

## Programa de Bolsas - DevSecOps

### Projeto: GitOps na prática

Time Compass	Thiago Geremias De Oliveira Yurisay Josefina Escudero Guevara Davi Santos Cardoso Da Silva Patricia Maria Moura dos Santos
--------------	---

### GitOps na Prática

O desenvolvimento moderno de aplicações exige entregas rápidas, seguras e escaláveis. Empresas como Netflix e Nubank utilizam plataformas como o Kubernetes para orquestrar centenas (ou milhares) de containers de forma eficiente, automatizada e resiliente. Ao mesmo tempo, surgiu a necessidade de tornar os processos de deploy mais auditáveis, previsíveis e versionados, e é aí que entra o GitOps — uma prática que usa o Git como fonte de verdade para toda a infraestrutura e aplicações.

Aprender Kubernetes permite entender como aplicações são executadas em ambientes distribuídos, como escalar cargas, lidar com falhas e automatizar o ciclo de vida dos serviços. Já o GitOps, com ferramentas como o ArgoCD, permite fazer deploys de forma automatizada e segura apenas com um git push, trazendo mais controle e rastreabilidade para os times de desenvolvimento e operações.

Esse projeto propõe uma imersão prática nesse universo, usando o GitHub, Rancher Desktop com Kubernetes e ArgoCD para implantar um aplicativo de forma automatizada. Assim, vivenciarão na prática como as empresas modernas operam em ambientes cloud-native.

### Objetivo:

Executar um conjunto de microserviços (Online Boutique) em Kubernetes local usando Rancher Desktop, controlado por GitOps com ArgoCD, a partir de um repositório público no GitHub.

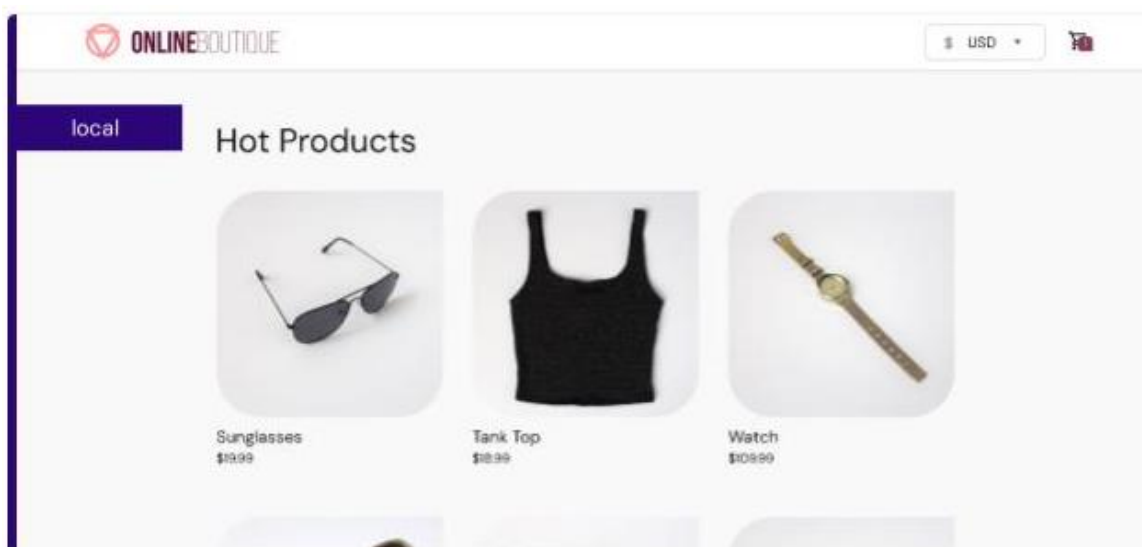


Figura 01: Site Online Boutique

[Link com os arquivos yaml](#)

## Etapas do Projeto

### Pré-requisitos:

- Rancher Desktop instalado (com Kubernetes habilitado);
- Kubectl configurado (kubectl get nodes funcionando);
- ArgoCD instalado no cluster;
- Conta no GitHub com repositório público;
- Git instalado;
- Docker funcionando localmente.

### Etapas 1 – Fork e repositório GitHub:

1. Fork do repositório oficial:
  - a. <https://github.com/GoogleCloudPlatform/microservices-demo>
2. Criar um repositório no GitHub com:
  - a. Apenas o arquivo release/kubernetes-manifests.yaml
  - b. Estrutura recomendada:

```
gitops-microservices/  
└─ k8s/  
    └─ online-boutique.yaml
```

## Etapa 2 - Instalar ArgoCD no cluster local

```
kubectl create namespace argocd  
kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml
```

## Etapa 3 – Acessar ArgoCD localmente

- Port-forward para acessar interface web.
- Acessar: <https://localhost:8080> com o usuário e senha padrões.

## Etapa 4 – Criar o App no ArgoCD

- Criar um vínculo com o repositório de código criado no github;
- Criar um App no ArgoCD;
- Sincronizá-lo para aplicar as mudanças no kubernetes local.

## Etapa 5 – Acessar o front-end

O frontend roda como um ClusterIP, então precisamos fazer port-forward.

## Entregas Esperadas

- ☒ Criar um repositório Git público com a estrutura de manifests YAML
- ☒ Fazer o deploy do ArgoCD corretamente
- ☒ Criar o App no ArgoCD apontando para o repositório Git
- ☒ Sincronizar a aplicação e garantir que os pods estejam rodando
- ☒ Acessar o frontend da aplicação via kubectl port-forward
- ☒ (Opcional) Customizar o manifest (ex: mudar número de réplicas de algum microserviço)