

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 РАЗВЕРТЫВАНИЯ»

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

Плеханов Михаил Анатольевич

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

Ход работы:

В папке “.github/workflows” создать файл с расширением “.yml”.

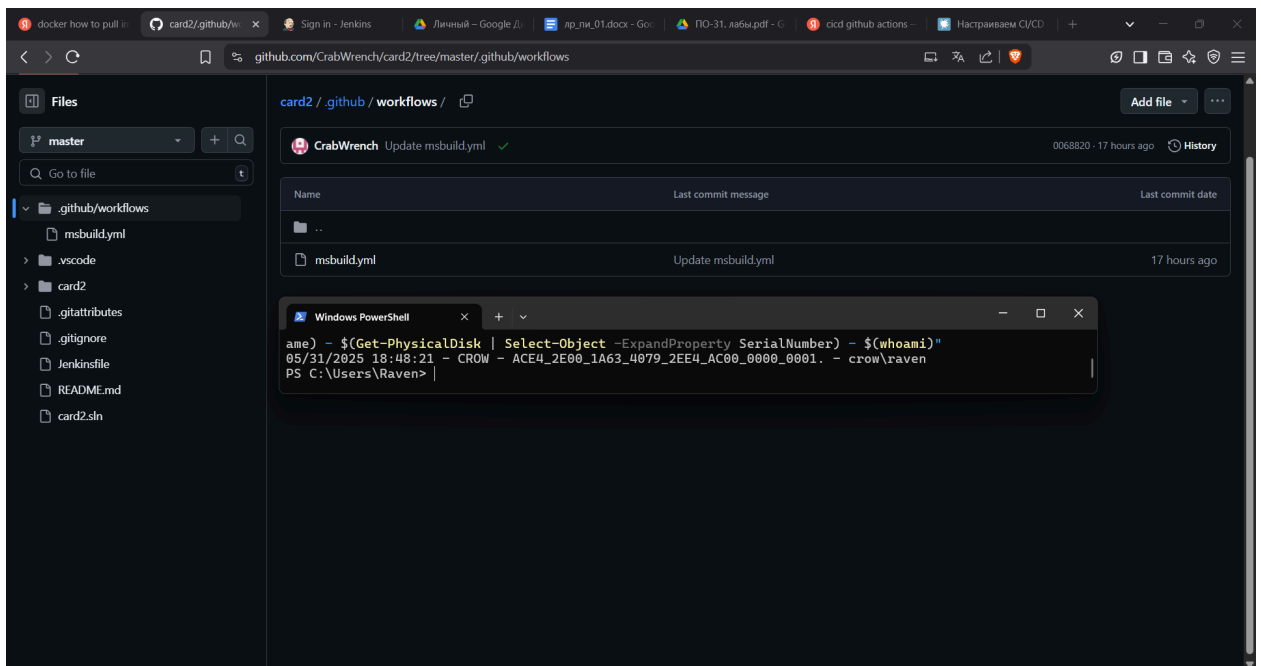


Рисунок 1 – файл с расширением “.yml”

В созданном файле пишем код для проверки работы проекта при пуше и пуле репозитория.

