**Ресурсы для первоначального ознакомления**

<https://proglib.io/p/git-for-half-an-hour/>

<https://www.gitignore.io/>

<https://git-scm.com/book/ru/v1/%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5-%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D1%8B-Git>

<https://githowto.com/ru>

**Источник инсталляций и сведения об учетной записи GitHub**

|  |  |
| --- | --- |
| Серверы:  Visual svn  <https://www.visualsvn.com/>  git  <https://github.com/>  <https://git-scm.com/download/win>  клиенты  <https://tortoisesvn.net/downloads.html>  <https://tortoisegit.org/>  <https://github.com/login>  логин  azureitstepkharkov  пароль  Azure$13 |  |

**Дополнительные материалы**

**Как начать работать с GitHub: быстрый старт**

<https://www.youtube.com/watch?v=mxDCimFA2dw>

<https://habrahabr.ru/post/125799/>

**Getting Started with TortoiseGit & Github**

<https://www.inboundnow.com/getting-started-tortoisegit-github/>

<https://github.com/moneymanagerex/moneymanagerex/wiki/Working-with-Git-and-TortoiseGit-in-Windows>

**Примеры**

Работа с Git и Tortoise

<https://github.com/moneymanagerex/moneymanagerex/wiki/Working-with-Git-and-TortoiseGit-in-Windows>

Видео, если не используем ssl соединение:

<https://www.youtube.com/watch?v=mxDCimFA2dw>

или, если используем ssl соединение:

<https://www.youtube.com/watch?v=fNPLuJTTto0>

(https://gist.github.com/svanas/87330eeb17313ea50d5cf9c265ab693f)

Работа с Git и Visual Studio

<https://www.infragistics.com/community/blogs/dhananjay_kumar/archive/2016/07/21/step-by-step-working-with-github-repository-and-visual-studio-2015.aspx>

или

<https://blogs.msdn.microsoft.com/benjaminperkins/2016/04/01/setting-up-and-using-github-in-visual-studio-2015/>

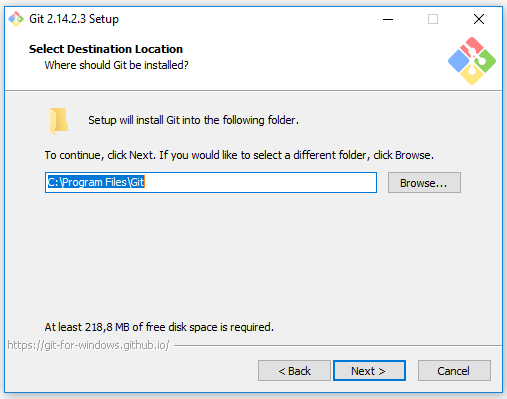
или

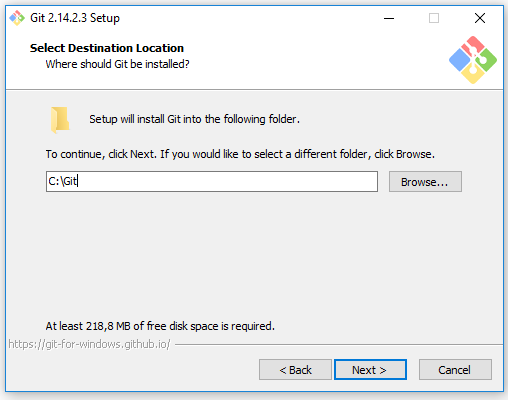
<https://visualstudio.github.com/>

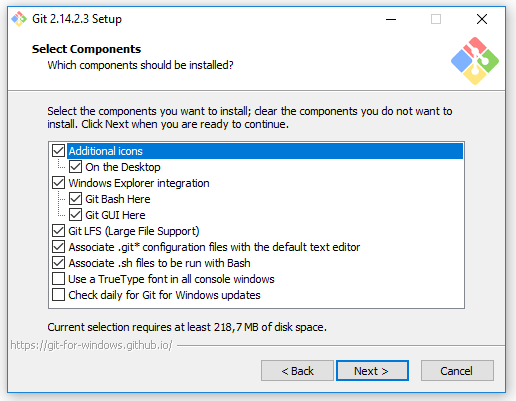
видео:

