Лабораторная работа №2

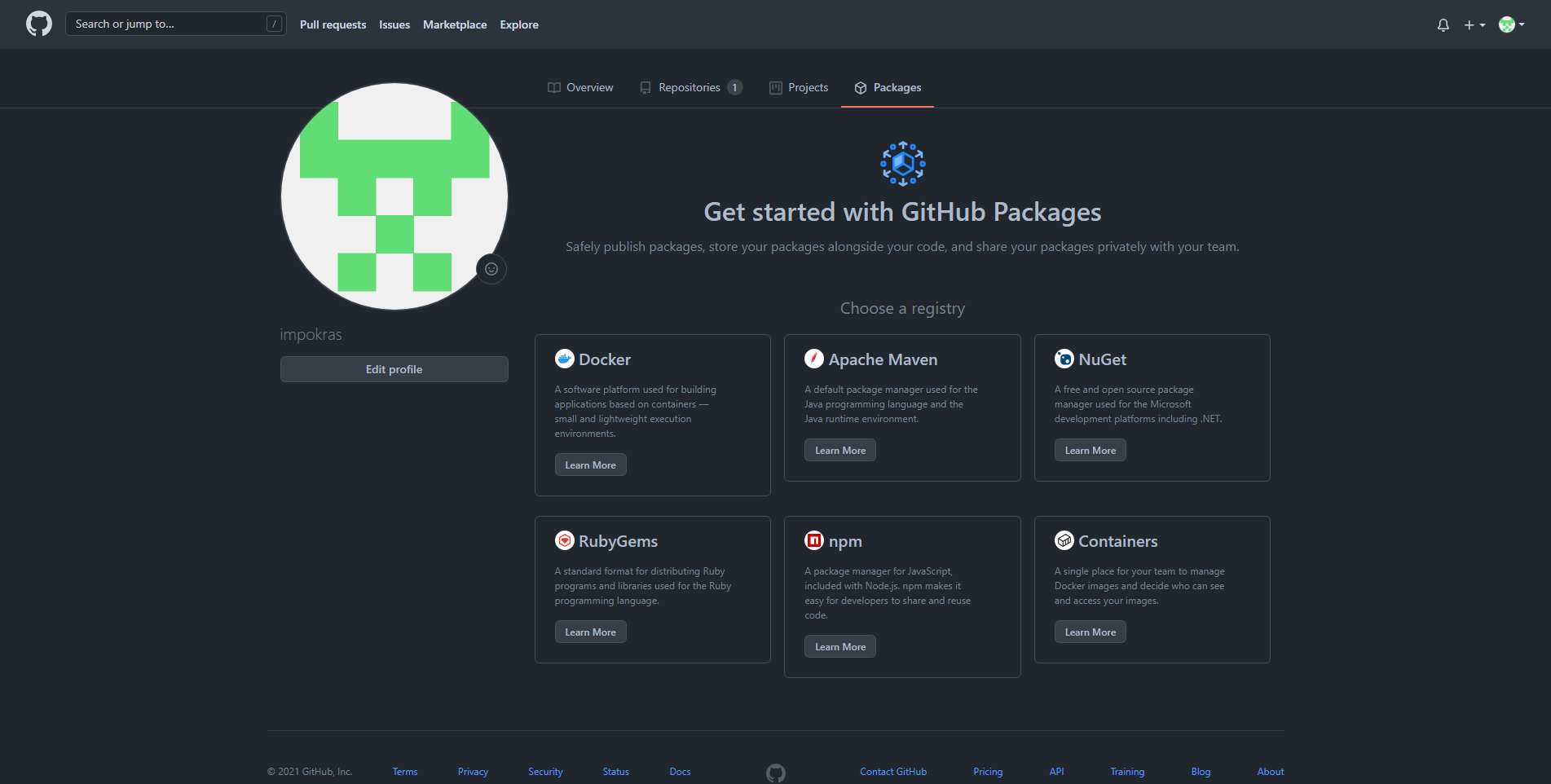
Покрас Илья Михайлович

## Цель работы

Изучить идеологию и применение средств контроля версий.

