

云原生社区Meetup第一期



主办方



承办方

UCLoud 优刻得



云原生社区上海站
Cloud Native Community Shanghai

赞助商



万科海外

Broadview

2020-11-28



上海站



程亮



VIPKID 资深架构师

曾任百度高级研发工程师，阿里巴巴技术专家。目前负责大班课后端总架构，VIPKID 监控系统。

云原生社区 Meetup
第一期 · 上海站 ×



云原生监控体系建设

演讲人：程亮 VIPKID

目 录

- VIPKID的传统监控系统
- 升级K8S发布的监控体系
- 日志流量链路监控
- 业务监控建设

机器监控 – zabbix/falcon

VipKid Falcon

仪表盘 Screen 服务 模板 Tree 表达式 Nodata 报警看板 欢迎 chengliang

搜索Endpoints

Endpoint

空格分割多个关键字

标签(eg: job=appstore-web)

全局搜索 Limit 50 page 1 删除endpoints

快速过滤

☐ 支持shift范围选择 刷新counter列表

☐ 10.10.2.1

☐ 10.21.27.1

☐ 10.20.10.17

☐ 10.20.10.19

搜索Counters

Counter

空格分割多个关键字

搜索 Limit 50 page 1 快速过滤 删除counters 看图

☐ Counters 类型 频率

机器监控 – falcon插件开发



GitLab interface showing the 'plugin' project details. The left sidebar contains navigation links: Project, Details, Activity, Cycle Analytics, Repository, Issues (0), Merge Requests (0), CI / CD, Operations, Wiki, Snippets, and Settings.

The main content area displays the project overview for 'plugin' (master branch). It includes a progress bar and buttons for adding changelog, contribution guide, enabling Auto DevOps, adding a Kubernetes cluster, and setting up CI/CD.

A commit history table is shown below the progress bar:

Name	Last commit	Last update
service	using post url	12 hours ago
sys	add service & sys dir	1 year ago
README.md	Initial commit	1 year ago

Below the table, the 'plugin' directory structure is visible, including a 'README.md' file and a 'plugin' subdirectory.

云原生监控体系建设 - 背景描述

1. VIPKID 内部服务已经基本实现k8s发布，基于k8s的监控完善
2. 基于内部系统需要，开发了各种k8s-operator，便于监控的自动化上线
3. 报警规则可以基于业务需要可以自定义
4. 针对老服务，监控自动适配，向前兼容
5. 监控手段多样，指标丰富。支持主动探测，上报接收，底层机器数据获取，业务数据获取等

云原生发布 – K8S发布系统

基于k8s的发布系统 - Ratel

☰

Ratel / 应用列表

输入应用的名字

选择业务端

搜索

所有应用

应用名	实例数			应用类型	级别
	预发环境	生产环境	QA测试		
gateway	1	3		meteor2	p0
user-service	1	3	1	meteor2	p0
mail-service	1			jar	p0

云原生监控 – thanos (Prometheus)

