

Федеральное государственное автономное
образовательное учреждение высшего
образования
«Национальный исследовательский университет
ИТМО»

Факультет Информационных технологий и программирования

Работа: Лабораторная работа по Git № 3

Выполнила студентка группы №М3103
Кравченкова Елизавета Александровна

Проверил:
Повышев Владислав Вячеславович

Санкт-Петербург

2022 г.

1. Системы управления репозиториями

1.1 Bitbucket

Bitbucket – это веб-сервис хостинга репозиториях контроля версий, принадлежащий Atlassian, для исходного кода и проектов по разработке, использующих системы контроля версий Mercurial или Git. Bitbucket предлагает, как коммерческие планы, так и бесплатные аккаунты. Он предлагает бесплатные аккаунты с неограниченным количеством частных репозиториях, которые могут иметь до пяти пользователей в случае бесплатных аккаунтов. Bitbucket в основном используется для кода и его проверки. Bitbucket поддерживает следующие функции:

1.2 GitHub

GitHub, Inc. является дочерней компанией Microsoft, которая предоставляет хостинг для разработки программного обеспечения и контроля версий с помощью Git.

GitHub предлагает свои основные услуги бесплатно. Его более продвинутые профессиональные и корпоративные услуги являются коммерческими. Бесплатные аккаунты GitHub обычно используются для размещения проектов с открытым исходным кодом.

GitHub предлагает неограниченное количество частных репозиториях для всех планов, включая бесплатные аккаунты, но разрешает размещать до трёх участников на каждый репозиторий. Начиная с 15 апреля 2020 года, бесплатный план позволяет иметь неограниченное количество участников, но ограничивает частные репозитории 2000 минутами действий в месяц. По данным на январь 2020 года, GitHub насчитывает более 40 миллионов пользователей и более 190 миллионов репозиториях, включая не менее 28 миллионов публичных репозиториях, что делает его крупнейшим хостом исходного кода в мире.

Основная цель GitHub – облегчить контроль версий и аспекты отслеживания проблем при разработке программного обеспечения. Ярлыки, контрольные точки, распределение ответственности и поисковая система доступны для отслеживания проблем. Для управления версиями Git позволяет pull запросами предлагать изменения в исходный код. Пользователи, у которых есть возможность просматривать предлагаемые изменения, могут видеть разницу в запрошенных изменениях и одобрять их. В терминологии Git это действие называется «committing», и один из его экземпляров – «commit». История всех коммитов сохраняется и может быть просмотрена позже.

1.3 GitLab

GitLab – это веб-инструмент жизненного цикла DevOps, предоставляющий Git-репозиторий, функции вики, отслеживания проблем и непрерывной интеграции и развёртывания, используя лицензию с открытым исходным кодом, разработанную GitLab Inc. Программное обеспечение создано украинскими разработчиками Дмитрием Запорожцем и Валерием Сизовым. Первоначально код был написан на языке Ruby, затем некоторые его части были переписаны на языке Go, первоначально как решение для

управления исходным кодом для совместной работы в команде по разработке программного обеспечения. Позднее он превратился в комплексное решение, охватывающее жизненный цикл разработки программного обеспечения, а затем - весь жизненный цикл DevOps. Текущий стек технологий включает в себя Go, Ruby on Rails и Vue.js.

Он следует модели разработки с открытым ядром, где основной функционал выпускается по лицензии с открытым исходным кодом MIT, а дополнительный функционал – по проприетарной лицензии

