Оглавление

[1. Однопользовательский, 1 ветка. 1](#_Toc520213473)

[2. Однопользовательский, 3 ветки 3](#_Toc520213474)

[3. Многопользовательский совместный 3](#_Toc520213475)

[4. Многопользовательский с архитектором 3](#_Toc520213476)

[5. Многопользовательский, 2 общих ветки (продакшен и препродакшен) 3](#_Toc520213477)

# Однопользовательский, 1 ветка.

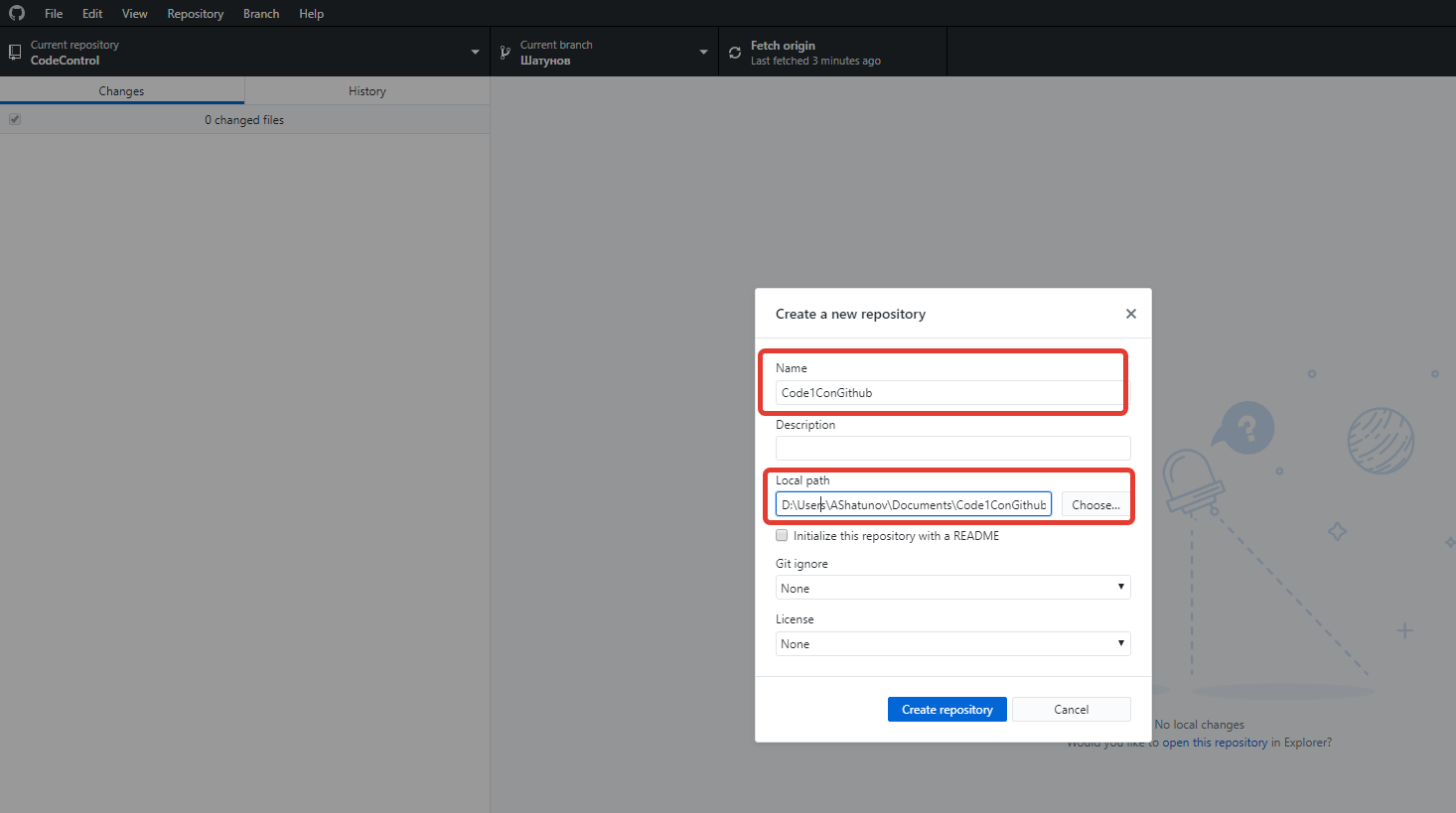
В этом случае мы извлекаем 2 функции гитхаба – систему контроля версий и облачное хранилище.

Ситуация: есть несколько компьютеров с интернетом, пользователь ведет разработку чего-либо.

