

# CI/CD



Александр  
Ульянцев



# АЛЕКСАНДР УЛЪЯНЦЕВ

G-Core labs, Backend Software Engineer

 [alex@uliantsev.name](mailto:alex@uliantsev.name)

 [@res1d3nt](https://t.me/res1d3nt)

---

# План занятия

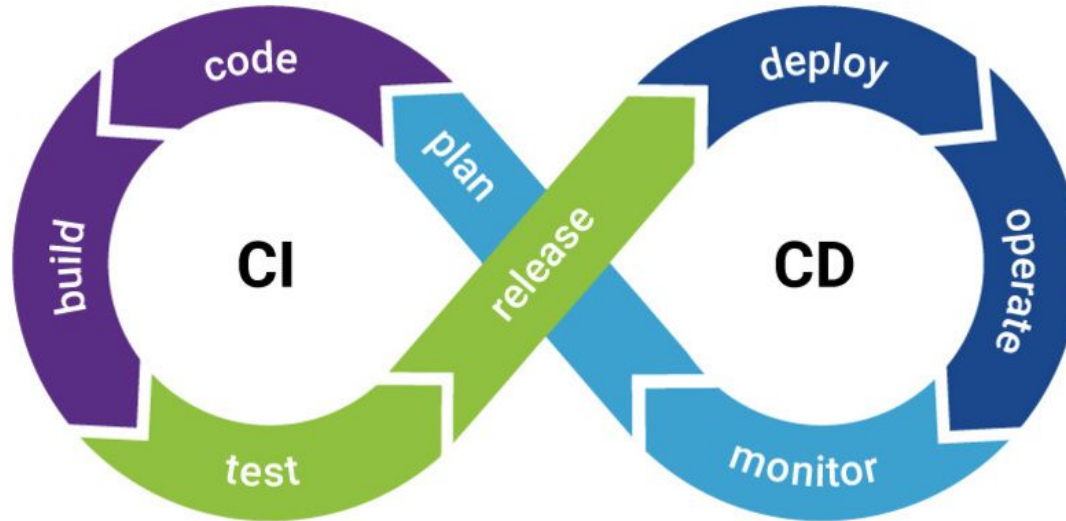
1. [Процессы](#)
2. [Сервисы](#)
3. [GitHub Actions](#)
4. [Heroku](#)
5. [Подготовка кода](#)
6. [Итоги](#)
7. [Домашнее задание](#)



# Процессы

# Определение

**CI/CD** (continuous Integration / continuous delivery) – комплекс мероприятий для непрерывной интеграции и доставки программного обеспечения.



Изображение: <https://www.synopsys.com/glossary/what-is-cicd.html>

Одна из задач DevOps-ов – настроить архитектуру для всех CI/CD процессов. Разработчику полезно владеть базовыми DevOps навыками для запуска своих проектов для демонстрации.

# Workflow

Существуют различные схемы взаимодействия процессов разработки и поставки.

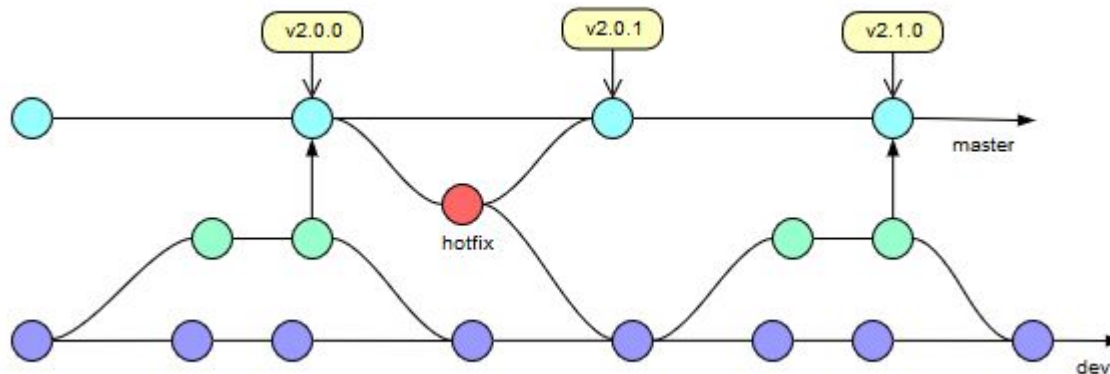
Centralized, Forking, Gitflow, GitHub, GitLab и др. workflow.

