Как разместить код на GitHub: Шаг за шагом

Описание

Внедрение контроля версий было бы простым делом, если бы вы работали только в одиночку с локальным репозиторием. Однако во многих профессиональных проектах дело обстоит иначе. Вместо этого команда объединяет ресурсы в удаленную репозиторию, используя такой хост, как GitHub. Поэтому важно научиться выполнять push на GitHub, поскольку это необходимо делать в рамках рабочего процесса.

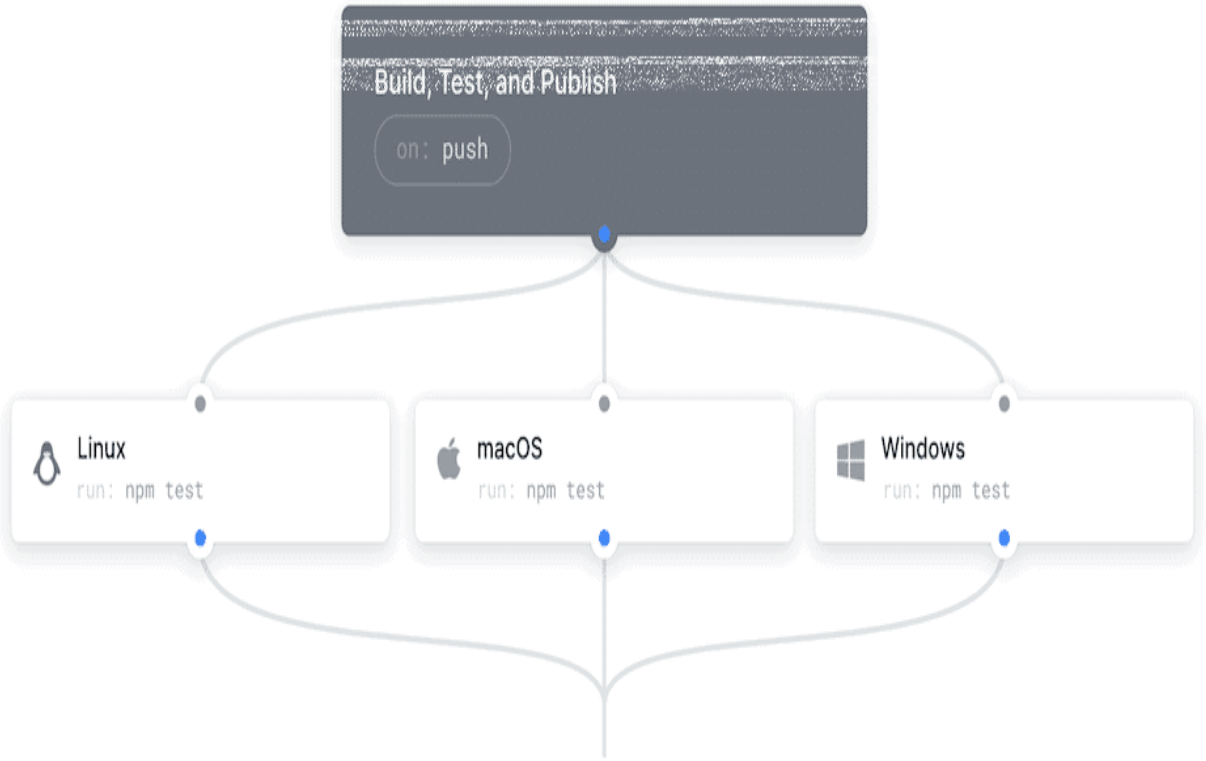
В этой статье мы расскажем о том, как выполнять push на GitHub с помощью командной строки. Кроме того, мы рассмотрим специализированные приложения, которые помогут сделать этот процесс более плавным. В конце статьи мы поговорим о том, почему следует изучать pull request и как они связаны с продвижением кода. Для начала давайте рассмотрим, как использовать GitHub для своих проектов.

Как разработчик или программист будет использовать GitHub

GitHub – незаменимый инструмент для разработчиков и программистов во всем мире по целому ряду причин. Он позволяет хранить код в централизованном месте, что облегчает доступ к нему и расширяет возможности сотрудничества с другими разработчиками.

Кроме того, вы можете отслеживать изменения, вносимые в код, и при необходимости возвращаться к предыдущим версиям. Кроме того, GitHub предоставляет инструменты для работы с проблемами и ошибками, что значительно упрощает сопровождение кодовой базы. Совместная работа – одна из ключевых причин, по которой вы можете использовать GitHub в качестве удаленной системы контроля версий (VCS). Она позволяет обмениваться кодом, отслеживать изменения и сотрудничать в решении проблем без лишних хлопот. Это повышает эффективность работы и улучшает качество кода.

GitHub также позволяет безболезненно управлять несколькими версиями кодовой базы, отслеживать изменения и при необходимости откатывать их назад. Крупные проекты и совместная работа с открытым исходным кодом – это лишь два способа, с помощью которых GitHub покажет свою ценность. Даже простые сценарии использования могут быть идеальными. Например, можно хранить код для проекта веб-разработки и отправлять удаленные обновления по мере внесения изменений. Кроме того, проекты непрерывной интеграции/непрерывного развертывания (CI/CD) выиграют от автоматизации в виде GitHub Actions на этапах сборки.



В целом GitHub и другие удаленные хосты ВКС, такие как GitLab, представляют собой платформу для совместной работы, контроля версий и других рабочих процессов разработки. Это позволяет оптимизировать процесс разработки и повысить качество кода. В связи с этим вам будет полезно узнать, как выполнять push на GitHub, поскольку эти знания пригодятся вам практически каждый день.

Как выполнить push на GitHub из терминала (командной строки)

В оставшейся части этой статьи мы расскажем вам о том, как выполнять push на GitHub. Этот процесс прост для понимания и выполнения. Однако необходимо предварительно убедиться в том, что вы настроили свой проект, в противном случае вы столкнетесь с ошибками. В первом разделе мы рассмотрим, какие инструменты и навыки вам понадобятся, а затем подробно остановимся на самом процессе.

