

Урок 8. Выполняем развертывание проекта

# ОГЛАВЛЕНИЕ

[ОГЛАВЛЕНИЕ 2](#_Toc100517438)

[ВВЕДЕНИЕ И РЕГИСТРАЦИЯ НА HEROKU 3](#_Toc100517439)

[ГОТОВИМ ПРОЕКТ К РАЗВЕРТЫВАНИЮ 4](#_Toc100517440)

[создаем и настраиваем файлы procfile и requirements.txt 4](#_Toc100517441)

[создаем удаленный репозиторий на GitHub 5](#_Toc100517442)

[создаем локальный репозиторий 5](#_Toc100517443)

[связываем локальный репозиторий с удаленным 6](#_Toc100517444)

[отправляем проект в удаленный репозиторий 8](#_Toc100517445)

[как получить токен в GitHub 9](#_Toc100517446)

[устанавливаем консольный клиент для Heroku 11](#_Toc100517447)

[ВЫПОЛНЯЕМ РАЗВЕРТЫВАНИЕ 12](#_Toc100517448)

[ЗАКЛЮЧЕНИЕ 15](#_Toc100517449)

[КЛЮЧЕВЫЕ ВЫВОДЫ 15](#_Toc100517450)



# ВВЕДЕНИЕ И РЕГИСТРАЦИЯ НА HEROKU

Мы подготовили демоверсию нашего бота, проверили работу, нашли и устранили возникшие ошибки. Теперь перед нами стоит не менее увлекательная задача – сделать проект доступным другим пользователям, т.е. выполнить «деплой» (развертывание) проекта. Эту задачу можно выполнить различными средствами. Мы воспользуемся возможностями сервиса Heroku, который предоставляет разработчикам услуги хостинга. У сервиса доступны как коммерческие, так и бесплатные тарифные планы.

Поэтому перейдем на ресурс Heroku по ссылке и выполним несложную процедуру регистрации: [**https://www.heroku.com**](https://www.heroku.com).

Важно, что нужно выбрать именно бесплатный аккаунт, т.е. опцию Free Account. Он имеет ограничения по количеству размещаемых приложений, но нам для учебных целей этого будет вполне достаточно.

Панель приложений, деплой которых выполнен, доступна разработчикам по ссылке: [**https://dashboard.heroku.com/apps**](https://dashboard.heroku.com/apps).

Она может выглядеть следующим образом:

# ГОТОВИМ ПРОЕКТ К РАЗВЕРТЫВАНИЮ

## создаем и настраиваем файлы procfile и requirements.txt

Оба файла нужно создать в корневой директории проекта. Первый файл будет содержать объявления процессов, функционирующих на сервере после деплоя.

В нашем случае в этом файле будет всего одно объявление:

worker: python telbot.py

Это наш главный запускаемый модуль.

Второй файл requirements.txt также играет важную роль. В нем мы указываем зависимости, т.е. дополнительные библиотеки, без которых корректная работа нашего бота невозможна. Этот файл можно создать командой на основе вашего текущего виртуального окружения, под которым вы запускаете бот в его локальной версии.

**pip freeze > requirements.txt**

Можно также опираться на заранее заготовленные файлы с зависимостями.

В нашем случае в этот файл достаточно добавить три строки:

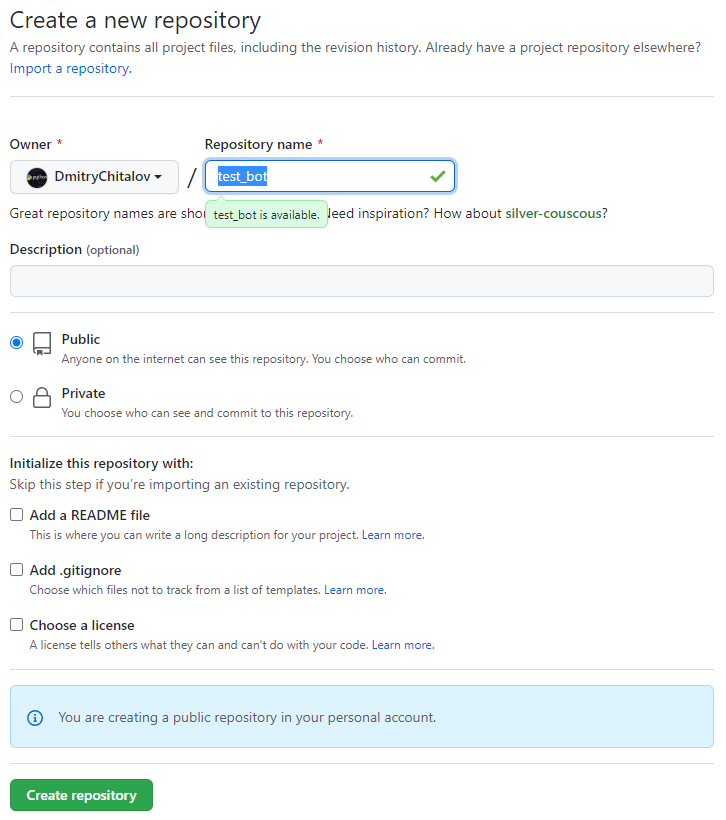
emoji==0.5.3

pyTelegramBotAPI==3.6.6

SQLAlchemy==1.3.6

## создаем удаленный репозиторий на GitHub

При деплое на Heroku код нашего проекта будет «подтягиваться» с удаленного репозитория на сервисе GitHub.

Создадим удаленный репозиторий.

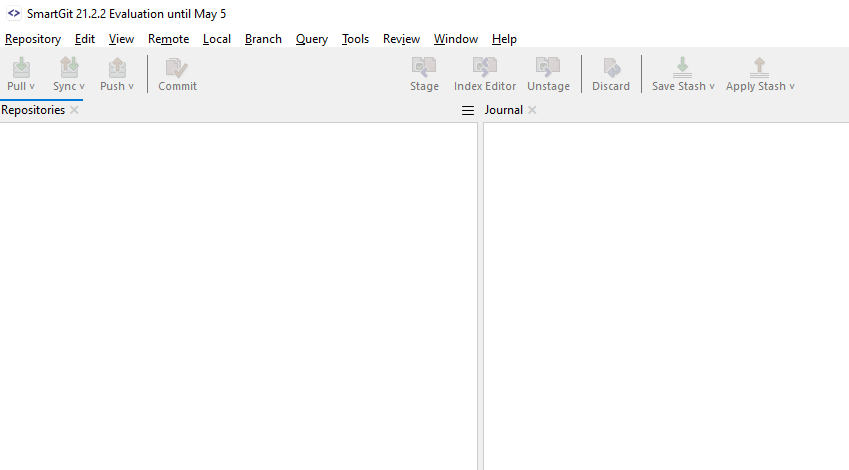
## создаем локальный репозиторий

Нам нужно код проекта отправить в удаленный репозиторий на GitHub. Для этого нужно инициализировать корневую директорию проекта в качестве локального репозитория. Это можно сделать разными способами, в том числе через консоль, но мы воспользуемся приложением SmartGit, которое предоставляет графический интерфейс, что конечно удобнее, чем вводить команды через терминал.

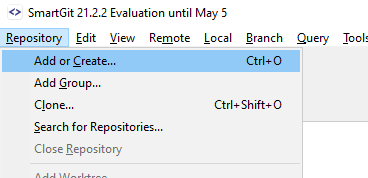
Приложение SmartGit включает бесплатную версию, ее установка очень простая.

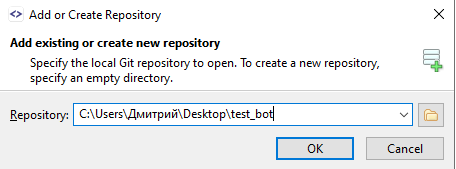
Загрузить приложение можно по ссылке: <https://www.syntevo.com/smartgit/download/>.

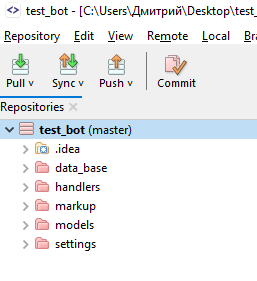
Окно приложения выглядит следующим образом:



Инициализируем локальную версию проекта в качестве локального репозитория:

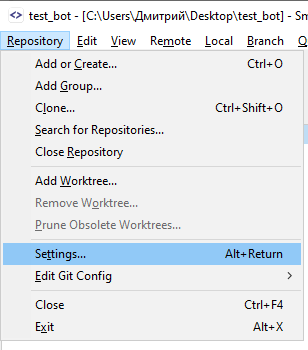
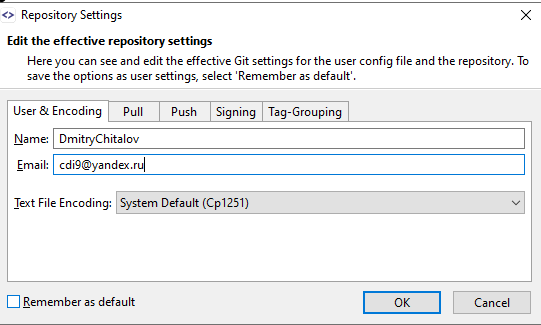




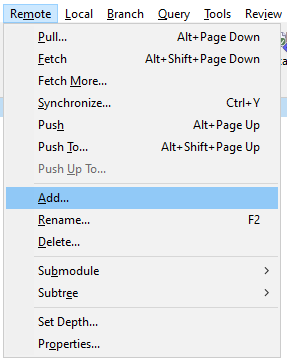


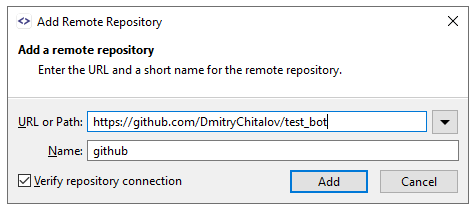
## связываем локальный репозиторий с удаленным

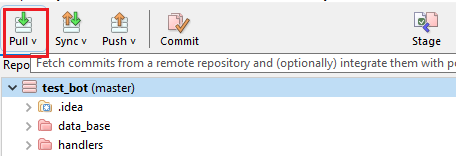
Укажем настройки пользователя в GitHub

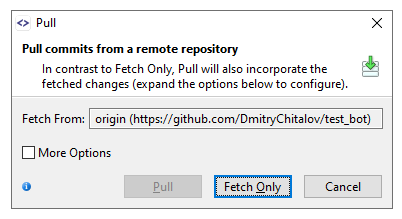


Подключим удаленный репозиторий.



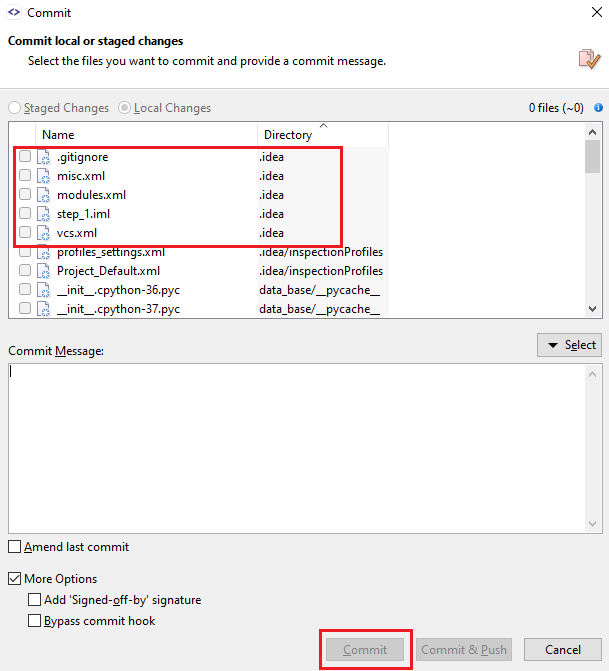


Связываем настройки удаленного репозитория с настройками локального.

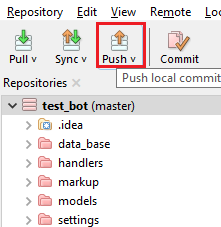


## отправляем проект в удаленный репозиторий

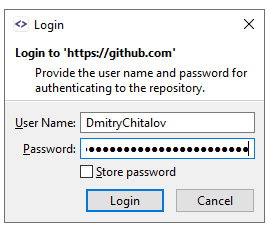
Выполним коммит (зафиксируем изменения) и отправим проект в удаленный репозиторий (команда «push»).

Перед выполнением коммита нужно выбрать компоненты (файлы и модули), которые должны быть переданы в удаленный репозиторий. При этом нужно выбирать не все компоненты, а только py-файлы, конфиги, txt-файлы, файлы баз данных. Но файлы с расширениями xml, pyc и т.д. добавлять не нужно.

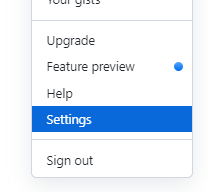
Далее выполняем push.