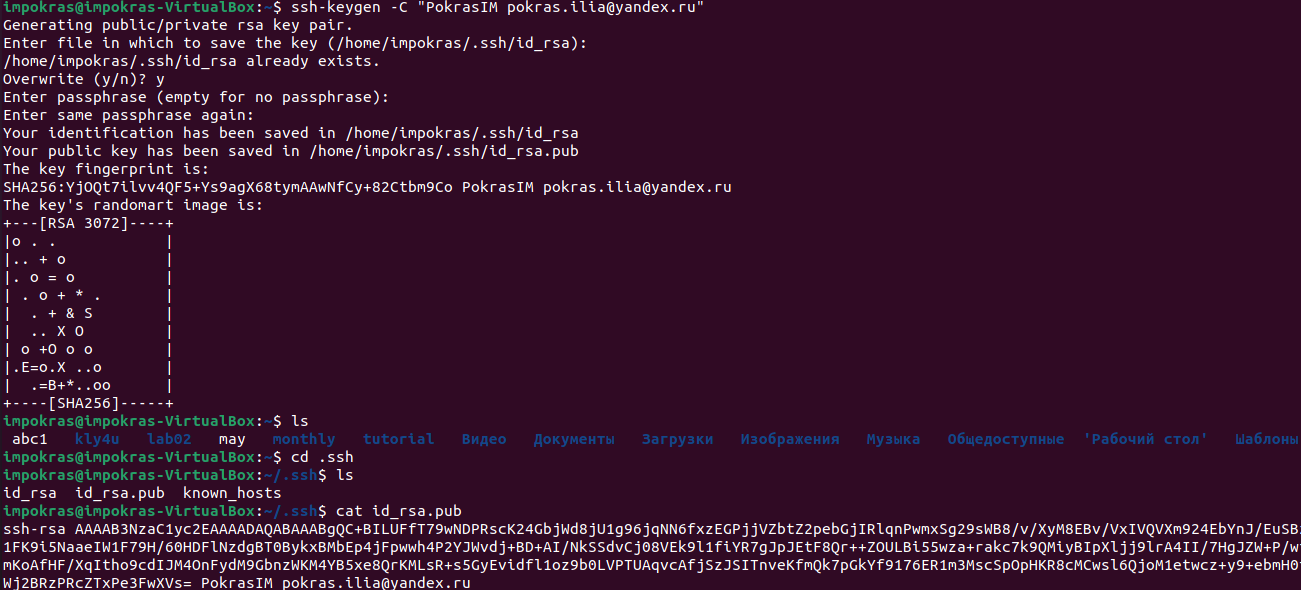
