

# 去哪儿网 自动化运维最佳实践

徐磊@qunar-opsdev

# 大纲

- 运维与日志
- 日志的收集与处理
- 日志服务化
- 开发参与运维

# 运维与日志

- 日志是什么？
  - 记录系统/应用行为
  - 遵循某种表达式规则
- 日志的价值？
  - 包含系统/应用做过的事情
  - 经过分析后才能体现价值

# 运维与日志



日志是运维的事件源之一！

# 日志的收集与处理

- 系统日志与应用日志分离
  - rsyslog（规则固定/变化几乎没有）
  - qunar-flume（规则变动大/处理多种多样）
- 建立机器与应用对应关系
  - 应用中心
- 智能收集Agent
  - 主动发现日志文件
  - 配置下发/热加载
  - 自动部署

# 日志的收集与处理

应用中心 » 应用列表 » 应用详情 »

应用详情

服务器列表

URL访问限制

DUBBO访问限制

会话服务

数据库资源授权状态












日志收集列表

日志收集黑名单列表

操作记录

主机列表

添加

操作	环境	主机	端口	监控状态	报警状态	版本	创建时间	agent版本	agent心跳时间	agent分析目录	http引流
	prod	████████████████████	8095	<input type="radio"/> OFF	<input type="radio"/> OFF	8.1.15	2015-09-21 17:09:22	未安装	未安装	<input type="text"/> <a href="#">更新</a>	<input type="radio"/> OFF
	prod	████████████████████.com	8095	<input type="radio"/> OFF	<input checked="" type="radio"/> ON	8.1.15	2015-12-10 20:34:36	1.6.9-SNAPSHOT	2016-01-03 14:31:54	<input type="text"/> <a href="#">更新</a>	<input type="radio"/> OFF
	prod	████████████████████	8095	<input type="radio"/> OFF	<input checked="" type="radio"/> ON	8.1.15	2015-12-10 20:35:01	1.6.9-SNAPSHOT	2016-01-03 14:32:20	<input type="text"/> <a href="#">更新</a>	<input type="radio"/> OFF
	prod	████████████████████.com	8095	<input type="radio"/> OFF	<input checked="" type="radio"/> ON	8.1.15	2015-12-10 20:35:29	1.6.9-SNAPSHOT	2016-01-03 14:31:54	<input type="text"/> <a href="#">更新</a>	<input type="radio"/> OFF
	prod	████████████████████	8095	<input type="radio"/> OFF	<input checked="" type="radio"/> ON	8.1.15	2015-12-10 20:35:50	1.6.9-SNAPSHOT	2016-01-03 14:32:32	<input type="text"/> <a href="#">更新</a>	<input type="radio"/> OFF
	prod	████████████████████	8095	<input type="radio"/> OFF	<input checked="" type="radio"/> ON	8.1.15	2015-12-10 20:36:22	1.6.9-SNAPSHOT	2016-01-03 14:31:49	<input type="text"/> <a href="#">更新</a>	<input type="radio"/> OFF
	prod	████████████████████.com	8095	<input type="radio"/> OFF	<input checked="" type="radio"/> ON	8.1.15	2015-12-10 20:36:45	1.6.9-SNAPSHOT	2016-01-03 14:32:11	<input type="text"/> <a href="#">更新</a>	<input type="radio"/> OFF
	prod	████████████████████	8095	<input type="radio"/> OFF	<input checked="" type="radio"/> ON	8.1.15	2015-12-10 20:37:39	1.6.9-SNAPSHOT	2016-01-03 14:31:59	<input type="text"/> <a href="#">更新</a>	<input type="radio"/> OFF
	prod	████████████████████	8095	<input type="radio"/> OFF	<input checked="" type="radio"/> ON	8.1.15	2015-12-10 20:37:18	1.6.9-SNAPSHOT	2016-01-03 14:32:27	<input type="text"/> <a href="#">更新</a>	<input type="radio"/> OFF
	prod	████████████████████.com	8095	<input type="radio"/> OFF	<input type="radio"/> OFF	8.1.15	2015-08-28 15:13:22	1.6.9-SNAPSHOT	2016-01-03 14:28:53	/home/q/www/web_hotdo <a href="#">更新</a>	<input type="radio"/> OFF
	prod	████████████████████.com	8095	<input type="radio"/> OFF	<input checked="" type="radio"/> ON	8.1.15	2015-08-28 15:13:23	1.6.9-SNAPSHOT	2016-01-03 14:29:41	/home/q/www/web_hotdo <a href="#">更新</a>	<input type="radio"/> OFF
	prod	████████████████████.com	8095	<input type="radio"/> OFF	<input type="radio"/> OFF	8.1.15	2015-08-28 15:14:13	1.6.9-SNAPSHOT	2016-01-03 14:29:14	/home/q/www/web_hotdo <a href="#">更新</a>	<input type="radio"/> OFF
	prod	████████████████████.com	8095	<input type="radio"/> OFF	<input type="radio"/> OFF	8.1.15	2015-08-28 15:15:51	1.6.9-SNAPSHOT	2016-01-03 14:29:29	/home/q/www/web_hotdo <a href="#">更新</a>	<input type="radio"/> OFF
	prod	ldclient13.wan.cn1.guany.com	8095	<input type="radio"/> OFF	<input checked="" type="radio"/> ON	8.1.15	2015-08-28 15:15:30	1.6.9-SNAPSHOT	2016-01-03 14:29:14	<input type="text"/> <a href="#">更新</a>	<input type="radio"/> OFF

















































