

Documentação - Deploy Automático

C. Santos; G. Soares; J. Martimiano; J. Sousa; L.Gomes; M. Ferreira; K. Matsumoto.

RESUMO: Este trabalho tem como finalidade detalhar as ferramentas utilizadas para realização de um deploy automático.

PALAVRAS-CHAVE: Deploy; CI; DevOps; CD, Teste, Build.

1. INTRODUÇÃO

Através das ferramentas de DevOps aplicaremos na prática todos os requisitos mapeados necessários para finalização de um projeto. Dentro deles é necessário que seja configurado ambientes específicos de forma coerente e a prova de falhas, sendo assim, sempre disponível o acesso do projeto e para modificações de estrutura futuras.

2. Requisitos

Para esse projeto foram solicitados os seguintes requisitos:

- Integração contínua
- Deploy
- Versionamento de código
- Testes unitários
- Canal de comunicação
- Distribuição de tarefas

3. Ferramentas

Para atendermos os requisitos solicitados ao projeto, foram pesquisadas diversas ferramentas que atendiam as funcionalidades propostas. Após conversas e análises foram definidas as seguintes ferramentas.:

- Travis CI
- Servidor na DigitalOcean
- GitHub
- UnitTest2
- Nose2
- Selenium
- Slack
- Trello

4. Configurações Iniciais

4.1. GitHub

O primeiro passo foi a criação do projeto no GitHub com o nome de ForceLineProject, após isso foram adicionados colaboradores no projeto, com todos os colaboradores com acesso ao projeto foram criadas as branches para separação dos códigos: master para o código em produção e develop para o código em desenvolvimento.

4.2. Travis CI

Primeiramente foi criada a conta no Travis CI e foi vinculado essa conta ao projeto criado no GitHub apontando para as branches necessárias para a integração contínua. Dentro da branch do GitHub foi adicionado o arquivo travis.yml para a preparação de todo o ambiente sendo ele um arquivo de configuração de ambiente. É possível fazer essa preparação a cada commit e a cada merge de código entre a branch principal configurada pelo Travis CI.

4.3. Servidor na DigitalOcean

Utilizado para a realização do deploy da aplicação em flask, e para armazenar o banco de dados e os testes após o deploy em conjunto com a integração contínua do Travis.

A mudança foi feita pois anteriormente na utilização do Heroku para deploy era necessário a contratação de uma conta para que tivéssemos maior liberdade nas configurações do ambiente, não sendo possível que administrássemos as ferramentas.

4.4. Slack

O Slack está sendo utilizado como o principal meio de comunicação usufruindo de suas funções tais como: criação de canais divididos por assunto, integração entre GitHub, Slack, etc.

Foi criada uma workspace ForceLine para o projeto no qual foram adicionados os colaboradores do projeto e criado os seguintes canais: #general para comunicação entre os colaboradores, #travis onde a cada integração é enviada uma notificação com informações sobre o status de execução.

4.4. Trello

O Trello foi a ferramenta utilizado para a distribuição e controle de tarefas foi utilizado no estilo Kanban com as seguintes colunas: To Do, In Progress e Done; para que sim todos tenham conhecimento do andamento do projeto.

4.5. UnitTest2

Utilizamos o UnitTest2 para realizar o teste de unidade, verificamos se a informação contida na codificação da página é exibida ao acessarmos ela.

4.6. Nose2

Utilizado para automatização do teste de unidade. Ele executa o que está dentro da pasta teste como se fosse um módulo.

4.7. Selenium

O Selenium está sendo utilizado para realizar os testes de tela. No teste configurado para ser realizado ele identifica se a cor do elemento está de acordo com o definido.

4. Fluxo

Abaixo é mostrado como fica o fluxo para o deploy automático.

