



第二届CCF·夜莺开发者创新论坛

中国北京 2024.7.26

主办方：中国计算机学会 | 承办方：CCF开源发展委员会、夜莺项目开源社区



中國計算機學會
CHINA COMPUTER FEDERATION



像 Google SRE 一样 OnCall - Flashduty 方法与工具剖析

于双羽

快猫星云, Flashduty 产品研发

中国北京 2024.7.26

主办方: 中国计算机学会 | 承办方: CCF开源发展委员会、夜莺项目开源社区

大纲

1

Google SRE 如何 OnCall

探讨 Google SRE 团队的 OnCall 实践和方法。

2

借助 Flashduty 降低 运维负载

介绍 Flashduty 如何帮助 减轻运维团队的工作负担。

3

借助 Flashduty 加速 告警响应

讲解 Flashduty 如何提高 告警响应速度和效率。

4

Flashduty 产品路线

展望 Flashduty 的未来发展方向和潜在改进。

大厂 OnCall 体验比拼

前蚂蚁金服员工[2]  

6 

当年去度蜜月全程背着15寸的电脑，以便随时响应oncall，登机起飞前的几分钟还被拉了个钉钉会议。回来后干了两件事，第一就是申请把电脑换成13寸的，15寸真的重；第二开始刷题换工作。

字节跳动员工  

真想请假写一天代码，每天被oncall搞死

蚂蚁金服员工[3]  

14 

现在理解什么是干电池被用尽了吧，我出去旅游什么都可以不带，但是电脑一定得带，在景区的时候都在应急。钉钉一响，整个人就跟惊弓之鸟一样

Amazon员工  

oncall到怀疑人生

Google员工  

按小时计费!

Facebook员工 

要oncall，破事儿还贼多

百度员工  

晚上接一个oncall电话，起来处理十几分钟，然后就睡不着了，第二天一天都废了，每次都这样，感觉身体受不了了

微软员工  

4 

回复带带da师兄：就这样啊，微软没有运维，所有运维都是工程师自己弄的，很多组都有724oncall值班

Google员工[3]  

2 

有的组专门招sre来oncall，但近几年sre的预算卡的紧，很多新的service都是swe在oncall。不过一般oncall都没什么事，还给双倍工资。。所以大家都抢着oncall 😭

《Google SRE 运维解密》

软件工程思维

通过软件工程思维解决运维问题，将技术创新应用于日常运维。

关注研发工作

长期关注研发工作，将琐事占比控制在 50% 以下，保持团队创新力。

琐事过多的影响

- 团队生产力与创新力下降
- 员工士气低落，人才流失
- 系统可靠性与稳定性下降

如何消除琐事？

中断性工作（一般告警）

- 基于 SLO 配置告警
- 告警降噪，将根本原因一致的告警聚合为故障

OnCall（紧急告警）

- 实施值班轮换，同时负责一般告警的处理
- 一个月 OnCall 时间不超过 25%



Google OnCall 方法

文化

- 强调 OnCall 工作的平衡
- 对 OnCall 工程师进行补贴
- 鼓励事后总结与分享
- 无指责，对事不对人的氛围

机制

- 建立主备 OnCall 值班
- 及时对问题进行响应 (SLO)
- 建立清晰的问题升级路线
- 建立明确的故障处置步骤

工具 - Outalator

- 接收公司所有告警
- 将告警降噪为故障
- 对故障打标，数据分析
- 生成故障报告

The screenshot shows the Outalator web interface. At the top left is the 'OUTALATOR!' logo with a red superhero character. A navigation bar includes 'agoogler | Settings | About | Feedback | Sign out'. Below the logo, there are 'Teams/Alert queues:' with 'doodle-serving' and 'shakespeare'. A search bar contains the text 'Search e.g. has:bug tag:cause:human-error -summary:"global"'. The main content area is titled 'Tickets / Outages' and includes buttons for 'Refresh', 'Combine/Create Outage', 'Report Mode', 'Statistics', and 'Create Handoff Email'. A table lists several tickets with columns for 'Team, From', 'Summary', and 'Date'. The table contains five rows of data, with the last row indicating '5 Outalations'.

Team, From	Summary	Date
shakespeare (3), agoogler (7)	prober ShakespeareBlackboxProbe_SearchFailure bug:94043	2015-07-24 11:32:59 PDT
shakespeare, agoogler	frontend TaskFlapping bug:90210 Close	2015-07-22 13:15:09 PDT
shakespeare, jrandom	frontend ManyHttp500s cl:8675309 bug:89191	2015-07-22 04:19:44 PDT
shakespeare, agoogler (2)	storage AnnotationConsistencyTooEventual	2015-07-21 19:31:12 PDT
shakespeare, jrandom	frontend HighSearchLatency cause:alert-tuning bug:89109 action:silence	2015-07-20 03:35:43 PDT

5 Outalations

Flashduty , 你的Outalator !