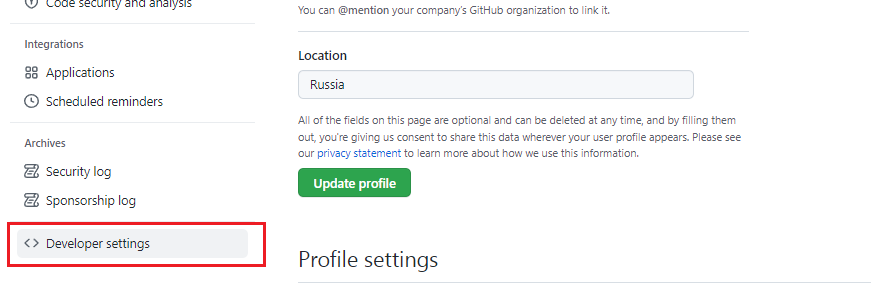
Укажем имя пользователя и токен.

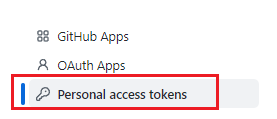


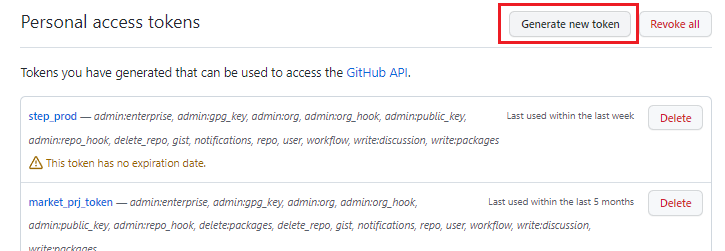
## как получить токен в GitHub

Нужно перейти на GitHub в раздел Settings.

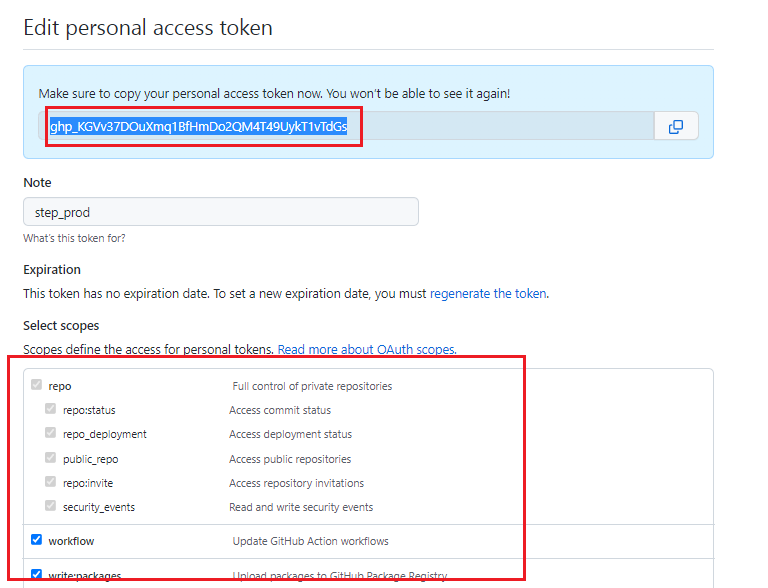
Перейдем в раздел настроек разработчика.

Перейдем в раздел токенов.



