

Flask

Heroku. Публикация приложения



На этом уроке

1. Рассмотрим запуск проекта на Heroku от создания нового приложения до подключения к базе данных и выполнения миграций.

Оглавление

[Теория](#)

[Практическое задание](#)

[Создание продакшн конфига](#)

[Создание аккаунта на Heroku](#)

[Создание проекта на Heroku](#)

[Подключение аккаунта GitHub](#)

[Создание Procfile для запуска gunicorn](#)

[Procfile](#)

[Подключение Postgres](#)

[Добавление SECRET_KEY](#)

[Выполняем миграции](#)

[Итоги](#)

[Практическое задание](#)

[Дополнительные материалы](#)

Теория

Heroku — облачная PaaS-платформа, поддерживающая ряд языков программирования.

Platform as a Service — модель предоставления облачных вычислений, при которой потребитель получает доступ к использованию информационно-технологических платформ: операционных систем, систем управления базами данных, к связующему программному обеспечению, средствам разработки и тестирования, размещённым у провайдера.

Продакшн — боевой сервер, доступный внешним клиентам, в отличие от тестовой среды.

Практическое задание

Создание продакшн конфига

В файле `blog/configs.py` добавляем новый класс:

```
class ProductionConfig(BaseConfig):  
    SQLALCHEMY_DATABASE_URI = os.environ.get("SQLALCHEMY_DATABASE_URI")  
    SECRET_KEY = os.environ.get("SECRET_KEY")
```

Создание аккаунта на Heroku

Регистрируемся на Heroku.

Sign up for free and experience Heroku today

Free account

Create apps, connect databases and add-on services, and collaborate on your apps, for free.

Your app platform

A platform for apps, with app management & instant scaling, for development and production.

Deploy now

Go from code to running app in minutes. Deploy, scale, and deliver your app to the world.

First name *

First name

Last name *

Last name

Email address *

Email address

Company name

Company name

Role *

Role

Country *

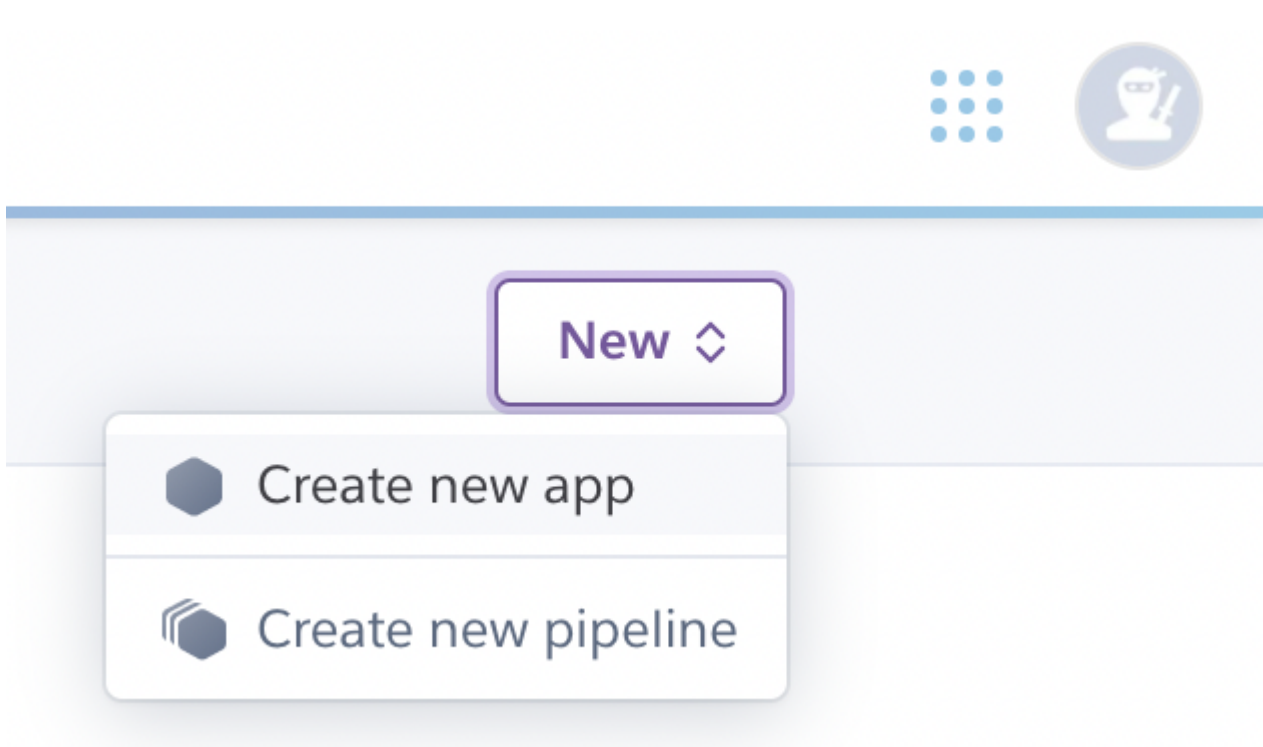
Country

Primary development language *

Select a language

Создание проекта на Heroku

Создаём новый проект.