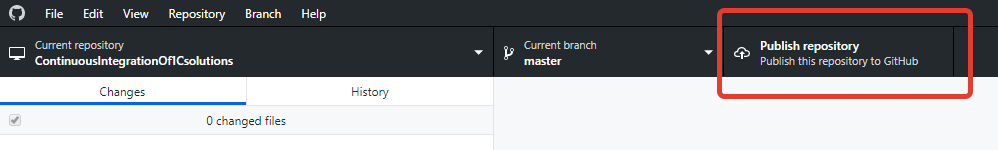
* 1. Создание репозитория

На любом из компьютеров открыть клиент и выполнить File/New repository

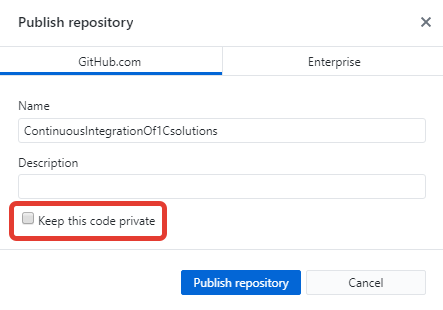
В появившемся окне указать имя проекта и путь к локальному хранилищу.



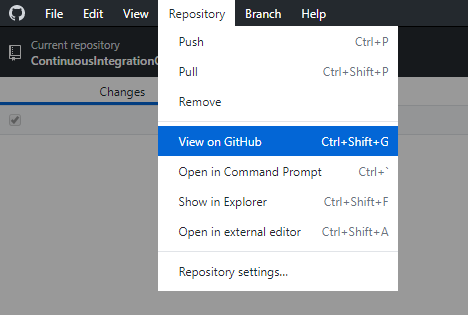
После создания нового проекта нужно отправить его в глобальный репозиторий. Для этого выполнить появившуюся команду Publish repository



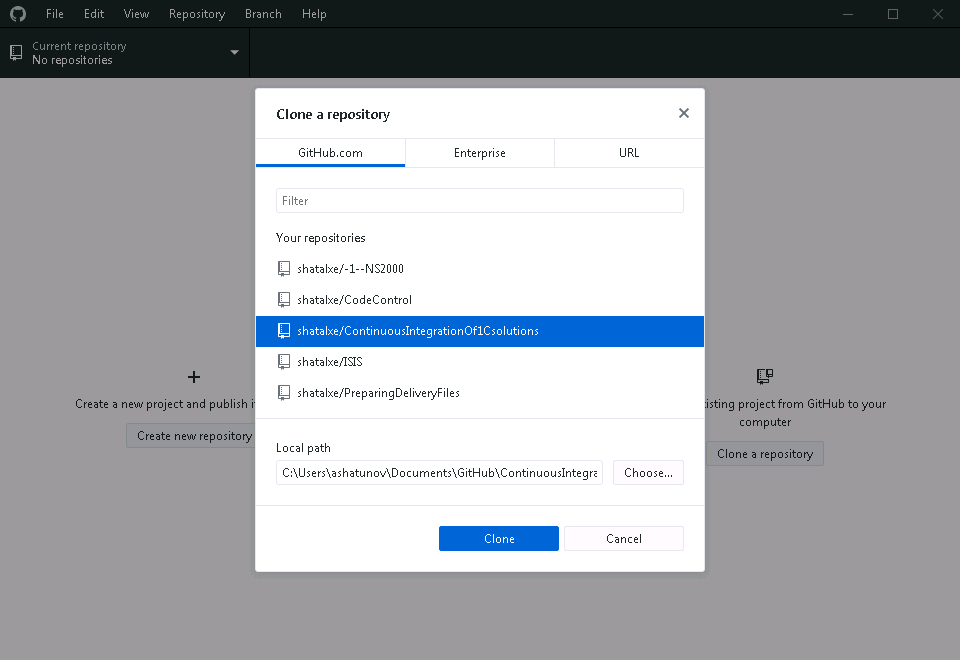
В новом окне убрать флаг Keep this code private



