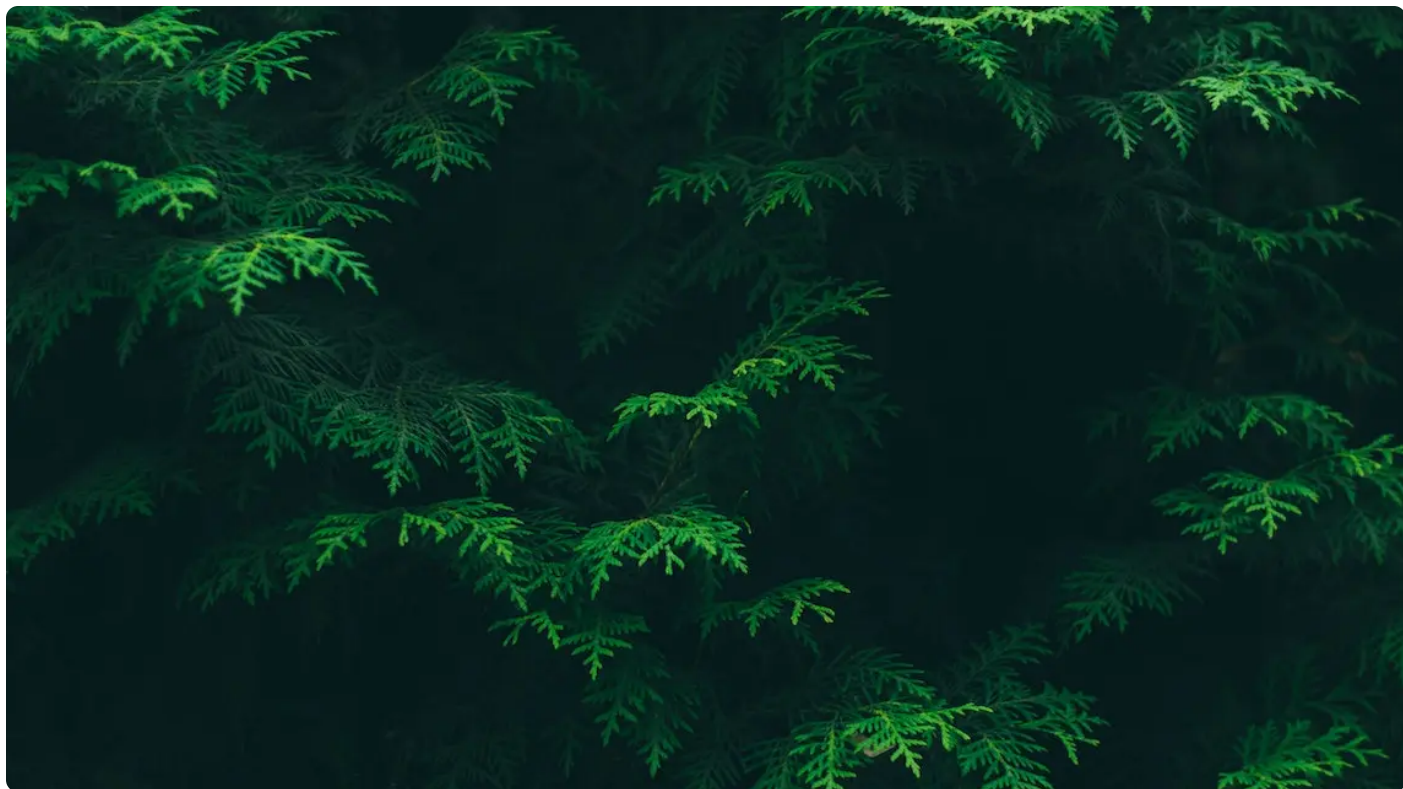


## 39 | GitOps 最佳实践，ArgoCD 凭什么脱颖而出？

2023-03-08 王伟 来自北京

《云原生架构与GitOps实战》

课程介绍 >



讲述：王伟

时长 12:21 大小 16.93M



你好，我是王伟。

2022 年 12 月，CNCF 宣布 Argo 项目从孵化阶段提升为毕业阶段。这意味着它和 Kubernetes、Prometheus 这些影响力巨大的项目一样，加入了毕业项目的行列。

你可能不知道，Argo 项目其实包括多个子项目，ArgoCD 是其中关注度最高且终端用户最多的项目。

CNCF 的统计数据显示，Argo 至少被 350 家企业用在了生产环境上，在社区贡献方面，有超过 2300 家公司和 8000 名个人为这个项目做出过贡献，Argo 项目在 GitHub 上拥有超过 20000 个 Star，它是 CNCF 开源社区中最活跃和最多样化的开源社区之一。

