

ENTREGA CONTÍNUA EM CLUSTERS KUBERNETES UTILIZANDO GITLAB E ARGO CD NA DTI/UFGM



<https://doi.org/10.22533/at.ed.7951725090512>

Data de aceite: 15/07/2025

Wallison Fabiano de Araújo

Diretoria de Tecnologia da Informação –
Universidade Federal de Minas Gerais
(UFGM)
Belo Horizonte – MG – Brasil

Gessy Caetano Junior

Diretoria de Tecnologia da Informação –
Universidade Federal de Minas Gerais
(UFGM)
Belo Horizonte – MG – Brasil

RESUM:. Este trabalho descreve o processo de implantação de Entrega Contínua no cluster Kubernetes utilizando Gitlab e Argo CD pela Divisão de Infraestrutura e Serviços da UFGM. Neste artigo são apresentadas as etapas dessa implantação, as ferramentas e os softwares utilizados para realizar operações de Entrega Contínua. Demostramos como o Argo CD aplica as mudanças no cluster Kubernetes para a aplicação de teste aplic01.

PALAVRAS-CHAVE: *Entrega Contínua. Kubernetes. ArgoCD. Gitlab.*

INTRODUÇÃO

A prática de desenvolvimento de *software* se torna mais eficiente quando há implementação de integração de código por meio de *builds* e *deploy* automatizados. Com a abordagem CI/CD (*Continuous Integration/Continuous Delivery*) é possível entregar aplicações com maior frequência aos clientes. A Integração Contínua (CI) é um conjunto de práticas em que as equipes entregam seu trabalho com muita frequência, geralmente ao menos uma vez ao dia (HUDSON D. C ANGELO, 2019). A Entrega Contínua (CD) se refere ao lançamento automático das mudanças feitas por uma pessoa desenvolvedora, do repositório à produção. Com isso, as equipes de operação não ficam sobrecarregadas com processos manuais que atrasam a entrega de *apps* (Gabriel Faraday, 2019).

Para melhor entendimento deste trabalho, vale destacar que devido a crescente necessidade de modernização do ambiente de TI da UFGM é utilizada

virtualização para melhor execução dos serviços. A Diretoria de Tecnologia da Informação (DTI) da Universidade Federal de Minas Gerais vem melhorando a infraestrutura de seus serviços com a implantação de Kubernetes para gerenciamento de cargas de trabalho e serviços distribuídos em contêineres, pois a utilização de contêiner fornece uma confiabilidade de melhoria na infraestrutura de TI (Pereira Ferreira and Sinnott, 2019).

O Kubernetes possibilita o processo de CI/CD por ser uma plataforma para executar aplicativos usando o modelo declarativo. Nesse modelo, você descreve o estado pretendido dos seus aplicativos e o Kubernetes tenta atingi-lo automaticamente (Google Developers, 2023).

A Divisão de Infraestrutura e Serviços da DTI propôs um projeto-piloto para avaliar a Entrega Contínua de aplicativos utilizando o Gitlab e o Argo CD em cluster Kubernetes. Neste artigo será demonstrada a etapa Entrega Contínua de uma aplicação denominada *aplic01*. Como resultados, espera-se que, com a abordagem CI/CD, haja maior rapidez no processo de publicação de aplicações no Cluster Kubernetes com a entrega de aplicações com mais frequência aos clientes.

MÉTODOS

Esta Seção está dividida da seguinte forma: (1) Estudo das tecnologias; (2) Descrição da arquitetura para implantação.

Estudos das Tecnologias

Inicialmente, foi realizado um estudo das tecnologias GitOps para implementação de CI/CD no Cluster Kubernetes. Como repositório *git*, adotamos o Gitlab. O Gitlab CI é uma aplicação *web* com uma API que armazena o seu estado em um banco de dados. Ele gerencia projetos e *builds*, fornecendo uma interface *web* agradável (Gitlab, 2023). O Gitlab possui número ilimitado de repositórios com Integração Contínua integrada. Porém, o recurso de GitOps no Gitlab é disponível somente na versão *Premium e Ultimate Edition*, portanto, não foi utilizado neste projeto.

No mercado, existem diversas ferramentas que implementam a abordagem que seja semelhante ao GitOps no Kubernetes, nesse estudo utilizamos o Argo CD. O Argo CD atua como uma ferramenta de sincronização dentro de um fluxo GitOps. Ele é implementado como um controlador, que monitora continuamente as definições e configurações da infraestrutura definida em um repositório *git*, comparando o estado especificado dessas configurações, com seu estado ativo no cluster Kubernetes (Aditya Soni, 2020).

Descrição da arquitetura para implantação

A arquitetura utilizada é composta por um cluster Kubernetes instalado na versão *v1.24*, uma máquina virtual com Gitlab instalado na versão *15.8* e o Argo CD, versão *2.5.8* instalado como *pod* dentro do cluster Kubernetes. Com isso, o Argo CD já possui acesso para realizar *deploy* de aplicações dentro do cluster Kubernetes.

