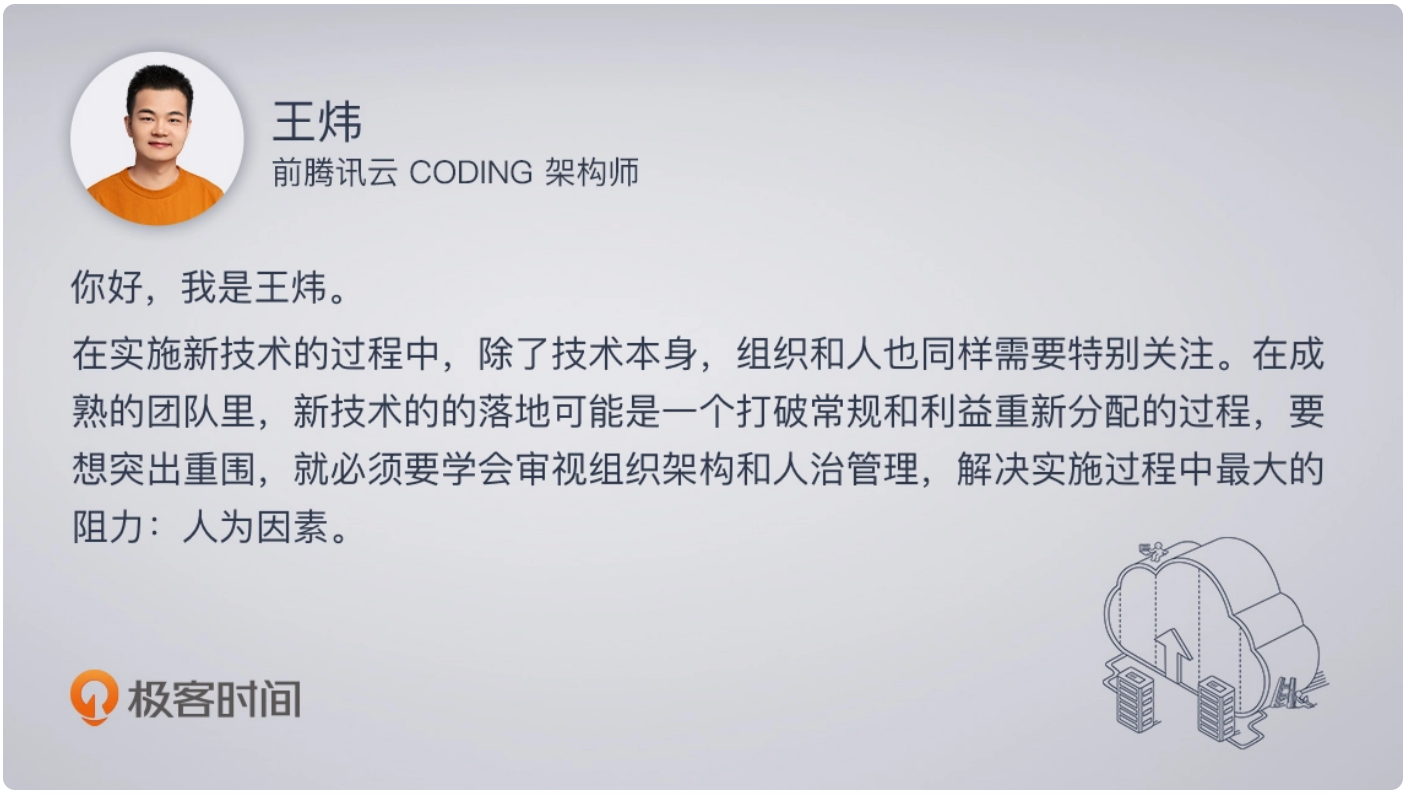
结束语丨下一步，我该如何在公司落地 GitOps？

2023-03-13 王炜 来自北京

《云原生架构与GitOps实战》 课程介绍 >



讲述：王炜

时长 08:36 大小 7.86M

你好，我是王炜。

首先恭喜你完成了专栏的学习，到这里，我想你已经掌握了 GitOps 的所有知识。现在，我相信你也已经跃跃欲试了。

今天这节课是我们专栏的最后一讲，我想抛开技术，跟你聊一下新技术方案落地过程中最大的影响因素：人。

我们知道公司是由人组成的，如果你有过从零到一落地一项新技术的经验，我相信你多少都会有一些不太好的回忆。当新技术来临时，组织内每个人的反应都是不同的，有好奇、有接纳、有拒绝，可能也有嫉妒。在我看来，我们很难用技术的先进性来说服迁移对象迅速接受一门新技术，你需要学会站在更高的角度思考问题。

这节课，我们就来讨论如何落地 GitOps、落地过程可能面临的挑战以及如何解决这些问题，带你从非技术的视角，并从组织层面去考虑问题。

如何说服团队？

在落地一项新技术时，我们首先要做的是说服团队。在这里，我建议你结合云原生架构和 GitOps 的优点，从项目的实际情况出发，向团队成员介绍新架构下的发布和运维优势，目的是对齐双方的目标：提升团队研发效率。

