



GOPS2017  
Beijing



# GOPS

# 全球运维大会

2017



北京站

指导单位：



数据中心联盟  
Data Center Alliance



开放运维联盟  
OOPSA Open OPS Alliance

大会时间：7月28-29日

主办单位：



高效运维社区  
GreatOps Community



DevOps 时代



CIO 时代  
CIO & I ASPI

大会地点：北京朝阳悠唐皇冠假日酒店



# Qunar硬件自动化运维体系介绍

刘亮



# 自我介绍



- 姓名：刘亮
- 部门：Qunar opsdev
- 介绍：硬件运维开发



# 目录

- ➔ **1** 背景概述
- 2** 工作阐述
- 3** 具体介绍
- 4** 总结回顾

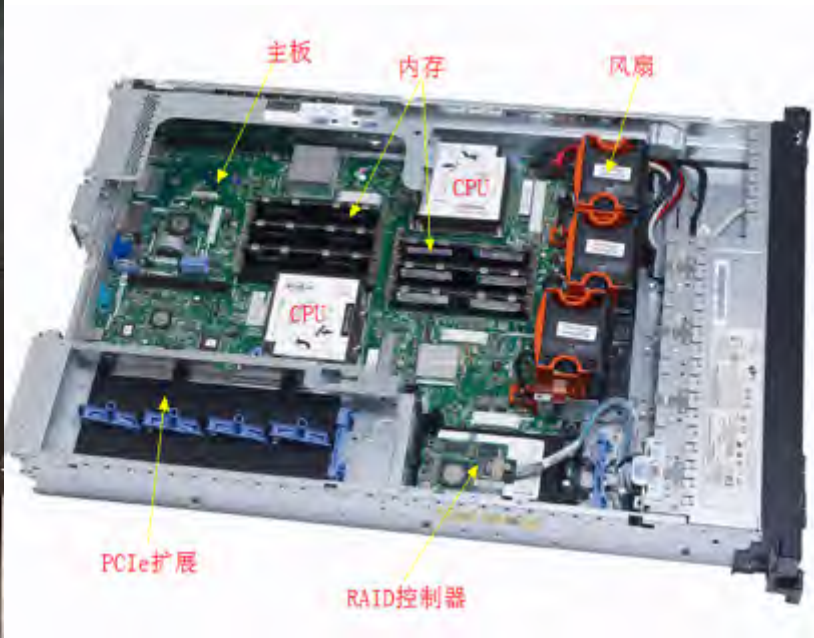


# 背景概述-硬件范畴



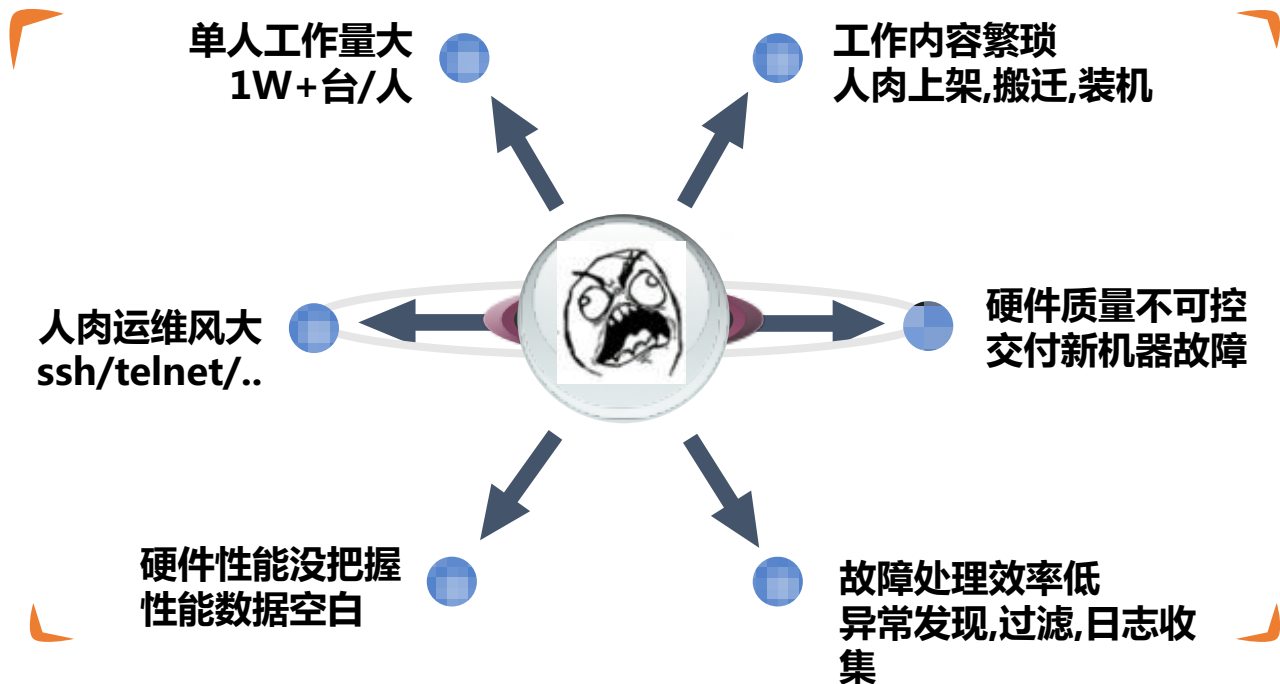


# 背景概述-硬件范畴

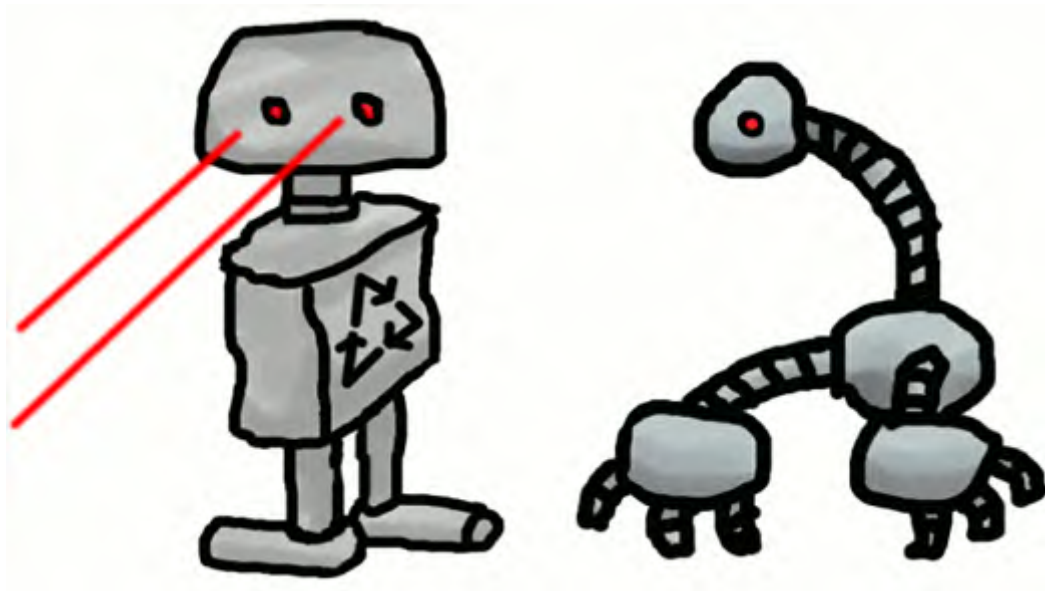