# 日志的收集与处理

应用中心 > 应用列表 > 应用详情 

应用详情 服务器列表 URL访问限制 DUBBO访问限制 会话服务 数据库资源授权状态 日志收集列表 日志收集黑名单列表 操作记录

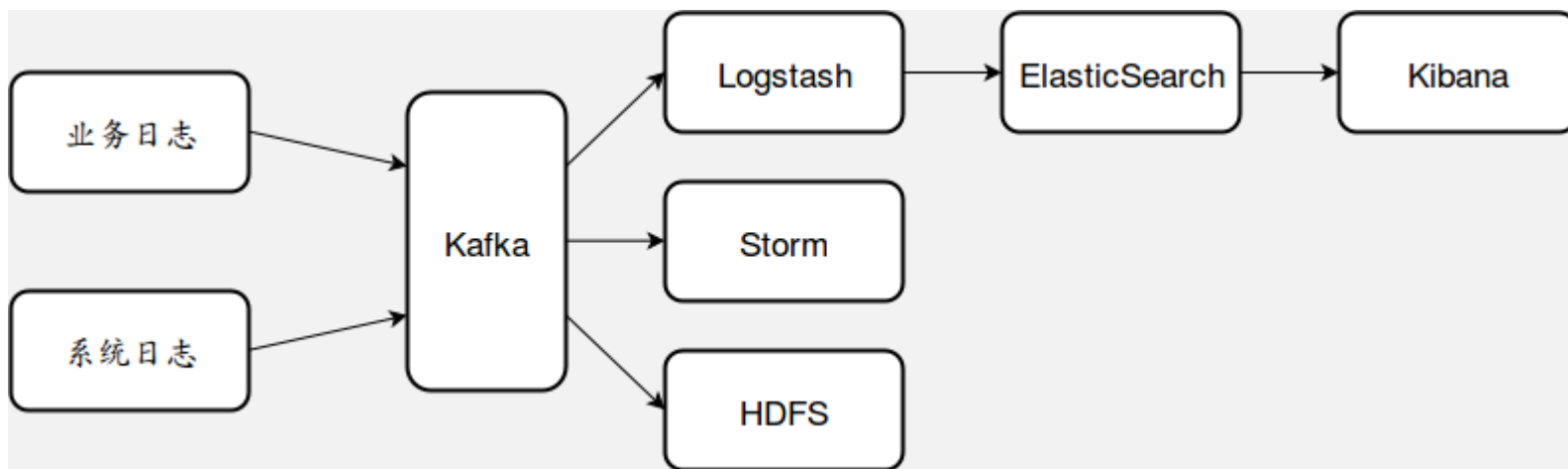
日志收集列表

打开

操作	名称	模式	前缀	主题	目录	收集状态	可靠级别	顺序消费	更新时间
 	未设置	access.{yyyy-MM-dd}.log.gz	按行收集	logs.logger.flight	/home/q/www/web_hotdog/logs		完全可靠		2015-11-25 14:55:06
 	androidError.log	androidError.log.{yyyy-MM-dd}	\d{2}\d{2}\d{2}:\d{2}\d{2}	logs.logger.flight	/home/q/www/web_hotdog/logs		完全可靠		2015-12-17 09:45:51
 	androidfc.txt	androidfc.txt.{yyyy-MM-dd-HH}.gz	\d{2}\d{2}\d{2}:\d{2}\d{2}	logs.logger.flight	/home/q/www/web_hotdog/logs		完全可靠		2015-11-25 14:55:06
 	bat.log	未设置	\d{2}\d{2}\d{2}:\d{2}\d{2}	logs.logger.flight	/home/q/www/web_hotdog/logs		完全可靠		2015-11-25 14:55:06
 	未设置	bat.log.{yyyy-MM-dd}.log.gz	\d{2}\d{2}\d{2}:\d{2}\d{2}	logs.logger.flight	/home/q/www/web_hotdog/logs		完全可靠		2015-11-25 14:55:06
 	BusinessSrv.txt	BusinessSrv.txt.{yyyy-MM-dd-HH}.gz	\d{2}\d{2}\d{2}:\d{2}\d{2}	logs.logger.flight	/home/q/www/web_hotdog/logs		完全可靠		2015-11-25 14:55:06
 	catalina.out	catalina.out.{yyyy-MM-dd}.gz	\d{4}-\d{2}-\d{2}	logs.logger.flight	/home/q/www/web_hotdog/logs		完全可靠		2015-11-25 14:55:06
 	cooper.log	cooper.log.{yyyy-MM-dd-HH}.gz	\d{4}-\d{2}-\d{2}	logs.logger.flight	/home/q/www/web_hotdog/logs		完全可靠		2015-11-25 14:55:06
 	未设置	dubbo-access-consumer.{yyyy-MM-dd}.log.gz	\d{4}-\d{2}-\d{2}	logs.logger.flight	/home/q/www/web_hotdog/logs		完全可靠		2015-11-25 14:55:06
 	error.log	error.log.{yyyy-MM-dd-HH}.gz	\d{4}-\d{2}-\d{2}	logs.logger.flight	/home/q/www/web_hotdog/logs		完全可靠		2015-11-25 14:55:06
 	gc.log	gc.log.{yyyy-MM-dd}.gz	\d{4}-\d{2}-\d{2}	logs.logger.flight	/home/q/www/web_hotdog/logs		完全可靠		2015-11-25 14:55:06
 	helper.log	helper.log.{yyyy-MM-dd-HH}.gz	\d{2}\d{2}\d{2}:\d{2}\d{2}	logs.logger.flight	/home/q/www/web_hotdog/logs		完全可靠		2015-11-25 14:55:06