俗话说不打无准备之仗，你需要提前准备好 GitOps 演示 Demo 和 PPT，并尝试把重点放在下面这几个点上。

1. 研发自助发布
2. 提升发布效率
3. Kubernetes 对业务的帮助
    a. 不中断发布
    b. 负载均衡，避免单点故障
    c. 服务自愈，业务不宕机
    d. 根据业务高低峰自动扩缩容
    e. 高级部署策略：蓝绿、灰度、金丝雀
4. 标准应用，多云可移植性

对于实施 GitOps 的团队来说，由于我们的客户是开发者，所以要想打动他们就需要从他们的角度和痛点出发，并设计能够解决他们核心痛点的技术方案和 GitOps 工作流，只要做到这一点，就几乎成功了一半了。

在 Demo 演示阶段，你同样需要做好充分的准备，确保在现场演示完整的发布过程，以便留下好的印象。通常，当演示完之后，接下来的环节将是对你心理承受能力的考验，这时候你往往会面临开发同学的灵魂发问，甚至遇到一些尖锐的问题，比如：

1. 新的方案在工作流程上对我的影响是什么？
2. 迁移大概需要多久的时间？

3. 我们需要怎样配合？
4. 迁移过程如果导致生产故障怎么办？
5. 会影响我们日常的工作吗？
6. 我们组暂时不想迁移行不行？

结合你团队的实际情况，你需要提前站在开发者的角度上设想他们可能会提出的问题，并准备好问题的答案。在大多数情况下，对于提升研发效率的事，管理层会表示支持，你也可以借机提出一些要求，比如将迁移工作纳入到季度的 OKR 或绩效考核的内容里，这会大大减小在实施迁移的过程中人为因素的阻碍。

迁移原则

在正式进行迁移时，涉及到的范围会非常广，例如涉及到不同团队的持续部署习惯、不同的工具链，不同的小组的技术氛围、技术水平和面临的业务也可能有很大的差异。这里我提供几点迁移的原则供你参考，遵循这些原则有利于减少迁移过程中的阻力。它们分别是：

- 尽量提供组织保障
- 工作流最小变更原则
- 合理利用已有的基础设施

尽量提供组织保障

提供组织保障是迁移过程中需要遵循的第一原则，也就是说，在迁移过程中要尽量将实施 GitOps 的小组独立出来，为迁移过程提供人力保障。

不同团队在不同的时期，根据业务情况可能会有不同的组织架构形式，在产品研发团队的组织架构中，有两种常见的组织架构：

- **集中式运维管理**
- **分散式小组高内聚和闭环**

集中式运维管理的组织形式通常是由 N 个业务研发团队加 1 个运维团队组成。运维团队负责所有生产环境所需的资料，例如运行环境、基础设施和集群网络等，对这些资料所有的变更都由该运维团队负责。

在这种组织架构形式下，运维团队对业务研发小组负责，并制定相关的标准。要实施 GitOps 相当于需要从他们手里拆分一些职权出来，这是有挑战的。

由于该团队的特殊和专业性，在实施迁移的过程往往也需要他们的配合，所以在迁移过程中他们可能会提出一些非产品层面的问题，例如网络、业务流程和安全等问题，值得注意的是，这并不是负责实施 GitOps 的团队所能解决的问题，我们需要把双方共同目标聚焦在迁移带来的效率提升和流程优化上。

简而言之，在这种组织架构模式下，实施 GitOps 的团队首先需要面临来自运维团队的挑战。

第二种常见的组织架构形式是：分散式小组高内聚和闭环。

这种组织架构常见于中大型的技术团队，他们崇尚敏捷文化并实施了 DevOps，小组内部由研发工程师兼任运维角色，他们负责整合工具链并提供自动化部署流程。在这种组织形式下，实施 GitOps 的关键是每个小组里负责整合工具链以及实施持续部署的工程师。

这意味着，负责实施 GitOps 的团队需要对每个小组进行落地，这是一个巨大的挑战，难点在于：不同的小组研发流程可能差异较大，没有 100% 适用于每个团队的迁移方案，有些小组的迁移方案甚至需要单独定制。

那么如何解决这个问题呢？我认为**可以采用 “各个击破”的方式。**

首先寻找有痛点、感兴趣并且愿意做出流程改变的 1 至 2 个团队，对他们进行深入调研后定制迁移方案和计划，并在迁移过程中提供 100% 的支持和持续跟进，最终完成他们的迁移目标。

迁移完成后，该团队便树立了一个优秀案例，我们也积攒了好的口碑。下一步，寻找机会在团队内继续宣讲客户案例和收益，并主动寻找其他团队的痛点。逐步覆盖，最终完成所有迁移小组的目标。迁移过程可能很漫长，短期内很难达到 100% 的全覆盖，这是正常的，团队要保持足够的耐心和决心。

