



全球运维大会

2016

DevOps 2.0: 重塑运维价值



北京站

会议时间：12月16日 - 12月17日

会议地点：北京国际会议中心

主办单位：



场景化运维—平台、引擎与自动化

李亚琼 博云CTO



目录



1

运维新挑战or新机遇？

2

从DevOps理念看运维

3

场景化运维技术分析

4

平台、引擎与技术

5

运维未来展望



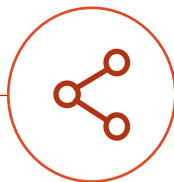
运维“新常态”

随着业务模式多样化及移动互联网用户接入需求，平台规模呈快速增长趋势。



规模大

技术栈复杂



各类操作系统、虚拟化平台、应用中间件、业务配置选项等加大了管理复杂度。

移动互联网的发展使得用户规模爆炸性增长，渠道接入流量更加剧了这种局面



流量大

变更频发



大量新的业务需要上线更新，秒杀、促销等业务变更频发，使得配置管理更加复杂。



某金融客户需求

规模挑战

管理了2000+台X86服务器，若干IBM小型机，Openstack虚拟化平台虚拟机5000+。

人员挑战

人员的技术背景不同，运维的经验也各不相同，特别是针对不同场景下的运维脚本编撰能力不同。

变更挑战

随着互联网业务的开展，特别是渠道流量的接入，系统变更的需求就更加的强烈，需要满足各种变更需求。

不因规模而变

运维平台的能力不因为管理的阶段数量而发生变化，也不因平台类型不同而发生变化

不因人力而变

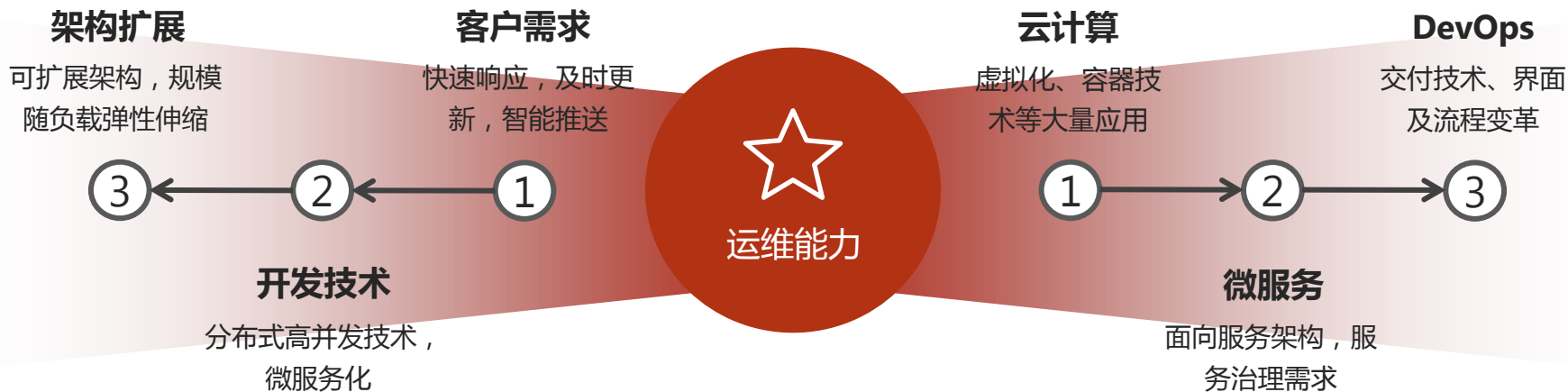
运维平台的能力不因为人员能力或者人员的增加/减少而发生变化，不依赖于运维人员的脚本能力

不因场景而变

运维平台的能力不因为不同平台或软件的变更而发生运维质量变化，具备支持多场景能力



技术发展



运维能力需要满足资源自服务交付、自动化响应、批量更新、
可编排任务、自动巡检与故障定位等



运维技术发展的机遇期？

旧运维模式

传统运维模式的三个重要特征：1，依赖于运维人员的运维管理技能与经营；2，以脚本作为配置管理的主要手段；3，各个系统之间没有打通，运维管理需要在不同的系统与平台间手动切换。

