Flashcat

让智能运维成为每个企业的标配

Nightingale

一站式智能观测平台

北京快猫星云科技有限公司

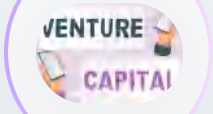
Flashcat

关于快猫星云

由开源项目夜莺
的核心开发团队组成

创始团队均来自
国内一线互联网公司

累计为上百家企业
提供服务



由国内顶级投资机构
连续投资

Nightingale

Nightingale 夜莺是一个 All-in-One 的开源监控工具，最早由滴滴开发并开源，是中国计算机学会接受捐赠并托管的第一个开源项目，在 GitHub 上有超过 11000 颗星，是广受关注和使用的开源监控工具。快猫星云技术团队是夜莺开源项目的重要开发力量。

Flashcat

Flashcat 是快猫星云以开源夜莺为内核打造的一体化观测平台，支持指标、日志、链路追踪数据的统一采集、存储、监控告警、可视化分析，只需一个Flashcat平台即可全面覆盖云上、云下、Kubernetes 的观测需求。Flashcat 预置了行业领先的故障发现定位最佳实践，并深度使用 AI 加速故障的分析过程，大幅缩短故障恢复时间。

FlashDuty

Flashduty 是一站式告警事件响应平台，支持告警聚合、降噪、排班、协作，内置飞书/钉钉/企微/短信/电话等多种通知方式，在IM和App中响应和处理告警，全生命周期管理告警事件、分析告警数据，提升 OnCall 体验，缩短 MTTA/MTTR。

Flashcat

快猫星云 ❤ Nightingale

CNCF landscape project

中国计算机学会接受捐赠并托管的第一个开源项目

11000+ GitHub Star

是最受关注的开源监控项目之一

150 Release

150多个版本发布，社区活跃

140 代码贡献者

上百位社区贡献者参与其中，群策群力

10000+ 企业用户

政企、金融、科技、互联网，众多公司信赖之选

Flashcat 解决什么问题

Flashcat

一体化的观测平台

组织好数据

✔**多对象**：
物理机 / 微服务 / 网络 / Kubernetes

✔**多平台**：
公有云、私有云、多云

✔**多维度**：
Metrics、Logs、APM、Events、RUM

故障发现定位的智能助手

用好数据

✔**发现故障**：
快速准确发现业务故障

✔**定位故障**：
一站式、引导式完成分析和定位

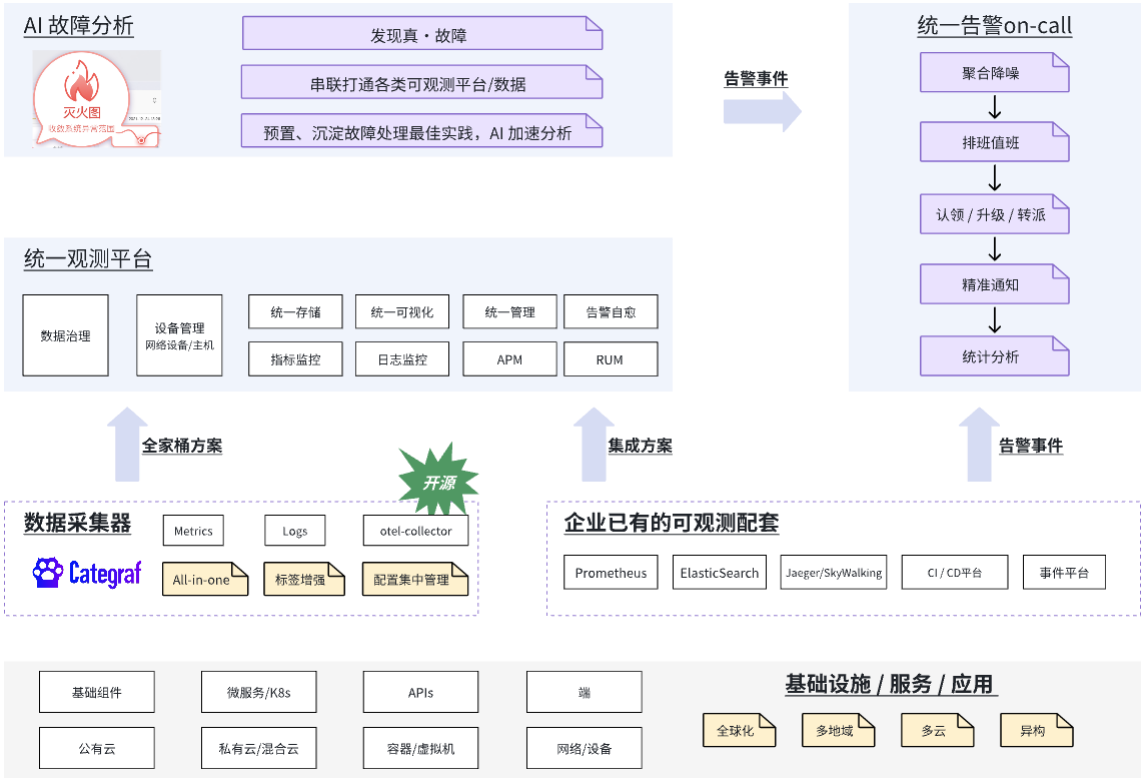
✔**最佳实践**：
持续积累和进化的稳定性保障实践

✔**智能化**：
深度使用 AI 加速问题分析和修复过程

Flashcat 旨在构建一个一体化的观测平台，并致力于解决服务稳定性保障过程中的核心难题

Flashcat

整体架构和特点



统一采集

采用插件化，内置集成上百种采集插件，GPU、服务器、网络设备、中间件、数据库、应用、业务，云上云下，均可监控，开箱即用。

统一集成

可集成企业内部已有的、云上云下的可观测配套系统，支持对接数十种数据源，已有工具无需推倒重来，充分利旧，减少落地阻力，快速见效，串联打通数据，发挥协同分析的价值。

统一观测平台

将Metrics、Logs、Traces、Events、Profiling 等多种可观测性数据拉通，一个平台集中观测，具备完整的平台能力。

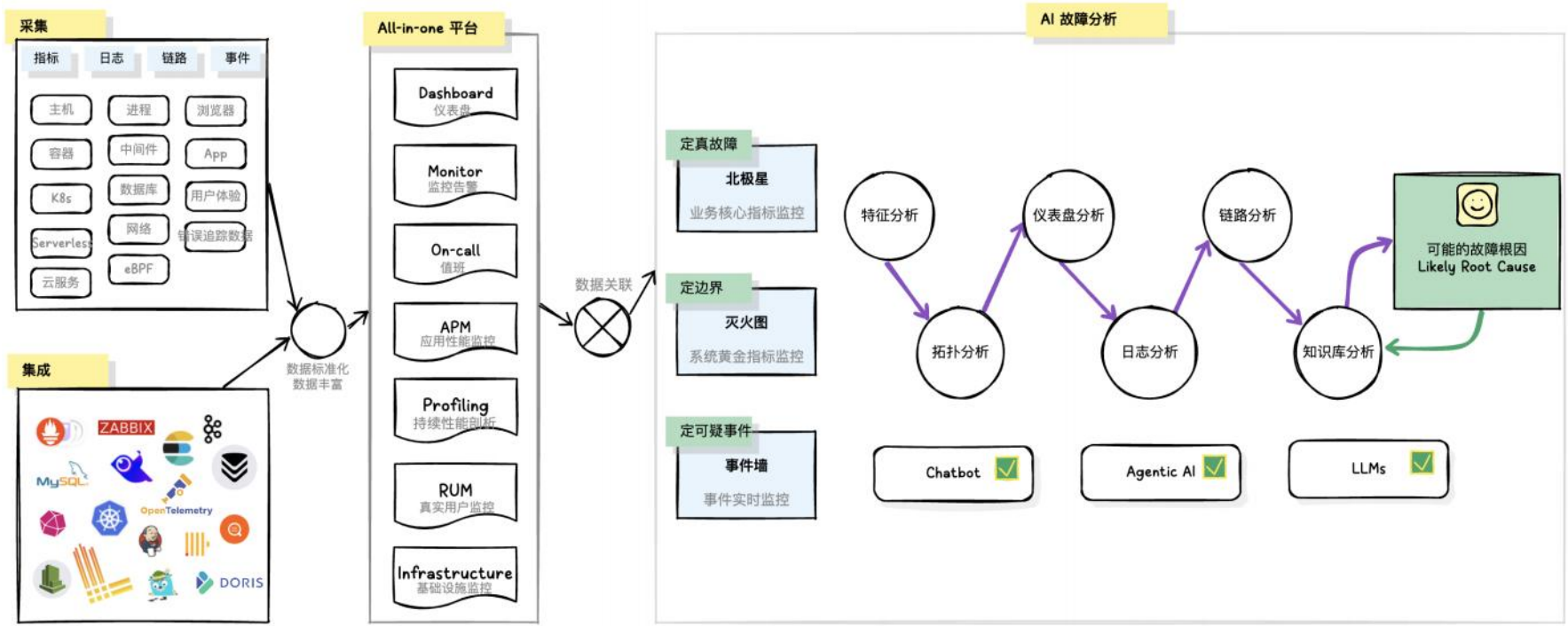
统一告警Oncall

支持指标告警、日志告警、智能告警，支持收集数十种各类监控系统的告警事件，进行统一的告警收敛、降噪、排班、认领、升级、协同，大幅提升告警处理效率，是各技术团队日常协同处理告警的统一入口。

AI 故障分析

预置稳定性保障行业最佳实践，既提供全局业务视角，又具备层层下钻、引导式、智能化的故障定位能力，有效缩短故障发现和定位时间，是各技术团队稳定性保障的统一入口。

Flashcat

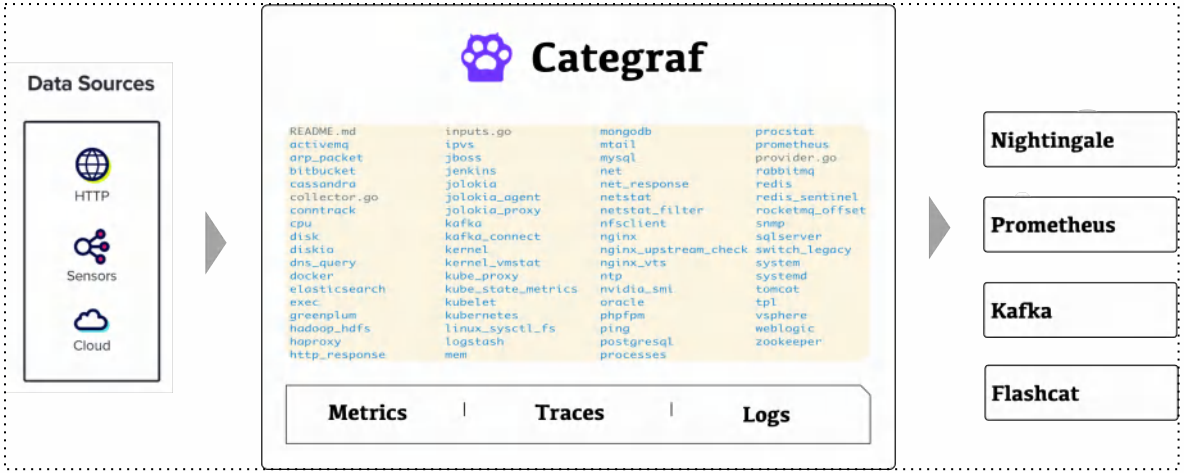

采集
指标
日志
链路
事件
主机
进程
浏览器
容器
中间件
App
K8s
数据库
用户体验
网络
错误追踪数据
Serverless
eBPF
云服务
集成
ZABBIX
MySQL
OpenTelemetry
DORIS
数据标准化
数据丰富
All-in-one 平台
Dashboard
仪表盘
Monitor
监控告警
On-call
值班
APM
应用性能监控
Profiling
持续性能剖析
RUM
真实用户监控
Infrastructure
基础设施监控
数据关联
AI 故障分析
定真故障
北极星
业务核心指标监控
定边界
灭火图
系统黄金指标监控
定可疑事件
事件墙
事件实时监控
特征分析
仪表盘分析
链路分析
拓扑分析
日志分析
知识库分析
可能的故障根因
Likely Root Cause
Chatbot
Agentic AI
LLMs

