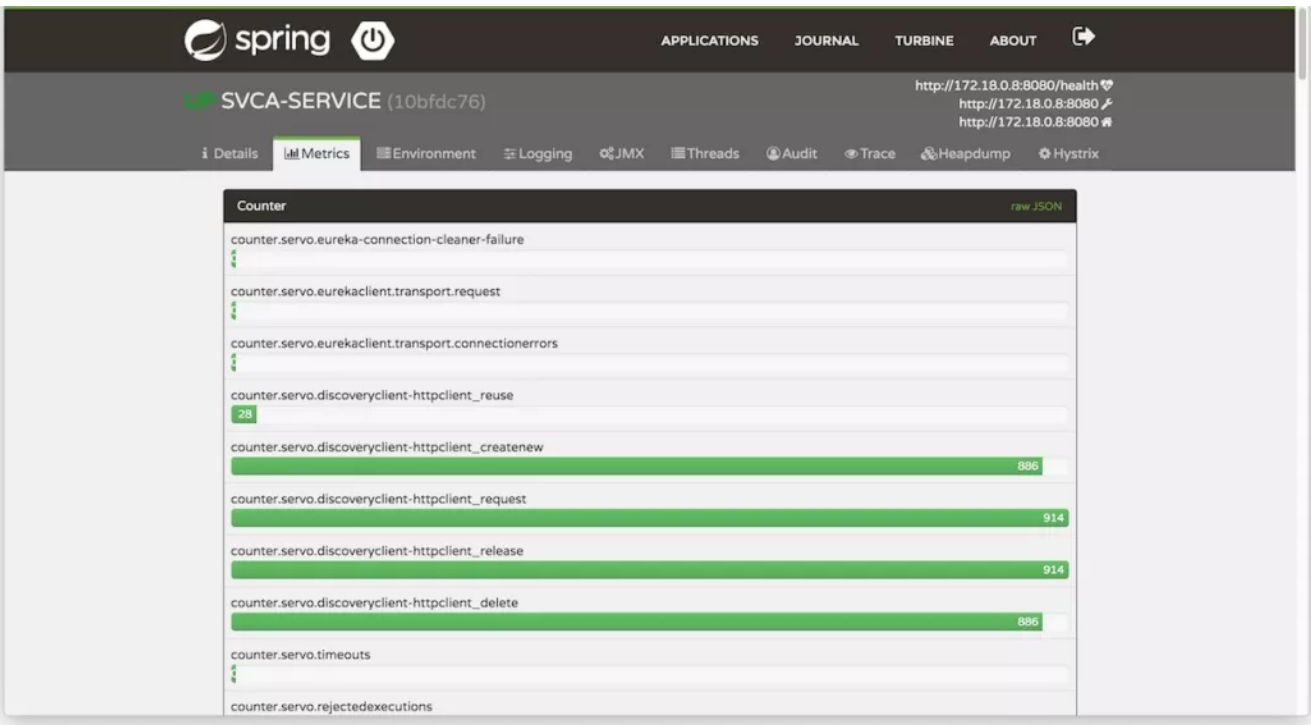
Turbine Hystrix面板



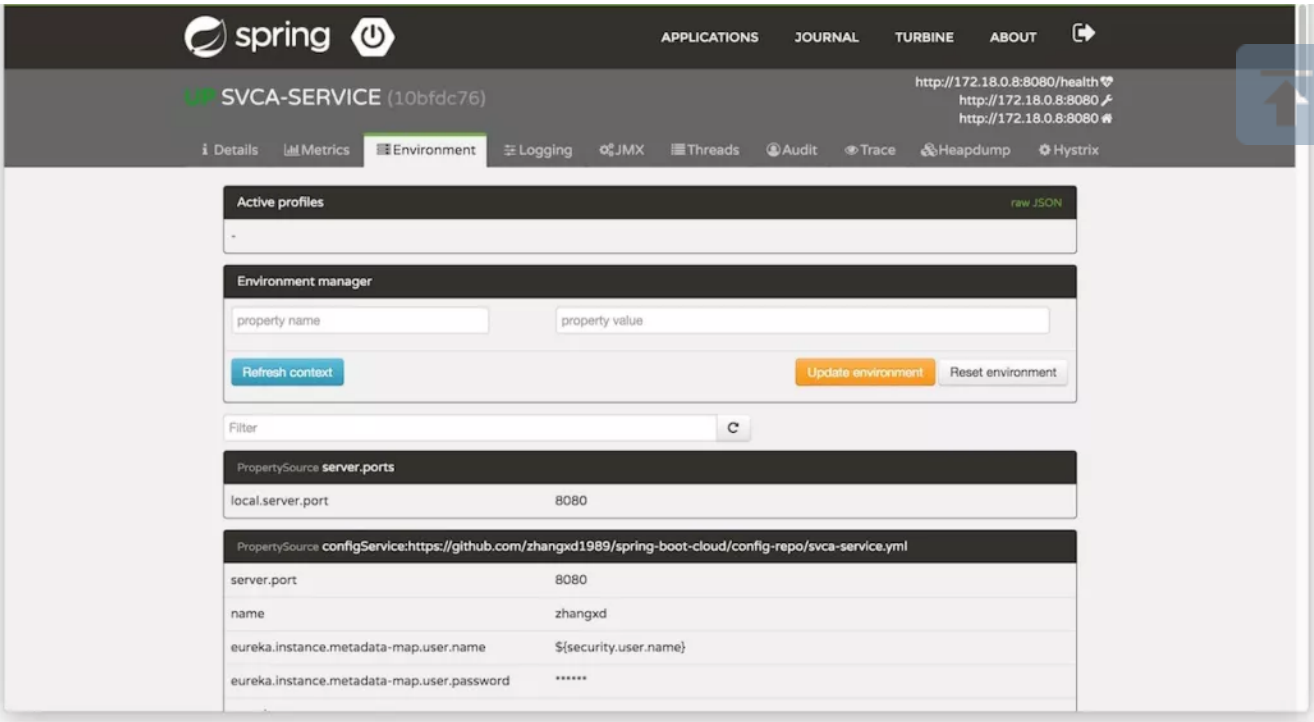
应用信息、健康状况、垃圾回收等详情



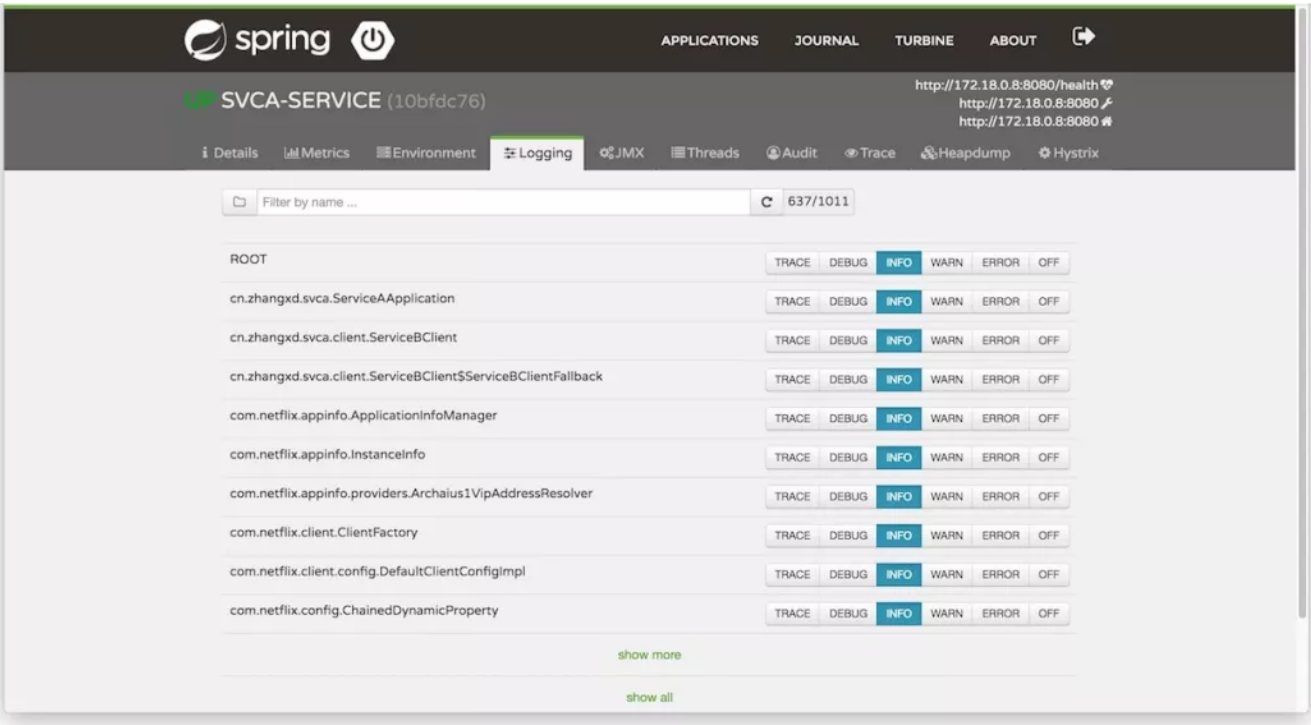
计数器



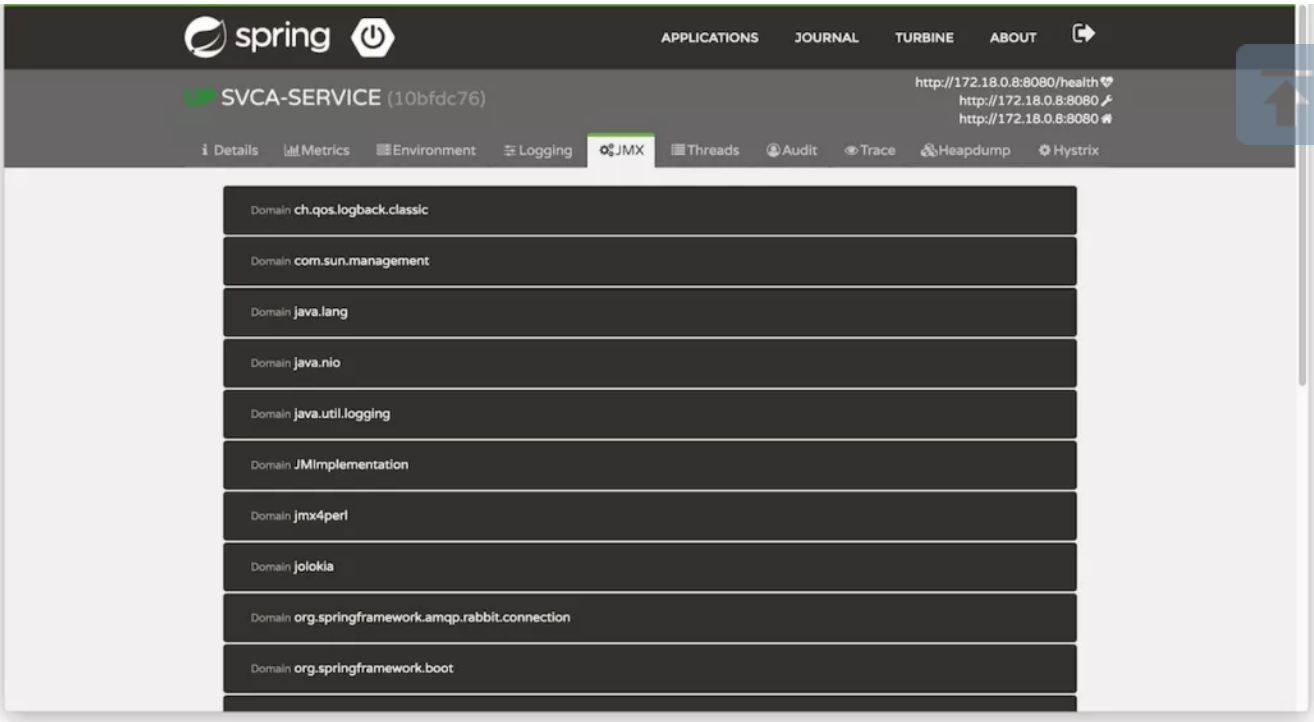
