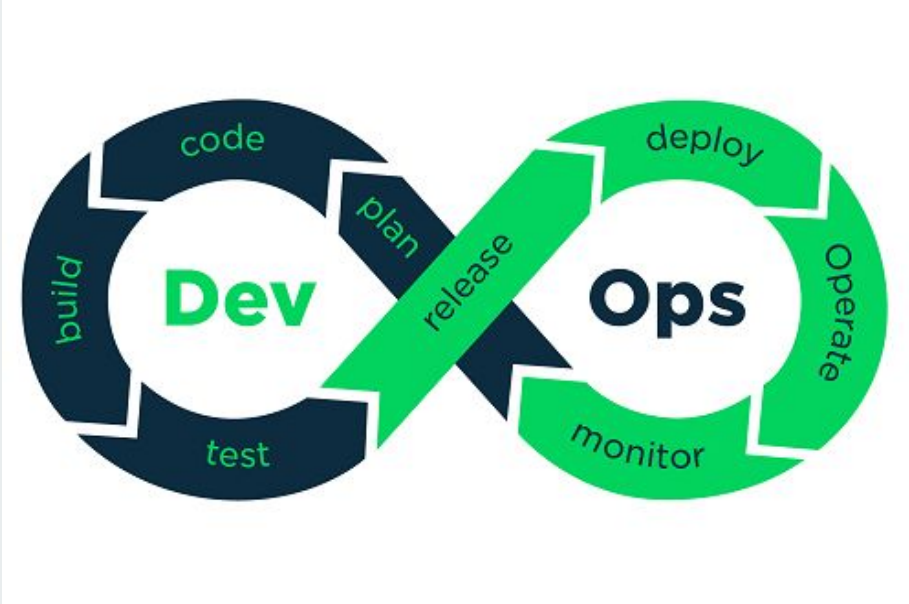
DevOps

Fundamentos de CI/CD

CSD

CSD (Continuous Software Development) é uma abordagem contínua de desenvolvimento de software que integra automação, testes e deploys frequentes.



CSD

Características:

- Integração Contínua (CI): Código integrado e testado automaticamente.
- Entrega Contínua (CD): Lançamento frequentes de confiaveis.
- Automação: Infraestrutura e processos automatizados.
- Feedback Contínuo: Monitoramento e melhorias constantes.

Benefícios:

- Maior rapidez na entrega de funcionalidades.
- Redução de erros e maior qualidade do software.
- Melhor adaptação a mudanças e necessidades do mercado.

O que é Integração Contínua (CI)?

A Integração Contínua é uma prática de desenvolvimento que envolve a automação do processo de construção e teste de código. O principal objetivo da CI é permitir que os desenvolvedores integrem suas alterações de código em um repositório compartilhado várias vezes ao dia. Cada integração é verificada por meio de testes automatizados, o que ajuda a detectar erros rapidamente.

O que é Integração Contínua (CI)?

Benefícios:

- **Detecção Precoce de Erros**: Com testes automatizados, os erros são identificados rapidamente, permitindo correções antes que se tornem problemas maiores.
- **Redução de Risco**: A integração frequente reduz o risco de conflitos de código, já que as alterações são integradas em pequenos incrementos.
- **Feedback Rápido**: Os desenvolvedores recebem feedback imediato sobre a qualidade do código, o que facilita a manutenção e a melhoria contínua.

O que é Entrega Contínua (CD)?

A Entrega Contínua é uma extensão da Integração Contínua, que se concentra na automação do processo de entrega de software. O objetivo da CD é garantir que o software esteja sempre em um estado que possa ser implantado em produção a qualquer momento. Isso significa que, após a integração e os testes, o código é automaticamente preparado para ser lançado.

O que é Entrega Contínua (CD)?

Benefícios:

- **Implantações Rápidas e Frequentes**: A CD permite que novas funcionalidades e correções sejam entregues aos usuários de forma rápida e eficiente.
- **Menos Erros em Produção**: Com um processo de entrega automatizado, a probabilidade de erros durante a implantação é reduzida.
- **Melhoria na Satisfação do Cliente**: A capacidade de lançar atualizações rapidamente melhora a experiência do usuário e a satisfação do cliente.

Relação entre CI x CD

A Integração Contínua e a Entrega Contínua estão interligadas e são partes essenciais de uma abordagem DevOps. Enquanto a CI se concentra na automação da construção e dos testes, a CD se concentra na automação da entrega. Juntas, essas práticas formam um ciclo contínuo que permite que as equipes de desenvolvimento entreguem software de alta qualidade de forma rápida e eficiente.

Ferramentas:

- Jenkins, Azure Pipelines, AWS CodePipelines, CodeShip, CodeCI, Github, Gitlab, etc...

Implementando CI/CD

Para implementar CI/CD Pipeline, devemos seguir algumas etapas:

1. **Configuração do Repositório de Código**: Utilize um sistema de controle de versão, como Git, para gerenciar o código-fonte.
2. **Automação do Build**: Configure um servidor de CI (como Jenkins, Travis CI ou CircleCI) para automatizar o processo de build.
3. **Testes Automatizados**: Desenvolva uma suíte de testes que cubra diferentes aspectos da aplicação, garantindo que o código funcione conforme o esperado.
4. **Implantação Automatizada**: Utilize ferramentas de automação de implantação, como Docker, Kubernetes ou AWS CodeDeploy, para facilitar a entrega contínua.
5. **Monitoramento e Feedback**: Implemente ferramentas de monitoramento para acompanhar o desempenho da aplicação em produção e colete feedback dos usuários.

Prática - Github Actions

Clonar repositório paeeglee/devops-api