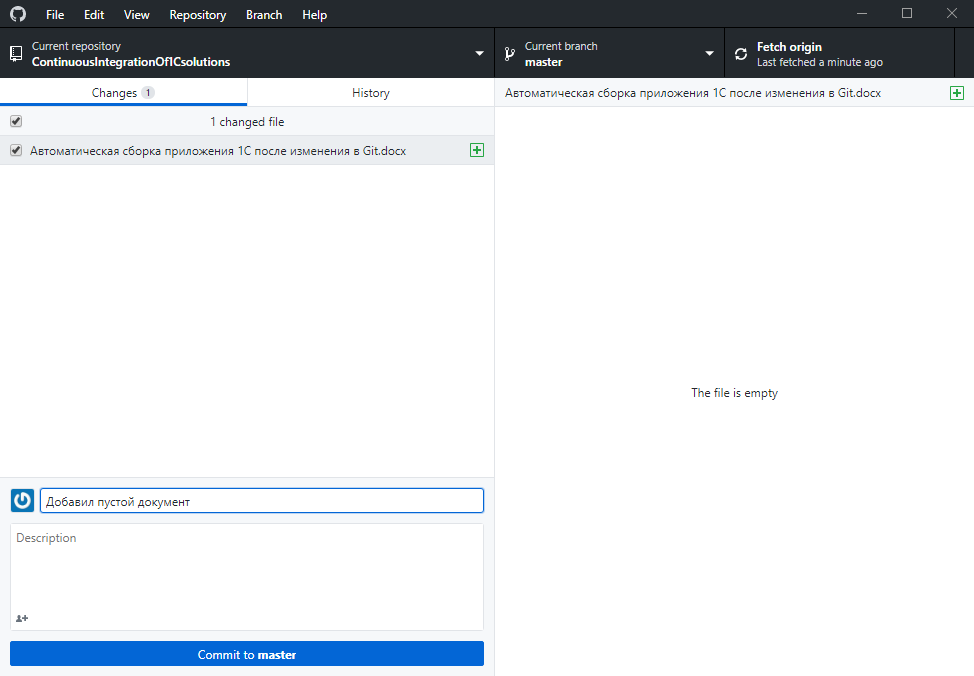
После публикации, за проектом можно следить через Веб-клиент.



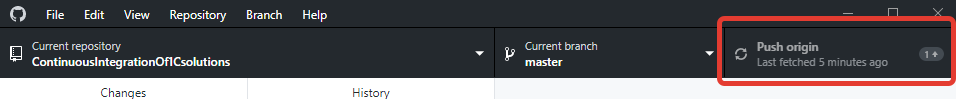
* 1. Подключение нового компьютера к репозиторию
     1. На втором устройстве открыть клиент GitHub
     2. Авторизоваться
     3. Выполнить File/Clone repository
     4. Указать нужный проект, нажать Clone



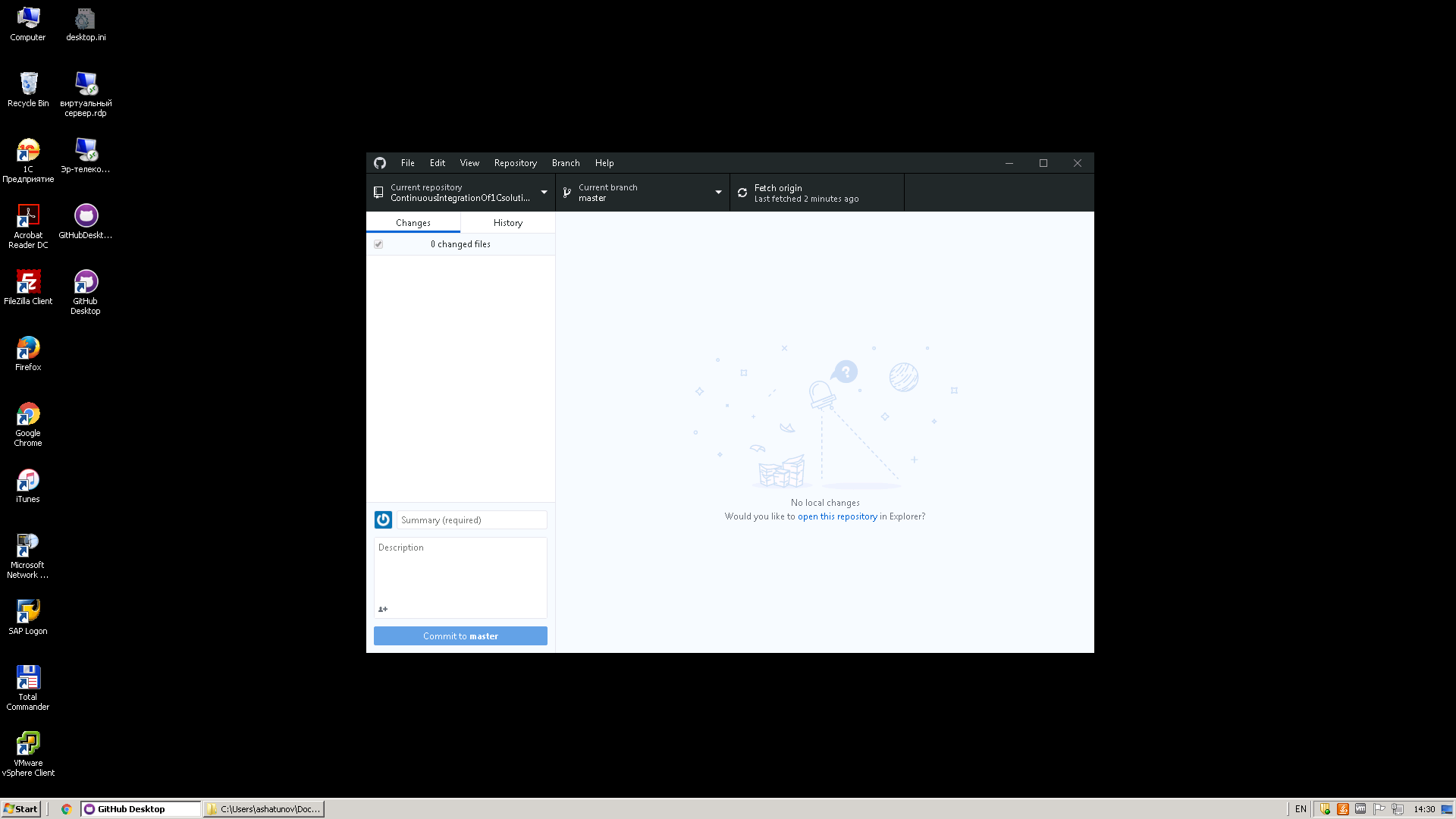
* 1. Выгрузка изменений в глобальный репозиторий
     1. Внести изменения в каталог проекта. Клиент GitHub покажет эти изменения.