查看和修改环境变量



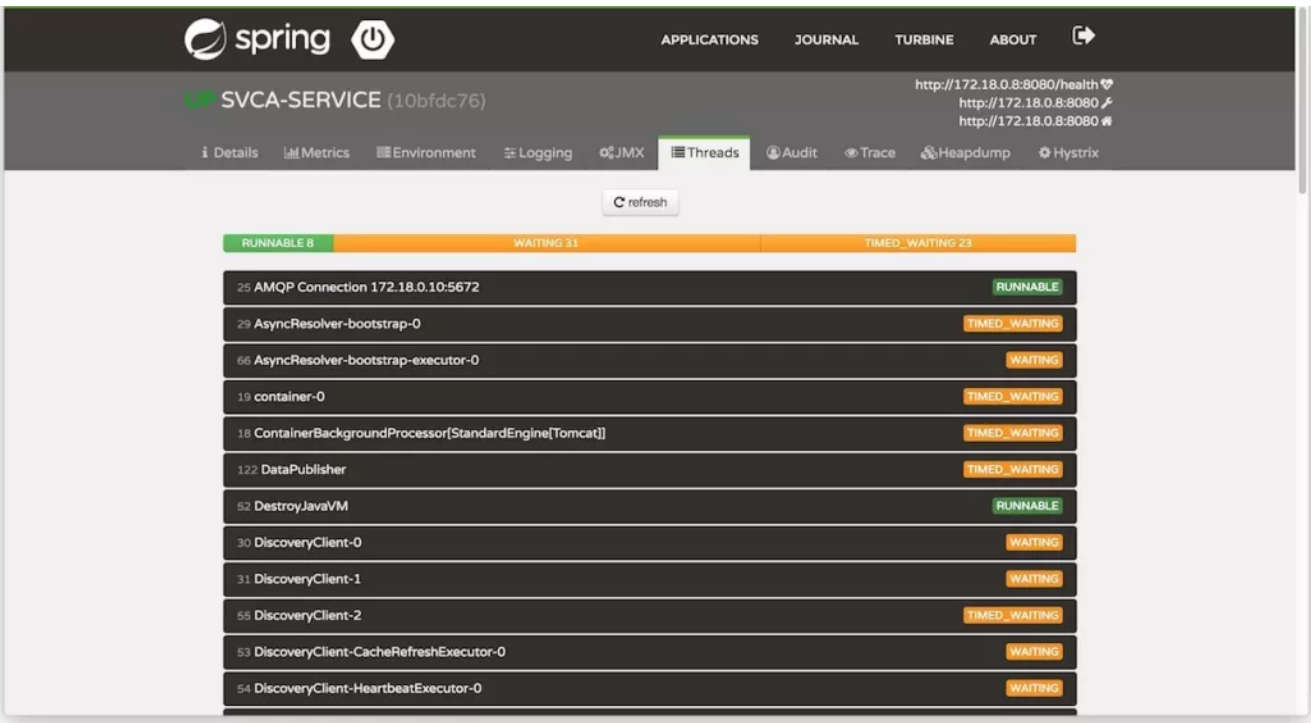
管理 Logback 日志级别



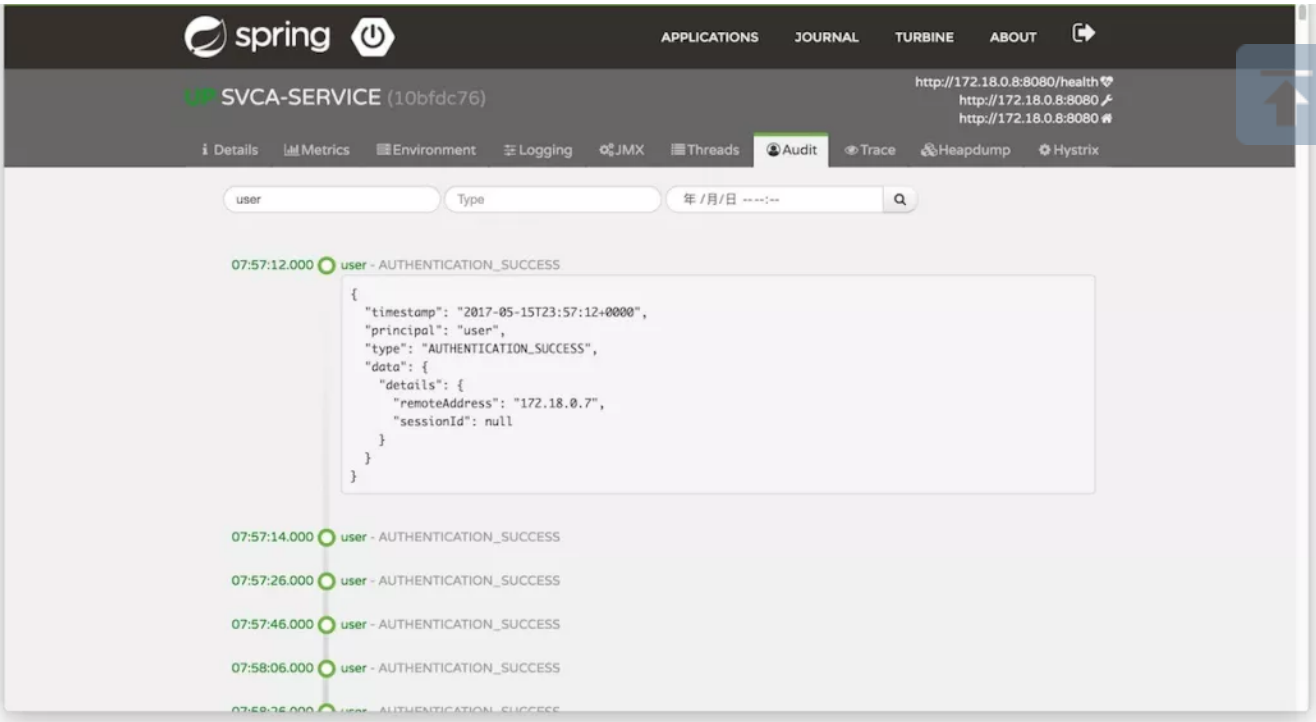
查看并使用 JMX


spring
APPLICATIONS
JOURNAL
TURBINE
ABOUT
UP SVCA-SERVICE (10bfdc76)
http://172.18.0.8:8080/health
http://172.18.0.8:8080
http://172.18.0.8:8080
Details
Metrics
Environment
Logging
JMX
Threads
Audit
Trace
Heapdump
Hystrix
Domain ch.qos.logback.classic
Domain com.sun.management
Domain java.lang
Domain java.nio
Domain java.util.logging
Domain JMImplementation
Domain jmx4perl
Domain jolokia
Domain org.springframework.amqp.rabbit.connection
Domain org.springframework.boot

