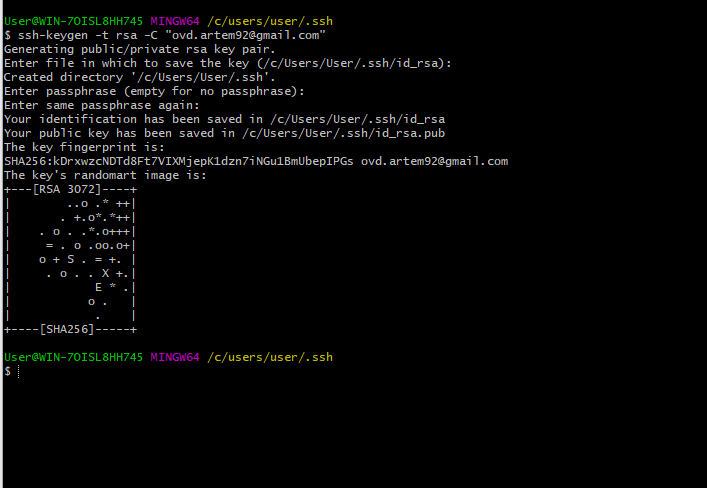
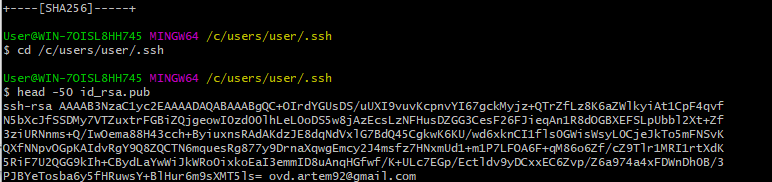
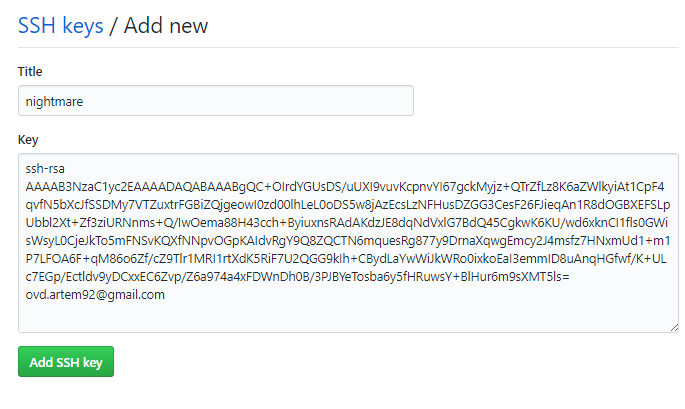
1.Установите git и сгенерируйте пару ssh ключей



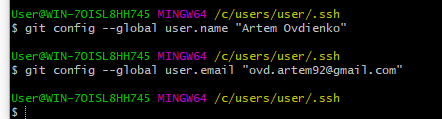
Переходим в директорию где храниться ssh ключ и читаем публичный ключ



Копируем его в буфер и переходи на github и вставляем свой ключ



2. Укажите свой user.name и user.email в git.



3. Создайте новый репозиторий на github.com и склонируйте его локально на свой компьютер.

Создаем на github репозиторий nightmare и копируем ssh ключ.

[git@github.com:ovdartem92/nightmare.git](mailto:git@github.com:ovdartem92/nightmare.git)

а. Создаем папку nightmare на диске c, в корневом каталоге



Б. Переходим в эту папку



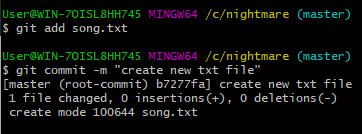
В. Создаем репозиторий в этой директории



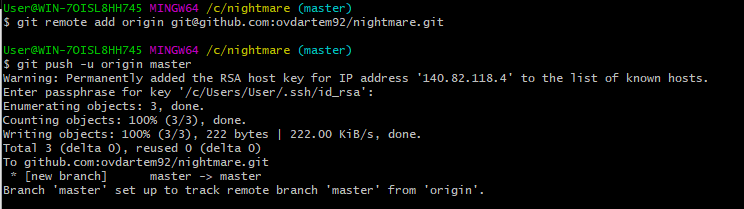
Г. Создаем файл song.txt



Д. Добавляем файл в индекс и создаем коммит

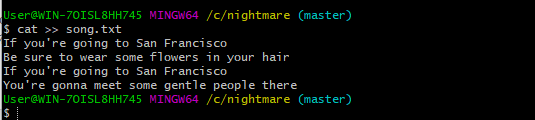


Е. Добавляем предварительно созданный пустой репозиторий и отправляем данные из локального репозитория в удаленный (ветку master)