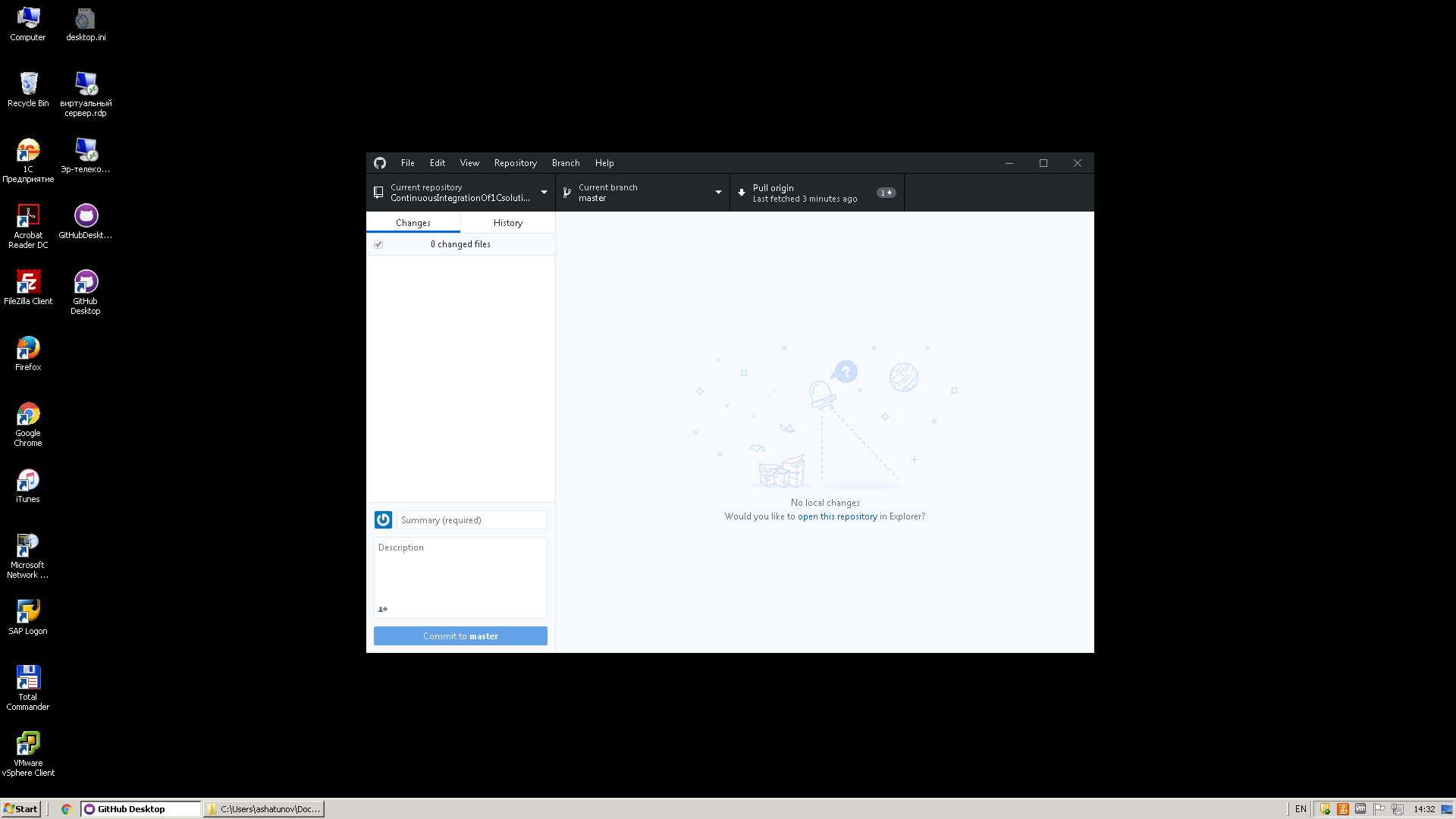
* + 1. В поле summary нужно ввести краткое описание аносимых изменений
    2. Выполнить commit to master
    3. Выполнить появившуюся команду Push origin