Flashcat

Categraf | All-in-one的采集器

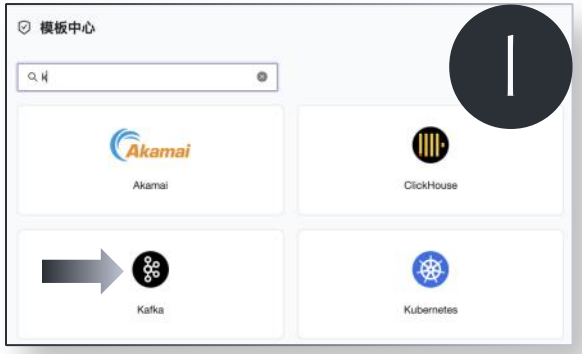
开源 / All-in-one / 开箱即用

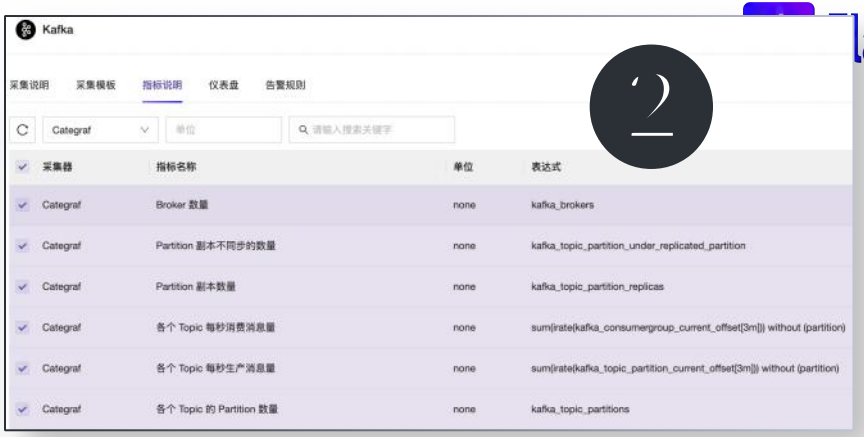
Flashcat

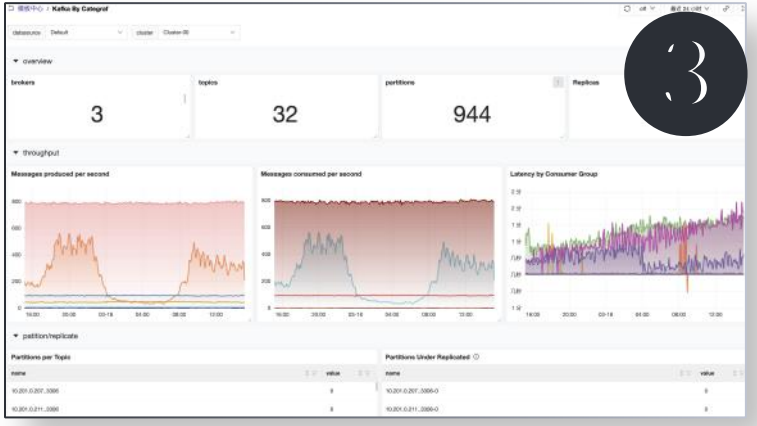


1、中心端集中管理所有采集点配置
2、支持agent自升级
3、支持标签增强
4、支持agent模式、Proxy模式运行
5、支持K8s部署
6、支持网络拨测
7、支持Pingmesh
8、支持持续性能剖析

100多种插件，开箱即用
超过130万次下载，广受信赖


模板中心
Akamai
ClickHouse
Kafka
Kubernetes
1


Kafka
采集说明
采集模板
指标说明
仪表盘
告警规则
Categraf
单位
请输入搜索关键字
采集器
指标名称
单位
表达式
Categraf Broker 数量 none kafka_brokers
Categraf Partition 副本不同步的数量 none kafka_topic_partition_under_replicated_partition
Categraf Partition 副本数量 none kafka_topic_partition_replicas
Categraf 各个 Topic 每秒消费消息量 none sum(irate(kafka_consumergroup_current_offset[3m])) without (partition)
Categraf 各个 Topic 每秒生产消息量 none sum(irate(kafka_topic_partition_current_offset[3m])) without (partition)
Categraf 各个 Topic 的 Partition 数量 none kafka_topic_partitions
2

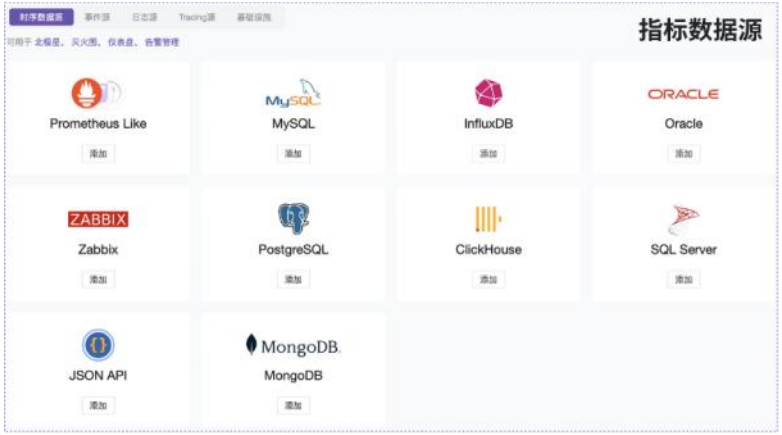

overview
brokers
3
topics
32
partitions
944
Replicas
throughput
Messages produced per second
Messages consumed per second
Latency by Consumer Group
partition/replicate
Partitions per Topic
Partitions Under Replicated
3

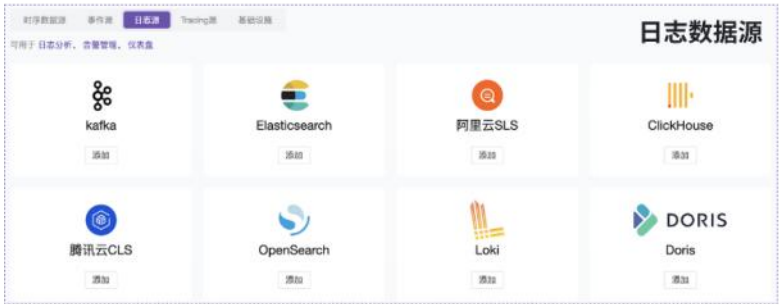

Kafka
采集说明
采集模板
指标说明
仪表盘
告警规则
kafka_by_categraf
请输入搜索关键字
名称
kafka 数据有丢失风险-副本数小于3
kafka 服务宕机
kafka 消费能力不足-延迟超过5分钟
4