# 背景概述-痛点



# 背景概述-自动化与智能化必要性







# 背景概述-宗旨





# 目录

**1** 背景概述

➔ **2** 工作阐述

**3** 具体介绍

**4** 总结回顾



# 核心-硬件生命周期



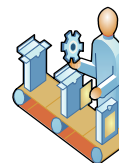
厂商供货



机器上架

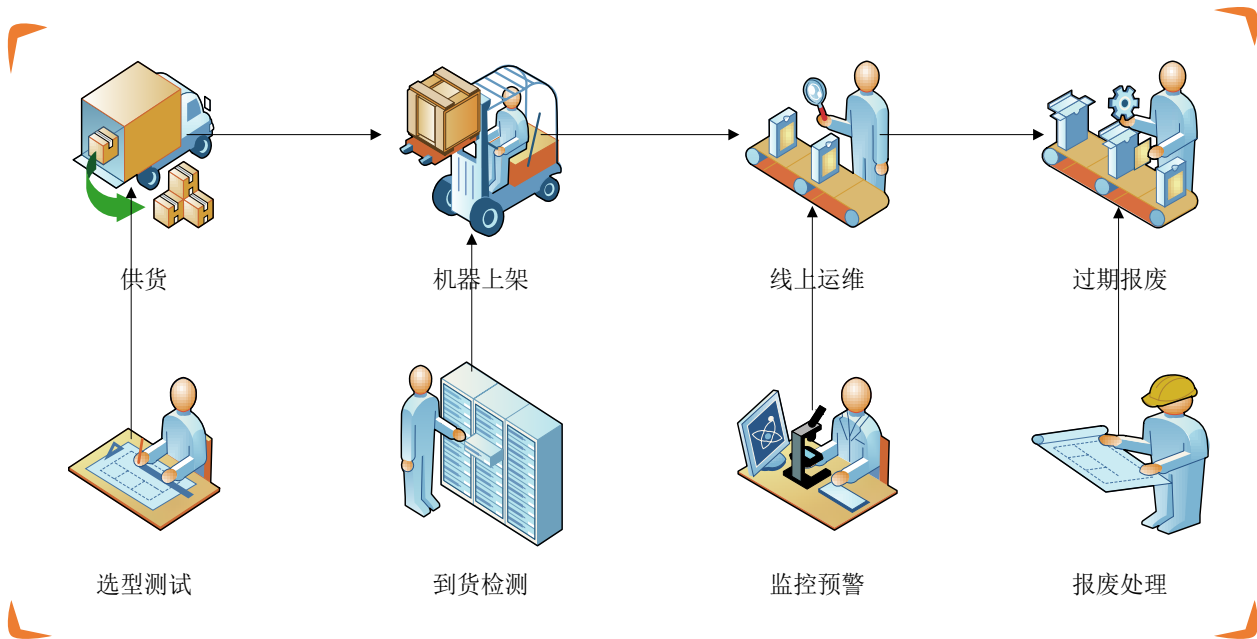


线上运维



报废处理

# 核心-硬件生命周期





# 目录

**1** 背景概述

**2** 工作阐述

**→ 3** 具体介绍

**4** 总结回顾

# 供应商数据有水分

供应商数

实际测试数据

据



50



5





# 实际测试看重性价比

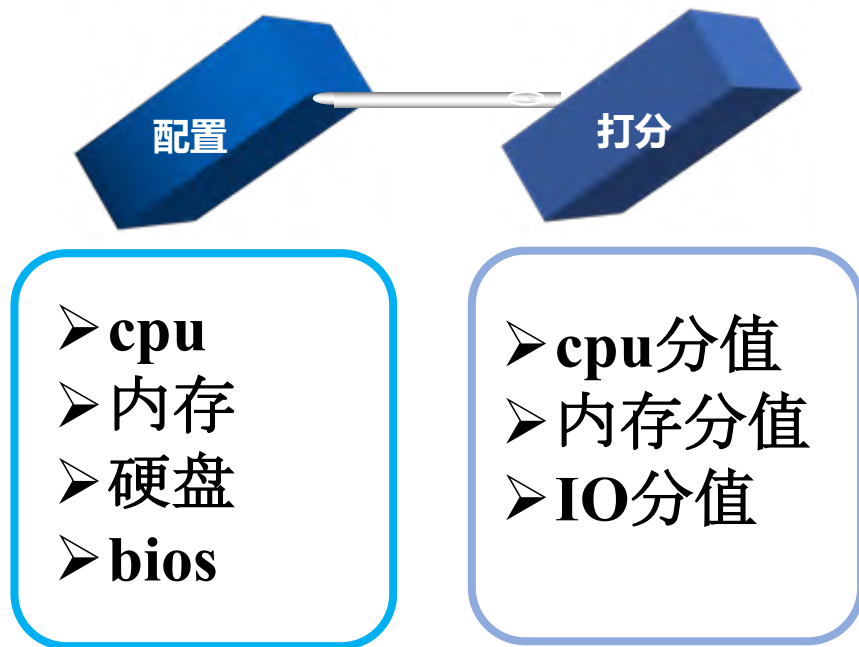


## □ 细节考虑

- ✓ 性能不是越高越好
- ✓ 最合适搭配
- ✓ bios设置是否正常
- ✓ raid卡设置
- ✓ 硬盘设置
- ✓ 系统设置