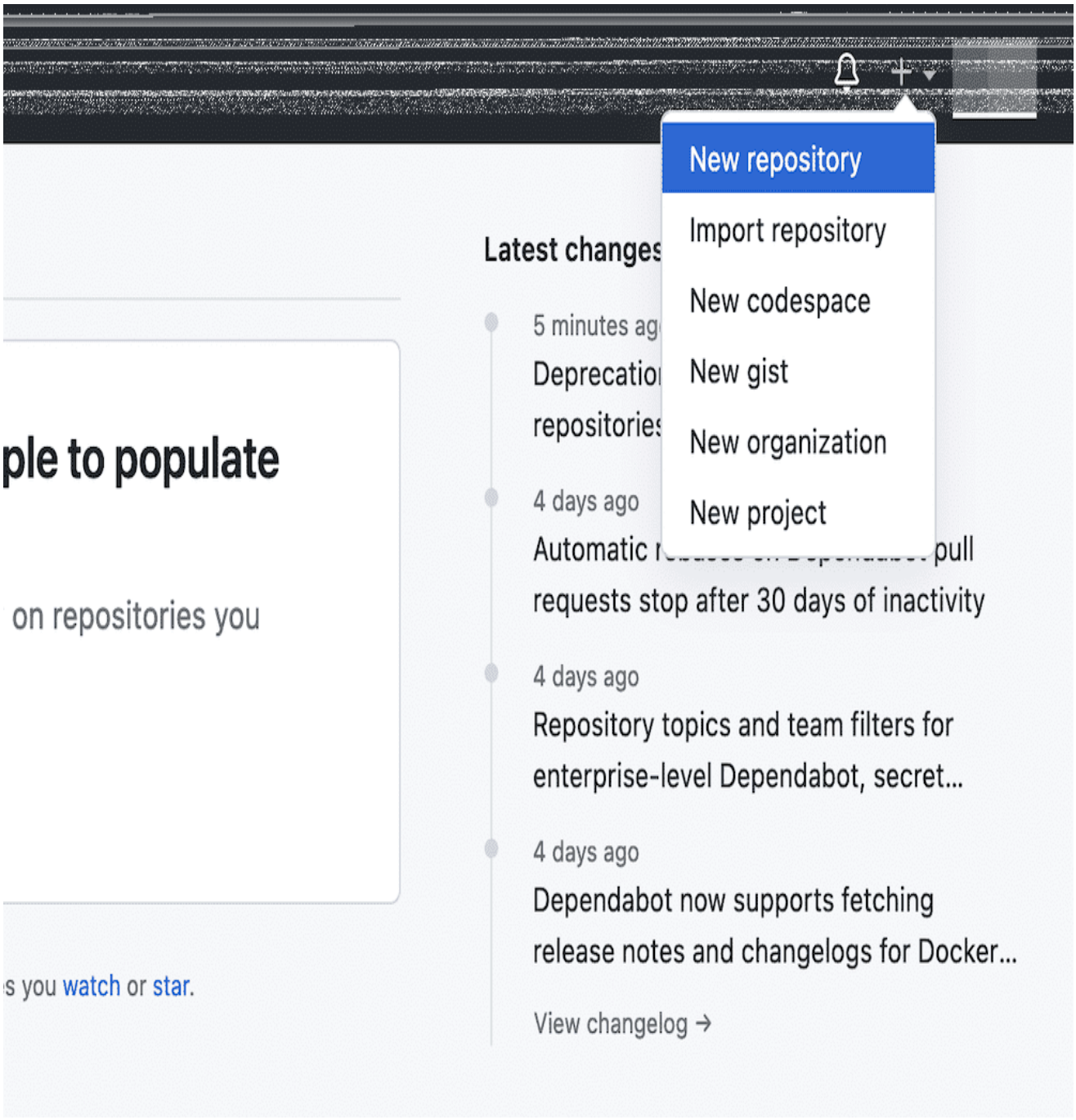
Что нужно для размещения проекта на GitHub

Очень важно настроить свой проект таким образом, чтобы он поддерживал использование удаленного репозитория и интегрировал push в рабочий процесс. Для этого, прежде всего, необходим Git-репозиторий – “репозиторий”, в котором будет храниться ваш код. Считайте, что это папка, содержащая файлы, связанные с вашим проектом. Весь процесс управления версиями начинается в локальной среде на вашем компьютере. Точные шаги для этого будут описаны позже, но, возможно, вы уже обладаете этими знаниями (или знаете, где их можно получить). Также вам потребуется учетная запись GitHub. На самом деле, можно использовать и другой онлайновый хост ВКС, например GitLab, BitBucket, Buddy и т.д. Инструкции, которые мы приводим здесь, в большинстве случаев можно переносить и на другие платформы. Однако сравнение этих хостов выходит за рамки данной статьи.

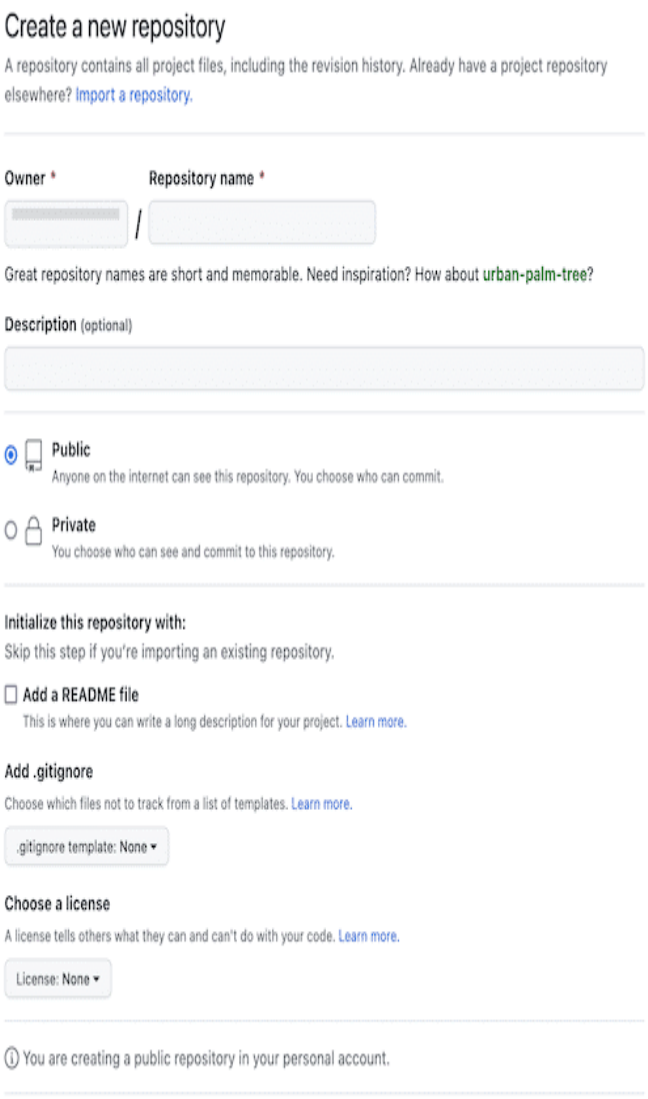
Для отправки кода на GitHub можно использовать как командную строку, так и графический интерфейс пользователя (GUI). Основная часть нашей статьи будет посвящена работе с командной строкой, но есть раздел и об использовании графического интерфейса, поскольку некоторые из них пользуются популярностью. Однако следует иметь в виду, что каждый графический интерфейс может иметь свой процесс отправки на GitHub, поэтому для получения максимальной отдачи от него необходимо придерживаться конкретного приложения. Наконец, убедитесь, что у вас есть правильный доступ к репозиторию. Документация GitHub содержит исчерпывающую информацию по этому вопросу, и вам следует обратить внимание либо на маркеры доступа HTTPS, либо на доступ Secure Shell (SSH). Без этого вы не сможете работать!