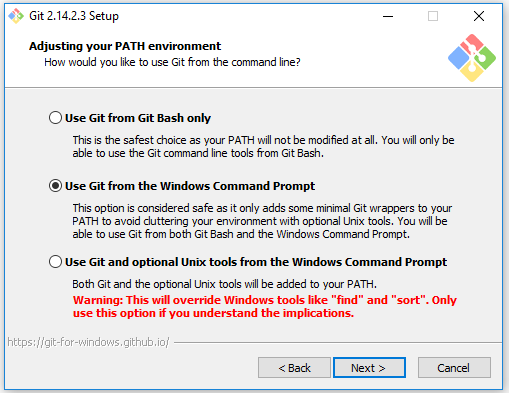
<https://www.youtube.com/watch?v=zNANRs0nZfU>

**Особенности установки**









**Проверка установки - запуск управляющей консоли Git (Git Bush – родной механизм)**

|  |  |
| --- | --- |
|  |  |

Любая команда к системе GIT начинается с трех симолов git

**Список команд см. файл git\_cmd.txt**

|  |  |
| --- | --- |
| **Команда Git** | **действие** |
| git --help | Вывод списка команд Git |
| git config --list | Текущие настройки системы Git на этом компьютере |
| git config --global --list | Вывод только глобальных настроек. Это настройки которые видны всем. |
| git status |  |
| git init | После выполнения этой команды в каталоге, это каталог помещается под наблюдение констроля версий Git: говорят гит репозиторий проинициализирован в каталоге |
| git add . | Индексирование изменений. Такой вид команды означает, что нужно проиндексировать все файлы из каталога. Только те файлы, что находятся в индексе отправляются (команда commit) на локальный сервер. |
| git add readme.txt newdata.txt | Пример команды, когда я перечисляю имена тех файлов, которые нужно добавить в индекс. |
| git reset readme.txt | Пример команды, которая исключает файл из индекса |
| git commit –m’my comment text’ | Пример команды commit. Ключ -m позволяет задать комментарий к комиту |
| git commit –a –m’comment’ | Самый распространенный вид команды: поместили все изменения в индекс и сразу же закоммитили |
| git log | Пример команды. Которая выводит список изменений |
| git log –p | Лог с расшифровкой изменений |
| git log -2 | Вывод только последних |

Переключение между командами в консоле – стрелки клавиатуры вверх-вниз

Первый тест – проверяем на какого пользователя настроен Git (Не позволяет работать анонимно. Если не заданы глобальные настройки, будут некорректно работать коммиты.)