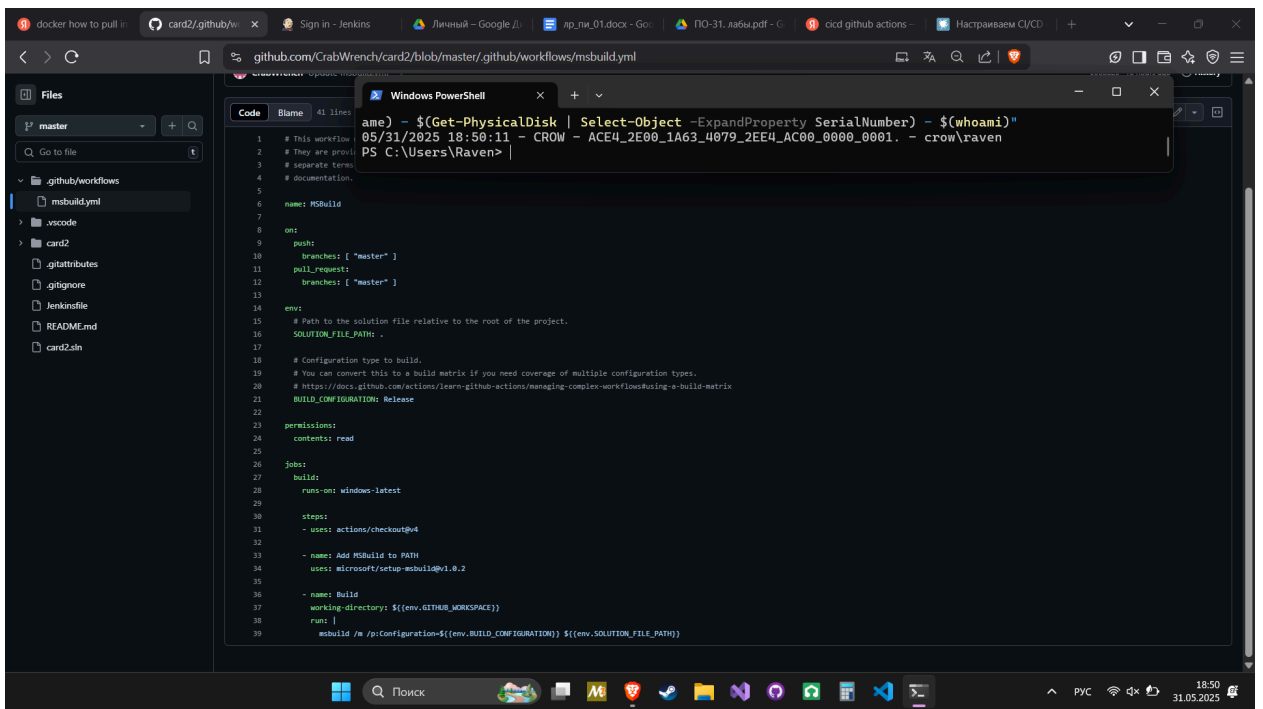


Рисунок 2 – код в “msbuild.yml”

Создаем папку “actions-runner” через PowerShell заходим в папку и создаем “Runner” под именем “CROW”.