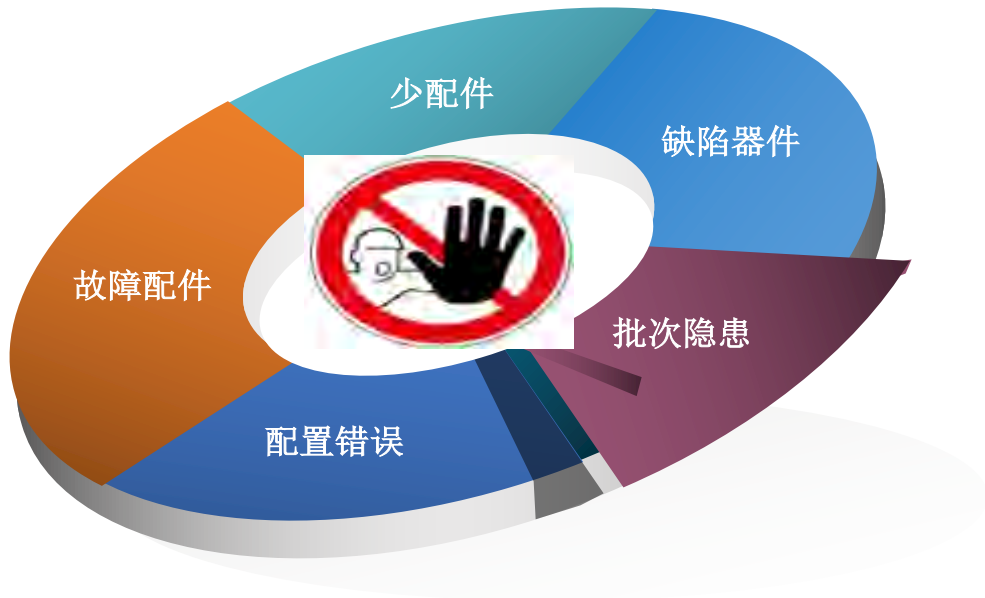
# 选型测试内容





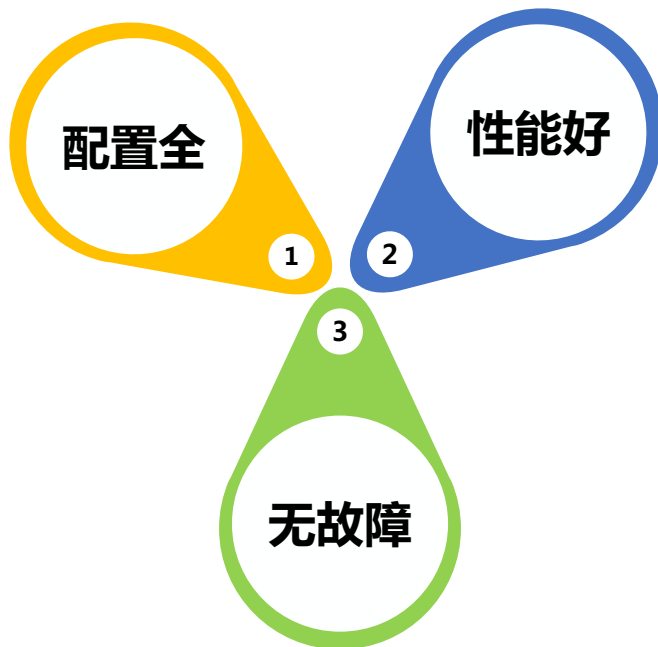


# 供货质量有缺陷





# 供货检测内容





# 数据收集展示需求杂乱

- 机器配置

cpu , 内存 , 硬盘



- 实时性能

cpu-idle , io-utils , load

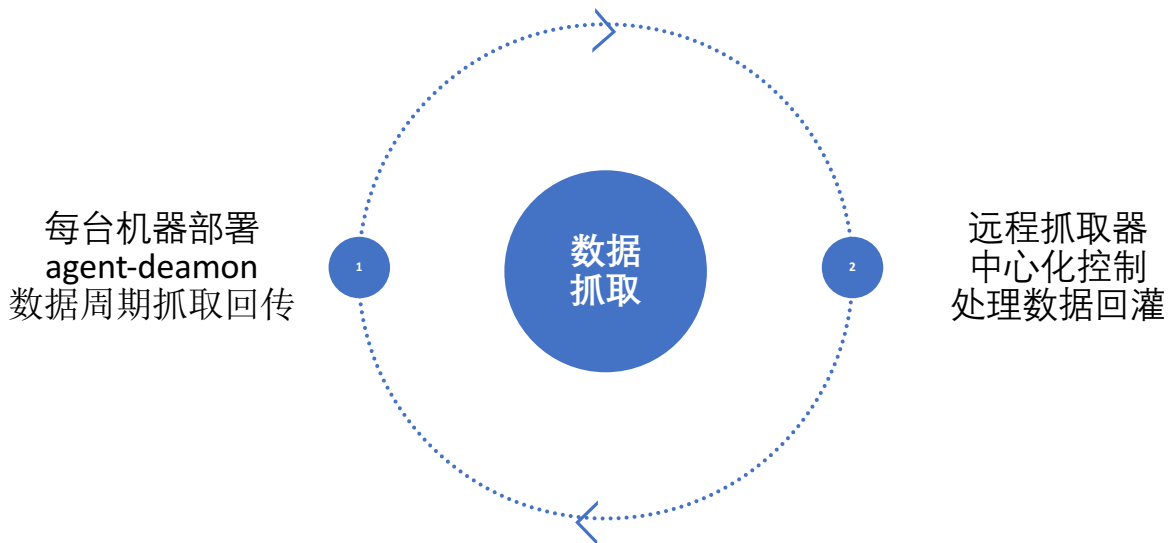


- 机房布局

机柜分布 , 机架布局



# 数据收集系统





# 故障发现困难 & 报修困难

累到吐血

报修效率低

人工筛  
查慢

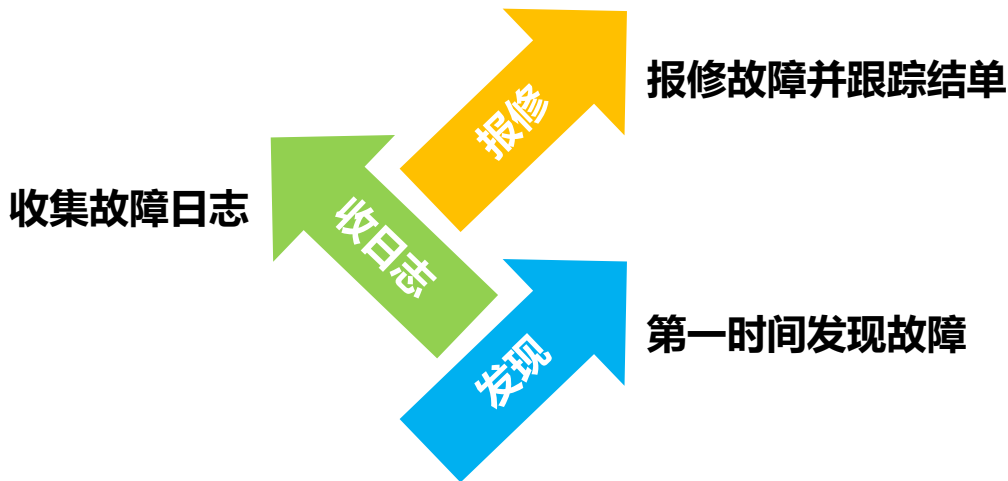
报修日志  
收集烦

售后跟踪  
累

故障分  
析难

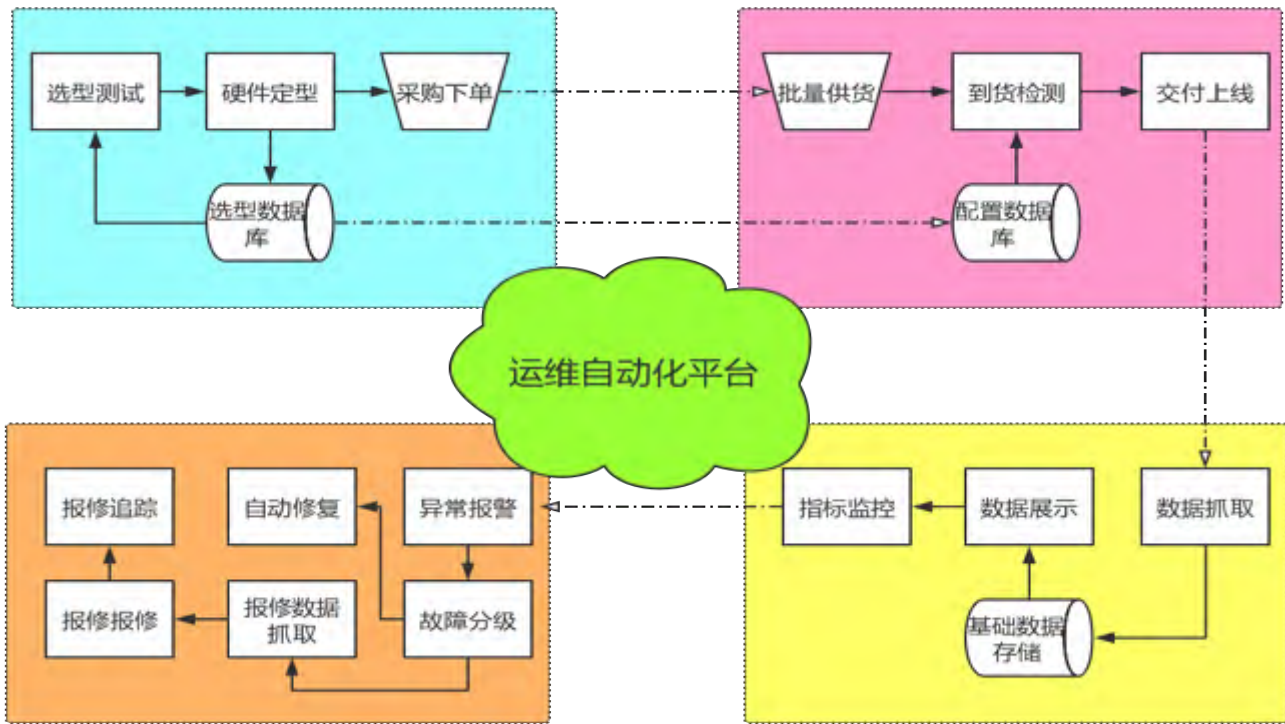


# 故障处理内容



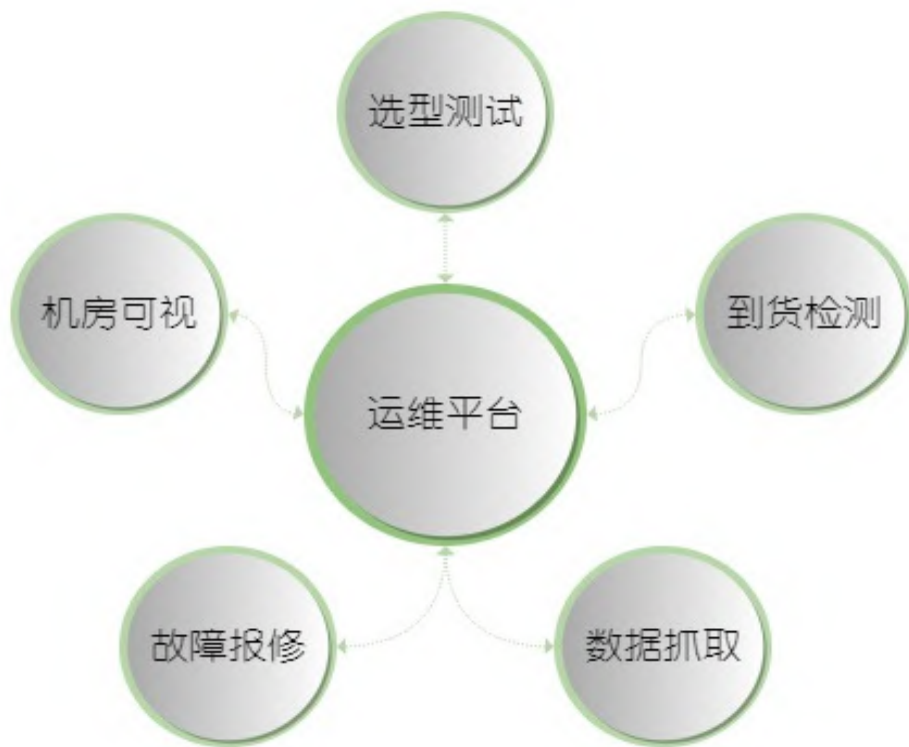


# 总体流程图





# 五大功能







# 复杂的数据源需求

硬件实时配置数据



机房机柜数据



配置定型数据

性能定型数据



系统性能时序指标



硬件故障数据



# 数据-功能

选型测试

到货检测

数据采集

故障报修

硬件运维平台

性能数据

配置数据

时序指标

故障数据



# opsdb-硬件配置数据

<a href="#">重建</a>	<a href="#">刷新</a>	<a href="#">回收通知</a>	<a href="#">编辑</a>	<a href="#">下架</a>	<a href="#">删除</a>
设备状态	active				
类型	server				
用途	prod				
操作系统	CentOS 6.7				
主机IP	██████████				
所在机房	cn1				
放置机架	2392				
机架中位置(U位)	24				
序列号	██████████				
设备型号	HP - ProLiant DL360 Gen9				
磁盘数量					
RAID级别					
硬件抓取ssd信息	品牌:SHANNON 容量:6400.00 GB 序列号:SF15121K755042				