基于K8S的监控，采用thanos, 并开发各种exporter

vkafka-exporter

Vkredis-exporter

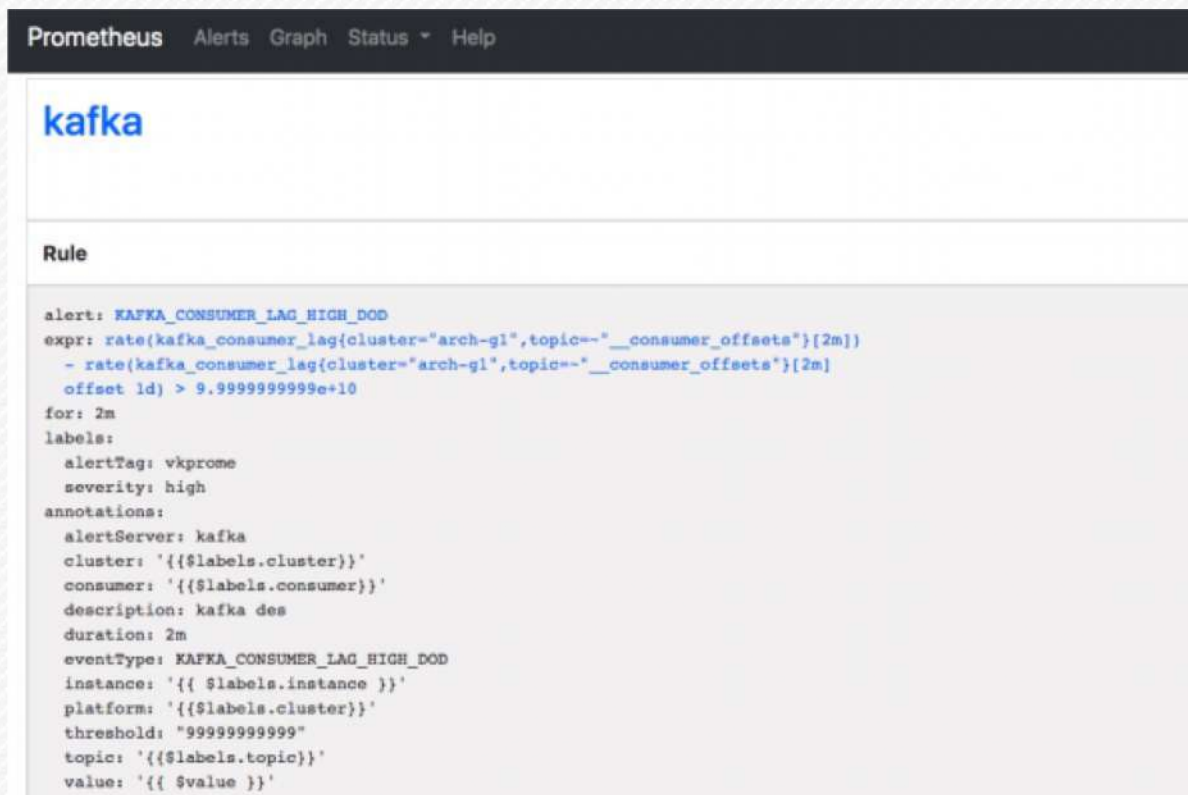
Doris-exporter

Custom-exporter

Vkos-exporter

node-exporter

...



The screenshot shows the Prometheus Alerts interface. The top navigation bar includes 'Prometheus', 'Alerts', 'Graph', 'Status', and 'Help'. The main content area is titled 'kafka' and displays a rule configuration under the 'Rule' section. The rule is named 'KAFKA_CONSUMER_LAG_HIGH_DOD' and is an alert. The expression (expr) is a PromQL query that calculates the rate of Kafka consumer lag for a specific cluster and topic, comparing it to a threshold. The rule includes labels for alertTag, severity, and annotations for alertServer, cluster, consumer, description, duration, eventType, instance, platform, threshold, topic, and value.