VS

新运维模式

新的运维具有三个方面的特征：1，运维管理不再依赖运维脚本，而是基于场景化的运维工具；2，运维平台强调自动化，能够进行自动化巡检、故障恢复等；3，强调可编排（编程）性，能够通过编排等手段支持复杂的运维场景。

已有技术基础



配置自动化工具

Ansible、Saltstack、Puppet、Chef等配置自动化工具风起云涌，快速发展。

容器技术

以Docker为代表的容器技术重新定义了云计算平台的架构与模型，为实现零配置部署提供了可能。



编排工具

Docker Compose、Kubernetes、Cloudify等编排工具快速发展，为应用自动化发布提供了支撑。

实时分析技术

Storm、Spark Streaming、Riemann等流处理引擎发展迅速，具备实施分析系统状态并作出响应能力。



围绕运维自动化需求，大量新的技术被开发应用。其中，以Ansible、Docker容器、Kubernetes编排、流处理技术等为代表，掀起了一场新的技术变革。



目录

1 运维新挑战or新机遇？



2 从DevOps理念看运维

3 场景化运维技术分析

4 平台、引擎与技术

5 运维未来展望



DevOps理念与功能

面向多种互联网创新应用，支持其一体化、高效、可靠开发和维护



代码构建



服务集成



应用编排



应用管理



产品运营

开发测试对接:

开发与测试人员的环境和现有应用发布平台对接，建设面向容器的开发测试平台，实现无缝对接。

组件/服务仓库:

无论开发端还是运营端都需要把优秀的基座组件或服务固化，降低新应用开发上线时间与风险。

架构连接一切:

开发端基于架构构建应用，运营端基于架构进行自动化配置、弹性伸缩、可用性保障等运维。

配置&部署:

应用的配置集中管理，不用登陆就可以实现应用配置管理、更新、比对、回滚等配置操作。

自动化运维:

固化平台常用的操作，如修改密码、文件比对、备份、安装部署等操作，降低人工操作风险。

DevOps平台



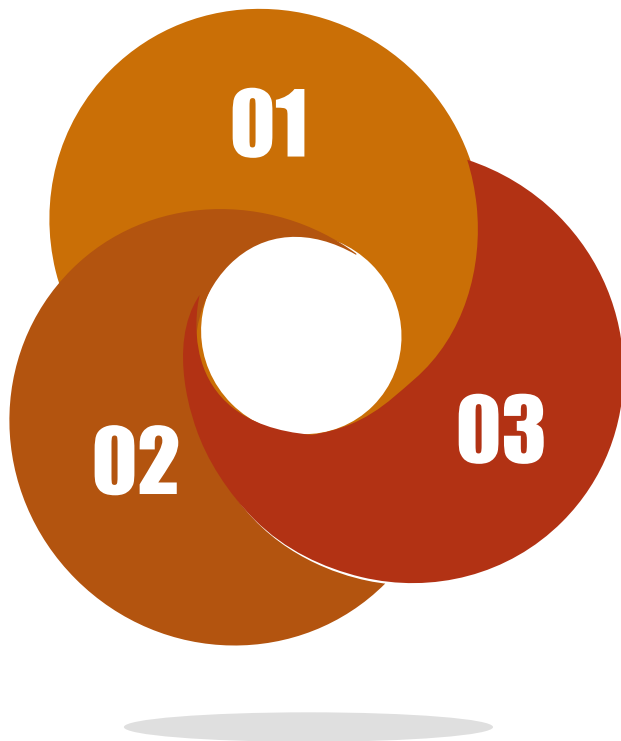
DevOps运维侧

业务零配置

运维人员可以实现业务应用的零配置或简单配置上线，无复杂的手工安装、配置工作。

自动化弹性

平台能够实时跟踪业务运行状态，根据业务负载实时调节其资源占用，弹性伸缩。



支持配置变更

业务版本升级、配置更新等操作可以实现自动化，配置过程能够实现批量化、远程化操作。



DevOps运维四个维度

自动化

运维平台能够自动化实现平台配置、应用部署、故障响应、状态巡检等运维动作。

可编排性

运维动作支持自定义，支持按照用户指定的执行顺序、并发模式、逻辑依赖进行流程编排。



场景化

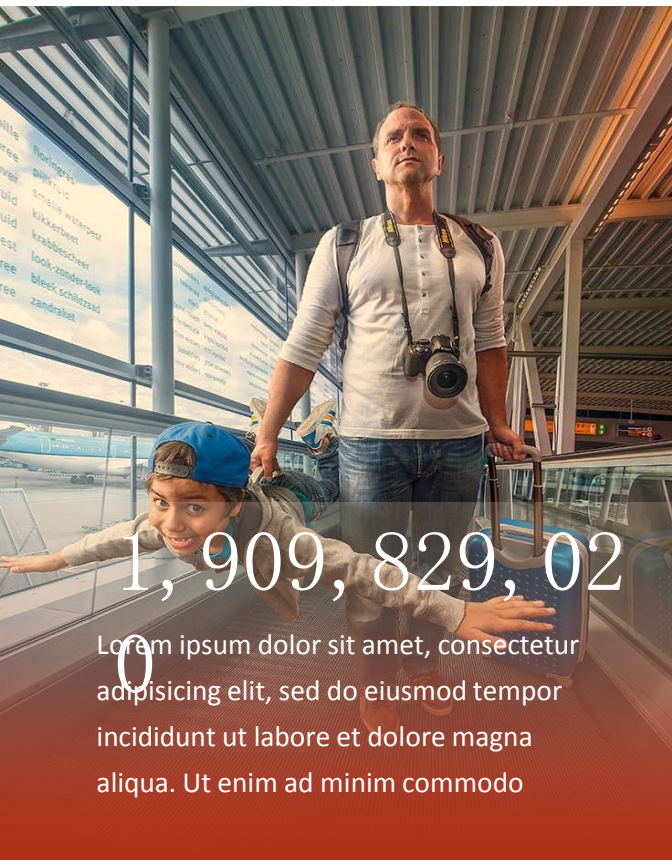
运维平台的功能需覆盖常例化运维中的业务上线、配置变更、故障响应、状态巡检等场景

可编程性

平台具备可编程性，包括运维动作可编程性、运维能力可编程性两个方面。



场景化运维



典型运维场景

1

业务部署

完成业务软件的安装、配置、验证测试、上线切换等动作。

2

配置变更

修改平台、软件、用户等的配置信息以及配置基线管理。

3

状态巡检

对系统运行状态进行常例化巡检，评估系统运行状态。

4

容量分析

实时分析系统容量信息，输出容量报表，管理阈值，触发容量告警。

5

故障响应

能够基于故障类型、业务类型等信息自动化对故障进行分析处理。



自动化运维

1

配置自动化

不依赖于手工而依赖于配置脚本（引擎）对配置进行自动化处理。

2

任务自动化

提供自动化任务执行能力，支撑计划任务、定时巡检、标准比对等自动化任务执行。

3

自动化响应