那么，为什么 ArgoCD 如此成功？它相比 GitOps 的鼻祖 FluxCD 有哪些优势呢？为什么 ArgoCD 会成为 GitOps 最受开发者欢迎的项目？

在这节课，我将从 ArgoCD 的历史说起，带你了解 ArgoCD 的发展历史，了解 ArgoCD 背后庞大的 Argo 项目。此外，我还会对 ArgoCD 和 FluxCD 做简单的对比，让你了解它们之间的差异，为你的技术选型提供参考。

## ArgoCD 的诞生

2017 年，Applatix 公司正式对外开源了 Argo 项目。2018 年，Applatix 公司被著名的 Intuit 公司收购。Intuit 公司成立于 1983 年，是一家老牌的金融和税务软件开发商。在收购 Argo 项目之前，Intuit 公司采用了 Netflix 开源的 CD 工具 Spinnaker。

随着业务的逐渐发展，Spinnaker 已经很难满足 Intuit 公司的需要了。所以，在收购 Applatix 公司之后，Intuit 要求其团队开发一个新的自助式的交付平台，并且需要用创新的方式来提高发布效率，降低复杂性。

在当时，Intuit 实际上已经使用了 Argo Workflow 项目了。但在构建这个全新的交付平台的时候，他们意识到在持续交付环节还缺少一款重要的产品，ArgoCD 在这一背景下就应运而生了。

由于在这之前 Intuit 公司采用了大量的开源项目，所以它们认为回馈开源社区是非常有必要的。因此，他们也将 ArgoCD 回馈给了开源社区。

在开源之后，ArgoCD 得到了开源社区极大的帮助，几年后，ArgoCD 成为了最火的 GitOps 工具。

## ArgoCD 的特点

ArgoCD 之所以能够在众多 CD 工具中脱颖而出，主要是因为它的特点突出。接下来，我就简单介绍一下 ArgoCD 的几个重要的特点，它们包括：

1. 支持多种应用标准
2. 开发者友好的 Dashboard
3. 支持多租户
4. 支持多集群
5. 漂移检测

6. 支持垃圾回收

7. 其他特性

## 支持多种应用标准

ArgoCD 几乎支持社区所有的应用封装格式，例如：

1. Kustomize

2. Helm

3. Ksonnet

4. YAML/JSON Manifest

不管你的 Kubernetes 应用是以哪种格式封装的，只要存储在 Git 仓库中，ArgoCD 都能够将它们作为应用导入。并通过对应的工具渲染成标准的 YAML Manifest，然后应用到集群内。

## 开发者友好的 Dashboard

ArgoCD 内置了开发者友好的 Dashboard，它可以对项目、应用和用户进行管理。通过 Dashboard，你可以很方便地创建一个应用，看到应用部署的状态。当集群资源和 Git 仓库的资源不一致的时候，Dashboard 还会发出提示。

此外，你也可以在 Dashboard 进行手动的刷新和同步应用操作。一旦应用被创建，Dashboard 还会提供应用的拓扑图，你可以非常方便地查看应用包含哪些资源，了解大致的流量拓扑等。

## 支持多租户

单个 ArgoCD 实例可以处理不同团队下不同的业务应用。它内置了项目的概念，你可以在某个项目下创建应用，并将它们映射到实际的团队，通过 ArgoCD Dashboard，特定的团队成员只能看到分配给他们的项目和应用程序，这种管理模型和 Kubernetes 命名空间与资源的关系非常类似。

借助多租户功能，我们可以很方便地用一个 ArgoCD 实例来管理组织中的多个团队和应用，实现多租户和隔离性。

## 支持多集群

ArgoCD 可以在它所运行的 Kubernetes 集群上同步应用，同时也可以管理外部集群。

其他集群的 API Server 的凭证会作为 Secret 对象存储在 ArgoCD 命名空间中，你可以很轻松地在 Dashboard 里管理多集群。在部署应用时，你可以选择不同集群进行部署，ArgoCD 内置的 RBAC 机制还可以控制用户对不同环境的访问权限。

多集群能力在大型的组织和应用下非常重要，在这种情况下，业务应用往往被部署在了不同的集群，如何更方便地管理多集群是一项很有挑战的工作。

## 可配置漂移检测

当集群的管理员在不通过 GitOps 工作流程（也就是将变更提交到 Git）的情况下更改资源的时候，Kubernetes 资源可能会和 Git 仓库存储的资源出现漂移。这是 GitOps 中的一个很常见的问题，但 ArgoCD 可以自动检测这些配置漂移，还可以自动将资源恢复为 Git 仓库里定义的状态。

不过，在 ArgoCD 控制台的应用配置中，你可以设置是否执行自动恢复策略。

## 支持垃圾回收

当部署的对象被从 Git 仓库中删除时，如果你使用 `kubectl apply` 重新部署，`kubectl` 并不会帮助你删除集群已存在的对象，但 ArgoCD 可以很好地解决这个问题。