* 1. Загрузка из глобального репозитория
     1. На втором устройстве выполнить Fetch origin



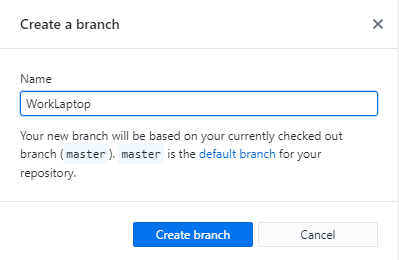
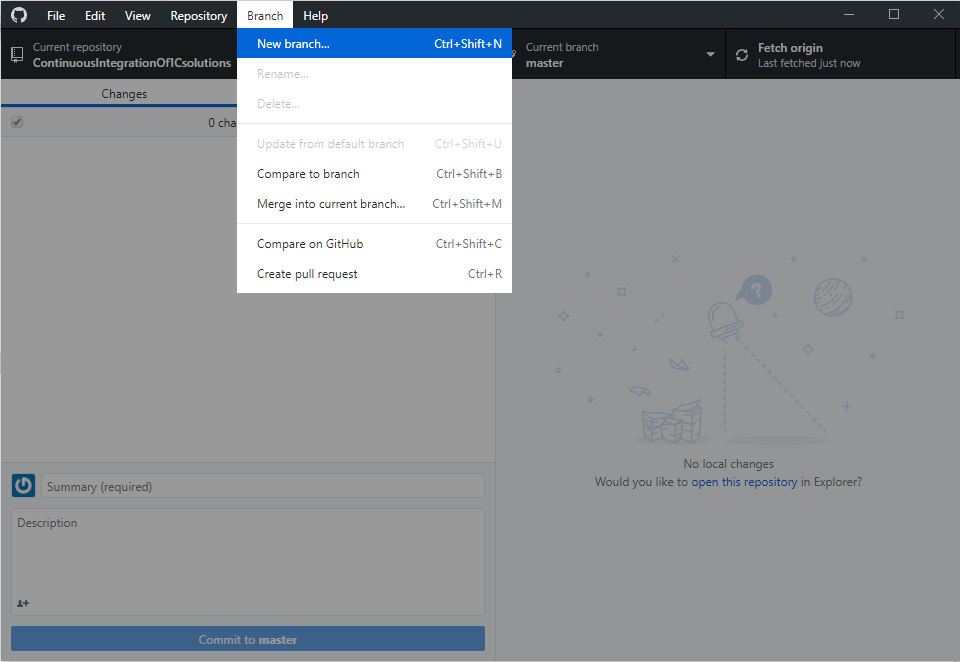
* + 1. Выполнить Pull origin



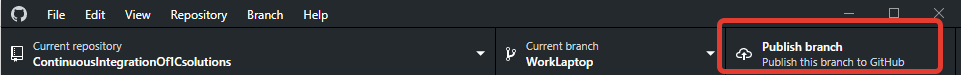
# Однопользовательский, 3 ветки

Вынесем работу на разных устройствах в разные ветки. Время от времени изменения будем отправлять в ветку master и обновляться из неё.