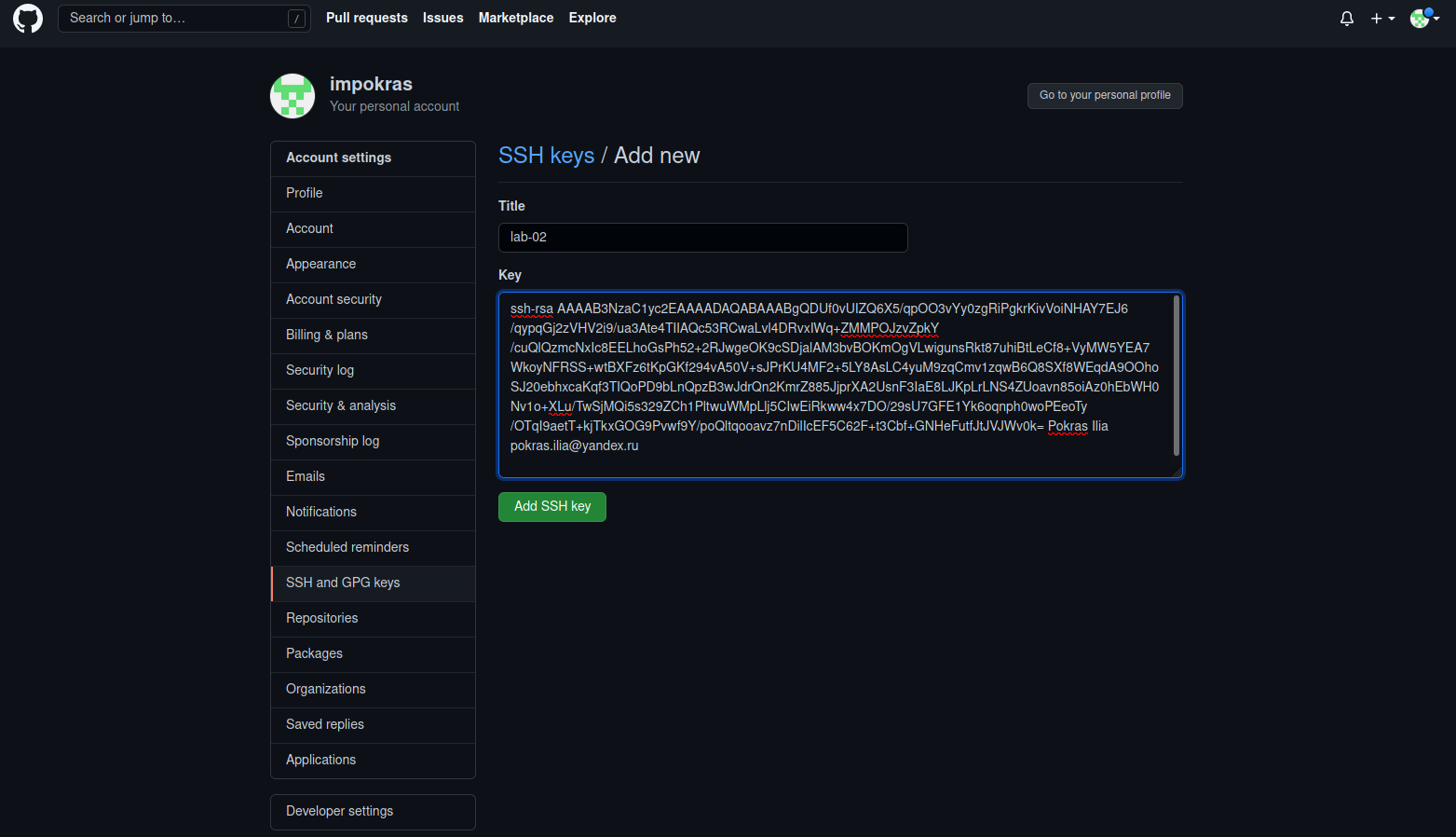
# Последовательность выполнения работы

