

01 | 为什么Netflix没有运维岗位？

2017-12-20 赵成

赵成的运维体系管理课

[进入课程 >](#)



讲述：黄洲君

时长 10:37 大小 4.86M



众所周知，Netflix 是业界微服务架构的最佳实践者，其基于公有云上的微服务架构设计、持续交付、监控、稳定性保障，都为业界提供了大量可遵从的原则和实践经验。

Netflix 超前提出某些理念并付诸实践，以至于在国内众多互联网公司的技术架构上也可以看到似曾相识的影子。当然殊途同归也好，经验借鉴也罢，这都不影响 Netflix 业界最佳实践者的地位。

同样，在运维这个细分领域，Netflix 仍然是最佳实践的典范。所以专栏的开始，就让我们一起看看世界顶级的互联网公司是如何定义运维以及如何开展运维工作的。

Netflix 运维现状

准确一点说，Netflix 是没有运维岗位的，和运维对应的岗位其实是我们熟知的 SRE（Site Reliability Engineer）。但我们知道 $SRE \neq$ 运维，**SRE 理念的核心是：用软件工程的方法重新设计和定义运维工作。**

也就是要改变之前靠人去做运维的方式，转而通过工具体系、团队协作、组织机制和文化氛围等方式去改变，同时将之前处于研发体系最末端的运维，拉回到与开发肩并肩的同一起跑线上。

但是 Netflix 的神奇和强大之处在于，连 Google 都非常重视和大力发展的 SRE 岗位，在 Netflix 却没有几个。按照 Netflix 一位技术主管（Katharina Probst）在今年 9 月份更新的博客中所描述，在 1 亿用户，每天 1.2 亿播放时长、万级微服务实例的业务体量下，SRE 人数竟然不超过 10 个，他们称这样的角色为 Core SRE。描述具体如下：

100+ Million members. 125+ Million hours watched per day.
Hundreds of microservices. Hundreds of thousands of instances. Less than 10 Core SREs.

不可否认，Netflix 拥有强大的技术实力和全球最优秀的工程师团队。按照 SRE 的理念，完全可以打造出这一系列的工具产品来取代运维和 SRE 的工作。但是能够做到如此极致，就不得不让人惊叹和好奇，Netflix 到底是怎么做到的？

为什么 Netflix 会做得如此极致？

这确实是一个很有意思的问题，带着这样的疑问，阅读了大量的 Netflix 技术文章和公开的演讲内容之后，我想这个问题可以从 Netflix 的技术架构、组织架构、企业文化等几个方面来看。

1. 海量业务规模下的技术架构和挑战

前面我也提到，Netflix 在业务高速发展以及超大规模业务体量的驱动下，引入了更为灵活的微服务架构，而且已经成为业界的最佳实践典范。

引入微服务架构后，软件架构的细化拆分和灵活多变，大大提升了开发人员的开发效率，继而也就提升了业务需求的响应和迭代速度，我想这也正是微服务思想在业界能够被广泛接受和采用的根本原因。

但是这也并不是说没有一点代价和成本，随之而来的便是架构复杂度大大增加，而且这种复杂度已经远远超出人的认知能力，也就是我们已经无法靠人力去掌控了，这也就为后续的交付和线上运维带来了极大的难度和挑战。这时，微服务架构下的运维就必须要靠软件工程思路去打造工具支撑体系来支持了，也就是要求微服务架构既要能够支持业务功能，还要能够提供和暴露更多的在后期交付和线上运维阶段所需的基础维护能力。

简单举几个例子，比如服务上下线、路由策略调整、并发数动态调整、功能开关、访问 ACL 控制、异常熔断和旁路、调用关系和服务质量日志输出等等，要在这些能力之上去建设我们的运维工具和服务平台。

讲到这里，我们可以看到，微服务架构模式下，运维已经成为整体技术架构体系中必不可少的一部分，而且与微服务架构相关的体系是紧密相连不可拆分的。

我们现在看到很多跟 SRE 相关的文章或内容，对于 SRE 产生原因的解释，大多是说为了缓解开发和运维之间的矛盾，树立共同的目标，让双方能够更好地协作配合。这样理解也没有太大的问题，但是我认为不够充分，或者说以上这些只能算是结果，但不是背后的根本原因。