Схема выбирается от целей команды.

Например, по триггеру приложение разворачивается на разных окружениях:

- production, финальная версия из основной ветки.
- stage, промежуточная версия для QA и бизнес тестов.
- develop, сырая версия из ветки с разработкой.

**Gitflow:**



Изображение: <http://blogs.stonesteps.ca/showpost.aspx?pid=67>



# Сервисы

# Сервисы интеграции и развертки

Существует различные инструменты интеграции, все они основаны на общем принципе конфигурации.

По факту целевого события скрипт интеграции запускает команду для запуска тестов или сборки.

Инструменты интеграции, встроенные в **git хостинги**:

- [GitHub Actions](#)
- [GitLab Pipelines](#)
- [Bitbucket Pipelines](#)

**Специализированные** инструменты интеграции, подключаемые к хостингу:

- [circleci](#)
- [semaphore](#)
- [jenkins](#)



# Типичный процесс

На примере Django проекта, после коммита изменений на внешнем репозитории (GitHub) автоматически запускать тесты и при успешных проверках разворачивать изменения на сервере (Heroku).

1. Создаём репозиторий в GitHub.
2. Отправляем в репозиторий исходный код (с валидными тестами и линтерами).
3. Настраиваем CI на GitHub для запуска проверок (чтобы убедиться, что тесты проходят и линтер не ругается).
4. Создаем Docker образ для публикации приложения.
5. Разворачиваем Docker образ на Heroku.

# Альтернатива

Для некоторых связок есть упрощенный вариант, как раз GitHub+Heroku+Django имеет очень удобный механизм интеграции:

1. Создаём репозиторий в GitHub.
2. Отправляем в репозиторий исходный код (с валидными тестами и линтерами).
3. Настраиваем CI на GitHub для запуска проверок (чтобы убедиться, что тесты проходят и линтер не ругается).
4. Разворачиваем проект на Heroku.



# GitHub Actions

# GitHub Actions

GitHub Actions сканирует каталог `.github/workflows` на наличие скриптов в **yaml** формате, соответствующий [шаблону](#).

На вкладке **actions** GitHub проекта доступны стандартные шаблоны для типовых проектов, например, «Python application».

**jobs** — перечень задач, сборка (build), тестирование (tests) или др.

**branches** — перечень веток, по действию в которых следует запускать интеграцию.

- Например, версия разработки будет собираться на тестовый сервер только из ветки `develop`, а финальная версия всегда соответствовать состоянию ветки `main`.

**steps** — последовательность действий для интеграции.

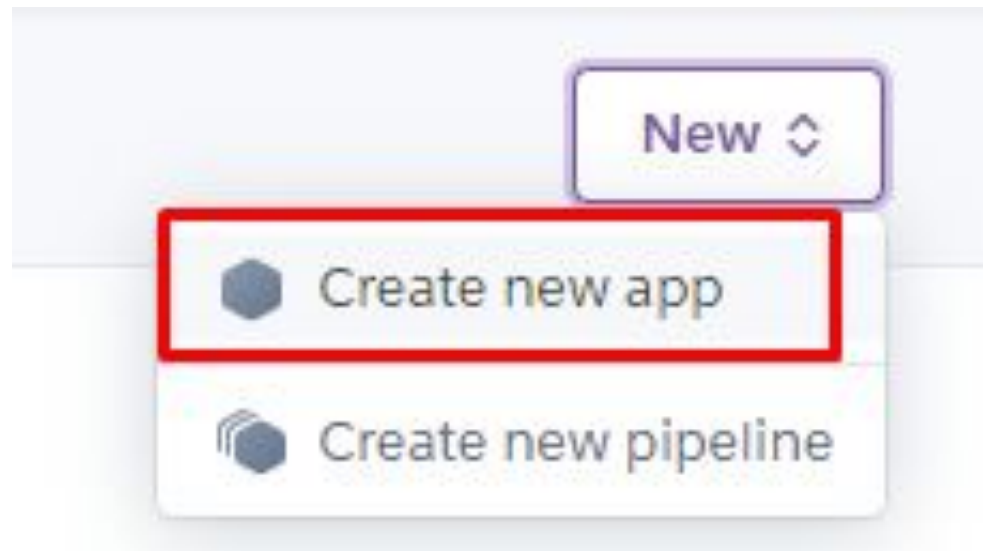
- Шаги могут использовать уже заготовленные скрипты, командой **uses**.
- Создавать новые командой **name** и **run**, где задается имя и что нужно выполнить.