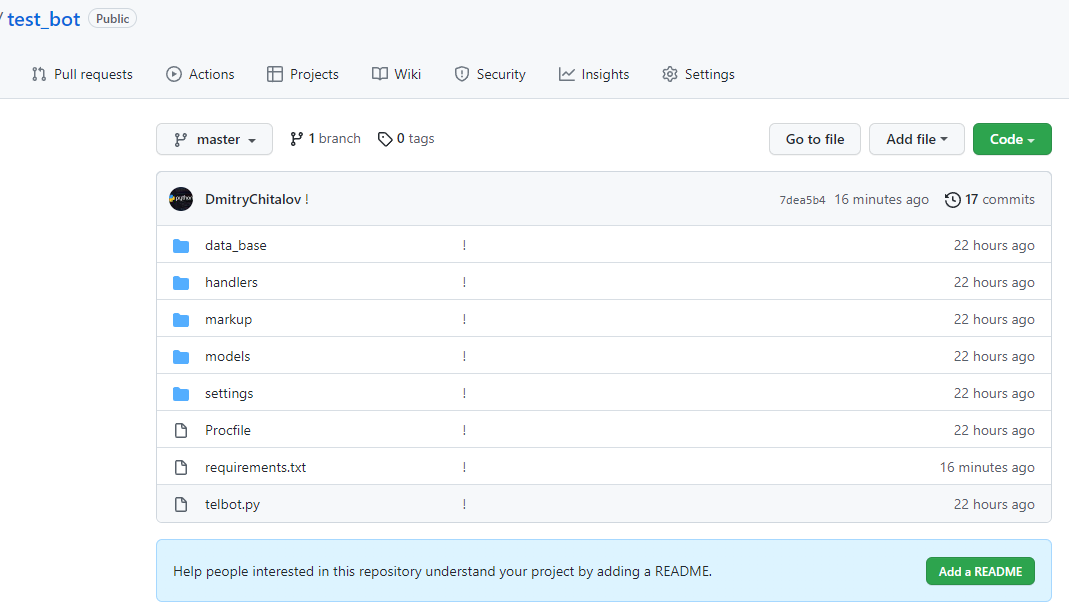
Здесь нужно сгенерировать новый токен и указать срок его действия.

Кроме того, укажем операции, на которые токен будет распространяться.



Токен нужно обязательно где-то сохранить, иначе его придется генерировать повторно.

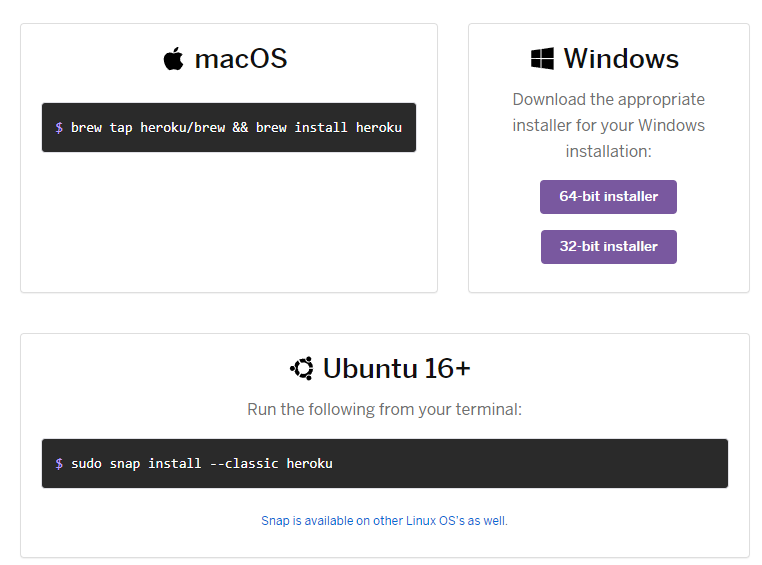
Проверим, что проект загружен на GitHub.



## устанавливаем консольный клиент для Heroku

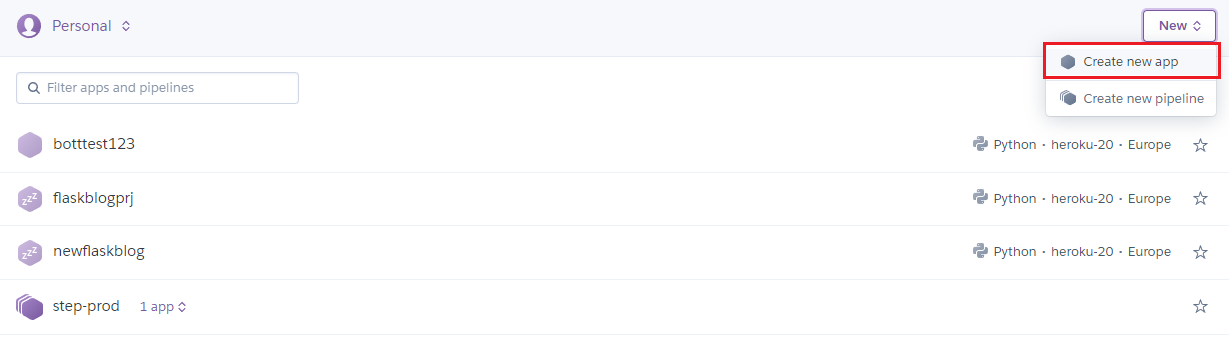
Нам потребуется выполнить несколько важных команд при развертывании.