Создание репозитория GitHub

Первым шагом является создание нового онлайн-репозитория в GitHub. Это можно сделать из командной строки, но не менее просто сделать это через веб-браузер. После входа в систему или регистрации на GitHub перейдите в правый верхний угол экрана и найдите выпадающее меню Plus рядом с аватаром вашего профиля. Если вы откроете его, то увидите несколько опций, в том числе New repository:



После этого откроется страница **Create a New Repository**. Здесь будет показан ряд параметров, которые помогут настроить удаленное хранилище:



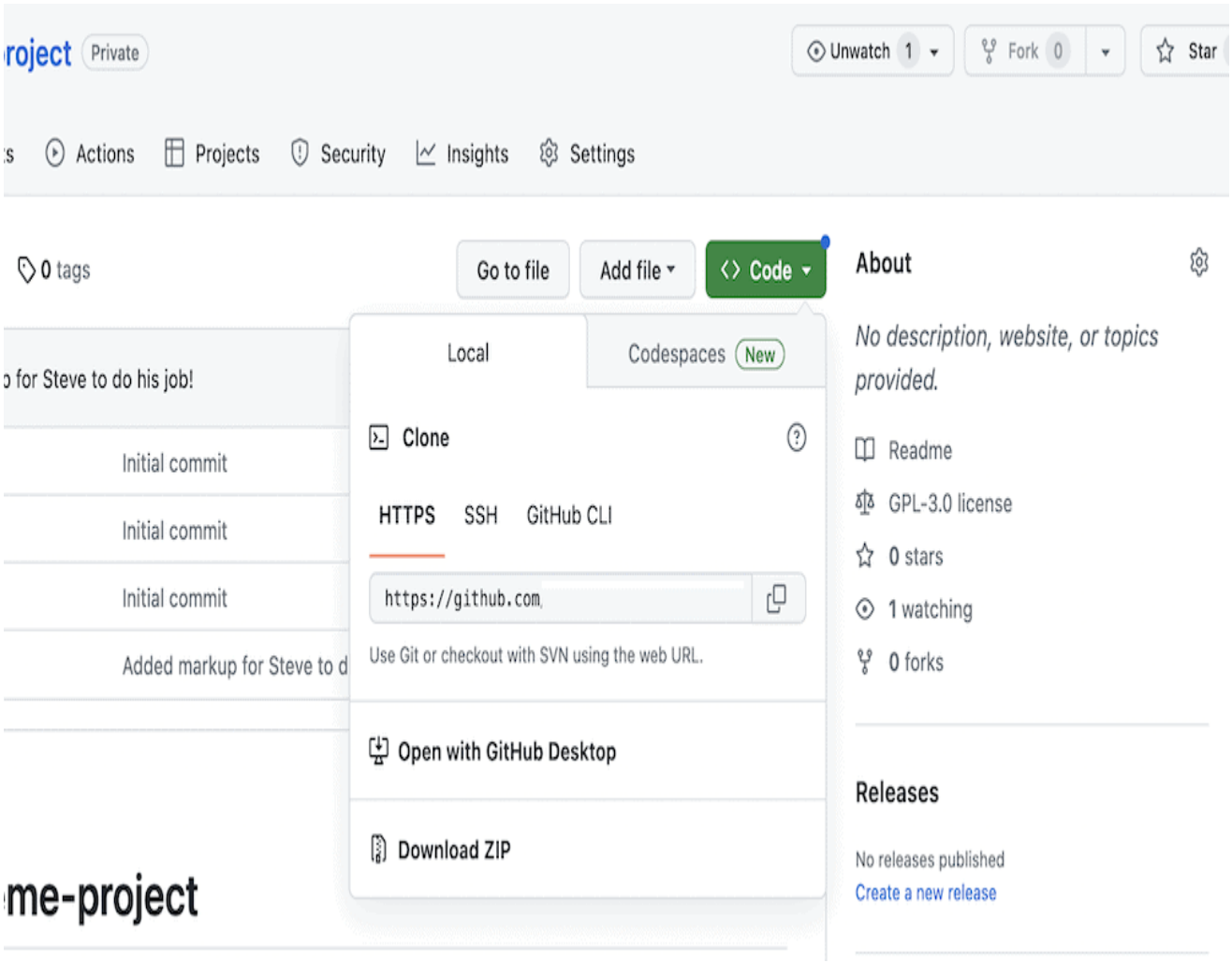
Параметры, которые вы здесь зададите, будут зависеть от потребностей вашего проекта. Однако если у вас уже есть локальный репозиторий, который вы хотите отправить на GitHub, мы будем ставить как можно меньше галочек, чтобы сохранить паритет между локальным и удаленным репозиториями. Далее нажмите кнопку Create repository, и GitHub настроит все под себя. На этом этапе вы попадаете на домашнюю страницу репозитория с инструкциями по настройке нового локального репозитория, связанного с удаленным, с помощью командной строки.

Путь, по которому вы пойдете, зависит от того, нет ли у вас еще репозитория или вы хотите клонировать содержимое существующего проекта. Если вы уже инициализировали Git и заполнили локальное репозиторий, то вам не нужно выполнять ничего из шага 2. Вместо этого можно сразу перейти к третьему шагу,

где мы рассмотрим процесс отправки кода на GitHub из локальной репозитории.

Клонирование удаленного Git-репозитория

Если у вас еще нет локального репозитория, то версия на GitHub будет единственной. Лучший способ синхронизировать оба места – использовать команду git clone на своем компьютере. Однако вам потребуется URL-адрес вашего репозитория. Чтобы получить его, перейдите в репозиторий на GitHub и найдите над списком файлов зеленую выпадающую опцию Code:



Если вы не видите этого, то, скорее всего, у вас еще нет заполненного репозитория. Обратите внимание, что URL-адрес репозитория можно скопировать из синего окна быстрой настройки в верхней части экрана. Просто переключитесь на HTTPS с помощью кнопок и скопируйте URL.

Однако мы предпочитаем генерировать файл .gitignore, поскольку он в любом случае будет вам необходим. Вы можете использовать сайт gitignore.io для поиска используемых вами инструментов и создания на его основе полного файла .gitignore для загрузки в репозиторий:

В любом случае, как только вы сможете открыть выпадающий список Code, в нем будут показаны URL-адреса вашего репозитория. Будут предложены варианты HTTPS, Secure Shell (SSH) и другие. Однако проще всего использовать HTTPS-адрес. Чтобы скопировать его в буфер обмена, можно щелкнуть на маленьком значке Copy рядом с URL. Далее перейдите в Терминал или приложение командной строки и выполните следующие действия:

```
git clone <full-github-url>
```

После выполнения команды Git скопирует удаленное репо в локальную среду.

Инициализация Git’а в локальной папке проекта

Для ситуаций, когда у вас еще нет локальной версии удаленного репозитория, вам необходимо инициализировать ее. Большая часть работы будет выполняться

локально, с регулярным переносом изменений на удаленный сервер. Ниже описаны шаги:

- Сначала перейдите по cd в папку, которую вы хотите использовать для своего проекта.
- Затем выполните команду git init. Это инициализирует Git в локальной папке проекта и создаст скрытый каталог .git.
- Добавьте файл .gitignore в корень локальной папки проекта, поскольку некоторые изменения, связанные с системными файлами, нежелательно вносить в сценарий.

На этом этапе необходимо проиндексировать текущие файлы. Это можно сделать обычным способом, используя git add, а затем зафиксировать изменения:

```
git add .
```

```
git commit -m “Initial Commit”
```

```
git branch -M trunk
```

Последняя строка меняет основную ветку на другую по вашему выбору, если вы еще не перешли с master. Последний вариант проблематичен, так как имеет негативную коннотацию с рабством, поэтому рекомендуется его изменить. Здесь мы использовали trunk, но допустимо и main. Если вы знаете, что эта строка вам не нужна, ее можно опустить. На этом этапе вы готовы к тому, чтобы узнать, как сделать push на GitHub!