1.4 Ключевые различия между Bitbucket, GitHub и GitLab

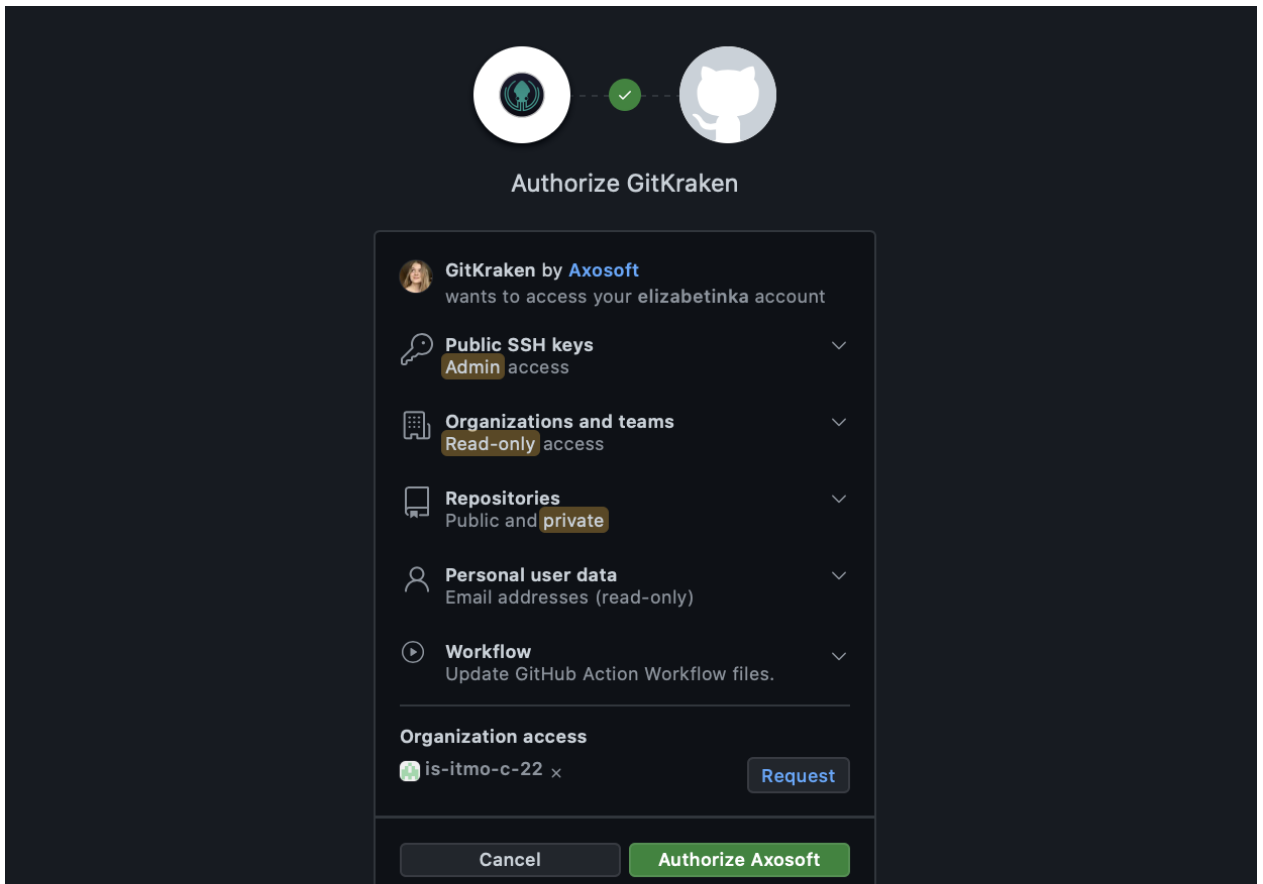
Фактор сравнения	Bitbucket	GitHub	GitLab
Бесплатные приватные репозитории	Bitbucket предоставляет пользователям функциональность бесплатных приватных репозиториях	В платформе GitHub такой функциональности нет.	GitLab также предоставляет пользователям функциональность бесплатных приватных репозиториях
Бесплатные публичные репозитории	Все платформы предоставляют пользователям функциональность бесплатных публичных репозиториях.		
Merge запрос или шаблоны проблем	В платформе Bitbucket такой функциональности нет.	Пользователь может запросить Merge запросы и шаблоны проблем.	
Встроенные функции CI (Continuous Integration)	Встроенные функции Continuous Integration предусмотрены на Bitbucket.	В GitHub такой функциональности нет. Если пользователь хочет использовать эту возможность, можно воспользоваться сторонним	Встроенные функции Continuous Integration предусмотрены на GitLab.

		приложением.	
Открытый исходный код	Платформа Bitbucket не является открытой. Эта платформа предоставляет некоторые функции, но полностью платформа не является открытой.	Платформа GitHub также не является платформой с открытым исходным кодом, и пользователь имеет ограниченные бесплатные возможности в этой платформе.	GitLab – это платформа с открытым исходным кодом, и пользователи могут использовать все функции без каких-либо затрат.
Удобство навигации	В платформе Bitbucket такой функционал не предусмотрен.	Удобство использования навигации предусмотрено в платформе.	
Файловое хранилище большого размера	Все платформы способны обеспечить возможность хранения файлов большого размера.		
Интеграция сторонних инструментов	Все платформы способны интегрировать в платформу инструменты сторонних производителей.		
Анализ проектов	Пользователь и могут график проекта диаграмму сгорания задач в Bitbucket.	Для платформы GitHub такой функционал не предусмотрен, а возможности пользователя ограничены.	Пользователь и могут график проекта диаграмму сгорания задач в GitLab.