Выполняем команды

git --help

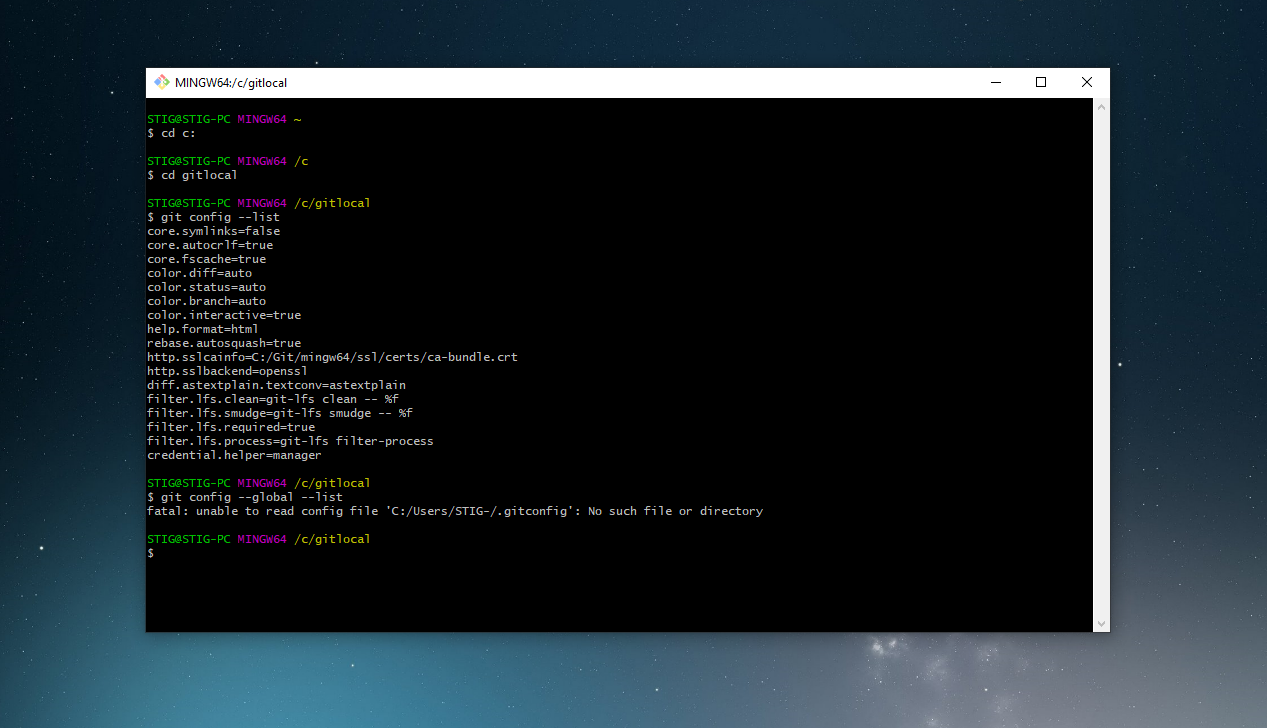
git config --list

git config --global --list

git config --global user.name 'New Name'

Для выяснения начального статуса Git.

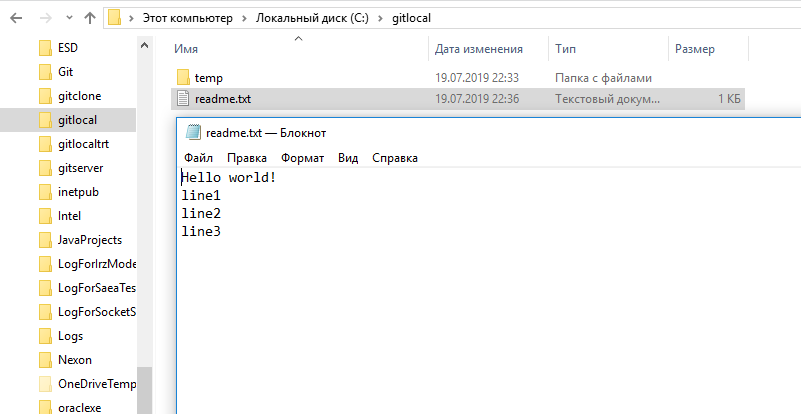
При первом запуске можем увидеть скрин



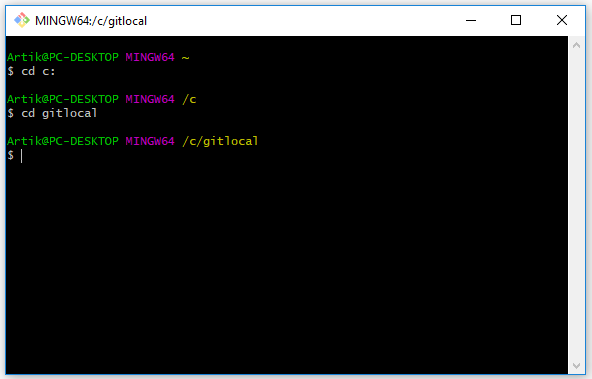
Это значит, что не установлены обязательные параметры (имя пользователя и почта)

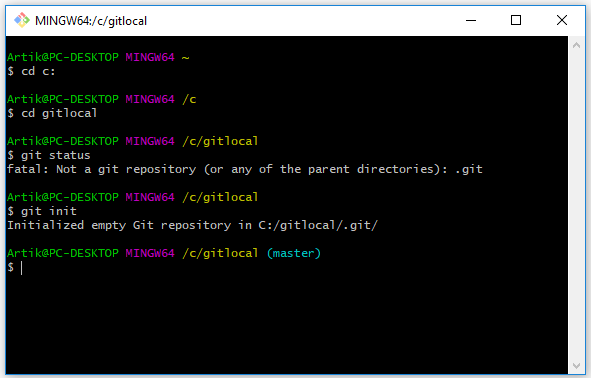
**Локальная работа с Git (сценарий работы, анологичен SVN)**

