Curso DevOps

WWW.LINUXFORCE.COM.BR

O que é Git

Git é um sistema open-source de controle de versões. Amplamente utilizado por desenvolvedores, vem se tornando uma cada vez mais utilizado pelo time de operação para criar histórico de alterações nos códigos e arquivos de configuração dos serviços.

Instalação Git

Vamos realizar a instalação do git para iniciar o controle de versões de nosso código.

Como estamos utilizando a maquina virtual Cliente Interno Novo, basta acessar o terminal e executar os comandos abaixo.

apt install git

Para verificar se o Git está instalado corretamente basta digitar o comando
git --version

Instalação no Windows poderá ser realizado acessando o site https://git-scm.com/download/win

Instalação no macOS poderá ser realizado acessando o site https://git-scm.com/download/mac

Repositórios Git

Cada diretório de trabalho do Git é um repositório com um histórico completo e habilidade total de acompanhamento das revisões, não dependendo de acesso a uma rede ou a um servidor central.

Criando Diretório para receber repositórios do git
mkdir /var/repositorio1
mkdir /var/repositorio2

Iniciando um Repositório

Para você começar a monitorar um projeto existente com Git, você deve ir para o diretório do projeto. Se você nunca fez isso, use o comando a seguir, terá uma pequena diferença dependendo do sistema em que está executando:

Acesse o repositório
cd /var/repositorio1

Definindo Repositório
git init

Status do Repositório
git status

Definindo Identidade Local
#git config --local user.email "rogerio@asf.com"
#git config --local user.name "Rogério"

Iniciando um Repositório

Para controlar o versionamento dos arquivos existentes, você devera iniciar a monitoração dos arquivos realizando um processo de commit, para isto é necessário adicionar os arquivos utilizando o comando git add para especificar os arquivos e sequencia executar o git commit conforme exemplo abaixo.

Adicionando arquivos ao diretório do repositório
cp /etc/passwd /var/repositorio1

Adicionando arquivos a serem monitorados pelo git
git add passwd

Status do Repositório
git status

Realizando o primeiro Commit
git commit –m “Primeiro Commit”

Clonando Projetos

Cópia de um repositório existente pode ser realizado através de clones, o comando para isso é o **git clone.** O git faz um copia completa de praticamente todos os dados que o servidor possui, cada versão de cada arquivo no histórico do projeto é obtida.

Clonando repositório local
mkdir /var/clonerepolocal
cd /var/clonerepolocal
git clone /var/repositorio1

Clonando repositório remoto
mkdir /var/clonereporemoto
cd /var/clonereporemoto
git clone `git clone https://github.com/libgit2/libgit2`

Salvando Alterações

Nesta etapa vamos adicionar novos arquivos em nosso repositório para serem monitorados, criando assim pontos de versionamento.

Acesse o repositório1
cd /var/repositorio1
git status

Copiando novos arquivos para repositório
cp /etc/hosts /var/repositorio1
cp /etc/shadow /var/repositorio1
git status

Adicionando todos os novos que estão dentro do repositório para serem monitorados pelo git.
git add .
git status
git commit –m “Adicionando arquivo hosts e passwd no repositório”
git status

Históricos

É possível visualizar todo o histórico de alterações que estão sendo realizadas dentro de nosso repositório para isto vamos aprender a utilizar o comando **git log**

Logs detalhados de todos os commit
git log

Logs resumidos em apenas 1 linha
git log --oneline

Logs de alterações de commit com comparação entre alterações realizadas
git log -p

Personalização da exibição de Logs
git log --pretty="format:%h %s"

Documentação
https://devhints.io/git-log

Ignorando Arquivos

Caso tenham algum arquivo presente dentro do repositório que não tenha necessidade de ser monitorado pelo git pode ser adicionado em um arquivo dentro do próprio repositório chamado **.gitignore** com isto o mesmo será ignorado ao realizar commits.