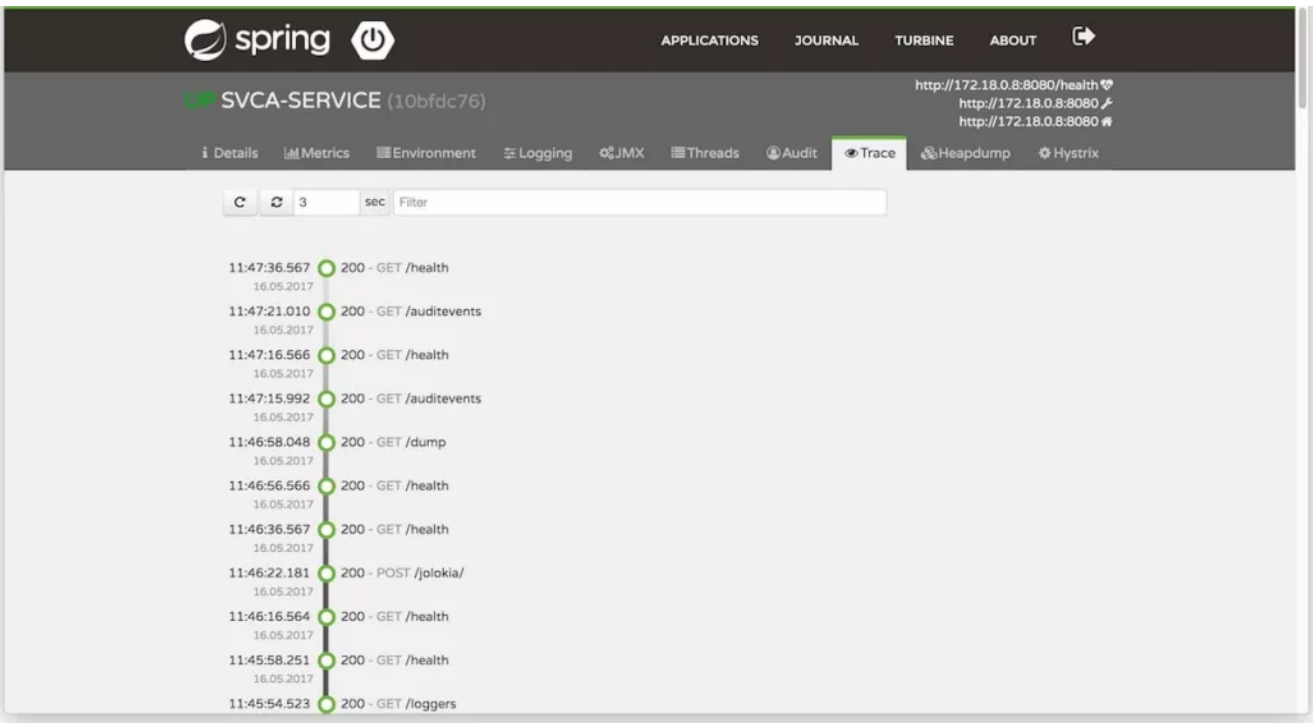
查看线程


spring
APPLICATIONS
JOURNAL
TURBINE
ABOUT
UP SVCA-SERVICE (10bfdc76)
http://172.18.0.8:8080/health
http://172.18.0.8:8080
http://172.18.0.8:8080
Details
Metrics
Environment
Logging
JMX
Threads
Audit
Trace
Heapdump
Hystrix
refresh
RUNNABLE 8
WAITING 31
TIMED_WAITING 23
25 AMQP Connection 172.18.0.10:5672 RUNNABLE
29 AsyncResolver-bootstrap-0 TIMED_WAITING
66 AsyncResolver-bootstrap-executor-0 WAITING
19 container-0 TIMED_WAITING
18 ContainerBackgroundProcessor[StandardEngine[Tomcat]] TIMED_WAITING
122 DataPublisher TIMED_WAITING
52 DestroyJavaVM RUNNABLE
30 DiscoveryClient-0 WAITING
31 DiscoveryClient-1 WAITING
55 DiscoveryClient-2 TIMED_WAITING
53 DiscoveryClient-CacheRefreshExecutor-0 WAITING
54 DiscoveryClient-HeartbeatExecutor-0 WAITING