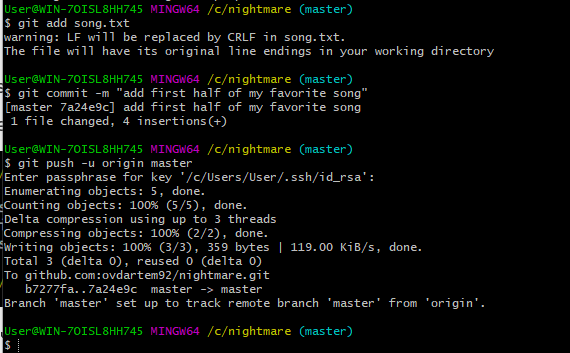
4. Создайте файл названием song.txt и поместите туда половину текста любимой песни.

Файл song.txt мы создали ранее поэтому нам необходимо только добавить в него текст.

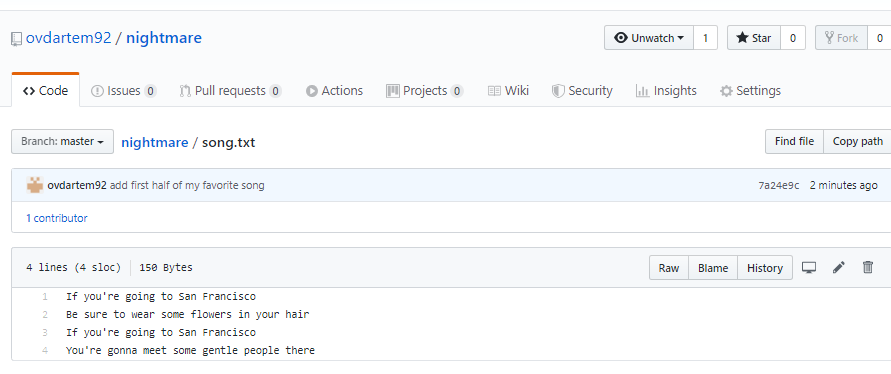


В конце используем связку ctrl-d чтоб закончить редактирование файла.

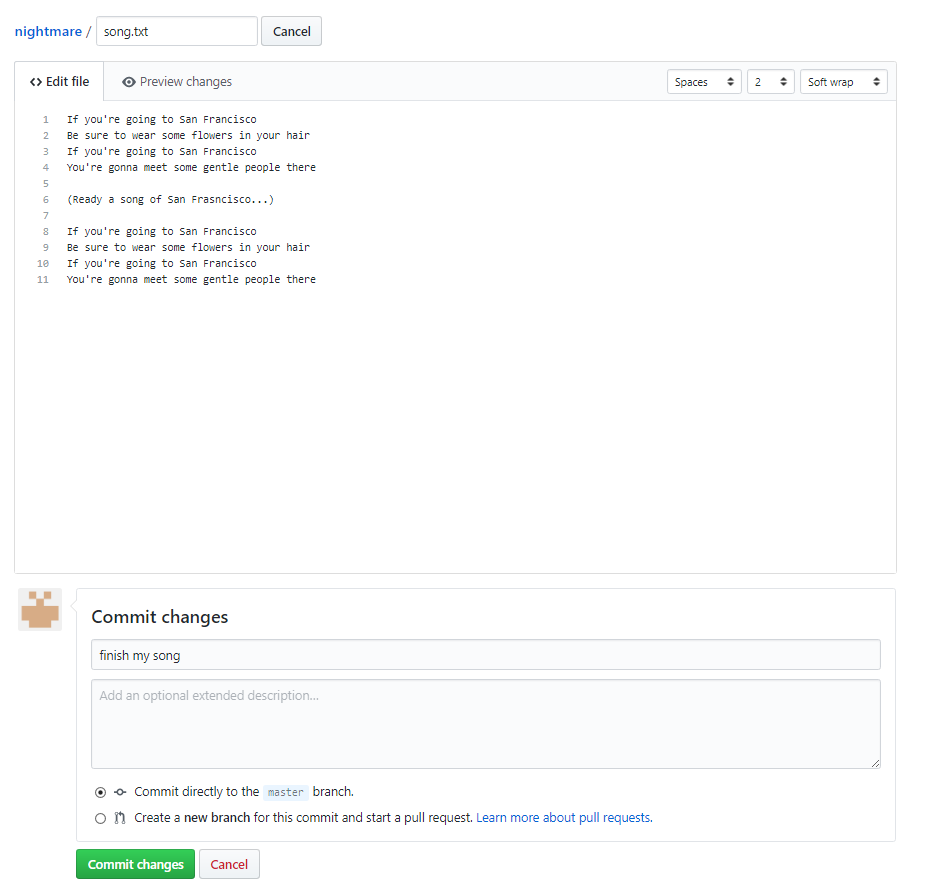
5. Сделайте коммит с названием "add first half of my favorite song" и отправьте его на сервер.