# Watcher-基础监控时序指标





# 生态圈





# 为啥要硬件配置格式统一

## □ 同一套搭配

- ✓ 不同服务器供货商
- ✓ 不同配件供货商
- ✓ 不同固件型号
- ✓ 不同批次
- ✓ 录入错误
- ✓ 数据缺失

## □ 同一套搭配

- ✓ 相同的性能
- ✓ 相同供货配置



# 格式化规则 *Rules*



## □ 配置格式统一

- ✓ 每台部署定制抓取工具
- ✓ 规则中心化管理
- ✓ 工具远程同步规则

## □ 格式化规则集

- ✓ 调用工具命令
- ✓ 输出格式化切分
- ✓ 规范输出格式

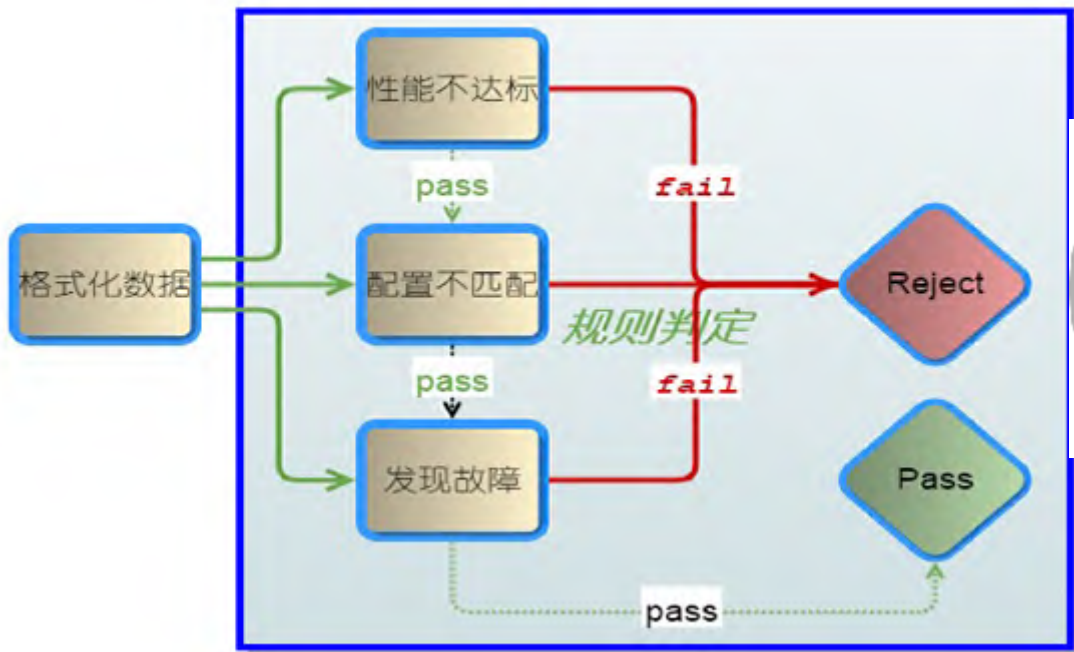


# 规则示例

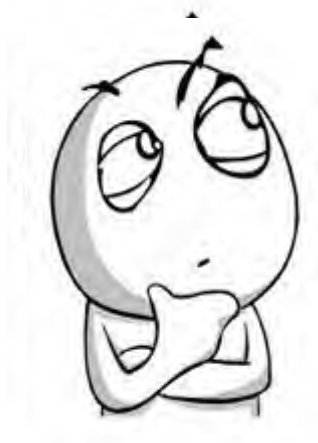
```
"cpu": {  
  "format": {  
    "corenums": "Core Count# #1",  
    "family": "Family# #1",  
    "freq": "Current Speed# #1",  
    "model": "Version# #1",  
    "sectionkey": "Processor Information",  
    "vendor": "Manufacturer# #1"  
  },  
  "runcmd": {  
    "cmdline": "dmidecode -t processor",  
    "tool": "dmidecode"  
  },  
  "transfer": {  
    "model": "e5-2630v4~e5-2630v4|e5-2630 v4|e5 2630v4|e5 2630 v4|e52630v4; e5-2620v4~e5-2620v4|e5-2620 v4|e5 2620v4|e5 2620 v4|e52620v4; e5-2660v3~e5-2660v3|e5-2660 v3|e5 2660v3|e5 2660 v3|e52660v3; e5-2650v4~e5-2650v4|e5-2650 v4|e5 2650v4|e5 2650 v4|e52650v4; e5-2650v3~e5-2650v3|e5-2650 v3|e5 2650v3|e5 2650 v3|e52650v3; e5-2640v2~e5-2640v2|e5-2640 v2|e5 2640v2|e5 2640 v2|e52640v2; e5-2630v3~e5-2630v3|e5-2630 v3|e5 2630v3|e5 2630 v3|e52630v3; e5-2620v3~e5-2620v3|e5-2620 v3|e5 2620v3|e5 2620 v3|e52620v3; e5-2420v3~e5-2420v3|e5-2420 v3|e5 2420v3|e5 2420 v3|e52420v3; e5-2450v3~e5-2450v3|e5-2450 v3|e5 2450v3|e5 2450 v3|e52450v3; e5-2637v3~e5-2637v3|e5-2637 v3|e5 2637v3|e5 2637 v3|e52637v3; e5-2660v2~e5-2660v2|e5-2660 v2|e5 2660v2|e5 2660 v2|e52660v2; e5-2650v2~e5-2650v2|e5-2650 v2|e5 2650v2|e5 2650 v2|e52650v2; e5-2630v2~e5-2630v2|e5-2630 v2|e5 2630v2|e5 2630 v2|e52630v2; e5-2620v2~e5-2620v2|e5-2620 v2|e5 2620v2|e5 2620 v2|e52620v2; e5-2450v2~e5-2450v2|e5-2450 v2|e5 2450v2|e5 2450 v2|e52450v2; e5-2430v2~e5-2430v2|e5-2430 v2|e5 2430v2|e5 2430 v2|e52430v2; e5-2420v2~e5-2420v2|e5-2420 v2|e5 2420v2|e5 2420 v2|e52420v2; e5-2637v2~e5-2637v2|e5-2637 v2|e5 2637v2|e5 