## Для начала работы я создал учетную запись в github.

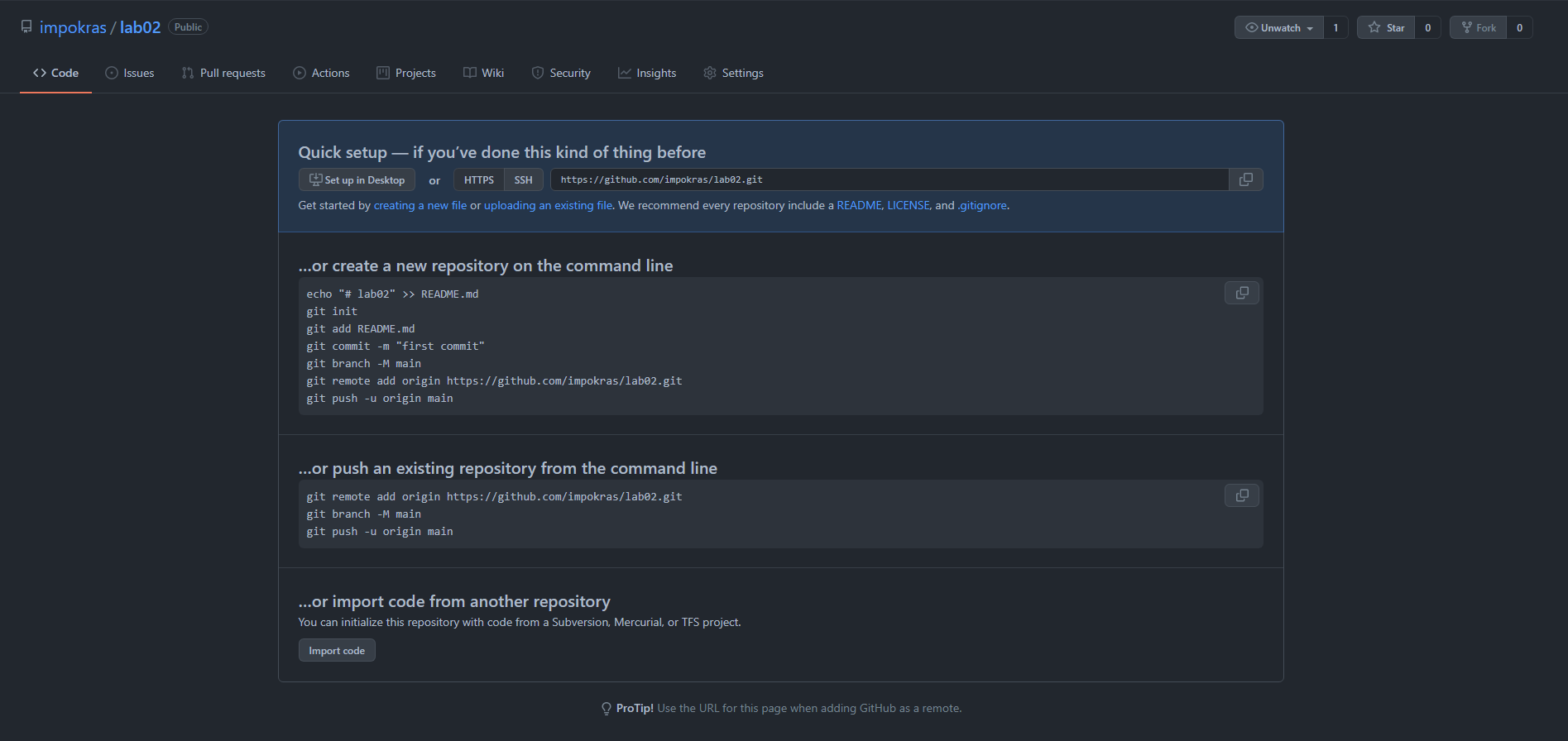


profile

## Для работы с сервером репозиториев я создал ssh-ключ с помощью команды ssh-keygen. После я скопировал его данные и через github вставил их в поле ‘key’