降低运维负载

- 对告警进行降噪
- 建立值班机制
- 数据分析, 减少长期中断

提升 OnCall 效率

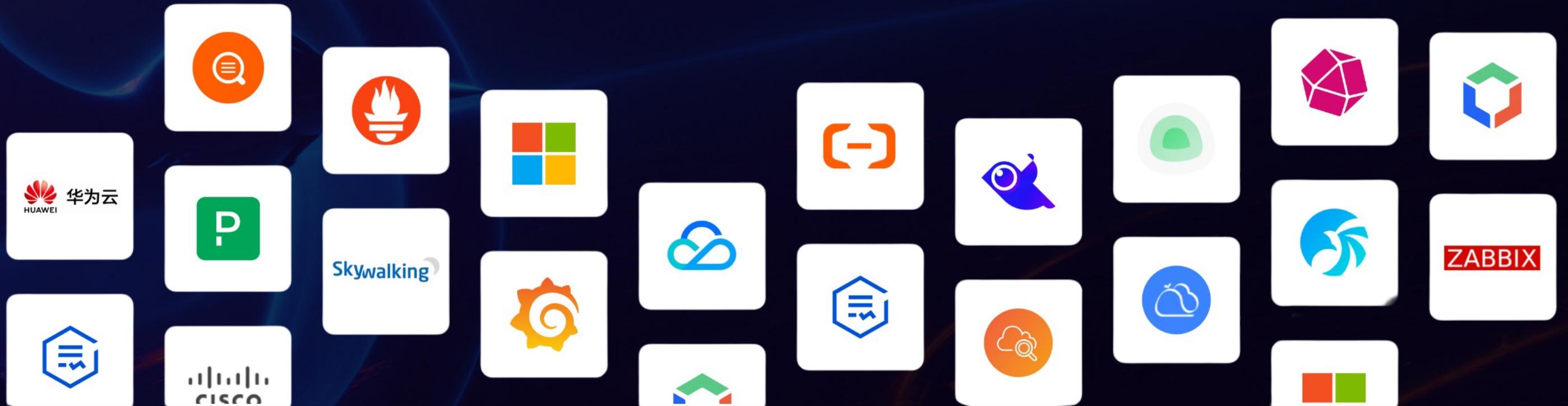
- 明确的故障升级路线
- 故障信息丰富
- IM协同增强
- 自定义操作扩展

The screenshot shows the Flashduty alert management interface. At the top, there are filters for '全部故障' (All Alerts), '未关闭' (Not Closed), and '最近 30 天' (Last 30 Days). Below the filters is a search bar for 'ID' and a '常用筛选' (Common Filter) dropdown. The main area displays a list of alerts, each with a checkbox, a status indicator (yellow for '待处理' - Pending, red for '处理中' - In Progress), an ID, a title, a duration, and a '关联告警' (Associated Alerts) count. The alerts are sorted by time, with the most recent at the top. The bottom of the interface shows a pagination bar with '总计 188 项' (Total 188 items) and a page number selector set to '1'.

ID	标题	状态	持续时间	关联告警	操作者	时间
#ABB3AB	出行系统->提现 / 连续飘红23138次	待处理	1小时53分	1	的空间	7月21日 15:46:52
#ABA8D4	sub-test / prom - dev-backup-01	待处理	2小时3分	1	n9e	7月21日 15:36:33
#23BA43	sub-test / prom - dev-n9e-01-vt220	待处理	2小时3分	1	n9e	7月21日 15:36:33
#AB5BD2	sub-test / prom - pushgw-agent	待处理	3小时20分	1	n9e	7月21日 14:19:33
#CE8B85	Cannot read properties of undefined (reading 'assigned_to')	处理中	3小时52分	1	guoyuhang	7月21日 13:48:09
#2346BE	Binlog同步延迟 / FlashcatSupport -	待处理	4小时33分	8	liufuniu	7月21日 13:07:02
#230226	Binlog同步延迟 / FlashcatSupport -	待处理	6小时7分	4	liufuniu	7月21日 11:33:01
#CDB9E2	flashcat-Fc服务宕机 / Flashcat-Self-Monitor - 10.32.247.241	待处理	7小时58分	1	liufuniu	7月21日 09:42:07
#CD8B78	CLS-多值展示到duty / cls_test	待处理	9小时15分	1	胡冲的空间	7月21日 08:25:03
#CD86CE	SystemDefault_acs_ecs_dashboard_vm.DiskUtilization / dev-flasheye-01,	待处理	9小时23分	1	测试环境	7月21日 08:16:44