Указываем имя, выбираем регион, нажимаем Create app:


A screenshot of the 'Create New App' form in Heroku. The form has a light blue header with the text 'Create New App'. Below the header, there is a section titled 'App name' with a text input field containing 'app-name'. Below that is a section titled 'Choose a region' with a dropdown menu showing 'Europe' and a flag icon. At the bottom of the form, there is a button labeled 'Add to pipeline...' and a large purple button labeled 'Create app'.


Подключение аккаунта GitHub

Выбираем GitHub как Deployment method и подключаем репозиторий.

Deployment method

 Heroku Git
Use Heroku CLI


 GitHub
Connect to GitHub

 Container Registry
Use Heroku CLI

Connect to GitHub

Connect this app to GitHub to enable code diffs and deploys.

Search for a repository to connect to

 mahenzon

repo-name



Search

Missing a GitHub organization? [Ensure Heroku Dashboard has team access.](#)


Выставляем настройки для автоматической публикации при появлении новых коммитов в master:



App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to  mahenzon/flask-lessons by  mahenzon


Disconnect...



 Releases in the [activity feed](#) link to GitHub to view commit diffs


 Automatically deploys from  master

Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

 You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions [here](#).

 Automatic deploys from  master are enabled

Every push to  will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch in GitHub is always in a deployable state and any tests have passed before you push. [Learn more](#).

☐ Wait for CI to pass before deploy
Only enable this option if you have a Continuous Integration service configured on your repo.

Disable Automatic Deploys


Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more](#).

Choose a branch to deploy

 master

Deploy Branch

Создание Procfile для запуска gunicorn

Procfile

Procfile — это файл конфигурации (инструкция по запуску), который прочитает Heroku и запустит app:

```
web: gunicorn wsgi:app --log-file -
```

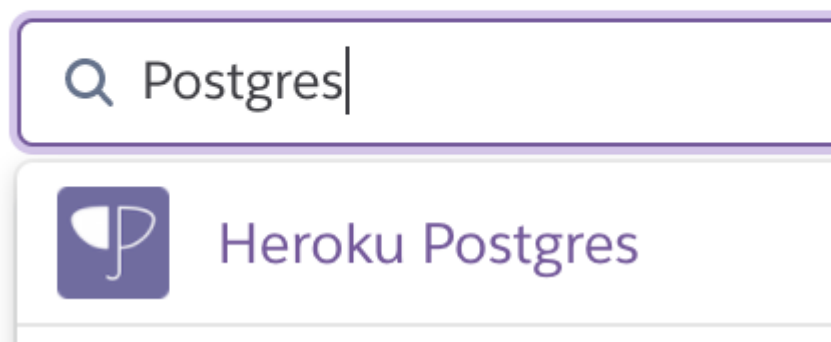
Нам не обязательно устанавливать gunicorn на локальную машину, но обязательно добавить его в requirements.txt.

Теперь можно выполнить push в master ветку. Будет выполнена автоматическая публикация.

Подключение Postgres

Нам необходимо подключить базу данных к нашему приложению. Heroku позволяет сделать это при помощи аддонов. В настройках приложения переходим на вкладку Resources, в секции Add-ons ищем Postgres и подключаем к нашему приложению:

Add-ons



К приложению будет автоматически добавлена переменная окружения `DATABASE_URL` с ссылкой на подключение к базе данных. Она нас немного не устраивает по формату и имени, поэтому копируем её из настроек аддона Postgres и создаём свою `SQLALCHEMY_DATABASE_URI`. Для этого в настройках приложения в разделе Settings добавляем переменные окружения:

Config Vars

Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.