Подробнее о синтаксисе скрипта в [документации](#).



# Heroku

## Создаем новое приложение



## Выбираем имя и регион

Create New App

App name

artrey-web-chat

artrey-web-chat is available

Choose a region

Europe

Add to pipeline...

Create app

Имя будет использоваться в url для доступа к вашему сайту.

Регион лучше выбрать поближе к себе территориально.

# Подключаем свой репозиторий

Personal > artrey-web-chat



Open app

More

Overview

Resources

Deploy

Metrics

Activity

Access

Settings

## Add this app to a pipeline

Create a new pipeline or choose an existing one and add this app to a stage in it.

## Add this app to a stage in a pipeline to enable additional features



Pipelines let you connect multiple apps together and **promote code** between them. [Learn more.](#)



Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. [Learn more.](#)

Choose a pipeline

## 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

artrey

web-chat

Search

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

artrey/web-chat

Connect



# Включаем автоматическую публикацию

## App connected to GitHub

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

Connected to  [artrey/web-chat](#) by  [artrey](#)

Disconnect...

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

## 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](#).

Enable automatic deploys from GitHub

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

Choose a branch to deploy



master



☒ Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

Enable Automatic Deploys

# Добавляем БД postgres

The screenshot shows the Heroku dashboard for the application 'artrey-web-chat'. The 'Resources' tab is highlighted in the top navigation bar. Below the navigation bar, a message states: 'This app has no process types yet. Add a Procfile to your app in order to define its process types. [Learn more](#)'. The 'Add-ons' section is visible, with a search bar containing 'postgre'. The search results list three add-ons: 'Draxlr for Postgres / MySQL', 'Heroku Postgres' (which is highlighted with a red box), and 'Trevor.io for Postgres / MySQL'. The 'Estimated monthly cost' for the add-ons is shown as '\$0.00'.

Personal > artrey-web-chat

GitHub artrey/web-chat master

Overview Resources Deploy Metrics Activity Access Settings

Dynos

This app has no process types yet  
Add a Procfile to your app in order to define its process types. [Learn more](#)




Add-ons Find more add-ons

Q postgre

- Draxlr for Postgres / MySQL
- Heroku Postgres
- Trevor.io for Postgres / MySQL

Estimated monthly cost \$0.00

# Выбираем бесплатный план



Heroku Postgresartrey-web-chat

By choosing "Online Order Form", this will add **Heroku Postgres** on your personal **artrey-web-chat** application.

Plan name

Hobby Dev – Free

[View add-on details in Elements Marketplace](#)

By submitting this order form, you agree that the Add-on is governed by the applicable provider's terms of use, and the Heroku Services are governed by the [Salesforce Master Subscription Agreement](#), unless (except for free customers) you have entered into a written Master Subscription Agreement executed by SFDC for the Heroku Services as referenced in the Documentation.

Submit Order Form



# Подготовка кода



# Документация

Актуальные правила для публикации Django-проектов находятся по этой ссылке:

<https://devcenter.heroku.com/articles/django-app-configuration>

# Действия

1. В [requirements.txt](#) добавить две библиотеки:

- gunicorn
- django-heroku

2. Создать файл [Procfile](#) в корне проекта с текстом:

```
web: gunicorn название\_вашего\_проекта.wsgi
```

3. В [settings.py](#) добавить строки:

- в самое начало:

```
import django_heroku
```

- в самый конец:

```
django_heroku.settings(locals())
```

# Миграции

Применить миграции для БД postgres на heroku можно несколькими способами:

- С помощью [heroku CLI](#) (для самостоятельного изучения)
- Добавить строчку в [Procfile](#) (в самое начало):

```
release: python manage.py migrate
```



# Пример проекта

<https://github.com/artrey/web-chat>





## Итоги

- Познакомились с автоматическим запуском тестов при публикации
- Настроили публикацию актуальной версии приложения из основной ветки

---

# Домашнее задание

- Обязательного домашнего задания по лекции **нет**
- Есть **необязательное** [домашнее задание](#)

Вопросы по теме задавайте в чате Slack!

**Задавайте вопросы и  
пишите отзыв о лекции!**

**Александр Ульянов**