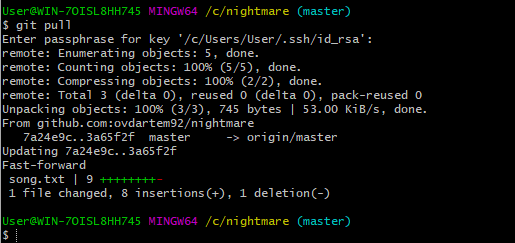
6. Убедитесь что на github есть файл song.txt с текстом песни.



7. Используя веб-интерфейс гитхаба добавьте вторую половину текста песни и сделайте коммит с названием "finish my song".



8. В локальном репозитории сделайте pull и убедитесь что коммит, который вы создали на github, подтянулся и у вас полный текст песни.





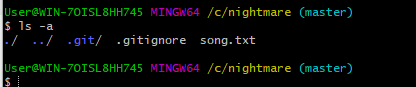
Данное задание выполняется сразу после предыдущего (I Can Win).

1. Добавьте в проект файл .gitignore и настройте так чтобы скрыть файлы с расширением **.db**, **.log** и директории с названиями **target** или **bin**.

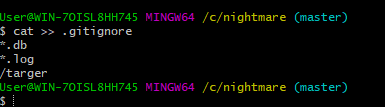
Создаем .gitignore



Просматриваем скрытые файлы командой ls -a



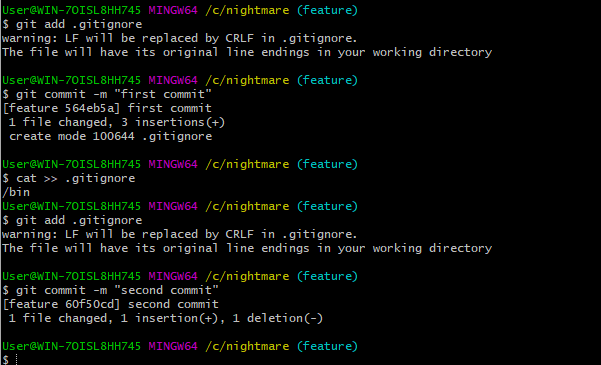
Добавляем данные в файл .gitignore



2. Создайте ветку feature



и добавьте в неё два коммита



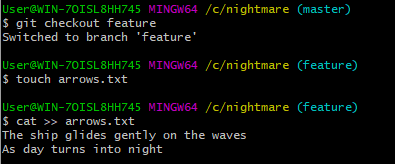
3. Смержите ветку feature в master

Переходим в ветку master и делаем merge

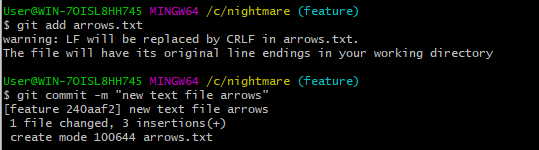


4. Вернитесь в feature и создайте файл arrows.txt cледующего содержания:

The ship glides gently on the waves  
As day turns into night

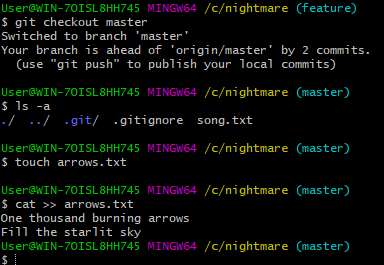


Выполните коммит.