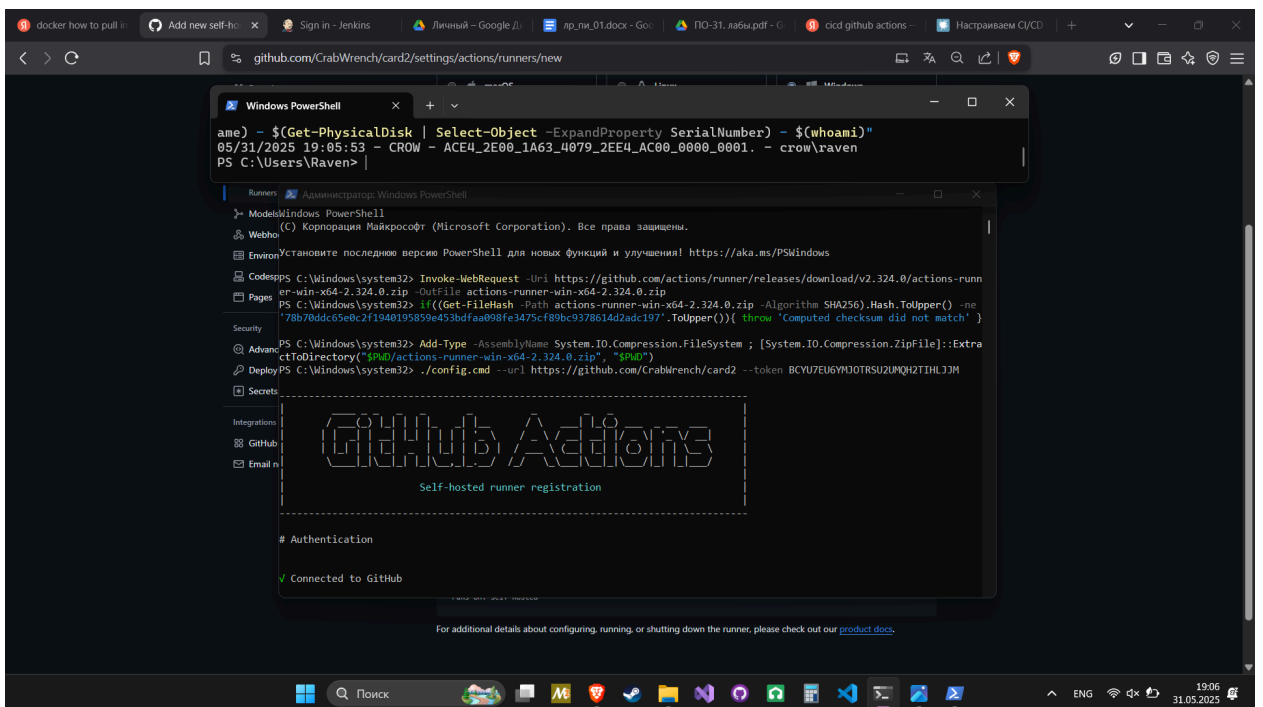


Рисунок 3 – вид консоли при создании ранера

Готовый ранер “CROW”.