2. Графический интерфейс для Git (GitKraken)

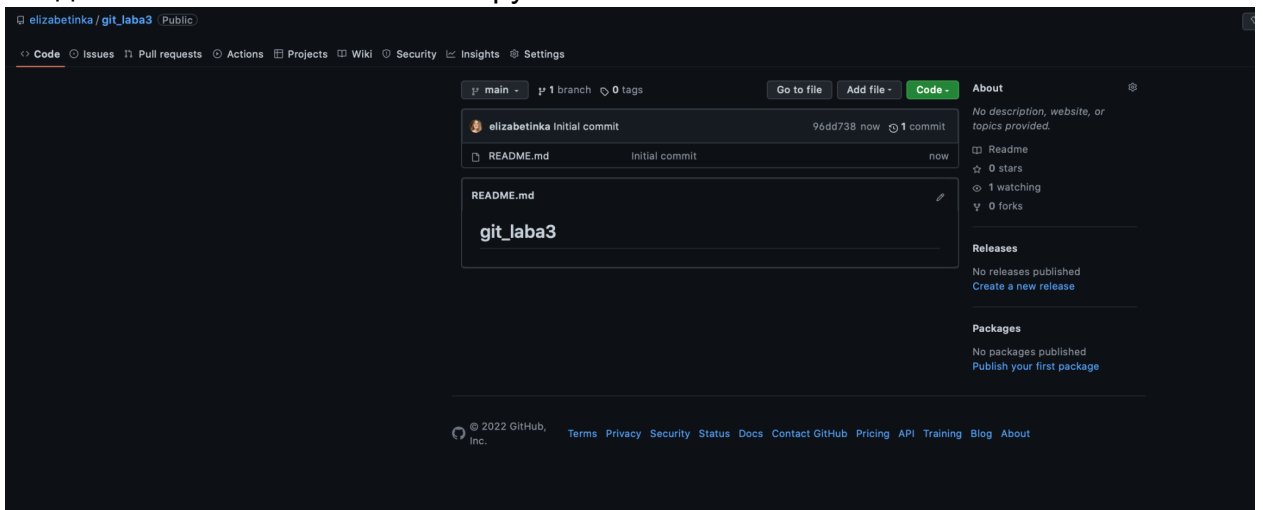
2.1 Установка

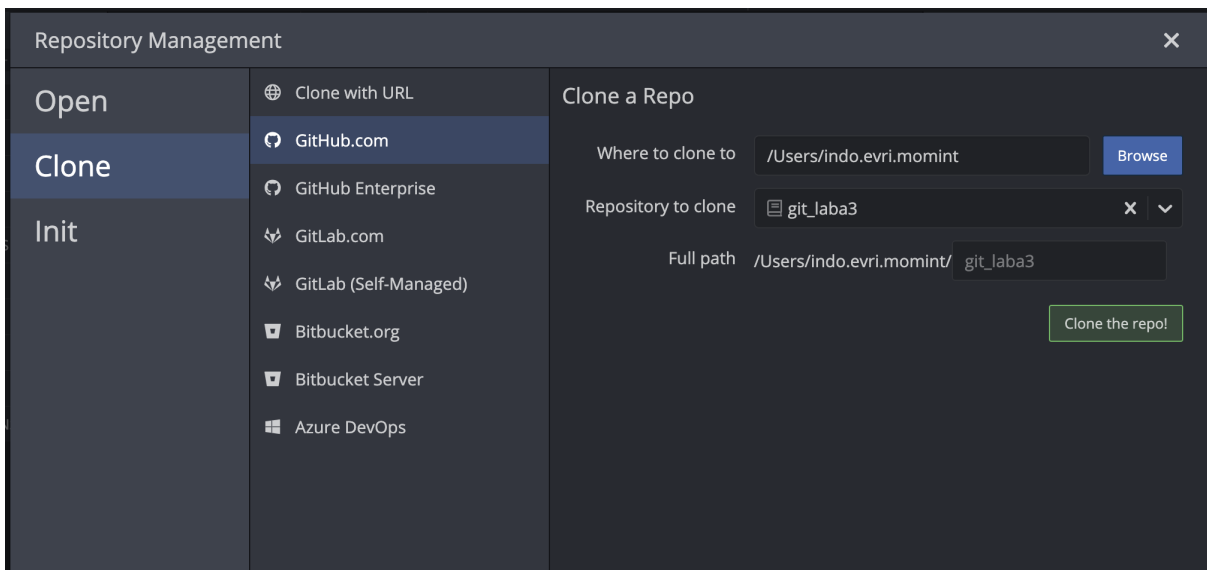
Переходим на <https://www.gitkraken.com/download> и скачиваем. Связываем его с аккаунтом в GitHub