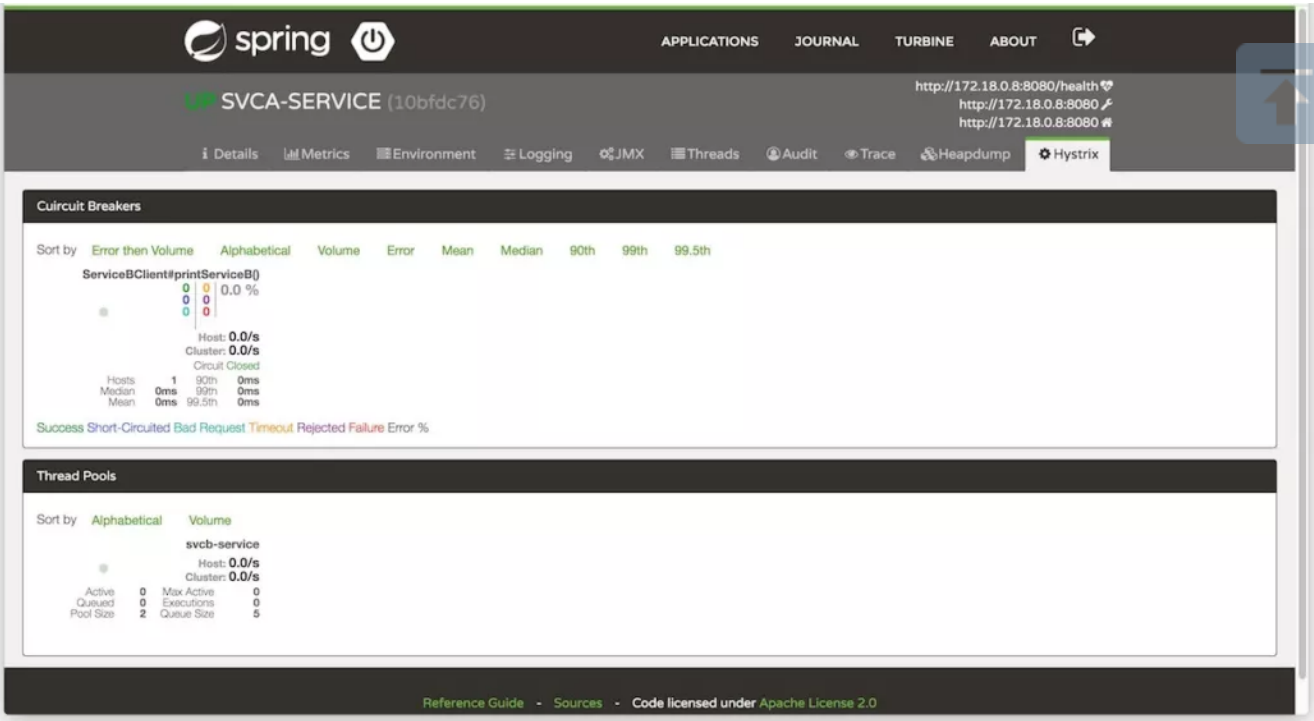
认证历史



查看 Http 请求轨迹



Hystrix 面板

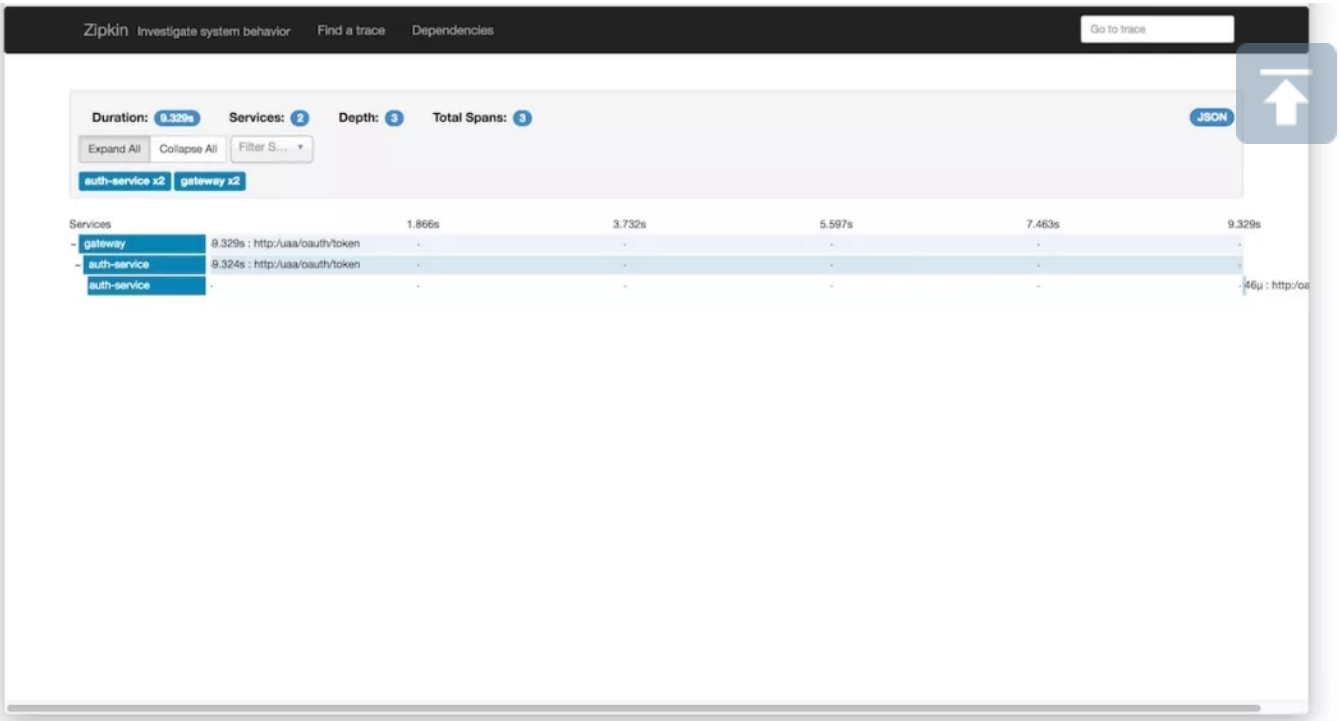