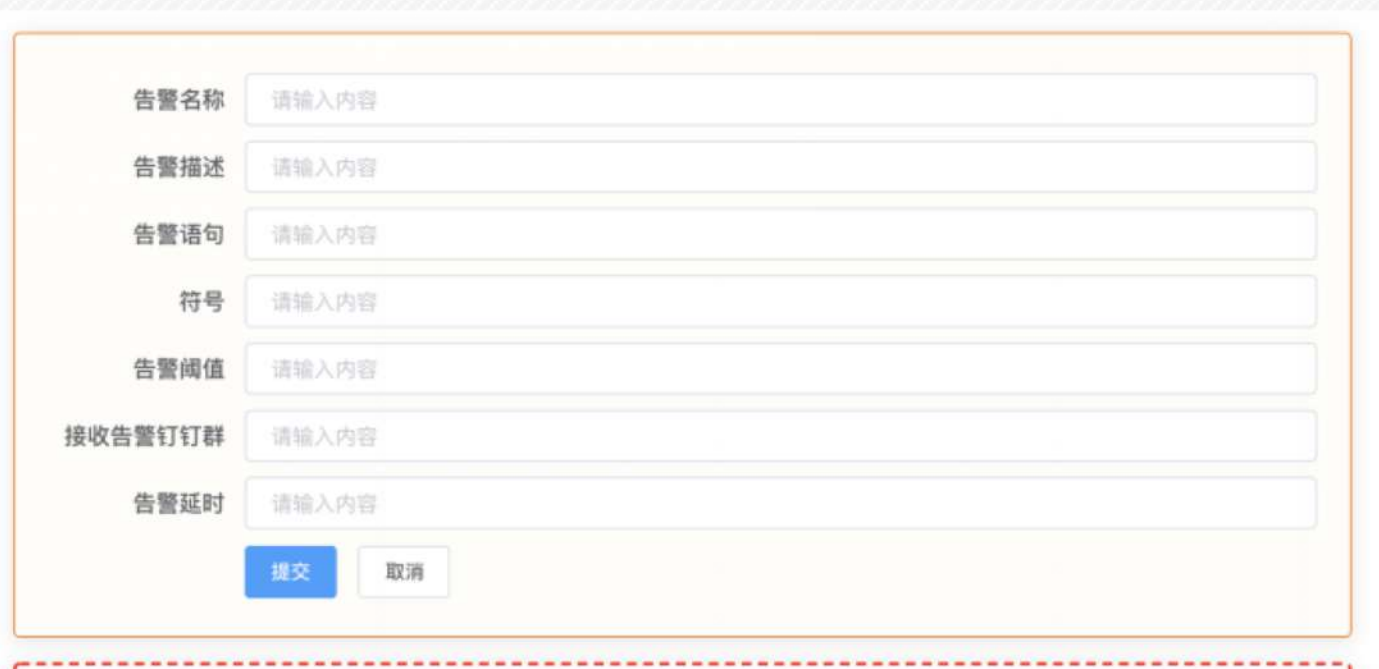
```

alert: KAFKA_CONSUMER_LAG_HIGH_DOD
expr: rate(kafka_consumer_lag{cluster="arch-g1",topic="__consumer_offsets"}[2m])
      - rate(kafka_consumer_lag{cluster="arch-g1",topic="__consumer_offsets"}[2m]
        offset 1d) > 9.9999999999e+10
for: 2m
labels:
  alertTag: vkprome
  severity: high
annotations:
  alertServer: kafka
  cluster: '{{ $labels.cluster }}'
  consumer: '{{ $labels.consumer }}'
  description: kafka des
  duration: 2m
  eventType: KAFKA_CONSUMER_LAG_HIGH_DOD
  instance: '{{ $labels.instance }}'
  platform: '{{ $labels.cluster }}'
  threshold: "9999999999"
  topic: '{{ $labels.topic }}'
  value: '{{ $value }}'
  