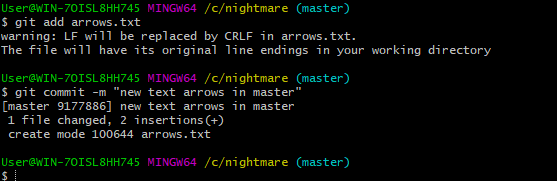


5. Перейдите в master. Создайте там файл arrows.txt и добавьте следующий текст:

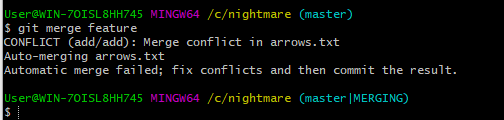
One thousand burning arrows  
Fill the starlit sky



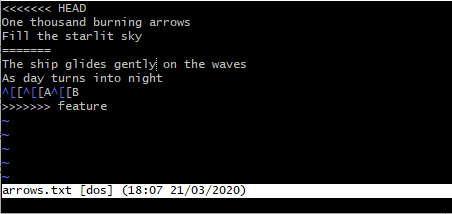
Выполните коммит.



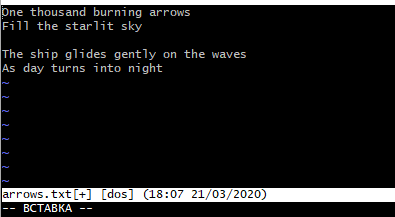
6. Смержите feature в master решив конфликт: сохраните все 4 строки в файле arrows.txt в порядке их добавления в пунктах 4 и 5.



Вызываем редактор vi командой vi arrows.txt



Редактируем текст. Для начала нужно нажать Insert и отредактировать текст.



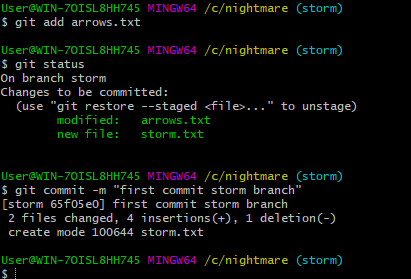
После нажать клавишу esc и ввести команду :wq

Данное задание выполняется после предыдущего (Bring It On).

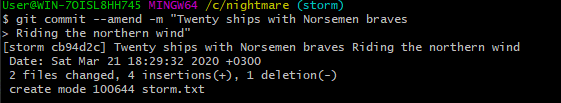
1. Создайте ветку storm и добавьте коммит в файл storm.txt:



В прошлом задании, в конце задания, не добавил в индекс arrows.txt, сейчас пришлось добавить два файла: arrows.txt и storm.txt в ветку storm, чтоб сделать коммит.

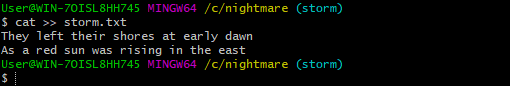


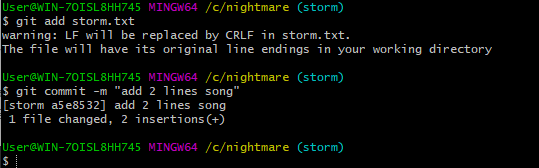
Специально ввел не другой текст коммита, чтоб показать как изменить текст, если допустил ошибку.



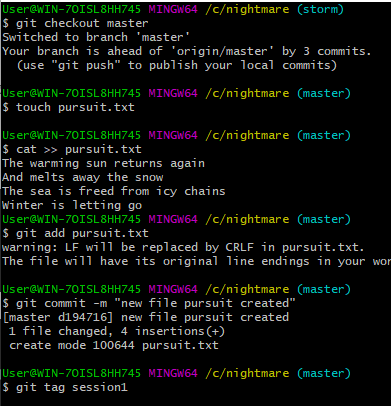
2. Добавьте еще 2 строки в storm.txt и сделайте еще один коммит:

They left their shores at early dawn  
As a red sun was rising in the east

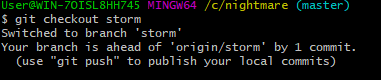




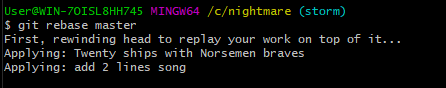
3. Вернитесь в master и создайте файл pursuit.txt с текстом ниже:

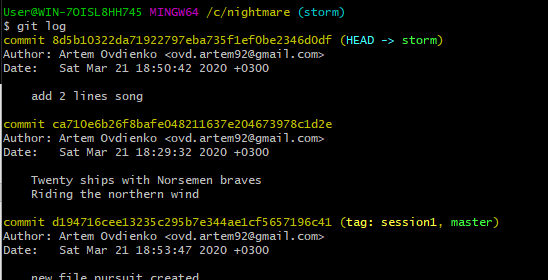


4. Отметьте коммит тегом session1 и перейдите в ветку storm

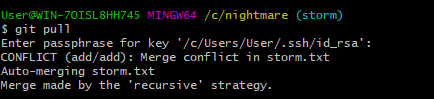


5. Сделайте rebase ветки storm так чтобы она содержала последний коммит из мастера.



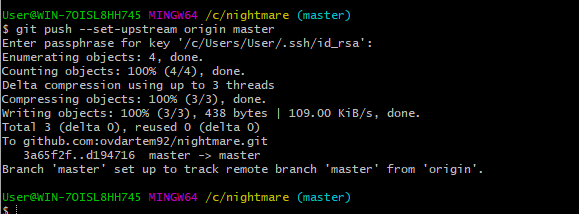


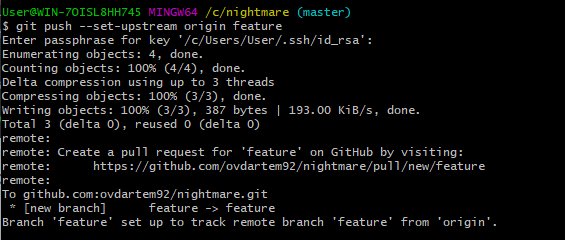
Сделать git pull и решить конфликт в редакторе vi

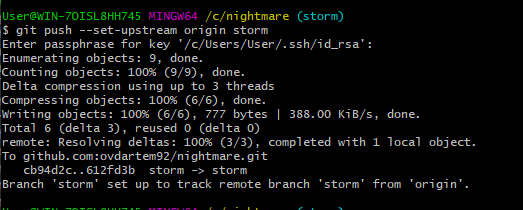


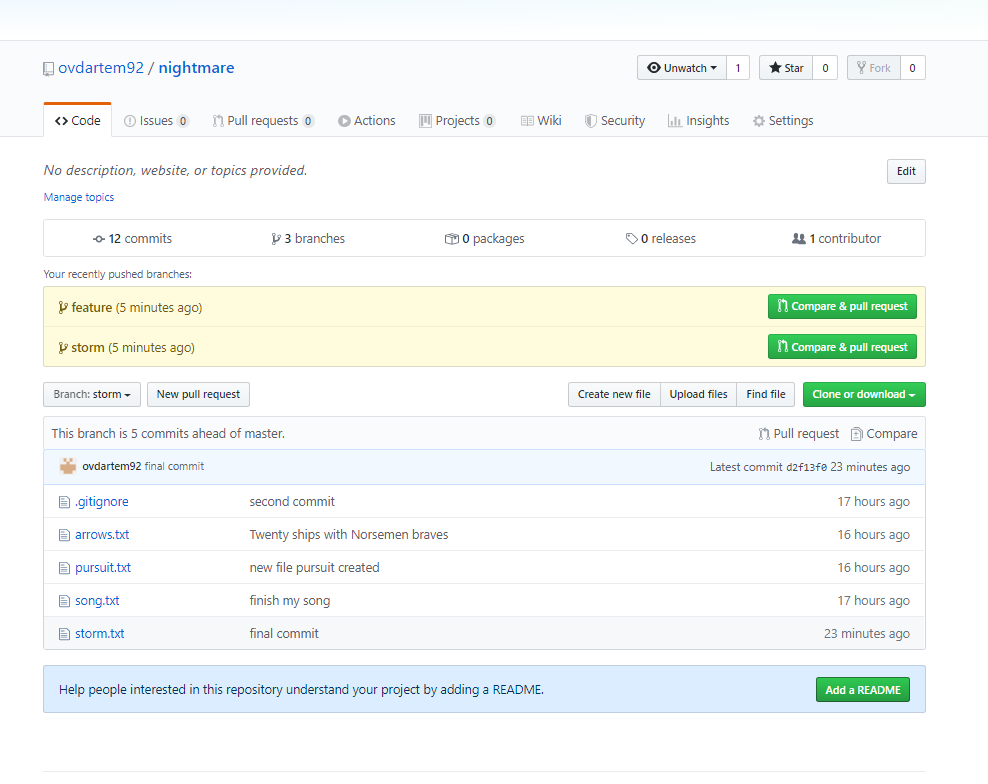
Данное задание выполняется сразу после предыдущего (Hurt Me Plenty).