链路跟踪

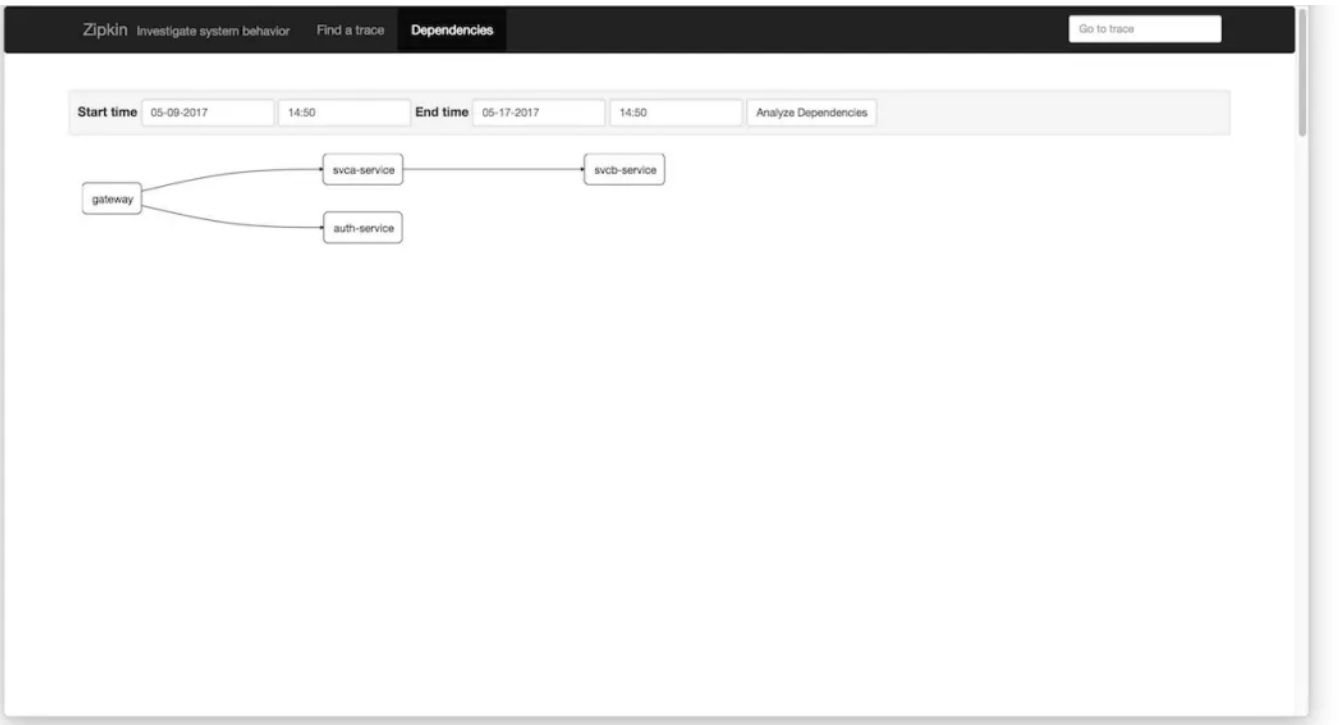
访问 http://localhost:9411/ 默认账号 admin，密码 admin