Crie um arquivo dentro do repositorio1
touch faq.txt
git status

Crie o arquivo do .gitignore e adicione uma linha contendo o nome do arquivo
vim .gitignore
git status

Adicione o .gitignore na relação de arquivos monitorados e realize o Commit da alteração.
git add .gitignore
git commit -m "Adicionando Arquivo .gitignore"
git status

Verificando Log Resumido
git log --oneline

Repositórios Remotos

Os repositórios remotos são locais onde podemos enviar nossas alterações para que outras pessoas possam acessa-las, este repositório somente recebera alterações não realizando alterações diretamente nele, para isto utilizaremos o comando **git init --bare** indicando que este será um repositório **puro** contendo apenas alterações realizadas e não copias completas dos arquivos.

Criação do repositório remoto e definição para conter apenas alteração
mkdir /var/repositoriobare
cd /var/repositoriobare
git init --bare

Adicionando repositório remoto no projeto do repositorio1
cd /var/repositorio1
git remote add local /var/repositoriobare
git remote -v

Sincronizando Dados

Nesta etapa vamos fazer com que o usuário sincronize dados do seu repositório com o servidor, o comando **git push local master** realizaremos o envio de todos os dados que estão na brach master para o repositório remoto que chamamos de local.

Acessando repositório Local Repositório 1 e enviando dados para repositório Remoto
git push local master

Acessando repositório Local Repositório 2 e criando repositório remoto
/var/repositorio2
git init
git remote add local /var/repositoriobare/
git remote –v

Atualizando repositório local a partir do repositório remoto
git pull local master
ls

Definindo Identidade Local
#git config --local user.email “kamila@asf.com"
#git config --local user.name “Kamila"

Github

Github é um serviço que permite fazer armazenamento de repositórios externo, após criar uma conta de forma gratuita podemos criar repositórios públicos ou privados, realizar commits e verificar históricos de forma rápida é simples.

1. Acesse o github e crie uma nova conta https://github.com/login
2. Clique em **Your repositories**
3. Clique em **New**
4. Defina o nome de seu repositório, marque a opção **public** e clique em **Create repository**

Procedimento para criar o repositório Remoto via command line
mkdir /var/repositorio3
cd /var/repositorio3
git init
git remote add origin https://github.com/rogerramossilva/devops1.git
git remote add local /var/repositoriobare/
git remote –v

Github

Atualizando a partir do repositório bare
git pull local master

Atualizando repositório remoto externo
git push origin master

Observação
Para que a atualização será realizado no repositório externo será necessário informar o usuário e senha do github para autenticação.

Acessar o github para verificar se todos os arquivos estão presentes no repositório criado.

Criar um novo arquivo no github, realizar o commit e executar o comando abaixo para atualizar o repositório local interno.
git pull origin master
git pull local master

Branches

Branches são ramificações independentes não qual é possível trabalhar de forma livre e independente, verionando quando entendermos que o código já está pronto, isto possibilita que inúmeros colaboradores trabalhem em um mesmo projeto sem compromete-lo.

A branch principal é chamada de máster, portanto será nossa branch estável o código desta vai estar pronto para ser entregue.

Para visualizar as branchs de um repositório basta digitar o comando baixo
git branch

Comando para criar nova Branch
git branch config1
git branch

Comando para selecionar um branch
git checkout config1
git branch

Branches

Realizando commit na branch config1
git checkout config1
echo “10.0.3.15 debian9” >> hosts
git status
git add hosts
git commit –m “Primeiro commit na branch config1”
git status

Comando para visualizar commit em todas as branchs
git log --all
git log --graph

Acesse o segundo repositório da Maria
cd /var/repositorio2

Comando para criar e selecionar um branch
git checkout –b config2

Branches

Comando para selecionar um branch
git checkout config2
git branch

Analisando o log não aparece o Commit realizando na Branch config1 do repositório 1
git log

Adicionando alteração no arquivo Hosts
echo “10.0.3.20 interno” >> hosts
git status
git add hosts

Realizando Commit e analisando Log
git commit –m “Primeiro commit na branch config2”
git status
git log