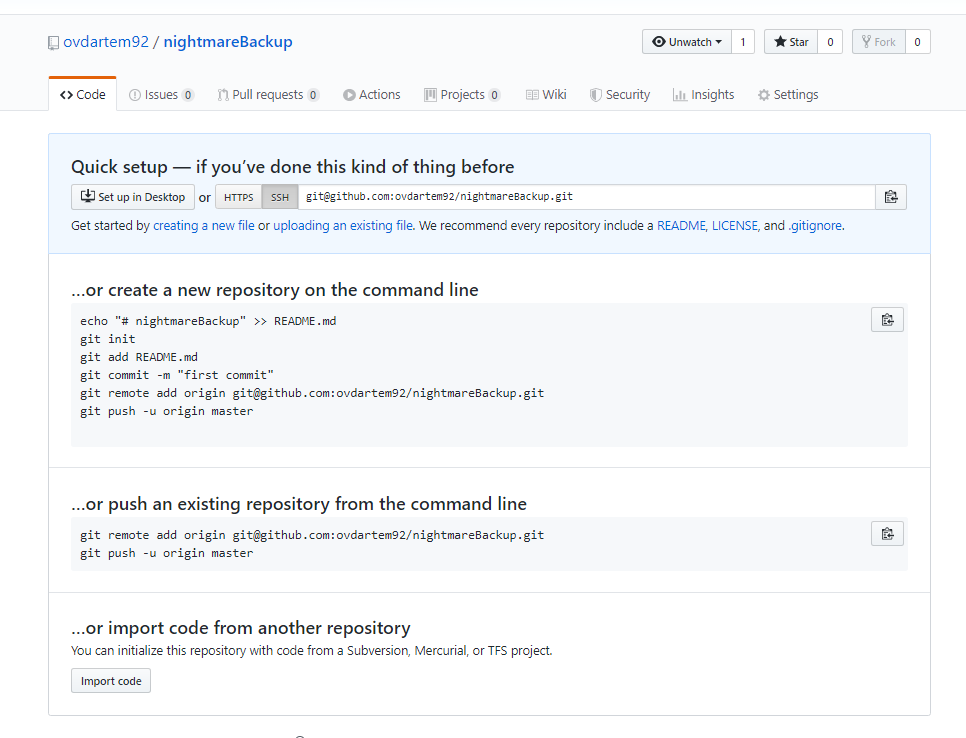
1. Сделайте push вашего репозитория и убедитесь, что все коммиты есть на github.



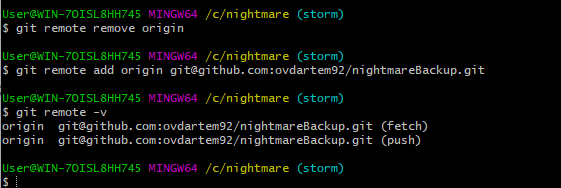




  
2. Сделайте новый репозиторий на github.



3. Смените remote в локальном репозитории так, чтобы fetch и push шел на новый репозиторий который был создан в предыдущем шаге.