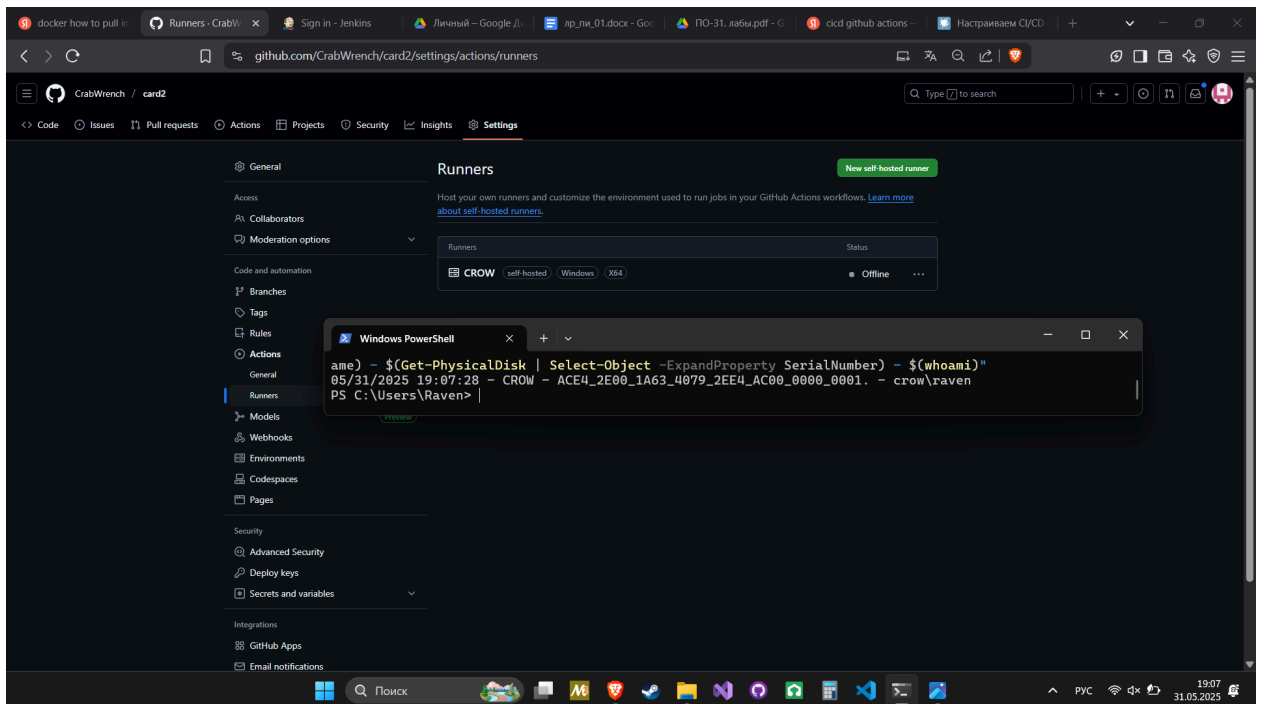


Рисунок 4 – готовый ранер “CROW”

Проверяем работу. Вносим изменения в проект и смотрим как проходит проверка и сборка.

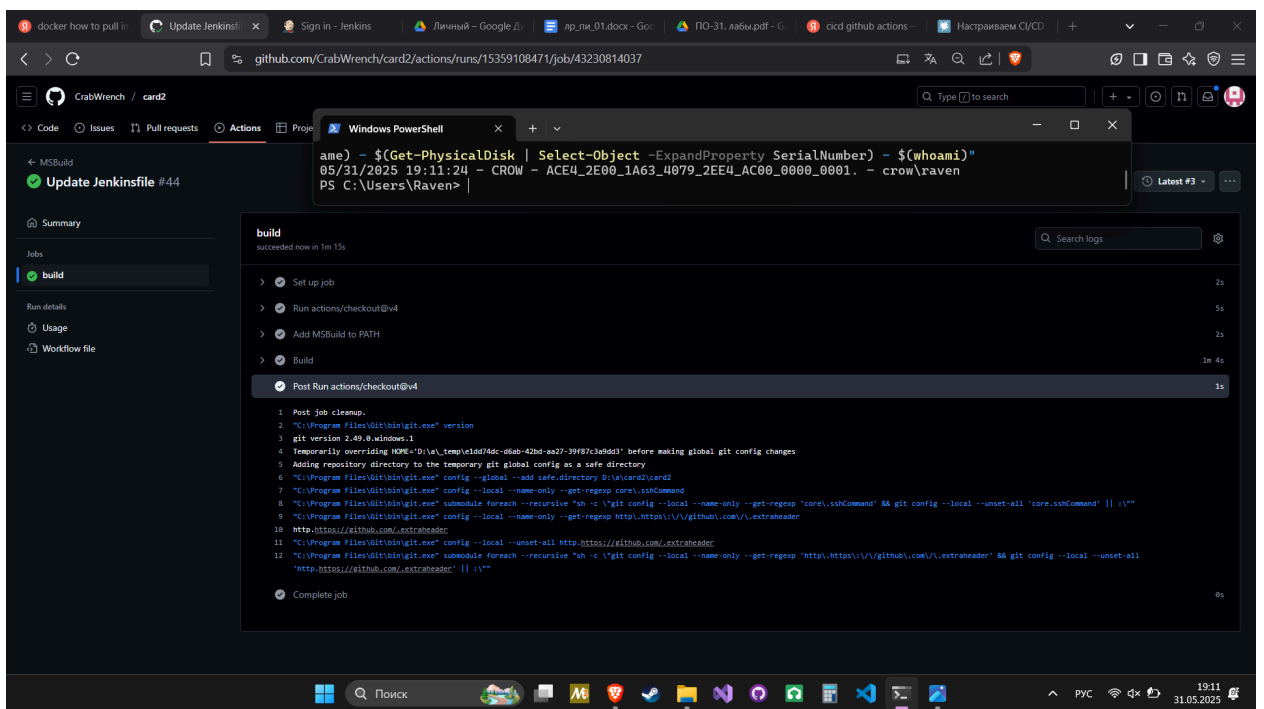


Рисунок 5 – успешная с проверка и сборка

Новая ветка "feature-branch".