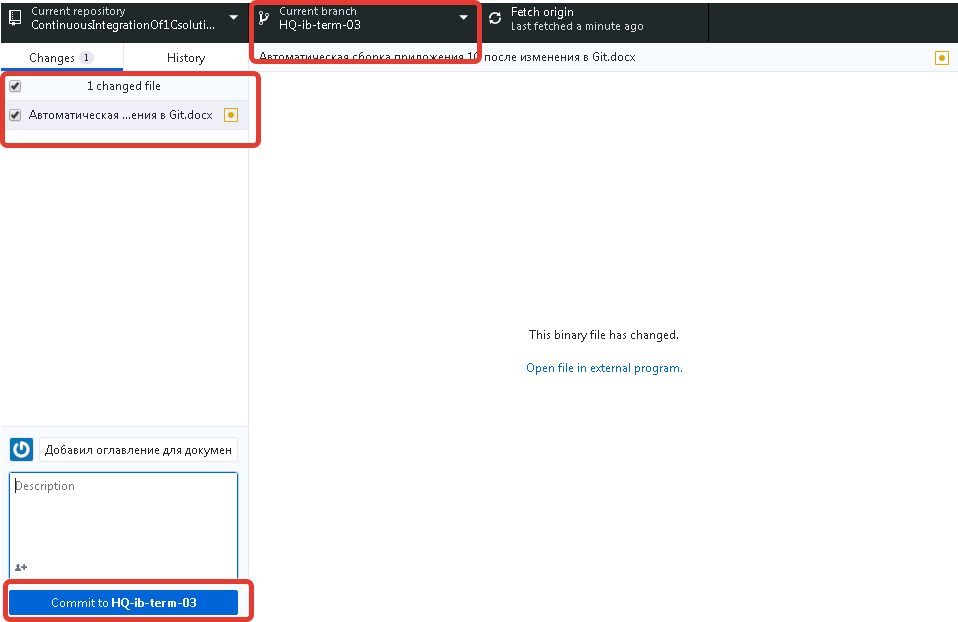
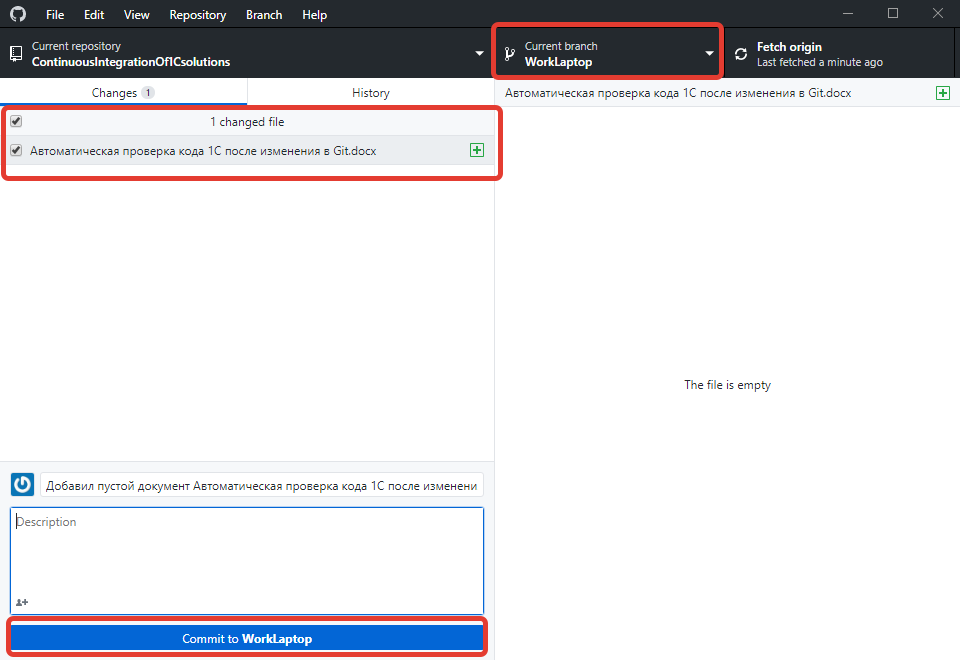
* 1. На обоих устройствах создадим по ветке



* 1. Отправим их в облако командой Publish branch



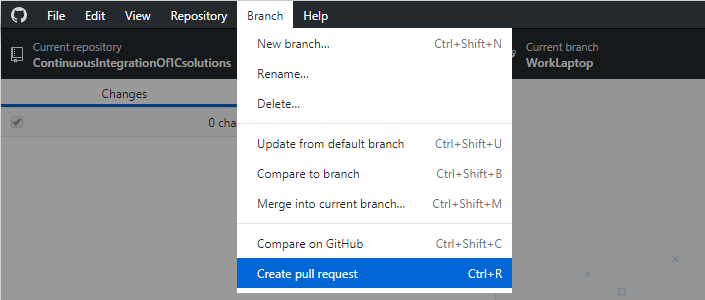
* 1. Добавим по коммиту на обоих устройствах.



