GIT

Распределённая система контроля версий (РСКВ). В РСКВ (таких как Git, Mercurial, Bazaar или Darcs) клиенты не просто скачивают снимок всех файлов (состояние файлов на определённый момент времени) — они полностью копируют репозиторий. В этом случае, если один из серверов, через который разработчики обменивались данными, умрёт, любой клиентский репозиторий может быть скопирован на другой сервер для продолжения работы. Каждая копия репозитория является полным бэкапом всех данных.

Преимущества:

- хранение бэкапов
- возможность синхронизировать код
- отмена изменений
- отслеживание изменений и владельцев
- окружение для хранения не завершенного кода

Конфигурация: при первом использовании надо сконфигурировать пользователя, репозиторий с которым работаем и т.д.

Главные этапы работы в ГИТ:

1. **Рабочий каталог** (working directory untracked/tracked) – это состояние репозитория после команды git init, тоесть когда он начал отслеживаться системой
2. **Плацдарм** (Stagging area) - это состояние репозитория после команды git add, когда измененные файлы добавлены на в stagging area
3. **Local repository** - это состояние репозитория после команды git commit -m, изменения сохранены локально
4. **Remote Repository** - это состояние репозитория после команды git push, изменения сохранены удаленно

Commands	
`init`	инициализация гит репозитория (создание папки .git)
`clone`	создание копии уже существующего репозитория.
`add`	добавить файлы в локальный репозиторий после того как в них произошли изменения
`status`	посмотреть состояние файлов которые хранятся локально (есть/нет изменений, untracked/tracked и тд)

`commit`	фиксация изменений произведённых в локальном репозитории с добавление «сообщения/описания»
`log`	список коммитов сделанных в репозитории в порядке их выполнения
`show`	просмотр информации о метке или коммите (в понятном человеке виде)
`fetch`	синхронизировать репозитории (БЕЗ СКАЧИВАНИЯ изменений)
`merge`	скачать недостающие файлы туда где их еще нет (после синхронизации)
`pull`	(fetch + merge) – одновременная синхронизация и получение изменений с удаленного репозитория
`push`	отправка изменений с локального репозитория на удалённый
`Work with branches`	
`Revert`	отменяет изменения внесённые на удаленный репозиторий (создает зеркальный коммит но без изменений которые мы отменили)
`Checkout – b "name_branch"`	создание ветки с именем name_branch
`Rebase`	изменение места ветвления
`Merge`	слияние веток
`Diff`	решение merge конфликта вручную в помощью Bash.
git revert HEAD -m 1	Отмена слияния, которое уже было отправлено в репу.
`git branch --delete old-branch`	Удаление локальной ветки
`Work with commits`	

`git commit --amend -m "add ft.tfvars to the NotificationNCService"`	Rename last commit in local repo