A configuração de instalação do Argo CD foi realizada dentro do repositório de configuração do próprio cluster Kubernetes e está armazenada em um repositório de configuração do cluster, mantendo infraestrutura como código. O Argo CD suporta diversos mecanismos de autenticação, para o nosso caso escolhemos utilizar o *OIDC (OpenID Connect)* para integrar todas as ferramentas utilizando *SSO (Single Sign-on)* institucional para acesso as aplicações.

RESULTADOS E ANÁLISES

Como resultados, apresentaremos o processo de configuração e a publicação de uma aplicação denominada *aplic01*. Esse artigo não abordará a funcionalidade da aplicação, mas o processo de como foi realizada a entrega contínua.

Configurações para a aplicação *aplic01* no Gitlab

Criou-se um grupo no Gitlab, depois foi criado o projeto sendo disponibilizado o endereço do repositório *git* para acesso. A integração do repositório da aplicação com o Argo CD foi feita utilizando uma chave *Deploy Token* criada para o grupo. Em seguida foram configurados no Argo CD o projeto (*default*) e o endereço de acesso ao repositório, utilizando-se a chave *Deploy Token* criada no GitLab, codificada em *base64*. O próximo passo foi configurar a aplicação com o endereço do repositório *git* da aplicação, e as opções de *syncOptions* (*prune=true*, *PruneLast=true*, *CreateNamespace=false*, *PrunePropagationPolicy=foreground*) e automação (*limit=5*, *duration=5s*, *fator=2*, *maxDuration:3m*). Finalizou-se criando os manifestos da aplicação *aplic01*, com isso o Argo CD já está configurado para realizar a Entrega Contínua da aplicação no cluster Kubernetes. A Figura 1 apresenta as etapas de configuração de aplicações entre o Gitlab e o Argo CD no Kubernetes.

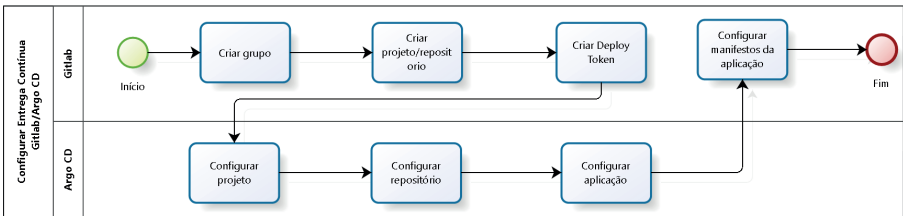


Figura 1 – Etapas para configuração da entrega contínua da aplicação.

Deploy automático de manifestos de aplicações

Com a Entrega Contínua configurada, o Argo CD ficará comparando o estado especificado nos manifestos da aplicação com o seu estado ativo no cluster Kubernetes. Caso o estado ativo não estiver de acordo com o que foi especificado nos manifestos, o Argo CD classificará a aplicação como *OutOfSync*. Neste caso ele reconfigurará automaticamente, o ambiente da aplicação no Kubernetes, a fim de que ele possa refletir o que foi especificado nos manifestos (Figura 2).

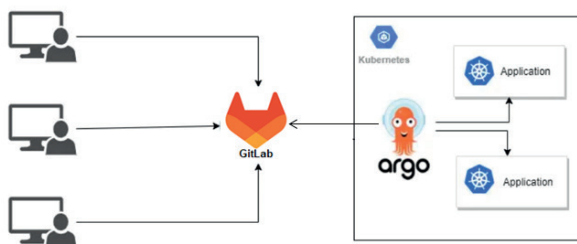


Figura 2 – Monitoramento do Argo CD comparando o estado especificado do repositório **git** do Gitlab com seu estado ativo

O processo de publicação da aplicação ocorrerá da seguinte forma: após criar ou alterar os manifestos da aplicação e realizado o *commit* no Gitlab, o Argo CD identificará que ocorreu alguma alteração e realizará a entrega contínua dela para o cluster Kubernetes. O Argo CD ainda permite que seja feita a análise dos logs da aplicação na própria interface de gerenciamento e caso ocorra algum erro é possível voltar à versão anterior deixando a aplicação disponível (Figura 3).

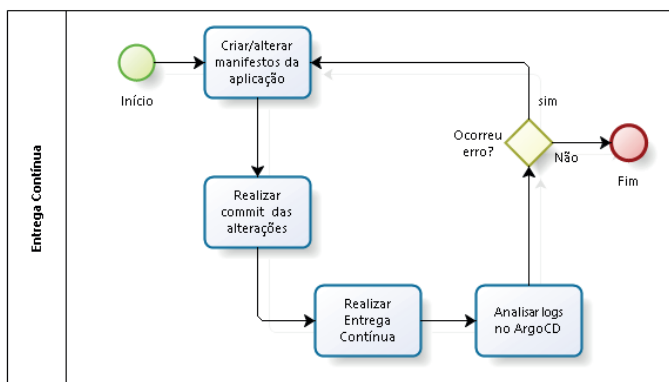


Figura 3 – Etapas da Entrega Contínua após **commit** de manifestos.