控制面板

链路跟踪明细

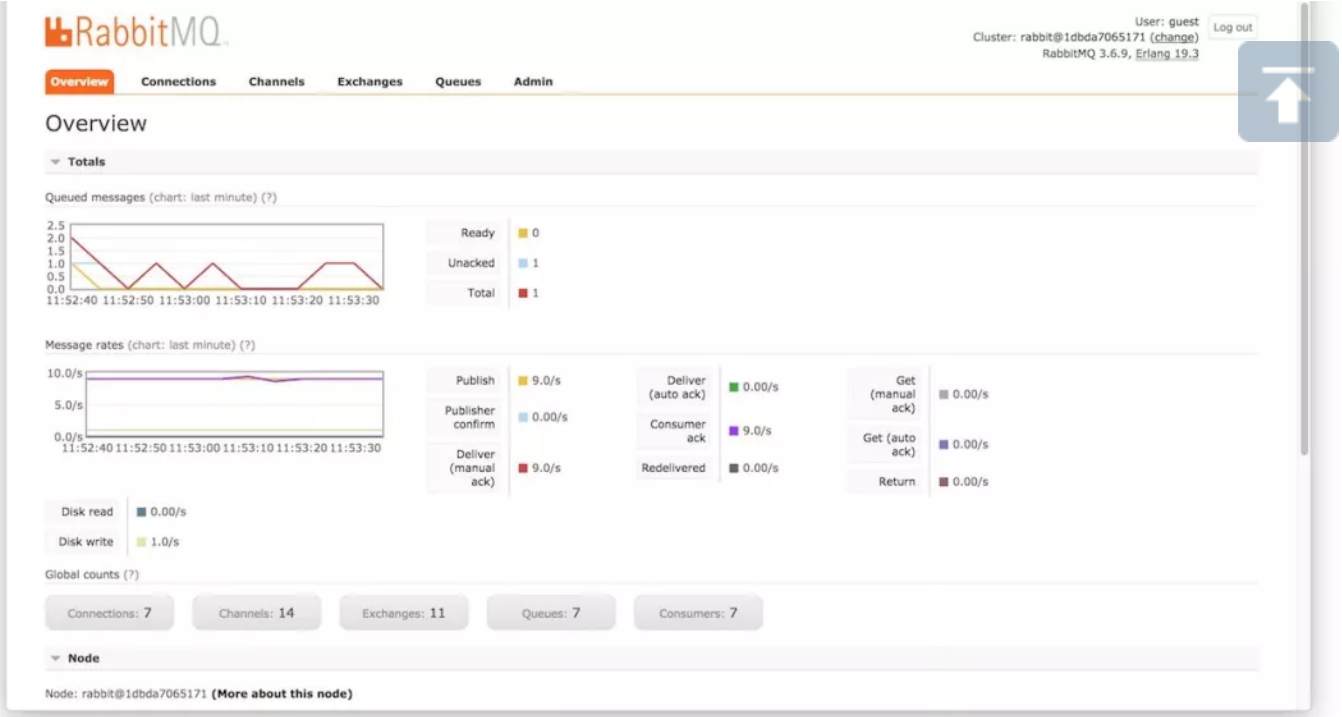


服务依赖关系



RabbitMQ 监控

Docker 启动访问 http://localhost:15673/ 默认账号 guest，密码 guest（本地 rabbit 管理系统默认端口 15672）



接口测试

1. **获取 Token**

> curl -X POST -vu client:secret http://localhost:8060/uaa/oauth/token -H "Accept: application/json" -d "password=password&username=anil&grant_type=password&scope=read%20write"

返回如下格式数据：

```
{
  "access_token": "eac56504-c4f0-4706-b72e-3dc3acdf45e9",
  "token_type": "bearer",
  "refresh_token": "da1007dc-683c-4309-965d-370b15aa4aeb",
  "expires_in": 3599,
  "scope": "read write"
}
```

2. **使用 access token 访问 service a 接口**

```
curl -i -H "Authorization: Bearer eac56504-c4f0-4706-b72e-3dc3acdf45e9" http://localhost:8060/svca
```

返回如下数据：

```
svca-service (172.18.0.8:8080)===>name:zhangxd
svcb-service (172.18.0.2:8070)===>Say Hello
```

3. **使用 access token 访问 service b 接口**

```
curl -i -H "Authorization: Bearer eac56504-c4f0-4706-b72e-3dc3acdf45e9" http://localhost:8060/svcb
```

返回如下数据：

```
svcb-service (172.18.0.2:8070)===>Say Hello
```

4. **使用 refresh token 刷新 token**

> curl -X POST -vu client:secret http://localhost:8060/uaa/oauth/token -H "Accept: application/json" -d "grant_type=refresh_token&refresh_token=da1007dc-683c-4309-965d-370b15aa4aeb"

返回更新后的 Token：

```
{
  "access_token": "63ff57ce-f140-482e-ba7e-b6f29df35c88",
  "token_type": "bearer",
  "refresh_token": "da1007dc-683c-4309-965d-370b15aa4aeb",
  "expires_in": 3599,
  "scope": "read write"
}
```

5. **刷新配置**

```
curl -X POST -vu user:password http://localhost:8888/bus/refresh
```

源码下载

https://github.com/souyunku/spring-boot-cloud.git

Contact

- 作者：鹏磊
- 出处：http://www.ymq.io
- Email：admin@souyunku.com