Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тихоокеанский государственный университет»

Высшая школа кибернетики и цифровых технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

ПО КУРСУ «ОСНОВЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ»

«АВТОМАТИЗАЦИЯ СБОРКИ И РАЗВЁРТЫВАНИЯ (CI/CD)»

Выполнил: студент 2 курса группы ПО(б) – 31

Якшин Кирилл Евгеньевич

Проверила: ассистент ВШ КЦТ

Балабасова Алиса Дмитриевна

Хабаровск 2025 г.

Цель работы:

Освоить настройку автоматической сборки и развертывания кода с использованием GitHub Actions и Jenkins.

Задание:

1. Автоматизировать любой проект с помощью GitHub Actions.

2. Автоматизировать любой проект с помощью Jenkins (или другого CI/CD на выбор).

3. Проверить работу при внесении изменений (скриншоты до/после).

Ход работы:

Создадим файлы workflow для GitHub Actions:

* lint\_py.yml – запускает pylint для проверки python файлов на стиль и ошибки в написании;
* release.yml – отвечает за развертывание проекта (выпуск релиза в GitHub).

Для Jenkins напишем скрипт pipeline. С его помощью Jenkins будет проверять репозиторий на обновления и развертывать новые версии в локальное окружение.

В файл README.md добавим бейджи статуса Release и Python Lint, количества коммитов с последнего релиза и бейдж даты последнего коммита (рисунок 1):

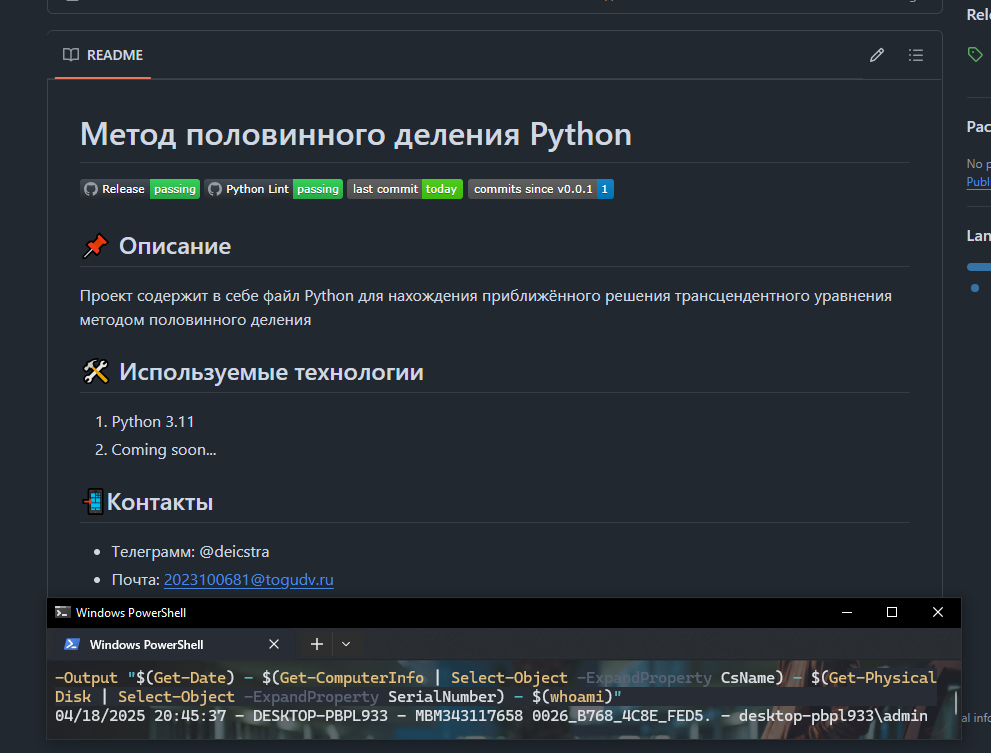


Рисунок 1 – Readme GitHub

Внесем изменения в файл LB\_1.py после чего создадим новую версию с помощью команды git tag и запушим её в репозиторий (рисунок 2).