# 日志的收集与处理

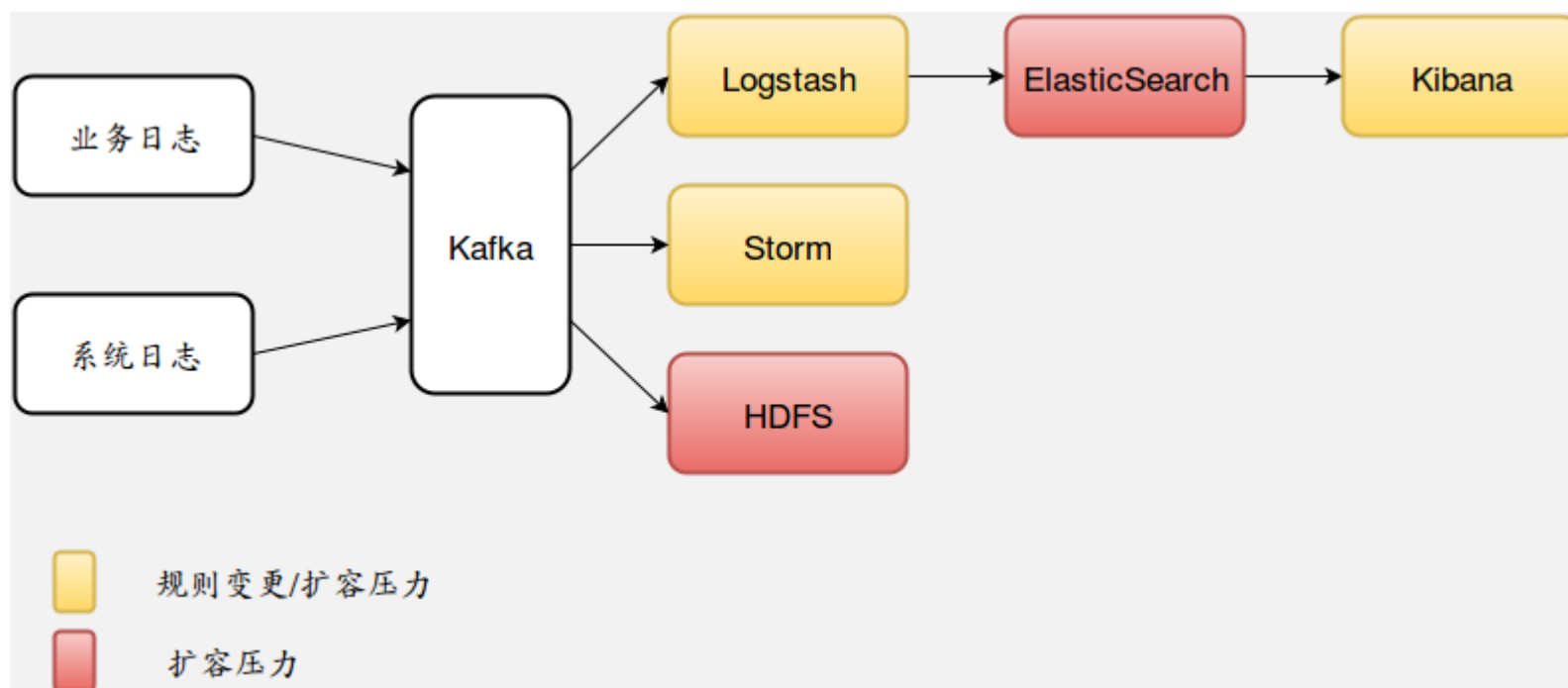
- 流行的Kafka + ELK架构
  - 利用**Kafka**的高吞吐量做为日志**Buffer**
  - ELK/Spark/Storm**消费**Kafka**消息进行分析
  - HDFS**归档日志
- 全开源架构