```

云原生监控 – PrometheusRule (CRD) 管理

PrometheusRule (CRD) 的管理

1. UI 管理规则
2. 自动对接内部的报警系统
3. 支持容器外服务发现



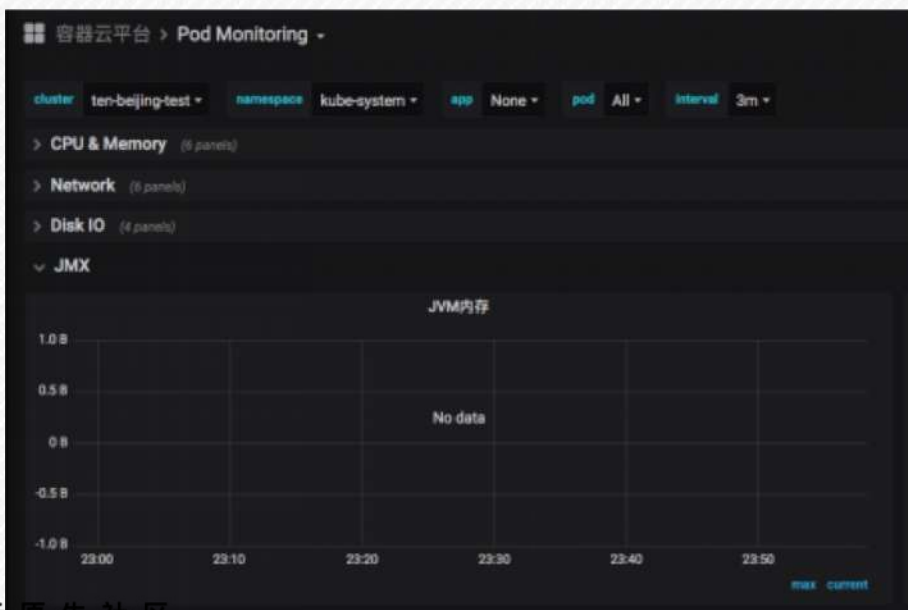
The image shows a web form for managing PrometheusRule (CRD) configurations. The form is enclosed in a light orange border and contains several input fields, each with a label and a placeholder text '请输入内容' (Please enter content). The fields are: 告警名称 (Alert Name), 告警描述 (Alert Description), 告警语句 (Alert Expression), 符号 (Symbol), 告警阈值 (Alert Threshold), 接收告警钉钉群 (Alert DingTalk Group), and 告警延时 (Alert Delay). At the bottom of the form, there are two buttons: a blue '提交' (Submit) button and a white '取消' (Cancel) button.