Config Vars

[Hide Config Vars](#)

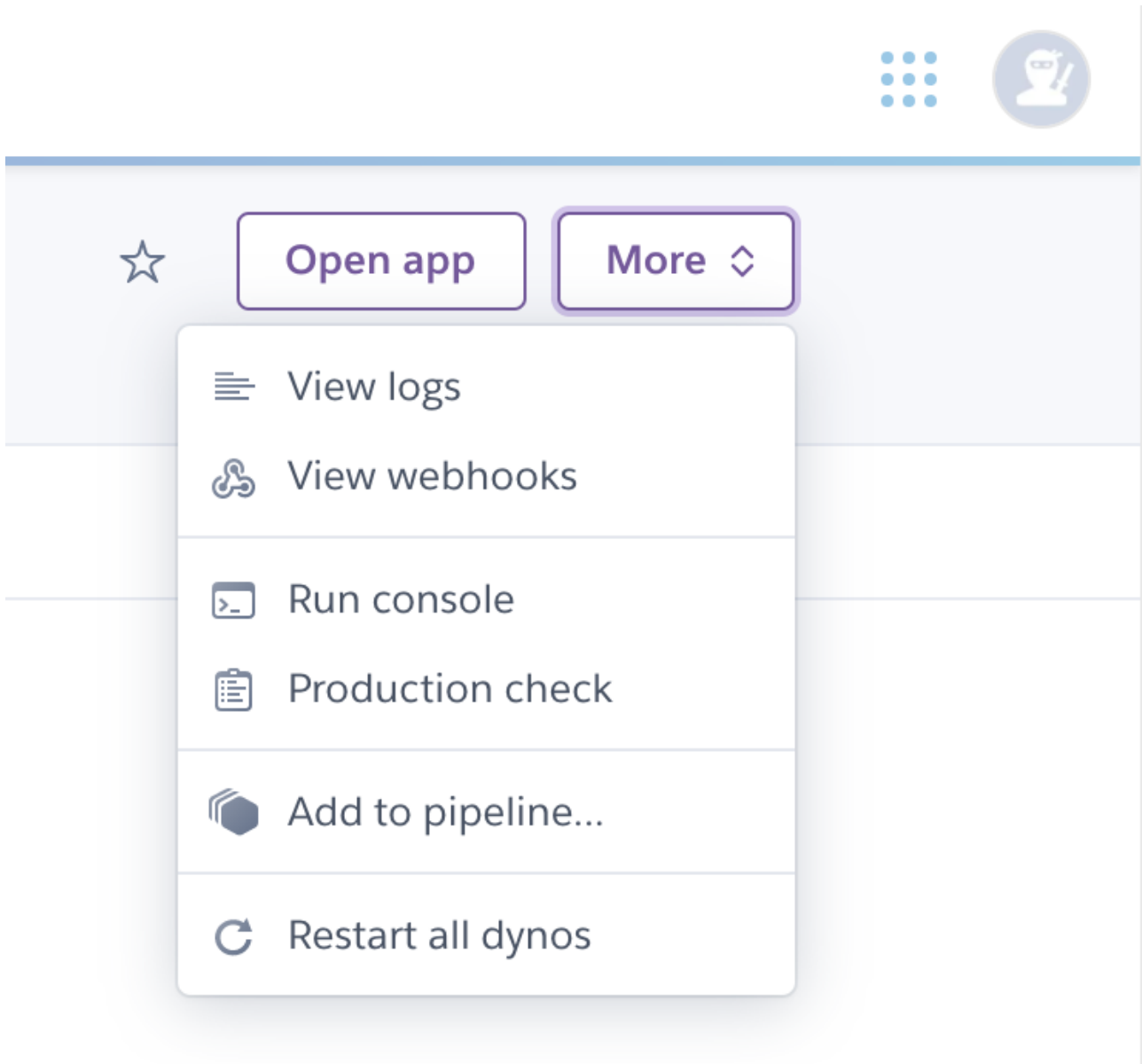
DATABASE_URL	postgres://tqjdgnoou	 
SECRET_KEY	\xe7\x01\xb0\xec\x	 
SQLALCHEMY_DATABASE_	postgresql://tqjdgno	 
KEY	VALUE	