2637 v2|e52637v2; e5-2660~e5-2660|e5 2660; e5-2650~e5-2650|e5 2650; e5-2630~e5-2630|e5 2630; e5-2620~e5-2620|e5 2620; e5-2450~e5-2450|e5 2450; e5-2430~e5-2430|e5 2430; e5-2420~e5-2420|e5 2420; e5410~e5410; e5420~e5420; e5440~e5440; e5520~e5520; e5604~e5504; e5620~e5620; e5645~e5645; e5649~e5649; x5650~x5650; e5-2609~e5-2609; e3-1240v3~e3-1240 v3; e5-2603v2~e5-2603v2|e5-2603 v2|e5 2603v2|e5 2603 v2|e52603v2; e5-2403~e5-
```



# 标准



# 检测标准



## □ 配置匹配

- ✓ 选型测试录入定型配置
- ✓ 定型配置生成配置标准
- ✓ 到货批量抓取实际配置
- ✓ 实际配置?=定型配置

## □ 性能达标

- ✓ 选型测试录入性能标准
- ✓ 到货批量抓取实际性能
- ✓ 实际性能?<性能标准

# 标准示例



CPU配置	浮点标准	整型标准	机器数量
E5-2620V3#2.40GHZ#6:2:ENABLE#ENABLE	36000#0.1	24000#0.1	770
E5-2660V3#2.60GHZ#10:2:ENABLE#ENABLE	64000#0.1	42000#0.1	10
E5645#2.40GHZ#6:2:ENABLE#ENABLE	10000#0.1	10000#0.1	34
E5-2630V2#2.60GHZ#6:2:ENABLE#ENABLE	32000#0.1	24000#0.1	1639
X5650#2.67GHZ#6:2:ENABLE#ENABLE	27000#0.1	20000#0.1	5
E5-2620V4#2.10GHZ#8:2:ENABLE#ENABLE	46000#0.1	30000#0.1	581
E5-2660V2#2.20GHZ#10:2:ENABLE#ENABLE	46000#0.1	32000#0.1	1
E5-2630V3#2.40GHZ#8:1:ENABLE#ENABLE	25000#0.1	18000#0.1	31
E5620#2.40GHZ#4:2:ENABLE#ENABLE	10000#0.1	10000#0.1	28
E5-2430#2.20GHZ#6:2:ENABLE#ENABLE	10000#0.1	10000#0.1	6

# 垃圾报警多



1	fan#high_speed	2065
2	raidcard#cache_battery_charging	1198
3	other#many_hpaucLi	991
4	raidcard#cache_battery_capacity_unknown	830
5	other#socket_timeout	699
6	other#hpasm_timeout	364
7	mboard#mboard_battery_unknown	280
8	other#connect_timeout	152
9	other#hpasmd_restart	126
10	raidcard#ld_ready	118
11	voltage#ps_not_available	108
12	raidcard#cache_battery_ready	94
13	other#connect_timeout2	87
14	raidcard#ctrl_accelerator_charging	80
15	raidcard#ctrl_accelerator_attention	63
16	disk#failed2	60
17	other#insufficient_rights	57
18	raidcard#fw_out_of_data	53
19	other#name_service_unknown	50
20	other#connect_timeout3	50
21	raidcard#degrade	48
22	disk#not_certified	44
23	voltage#voltage_not_found	39