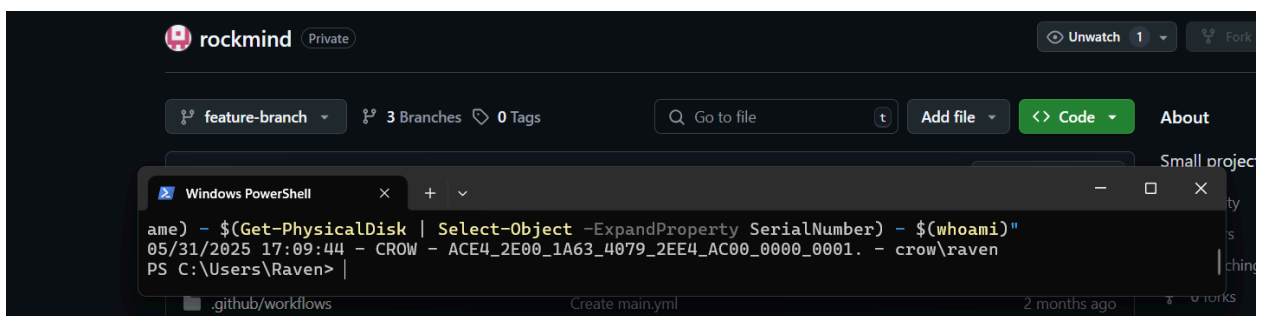


Рисунок 6 – ветка "feature-branch"

В корневой папке проекта создаем “Jenkinsfile”.