* 1. Отправим изменения в облако

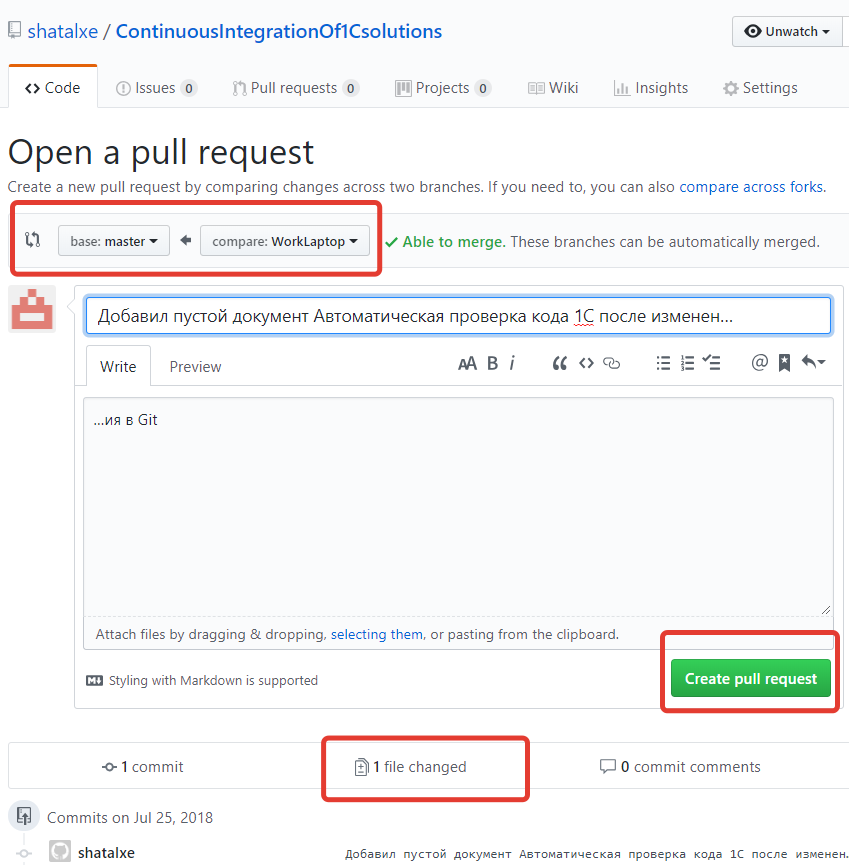


* 1. Отправляем запрос на слияние с веткой Master

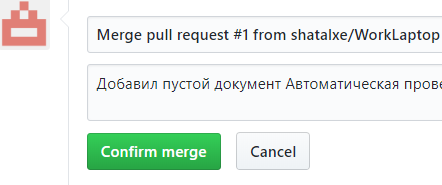
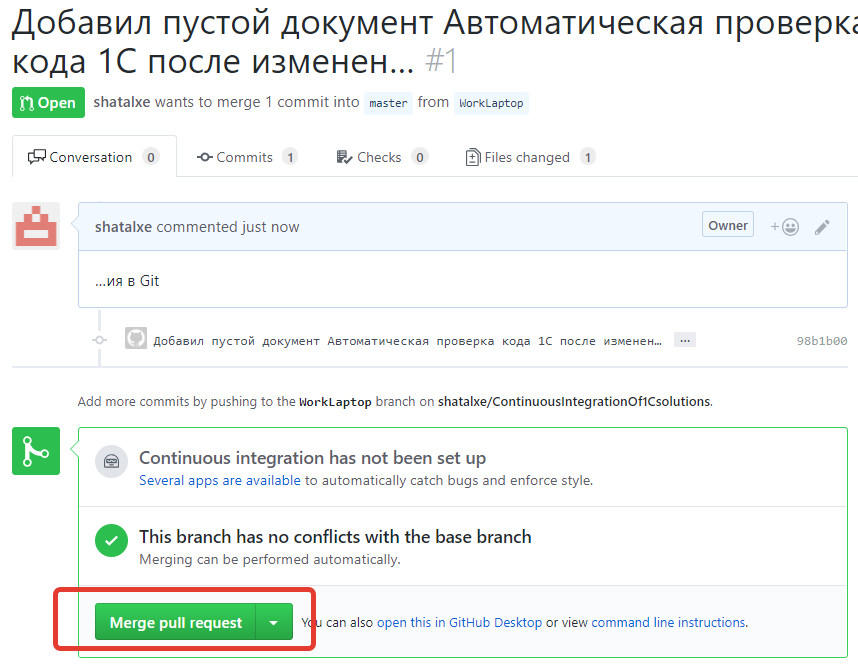


Откроется веб-интерфейс:

* 1. После проверки и устранения вероятных конфликтов жмем Create pull request

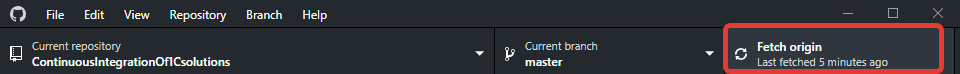


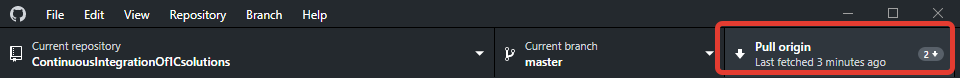
* 1. Затем Merge pull request -> Confirm merge



* 1. Повторяем процедуру слияния на других устройствах.

Возвращаемся к работе на локальном клиенте.

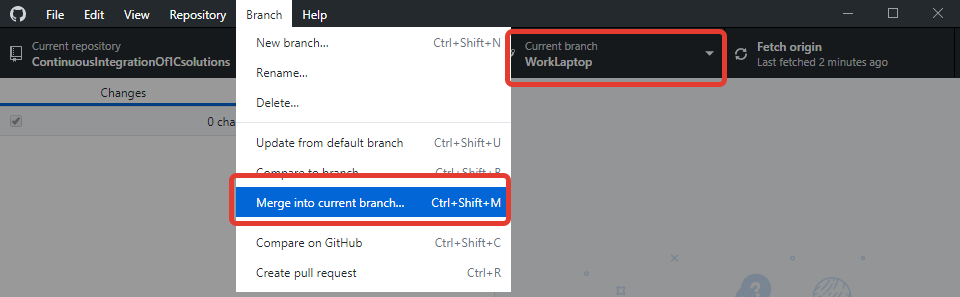
* 1. Обновляем локальный репозиторий командой Fetch origin
  2. Если есть изменения, появится команда Pull origin – выполняем её.