# 定制监控工具



降90%报警

原有的报警工具发出大量的报警只用少量的是真正的硬件故障需要报修



精准定位故障

分为ok,warning,critical三级，critical需及时报修，warning需要观察分析，ok忽略。

# 人肉报修慢





# 故障报修要点

收集报修日志  
故障配置信息



获取报修邮箱  
发生格式化邮件



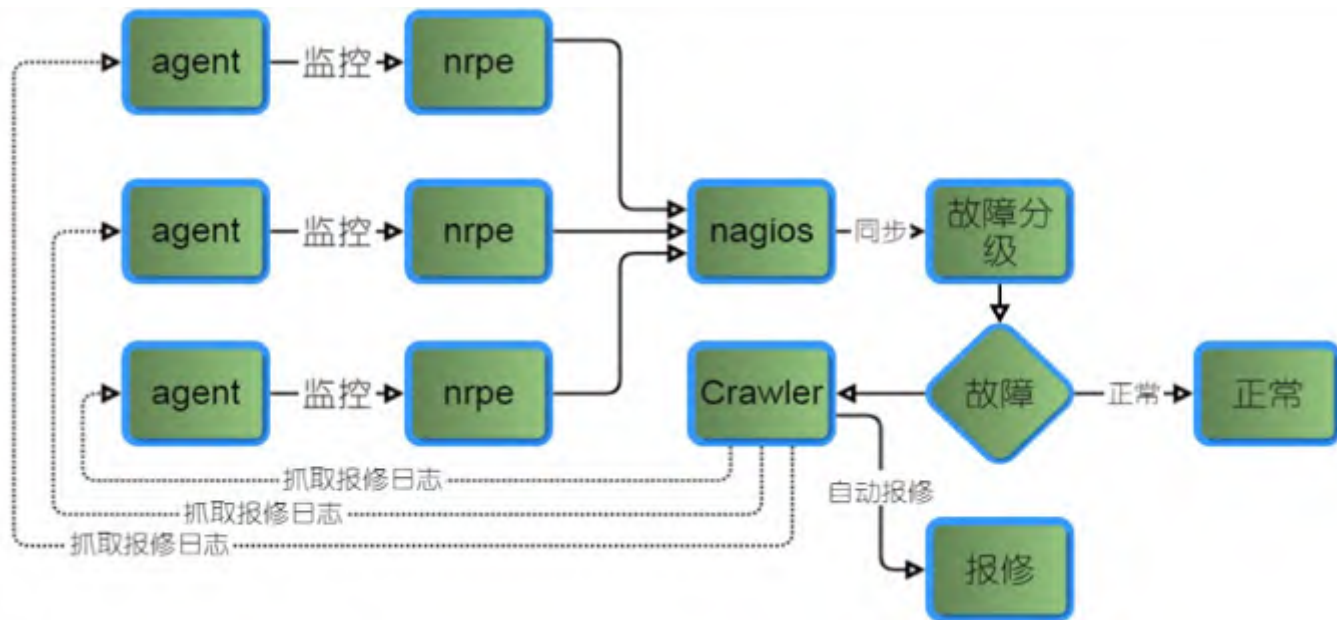
报修  
要点



跟踪故障维修进度  
及时故障结单



# 自动化报修





# 机房运维杂事多

## □ 上架布局

- ✓ 机柜空闲情况
- ✓ 机房布局情况

## □ 机房巡检

- ✓ 机柜温度
- ✓ 机房电流





# 可视化平台





# 目录

**1** 背景概述

**2** 工作阐述

**3** 具体介绍

**➔ 4** 总结回顾

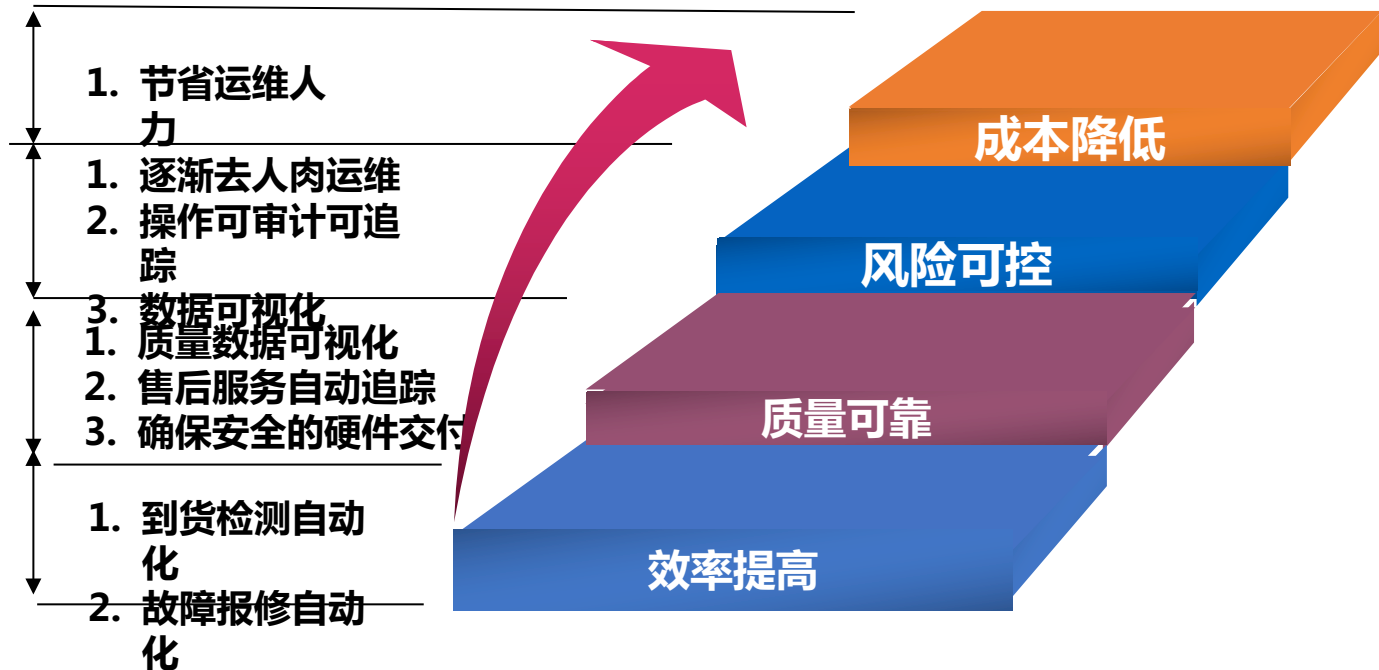


# Before VS. Now

Before	Now
<ul style="list-style-type: none"><li>● 选型靠读文档</li><li>● 到货人工核查</li><li>● 故障人肉报修</li><li>● 配置变更跑命令</li><li>● 故障率分析空白</li></ul>	<ul style="list-style-type: none"><li>● 选型性能自动采集分析对比报告</li><li>● 到货一体化检测</li><li>● 故障自动报修追踪</li><li>● 硬件配置一键变更</li><li>● 故障率分析发现批次问题</li></ul>



