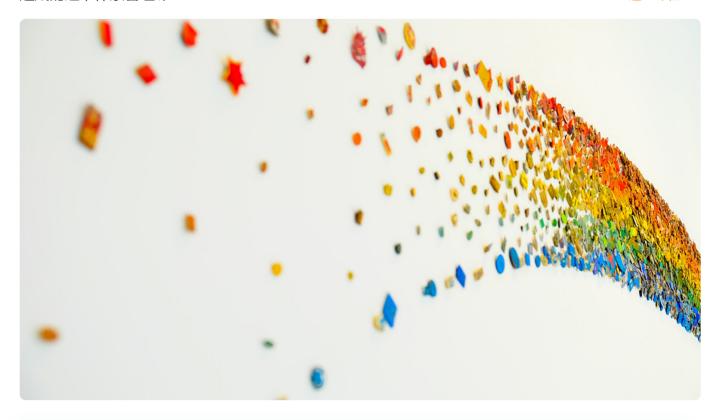
08 | 有了CMDB, 为什么还需要应用配置管理?

2018-01-05 赵成

赵成的运维体系管理课

进入课程 >



讲述: 黄洲君

时长 06:46 大小 3.10M



今天我们分享的主题是:有了 CMDB,为什么还需要应用配置管理?

你不妨先停下来, 思考一下这个问题。

我抛出的观点是: CMDB 是面向资源的管理,应用配置是面向应用的管理。

请注意,这里是面向"资源",不是面向"资产",资源≠资产。

CMDB 是面向资源的管理,是运维的基石

我们一起来梳理一下,在建设运维的基础管理平台时通常要做的事情。

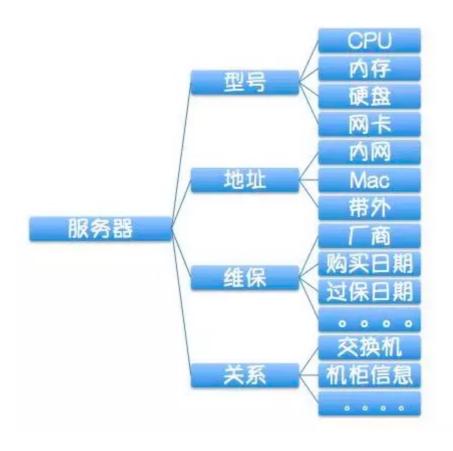
- 第1步,把服务器、网络、IDC、机柜、存储、配件等这几大维度先定下来;
- **第2步**,把这些硬件的属性确定下来,比如服务器就会有 SN 序列号、IP 地址、厂商、硬件配置(如 CPU、内存、硬盘、网卡、PCIE、BIOS)、维保信息等等;网络设备如交换机也会有厂商、型号、带宽等等;
- **第3步**, 梳理以上信息之间的关联关系, 或者叫拓扑关系。比如服务器所在机柜, 虚拟机所在的宿主机、机柜所在 IDC 等简单关系; 复杂一点就会有核心交换机、汇聚交换机、接入交换机以及机柜和服务器之间的级联关系;
- **第 3.5 步**,在上面信息的梳理过程中肯定就会遇到一些规划问题,比如,IP 地址段的规划,哪个网段用于 DB,哪个网段用于大数据、哪个网段用于业务应用等等,再比如同步要做的还有哪些机柜用于做虚拟化宿主机、哪些机柜只放 DB 机器等。

以上信息梳理清楚,通过 ER 建模工具进行数据建模,再将以上的信息固化到 DB 中,一个资源层面的信息管理平台就基本成型了。

但是,**信息固化不是目的,也没有价值,只有信息动态流转起来才有价值**(跟货币一样)。接下来我们可以做的事情:

- **第4步**,基于这些信息进行流程规范的建设,比如服务器的上线、下线、维修、装机等流程。同时,流程过程中状态的变更要同步管理起来;
- 第5步, 拓扑关系的可视化和动态展示, 比如交换机与服务器之间的级联关系、状态 (正常 or 故障) 的展示等, 这样可以很直观地关注到资源节点的状态。