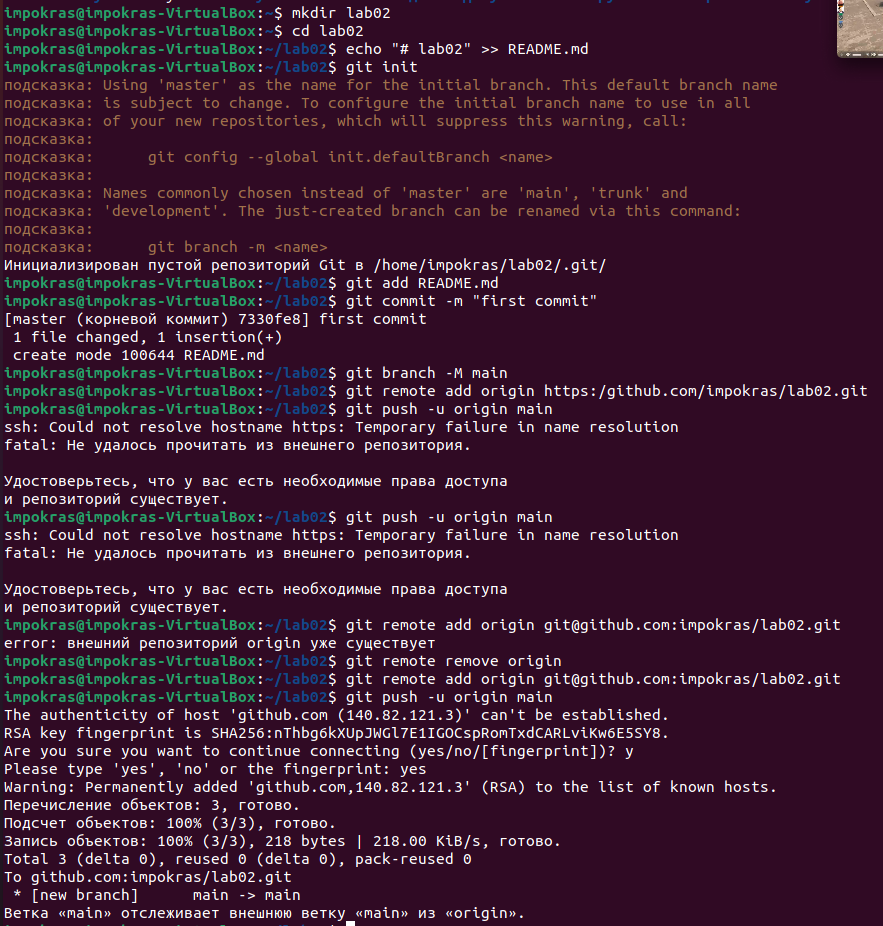
 

## Я создал пустой репозиторий lab02 для дальнейшей работы с ним.



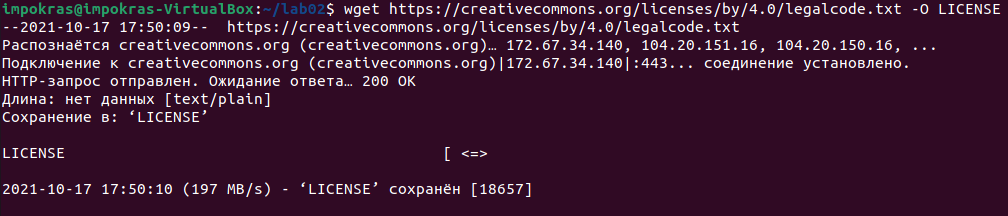
empty repository