能够根据异常/事件的上下文信息自动进行故障响应，响应操作包括告警、自动恢复、弹性伸缩等。



可编排性运维



原子任务

运维平台的一个操作或者一个可以执行的脚本，可以被任务引擎直接一次性或周期性执行。

编排任务

根据一定的组成原则由原子任务构成的复合任务，由编排引擎负责解析后下发到任务引擎执行。

任务执行引擎

执行任务的计划任务框架，能够支持单次、循环、定时等任务的执行。



可编程性运维

1

运维脚本可编程性

理论上可以基于脚本实现绝大多数的运维操作，脚本语言的特性赋予了脚本化运维的强大表现力。

2

运维响应可编程性

事件响应是运维的另一个重要工作，可编程性响应模型可以针对不同的事件编程不同的响应动作，支撑负责的自动化响应需求。

3

运维服务可编程性

待运维的对象及操作的sdk化，甚至运维操作的DSL化，使得运维服务作为一种能力能够支持业务应用主动弹性等需求。



目录

1 运维新挑战or新机遇？

2 从DevOps理念看运维

➔ 3 场景化运维技术分析

4 平台、引擎与技术

5 运维未来展望



应用上线场景

业务架构/资源

根据业务需求确认业务的部署拓扑以及资源需求，确认部署拓扑正确性、资源分配合理性。

支撑技术：**拓扑编排、配置自动化**

1

方案确认

2

部署配置

安装与配置

根据拓扑依赖对组成系统的模块/子系统进行按顺序安装部署，同时根据配置不同在不同的阶段进行配置。

支撑技术：**自动部署、自动配置**

3

测试验证

自动化验证

利用自动化测试平台与测试脚本对已部署的系统进行测试验证，确保已部署平台正确、可靠。

支撑技术：**自动化测试**

4

上线切换

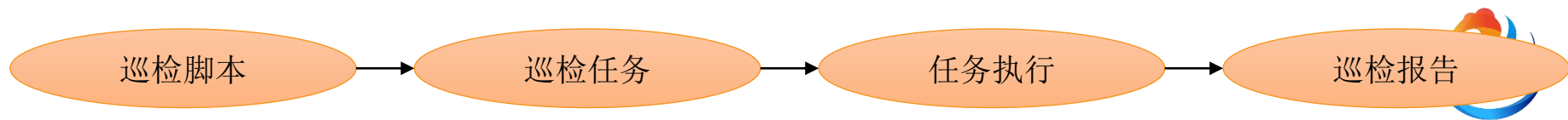
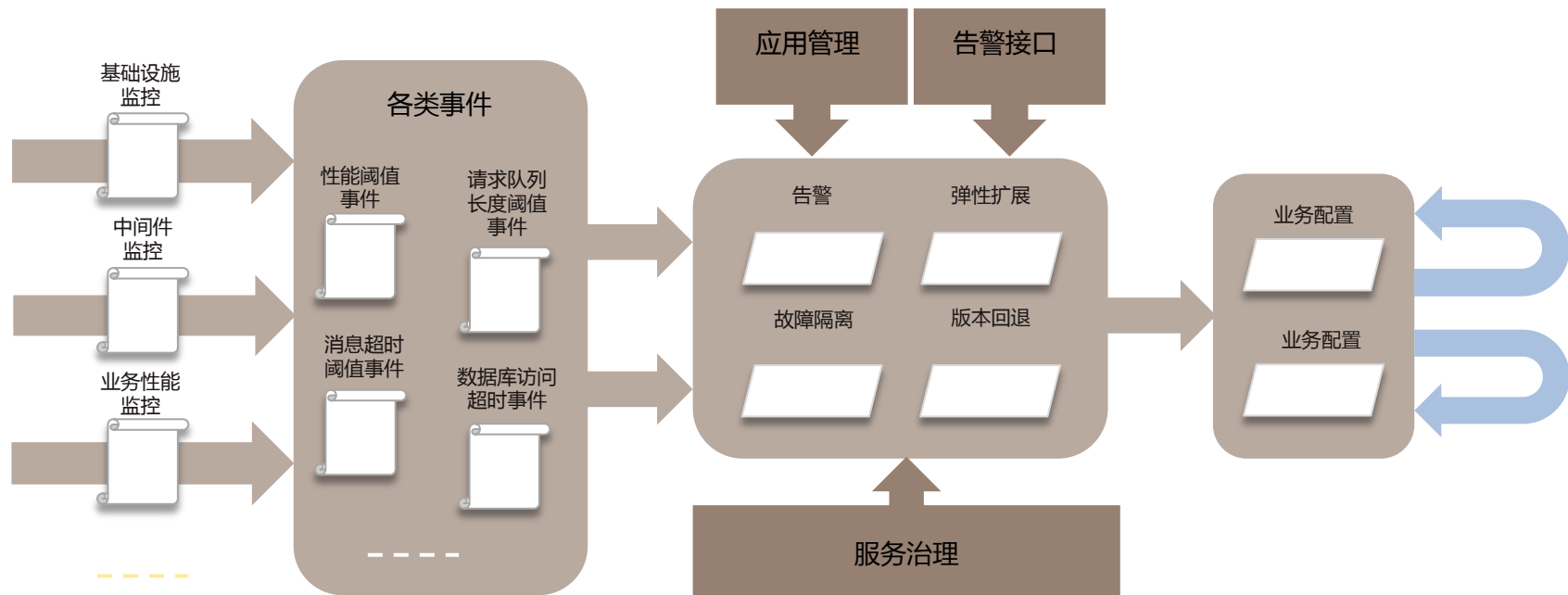
对外发布