云原生监控 – 监控数据展示



Grafana Prometheus 数据展示

1. 基础监控数据展示

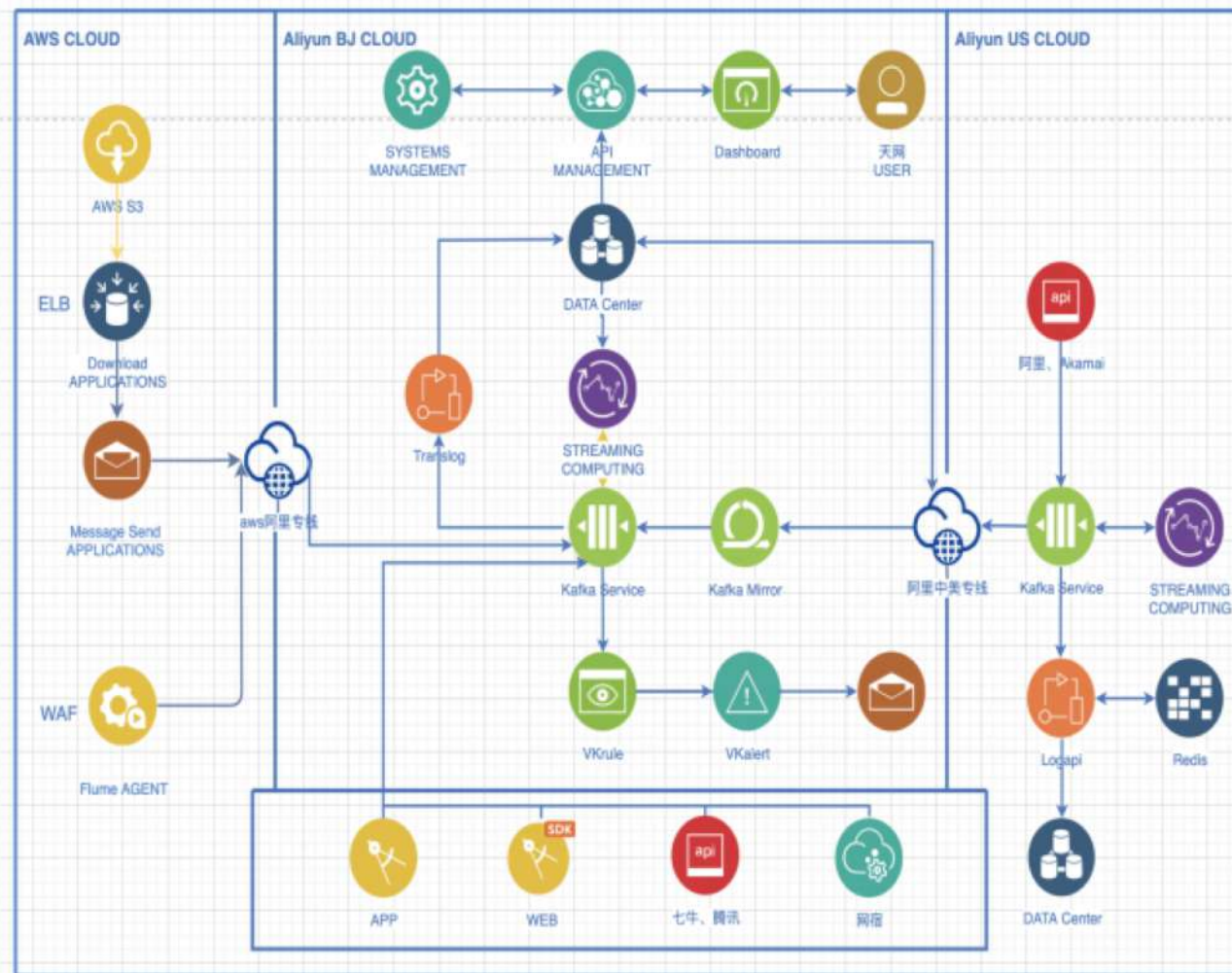


2. exporter数据展示

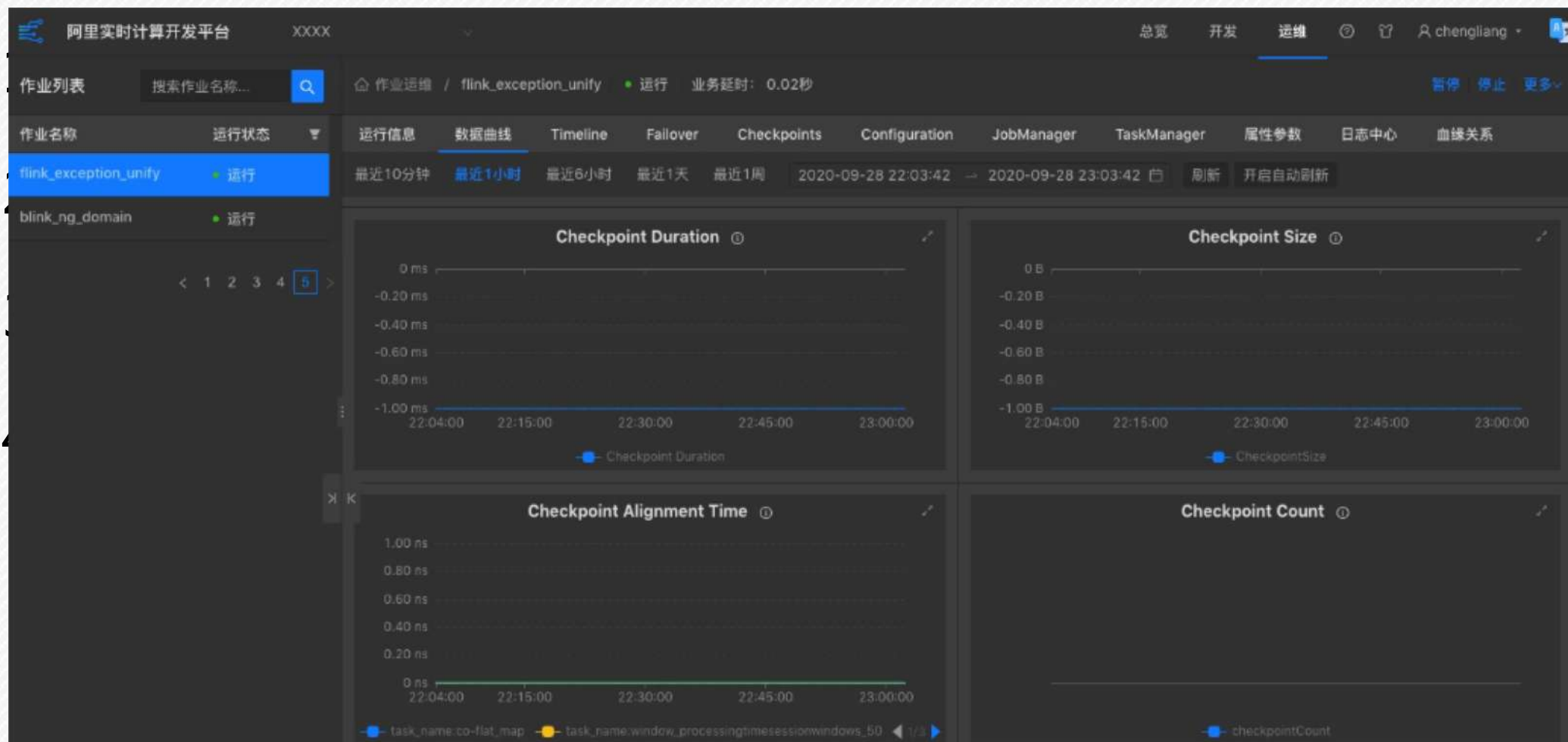


流量链路监控-架构

1. 基于日志统计
2. CDN LB日志收集
3. 跨国多集群搭建
4. 实时流flink统计
5. 自定义报警阈值



流量链路监控-实时流量统计



流量链路监控-展示



流量链路监控 – 规则设置

1. 域名/path监控
2. 阈值/同环比报警
3. 多种metrics 2xx 4xx
4. 多种报警途径

The screenshot shows a web-based configuration form for monitoring a specific API endpoint. At the top, the target URL is `www.vipkid.com/api/test/controller1/test1/test2`. The '是否监控' (Whether to monitor) option is set to '是' (Yes). The '监控开始时间' (Monitoring start time) is set to 00:00 and the '监控结束时间' (Monitoring end time) is set to 23:59. The '请求方式' (Request method) is set to 'ALL'. The '告警类型' (Alert type) is set to '阈值设置' (Threshold setting). Under '设置1:' (Setting 1), the '监控指标' (Monitoring metric) is 'count403', the '监控方式' (Monitoring method) is '数值' (Value), the '监控条件' (Monitoring condition) is '10' minutes, and the '比较类型' (Comparison type) is '大于' (Greater than) with a '阈值' (Threshold) of '1000'. A green button labeled '添加子条件' (Add sub-condition) is located below these settings. At the bottom, the '发送方式' (Sending method) is configured with '钉钉群' (DingTalk group) checked, and '短信' (SMS), '邮件' (Email), and '电话' (Phone) unchecked.

www.vipkid.com/api/test/controller1/test1/test2

是否监控 ☒ 是 ☐ 否

监控开始时间 00:00 监控结束时间 23:59

请求方式 ALL

告警类型 ☒ 阈值设置 ☐ 同环比设置

设置1:

监控指标 count403

监控方式 数值

监控条件 10 分钟 10 次

比较类型 大于 阈值 1000

添加子条件

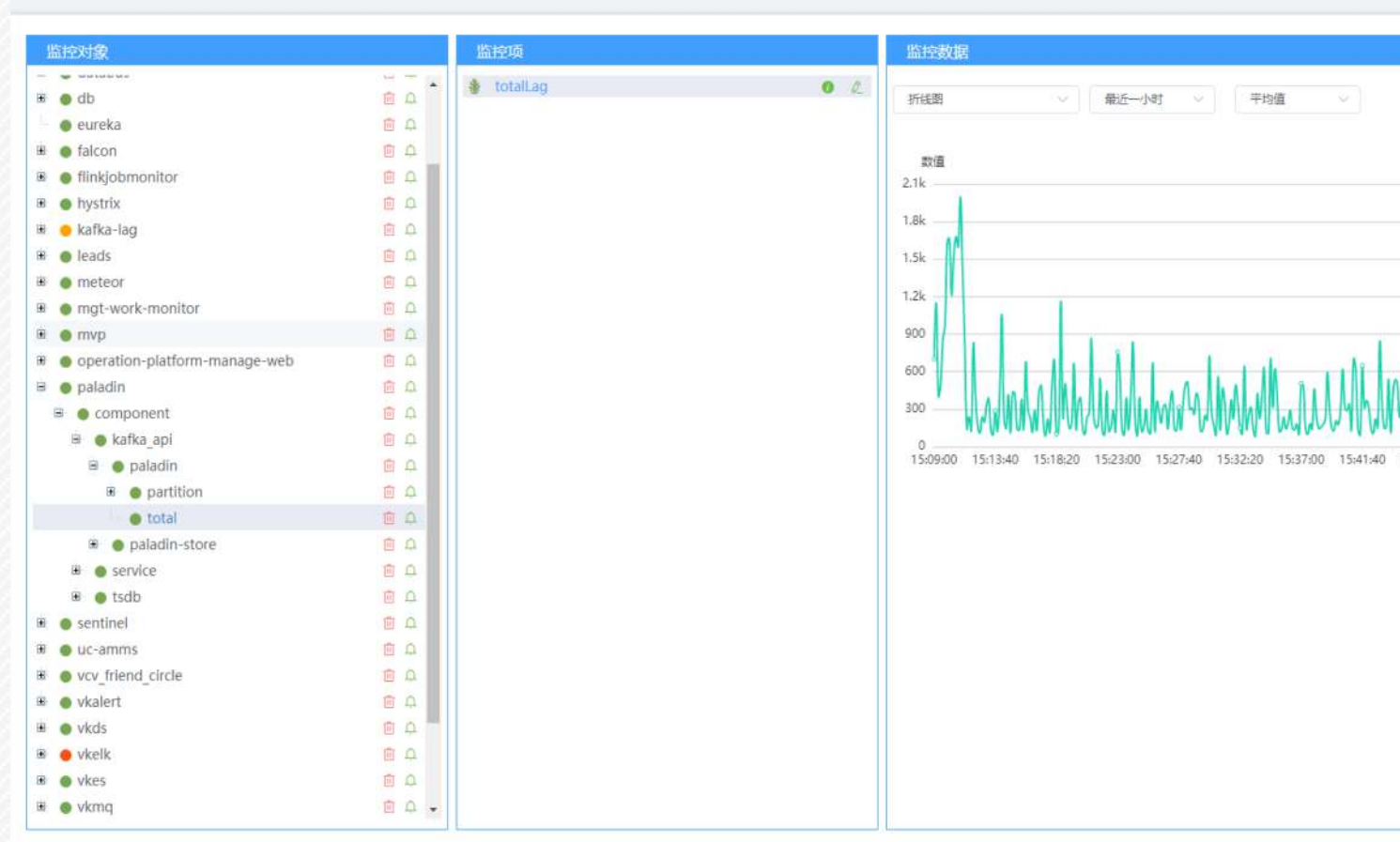
发送方式 ☒ 钉钉群 ☐ 短信 ☐ 邮件 ☐ 电话

1. 接口上报，固定的接口，数据格式相对固定，报警规则在固定字段上配置，可以阈值比较，同环比报警等
2. 脚本自定义，通过exporter执行器转换，prometheus触发
3. falcon插件，通过falcon-dashborad配置报警
4. 基于数据或者接口的定时回归检查，并判断返回值报警

业务监控 – 接口上报



1. 提供api，上报数据
2. 设置阈值，可以根据上报的字段配置
3. 提供dashborad展示



业务监控 – Prometheus-自定义exporter适配

- 1.支持自定义脚本扫描
- 2.支持metrics暴露
- 3.依赖prometheus定时

4.服务自发现CRD配置

```
services:  
  test-ps:  
    metrics:  
      - name: custome_ps_num_test  
        script: ./test.sh  
        args: agent,mq_mcd_2_ccd  
        help: helpinfo  
        type: gauge  
        script: /root/worker.sh  
        args: arg1  
        help: helpinfo  
        type: counter
```

```
imagePullPolicy: IfNotPresent  
name: lvc-relay-instance  
ports:  
  - containerPort: 50091  
    protocol: TCP  
  - containerPort: 18080  
    protocol: TCP  
  - containerPort: 8080  
    protocol: TCP  
  - containerPort: 9999  
    protocol: TCP  
resources:  
  limits:  
    cpu: 650m
```

业务监控 – 数据/接口监控



1. 数据获取/接口调用

2. 返回阈值判断

✕ 第3步

查询数据库

DB组件 ▼

Sql语句

select * from vipkid_order.pack_order where student_id = 293700190 order by id desc limit 1

Output

db_create_date_time

\${T:{data[0].create_date_time},%Y-%m-%d}

🗑️

+

云原生社区Meetup

第一期·上海站



THANKS