我理解的这背后最根本的原因是，微服务架构复杂度到了一定程度，已经远远超出单纯的开发和单纯的运维职责范畴，也远远超出了单纯人力的认知掌控范围，所以必须寻求在此架构之上的更为有效和统一的技术解决方案来解决复杂度认知的问题。进而，在这一套统一的技术解决方案之上，开发和运维产生了新的职责分工和协作方式。

目前业界火热的 DevOps 理念及衍生出来的一系列话题，我们可以仔细思考一下，其实也是同样的背景和逻辑。DevOps 想要解决的开发和运维之间日益严重的矛盾，究其根本，还是微服务架构背后带来的技术复杂度在不断提升的问题。

Netflix 带给我们的启示一：微服务架构模式下，我们必须换一个思路来重新定义和思考运维，运维一定要与微服务架构本身紧密结合起来。

2. 更加合理的组织架构和先进的工具体系及理念

我上面提到，在微服务架构模式下，运维已经成为整体技术架构和体系中不可分割的一部分，两者脱节就会带来后续一系列的严重问题。

从这一点上，不得不说 Netflix 的前瞻性和技术架构能力还是很强的。因为早在 2012 年，甚至更早之前，Netflix 就已经意识到了这个问题。在组织架构上，将中间件、SRE、DBA、交付和自动化工具、基础架构等团队都放在统一的云平台工程（Cloud and Platform Engineering）这个大团队下，在产品层面统一规划和建设，从而能够最大程度地发挥组织能力，避免了开发和运维的脱节。

当然，这个团队不仅没让大家失望，还给我们带来了太多惊喜。业界大名鼎鼎的 NetflixOSS 开源产品体系里，绝大部分的产品都是这个团队贡献的。

比如持续交付系统 Spinnaker；稳定性保障工具体系 Chaos Engineering，这里面最著名的就是那只不安分的猴子，也正是这套稳定性理念和产品最大程度地保障了 Netflix 系统的稳定运行；被 Spring Cloud 引入并得以更广泛传播的 Common Runtime Service&Libraries，而且大家选用 Spring Cloud 基本就是冲着 Spring Cloud Netflix 这个子项目去的。当然还有很多其它优秀的开源产品，这里我就不一一介绍了。

Netflix 带给我们的启示二：合理的组织架构是保障技术架构落地的必要条件，用技术手段来解决运维过程中遇到的效率和稳定问题才是根本解决方案。

3. 自由与责任并存的企业文化

上面讲了这么多，看上去好像就是 SRE 常见的理念和套路，只不过 Netflix 在开源和分享上更加开放和透明，让我们有机会更多地了解到一些细节。但是为什么 Netflix 会做的如此极致呢？好像我们一直没有回答到这个问题，那这里就不得不提 Netflix 的企业文化了。

Netflix 的企业文化是 **Freedom & Responsibility**，也就是自由和责任并存，高度自由的同时，也需要员工具备更强的责任心和 Owner 意识。

体现在技术团队中就是，You Build It, You Run It。工程师可以随时向生产环境提交代码或者发布新的服务，但是同时你作为 Owner，要对你发布的代码和线上服务的稳定运行负责。

在这种文化的驱使下，技术团队自然会考虑从开发设计阶段到交付和线上运维阶段的端到端整体解决方案，而不会是开发就只管需求开发，后期交付和维护应该是一个叫运维的角色去考虑。No，文化使然，在 Netflix 是绝对不允许这种情况存在的，你是开发，你就是 Owner，你就要端到端负责。

所以，短短两个英文单词，Freedom & Responsibility，却从源头上就决定了团队和员工的做事方式。

Netflix 带给我们的启示三：Owner 意识很重要，正确的做事方式需要引导，这就是优秀和极致的距离。

总结

通过上面的分析，我们可以看到，Netflix 在其技术架构、组织架构和企业文化等方面的独到之处，造就了其优秀的技术理念和最佳实践。从运维的角度来说，无论是 SRE 也好，还是 DevOps 也罢，都被 Netflix 发挥到了极致。