# 实施效果





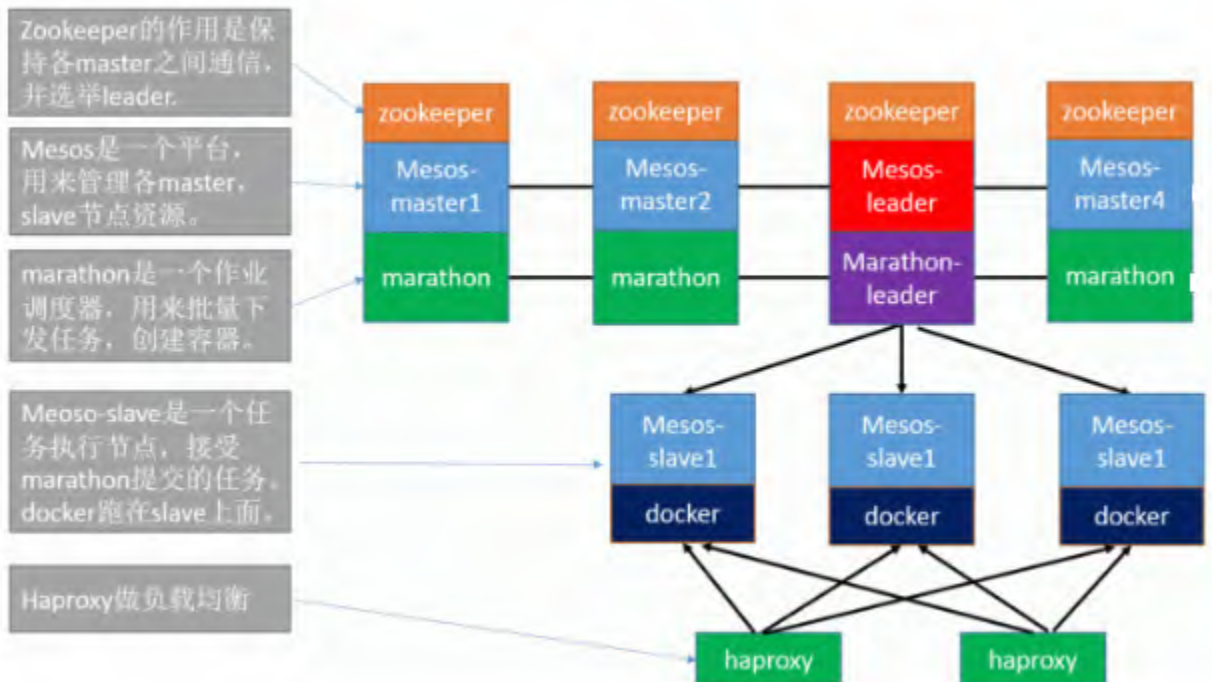
# 数据抓取实施难点



- 弹性线性扩展
  - ✓ 多目标抓取docker实例
- 自动HA
  - ✓ Marathon+Mesos
- 异步任务调度
  - ✓ Celery拆分独立模块

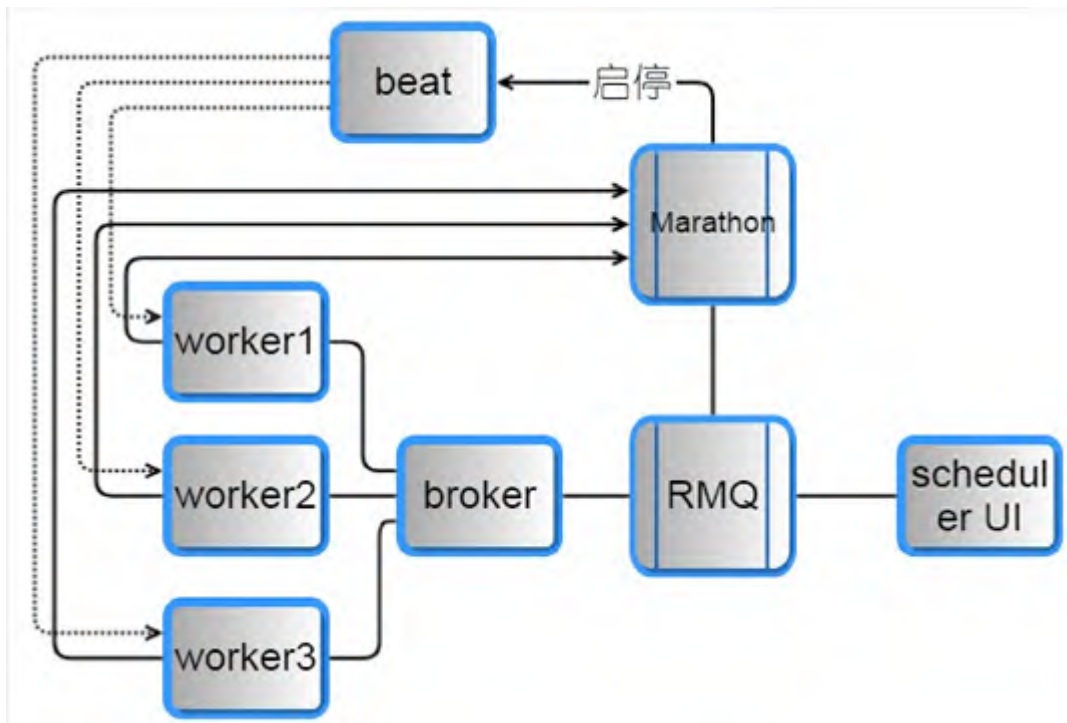


# Mesos+Marathon+Docker





# 分布任务调度集群







高效运维社区  
GreatOPS Community



## 会议

## 培训

## 咨询

- 8月18日 DevOpsDays 上海
- 全年 DevOps China 巡回沙龙
- 11月17日 DevOps金融上海

- EXIN DevOps Master 认证培训
- DevOps 企业内训
- DevOps 公开课
- 互联网运维培训
- 企业DevOps 实践咨询
- 企业运维咨询



商务经理：刘静女士  
电话 / 微信：13021082989  
邮箱：liujing@greatops.com



# Thanks

高效运维社区  
开放运维联盟

荣誉出品