Добавление нового удаленного источника и отправка кода на GitHub

После создания нового удаленного репозитория на GitHub необходимо добавить новый “удаленный источник” в локальный репозиторий. По сути, это ссылка на ваш удаленный репозиторий, чтобы ваш локальный репозиторий знал, куда отправлять изменения из восходящего потока. Для этого введите в терминале следующую команду:

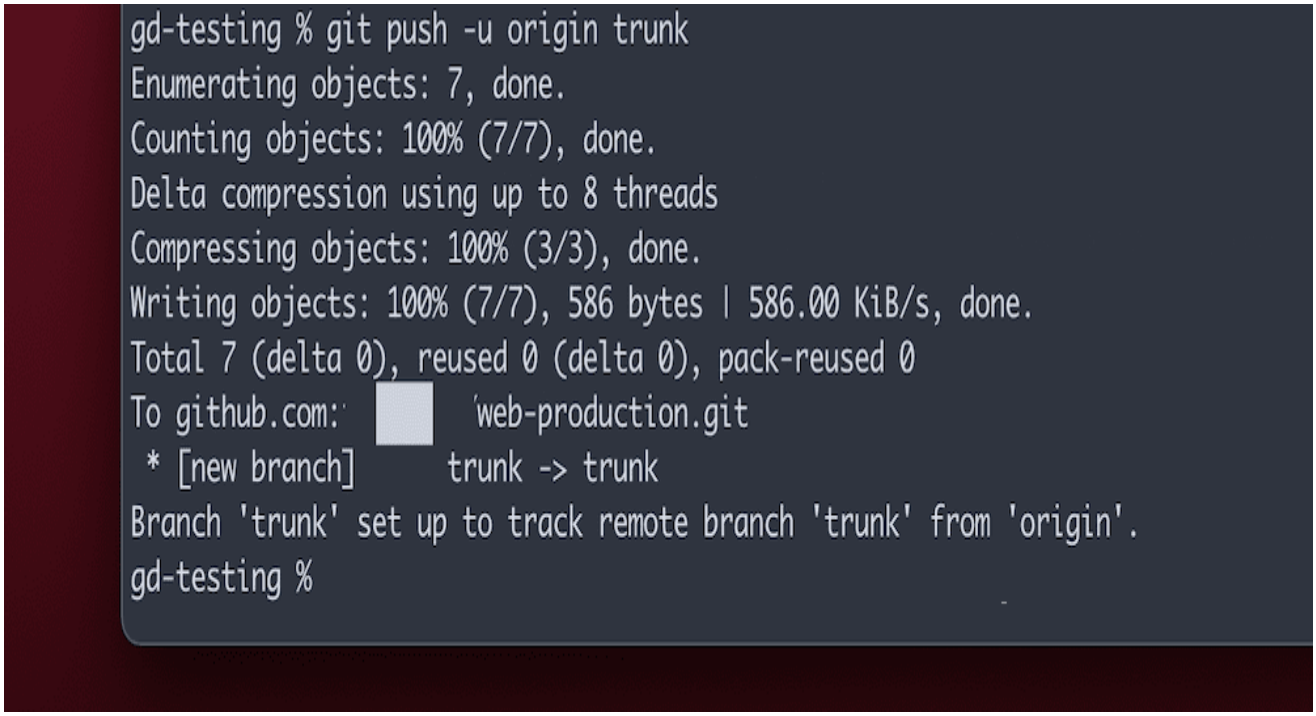
```
git remote add origin <github-url>
```

В техническом смысле добавляемый пульт может иметь любое название. Однако большинство называют его “origin”, поскольку вы используете только один

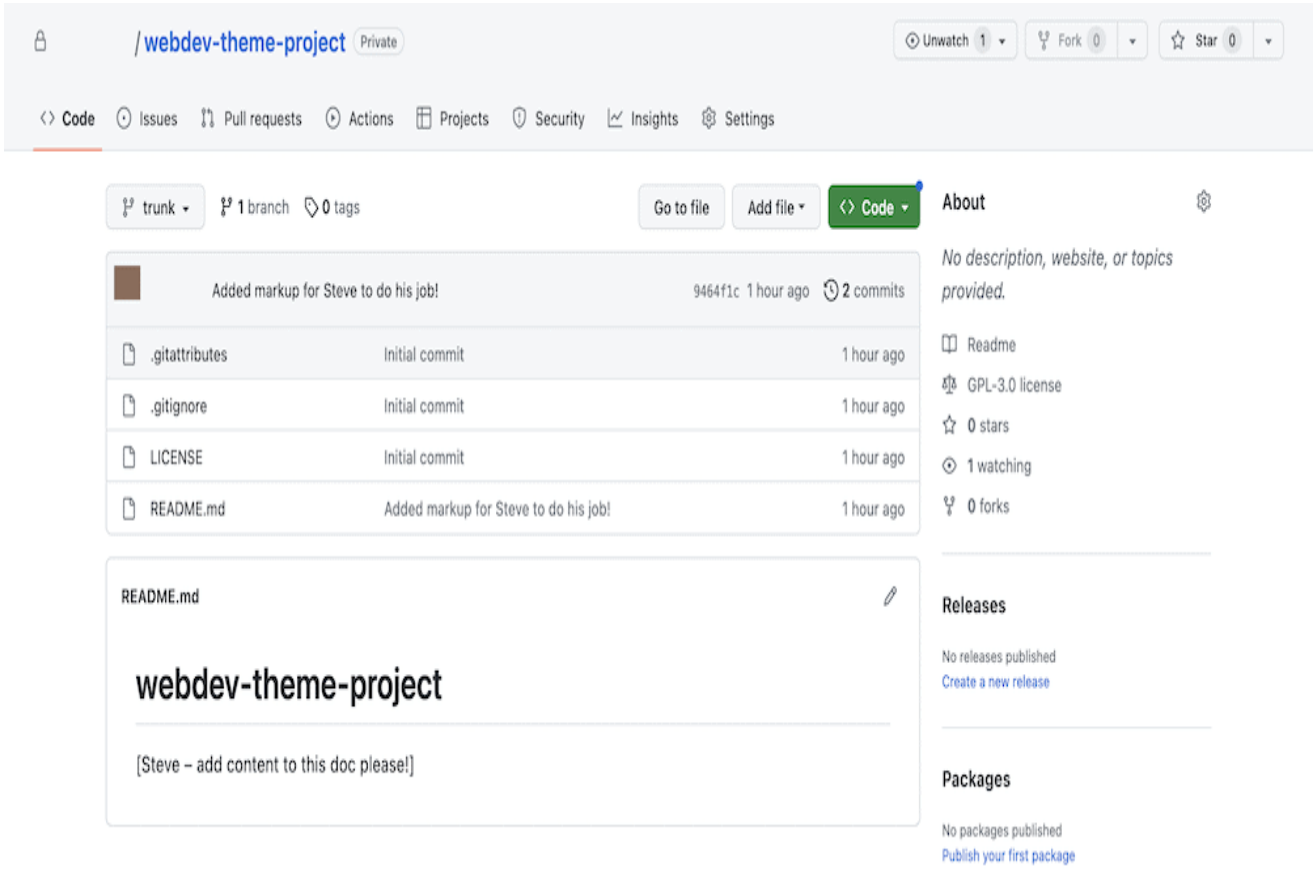
добавляемый пульт, и это обеспечивает абсолютную ясность. На этом этапе можно выполнить push на GitHub, используя следующее:

```
git push -u origin trunk
```

Эта команда переместит ваш код в новое удаленное хранилище с именем “origin” и установит ветвь upstream в “trunk”. При необходимости вы также можете добавить любую ветку в удаленное хранилище.