## 其他特性

除了上面提到的这些特点，ArgoCD 还提供了下面这些功能。

- 可选手动或自动将应用程序部署到 Kubernetes 集群。
- 可以自动同步 Git 仓库的变更。
- 拥有 Web 用户界面和命令行界面（CLI）。
- 可视化部署并检测和修复错误。
- 拥有基于角色的访问控制（RBAC）。
- 支持使用 GitLab、GitHub、OAuth2 和 SAML 2.0 等服务的单点登录（SSO）。

- 支持 GitLab、GitHub 和 BitBucket 的 Webhook。

## ArgoCD 和 FluxCD 对比

ArgoCD 和 FluxCD 各有自己的特点。首先，FluxCD 整体架构比较简单、轻量，也比较容易维护。不过，FluxCD 并没有像 ArgoCD 一样提供 Dashboard，对于喜欢通过界面操作的开发者来说，ArgoCD 是更好的选择。

其次，在单实例的条件下，FluxCD 缺少多租户和多集群的支持。在中大型团队中，ArgoCD 是一个更好的选择。当然，FluxCD 多实例的部署方式虽然能间接实现多租户和多集群的支持，但在管理上也带来了更高的复杂度。

然后，在 GitOps 部署能力的支持上，两者并没有太大的差异。它们都只是部署工具，无法帮助你自动构建镜像，并且都符合 GitOps 所要求的几大原则。


最后，如果你希望通过二次开发的方法构建自己的 GitOps 工具，同时根据自己的组织实现多租户，那么 FluxCD 可能是更好的选择。

## Argo 生态

除了 ArgoCD 自身的优势以外，Argo 丰富的生态也是 ArgoCD 能够脱颖而出的一个重要因素。ArgoCD 可以结合 Argo 生态内的其他工具一起使用，这就极大扩展了 ArgoCD 的能力。

接下来，我就简单介绍一下 Argo 生态的其他项目。

### Argo Workflows

 **Argo Workflows** 是一个开源工作流编排引擎，用于在 Kubernetes 上编排并行任务。它通过在容器里运行工作流定义的每一个步骤，并将工作流构建成 DAG 来控制任务的依赖关系，特别适合调度密集型的计算任务。

当然，你也完全可以通过 Argo Workflows 来实现 CI/CD 流水线。

### Argo Rollouts

在之前的课程中，我们已经介绍了如何使用  **Argo Rollouts** 来实现蓝绿发布、灰度发布以及自动金丝雀发布。Argo Rollouts 提供了一组 Kubernetes 控制器和 CRD，让渐进式交付变得

非常简单。

此外，Argo Rollouts 可以与 Ingress 和服务网格集成，并结合它们的流量控制功能在更新期间逐步将流量切换到新版本。它还可以在发布过程中查询指标，以此验证部署结果并执行升级或回滚。

## Argo Event

🔗 **Argo Event** 是一个事件管理的扩展工具。在 Kubernetes 环境中，通常我们需要管理来自不同系统的事件，并针对事件编写相关的业务逻辑。Argo Events 提供了一个可扩展的解决方案，支持自定义事件监听器。

Argo Event 支持多达 20 多个不同的事件源（例如 webhook、S3、cron 和消息队列等），并支持 10 多种触发器。通过触发器，通过 Argo Event，你可以将事件源和触发器连接起来，比如，在接收到触发器之后，创建 Kubernetes 对象。

Argo Events 一般不独立工作，它需要和其他项目集成，比如 Argo Workflows 和 Argo CD。

## Argo Autopilot

对于新手来说，从零开始配置 ArgoCD 和建立完整的 GitOps 工作流可能存在一定难度。

🔗 **Autopilot** 项目是专门面向 GitOps 新手的 CLI 工具，它可以一键帮你配置好 Git 仓库并安装 ArgoCD，同时在 ArgoCD 里配置好示例应用。

## Argo Image updater

在之前的课程中，我介绍了如何使用 🔗 **Argo Image Updater** 来监听镜像版本的修改，自动触发 GitOps 工作流。

Image updater 的主要功能包括下面几个。

- 更新 Helm 或 Kustomize 创建的 Argo CD 应用的镜像版本。
- 支持多种更新策略来更新应用镜像。
- 支持大部分容器镜像仓库以及私有镜像仓库。
- 支持将镜像版本回写到 Git。