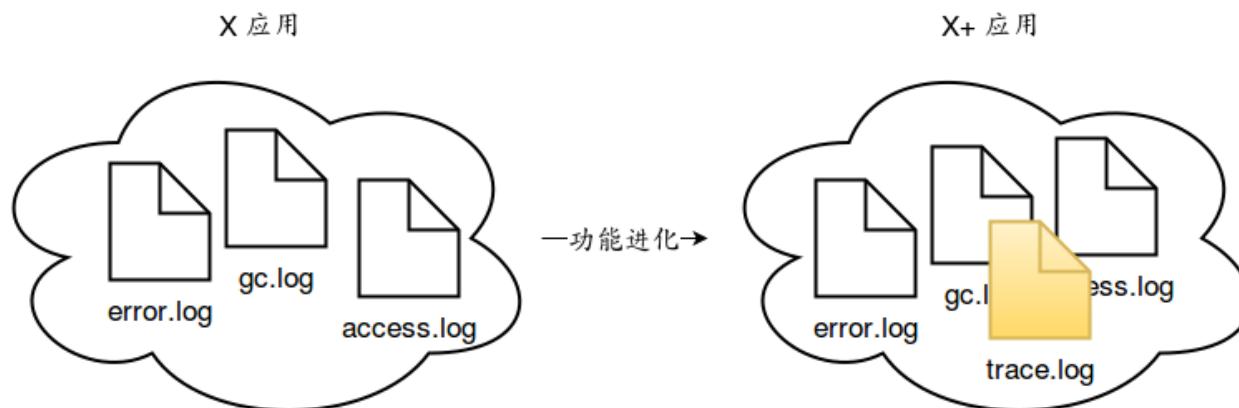


# 日志的收集与处理

- 应用日志变化频繁



# 日志的收集与处理



```
ruby {
  code => "event['log_duration'] = (Time.parse(event['@timestamp']).to_s).to_f * 1000).to_i
}

mutate {
  add_field => { "host" => "%{source_host}" }
}

ruby {
  code => "event['log_duration'] = (Time.parse(event['@timestamp']).to_s).to_f * 1000).to_i
}

grok {
  match => {
    "source_path" => "%{GREEDYDATA}/%{WORD:log_name}..."
  }
}

if "access" in [log_name] {
  grok {
    match => {
      "content" => [
        "%{IPV4:src_ip} - - [%{HTTPDATE:logtime}] %{GREEDYDATA} %{INT:http_status:int}"
        "%{IPV4:src_ip} - - [%{HTTPDATE:logtime}] %{GREEDYDATA} %{INT:http_status:int}"
      ]
    }
  }
  date {
    # example 17/Jun/2015:07:59:58 +0800
    match => { "logtime", "dd/MMM/YYYY:HH:mm:ss Z" }
    target => "logtime"
  }
}
```

变更

```
grok {
  match => {
    "source_path" => "%{GREEDYDATA}/%{WORD:log_name}..."
  }
}

if "access" in [log_name] {
  grok {
    match => {
      "content" => "%{IPV4:src_ip} - - [%{HTTPDATE:logtime}] \[%{?WORD:verb} %%(URI:PATH:path) ?%{GREEDYDATA}"
    }
  }
  geoip {
    source => "client"
  }
  date {
    # example 17/Jun/2015:07:59:58 +0800
    match => { "logtime", "dd/MMM/YYYY:HH:mm:ss Z" }
    timezone => "+08:00"
  }
}

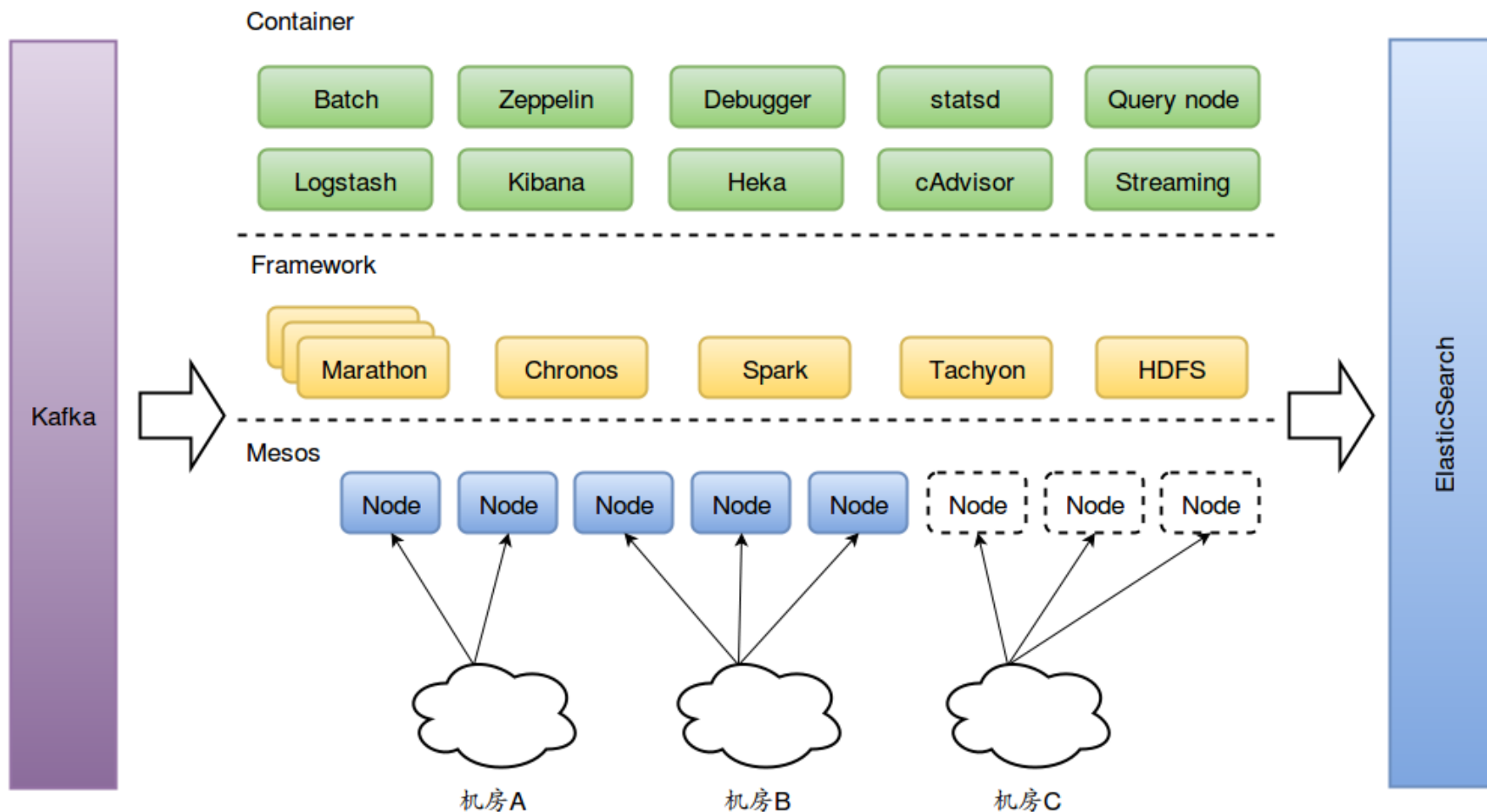
if "wirelessapi" in [log_name] {
  grok {
    match => {
      "content" => "%{TIMESTAMP_ISO8601:logtime}%{GREEDYDATA}businessParams=%{(GREEDYDATA:businessParams)}"
    }
  }
  json {
    source => "businessParams"
    target => "BParam"
  }
  json {
    source => "commonParams"
    target => "CParam"
  }
  json {
    source => "result"
    target => "rst"
  }
  json {
    source => "resultStatus"
    target => "rstStatus"
  }
}
```