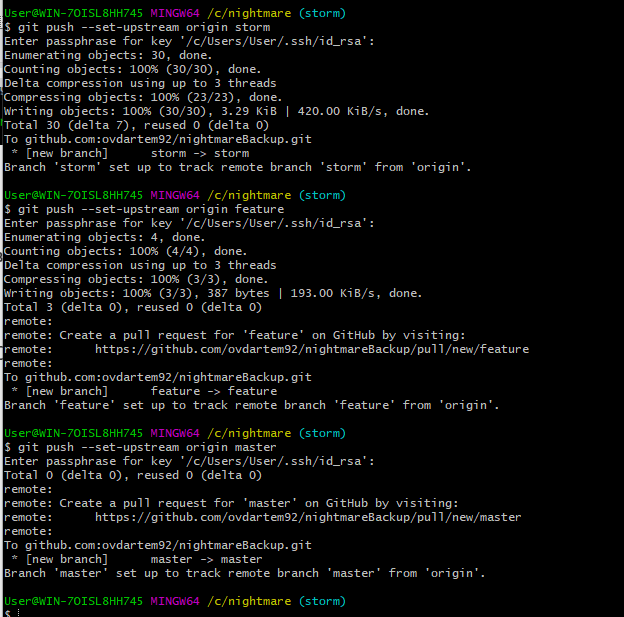


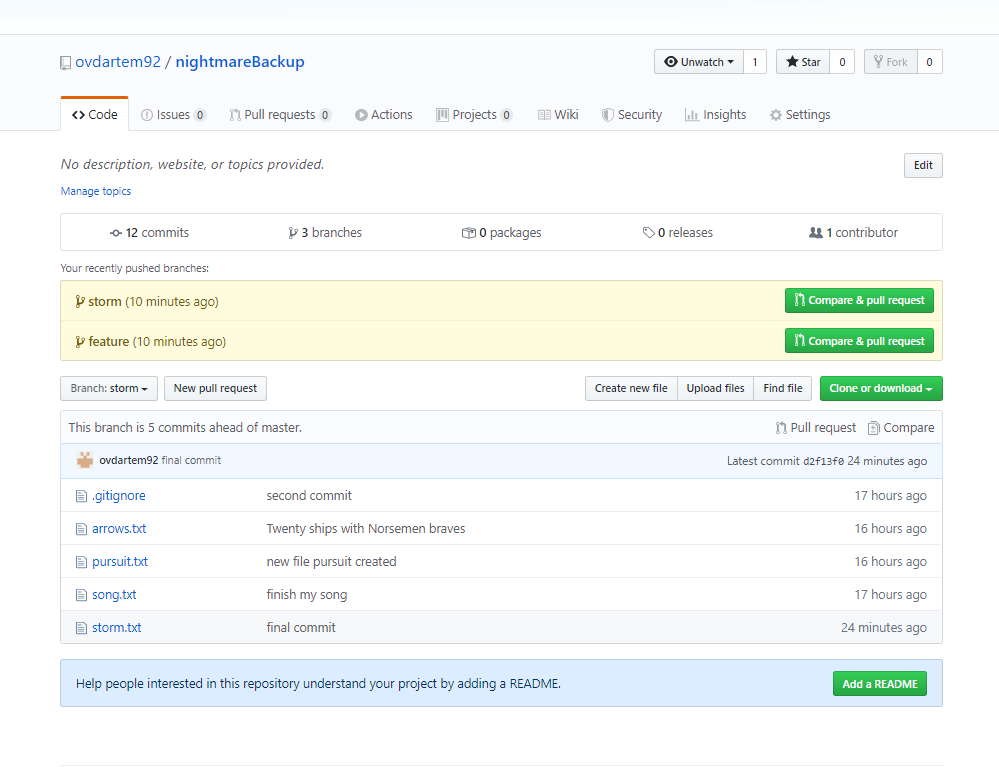
Git remote remove origin - удаляет старую ссылку на репозиторий github

Git remote add origin [git@github.com:ovdartem92/nightmareBackup.git](mailto:git@github.com:ovdartem92/nightmareBackup.git) - добавляем новый репозиторий

Git remote –v - проверяем какой репозиторий мы используем

4. Сделайте push и убедитесь, что второй репозиторий на гитхабе выглядит так же, как и первый.





5. Верните настройки remote в исходное состояние: пул и пуш первого локального репозитория ведет в один удаленный репозиторий на гитхабе.