至此,从资源维度的信息梳理,以及基于这些信息的平台和流程规范建设就算是基本成型了。这个时候,以服务器简单示例,我们的视角是下面这样的:



应用配置管理是面向应用的管理,是运维的核心

上面说明了 CMDB 的基础信息部分,如果从传统的 SA 运维模式,这些信息已经足够,但是从应用运维的角度,这些就远远不够了。

这时我们就需要一个非常非常重要的句柄: **应用名**,或者叫应用标识。至此,应用运维里面最最重要的一条联系也就产生了: "**应用名-IP**"的关联关系(这里也可以是定义的其它唯一主机标识,如主机名、容器 ID 等等,因为我们使用的方式是 IP,所以这里就以 IP 示例)。

之所以说"应用名"和"应用名-IP关联关系"非常重要,是因为它的影响力不仅仅在运维内部,而是会一直延伸到整个技术架构上。后面我们会介绍到的所有平台和系统建设,都跟这两个概念有关。

CMDB 是 IP 为标识的资源管理维度,有了应用名之后,就是以应用为视角的管理维度了。 首先看一下应用会涉及到的信息:

应用基础信息,如应用责任人、应用的 Git 地址等;

应用部署涉及的基础软件包,如语言包(Java、C++、GO等)、Web 容器(Tomcat、JBoss等)、Web 服务器(Apache、Nginx等)、基础组件(各种agent,如日志、监控、系统维护类的 tsar 等);

应用部署涉及的目录, 如运维脚本目录、日志目录、应用包目录、临时目录等;

应用运行涉及的各项脚本和命令,如启停脚本、健康监测脚本;

应用运行时的参数配置,如 Java 的 jvm 参数,特别重要的是 GC 方式、新生代、老生代、永生代的堆内存大小配置等;

应用运行的端口号;

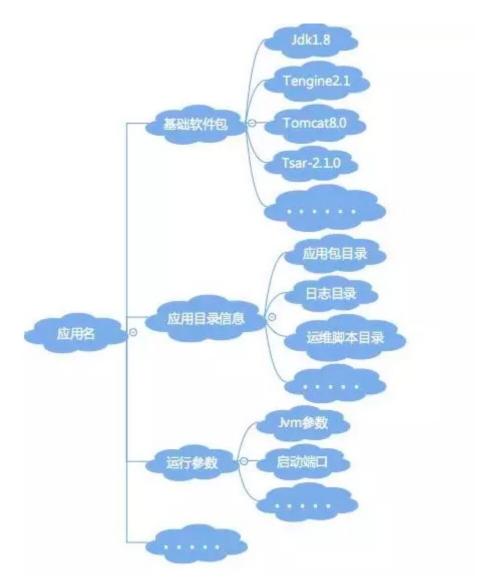
应用日志的输出规范;

其他。

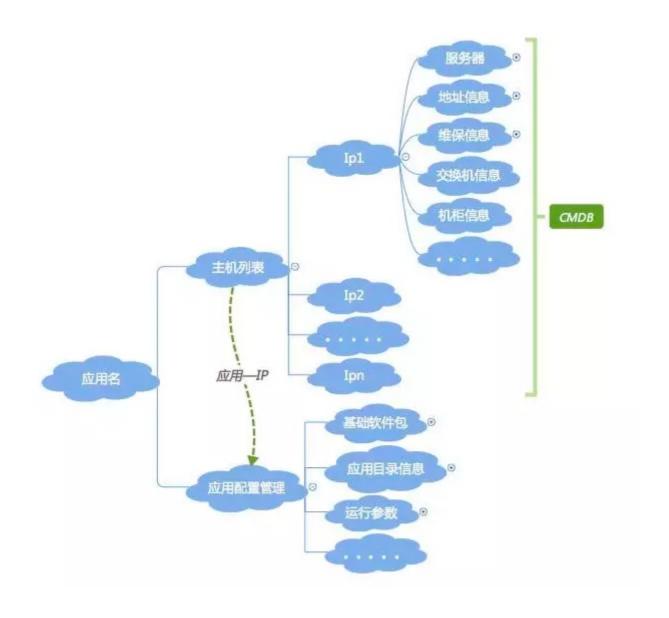
上面的梳理过程实际就是标准化的过程。我们梳理完上述信息后就会发现,这些信息跟 CMDB 里面的资源信息完全是两个维度的东西。所以从信息管理维度上讲,把资源配置和 应用配置分开会更清晰,解耦之后也更容易管理。