Merges

Merge é a junção de branchs

Acessar o repositorio3 para realizar merge com a branch master
#cd /var/repositorio3
#git checkout master
#git merge config1
#git log

Acessar o repositorio2 para realizar merge com a branch master
#cd /var/repositorio3
#git checkout master
#git merge config2
#git log

Atualizando Branch

Acesse o Repositório 2 para fazer o envio das alteração para repositório bare
cd /var/repositorio2
git push local master

Acesse o Repositório 3 para fazer o envio das alteração para repositório bare
cd /var/repositorio3
git push local máster

Após o erro informando que tem conflitos, devemos atualizar a branch master
git pull local master

Ajustando o Arquivo para posteriormente realizar novamente o Commit
vim /etc/hosts
git add hosts
git commit –m “Correção após Merge”

Enviando informações para repositório bare
git push local master

Resolvendo Conflitos

Removendo mudanças após edição do arquivo e antes de realizar add para monitorar
cd /var/repositorio3
vim hosts
git status
git checkout -- hosts

Remove mudanças após o add para monitoração antes de realizar Commit
vim hosts
git status
git add hosts
git status
git reset HEAD hosts
git status
git checkout – hosts

Revertendo Commit

Para reverter algum commit aplicado é necessário obter informações sobre sua identificação e analisar o impacto que esta reversão pode trazer.

Acessar o Arquivo que se encontra no Repositorio 3
cd /var/repositorio3
vim hosts
git add hosts
git commit -m “Teste para reverter para Commit Anterior”

Após realizar o Commit acessar o log para Analise e identificação
git log –p

Com o comando git revert e possível reverter para um ponto especifico.
git revert e917b147c9f11637530295b8f9aacf4b7be1f1e2
git status
cat hosts

Stash

Acesse o Repositório 3 para realizar uma alteração no Arquivo hosts e posteriormente esconder.
cd /var/repositorio3
vim hosts
git status
git stash

Para listar os trabalhos que estão guardados utilize o comando abaixo
git stash list

Realizando o Commit do Arquivo
vim hosts
git status
git add hosts
git commit -m “Alterando arquivo pós Stash"

Stash

Voltando o Estado de um Trabalho em Stash
git stash list
git stash apply 0

Resolvendo Conflitos pós apply stash
vim hosts
git add hosts
git commit –m “Commit executado pós resolução de Conflitos Stash”

Removendo Stash
git stash drop 0

Enviando alteração para repositório bare
git push local master

Vendo Alterações

Todas as alterações são gravadas em logs possibilitando verificar pontos de alteração que ocorreu entre os commits realizados.

Exibição de Logs
cd /var/repositorio3
git log -p
git log --oneline

Para verificar a diferença entre intervalo de commits basta colocar o intervalo separado por ..
git diff 4b9e783..0a63fd1

Alterar um arquivo e ver a diferença antes de realizar o Commit
vim hosts
git status
git diff

Removendo as mudanças e voltando o arquivo ao estado anterior.
git checkout -- hosts
git status

Tags e Releases

Após conclusão as fases de um Projeto chega o momento de criar um marcação indicando o surgimento de uma nova versão, este será um ponto onde sinalizaremos que não poderá mais sofrer mudanças, qualquer mudança devera ser incluída em versões posteriores. Para criar estes marcos utilizaremos o conceito de Tags.

Acesse o Diretório do Repositório 3 e vamos criar nossa primeira versão
cd /var/repositorio3
git log -n2
git tag -a v0.1.0 -m "Lançado primeira versão"

Para verificar as Tags Criadas digitar o Comando abaixo
git tag

Enviando Mudanças para Repositório Bare
git push local master

Enviando Mudanças para Tag Criada
git push local v0.1.0

Tags e Releases

Enviado Mudanças e Tags para repositório Remoto
git remote –v
git push origin master
git push origin v0.1.0

Após realizar as ações acessar o Github para verificar as informações.

Release é a versão pronta para ser lançada ou baixada por qualquer pessoa que queria utilizar.

Para baixar basta ir no Github e acessar o repositório e baixar os arquivos zip ou tar.gz