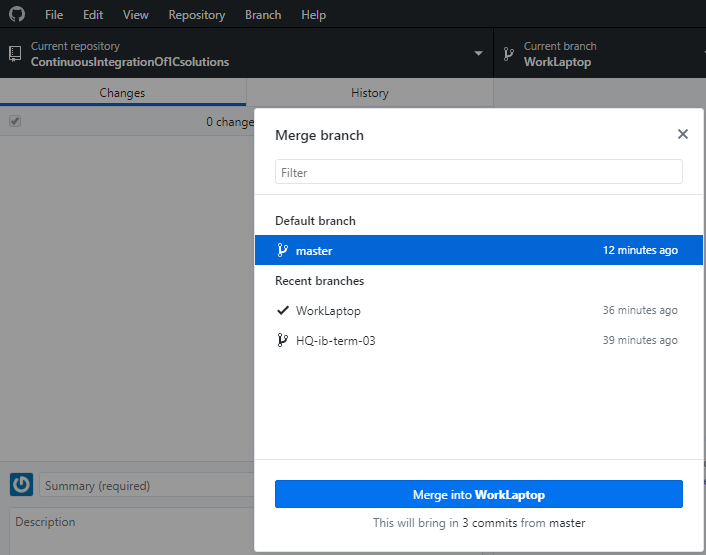
Таким образом, в ветке master собраны все актуальные изменения из других веток.

Теперь обновим остальные ветки из ветки master:

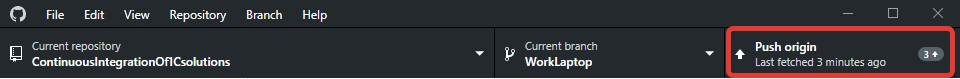
* 1. Убеждаемся, что выбрана нужная нам ветка, жмем Branch/Merge into current branch..



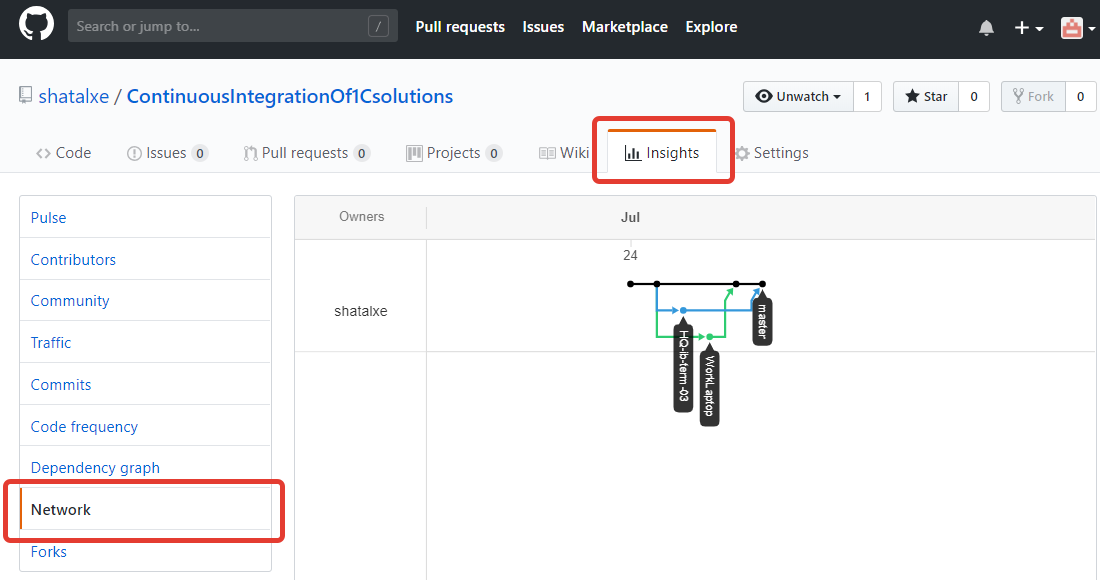
* 1. Выбираем ветку mater, жмем Merge into [имя ветки] (если ветки равны, то команда не будет доступна)



* 1. Отправляем результат слияния в глобальный репозиторий



Изменения глобального репозитория в графическом виде можно посмотреть в веб-интерфейсе



# Многопользовательский совместный

# 4. Многопользовательский с архитектором

# 5. Многопользовательский, 2 общих ветки (продакшен и препродакшен)