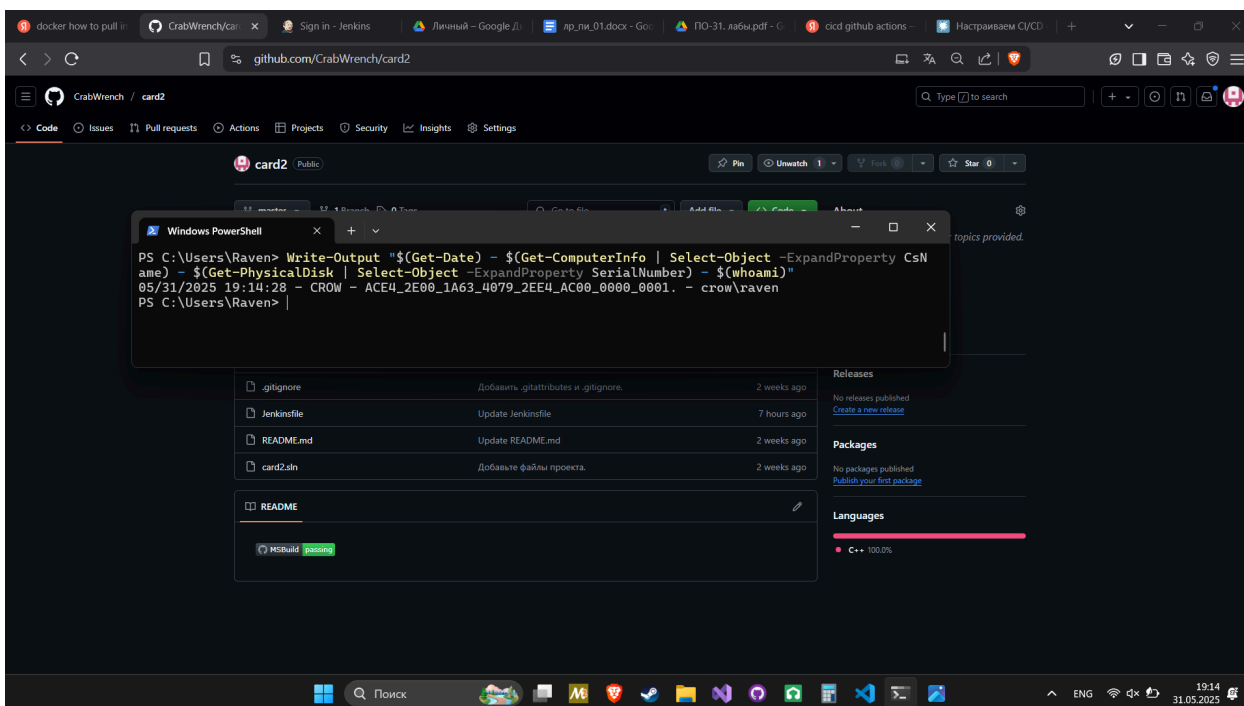


Рисунок 7 – “Jenkinsfile” в проекте

Скачиваем “Jenkins” на устройство и регистрируемся. Заходим в свой аккаунт.

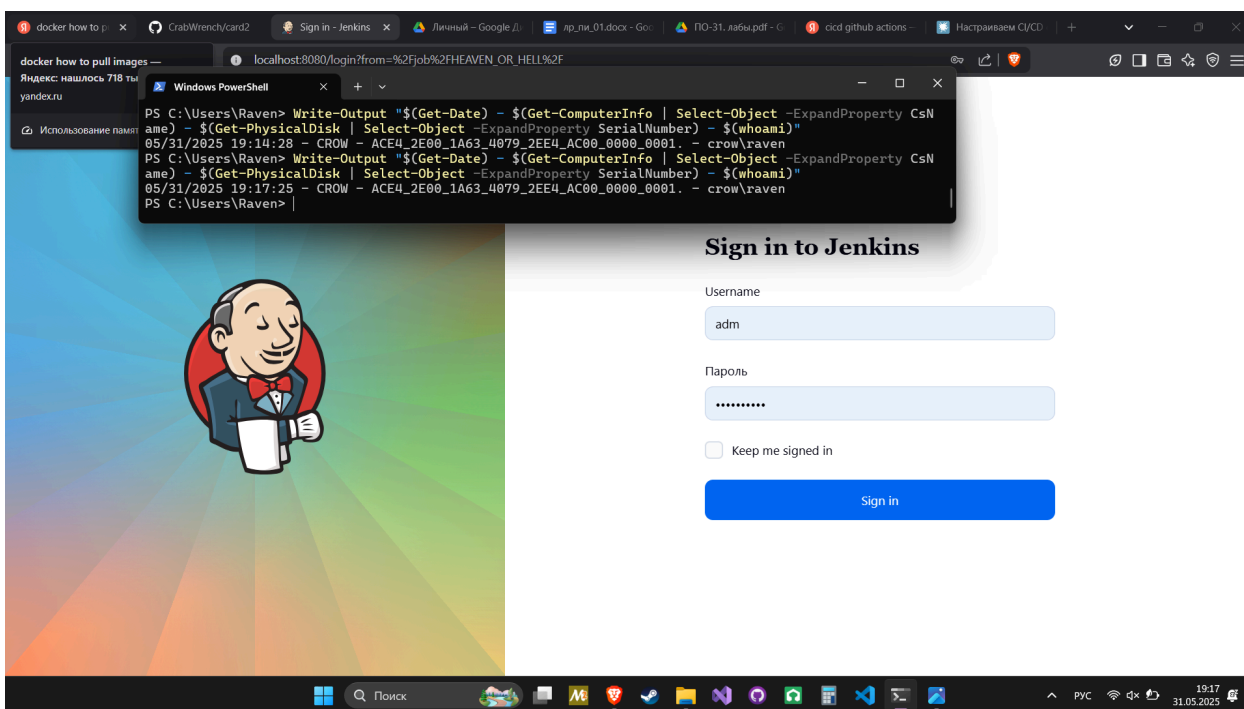


Рисунок 8 – вход в “Jenkins”