切换到对外服务模式，配置前端负载均衡器、打开IP及端口映射、防火墙规则等。

支撑技术：**网络管理**



监控/巡检场景



配置变更场景



01

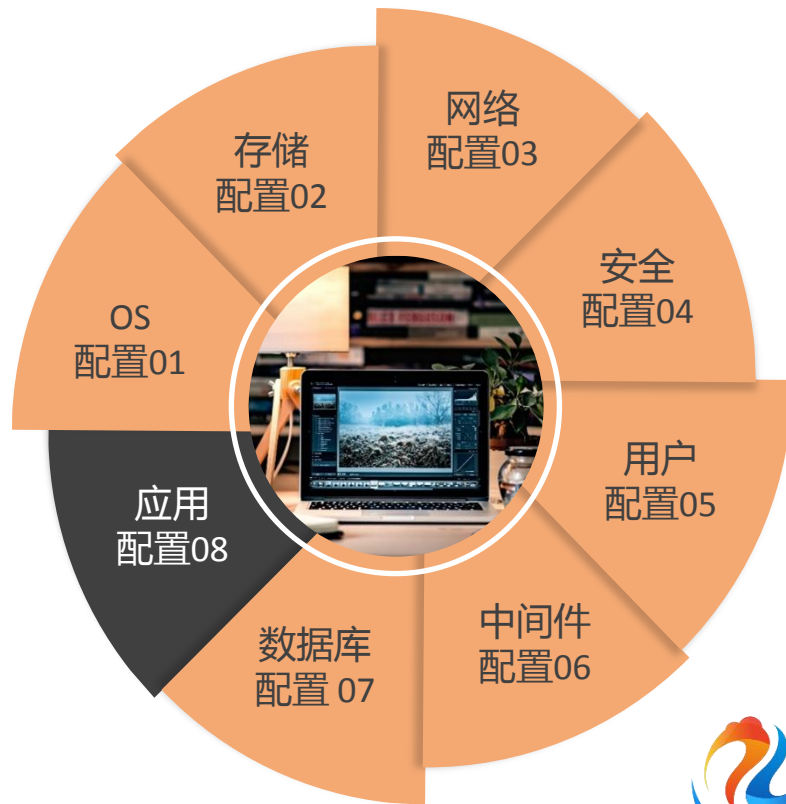
配置变更

满足运维人员在资源分配、环境配置、安全策略、网络隔离、性能优化、软件配置等方面的配置需求

02

基线管理

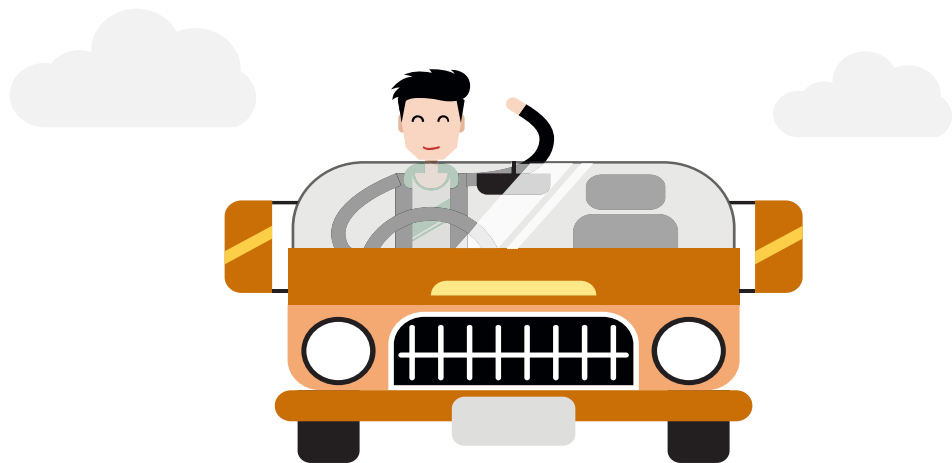
管理平台的各类配置基线，包括配置文件、环境变量、配置项等



应急响应场景

对性能、日志、容量、可用性等事件作出应急响应

事件类型覆盖性能变化、异常（错误）日志、容量告警、可用性异常等，实时分析事件类型、根因等并触发相应的应急响应操作。



规则动态配置

自动响应规则能够实现自定义配置，相关配置规则能够动态生效



对接告警平台

根据需要应急响应流程可以对接各类告警平台，执行邮件、短信等告警



可编程动作

响应动作可编程，具备和其他平台联动能力，能够实现复杂响应动作



事件可升级

应急事件可以进行升级，升级后的事件同样进入响应流程进行响应

目录

1 运维新挑战or新机遇？

2 从DevOps理念看运维

3 场景化运维技术分析

➔ 4 平台、引擎与技术

5 运维未来展望



博云运维产品思路

“聚焦运维自动化”