A figura 4 apresenta a aplicação *aplic01* sendo publicada. Percebe-se que na primeira imagem, a aplicação *aplic01* está no estado de *Progressing e Syncing*. Após a publicação, o estado muda para *Healthy e Sync OK*, isso mostra que a aplicação foi publicada com sucesso e está de acordo com o que foi configurado em seus manifestos.

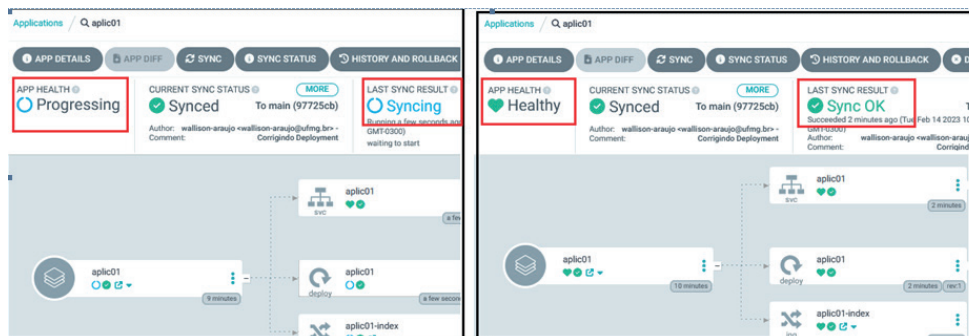


Figura 4 – Tela do Argo CD publicando a aplicação **aplic01**.

CONCLUSÃO E TRABALHOS FUTUROS

Constatou-se que o processo foi bastante eficiente. Com testes realizados a solução mostrou-se funcionar corretamente. O processo de publicação de aplicações, que atualmente se dá apenas pela equipe de Infraestrutura, poderá ser realizado pela equipe de desenvolvimento através da Entrega Contínua, sendo possível, publicação de aplicações, análise de logs, interação com *pods* e restauração rápida da versão anterior em caso de falhas. Entretanto, a criação de armazenamento, com a criação de PV (*persistente volumes*) no Cluster Kubernetes continuará a ser feita pela equipe de Infraestrutura. A utilização de conceito de projetos tanto no Argo CD quanto no Gitlab permite uma melhor administração e controle de acesso de aplicações e usuários. Atualmente o cluster Kubernetes está em fase de testes de homologação, sendo necessárias configurações de outras ferramentas antes ser disponibilizado em produção.

O GitOps com Kubernetes leva a infraestrutura como código a um outro nível. O estilo declarativo e o modelo de reconciliação do Kubernetes proporcionam um ambiente extremamente favorável para a sincronização e automação de infraestruturas através do *git*. O Argo CD é uma excelente ferramenta para esse trabalho podendo ser usado para gerenciar várias aplicações em diferentes clusters Kubernetes a partir de um único ponto.

Programadores de código ou operadores podem revisar rapidamente quaisquer alterações de aplicativo ou configuração no cluster Kubernetes, tudo a partir da interação com as ferramentas, além de garantir que as implantações no Kubernetes sejam auditáveis.

Para trabalhos futuros, sugere-se que antes desse processo ser disponibilizado em produção, deve-se melhorar a maturidade na equipe de desenvolvimento e melhorar a cobertura de testes, caso contrário pode-se enviar um erro drástico direto para a produção.

REFERÊNCIAS

Aditya Soni. (2020). Disponível em: <<https://blog.searce.com/argocd-gitops-continuous-delivery-approach-on-google-kubernetes-engine-2a6b3f6813c0>>. Acessado em: 10/02/2023.

Gitlab. (2023). What is GitOps? Disponível em: <<https://about.gitlab.com/topics/gitops/>>. Acessado em 05/02/2023.

Gabriel Faraday. (2019). O que é CI/CD? Onde eu uso isso? Disponível em: <<https://gabriel-faraday.medium.com/o-que-%C3%A9-ci-cd-onde-eu-uso-isso-57e9b8ad8c73>>. Acessado em: 13/02/2023.

Google Developers. (2023). Kubernetes e os desafios da entrega contínua de software. Disponível em: <<https://cloud.google.com/architecture/addressing-continuous-delivery-challenges-in-a-kubernetes-world?hl=pt-br>>. Acessado em 27/02/2023.

Hudson D. C. Ângelo. (2019). Integração Contínua e seus benefícios. Disponível em: <<https://meuartigo.brasilecola.uol.com.br/informatica/integracao-continua-seus-beneficios.htm>>. Acessado em: 14/02/2023.

Pereira Ferreira, A. and Sinnott, R. (2019). A performance evaluation of containers running on managed kubernetes services. In 2019 IEEE International Conference on Cloud Computing Technology and Science (Cloud Com), pages 199–208.