

Podcast

Disciplina: Práticas da cultura DevOps no desenvolvimento de sistemas

Título do tema: Tecnologia de versionamento GIT e de integração continua

Autoria: Anderson Pereira de Lima Jerônimo

Leitura crítica: Stella Marys Dornelas Lamounier

Olá, ouvinte! No podcast de hoje vamos falar sobre Tecnologia de versionamento GIT e de integração continua. A necessidade de ter um código único para todos os membros da equipe de desenvolvimento é um desafio para quem não tem o costume de usar a tecnologia de versionamento. Imaginemos um seguinte cenário: criação de várias cópias do projeto, sendo cada uma disponibilizada para cada membro da equipe de desenvolvimento, e cada vez que um membro for oficializar uma nova versão, precisará integrar o seu código gerado com os demais, mas esse procedimento sem utilizar um sistema de versão, se torna uma tarefa árdua, pois, haverá mais tempo para analisar todo código e verificar se algum outro membro da equipe também possui alguma modificação que possa fazer a mesclagem do código para gerar essa nova versão do projeto.

Diante desse contexto, os sistemas de controle de versionamento, disponibilizados no mercado são ferramentas essenciais para as equipes de desenvolvimento, também como melhoria contínua do código fonte. Podemos destacar dentre as tecnologias de versionamento, a ferramenta GIT, comumente conhecida como sistema de versionamento, que facilita o trabalho das equipes de desenvolvimento realizarem todo processo de monitoramento, análise e modificações do projeto. Todas essas funcionalidades embarcadas no GIT auxiliam as equipes DevOps a terem um código único. Outra característica que podemos falar sobre o GIT é sua popularidade devido seu modelo distribuído, que permite que os projetos sejam compartilhados de forma mais rápida. Essa rapidez acontece devido todo processo ser feito através de repositórios locais, para armazenar seus códigos e gerenciá-los. Isso ajuda as equipes DevOps sempre desenvolver com versão mais atualizada, e o compartilhamento para servidor remoto. Como GIT trabalha de forma local, há alguns serviços Web que disponibilizam uma espécie de rede social de códigos fontes, podemos citar: GitLab, GitHub, Bitbucket e entre outros. Cada um com sua especificidade.

O GIT garante um gerenciamento mais rápido do que outras ferramentas de versionamento, as demais tecnologias de versionamento trabalham em modo cliente-servidor, nesse modelo requer uma comunicação constante com servidor, porque a máquina local não controla os estágios de modificações, e apenas o servidor fica monitorando, ou seja, esse modelo cliente-servidor caso tenha uma equipe grande de desenvolvimento pode haver uma grande demanda de requisições ao mesmo tempo e, uma comunicação em tempo real, cada vez que um arquivo seja adicionado, alterado ou removido do projeto.

No modelo distribuído, todo processo ocorre localmente, isso faz a diferença na comunicação quando for compartilhar com servidor remoto, pois ele só vai se comunicar com o servidor remoto quando precisar atualizar o repositório local, e depois de consolidar o versionamento localmente, isso garante menos requisições constantes no servidor remoto. O processo de instalação do GIT, é simples, disponível em diferentes plataformas de sistema operacionais, tais como: Linux, Windows e MacOS., e podemos baixar o executável e/ou arquivos para instalação através do site GIT. Após a instalação, caso deseje verificar se a instalação ocorreu corretamente podemos verificar através do comando “git – version”, que deve ser colocado no terminal no caso Linux e MacOS e no ambiente Windows no cmd. Caso o retorno do comando for a versão do GIT instalado, é porque deu tudo certo. Na ferramenta GIT, temos alguns estágios, o primeiro deles é o estado chamado de **modificado**, ele indica se um arquivo foi adicionado, alterado ou removido em seu conteúdo. Outro estado é conhecido como **preparado**, esse é usado no momento que adicionamos no versionamento através do comando “git add .” nesse comando permite ao GIT trabalhar com todos os arquivos adicionados no versionamento. Temos ainda o estado **consolidado**, após os arquivos serem preparados o próximo passo é consolidar todos os arquivos adicionados, isso é feito através do comando **git commit** que cria uma versão do projeto atual, e guarda essa versão no seu repositório local. Esse procedimento funciona como se estivéssemos tirando uma foto do momento atual, que servirá como referência para outros membros que estejam trabalhando no mesmo projeto. Alguns servidores remotos web (GitHub e GitLab) que trabalham com GIT permite guardar essas versões em repositórios remotos conhecidos de branches.

Este foi nosso podcast de hoje! Até a próxima!