围绕基础设施、虚拟化平台、中间件、应用等运维管理需求，利用自动化技术来支撑软件部署、配置变更、标准比对、故障自动响应等功能。

- 配置自动化
- 部署自动化
- 巡检自动化
- 响应自动化



配置自动化围绕基础设施平台、操作系统、中间件、数据库、应用等的配置管理，通过自动化技术实现配置的下发。



软件自动化部署重点解决软件部署需求，对不同的软件包、不同的组件基于应用拓扑进行应用部署与配置。

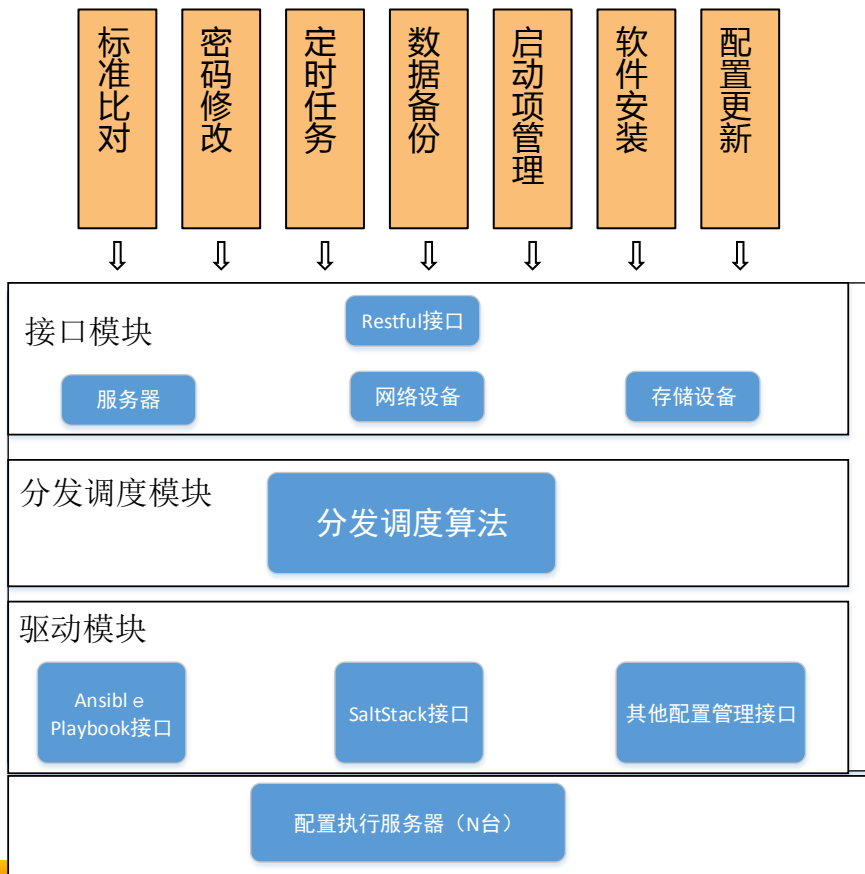


针对常例化运维管理，提供脚本自动执行、定时任务、任务编排等功能，支撑运维操作中的场景化需求。



采集系统运行状态，实时分析系统状态数据，对平台内发生的各类事件进行自动化处理与响应。