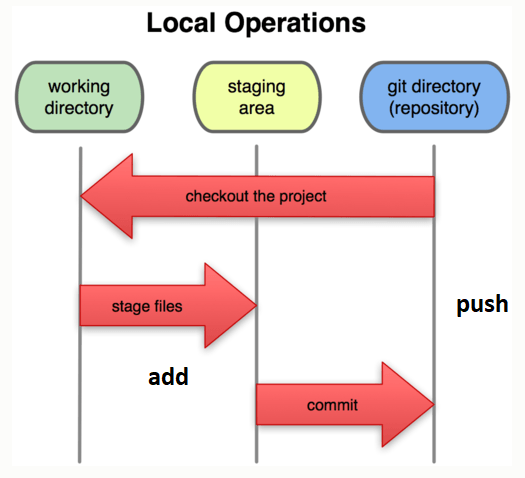
Все команды, которые мы будем использовать – это те команды, которые не требуют подключения к интернету.



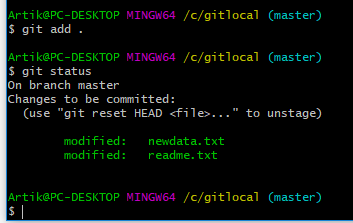
Переход в наш каталог командами консоли



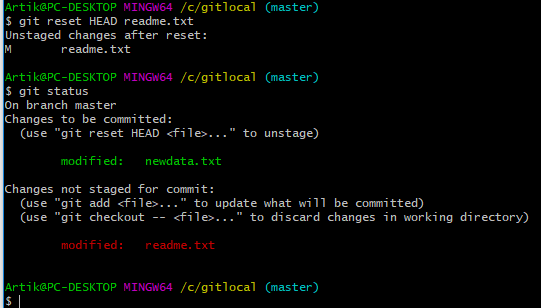




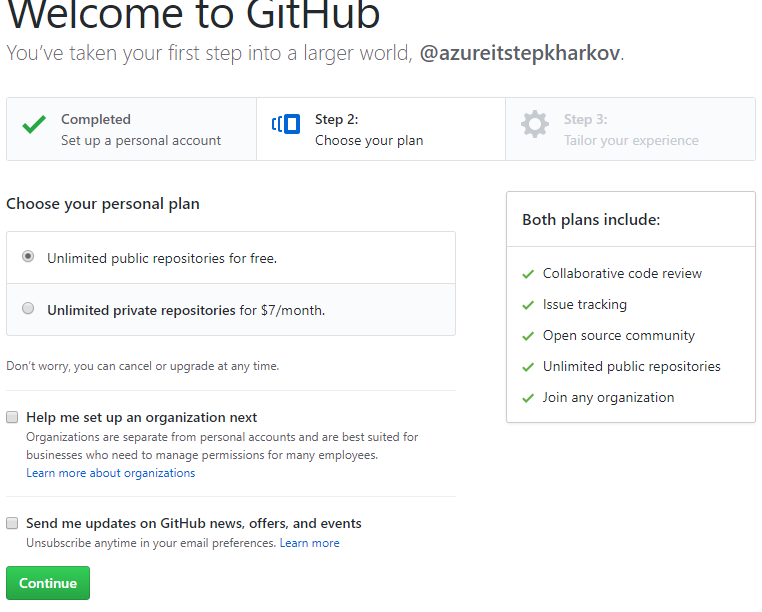
Вывод данных из области подготовленного к комиту