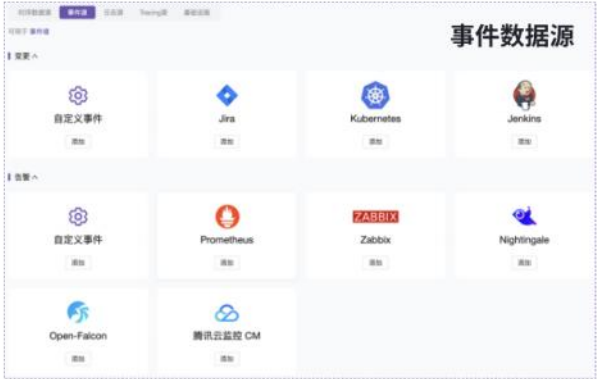
Flashcat

统一的数据源集成

多数据源集成





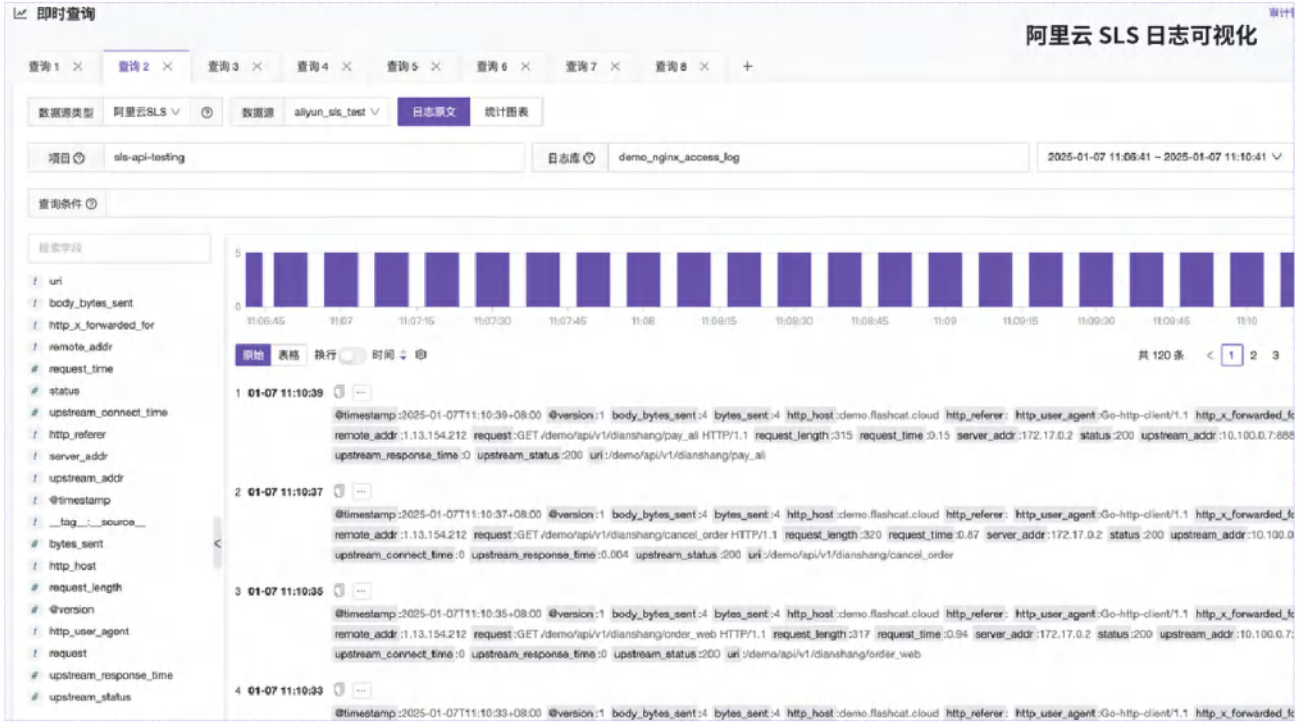


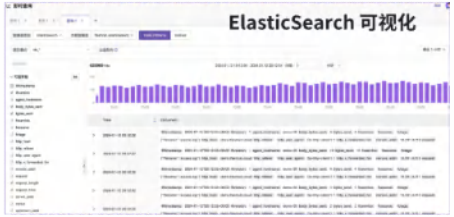
对接40多种数据源

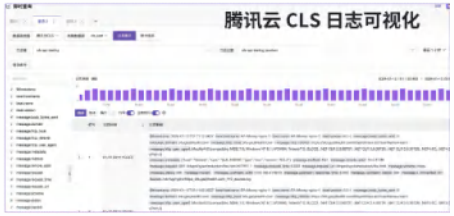
与企业已有的可观测性配套，深度集成融合，对其进行统一分析

日志统一可视化

Flashcat

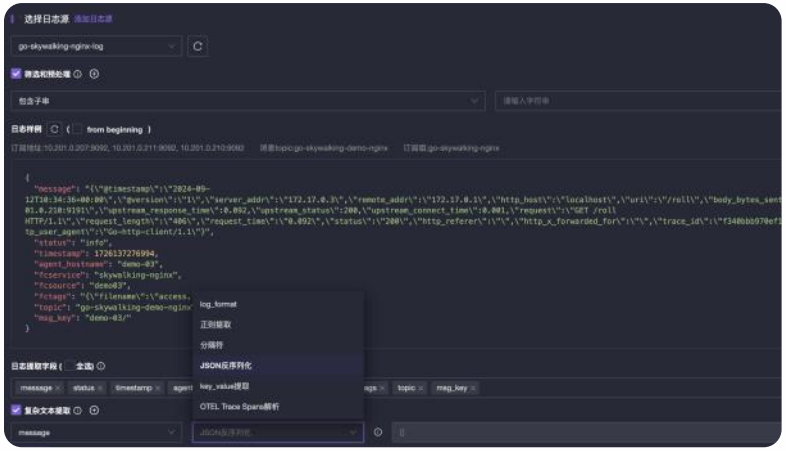


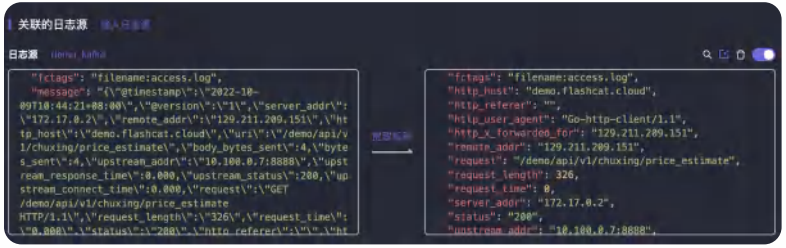


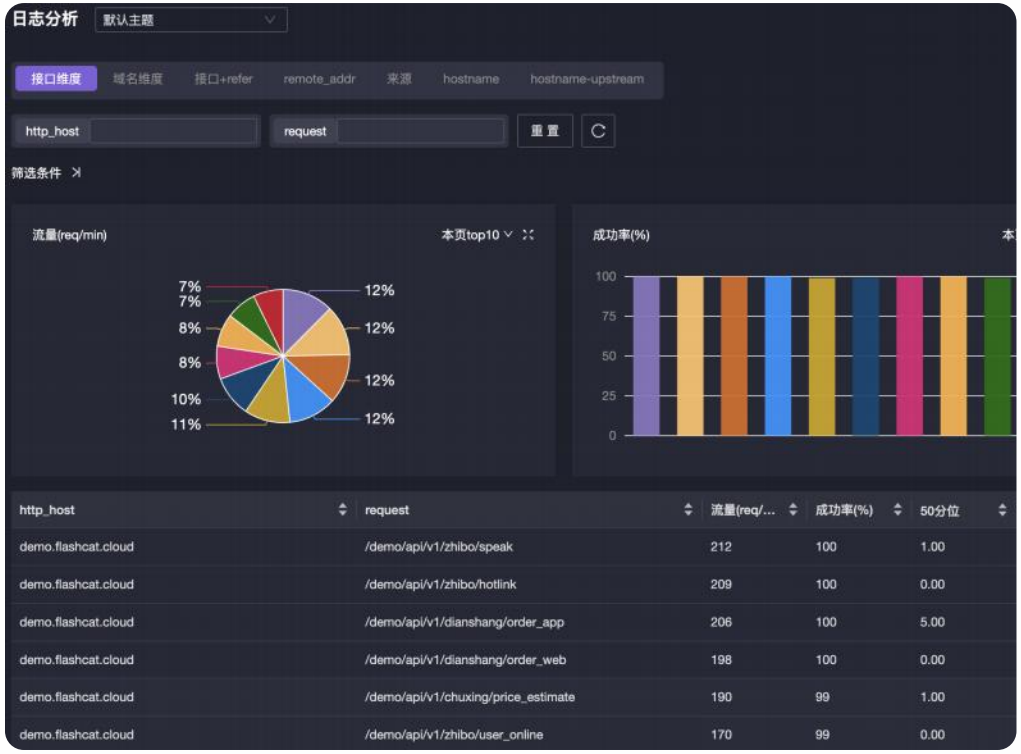


日志转换和报表

Flashcat





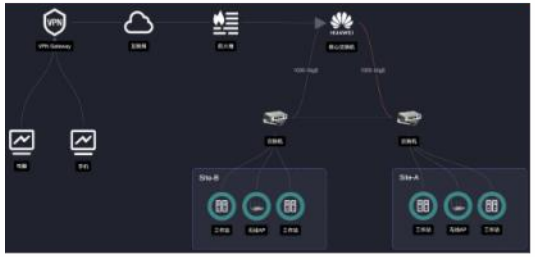


http_host	request	流量(req/...	成功率(%)	50分位
demo.flashcat.cloud	/demo/api/v1/zhibo/speak	212	100	1.00
demo.flashcat.cloud	/demo/api/v1/zhibo/hotlink	209	100	0.00
demo.flashcat.cloud	/demo/api/v1/dianshang/order_app	206	100	5.00
demo.flashcat.cloud	/demo/api/v1/dianshang/order_web	198	100	0.00
demo.flashcat.cloud	/demo/api/v1/chuxing/price_estimate	190	99	1.00
demo.flashcat.cloud	/demo/api/v1/zhibo/user_online	170	99	0.00