2.2 Создаем репозиторий

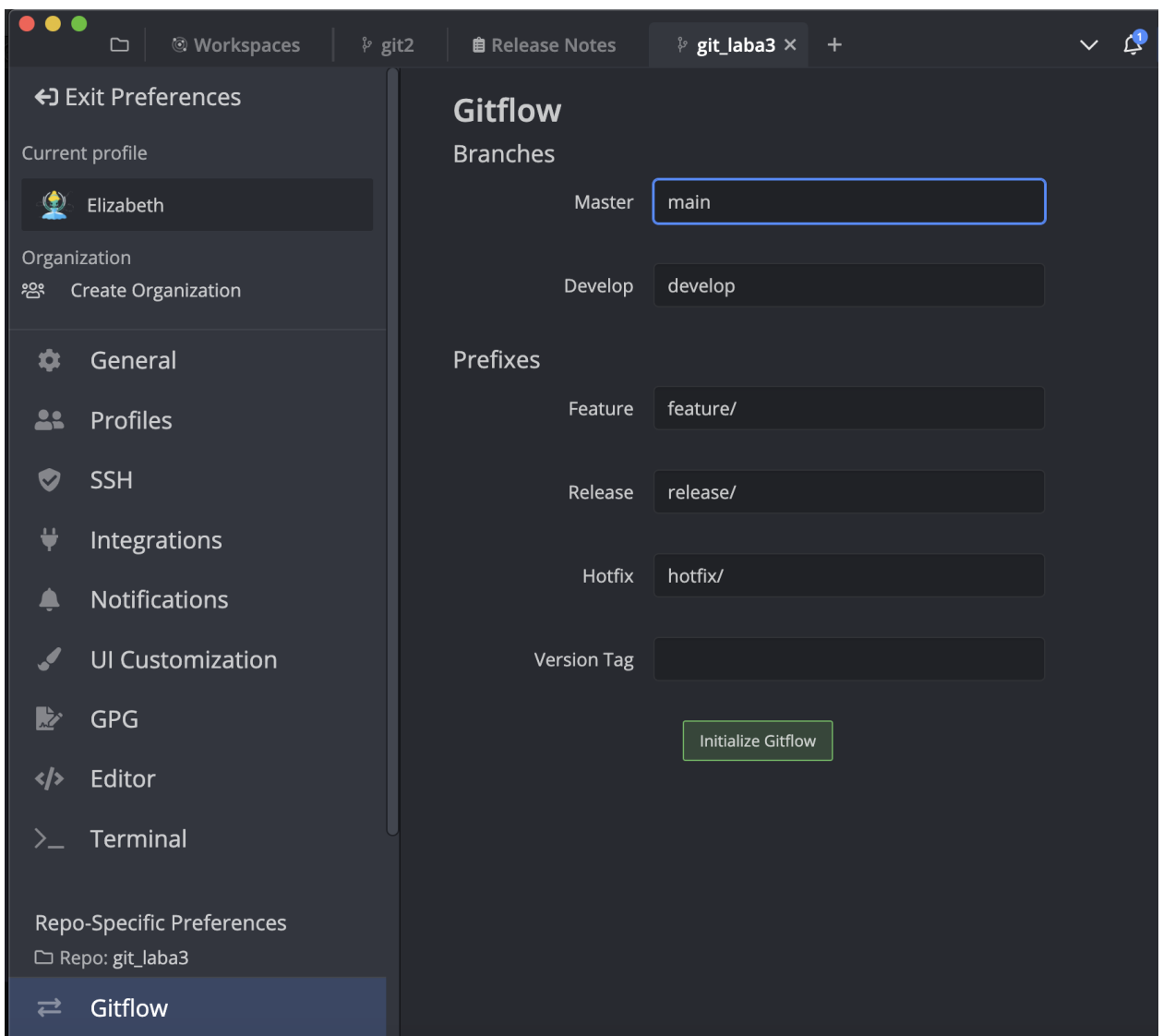
Создаем в GitHub. Потом клонируем его в GitKraken