我们需要快速应对业务线的变化

# 日志的收集与处理

- 变更后的部署？
  - Salt/Puppet/Chef/Ansible？
- 资源管理？
- Failover/Scheduler/HA...？
- 我们需要的是综合方案
  - Mesos
  - Docker
  - Marathon/Chronos
- 日志服务化！

# 日志服务化



# 日志服务化

- **Docker + Git**应对日志解析规则变更
- 多机房统一管理/日志白名单
- 自动扩容应对业务高峰压力
- 版本控制
  - **Git**和**Marathon**双重保障
  - 快速回滚
- 系统扩容
- 日志收集/解析自助发布
- 监控！告警！

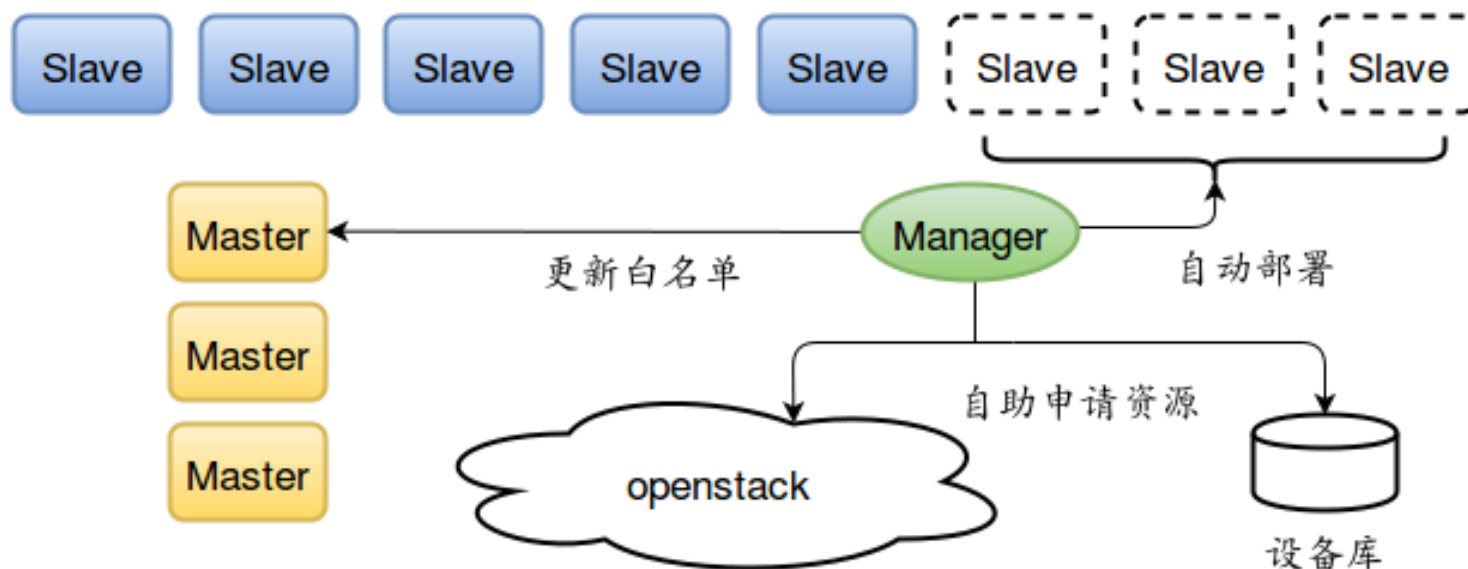
# 日志服务化

- 处理能力
  - 每天处理90~100亿条日志
  - 峰值143亿条日志
  - 平均延迟稳定在150ms
- 存储规模
  - 100TB+的存储池
  - 35亿+document
- 扩容响应
  - 3秒内扩容完成150个实例

# 日志服务化

- 系统扩容
  - 绑定设备库，优先使用未分配的机器
  - 紧急情况申请OpenStack的虚拟机

Mesos Cluster



# 日志服务化

- 日志收集/解析自助发布

- 会用Git就能发布
- 自动添加监控/告警
- 自动关联责任人
- 定时备份Kibana信息
- 自定义数据保存时间
- ...