После завершения процесса необходимо убедиться в том, что толчок прошел успешно. Это можно сделать несколькими способами. Например, можно зайти в репозиторий на GitHub и проверить, внесены ли изменения:

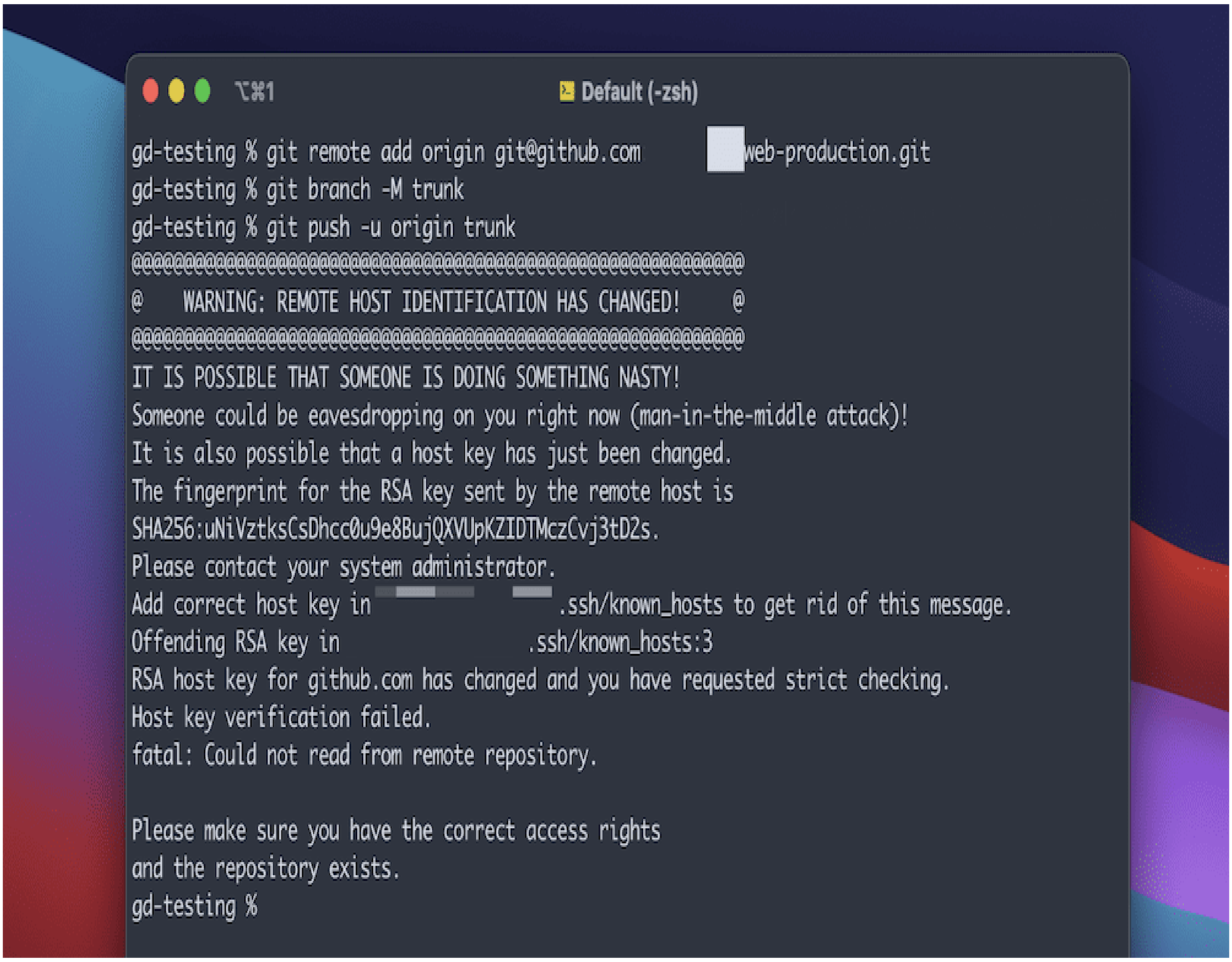


Однако можно также запустить git log из командной строки:

Эта команда отображает все фиксации для вашего репозитория, включая ту, которую вы только что вытолкнули. Таким образом, если фиксация находится в журнале, значит, толчок был успешным.

Как выполнить push на GitHub без ошибок

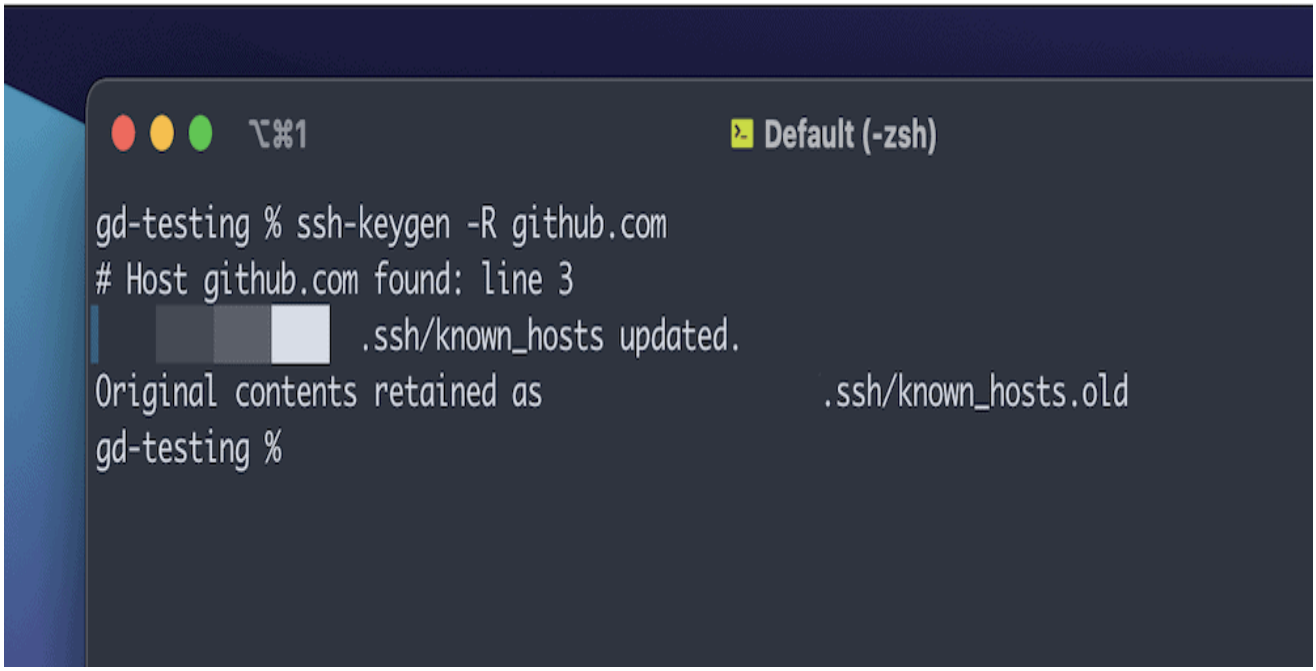
В некоторых случаях при попытке отправки кода на GitHub может возникнуть ошибка:



Это происходит, когда у вас уже есть защищенное соединение с GitHub через старый проект, но используется старый ключ RSA. У нас есть руководство по устранению этой проблемы в целом. Однако для исправления этой проблемы конкретно для GitHub можно выполнить следующее:

```
ssh-keygen -R github.com
```

Это приведет к обновлению файла ‘known hosts’, после чего появится подтверждающее сообщение:



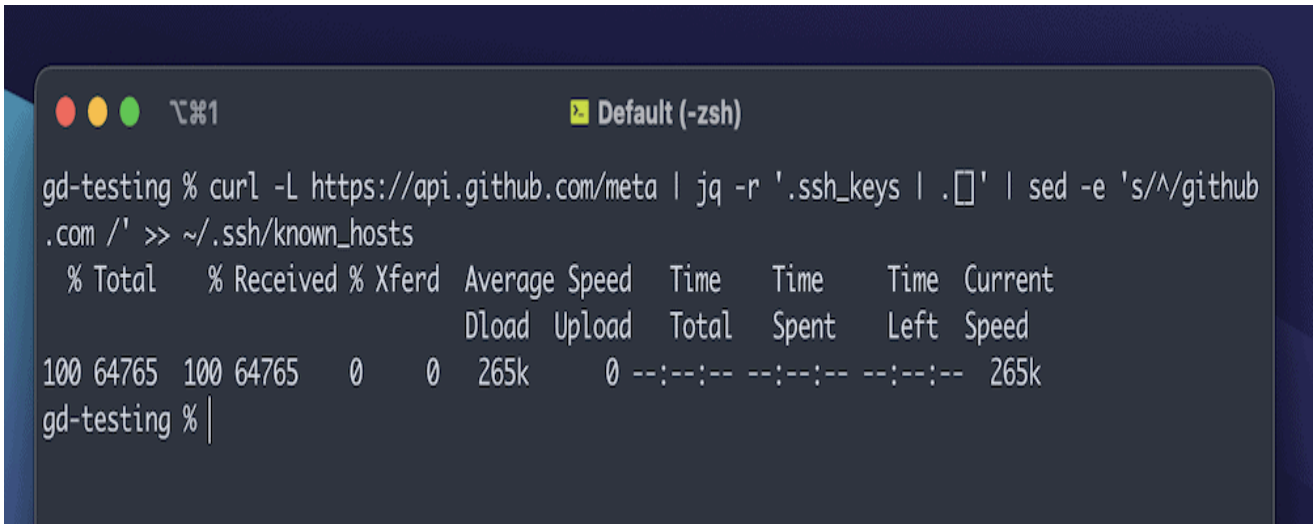
Отсюда выполните следующие действия, чтобы добавить новый RSA-ключ в файл известных хостов:

```
curl -L https://api.github.com/meta | jq -r '.ssh_keys | .[]' | sed -e 's/^/gi
```

На самом деле, здесь также может возникнуть ошибка, связанная с пакетом jq. В этом случае, в зависимости от операционной системы, можно выполнить одно из следующих действий:

- **Windows:** curl -L -o /usr/bin/jq.exe https://github.com/stedolan/jq/releases/latest/download/jq-win64.exe
- **Mac:** brew install jq
- **Linux:** apt-get update | apt-get -y install jq

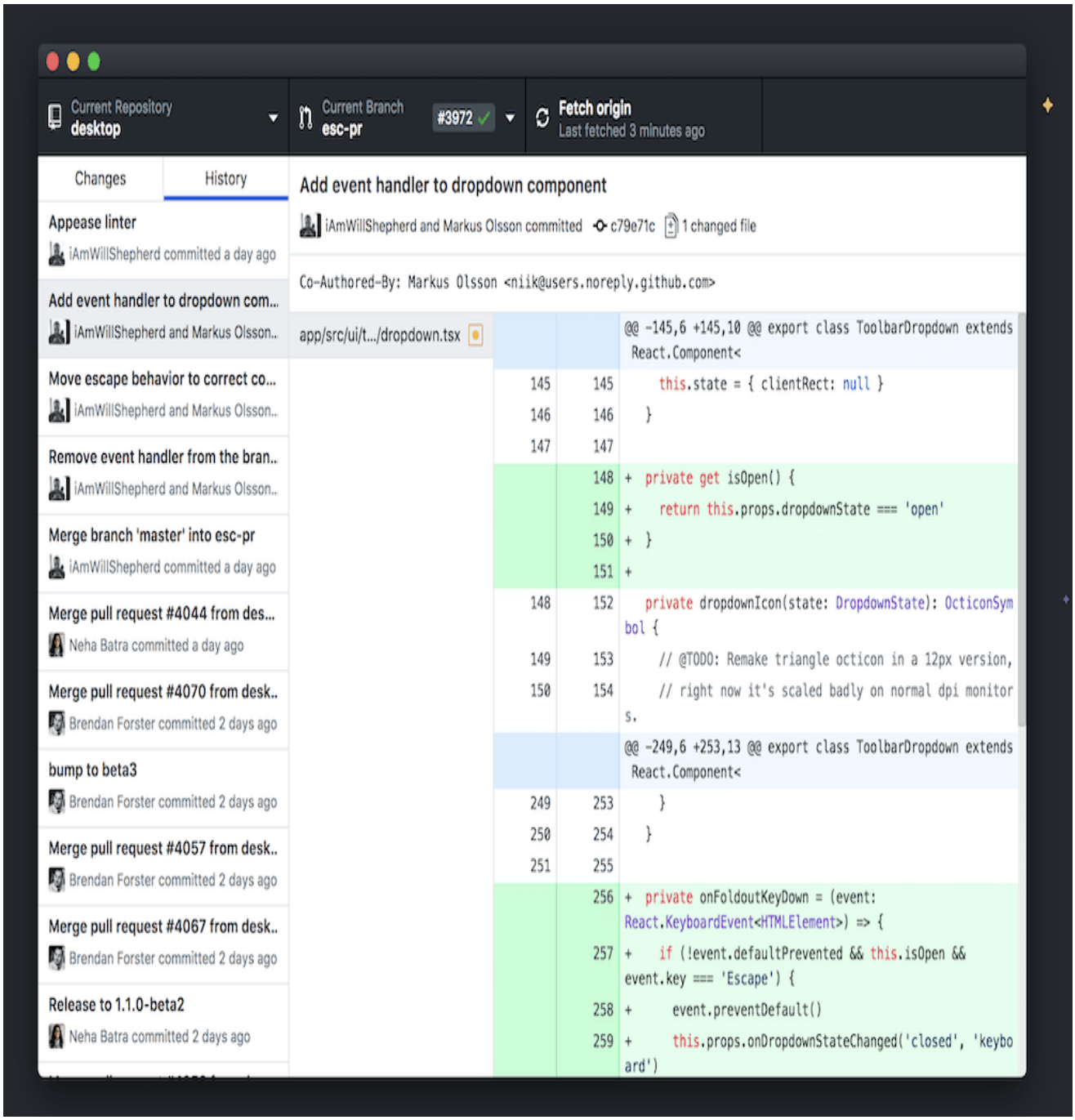
После установки снова выполните команду и дождитесь ее завершения:



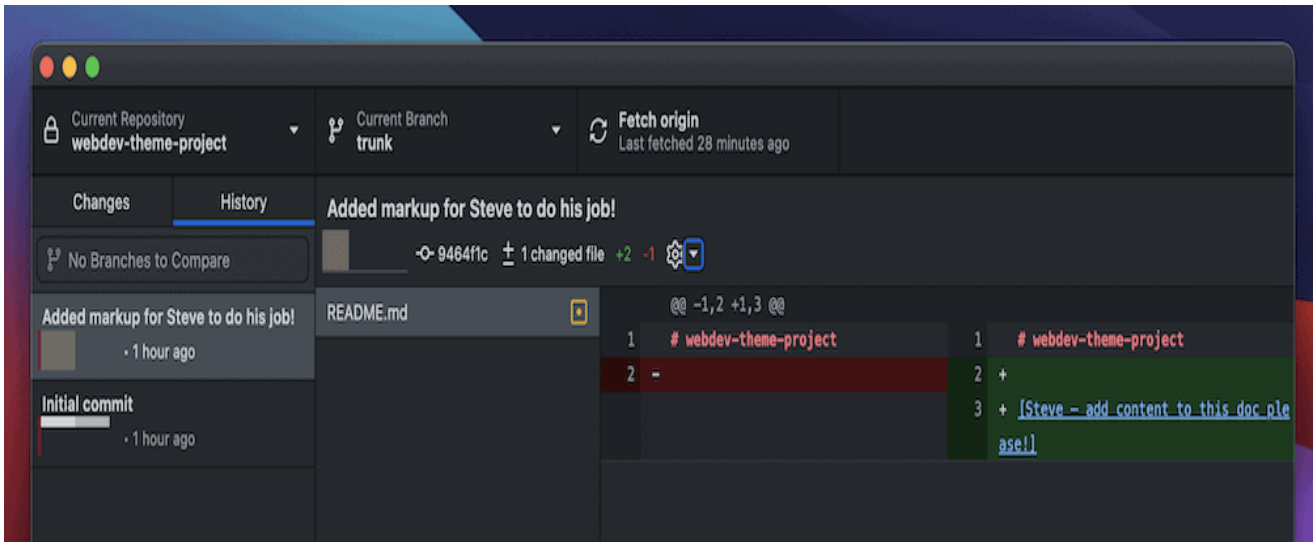
Наконец, можно снова выполнить команду git push, и на этот раз процесс должен завершиться. Если этого не произошло, то проблема, скорее всего, связана с неправильными SSH-ключами или даже с тем, что необходимо настроить безопасные соединения с помощью специальных ключей. У нас есть полное руководство по генерации новых SSH-ключей для GitHub, а также подробная документация GitHub.

Использование графического интерфейса для отправки кода на GitHub

Хотя процесс отправки кода на GitHub прост после его настройки, существует множество шагов, предостережений и подпроцессов, которые необходимо учитывать. Графический интерфейс пользователя может упростить этот процесс. Например, вы получаете все функции командной строки, но с более приятным интерфейсом (в некоторых случаях с возможностью перетаскивания). Более того, часто проще визуализировать изменения и управлять ими с помощью графического интерфейса, особенно если вы не знакомы с инструментами командной строки. Если вы знаете, что вам никогда не понадобится использовать приложение с графическим интерфейсом для доступа к другому удаленному узлу VCS, GitHub Desktop может стать идеальным вариантом.



Приложение позволяет создавать и управлять репозиториями, фиксировать изменения и отправлять их на GitHub всего несколькими щелчками мыши. Оно работает по принципу drag-and-drop, а также имеет инструмент “визуальный дифф”, упрощающий выявление изменений в коде между версиями:

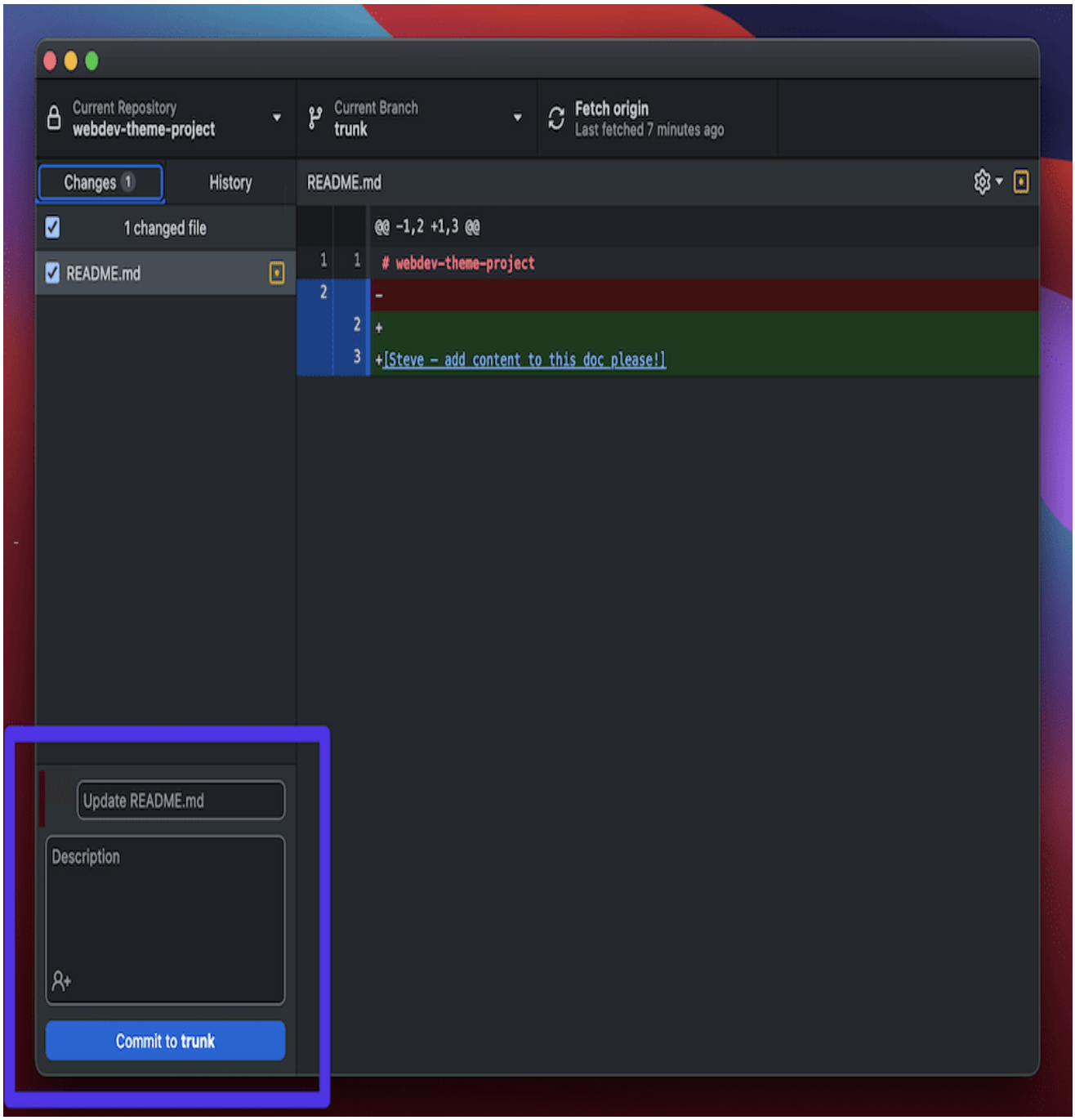


Sourcetree – еще один популярный графический интерфейс Git, который мы рассматриваем в нашем обзоре инструментов для веб-разработки. Хотя предпочтительным VCS является BitBucket (из-за того, что это продукт Atlassian), вы можете использовать этот инструмент и с GitHub. Инструмент разрешения конфликтов слияния также удобен и является одной из наиболее ярких особенностей.

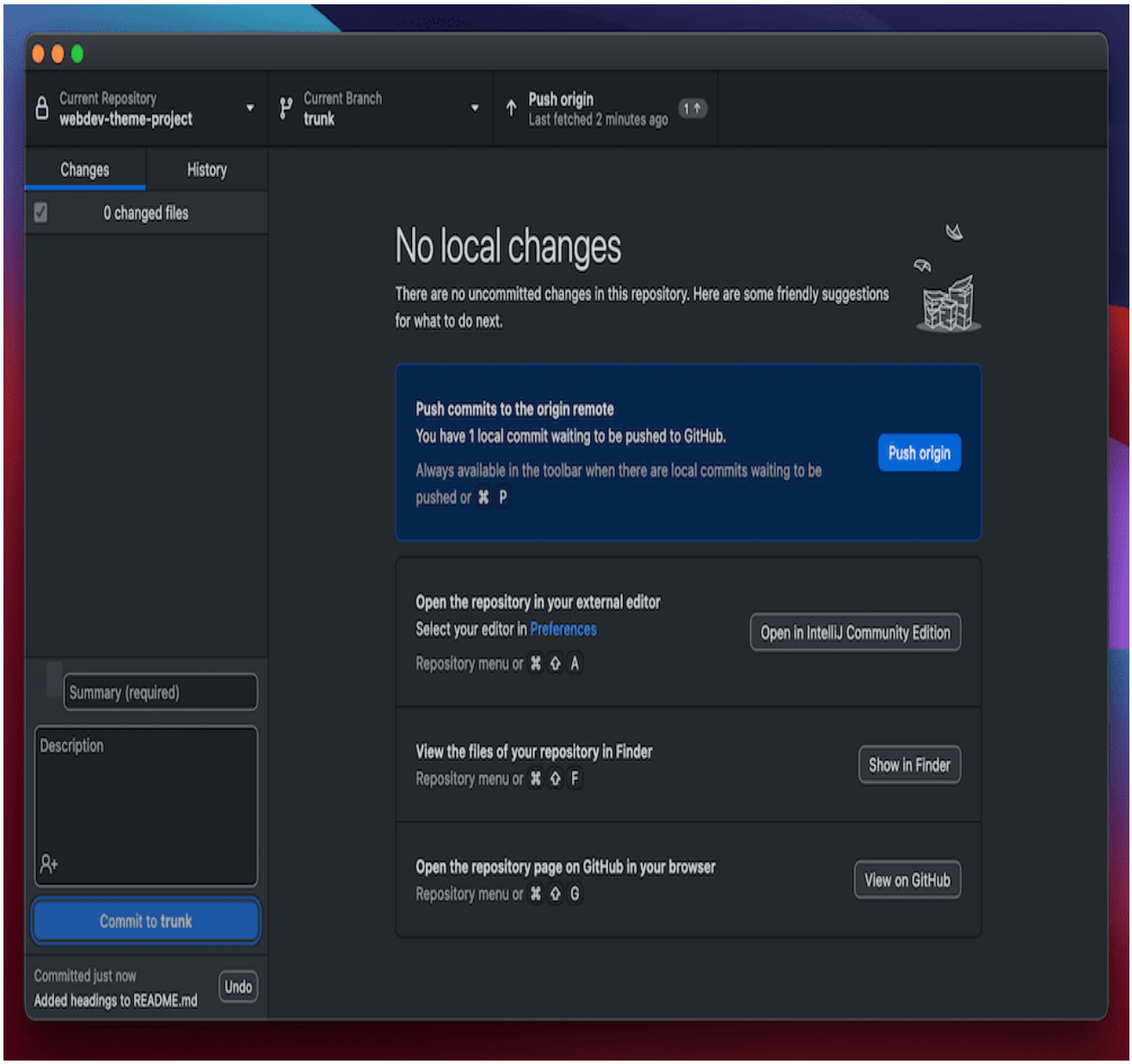
GitKraken – это, пожалуй, лучшее приложение с графическим интерфейсом, предлагающее разумную бесплатную версию для локальных и публичных репозиториев. Оно поддерживает все основные хосты VCS – конечно, GitHub, но также GitLab и BitBucket, и другие. Нам нравится его визуальное представление вашего репозитория, а также продуманная функциональность для команд.

Использование GitHub Desktop для отправки на GitHub

Хотя процесс работы с каждым приложением будет несколько отличаться, GitHub Desktop очень удобен. Вы работаете на одном экране, на котором используются различные окна и панели. После внесения изменений в файл (который можно открыть в выбранном редакторе с помощью контекстного меню, вызываемого правой кнопкой мыши) выполняется фиксация с помощью небольшого виджета на экране:



Этот коммит станет частью секции Push Origin на верхней панели инструментов: Если у вас нет изменений для фиксации, то также появится уведомление о том, что вы хотите отправить свои коммиты на удалённое хранилище origin:



Это решение, позволяющее одним щелчком мыши перенести изменения в репозиторий GitHub. Весь рабочий процесс выполняется быстро, безболезненно и просто.

Заключение

GitHub – незаменимый инструмент для разработчиков и программистов. Он представляет собой централизованный репозиторий для хранения, отслеживания и совместной работы над кодом. Научившись отправлять свой код на GitHub из локального хранилища, вы сможете присоединиться к этой совместной работе. Использование командной строки упрощает процесс отправки кода на GitHub и

требует выполнения всего нескольких команд после того, как все настроено. Однако, возможно, вам стоит рассмотреть возможность использования специализированных приложений с графическим интерфейсом, таких как GitKraken или GitHub Desktop. Они позволяют исключить командную строку из уравнения и выполнять практически все необходимые задачи с помощью Git из привычного интерфейса.

Дата Создания
10.07.2023