В “Jenkinsfile” пишем следующее

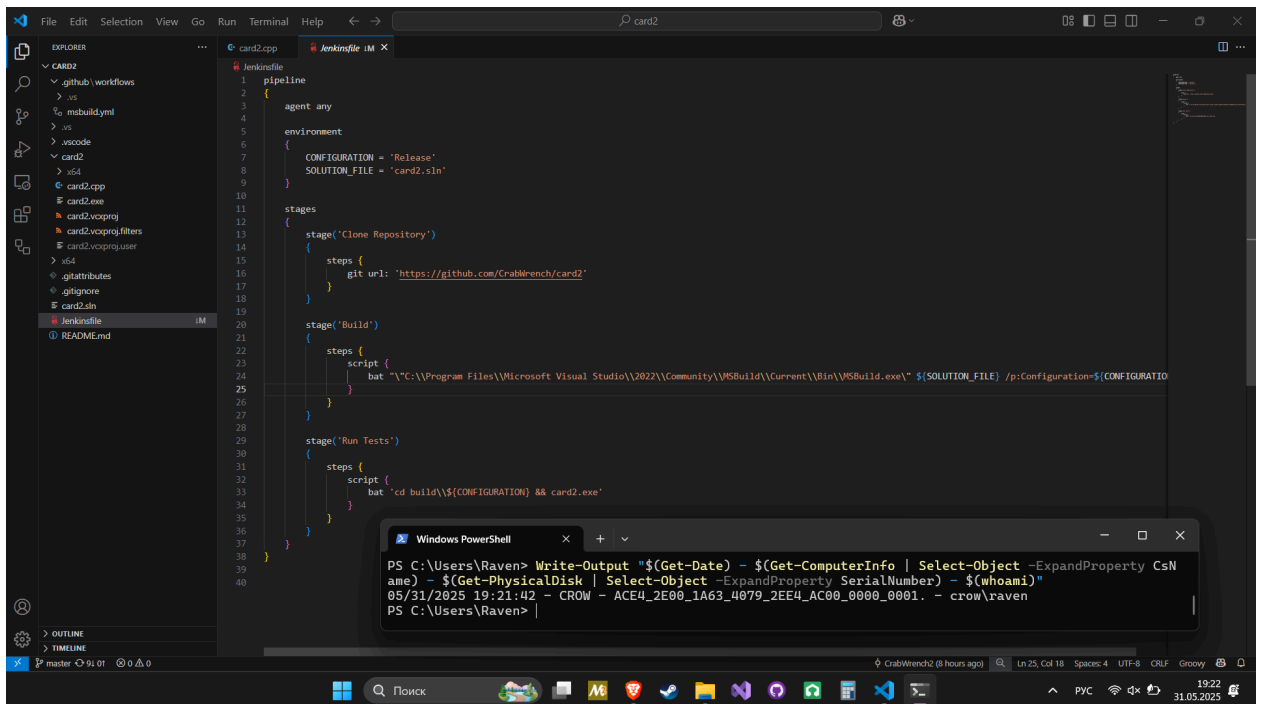


Рисунок 9 – код в “Jenkinsfile”

Создаем пайплайн.