## Для подключения репозитория к github я создал readme.md файл и сделал коммит.



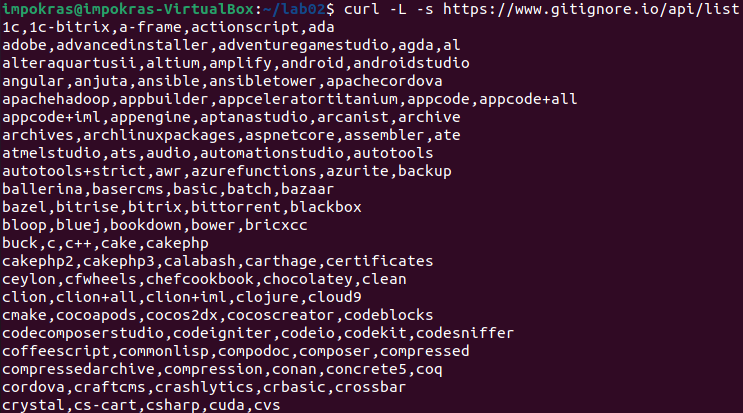
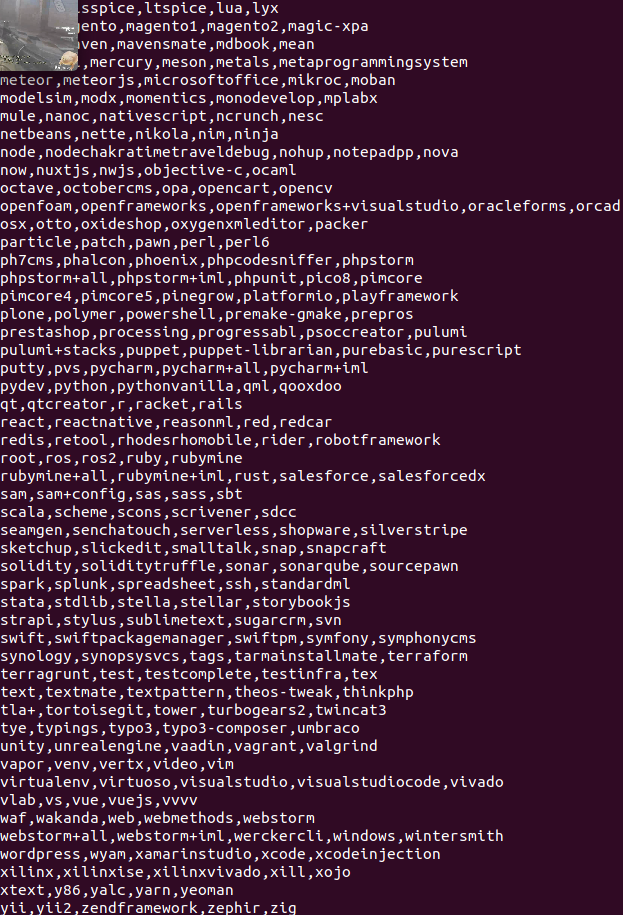
README