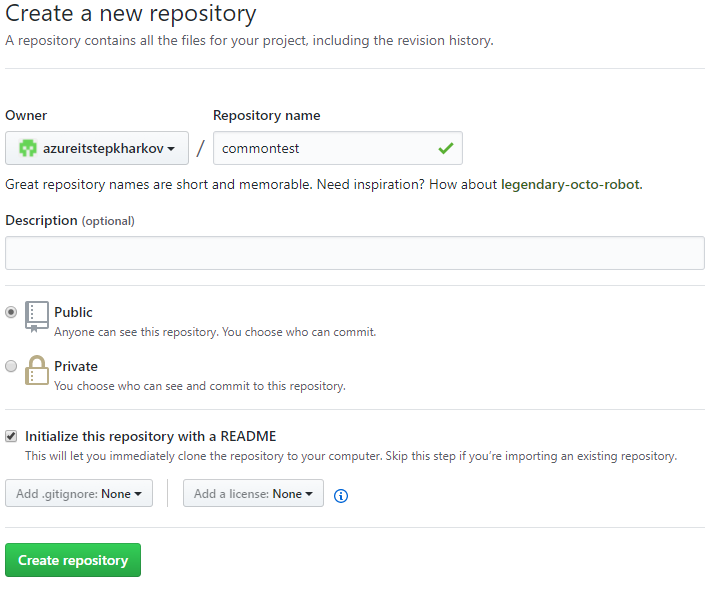
git reset HEAD readme.txt



**Состояние учетной записи Git после регистрации**



**Создание и использование репозитория**

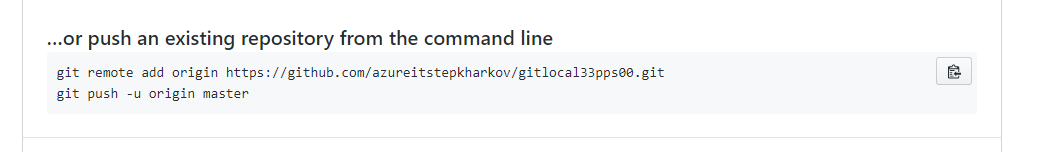


Далее у нас два варианта развития событий:

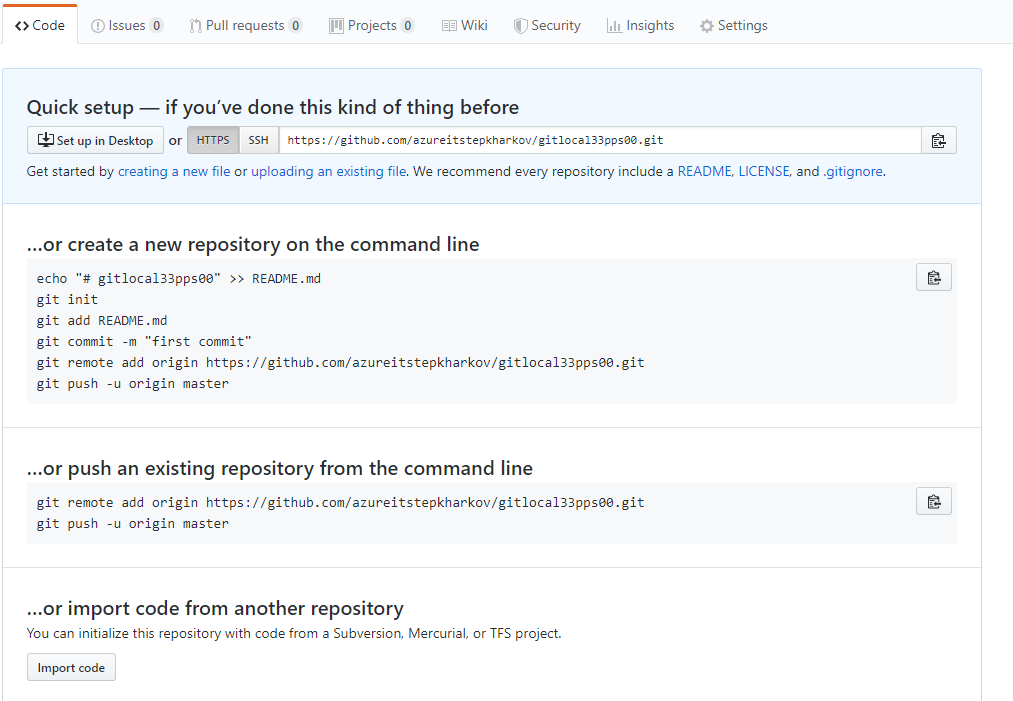
Вариант 1 – воспользоваться созданной здесь ссылкой и загрузить в GitHub наш локальный репозиторий.

Вариант 2 – выполнить полную настройку репозитория на сайте, а потом склонировать его в локальный репозиторий.