- 能让程序搞定就不动手！



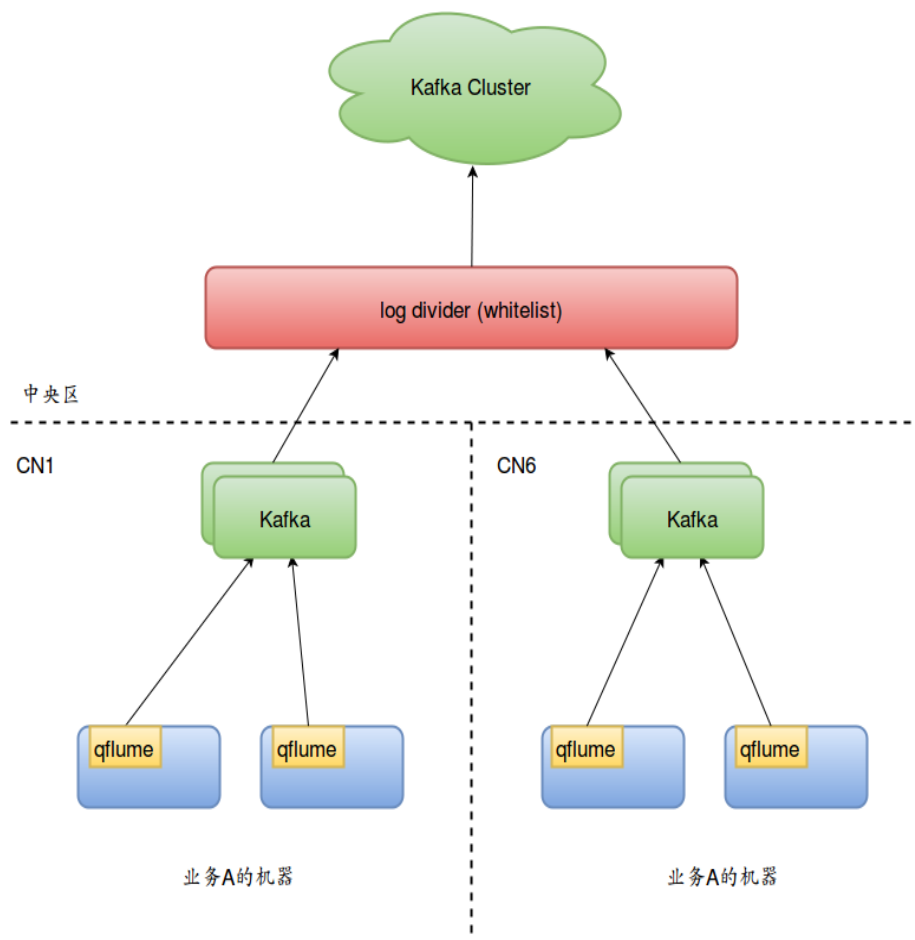
The screenshot shows the Butcher dashboard with a dark header. The main content area is divided into two sections. The top section, titled '统计' (Statistics), displays five metrics: '正在运行任务数' (96), '总共占用资源数' (CPU:630.5 MEM:847.5G), '总任务提交数' (1012), '发布错误数' (91), and '昨日日志量' (73亿). The bottom section, titled '最近的7条队列信息' (Recent 7 queue information), is a table with 7 rows. Each row contains a sequence number, a tag, a server name, a status, a type, a release time, and two action buttons: '重新发布' (Re-release) and '取消发布' (Cancel release).

序号	tag	机房	状态	类型	发布时间	操作
1	[REDACTED]	[REDACTED]	PUBLISHING	GITLAB	2016-01-03 15:22:06.237319	<button>重新发布</button> <button>取消发布</button>
2	[REDACTED]	[REDACTED]	PUBLISHING	GITLAB	2015-12-31 18:29:03.219414	<button>重新发布</button> <button>取消发布</button>
3	[REDACTED]	[REDACTED]	PUBLISHING	GITLAB	2015-12-31 14:23:33.631413	<button>重新发布</button> <button>取消发布</button>
4	[REDACTED]	[REDACTED]	PUBLISHING	GITLAB	2015-12-30 15:27:53.703496	<button>重新发布</button> <button>取消发布</button>
5	[REDACTED]	[REDACTED]	FINISH	GITLAB	2015-12-30 01:36:49.009882	<button>重新发布</button> <button>取消发布</button>
6	[REDACTED]	[REDACTED]	FINISH	GITLAB	2015-12-29 16:19:14.480710	<button>重新发布</button> <button>取消发布</button>
7	[REDACTED]	[REDACTED]	FINISH	GITLAB	2015-12-29 15:34:54.096393	<button>重新发布</button> <button>取消发布</button>