## Первичной конфигурацией я использовал создание файла лицензии.

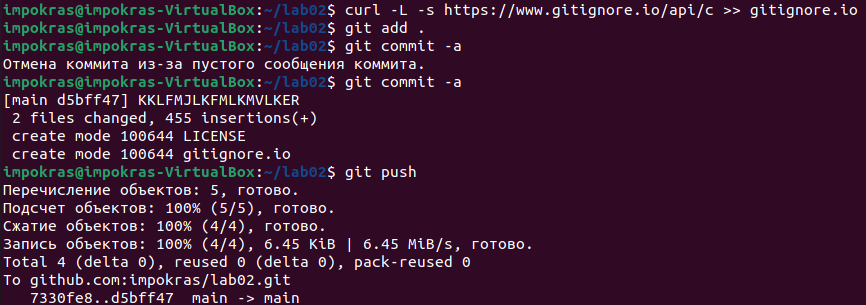


License

## Далее я добавил шаблон конфигурируемых файлов с помощью команды curl.

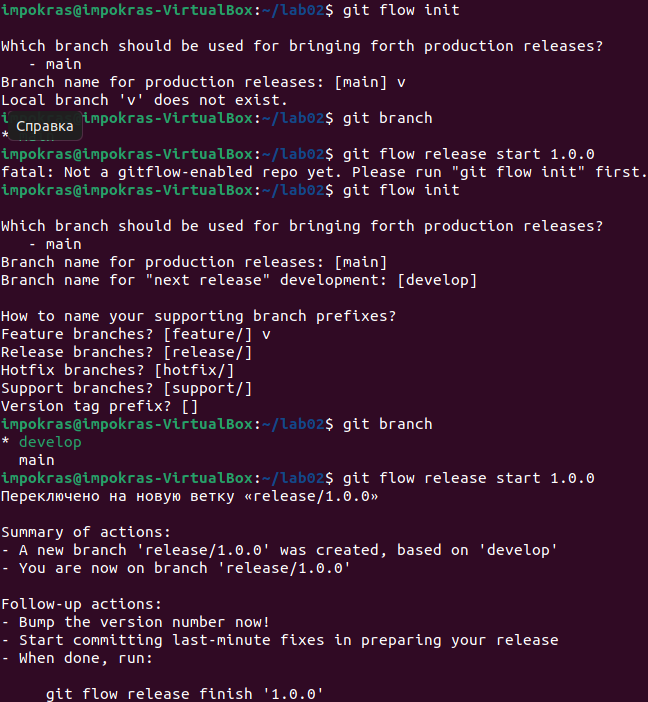
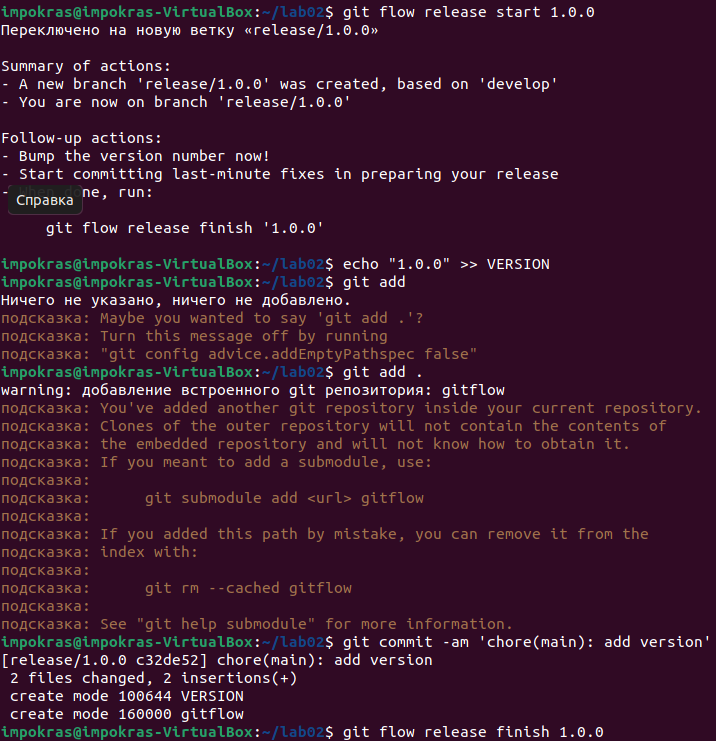
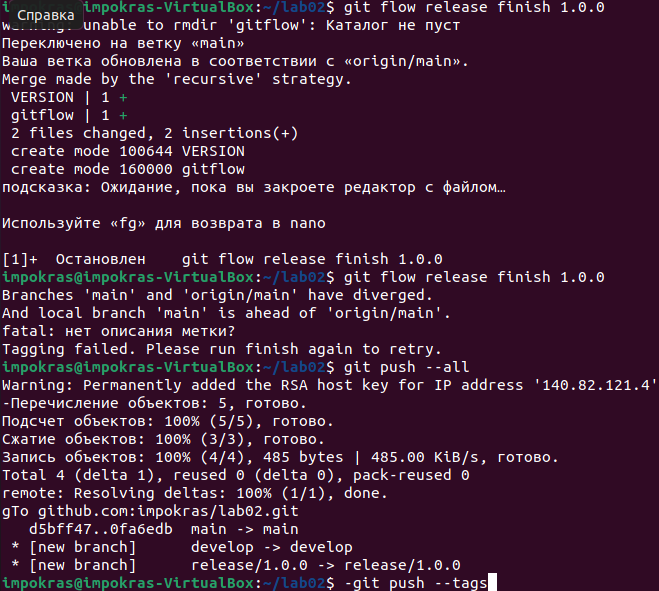
 