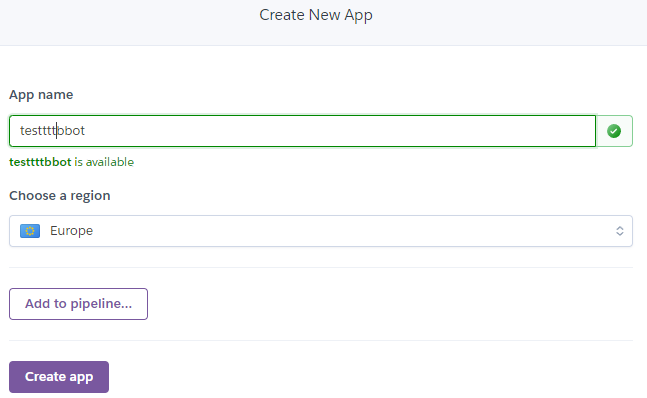
Для этого нам предстоит загрузить и установить приложение «интерфейс командной строки для Heroku».

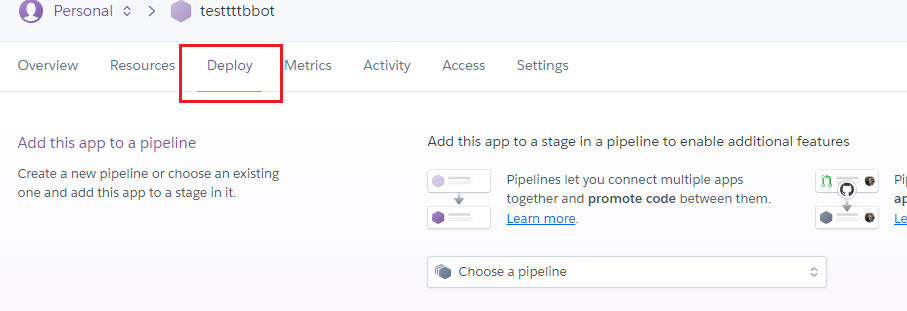
Переходим по ссылке <https://devcenter.heroku.com/articles/heroku-cli>, скачиваем и устанавливаем:

# ВЫПОЛНЯЕМ РАЗВЕРТЫВАНИЕ

Перейдем по ссылке:

<https://dashboard.heroku.com/apps> и выберем опцию «Create new app» (создать новое приложение).

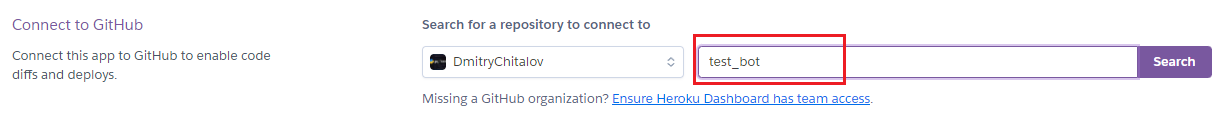
Заполним форму:

Перейдем на вкладку Deploy.

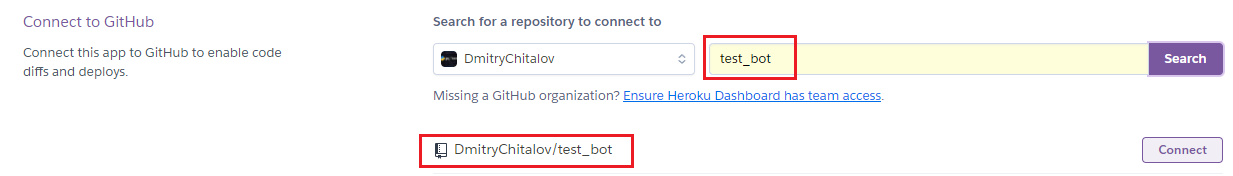
Выберем способ развертывания – через GitHub.