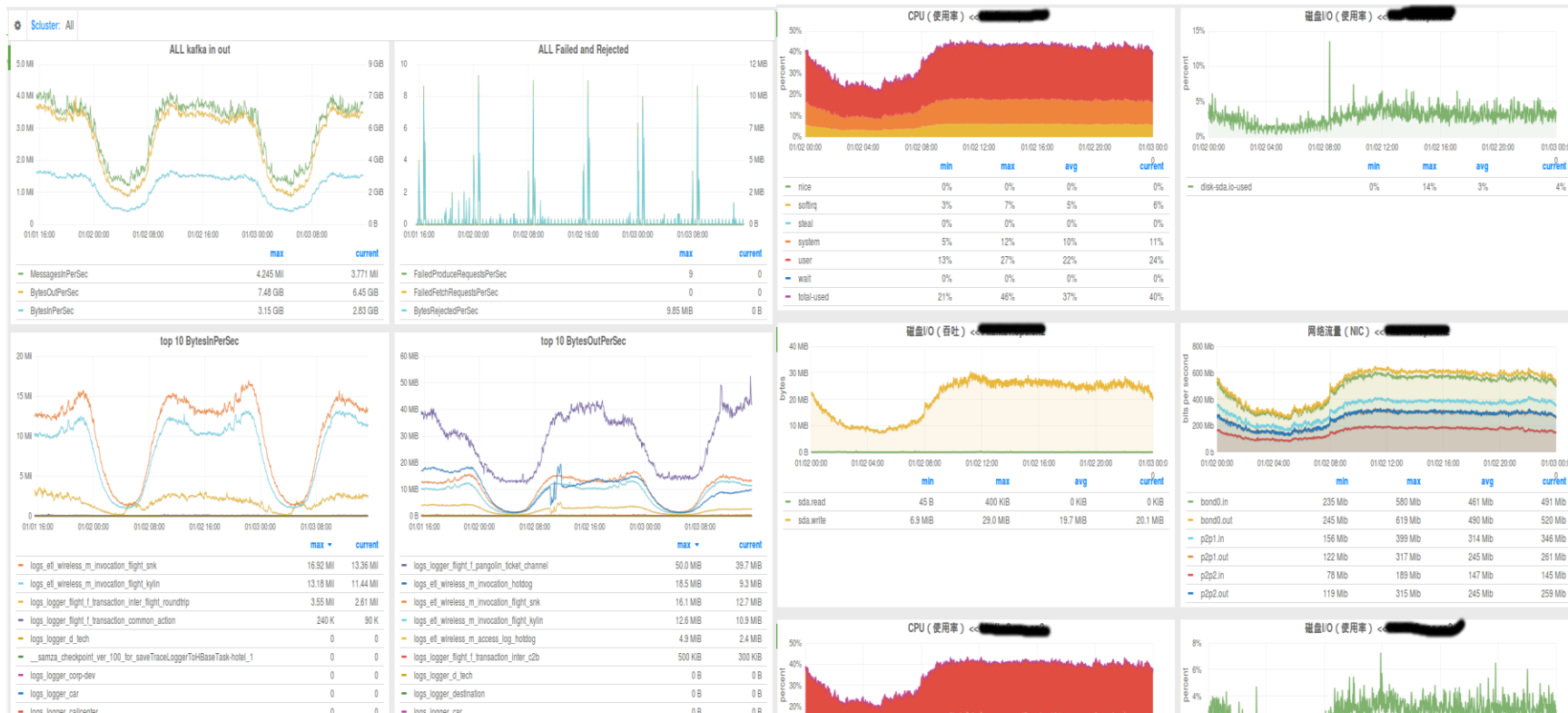
# 日志服务化

- 保证日志的可用性
  - 每个机房独立Kafka
  - 统一的数据引流
  - 日志白名单
  - 传输压缩
- kafka-manager



# 日志服务化

## • 更细节的监控——Kafka



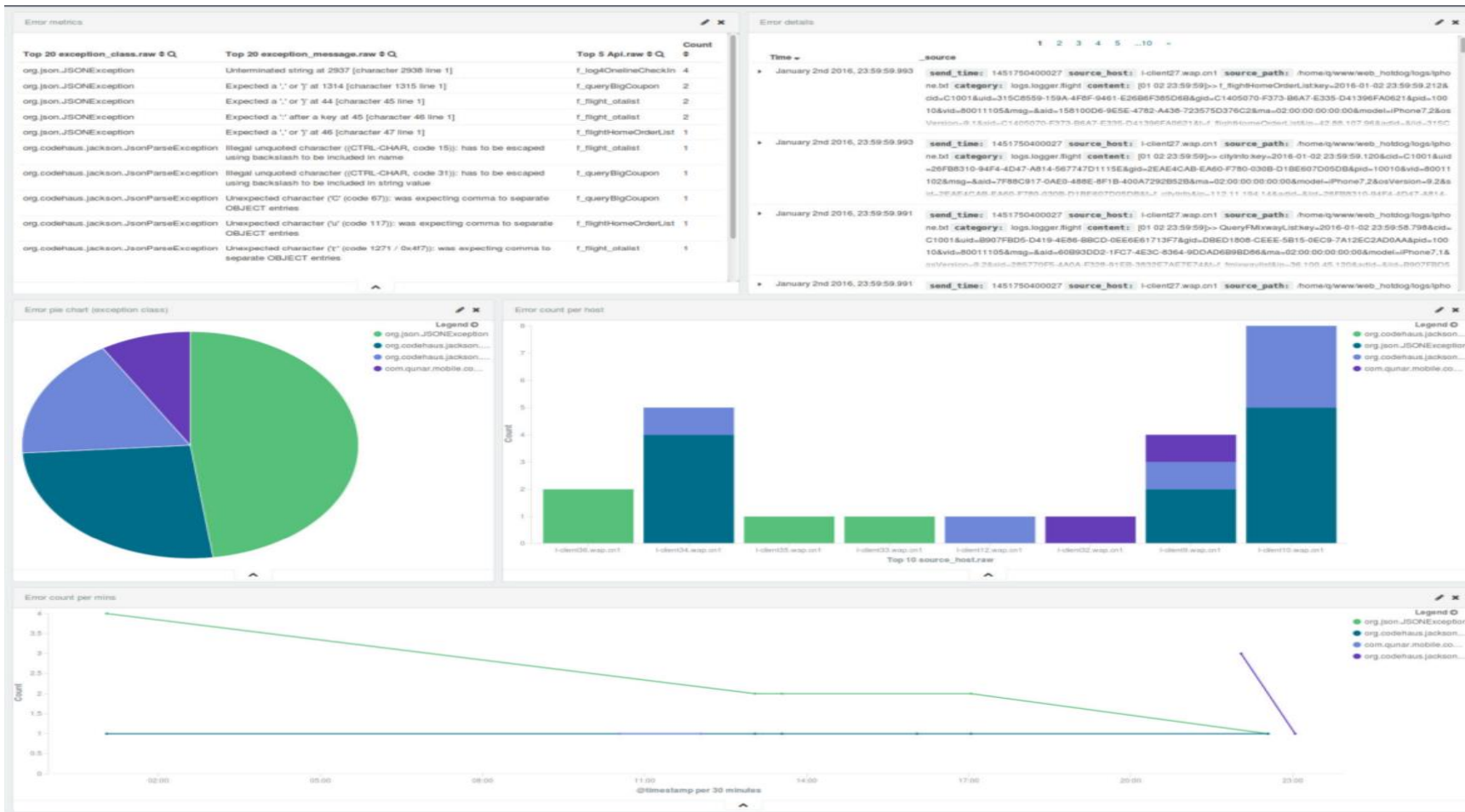
# 日志服务化

- 监控/报警
  - 匹配错误信息，发送到监控平台
  - 严重问题直接发邮件给负责人
- 更多的应用
  - 机器报修
  - cronjob通知