2.3 Инициализируем GitFlow и LFS

Переходим в preferences



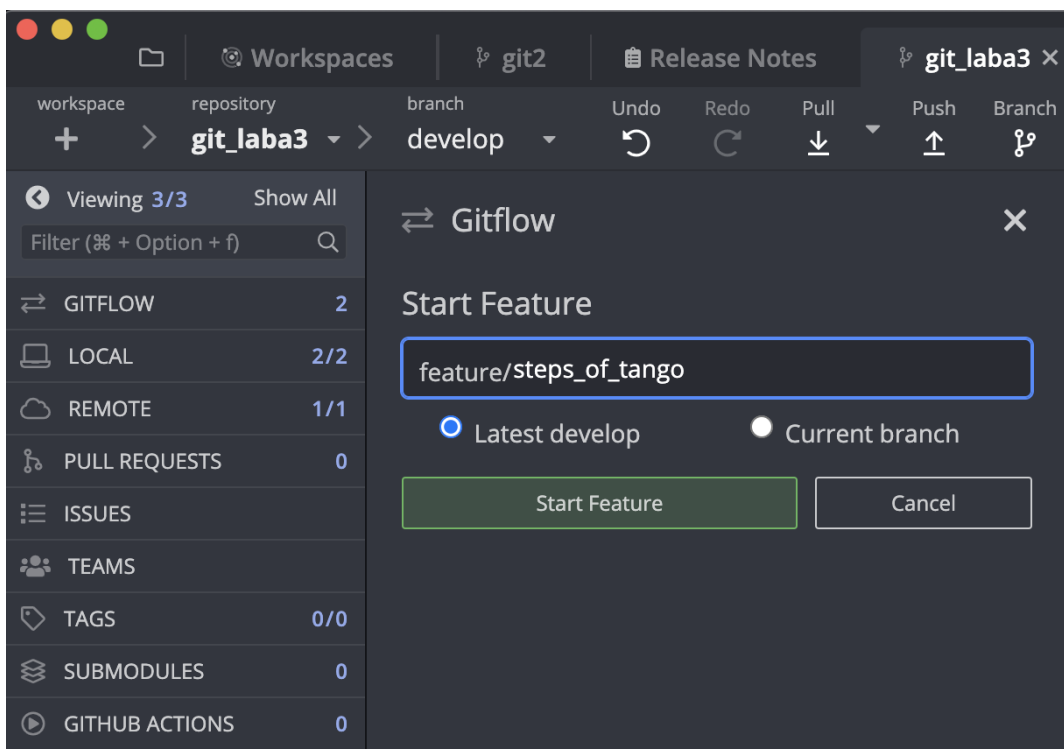
LFS

Initialize LFS

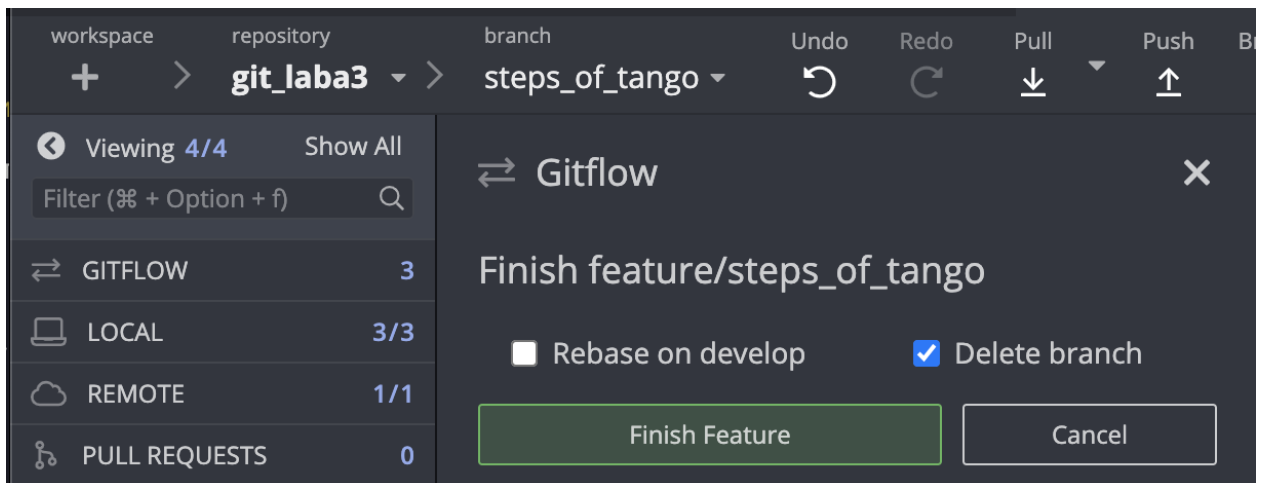
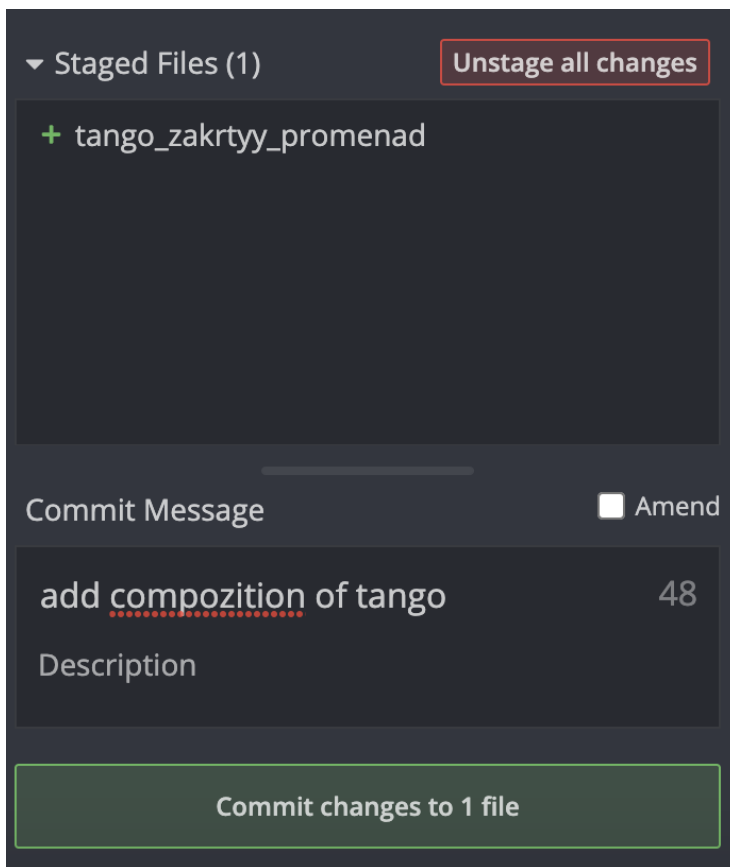
LFS has not been initialized on git_laba3

