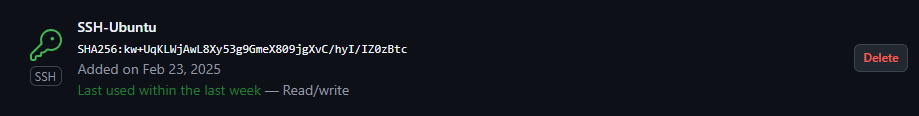
# Л13. CI/CD – непрерывная интеграция, развертывание и доставка

Харисов Ринат ПМИ-5

Проверяем что публичный ssh ключ добавлен в профиль на гитхабе:

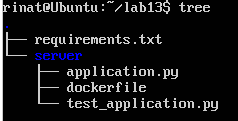




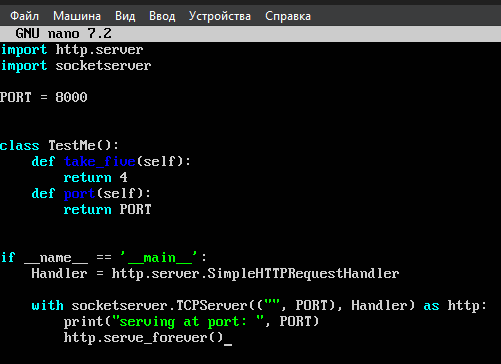
Создаем репозиторий и клонируем его:



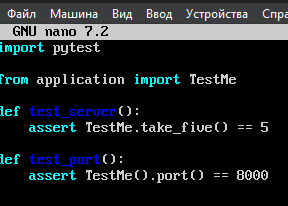
Подготавливаем структуру каталогов:



server/applcation.py



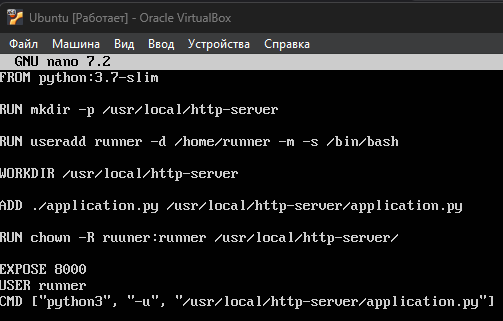
server/test\_applcation.py



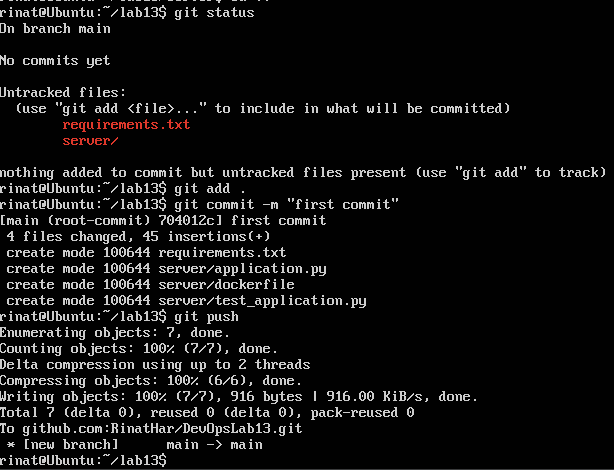
requirements.txt



dockerfile



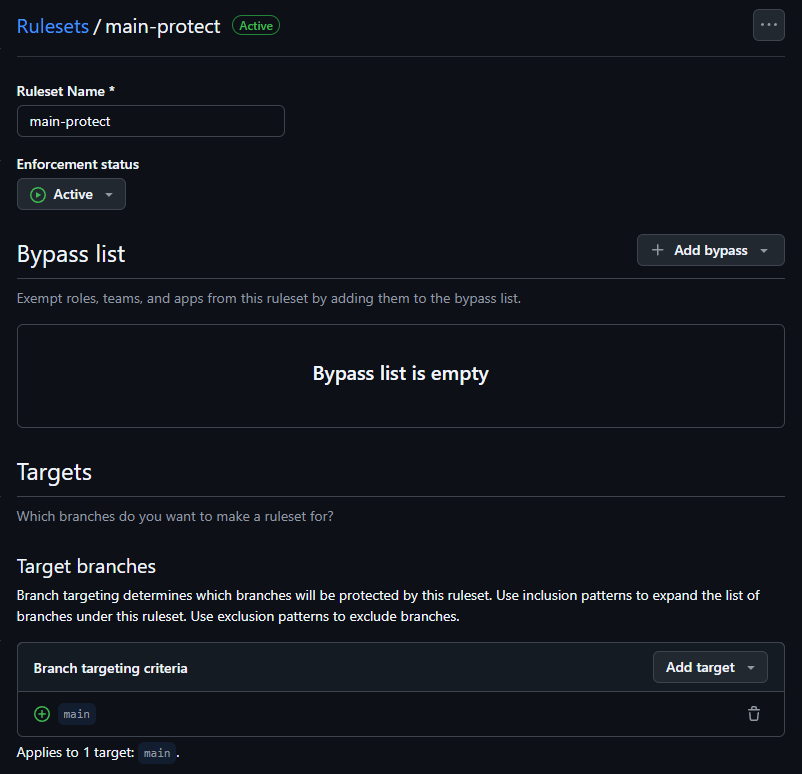
Загружаем изменения в гит:

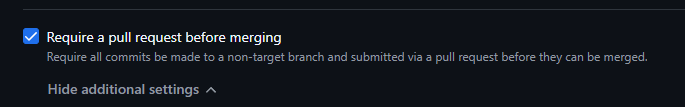


Переводим разработку в отдельную ветку dev:

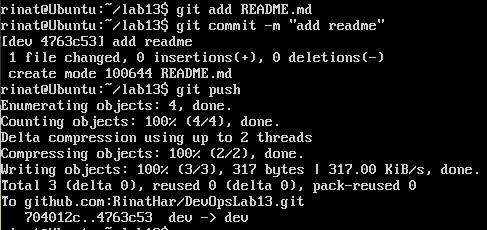


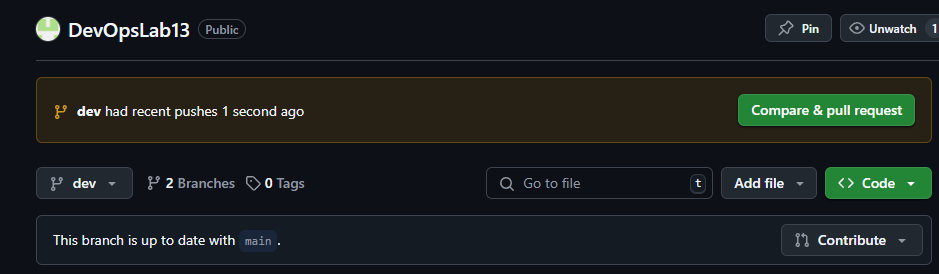
Защищаем ветку master от прямых изменений, теперь только merge-requests:

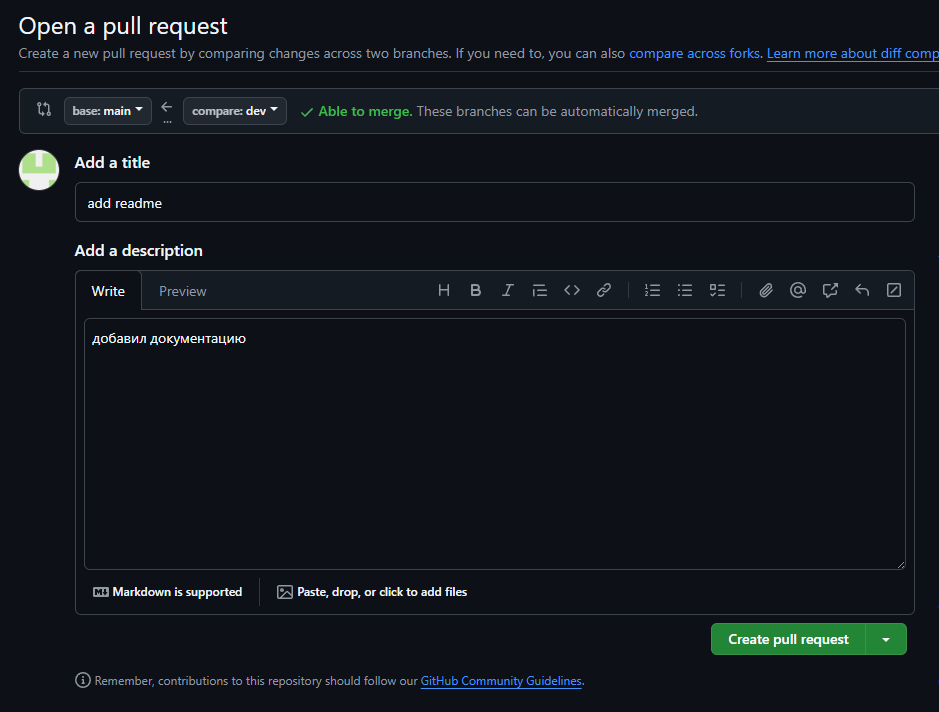


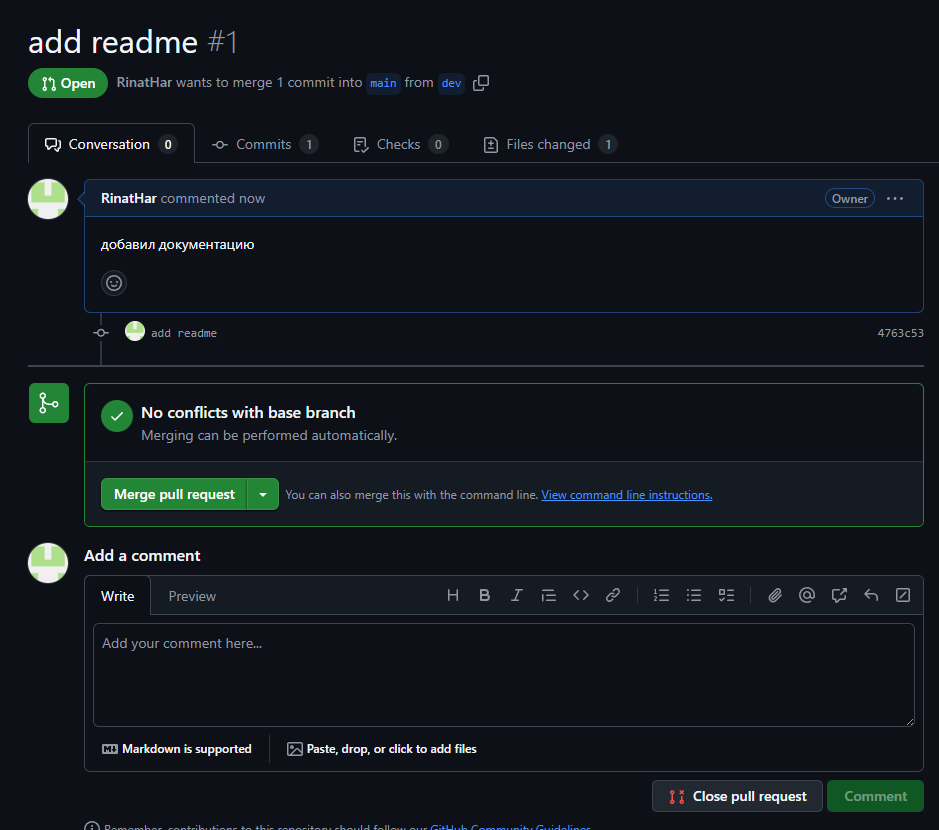


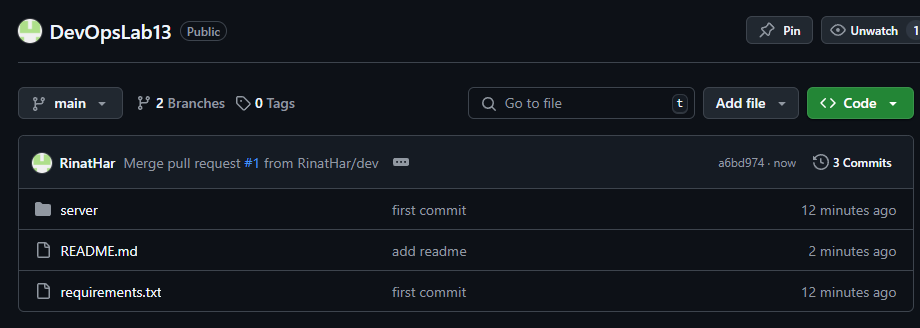
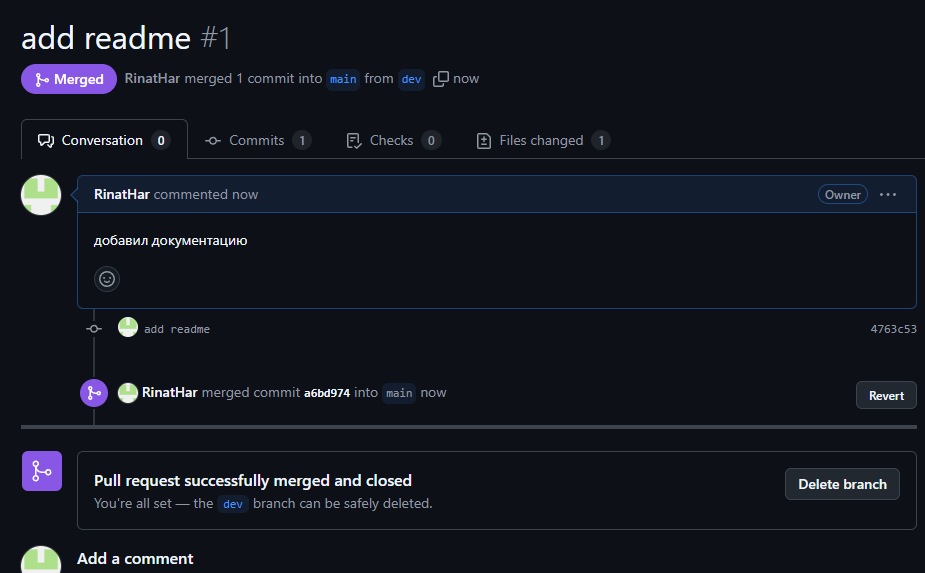
Добавляем README.md, пушим, открываем pull/merge-request:





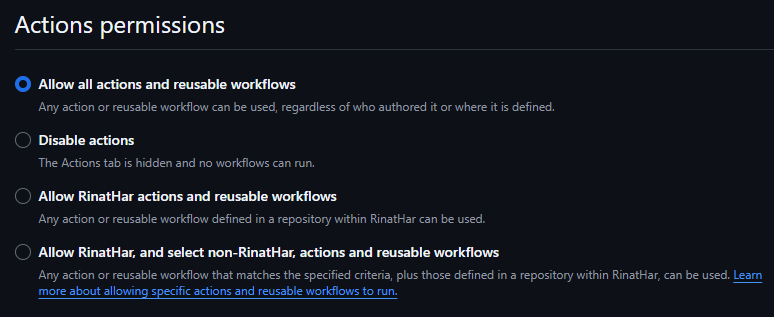




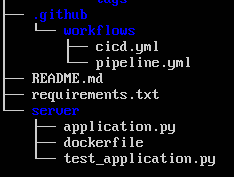


Добавляем CI-пайплайны (сценарии) – Github Actions

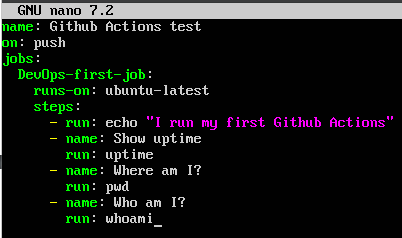
Проверяем права:



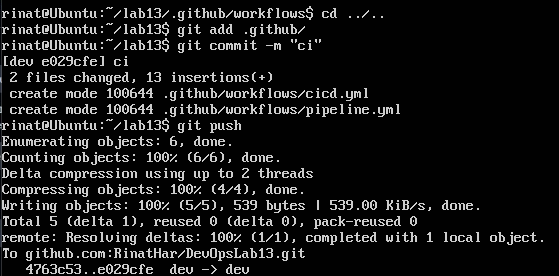
Подготавливаем структуру:

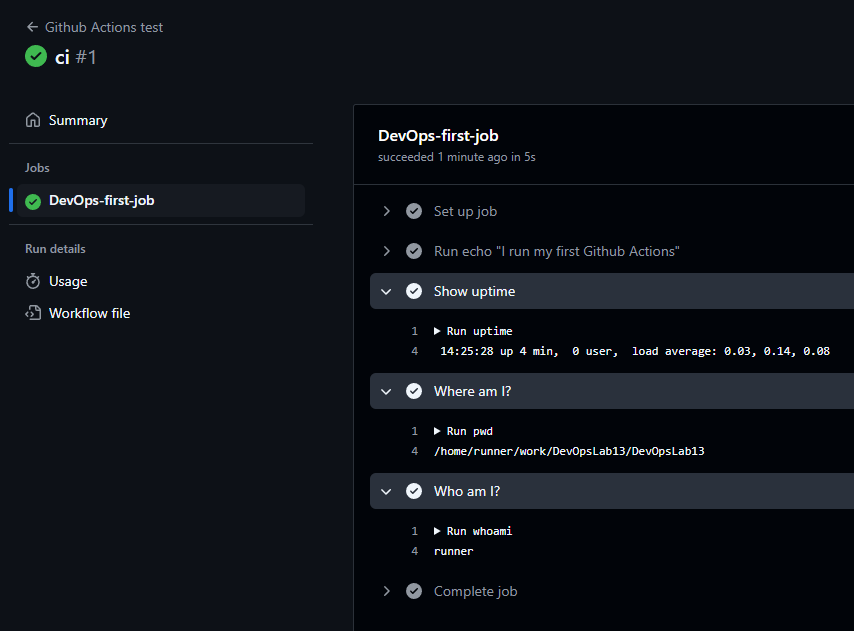


Пишем тестовый сценарий в pipeline.yml:

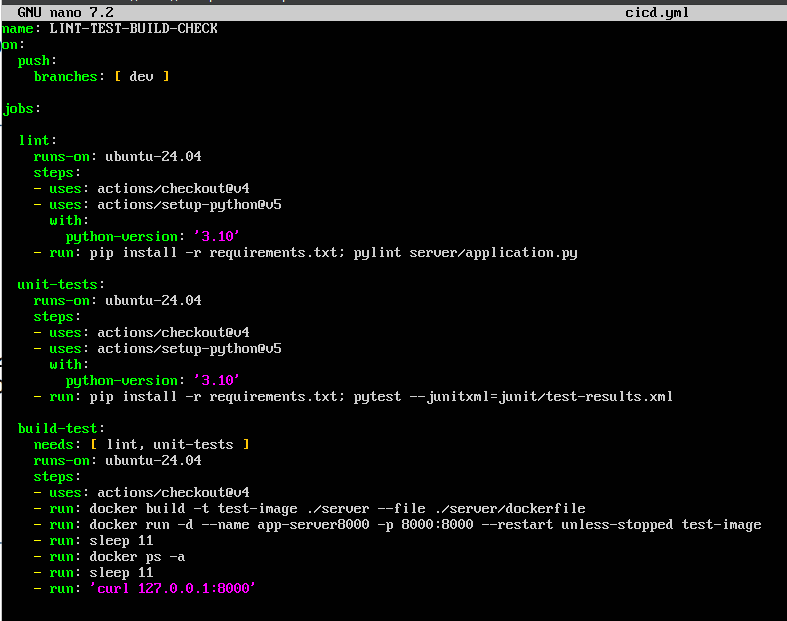


Добавляем сценарий в гит:

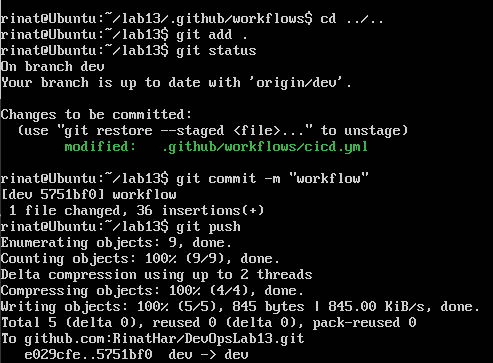




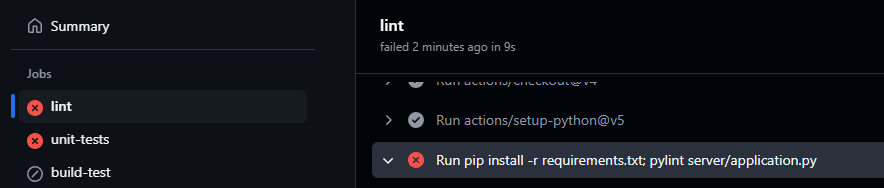
Пишем боевой сценарий:



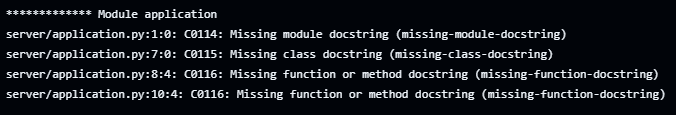
Добавляем сценарий в git:

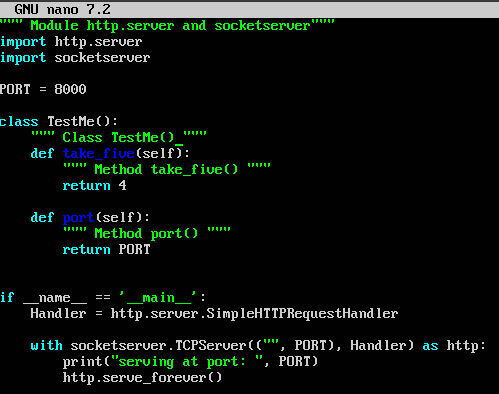


Видим, что есть ошибки:

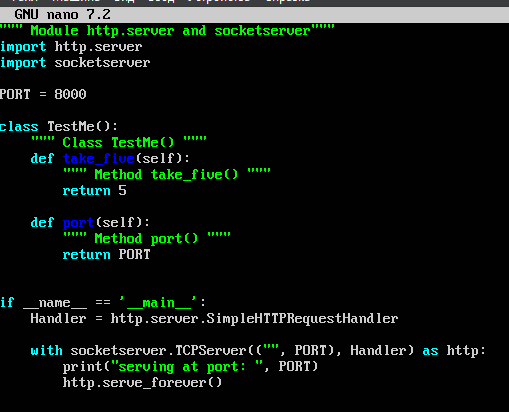


Чиним линтер:

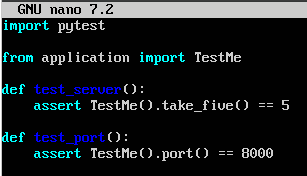




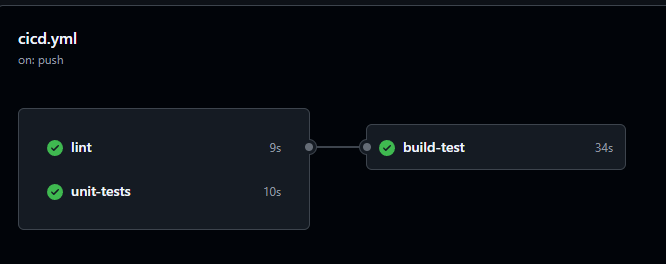
Изменяем 4 на 5 в application.py:

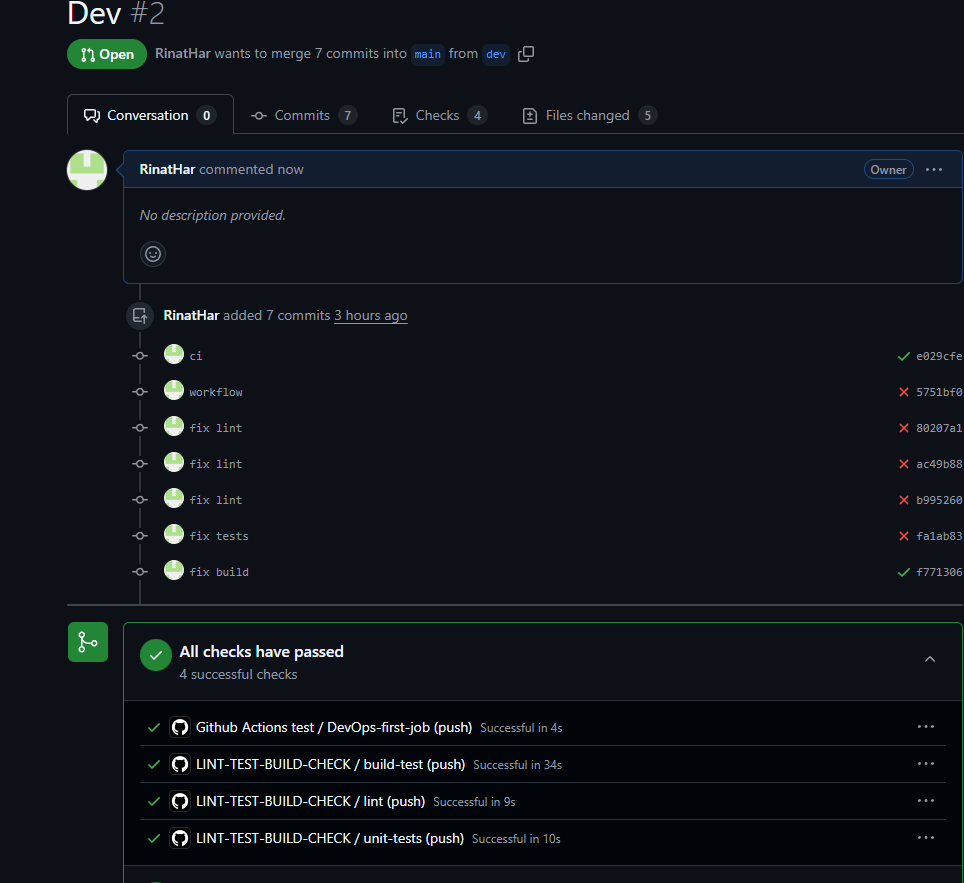


Изменяем TestMe на TestMe() в test\_application.py:



В итоге pipeline полностью выполнился:

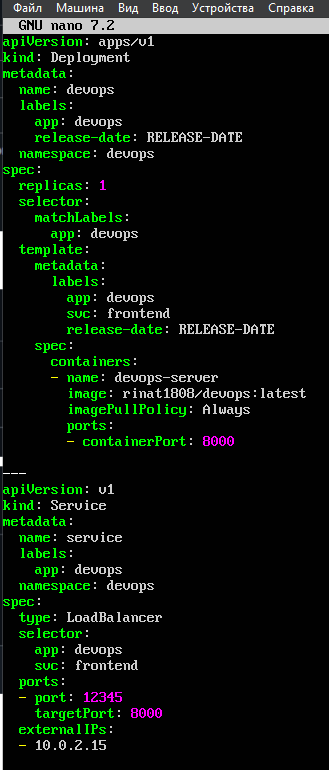




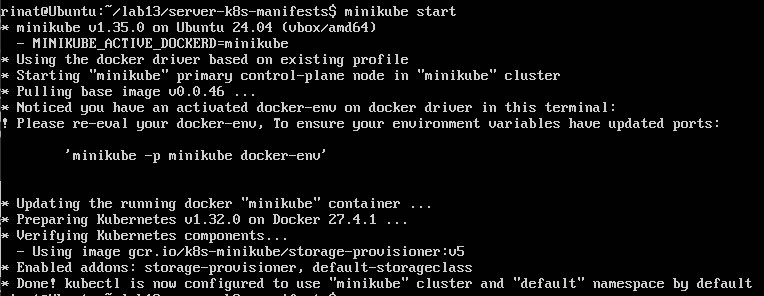
Создаём для приложения k8s-манифесты:



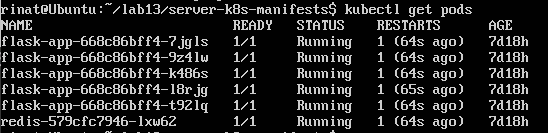
Создаем K8s манифест devops.yml:



Запускаем minikube:



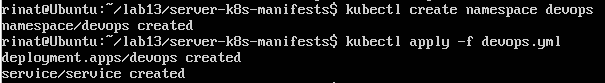
Удаляем старые контейнеры:







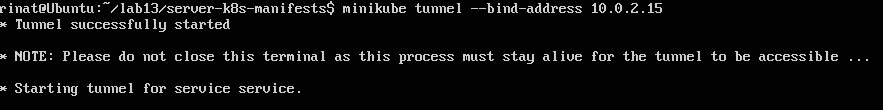
Применяем манифесты:



Собираем образ:

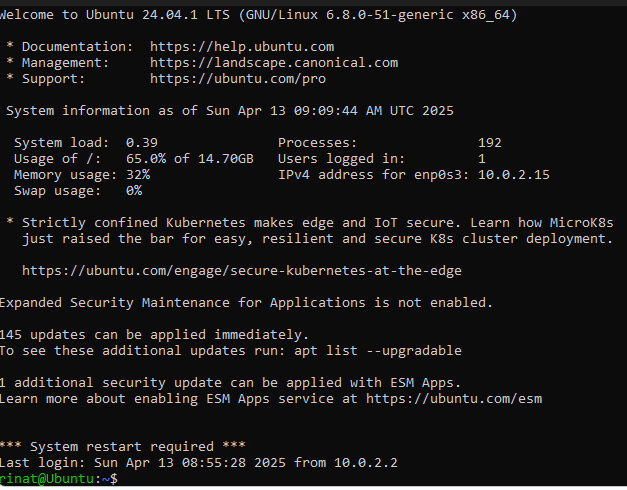




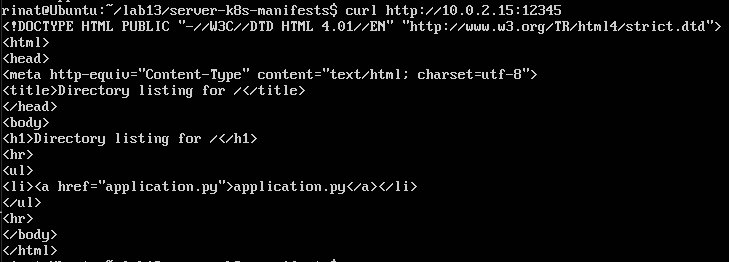
Открываем туннель:  