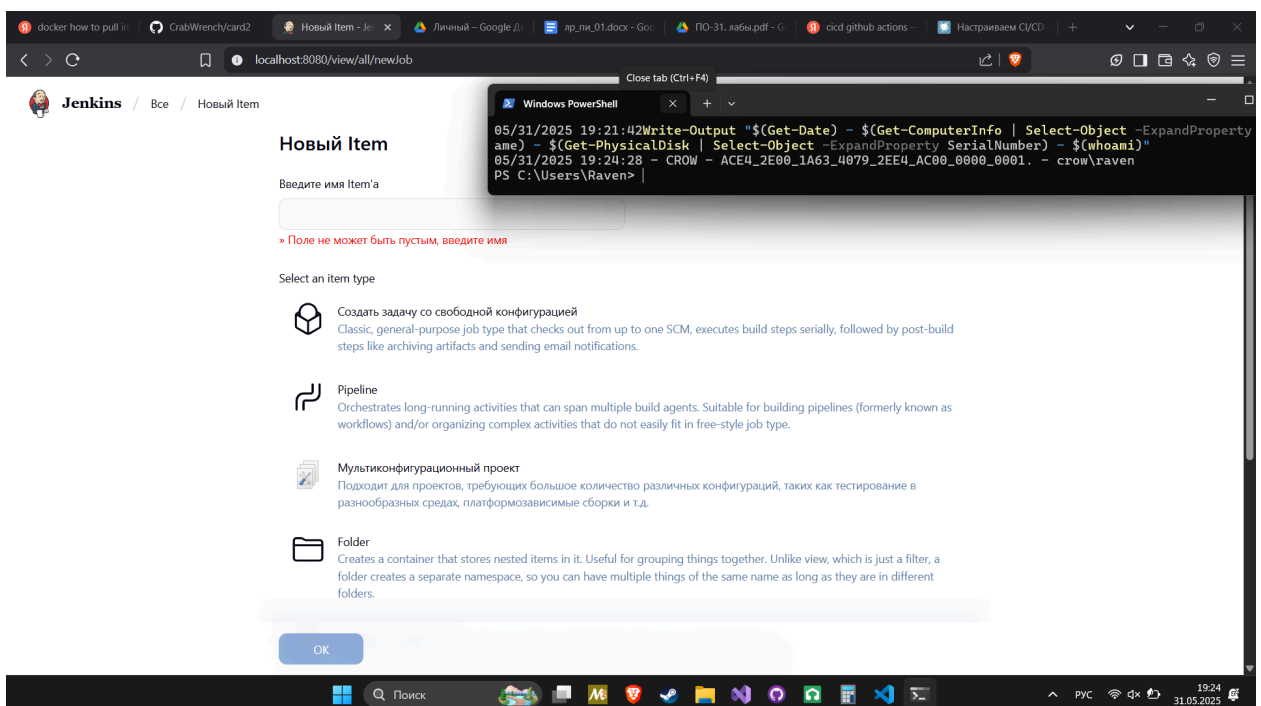


Рисунок 10 – создание пайплайна

Проверяем работу пайплайна

The screenshot displays the Jenkins web interface for a pipeline named 'HEAVEN_OR_HELL'. The status is 'SUCCESS' with a green checkmark. The pipeline consists of five stages: 'Declarative: Checkout SCM', 'Clone Repository', 'Build', 'Run Tests', and 'Declarative: Post Actions'. The 'Run Tests' stage is currently active, showing a progress bar and a duration of 5s. A Windows PowerShell terminal window is overlaid on the right, displaying the output of the 'Run Tests' stage, which includes the command 'Get-PhysicalDisk' and the output 'CROW - ACE4_2E00_1A63_4079_2EE4_AC00_0000_0001. - crow\raven'. The terminal also shows the 'whoami' command output 'CROW'.

Stage	Declarative: Checkout SCM	Clone Repository	Build	Run Tests	Declarative: Post Actions
Average stage times: (full run time: ~16s)	2s	2s	2s	5s	143ms
#311 май 31 12:19 No Changes	2s	2s	1s	5s	135ms
#30 май 31 12:16 3 commits	2s	2s	2s	6s	152ms
#29 май 31 12:13 No Changes					
#28 май 31 12:12 No Changes					
#27 май 31 No					

Рисунок 11 – успешная проверка

Вывод:

В ходе выполнения лабораторной работы, целью которой было освоение настройки автоматической сборки и развертывания кода с использованием GitHub Actions и Jenkins, я успешно реализовал автоматизацию как для одного из проектов на GitHub, так и для локального проекта с использованием Jenkins.