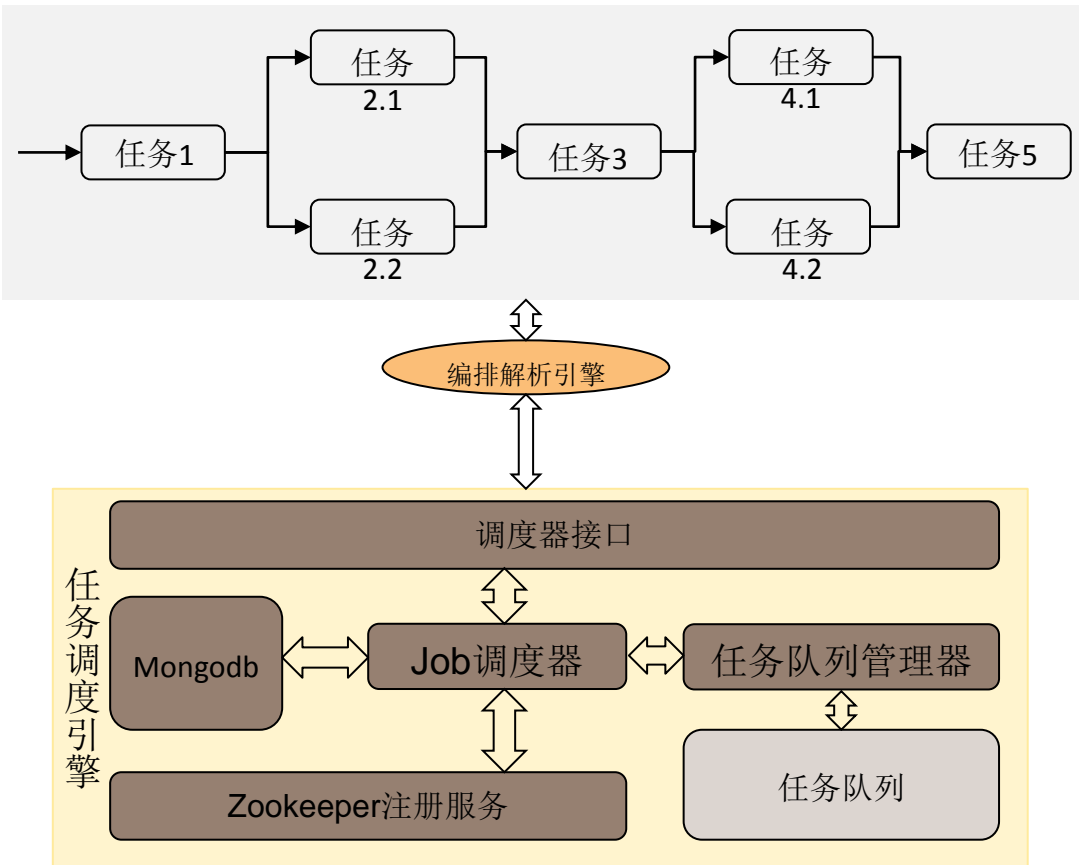
配置自动化



- 核心调度子系统负责平台的任务分发、调度与执行。调度的目标有小型机、x86服务器、虚拟机、网络设备和存储设备。
- 分为接口模块，对其他子系统提供restful接口；调度模块和驱动模块。
- 驱动模块通过三种方式与主机交互。



编排与定时任务



复杂任务编排:

- ◆ 支持多阶段任务顺序执行
- ◆ 支持任务并行执行
- ◆ 可视化拖拽编排（正在实现）
- ◆ 任务可以为脚本执行或配置下发

编排解析引擎:

- ◆ 基于编排生成任务序列
- ◆ 广度优先搜索（基于可视化拖拽）
- ◆ 支持任务权重设置（正在实现）

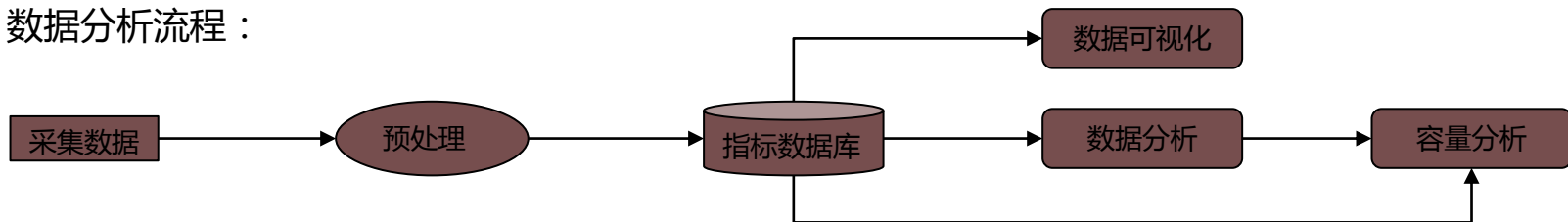
任务调度引擎:

- ◆ 支持任务执行跟踪
- ◆ 支持任务执行结果确认
- ◆ 支持任务批量执行
- ◆ 支持任务分布式执行

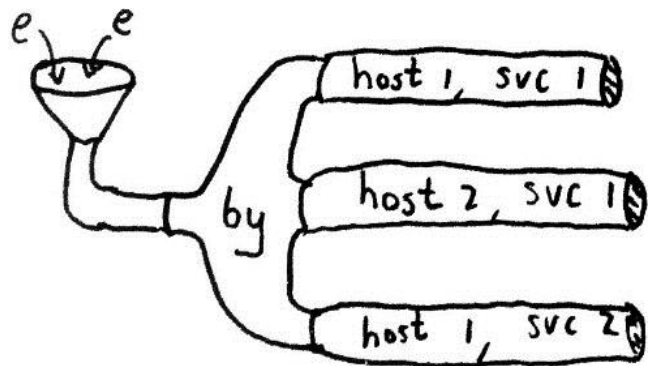


流处理引擎

数据分析流程：



riemann



性能统计

相关聚类

状态变化

依赖关系

安全分析

可用性变化

配置即代码

```
(streams
  #(info %) ; First, measure here
  (where (service #"^riak.+")
    #(info %) ; Then move the info stream here to check the filter
    (by :service
      (coalesce
        #(info %) ; Third, check the coalesced vector of events
        (snap folds/maximum
          #(info %) ; Fourth, probe here to check the maximum calculation
          (with :host nil
            #(info %) ; Finally, check exactly what events are being applied
              ; to the index.
            index))))))
```


统一化管理

云管理平台



资源自服务



基础设施管理



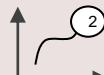
混合云管理



配额与流程审批



多租户支持



监控告警

租户portal

虚拟机
申请

容器部
署

资产管
理

其他目
录

管理员portal

VMWare管
理

Openstack管
理

容器Docker
管理

云环境配置
管理

监控
告警

多租户
配额



VMWare
Provider

Openstack
Provider

Docker
Provider

Aliyun/Azur
e provider

物理dev
Provider

其他服务
provider

配置自动化

监控/告警

申请/审批

日志/审计

用户/角色



VMWare云平台

集群
管理

虚拟机
管理

存储
管理

网络
管理

配置
管理

监控
管理

Openstack云平台

Monitoring

Metering

Backup/Restore

Image

Nova

Cinder

Swift

Neutron

Glance

Openshift云平台

镜像
仓库

容器
管理

持续
集成

持续
部署

编排
引擎

配置
管理

基础设施平台



服务器



存储



网络



安全

目录

1 运维新挑战or新机遇？

2 从DevOps理念看运维

3 场景化运维技术分析

4 平台、引擎与技术

➔ 5 运维未来展望



运维DSL

运维管理语言?

```
Select cpus, mems,  
io_band from Servers  
where ip=10.0.11.12;
```

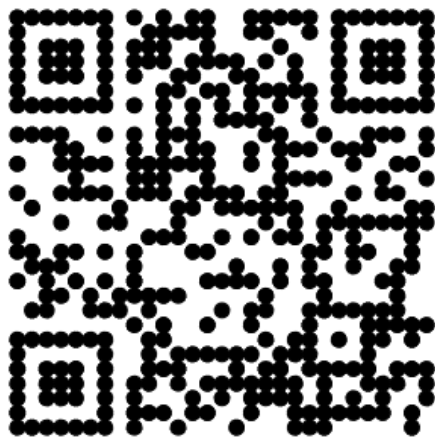


DevOpsDays 即将首次登陆中国



DevOps 之父 Patrick Debois 与您相约

DevOpsDays 北京站 2017年3月18日



门票早鸟价仅限前100名，请从速哟

<http://2017-beijing.devopsdayschina.org/>



GOPS2016
Beijing



想第一时间看到
高效运维社区公众号
的好文章吗？

请打开高效运维社区公众号，点击右上角小人，如右侧所示设置就好





Thanks

高效运维社区
开放运维联盟

荣誉出品