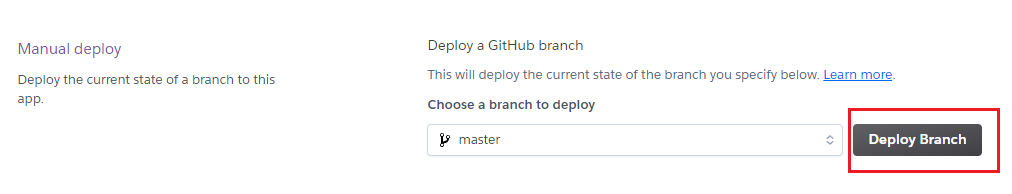


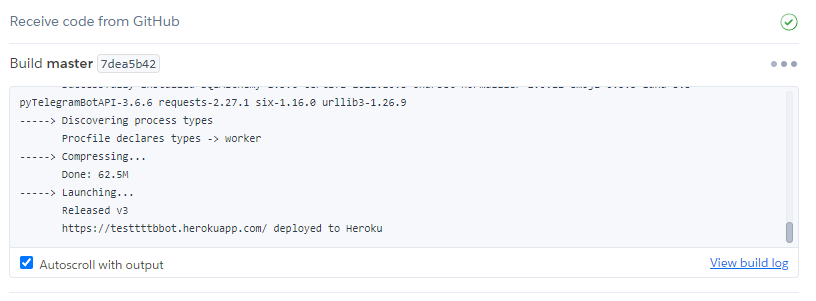
Выберем наш удаленный репозиторий.



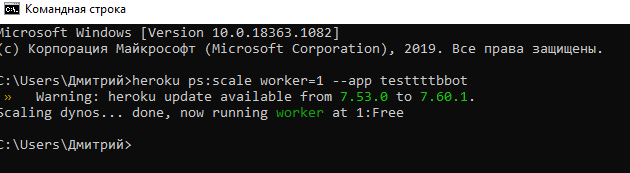
Нажмем Search и Connect.

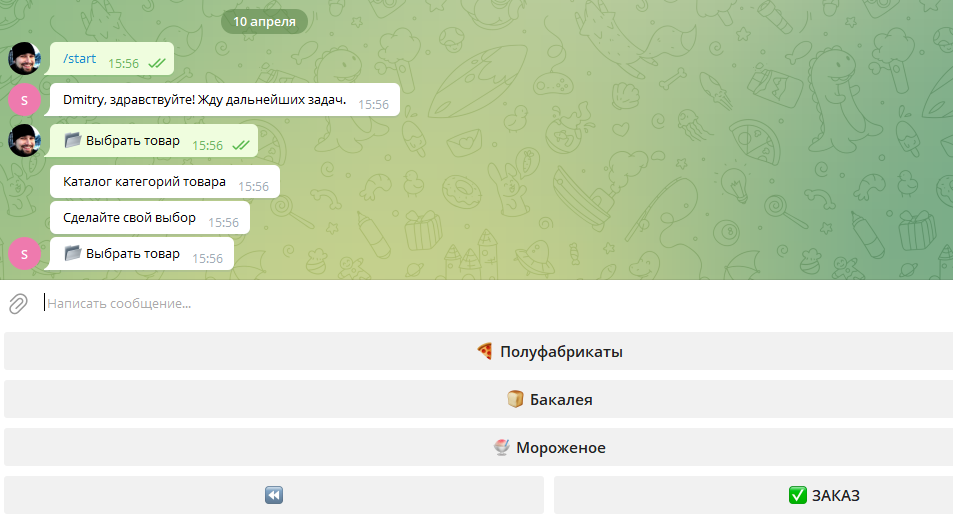
Выполним развертывание.

При успешном выполнении развертывания получим следующий результат.

Далее перейдем в командную строку и выполним:

heroku ps:scale worker=1 --app telebot\_store

Откроем Telegram, выполним поиск бота по имени (SimbaBot), выполним его запуск.

Все получилось. Деплой проекта успешно выполнен. 

**=**

# Вид сверху на рукопожатие над деловыми документамиЗАКЛЮЧЕНИЕ

Наш Telegram-бот доступен всем пользователям. Мы осуществили несколько шагов, в том числе подготовили проект к развертыванию, добавив служебные файлы, загрузили проект на GitHub и выполнили непосредственно деплой на Heroku.

## КЛЮЧЕВЫЕ ВЫВОДЫ

* Сервис Heroku предоставляет возможности как для учебного, так и «боевого» развертывания Python-проектов. Он надежен, функционален и популярен у разработчиков. С этим сервисом мы можем работать как посредством встроенного веб-интерфейса, так и посредством традицонной командной строки.
* Перед развертыванием мы создали дополнительные служебные файлы с настройками.
* В процессе развертывания мы через приложение SmartGit загрузили исходный код проекта на сервис GitHub, с которого далее отправили код уже на Heroku.
* В процессе развертывния может возникнуть немало ошибок, которые нужно устранить.