好了,按照上面 CMDB 说的套路,梳理完成后,就是要进行信息的建模和数据的固化,这时就有了我们的"应用配置管理"。再往后,就是基于应用配置管理的流程规范和工具平台的建设,这就涉及到我们经常说的持续集成和发布、持续交付、监控、稳定性平台、成本管理等等。

从应用的视角,我们配置管理,应该是下面这样一个视图(简单示例,不是完整的):



好了,有了资源配置信息和应用配置信息,这两个信息应该怎么统一管理起来呢。直接看图:



至此, CMDB 和应用配置管理的分层分解就完成了, 应用名关联着应用配置信息, IP 关联着资源信息, 二者通过"应用名-IP"的对应关系, 联系到一起。

总结

CMDB 是运维的基石,但是要发挥出更大的价值,只有基础是不够的,我们要把更多的精力放到上层的应用和价值服务上,所以我们说应用才是运维的核心。

你可以看到,如果仅仅基于 CMDB 的资源信息作自动化,最多只能做出自动化的硬件资源 采集、自动化装机、网络 - 硬件拓扑关系生成等资源层面的工具,这些工具只会在运维层 面产生价值,离业务还很远,就更谈不上给业务带来价值了。

但是基于应用这一层去做,就可以做很多事情,比如持续集成和发布、持续交付、弹性扩缩容、稳定性平台、成本控制等等,做这些事情带来的价值就会大大不同。

以上就是我抛出的观点,CMDB 是面向资源的管理,应用配置是面向应用的管理。希望能够抛砖引玉,听到更多你的观点和反馈。

如果今天的内容对你有用,也希望你分享给身边的朋友,我们下期见!



© 版权归极客邦科技所有,未经许可不得传播售卖。 页面已增加防盗追踪,如有侵权极客邦将依法追究其法律责任。

上一篇 07 | 聊聊CMDB的前世今生

下一篇 09 | 如何在CMDB中落地应用的概念?

精选留言 (6)



心 3



兵又

2018-01-06

请教赵老师,CMDB和应用配置管理也是持续集成和发布系统的基石,但如果现状是没有应用配置管理,该如何做好持续发布系统?对于持续发布这一块您有什么好的实践吗? 展开〉 作者回复: 规模不大,应用不多的时候,这两个东西没有问题也不大。但是这种情况,就成了我前面说的直接就冲着各种工具去了,忽略了基础,当规模变大时,就会有各种信息不同步,不统一,即使有工具效率也上不去。

持续发布我后面有专门一个系列介绍。



心 2

对于公有云的CMDB的建设,总感觉是重复建设,动力不是那么足,只是说考虑到今后混合云的可能,做一层抽象层来解耦。公有云的CMDB的构建有什么特别的地方么?

作者回复: 我的建议是从你实际的运维对象入手,识别出他们,前面应用生命周期的文章有介绍对应的套路。

再就是CMDB是我们的手段,不是问题,所以思路不一定是从CMDB入手,而是从你遇到的问题入手,所以可以问一下自己现在遇到的具体问题场景是什么?



杨陆伟



2019-03-30

老师的思路很开阔,方案很大,请问下在老师的公司实现了吗? 展开~



மி

太棒了, 听君一席话 胜读十年书

展开٧



ľ

后悔没有第一时间来追老师的剧,老师写完专栏才来。读了前面几篇就已经如获至宝了, 多年的运维积累感觉可以有可能被串起来抽象化平台化了。按耐住一口气读完的冲动,细



ம

请教下赵老师,有开源的同时支持资源配置和应用配置的cmdb开源软件吗? 展开~