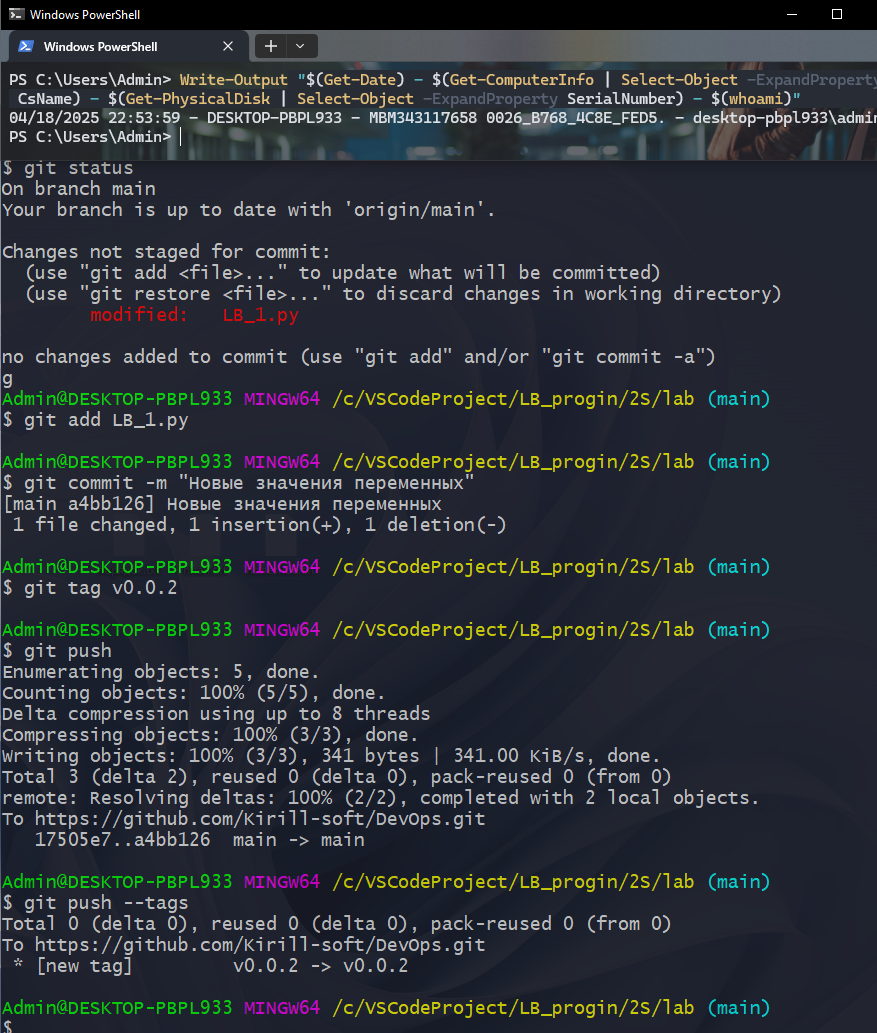


Рисунок 2 – Создание новой версии (тега)

Это запустит развертывание в GitHub и Linter для проверки Python файла.

Через несколько секунд, когда работа GitHub Actions завершиться можно будет увидеть краткий результат выполнения работ (рисунок 3), какие были выполнены, а какие пропущены. Так работа create\_tag была пропущена, так как workflow был запущен пушем тега и, следовательно, создавать его не требуется.

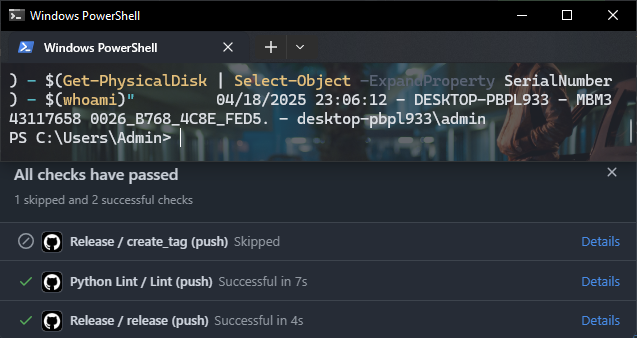


Рисунок 3 – Результат исполнения работ GitHub Actions

В результате создаться следующих релиз GitHub (рисунок 4):