Рассмотрим Вариант 1.



Из общего меню:



Необходимо передать команду

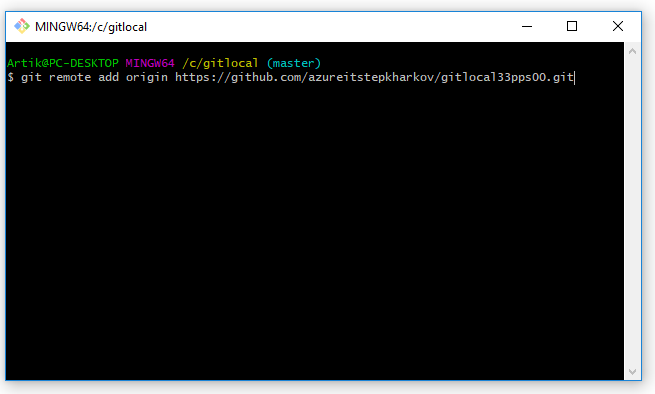
git remote add origin https://github.com/azureitstepkharkov/gitlocal33pps00.git

или

git remote add origin https://github.com/azureitstepkharkov/31pr12server.git

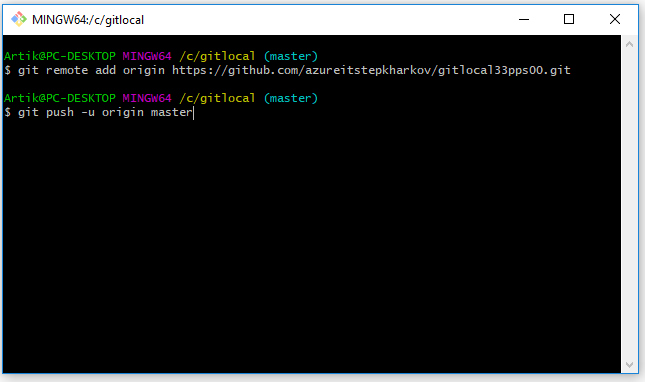
в мой локальный репозиторий.

|  |  |
| --- | --- |
| Шаг 1. Вызвать контекстное меню на моем каталоге  Шаг 2 Выполнить команду Bush Here  Шаг 3 в окне Bush консоли вызвать контекстное меню и выбрать пункт вставить (shift insert) |  |



“remote add” говорит, что мы добавляем данные для удаленного репозиторий

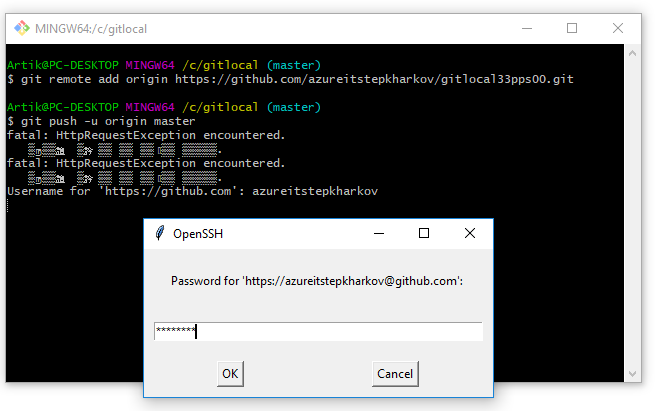
“origin” это псевдоним для полного пути (имени) репозитория для которого мы подготовили данные

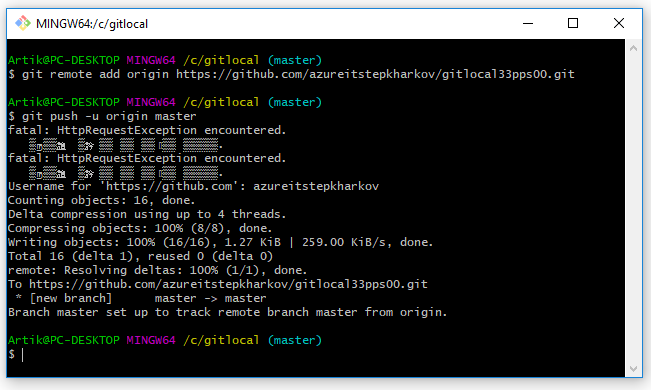


Команда «push» и выполняет загрузку подготовленных данных в ветку master (но уже на сервере)