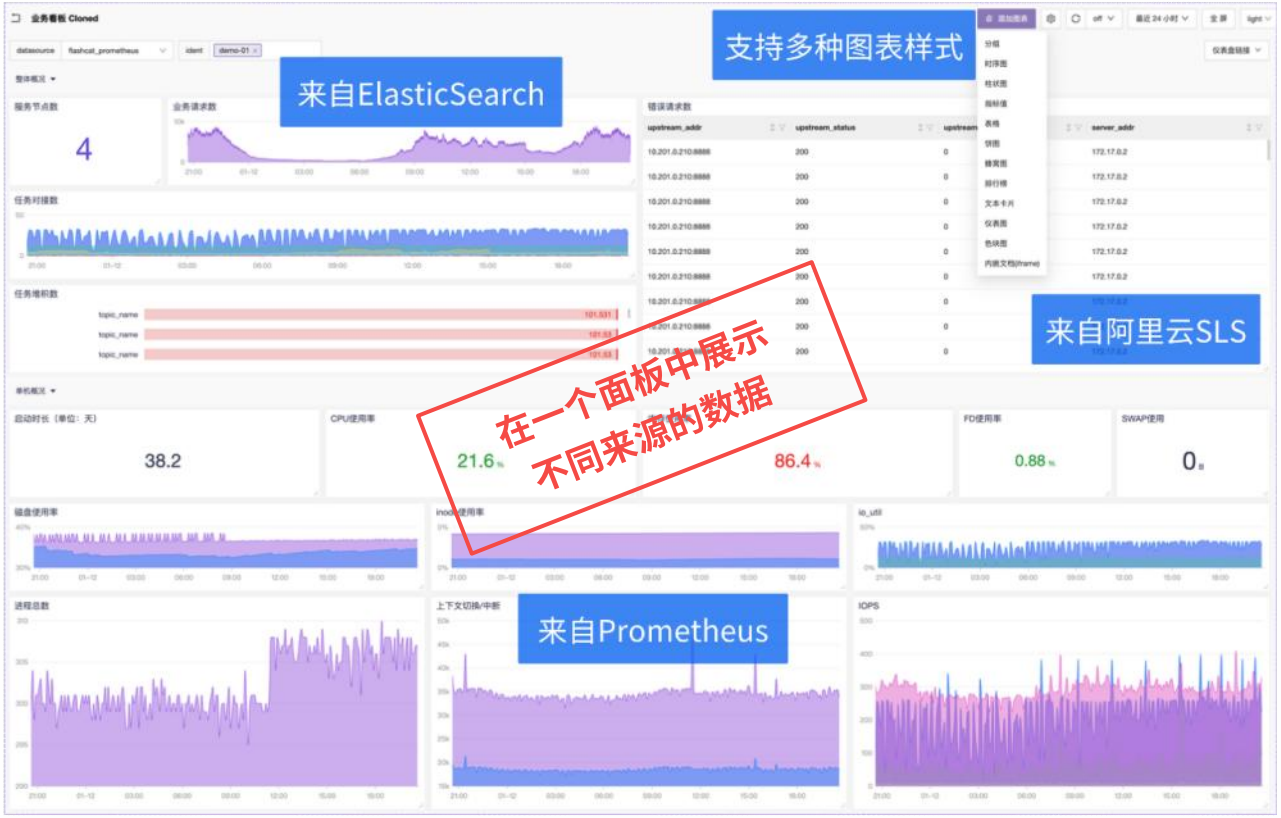
统一的仪表盘

Flashcat

自定义绘制拓扑图



支持导入Grafana仪表盘



针对各种数据源集中配置和管理告警策略

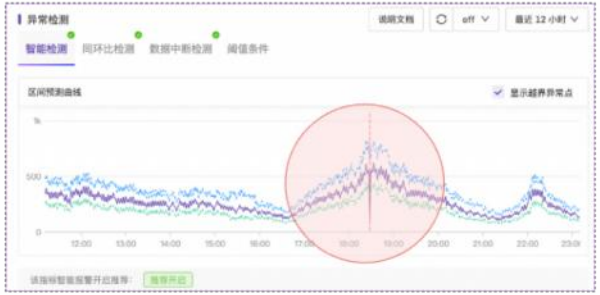
Flashcat

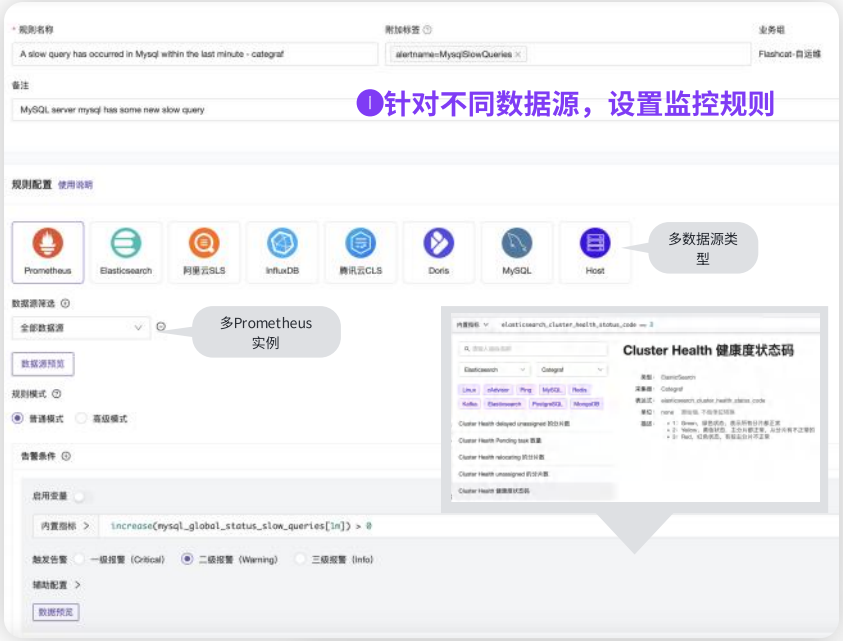
多集群Prometheus监控告警

阿里云SLS监控告警

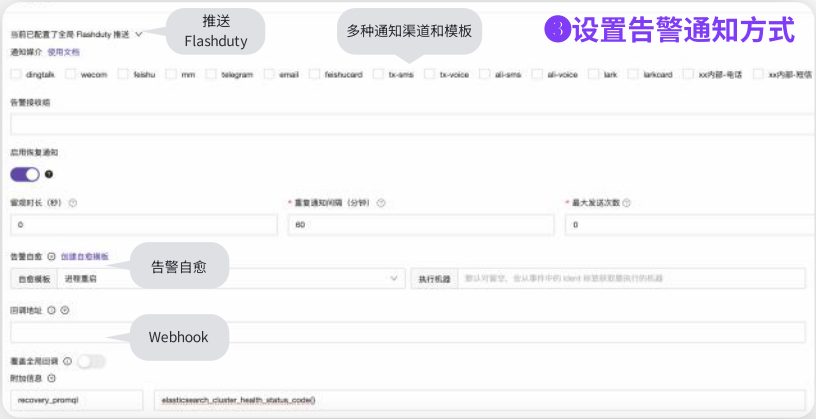
ElasticSearch日志监控告警

智能监控



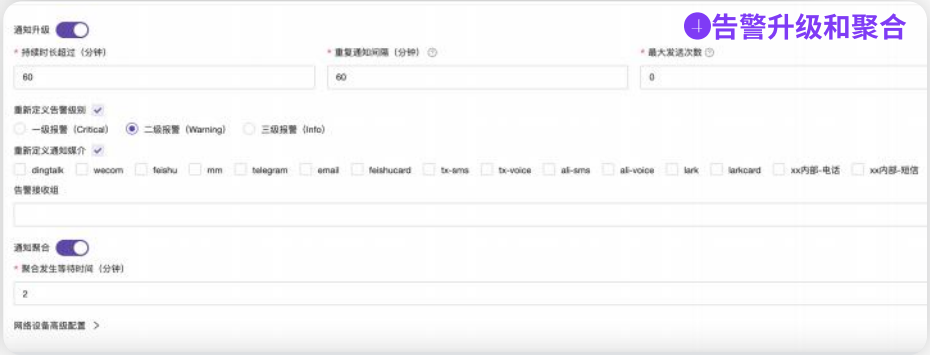


❶针对不同数据源，设置监控规则



❸设置告警通知方式

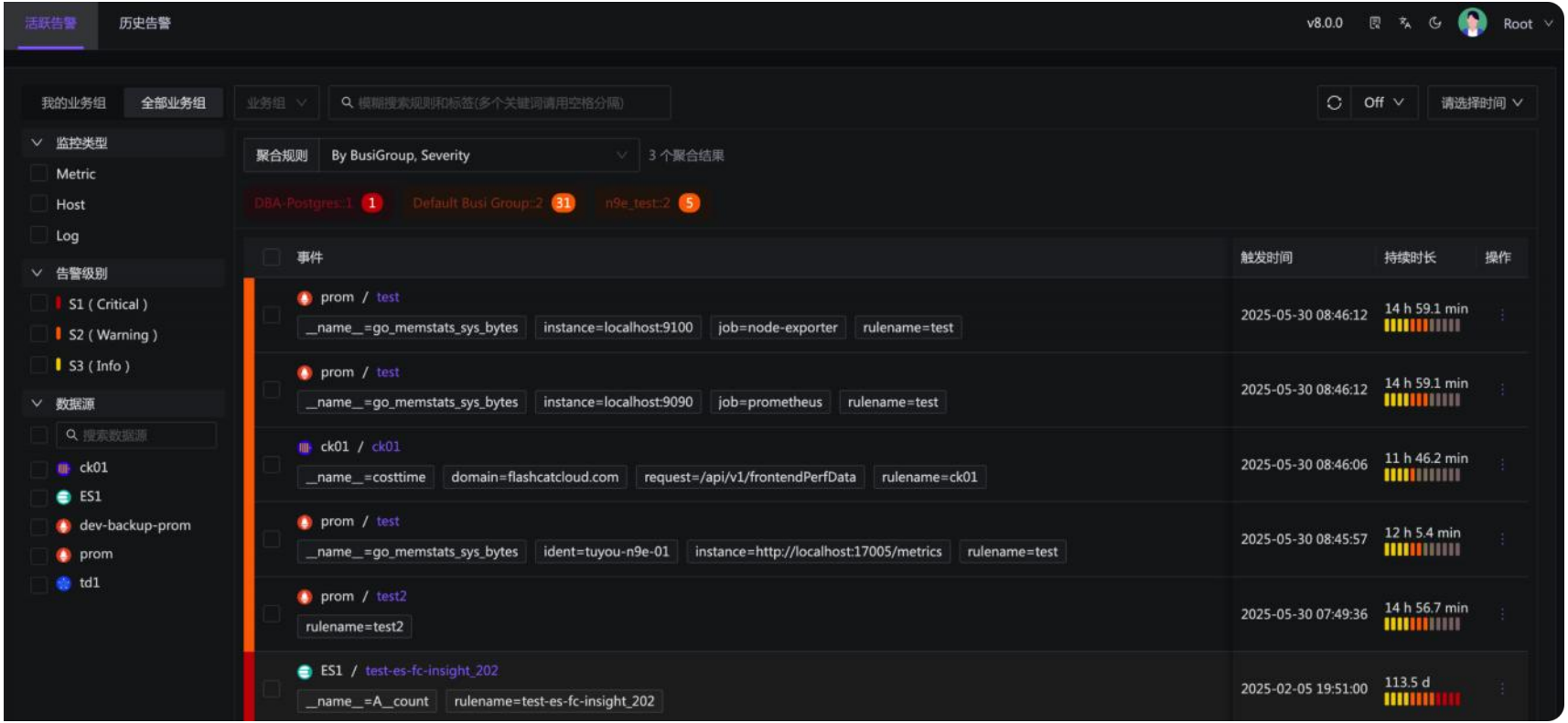
❷自定义监控生效周期



❹告警升级和聚合

Flashcat

告警信息全景看板



监控告警大屏看板

Flashcat

FlashDuty

Flashduty | 统一告警On-call平台

告警 On-call 领域面对的挑战

FlashDuty

01

告警散落各处 告警多

云上、云下多套监控系统，告警散落各处，没有集中的分派通知策略，缺少全局视图

02

告警风暴

底层故障经常导致告警风暴，手机震动不止，影响故障处理，夜间尤其痛苦，工程师体验差

03

告警遗漏

重要告警容易被淹没、遗漏；告警责任不到人，没有值班、备份、升级等机制，告警长时间无人响应

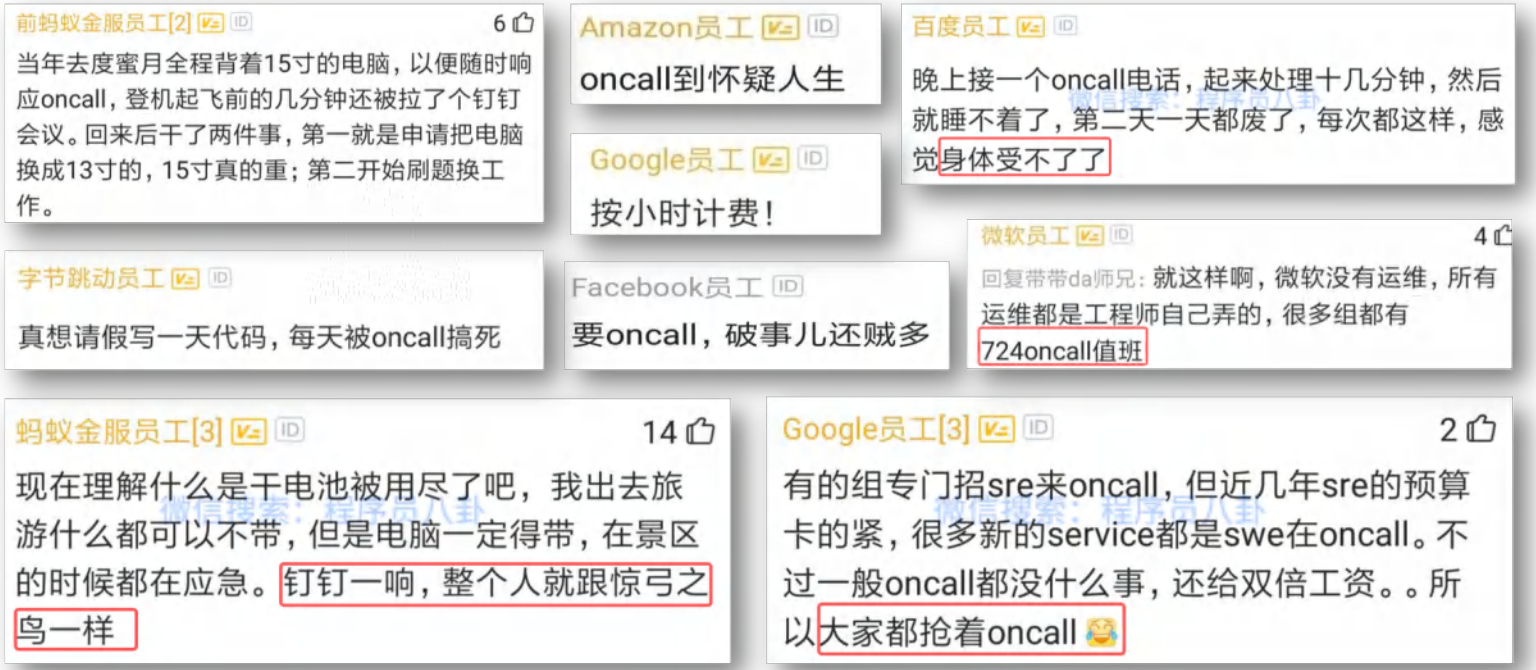
04

处理效率低 协同差

告警处理缺乏协同，过程不透明，信息难以共享，知识难以沉淀。不方便移动端查看处理

用户往往先于技术团队发现故障，久而久之，对 IT 的满意度持续走低

大厂 On-call 体验比拼



一站式告警On-call平台，加速企业告警响应

FlashDuty

Step 1

告警集成

将各种不同监控工具发出的告警，实时的收集到统一的平台中，开箱即用

Step 2

标签增强

与CMDB 等元数据中心打通，实时丰富告警标签和和告警上下文

Step 3

聚合降噪

对相似的告警进行聚合、对频发的告警进行收敛，显著降低告警数量

Step 4

排班值班

灵活排班，满足日常值班、节假日值班、调班等场景，避免整个团队被频繁打断

Step 5

告警分派/认领/升级

集中管理告警分发策略，按时段、按标签灵活分发告警到不同的对象

Step 6

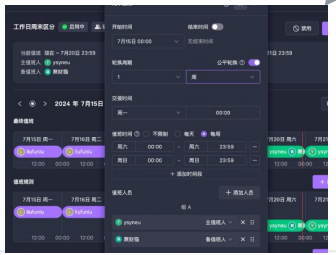