总之，无论是哪种组织架构模式，最重要的都是“提供组织保障”。也就是说，在迁移过程要单独成立实施 GitOps 的小组或者将职责纳入到已有的基础架构部门，为迁移提供人力资源保障以及合法性。

最后，迁移过程并不是一蹴而就的。在迁移之前，应当制定详细的目标和实施计划，逐步覆盖，最终完成迁移任务。

工作流最小变更原则

工作流最小变更的原则指的是在部分环节尽可能复用之前的流程。

比如，在实施 GitOps 之前，在团队内很常见的实践是通过 Jenkins 来执行 CI/CD 流水线。在这种情况下，工作流最小变更的原则就显得很重要了。由于在实施 GitOps 工作流时涉及非常多的工具链以及上下游的连接，为了能够平滑迁移，你可以设计两期的迁移过程，在第一期迁移时，仍然使用 Jenkins 作为 CI 工具，但把持续部署流程拆分到 GitOps 工作流。在进行第二期迁移时，把 Jenkins 替换为其他的云原生构建工具。

这种迁移原则，实际上是在照顾一些短时间内难以改变习惯的开发者，在迁移的早期阶段，开发者仍然部分使用原来的技术栈和工具是一个很好的开始，对于成熟的业务和团队，我推荐你通过这种方式来稳步推进迁移。

不过，在评估是否应当遵循这个原则的时候，你需要综合考虑团队的现状，如果团队目前没有成熟的工具链和使用习惯，并且正面临一系列的发布问题，这时候应该全量迁移到 GitOps 工作流中。

合理利用已有的基础设施

在一些中大型的团队中，生产环境的资源一般会由运维团队专门管理，而在将环境迁移到 GitOps 的过程中，往往需要资源来安装 GitOps 的工具链，这时我们同样需要考虑运维团队的诉求。

在大部分情况下，为了减小运维团队对基础设施的维护成本以及资源开支，你可以考虑复用已有的基础设施，例如 Redis、MySQL 和 Kubernetes 集群等。当然，这也不是绝对的，有一些运维团队认为故障隔离问题优先于维护成本和开支问题，他们宁愿使用新资源来承载新的业务，这也是一个选择。

GitOps 工具链和已有的基础设施整合度越高，其在团队内的不可替代性就越高，这对巩固 GitOps 的推广成果是很有帮助的。

总结

好了，以上是我在不同规模的公司从零到一实施新技术的一些经验。这些道理并不高深，只是由于组织和个人存在特殊性，因此在落地 GitoOps 的过程中，情况容易变得复杂。其实在任何时候，你只需要记住一点：**在公司体系下，切勿凭借满腔热血单打独斗。**你需要在实施迁移时找到与你有相同利益的管理者，并争取他的必要支持，配合同级其他业务组的团队成员，一起打配合“搓麻将”。

特别是在中大型公司，新技术的落地，往往意味着打破常规和利益重新分配的过程，如果没有至上到下的支持和推动，仅仅凭借自己的热情是很难实现的。如果执意单打独斗，最终很容易落得“壮志难酬”和“此处不留爷，自有留爷处”的结果，受伤的终究还是自己。

说了这么多，我们并不是鼓励“办公室政治”，而是需要你多个心眼并留意“办公室政治”，技术人往往都比较单纯，我们需要避免自己成为“办公室政治”的牺牲品。

到此，我们的《云原生架构和 GitOps 实战》也就告一段落了。课程结束并非终点，我们还可以在留言区互动交流，也祝你享受成长，学有所成。

分享给需要的人，Ta购买本课程，你将得 18 元

生成海报并分享

赞 1　提建议

上一篇　40丨命令式和声明式，谁才是驱动云原生的“引擎”？

下一篇　期末测试 | 来赴一场满分之约吧～

精选留言 (3)

写留言

Amos

2023-03-13 来自广东

第一个读完的，不得不说老师该系列的文章内容真的很值得学习。目前各云厂商对 gitops 支持并不多，而原生 argocd 从 web 页面 、私有 git 仓库、企业微信&钉钉通知集成等方面并不适用所有企业。因此如果课程里能有一些二次开发的文章就更加完美了

作者回复: 感谢支持和认可！
很好的建议，谢谢，同时希望你能有所收获，在职业发展的路上更上一层楼。

1

橙汁
2023-03-30 来自北京

课很值得，学完换了份工作用的就是gitops形式 牛逼

作者回复: 能坚持学完真的很棒！很高兴能为你助力新的职业发展！

黑鹰
2023-03-16 来自北京

老师，能从你的落地经验上，介绍下GitOps和DevOps的核心区别吗？

作者回复: GitOps 其实是实现 DevOps 的一种路径，他们并不冲突。只是 DevOps 的工具链已经很难满足云原生场景下的需求了，所以需要革新工具链。
本质上他们都实现了运维和发布能力的左移，使开发者能自助处理发布流程，打破部门间的信息墙，提高开发和发布效率。

1