当然，Netflix 能做到这一点，是需要非常强大的技术实力和人才储备的。当前我们虽然没法直接套用，但是这并不妨碍我们在某些经验和思路上去借鉴和学习。

比如，现在很多公司在采用了微服务架构后，就没有充分考虑到后续基于微服务架构的运维问题。而且在运维团队设置上，仍然是脱离整个技术团队，更不用说将其与中间件和架构设计等团队整合拉通去建设，自然也就谈不上在产品层面的合理规划和建设了。

因此导致的问题就是运维效率低下，完全靠人工，线上故障频发，但是处理效率又极低，开发和运维都处于非常痛苦的状态之中，运维团队和成员也会遭遇到转型和成长的障碍。

以上这些问题都很棘手，亟待解决。

通过今天的分享，我们了解了 Netflix 的技术团队运作模式和思路，不知道给你带来哪些启示呢？

如果今天的内容对你有帮助，也请你分享给身边的朋友。

欢迎你留言与我一起讨论。

赵成的运维体系管理课

带你直击运维的本质

赵成

美丽联合集团技术
服务经理



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 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 开篇词 | 带给你不一样的运维思考

下一篇 02 | 微服务架构时代，运维体系建设为什么要以“应用”为核心？

精选留言 (16)

写留言



宵伯特

2017-12-21

10

就像自动化取代传统手工业，技术的发展使得生产率提升的同时，也使得行业内的职能配置不断发生变化。现在大部分的软件行业由于敏捷开发的流行也逐渐的淘汰或替换了传统的QA部门，运维的职能也逐渐的从传统的手工业向自动化不断的改进，软件的生命周期也早已不是传统瀑布开发所规定的那样简单。从软件开发的历史来看，代码层级的一些规范也逐渐的向架构层次延展，像微服务的出现不就是kiss原则和dry原则在架构设计上的应...

展开

作者回复：赞！技术应用也是需要看场景和时机的。



白开水

2017-12-21

👍 4

有具体的Netflix原则和最佳实践参考吗？

展开 ▾

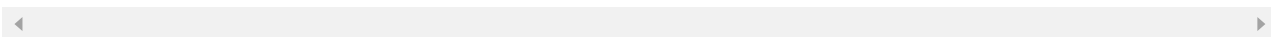
作者回复: 推荐两个Netflix的技术网站(需要翻越):

Netflix的Tech Blog: <https://medium.com/netflix-techblog>

Netflix公开分享的PPT合集: <https://www.slideshare.net/netflix>

特别是第一个, Netflix会持续更新他们的技术动态和实践, 很好的学习案例。slideshare这个虽然内容大部分是前些年的, 不过仍然会有很大的启发意义, 在这里会体验到Netflix的技术前瞻性。

后面我也会会有文章分享我们的一些实践, 期望对你会有帮助。



朱雯

2019-04-18

👍 2

运维价值

1:定义范围:除了业务需求实现层面, 其他都是运维

价值所在: 运维能力就是架构能力,运维层面爆发问题或者故障, 一定是技术架构中的问题, 单纯看待没有意义

矛盾:割裂运维和开发, 按照软件生命周期开始划分运维和开发, 所以应该先转换运维思...

展开 ▾



岑崙

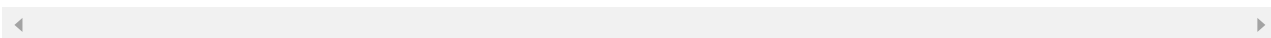
2017-12-20

👍 2

在组织架构设计方面有没有什么方法论层面的内容和方向

展开 ▾

作者回复: 会有的, 专门有两篇介绍组织架构和协作, 以及个人转型的文章。



玉剑冰锋

2018-09-03

👍 1

老师您好，听了两小节了课程了，有几个想法想听听您的见解:1.案例中提到的毕竟是整个行业的风向标或者说是趋势，但是这毕竟是大公司或者具备很强团队才能达到预期效果，如果一个公司才几十个人或者上百人，这种情况该如何处理？2.我相信大部分中小型公司运维岗位还是必须要设立的，而且更多的是人肉运维，是否能给这些人一些建议或者发现方向，谢谢！

展开 ∨



诺克大叔

2018-03-26

👍 1

owner 直接体现了自带责任属性 可以避免好多推锅和定位故障难问题

展开 ∨

作者回复: 把事情当作自己的，处理起来最高效



zyq

2017-12-25

👍 1

我能明白市场选择所带来的阵痛，不管是对企业对个人。其实运维主要面临两个问题：一个是企业方面，如果市场趋势如此，那企业在选择业务服务的话，是否应该把运维独立出来以服务的形式提供，并且押宝在上面。另一个就是运维人员，因为如果按照这种趋势下去，运维最终会由工具的使用者转变为工具的制造者。这个转变是巨大的，也有会一部分开发者进入这一领域，对传统运维冲击很大。另外就是对第三方独立应用来说，如何提...