转派/协同

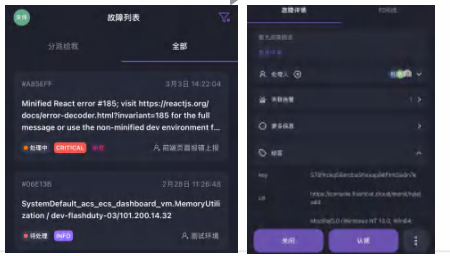
正确的时间通知需要的人，共享告警处理上下文信息，避免告警漏处理

Step 7

告警通知

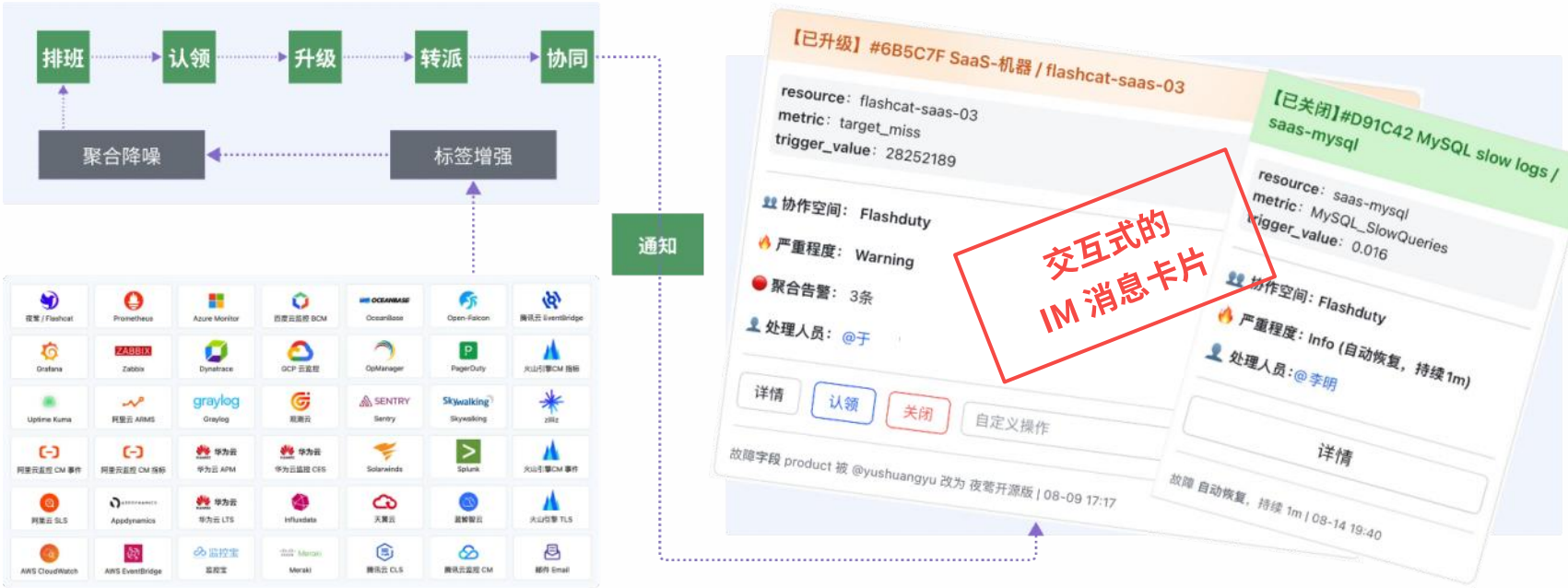
内置多种告警通知方式，及时、准确的通知相关人员，在 IM 中实时交付响应告警





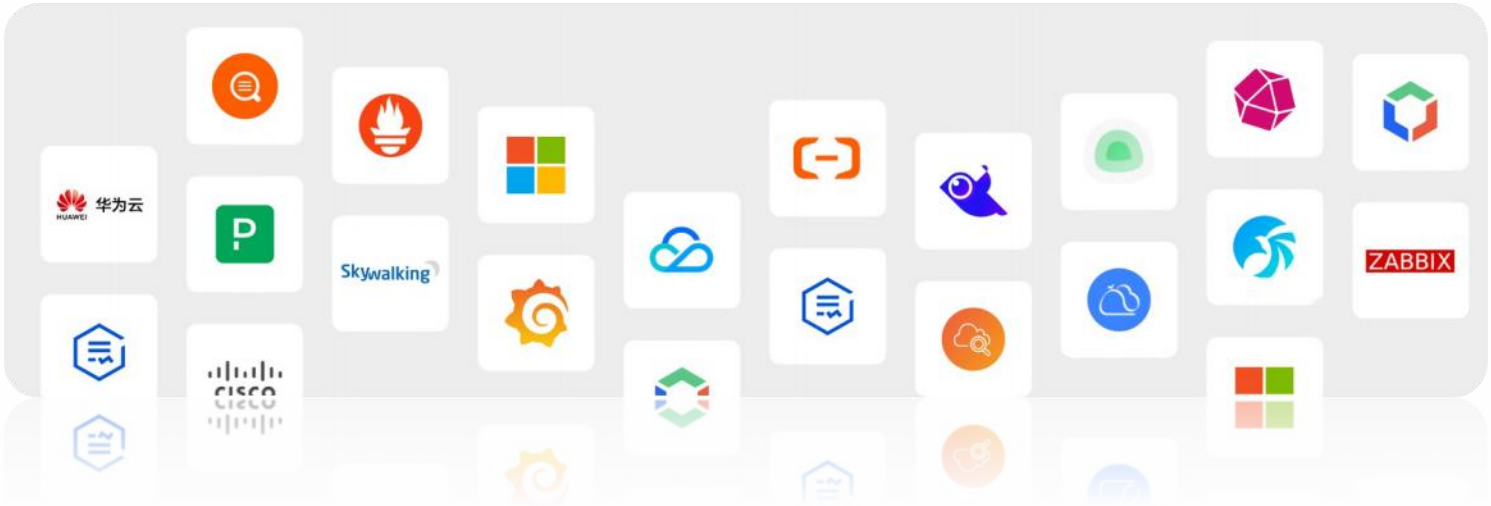
一站式告警On-call平台，加速企业告警响应

FlashDuty



一个平台集中处置所有告警

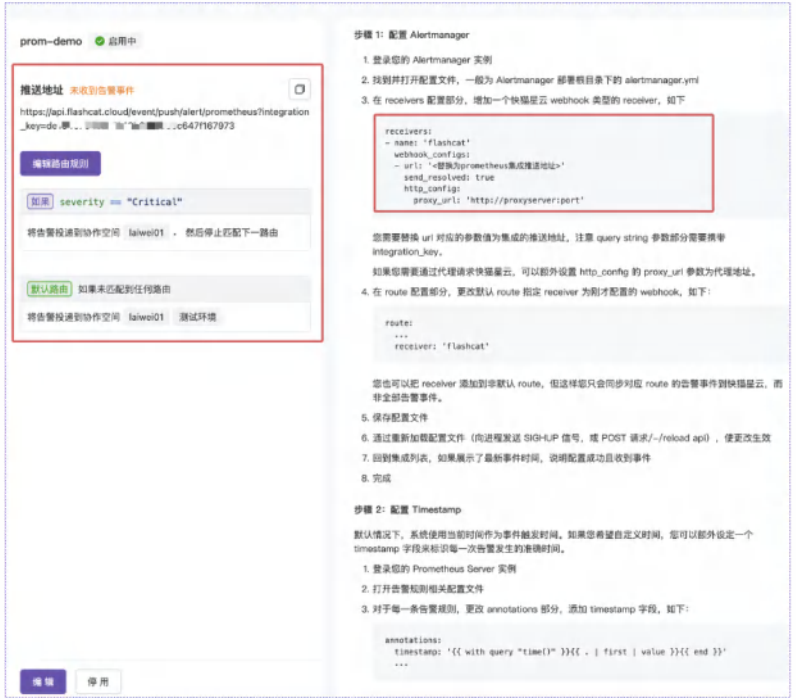
30+常见监控工具（开箱即用）+ 1 套标准 HTTP 协议 + 邮件解析（覆盖自研监控）



对接各种告警事件

Flashcat

选择数据源 → 获取告警推送地址 → 修改 AlertManager Webhook 地址 → 集中查看和处理告警



数据增强，丰富告警上下文

Flashcat

历史变更事件

70% 的故障由变更导致

CMDB 元数据

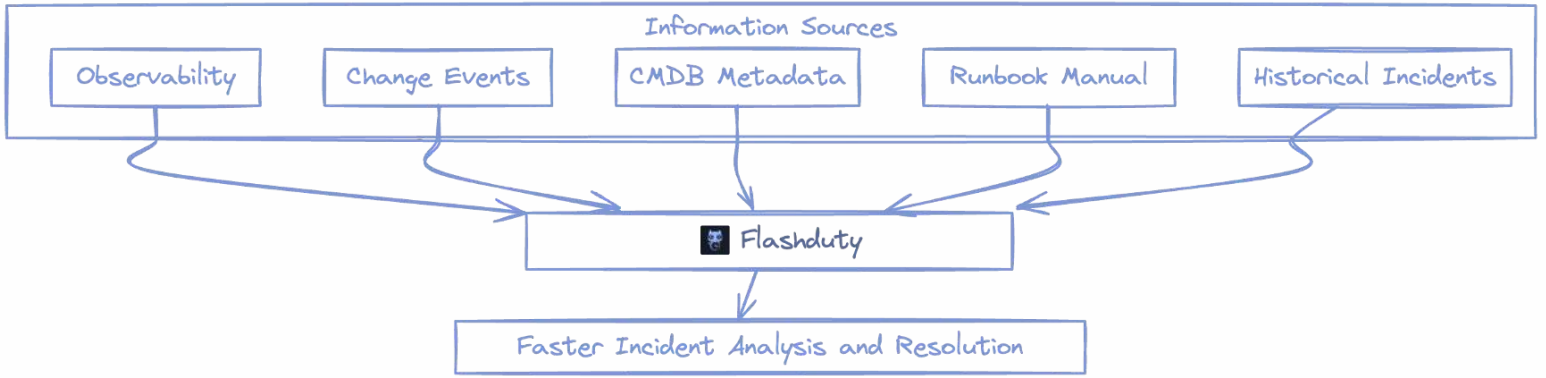
资产关系依赖映射

知识库和 SOP

在故障信息中展示 SOP

历史故障记录

参考相似故障的解决办法



告警降噪，显著降低告警数量

Flashcat

对相似告警进行聚合

- 事件 => L1 告警 => L2 故障
- 减少通知，避免告警风暴
- 规则聚合、智能聚合，至多降噪90+%

对频发告警进行收敛

- 避免狼来了效应
- 避免频繁被打断

排班值班，避免整个团队被频繁打断

Flashcat

1

满足各类场景

日常、节假日、调班、限时、公平轮换

2

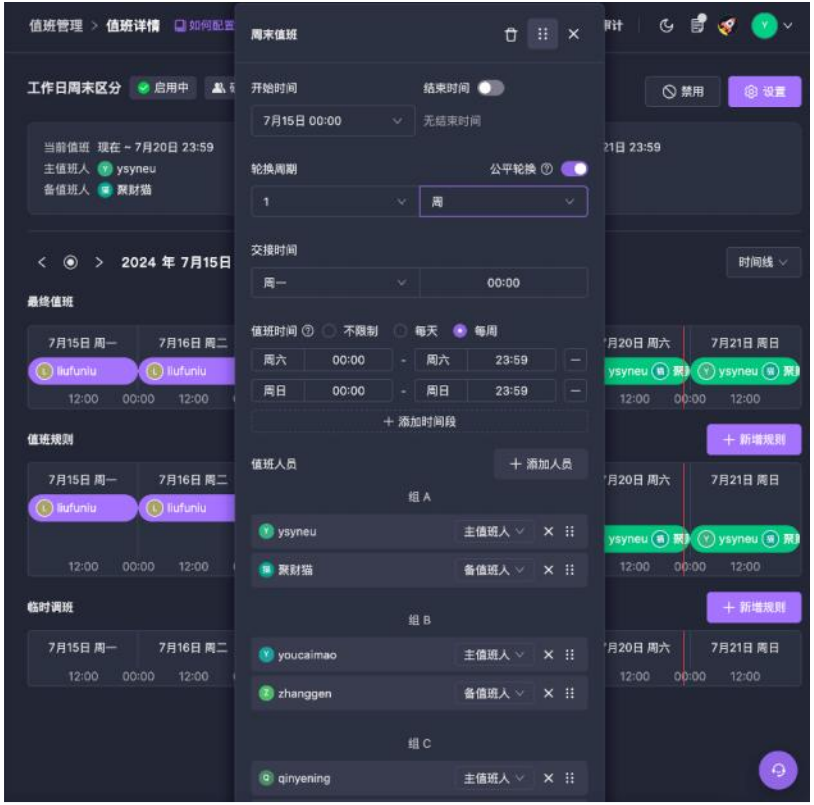
建立主备值班机制

支持多人同时，按角色值班

3

分派告警到值班人

不要随机分派，更不要分派到整个团队



告警升级和转派，正确的时间通知需要的人

Flashcat

1. **制定升级路线**

   为故障制定清晰的升级路线

2. **A：确保问题得到响应**

   不因漏处理告警而引发更大损失

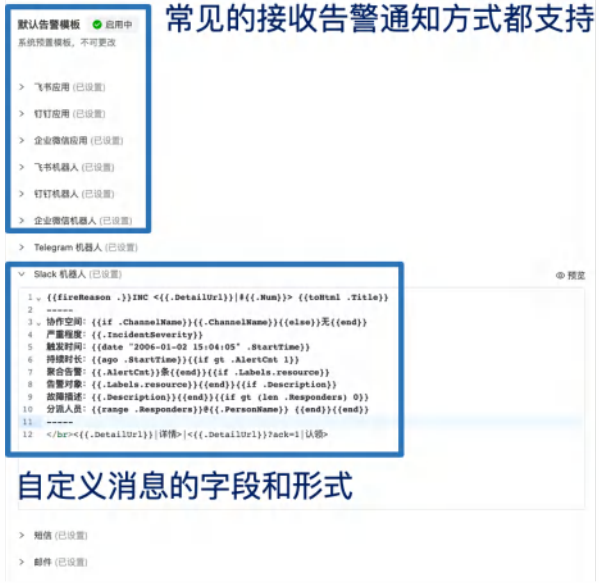
3. **B：从容应对紧急情况**

   发生紧急情况不慌张，有协作

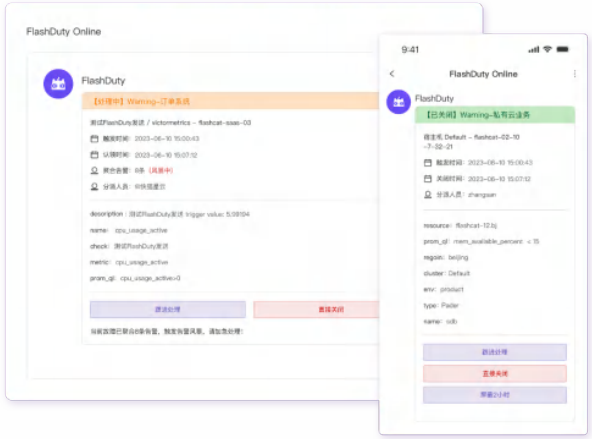
通知，及时精准高效的触达干系人

Flashcat

常见的接收告警通知方式都支持



交互式的 IM 消息卡片



手机App(安卓、iOS)