## После я добавил новые файлы, сделал коммит и отправил их на github.

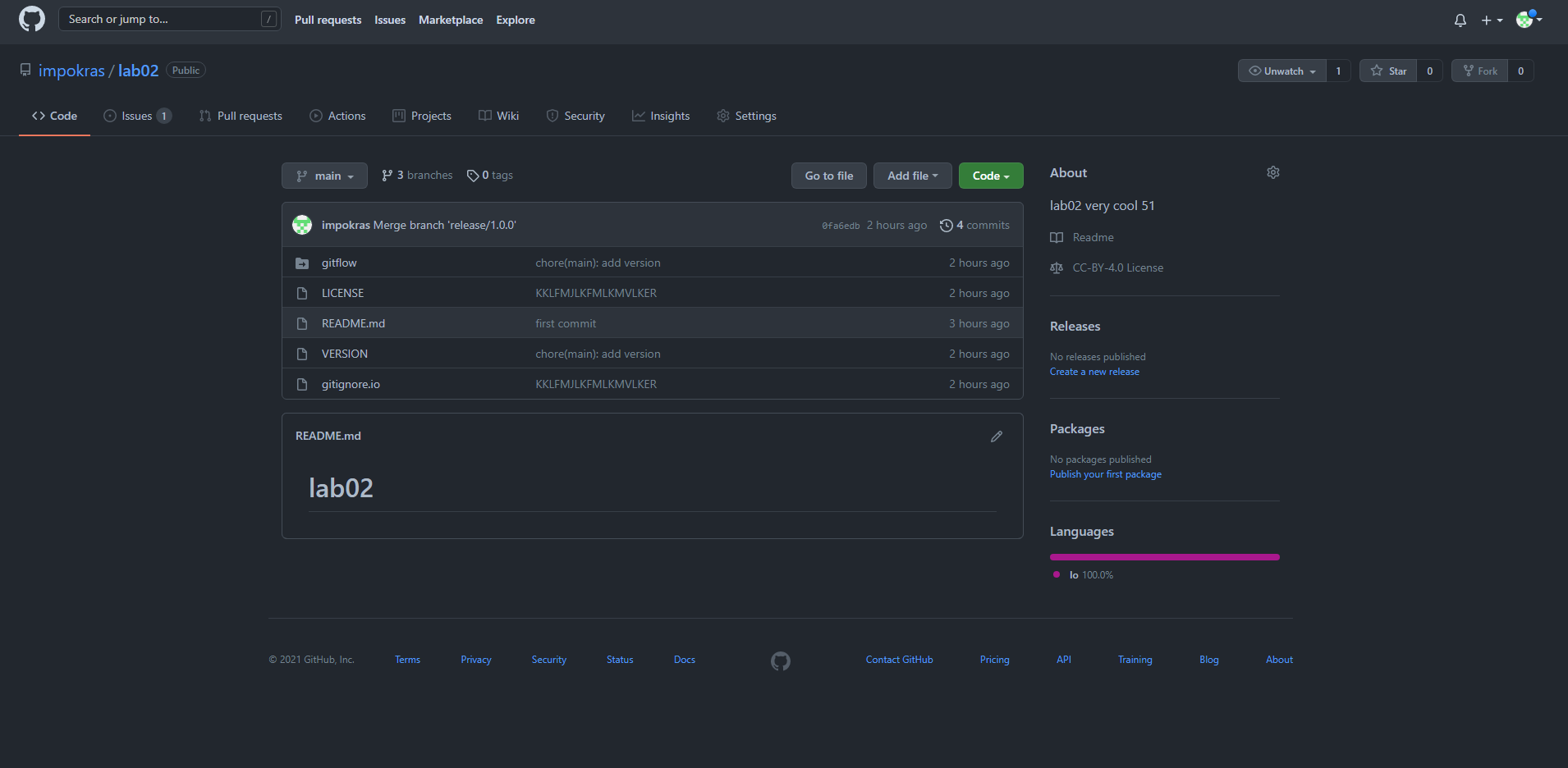


gitconfig - fin

## Я инициализировал git flow и создал новый релиз, указав версию и дбавив индекс.

  ## После чего я залил релиз на github. 

# Результат

 ## Вывод:

Я изучил идеологию и применение средств контроля версий.

# Ответы на контрольные вопросы:

1. система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется.
2. • некоторое место, где хранятся файлы • commit - отдельная версия программы • история - информация о всех изменениях в проекте • рабочая копия - текущее состояние файлов проекта
3. децентрализованные хранятся на многих устройствах и передаются многим устройствам(Mercurial), централизованные - файлы хранятся в одном устройства(CVS)
4. изменяем, делаем коммиты, закидываем на репозиторий, если необходимо делаеи релизы.
5. делаем пул нынешних изменений, изменяем, добавляем свои изменения в ветку, коммиты и т.д.
6. контроль версиями, одновременная работа нескольких людей над 1 проектом и многое другое
7. -init = инициализация локального репозитория -status = показ всех измененных и добавленных в коммит файлов -add = добавление файлов
8. проект с закрытым исходным кодом используют локальные репозитории, проекты с открытым исходным кодом, проекты выполняемые несколькими людьми, удаленно - удаленные репозитории
9. ветки используются для удобства работы с версиями и изминениями программы
10. многие конфигурационные файлы с личными или важными файлами не должны бытб доступны, а также есть примеры папок/файлов, которые создаются для кэширования, которые независимы от проекта и от которых проект не зависит.