集成，一个平台处理所有告警

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）



降噪，显著降低告警数量

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
- 至多降噪 99%

对频发告警进行收敛

- 避免狼来了效应
- 避免频繁被打断



值班，避免整个团队被中断

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

保持住流状态：如果你的目标只是做中断性工作，那么中断性工作就不再是中断。

中国北京 2024.7.26

值班管理 > 值班详情 如何配置

工作日周末区分 启用中

当前值班 现在 ~ 7月20日 23:59
主值班人 ysyneu
备值班人 聚财猫

2024年7月15日

最终值班

7月15日 周一	7月16日 周二
liufuniu	liufuniu
12:00 - 00:00	00:00 - 12:00

值班规则

7月15日 周一	7月16日 周二
liufuniu	liufuniu
12:00 - 00:00	00:00 - 12:00

临时调班

7月15日 周一	7月16日 周二
12:00 - 00:00	00:00 - 12:00

周末值班

开始时间: 7月15日 00:00 | 结束时间: 无结束时间

轮换周期: 1 | 周 | 公平轮换

交接时间: 周一 | 00:00

值班时间: 不限制 每天 每周

周六	00:00 - 23:59
周日	00:00 - 23:59

值班人员

- 组 A: ysyneu (主值班人), 聚财猫 (备值班人)
- 组 B: youcaimao (主值班人), zhanggen (备值班人)
- 组 C: qinyening (主值班人)

数据分析，驱动长期改进



运维负载

本周处理了多少告警？周告警数量处于什么趋势？



SLO指标

紧急告警的 MTTA 是多少？是否满足 SLO？



TopK 告警

哪些告警频繁产生？哪些主机频繁告警？



可操作性分析

团队对告警的响应比和降噪比如何？

中国北京 2024.7.26

分析数据

监控管理 故障管理 费用中心 访问控制 审计

过去一周降噪比例

98.65% ↑1.17%

过去一周故障 MTTA

31 分钟 ↓88.95%

过去一周故障 MTTR

2 小时 ↓80.47%

过去一周响应比例

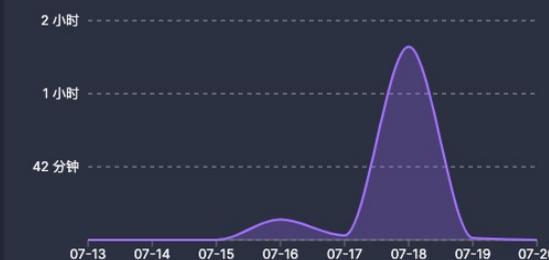
19.37% ↓13.83%

过去一周故障数量

444 条 ↑103.67%

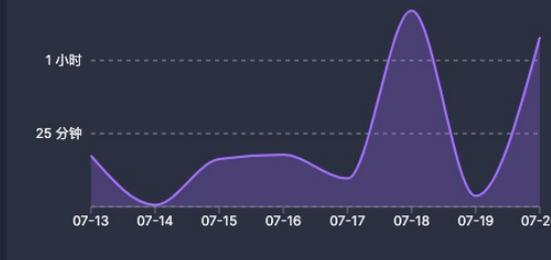
MTTA故障变化趋势-by协作空间

协作空间 严重程度 +1 最近7天



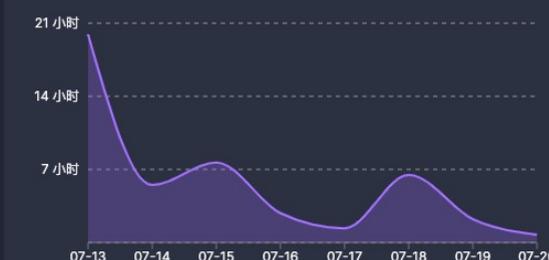
MTTA故障变化趋势-by团队

团队 严重程度 最近7天



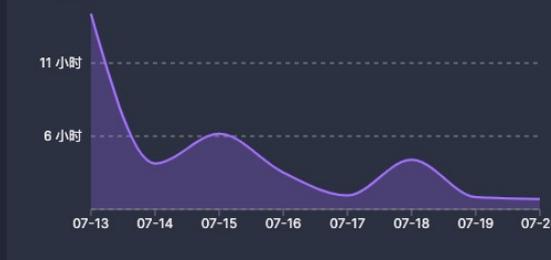
MTTR故障变化趋势-by协作空间

协作空间 严重程度 +1 最近7天



MTTR故障变化趋势-by团队

团队 严重程度 最近7天



Top20告警检查项

协作空间 最近7天

名称	数量
sub-test	29576
sub-test2	3507
Binlog同步延迟	1858
CLS-多值展示到duty	147
SaaS-HTTP请求出错	116