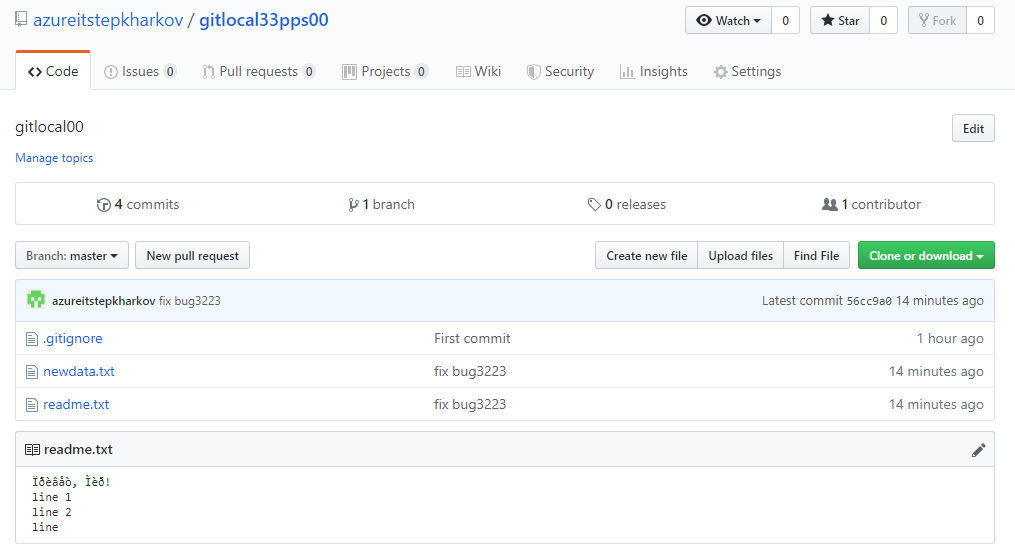
Ключ «-u» освобождает нас от дальнейшей необходимости после команды push указывать псевдоним origin и имя основной ветки master, то есть теперь каждый push это

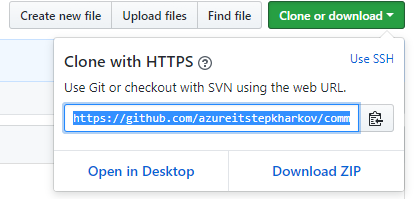
git push origin master



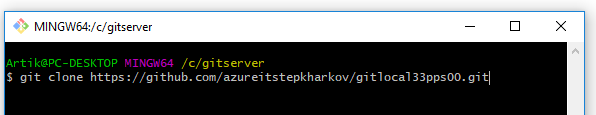


После обновления страницы нашего проекта в GitHub мы увидим историю наших коммитов и получим возможность клонировать проект.

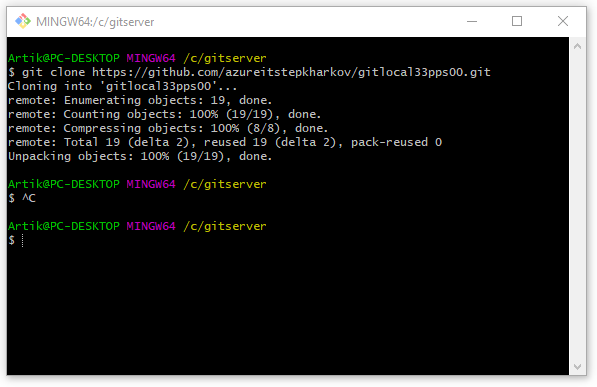




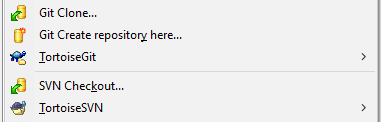
Клонирование проекта можно выполнить используя как консоль управления,



Команда git clone



так и клиент TortoiseGit:



**Основы работы с Git в режиме нескольких веток: dev и master**

|  |  |
| --- | --- |
| git branch  git branch dev1  git branch  git checkout dev1 |  |