自定义操作，集成工作流

Flashcat

API 集成

以 按钮 形式集成到控制台、IM 消息卡片

自动化流程

集成任何自动化、SOP 流程

典型场景

- 重启主机
- 回滚变更
- AI 根因分析
- 一键拉群
- 发布 Status Page

IM(飞书、企微、钉钉、手机App)

数据驱动， 推动告警处理流程持续优化

MTTx 统计

MTTx 分类趋势

TopN 告警统计

工作量统计

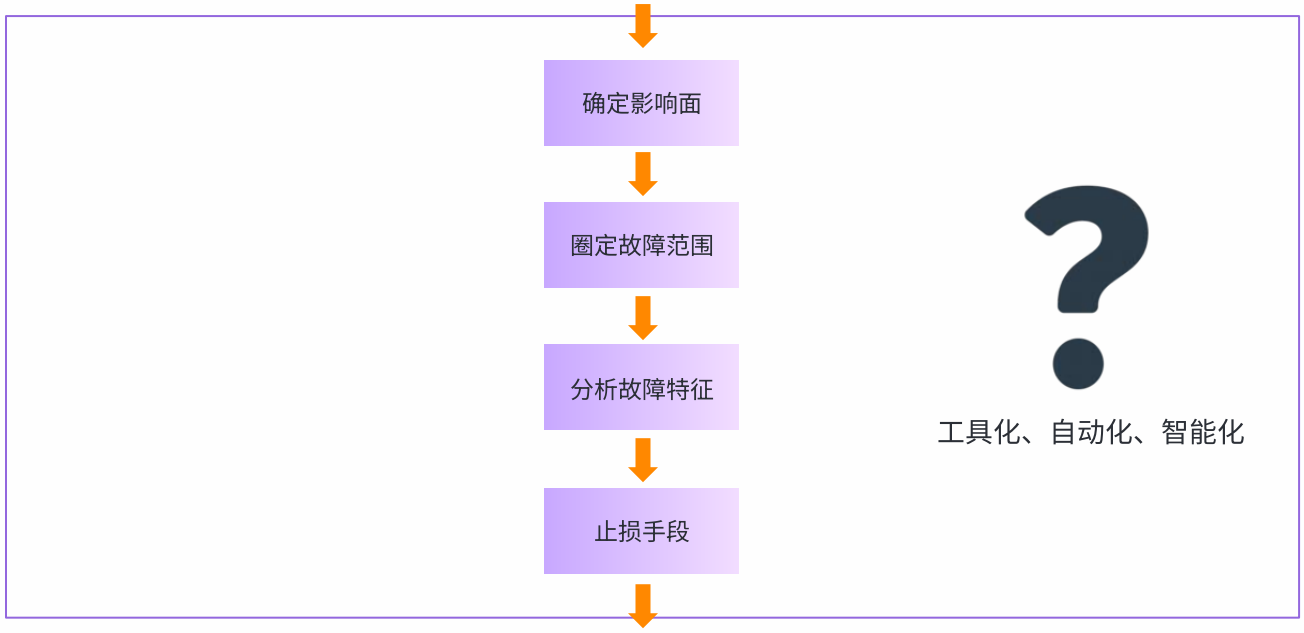
数据统计，通过数据驱动不断推动告警治理和优化

Flashcat | 智能的故障发现定位平台

故障发现定位模型

Flashcat

主动/被动发现业务受损



工具化、自动化、智能化

流量调度 Traffic Control	单点切换 Single Point Switch Over	服务降级 Service Degradation	限流 Throttle Control	自动熔断 Auto Circuit Breaker	一键回滚 One-button Rollback	一键重启 One-button Reset	一键扩容 One-button Scale up

故障发现定位模型

Flashcat

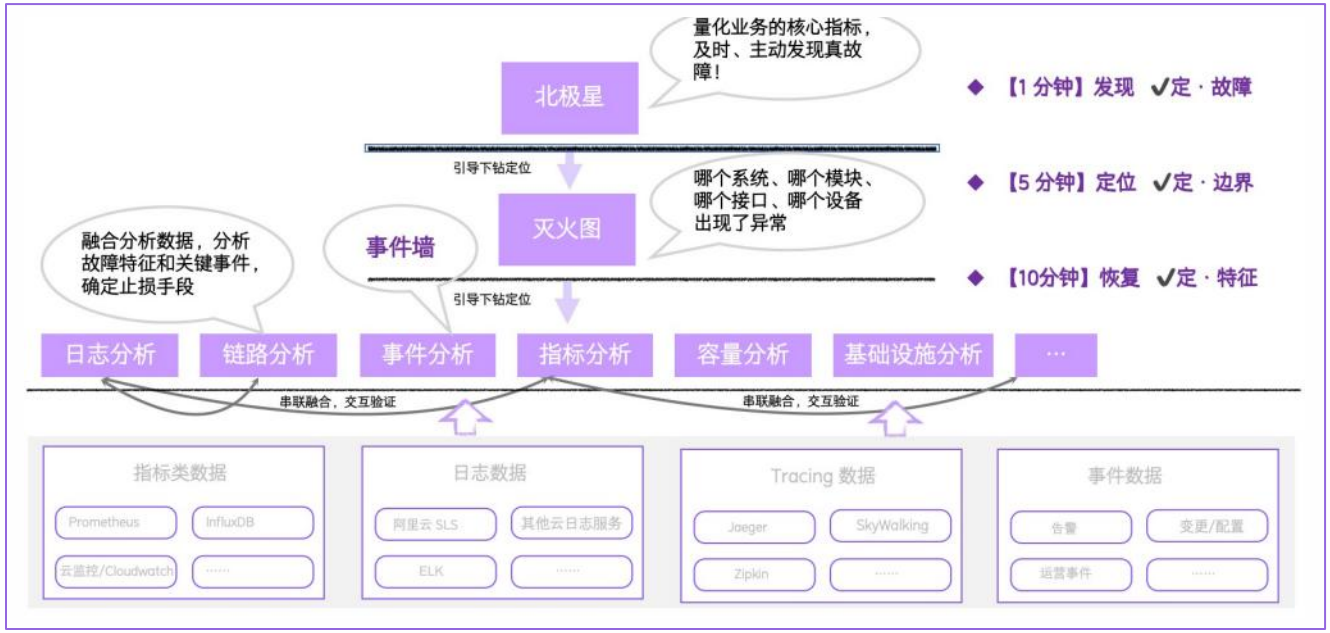
主动/被动发现业务受损

确定影响面

圈定故障范围

分析故障特征

止损手段



流量调度	单点切换	服务降级	限流	自动熔断	一键回滚	一键重启	一键扩容
Traffic Control	Single Point Switch Over	Service Degradation	Throttle Control	Auto Circuit Breaker	One-button Rollback	One-button Reset	One-button Scale up