Top20告警对象

协作空间 最近7天

名称	数量
dev-n9e-01-vt220	8545
dev-flasheye-01	8267
dev-backup-01	7875
pushgw-agent	4942
localhost-192.168.10.114	3454

个人指标

成员 团队 成员 +10 严重程度 最近7天

姓名	被分派故障	认领故障	关闭故障	MTTA	MTTR
gu	12	12	8	1分42秒	4小时38分
du	1	0	0	0秒	0秒
yu	12	7	6	31秒	1分18秒
发	25	4	4	28分19秒	37分36秒
liu	106	3	0	5分18秒	0秒
qir	0	0	0	0秒	0秒

升级，正确时间通知正确的人

1

制定升级路线

为故障制定清晰的升级路线

2

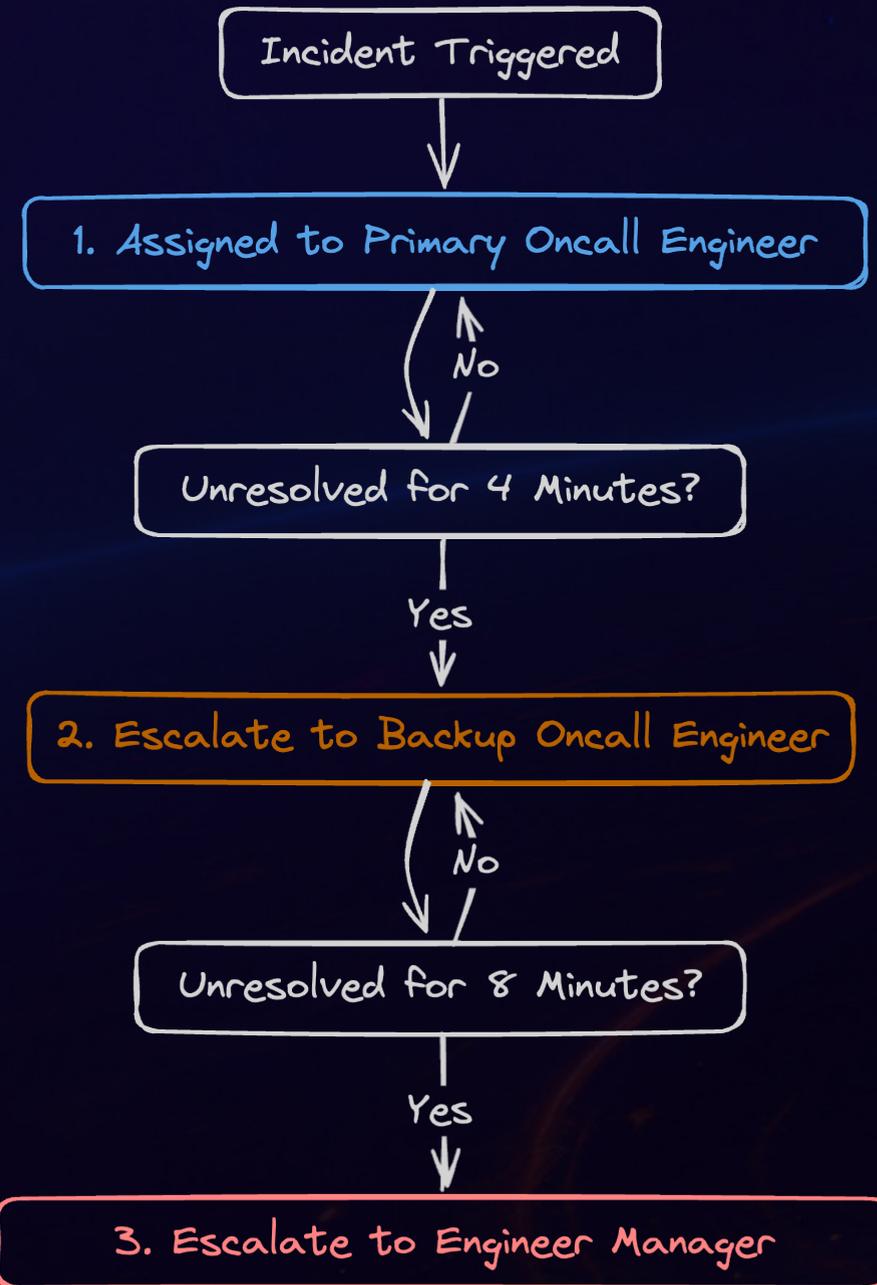
A：确保问题得到响应

不被漏处理导致更大资损

3

B：从容应对紧急情况

发生紧急情况不慌张，有后盾



数据增强，丰富故障上下文



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



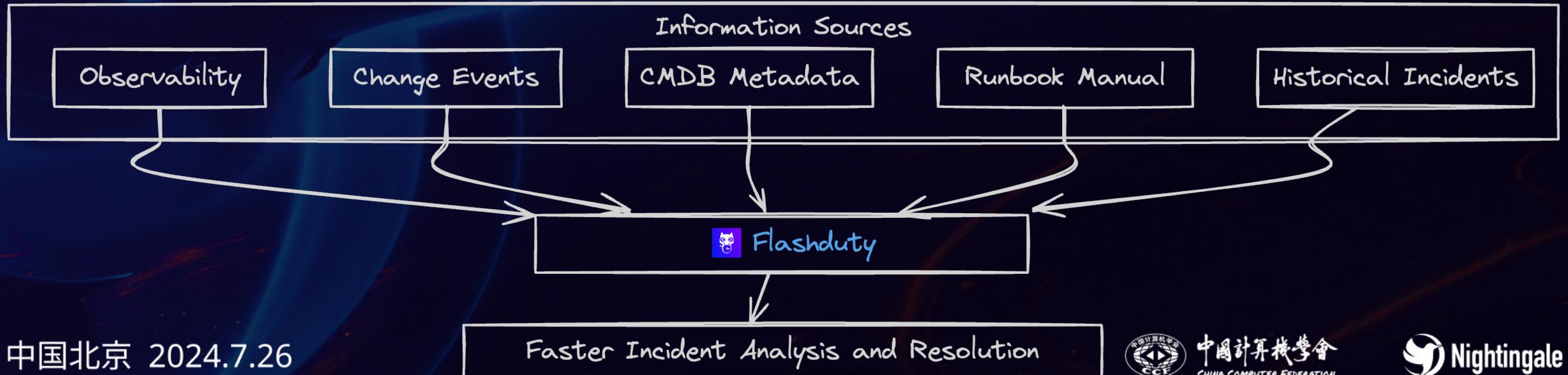
知识库和 SOP

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



IM集成，随时随地处理故障

1 实时通知

将故障的任何变化及时投递到 IM

2 多端操作

在 IM 内查看、处理故障，操作多端同步

3 加强协作

一键拉起作战室，关键信息回传到平台

中国北京 2024.7.26

The screenshot displays a WeChat group chat titled "回归测试群" (Regression Testing Group). A message from "FlashDuty Online 机器人" (FlashDuty Online Bot) is shown, indicating a critical alert: "【待处理】#CBD8B6 实时订单量越过阈值下界" (Pending Processing #CBD8B6 Real-time order volume exceeds threshold lower limit). The alert details include: "触发时间: 2024-07-20 22:27:46" (Trigger time: 2024-07-20 22:27:46), "严重程度: Critical" (Severity: Critical), "协作空间: 回归测试" (Collaboration space: Regression Testing), and "处理人员: @于双羽" (Handler: @Yushuangyu). The alert content specifies: "business: 电商" (Business: E-commerce), "dashboard: https://demo.flashcat.cloud/polaris/screen/detail?id=213667376571599" (Dashboard: https://demo.flashcat.cloud/polaris/screen/detail?id=213667376571599), "env: 线上" (Environment: Online), "source: 北极星" (Source: Polar Star), and "value: 1939单每分钟" (Value: 1939 orders per minute). At the bottom of the alert, there are buttons for "详情" (Details), "认领" (Claim), "关闭" (Close), and "自定义操作" (Custom Operation). The chat input area at the bottom shows "发送给 回归测试群" (Send to Regression Testing Group) and various formatting icons.

自定义操作，集成 workflows

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

中国北京 2024.7.26



FlashDuty Online 机器人 | 企业统一事件响应平台 19:22

【待处理】 #20169F TLS握手失败 / api.flashcat.cloud

🕒 触发时间: 2024-07-20 18:13:53

🔥 严重程度: Warning (已恢复, 持续1h8m)

👥 协作空间: Flashduty

👤 处理人员: @三月

resource: api.flashcat.cloud

metric: ListenerClientTLSNegotiationError

trigger_value: 17.14

详情

认领

关闭

自定义操作 ^

Reboot Server

发送给 快猫消防群

Aa 😊 @ ✂️ + ↗️ ➡️

感谢聆听

Thank you for listening