Initialize LFS on this repo

2.3 Из ветки develop сделаем ветку feature

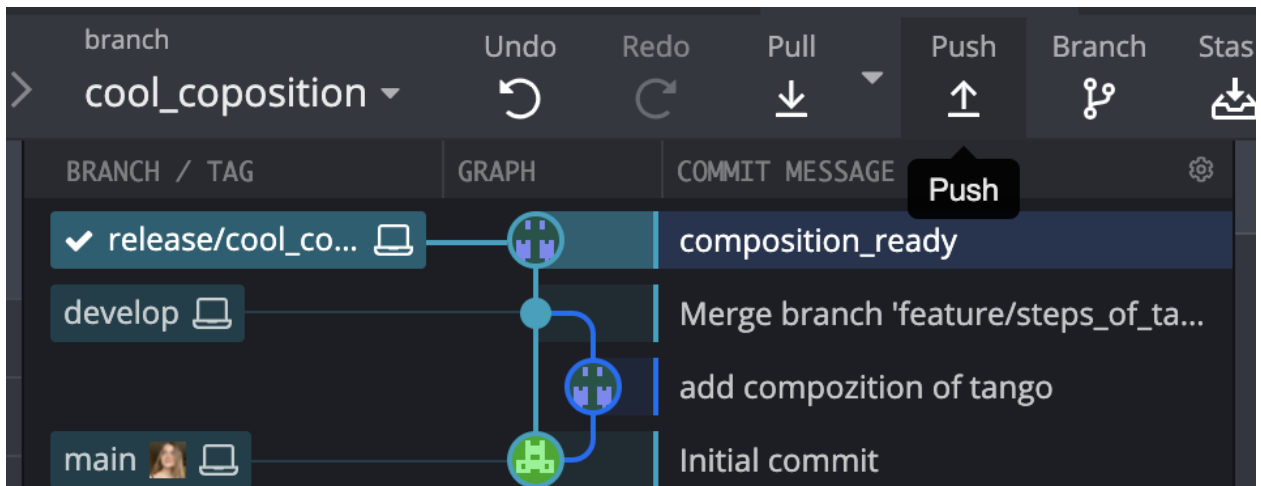
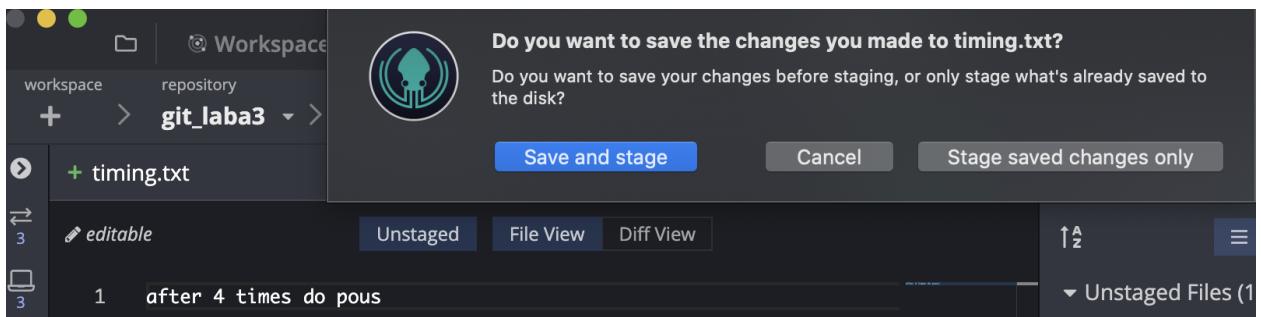
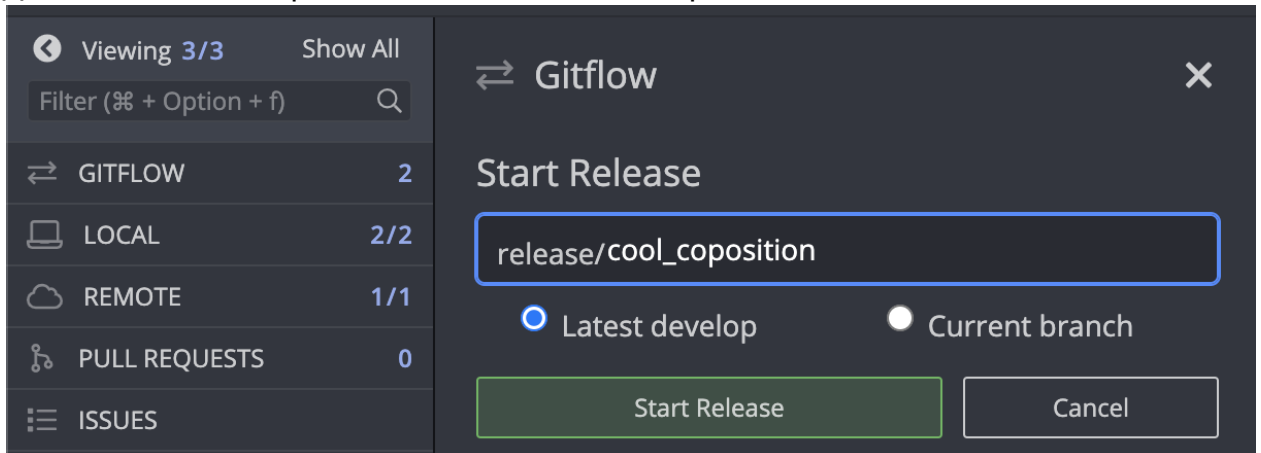