Flashcat北极星——发现真故障，确定影响面和程度

✔ 北极星指标必须是公司上下人人都**容易直观理解**的，含义和重要性不言自明的。

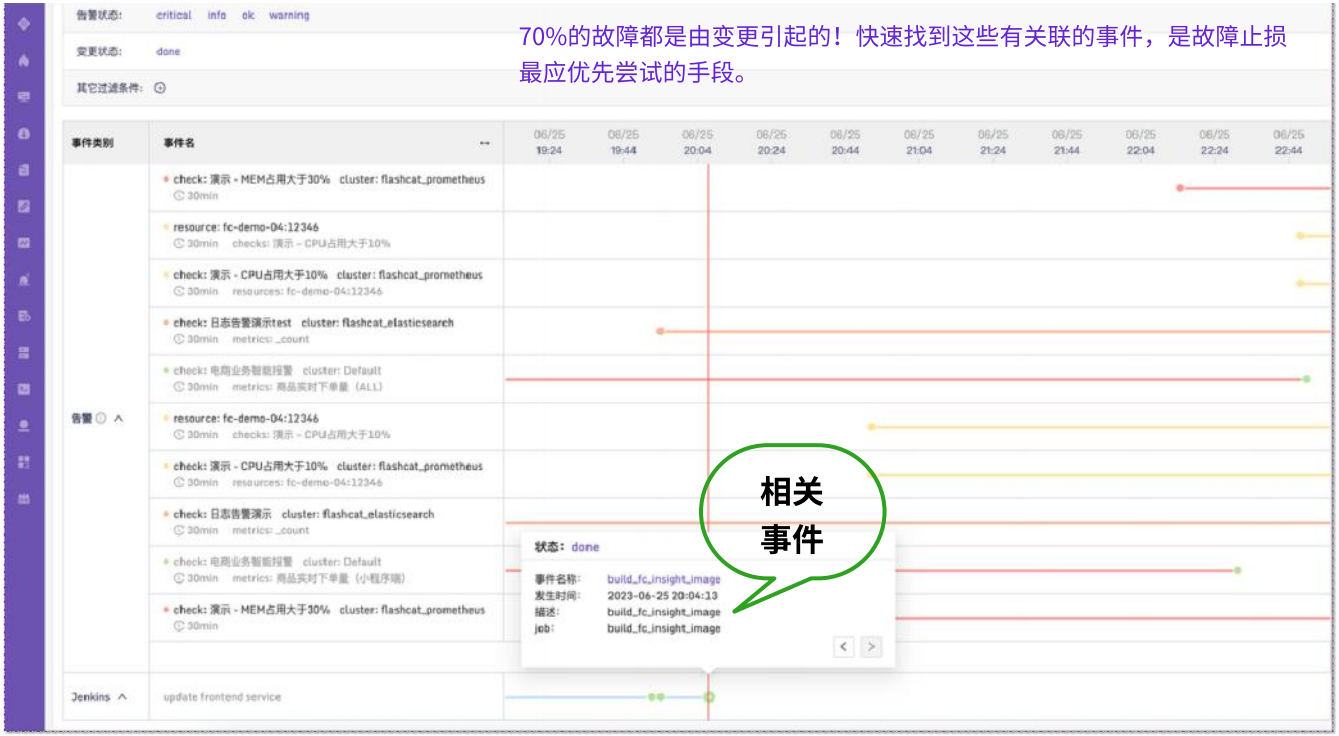
✔ 北极星指标的配置和生成应该是非常**简便易得**的。

✔ 北极星指标必须是**实时**的，这样才能第一时间发现业务受损的情况。

✔ 针对北极星指标的异常波动检测是要非常**准确**的，否则就是狼来了。

✔ 针对北极星告警，需制定明确的应急响应流程。

Flashcat事件墙——快速确定或者排除相关事件



70%的故障都是由变更引起的！快速找到这些有关联的事件，是故障止损最应优先尝试的手段。

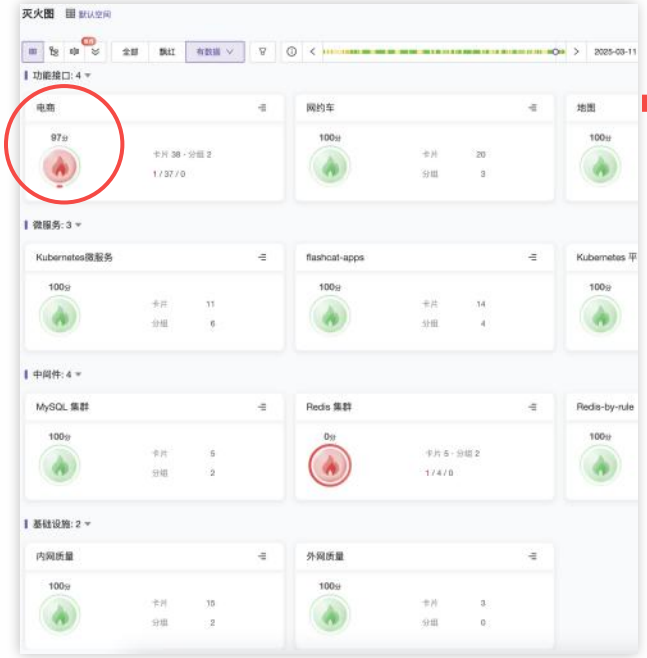
关键告警

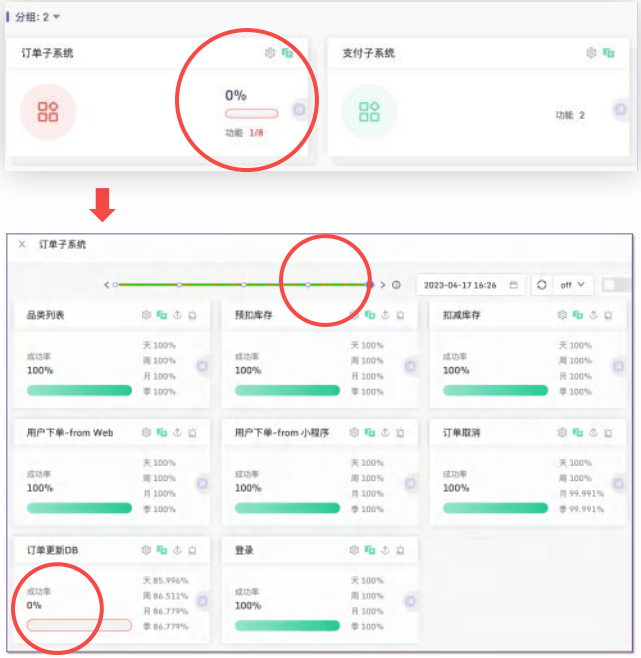
关键上线

配置变更

促销活动

Flashcat灭火图——圈定故障范围和对象





Metrics

Logs

Traces

Events

Profling

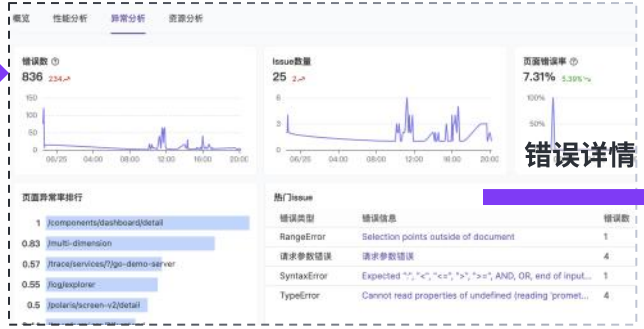
Dashboard

1、不断的将更多的经验、知识、能力，沉淀固化到下钻链路上，让每位工程师都能拥有轻松调查故障的能力。
2、在一个平台上完成指标、日志、链路追踪数据的统一存储、可视化与分析。

RUM —— 真实用户体验监控

Flashcat

错误分析



错误详情

性能分析

资源分析

性能详情

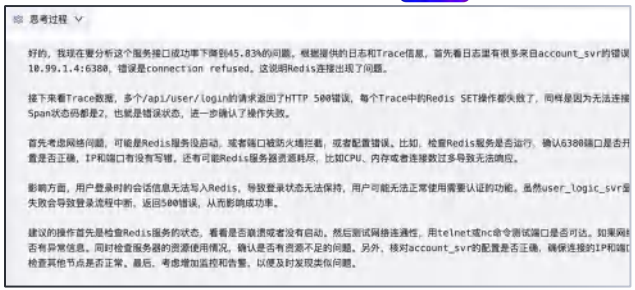
APM —— 应用性能监控

Flashcat

AI 分析


服务接口
成功率(%)(平均)
52.381%
流量(求和)
21
1391
5


核心问题
Redis服务不可达：日志及Trace均指向Redis服务器 10.99.1.4:6380 连接失败，导致用户登录接口 /api/user/login 出现HTTP 500错误，最终引发服务成功率暴跌至45.83%。
直接证据
1. 日志层：
account_svr 持续报错 set session to redis failed dial tcp 10.99.1.4:6380: connect: connection refused
错误时间分布密集（5分钟内连续5次），表明Redis连接持续不可用
2. Trace层：
所有失败请求的根Span /api/user/login 均返回500状态码
Redis SET 子Span明确记录连接拒绝错误，Span状态码为2（错误）
3. 网络特征：
目标地址为内网IP 10.99.1.4，排除公网网络问题
端口 6380 为非常用Redis端口（默认6379），需检查配置合理性


思考过程
好的，我现在要分析这个服务接口成功率下降到45.83%的问题，根据提供的日志和Trace信息。首先看日志里有很多来自account_svr的错误
10.99.1.4:6380，错误是connection refused。这说明Redis连接出现了问题。
接下来看Trace数据，多个/api/user/login的请求返回了HTTP 500错误，每个Trace中的Redis SET操作都失败了，同样是因为无法连接
Span状态码都是2，也就是错误状态，进一步确认了操作失败。
首先考虑网络问题，可能是Redis服务没启动，或者端口被防火墙拦截，或者配置错误。比如，检查Redis服务是否运行，确认6380端口是否开
置是否正确，IP和端口有没有写错。还有可能Redis服务器资源耗尽，比如CPU、内存或者连接数过多导致无法响应。
影响方面，用户登录时的会话信息无法写入Redis，导致登录状态无法保持，用户可能无法正常使用需要认证的功能。虽然user_logic_svr
失败会导致登录流程中断，返回500错误，从而影响成功率。
建议的操作首先是检查Redis服务的状态，看看是否崩溃或者没有启动，然后测试网络连通性，用telnet或nc命令测试端口是否可达。如果网
否有异常信息。同时检查服务器的资源使用情况，确认是否有资源不足的问题。另外，核对account_svr的配置是否正确，确保连接的IP和端口
检查其他节点是否正常。最后，考虑增加监控和告警，以便及时发现类似问题。

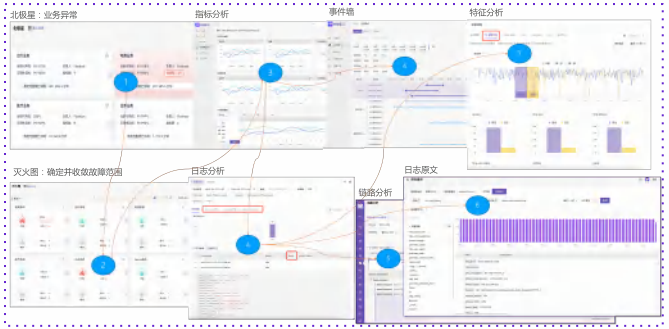

异常分析
1. 异常现象
检测到多条来自 account_svr 服务的 error 级别日志，错误内容均为 set session to redis failed dial tcp
80: connect: connection refused 。
2. 原因推测
网络/服务不可达：Redis 服务器 10.99.1.4:6380 无法连接，可能是由于以下原因：
Redis 服务未启动或崩溃
网络防火墙/安全组规则阻断了 6380 端口
IP 地址或端口配置错误
依赖服务异常：Redis 集群节点故障或资源耗尽（如 CPU、内存、连接数等）
3. 影响范围
会话（Session）信息无法写入 Redis，可能导致用户登录状态失效、鉴权失败等问题。
虽然 user_logic_svr 显示账户创建成功，但会话存储失败可能导致后续业务流程中断。
4. 建议排查方向
检查 Redis 服务 10.99.1.4:6380 的存活状态及日志
使用 telnet 10.99.1.4 6380 或 nc -vz 10.99.1.4 6380 验证端口连通性
检查 account_svr 的 Redis 连接配置（IP、端口、密码等）
监控 Redis 服务的资源使用情况（CPU、内存、连接数）


Tracing
发现异常：
检测到异常Trace
分析结论
以下Trace存在异常，原因如下：
1. HTTP 500错误：根Span /api/user/login 的 http.status_code 为500，表示服务器内部错误。
2. Redis连接失败：子Span Redis SET 中均出现错误事件 Redis SET failed: dial tcp 10.99.1.4:6380: connect: connection refused ，表明Redis服务（IP: 10.99.1.4:6380 ）无法连接。
3. Span状态码异常：相关Span的 status.code 为2（错误状态），进一步确认操作失败。
结论：所有异常Trace均因Redis服务不可用导致，需检查Redis实例（10.99.1.4:6380）的网络连通性或服务状态。
ai-dev-test-jaeger Service: user_logic_svr Operation: /api/user/login
疑似异常Trace: 共 5 个Trace 收起
user_logic_svr: /api/user/login 6e22151b025d577c
123.88ms
3 Spans
user_logic_svr 1
account_svr 2
Today 9:55:53am
14 分 前
user_logic_svr: /api/user/login 1c3e7ea5ae941a54
79.98ms

面向不同人群建立针对性的视图

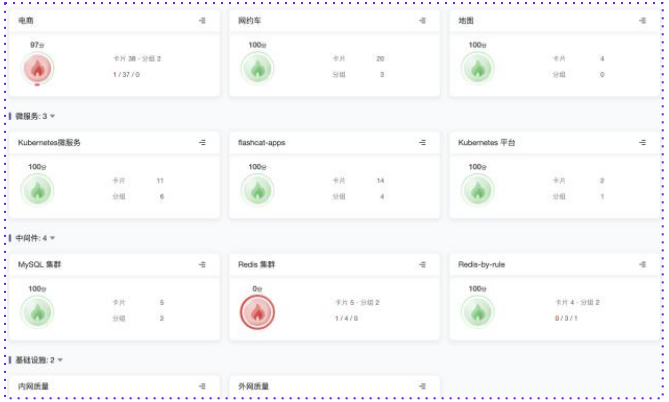
1、业务/管理层视图

业务状态、全局视图



2、研发视图

串联数据 排查问题
细节分析



3、运维视图

全局IT状态
故障/问题/告警视图

用户案例

Flashcat

医疗及连锁

香港医管局、高济健康、益丰大药房

餐饮连锁

星巴克、海底捞、金拱门、吉野家、零食很忙

游戏

莉莉丝游戏、悠星网络、鹰角网络、途游游戏

交通出行

小马智行、六分科技、真点科技

路特斯科技、保时捷、吉利汽车

嘀嗒出行、阳光出行、UU跑腿、八维通

科技

地平线科技、小牛电动、安克创新、影石Insta360

鹿客科技、Zenlayer、新钛云服、融云科技

金融

国泰君安期货、华盛证券、东莞证券

灵均投资、萨摩耶、顽岩资本

其他

海大集团、旭辉集团、中国电信、盛大、中免日上

知乎、Klook客路、作业帮、快看漫画、金山办公

Flashcat

附录

Flashcat

系统架构

水平扩展的部署案例

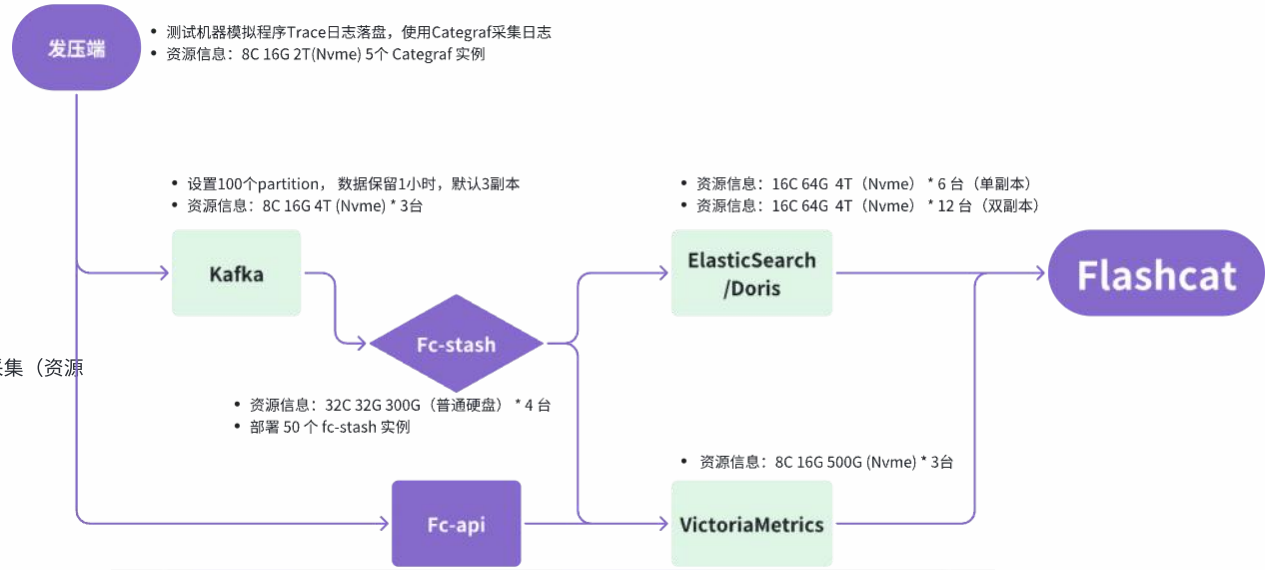
Flashcat

测试目标：

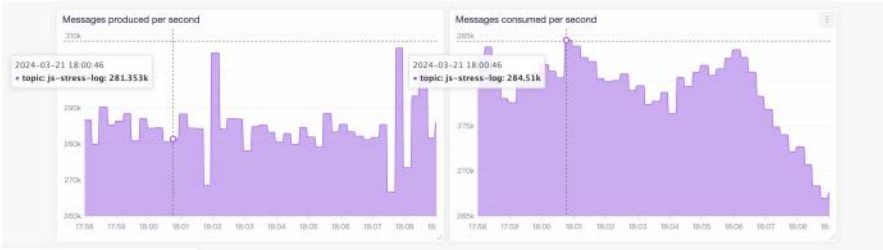
1. 支持峰值20万tps的数据写入
2. 支持每天15TB数据的处理和分析
3. 各组件均可水平扩展