Добавление SECRET_KEY

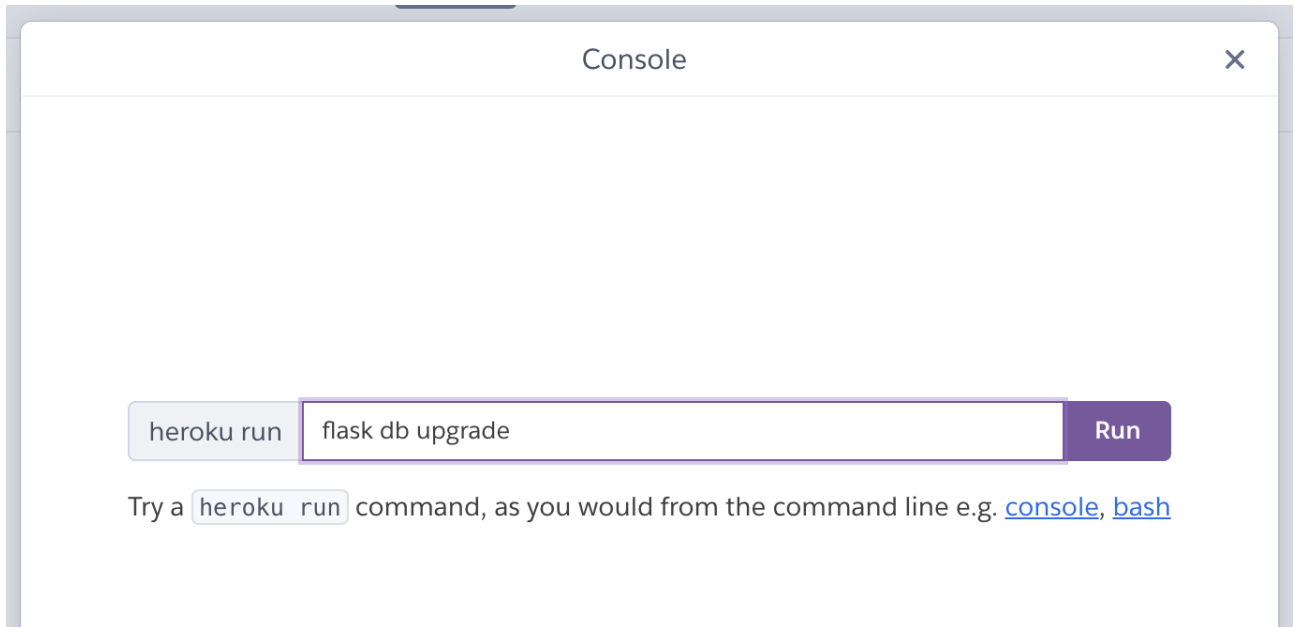
Создаём случайный секретный ключ командой `python -c 'import os; print(os.urandom(16))'` и добавляем переменную окружения `SECRET_KEY`. Теперь мы можем подключаться к базе данных, а также работать с системой авторизации.

Выполняем миграции

Нам осталось создать необходимые таблицы, а также пользователя администратора. Для этого нам нужен терминал на удалённой машине. Чтобы получить к нему доступ, в настройках приложения справа выбираем действие Run console.



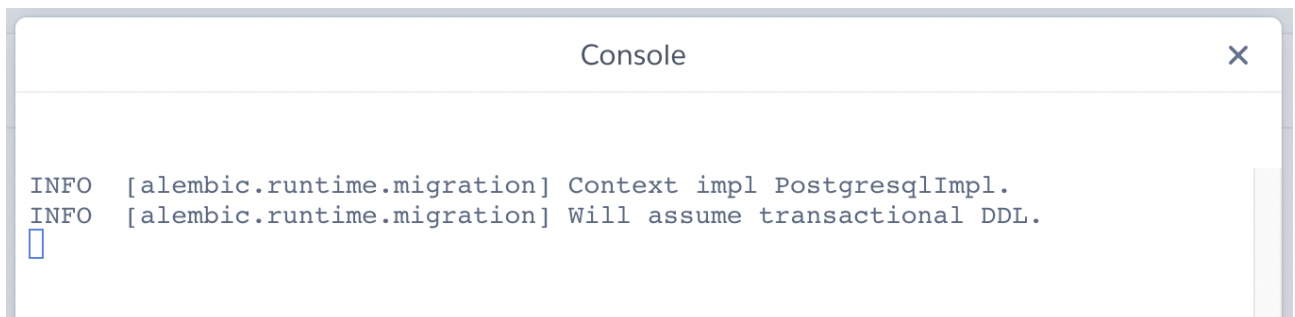
Нам откроется консоль:



Выполняем там команды миграций:

- flask db upgrade;
- flask create-admin.

Мы получим соответствующий вывод в консоли (на скриншоте применение миграции, когда все миграции уже выполнены. При первом прогоне будут написаны все выполненные миграции).



Теперь можно перейти к приложению при помощи кнопки Open app. Этой ссылкой можно поделиться с друзьями, родственниками и т.д. — она доступна из любой точки мира, если там доступны сервера Heroku.

Итоги

На занятии мы запустили свой Flask проект на Heroku.

Практическое задание

1. Запустить свой Flask проект на Heroku.

Дополнительные материалы

1. [Custom Commands](#).
2. [User class Flask-Login required props](#).
3. [flask_login.UserMixin](#).
4. [SECRET_KEY](#).
5. [Remember Me](#).
6. [Alternative Tokens](#).