2.4 Создадим файл, сделаем коммит и удалим ветку feature



2.5 Из ветки develop делаем ветку release

Добавляем новый файл. Коммитим и делаем push



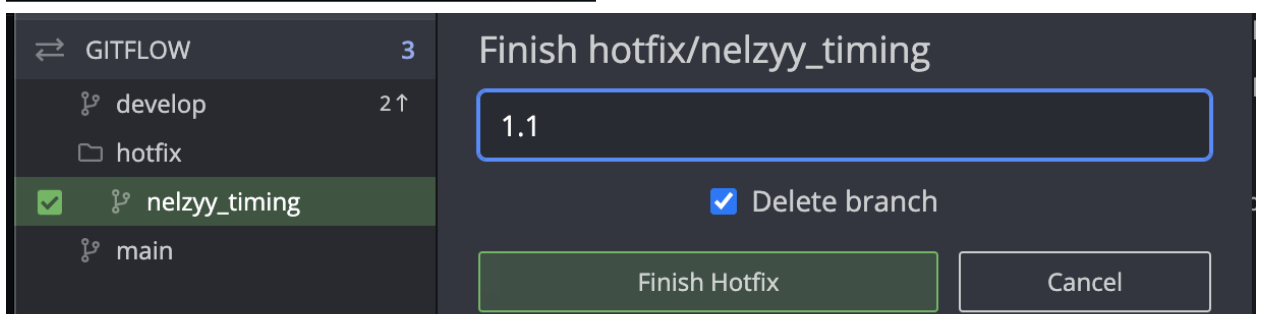
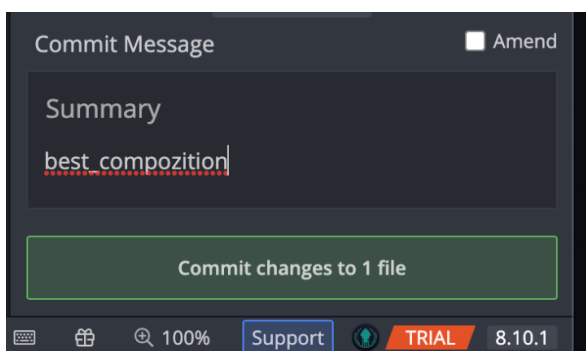
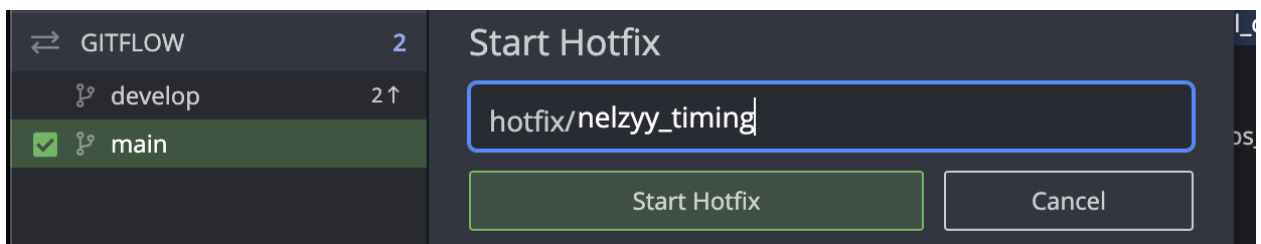
2.6 Удаляем ветку release.