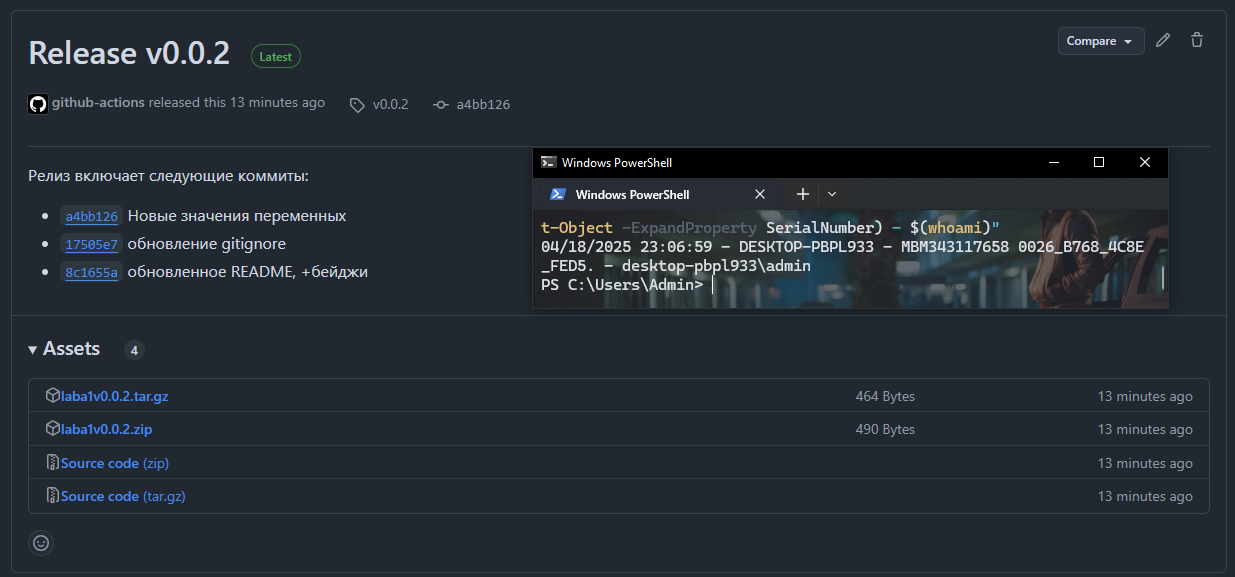


Рисунок 4 – Релиз GitHub

В это же время Jenkins автоматически обнаружит изменения в репозитории и развернет новую версию в локальной папке (рисунок 5).

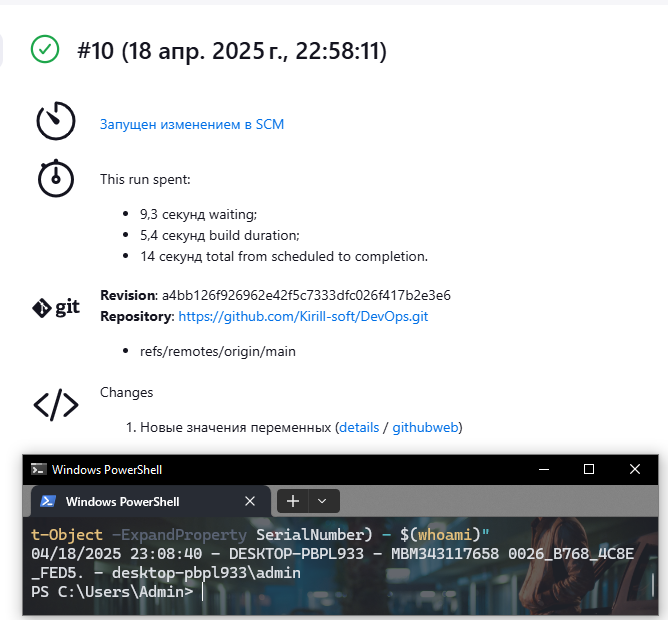


Рисунок 5 – Сводка результата выполнения Jenkins

Содержимое локальной папки, в которую был развернут релиз выглядит следующим образом (рисунок 6):

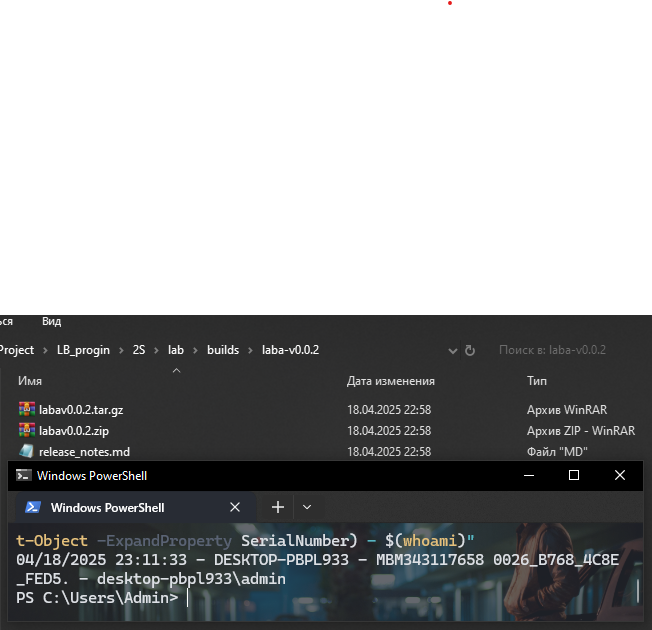


Рисунок 8 – Папка с развернутым релизом Jenkins

В данной папке можно увидеть файлы проекта в архивах для Windows и Linux, а также список изменений в файле release\_notes.md – такой же список изменений используется в GitHub.

Вывод: в результате выполнения работы была реализована автоматическая сборка и развертывание кода с помощью GitHub Actions и Jenkins как в локальном, так и вы облачном окружении. Опыт, полученный в результате выполнения данной работы будет полезен в дальнейшем при использовании систем CI/CD в разработке проектов.