  - salt部署监控
  - Nginx/DNS监控
  - ...

# 开发参与运维

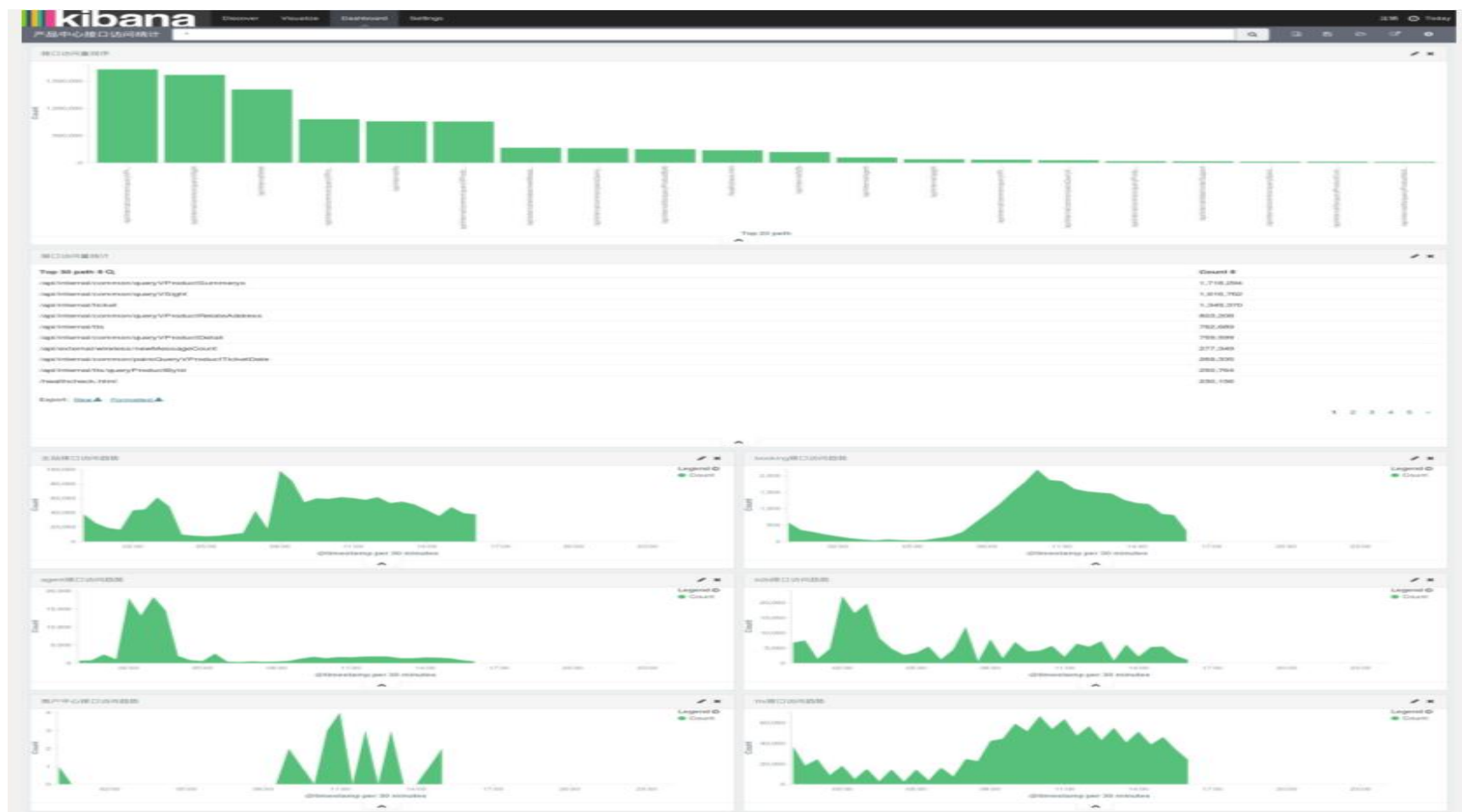
- 日志集中查询/展示
  - 不再申请机器帐号
  - 用可视化工具替代grep/awk/perl等
  - 统计/报表
- 运营和产品也可以用
- 运维不难懂，只是缺工具！
  - 将日志处理好并呈现给他们
  - 帮助开发减少定位的周期

# 开发参与运维



## 应用异常统计：针对线上问题排期修复

# 开发参与运维



服务调用统计：接口延时

谢谢大家