Нам предлагают дать имя тегу. Пушим тег



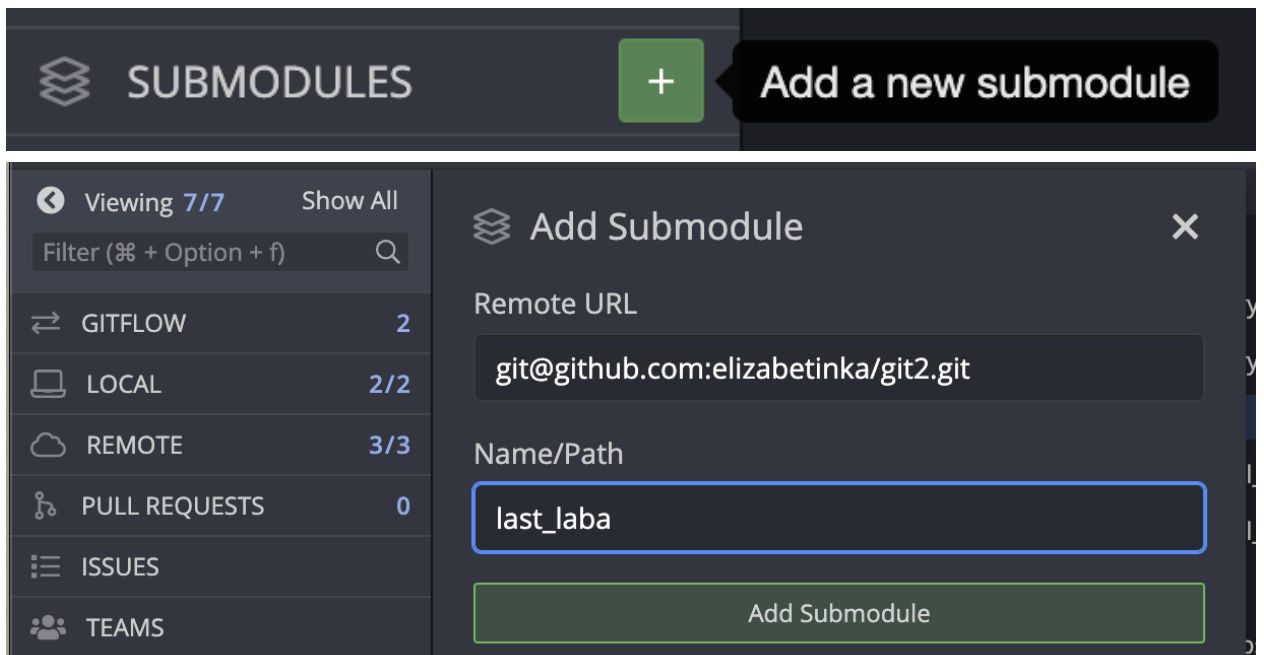
2.7 Работа с веткой hotfix.

Из main создаем ветку hotfix. Меняем файл. Делаем коммит. Удаляем ветку. Нам также предлагают сделать тег



2.8 Submodules

Добавим submodule



2.8 LFS

При инициализации укажем, что LFS отслеживает файлы расширения bmp. Добавим файл с расширением bmp. Сделаем КОММИТ