测试结论：

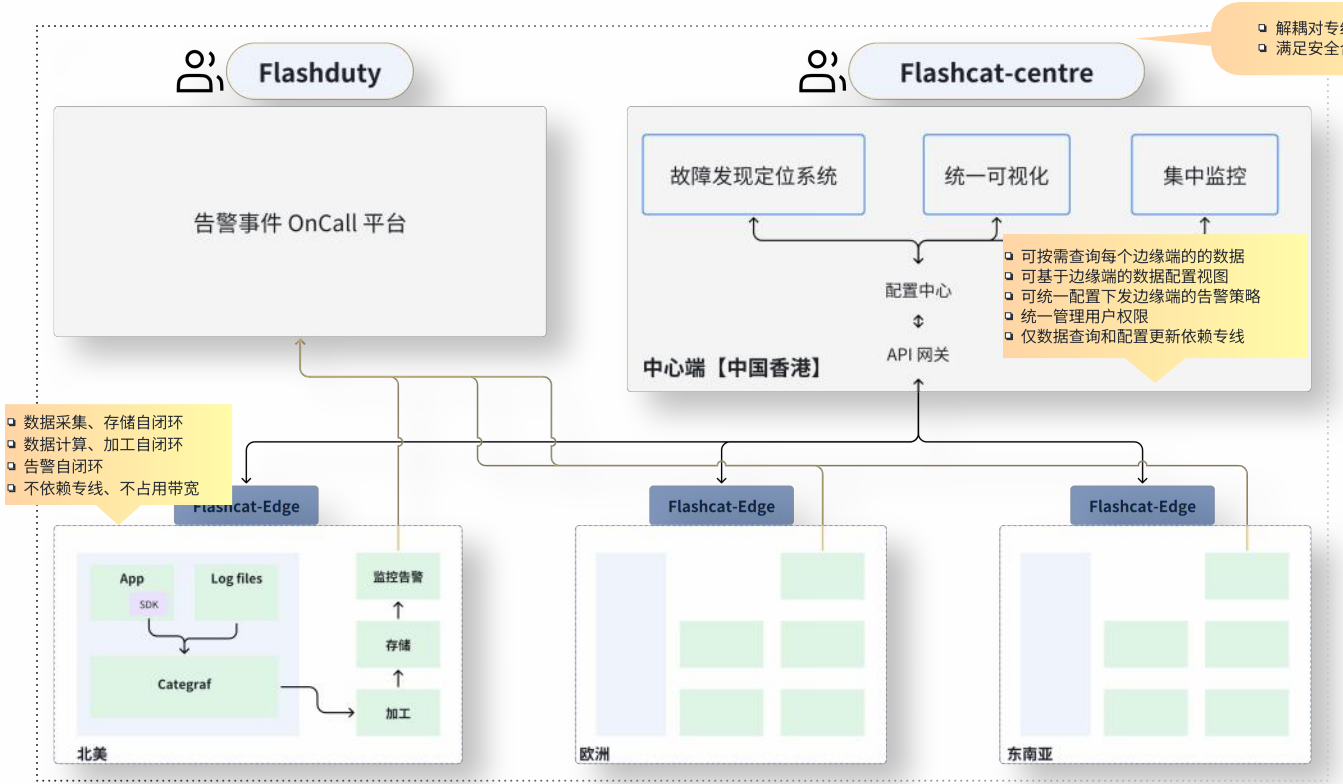
1. 各组件均可通过水平扩展增加来提升处理能力
2. 其中数据采集器Categraf单机达到了单机每秒5万条日志的采集（资源消耗3Core/300MB内存）
3. 最终系统处理数据的峰值达到了28万tps，并稳定运行



1. 在同等资源消耗的前提下，Categraf对日志的采集速度比 filebeat 提升了25%
2. 压测过程中，在继续增加资源的情况下，系统处理数据的峰值达到了58万tps



边缘部署模式

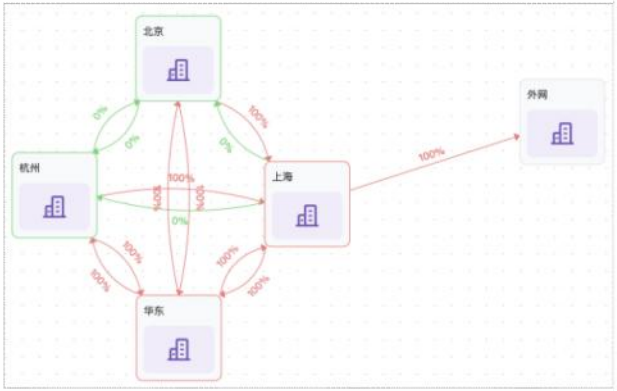


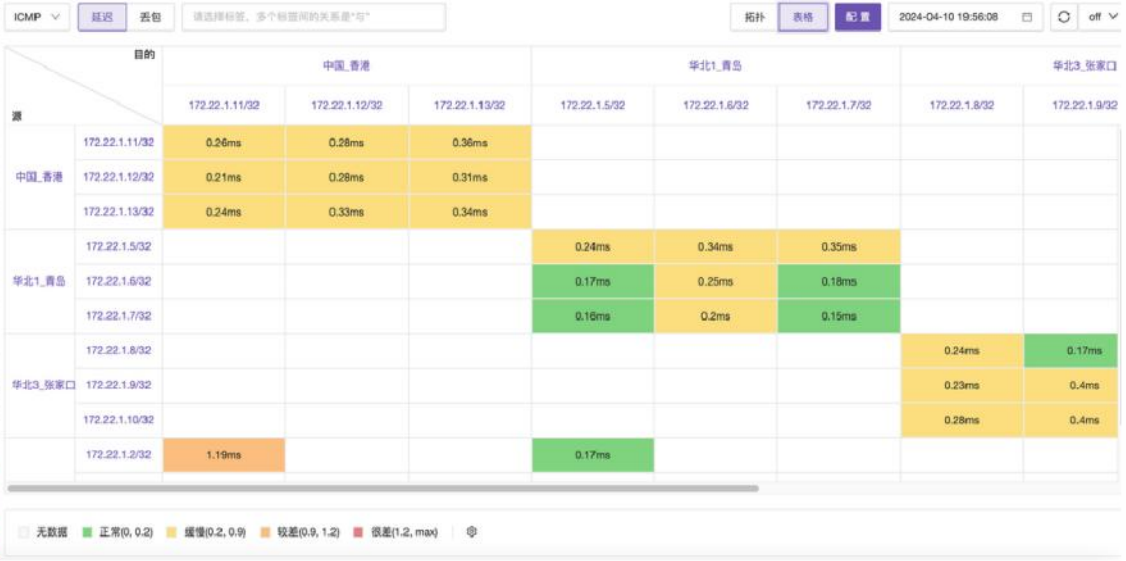
数据隐私保护 vs. 管理便捷性

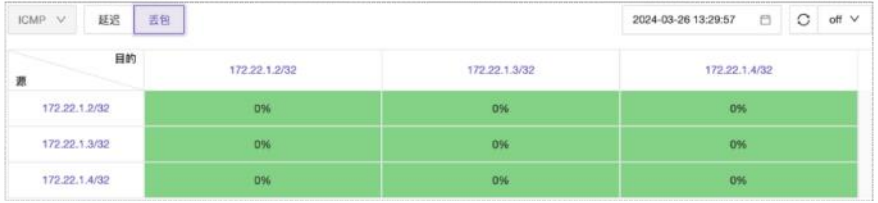
跨境数据传输和集中存储，会带来数据隐私保护方面的隐患，也会消耗更多的跨境传输带宽，成本更高，监控数据延迟大。

把数据闭环存储到各个独立的国家或者市场，在数据分析、监控告警管理层面又会带来不便和挑战。

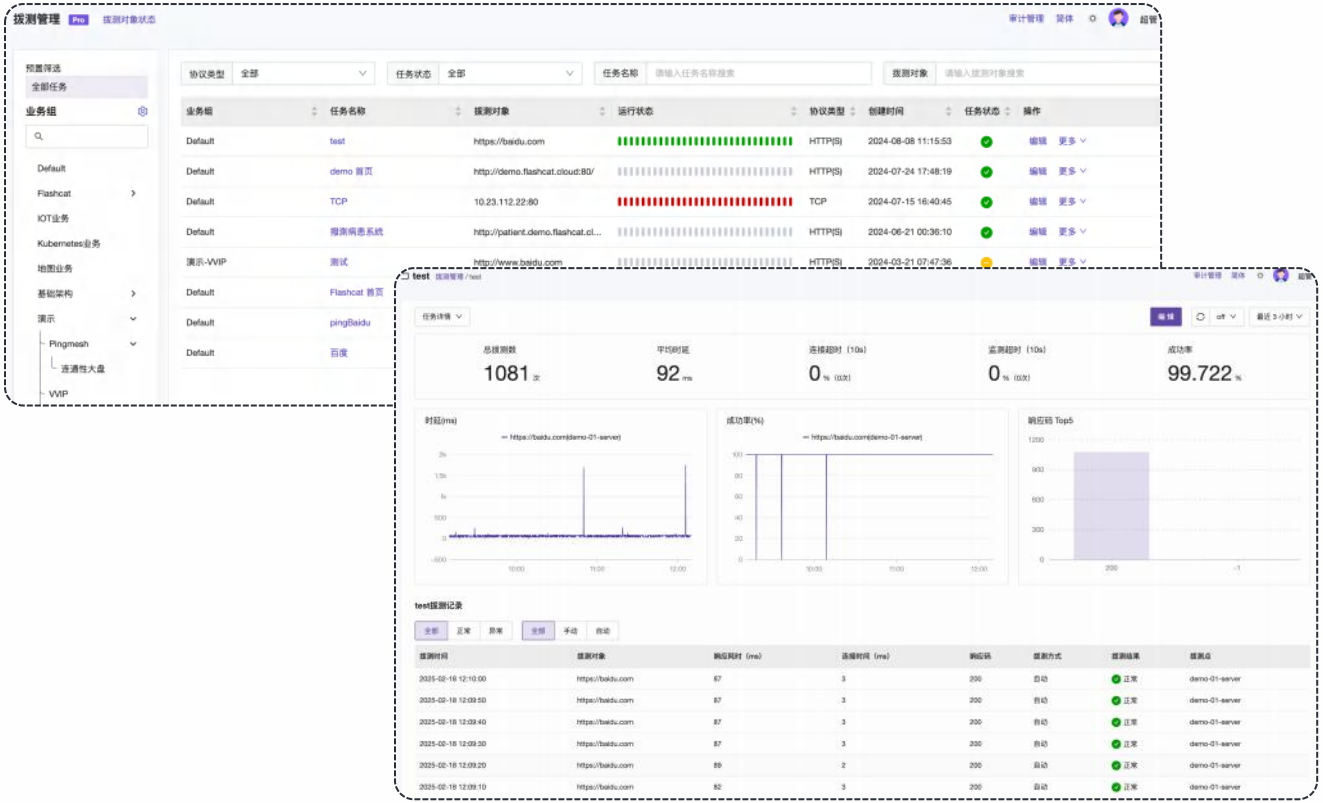
采集器增强 - Pingmesh







采集器增强 - 网络拨测



多协议

HTTP、TCP、UDP、ICMP、WSDL

针对 Reponse 做匹配

多拨测点

可选择安装了Categraf的一个或多个设备

采集器增强 - 持续性能剖析

针对Java等技术栈进行性能分析，生成火焰图

Flashcat

谢谢

https://flashcat.cloud