展开 ∨



zyq

2017-12-22

👍 1

作为运维人员如何权衡自己的发展呢？如果将来的运维人员的发展方向更偏于容器，k8s，自动化。那企业内应用，比如说windows的AD运维，exchange运维。企业如何在没有开发团队的基础上实现运维？开发团队如何在远离生产环境的模式实现owner&build？

作者回复: 我看下来应该是三个问题，也确实是一个个非常典型的现状，我的答复如下：

第一个，关于如何衡量个人发展，这个问题说实话有点大，我可能无法回答全面，我建议可以把问题再具体一些。然后，个人是否能够有发展，是否能够发展地还不错，这里暗含着一个关键的前提，就是要看你个人想要什么，想要成长成什么样子。

第二个，容器、k8s和自动化，一定是未来的演进趋势，因为它们确实能够带来更高的效率，让人

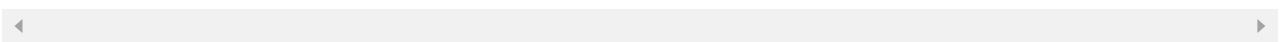
工作的更轻松，任何技术的发展只要能够起到上面这两个作用，就一定是有生命力的，代表着未来趋势。

第三个，关于企业内第三方产品的运维和开发问题，我的建议是当内部无法基于这些产品进行运维效率和稳定性提升方面的二次开发时，还是尽量依赖产品方的服务，看产品方是否可以提供一些运维方便产品技术支持。

从你反馈的单个案例来看，如果还是保持现状，你提到的运维和开发这两个事情都很难改变，因为受限于产品形态和合作模式。但是从整个业界来看，你提到的这两个产品，其实微软都已经将它们上云了，如果是在云上应用，这些问题是不是就不是问题了？

可能有些问题的答案是在问题之外的，这一点可以多考虑一下，因为随之而来的还会有其它一系列问题要去考虑。

思考之后，可以继续留言给我。



leo

2017-12-21

👍 1

期待后续更新

展开 ▾



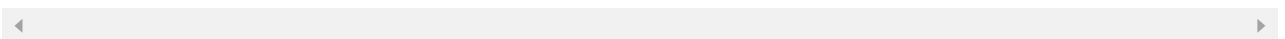
zyq

2017-12-21

👍 1

传统应用的运维模式会逐渐转变到这种模式下吗？那运维外包业务岂不是会慢慢萎缩，运维人员的成长也被局限了

作者回复: 必然会向这个方向发展，是不是会萎缩不一定，但是发展空间一定会受限。后面我也会有一篇也想来介绍这个情况。



苦行僧

2019-03-27

👍

you build it , you run it

展开 ▾





趁早

2019-01-08



那具体的话，作为一个owner如何保障自己“随意”每次提交的代码没有问题？



cc

2018-11-19



很有感触，我觉得这种模式正是需要的，运维不能只做背锅侠

展开 ▾



kkggo

2018-09-05



这种技术储备要掌握那些技能

展开 ▾



大爷静悄悄

2018-05-20



Netflix 带给我们的启示二：合理的组织架构是保障技术架构落地的必要条件，用技术手段来解决运维过程中遇到的效率和稳定问题才是根本解决方案。



希望

2018-01-23



从专精的角度上考虑，开发和运维技术栈，看待问题的角度还是有所区别的，我比较好奇Netflix的开发人员如何做到既能关注自身开发的功能，又能关注到产品上线后的运维和部署架构？他们到达今天这个程度是开发的能力推动，还是因为运维基础平台的完善推动？

作者回复: 这个是自驱动型的，当然，很重要的一个前提是，他们技术能力非常强大，这个是国内达不到的，所以更多的需要组织架构上互补。