- 能够根据规则过滤特定的镜像版本。

## Argo ApplicationSet

在 [🔗 第 27 讲](#) 中，我们的多环境管理就是通过 ApplicationSet 来实现的。

普通的 Argo CD 应用只能从单个 Git 仓库部署到单个目标集群或者命名空间中，ApplicationSet 则能够提供更高级的功能，它提供的模板可以一次创建多个 Argo CD 应用。ApplicationSet 在设计上和 Application 是解耦的，ApplicationSet 可以通过规则来生成多个 Application CRD 对象，以此来实现多应用的方式。

ApplicationSet 主要提供以下两个功能。

- 一次将多个 ArgoCD 应用部署到一个或多个 Kubernetes 集群。
- 一次从单个 Git 仓库中以模板的方式部署多个 ArgoCD 应用。

## ArgoCD Operator

[🔗 ArgoCD Operator](#) 可以管理 Argo CD 及其所需组件。它的目标是自动配置 ArgoCD，例如升级、备份、恢复以及安装。它的另一个很大的优点是可以自动配置 Prometheus 和 Grafana，为 ArgoCD 提供可观察性。

该项目现在仍然在积极开发中，它主要提供以下功能。

- 使用默认配置安装 ArgoCD。
- 支持 ArgoCD 升级。
- 支持备份和恢复 ArgoCD。
- 支持配置 Prometheus 和 Grafana。
- 支持为 ArgoCD 配置 HPA。

## Argo Vault 插件

[🔗 ArgoCD Vault 插件](#) 用于从各种密钥管理工具中提取密钥并将它们注入到 Kubernetes 集群中，它支持丰富的第三方密钥管理工具例如 HashiCorp Vault、IBM Cloud Secrets Manager



以及 AWS Secrets Manager。

Argo Vault 插件旨在解决 GitOps 和 Argo CD 的秘密管理问题。它希望通过一种不需要 CRD 和 Operator 的方法来管理密钥，这个插件不仅可以用来管理密钥，还可以用来部署 configMaps 或任何其他 Kubernetes 资源。

在 [第 29 讲](#) 中，我们提到了如何使用 Sealed-Secrets 来管理密钥，这是一种需要 Operator 和 CRD 对象的密钥管理方式。如果你在 GitOps 工作流中使用了外部的密钥管理工具，那么在使用 ArgoCD 实施 GitOps 工作流时，你就需要用到 Argo Vault 插件将这些密钥提取出来并应用到 Kubernetes 集群内。

## 总结

总结一下，这节课，我带你简单学习了 ArgoCD 诞生的历史、特点，它和 FluxCD 的对比以及 Argo 生态。

ArgoCD 能在众多 CD 工具中脱颖而出，除了 GitOps 大背景的推动以外，其自身也具备非常多优秀的特性。实际上，通过了解众多优秀的开源项目，我们会发现一个共同点，那就是，最初开源这款产品背后的公司，它使用项目的规模很大程度上决定了项目的技术视野和发展。

无论是 Kubernetes 还是 ArgoCD，我们会发现其背后的公司在产品开源之前就已经在团队内部进行了大规模的使用，产品实际上已经考虑到了在不同规模的团队下的痛点和诉求，所以，这些产品在开源的时候就已经相对完善了，产品体验和文档也比较完善。

在建立 GitOps 工作流时，通常我们有两种技术选型，分别是 FluxCD 和 ArgoCD。它们有不同的特点，你可以根据团队需要进行选型。不过，对于一般的业务团队来说，如果想给团队带来自助式体验的发布过程，ArgoCD 可能是更好的选择。

最后，Argo 丰富的生态系统也是 ArgoCD 能够脱颖而出的一个重要原因。实际上，Argo 生态的项目能解决 GitOps 工作流中的大部分问题，例如和外部系统的对接、持续集成和持续部署等。



上一篇 38 | GitOps 为什么成为云原生交付的事实标准？

下一篇 40 | 命令式和声明式，谁才是驱动云原生的“引擎”？

## 更多课程推荐

### 李三红·搞定 Java 开发基础

极客时间 × 阿里云开发者社区联合出品

李三红  
阿里云程序语言  
与编译器技术总监  
Java Champion

免费订阅



## 精选留言 (3)

💬 写留言



jarvan\_1995

2023-03-09 来自上海

老师我们的Argocd接入了Idap之后能登录，但是鉴权有问题，按照官网的例子配置了也不管用，请问是什么问题呢

作者回复: 是用 dex 集成的吗？

共 2 条评论 >



黑鹰

2023-03-08 来自北京

这个专栏咱们有微信交流群吗？实践部分遇到问题，在留言区不能发截图，不太方便表达和交流。

作者回复: 暂时没有哦，可以把代码贴到留言区。

共 2 条评论 >



黑鹰

2023-03-08 来自北京

ArgoCD确实挺强大的，它提供的命令行工具可以和Tekton这类CI工具很好的集成，实现CI/CD 全流程



👍 1