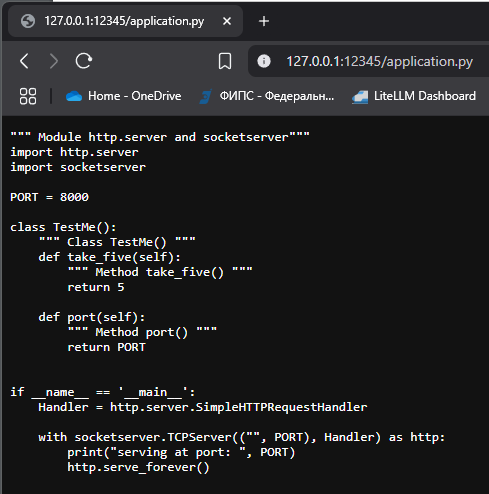
На хостовой машине подключаемся к виртуалке:





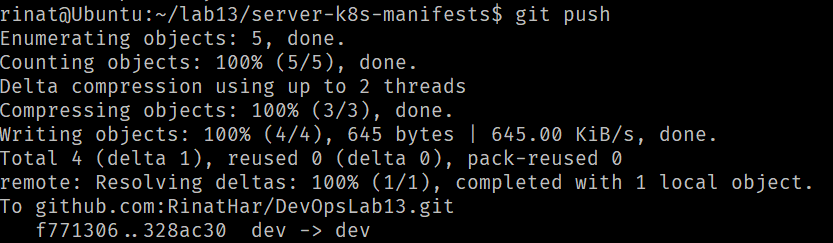
Проверяем работу:



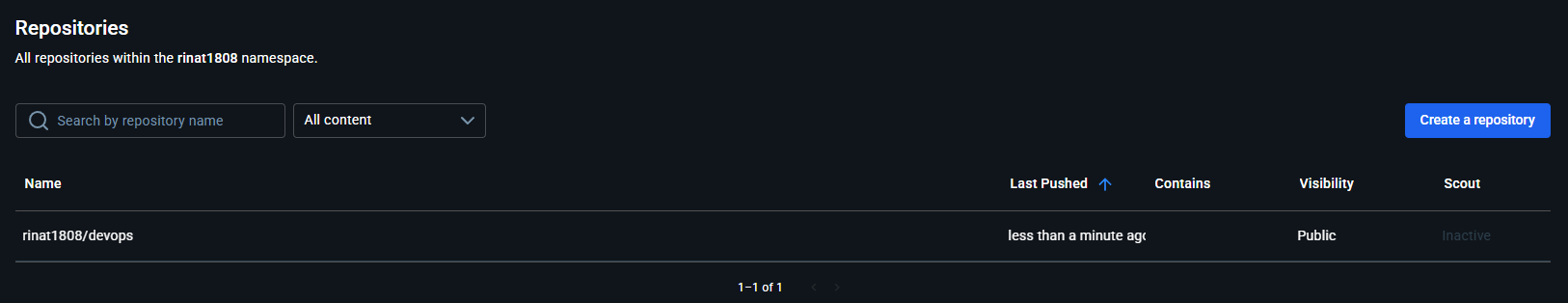


Сохраняем в github:

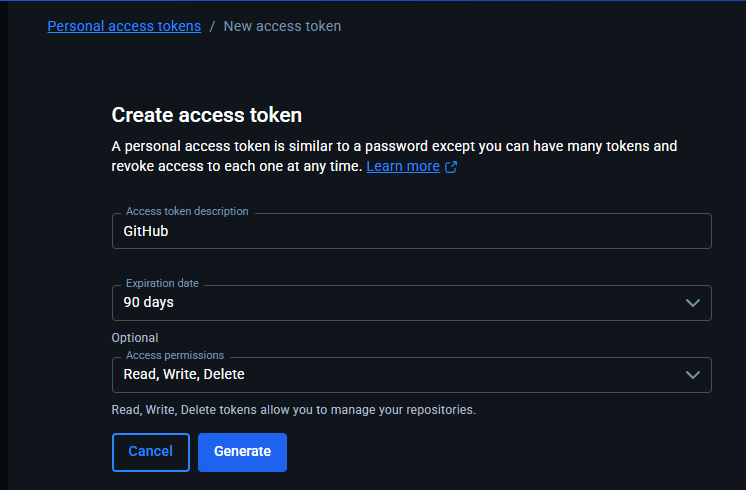




Создаем публичный репозиторий в DockerHub:

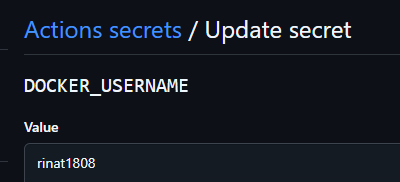


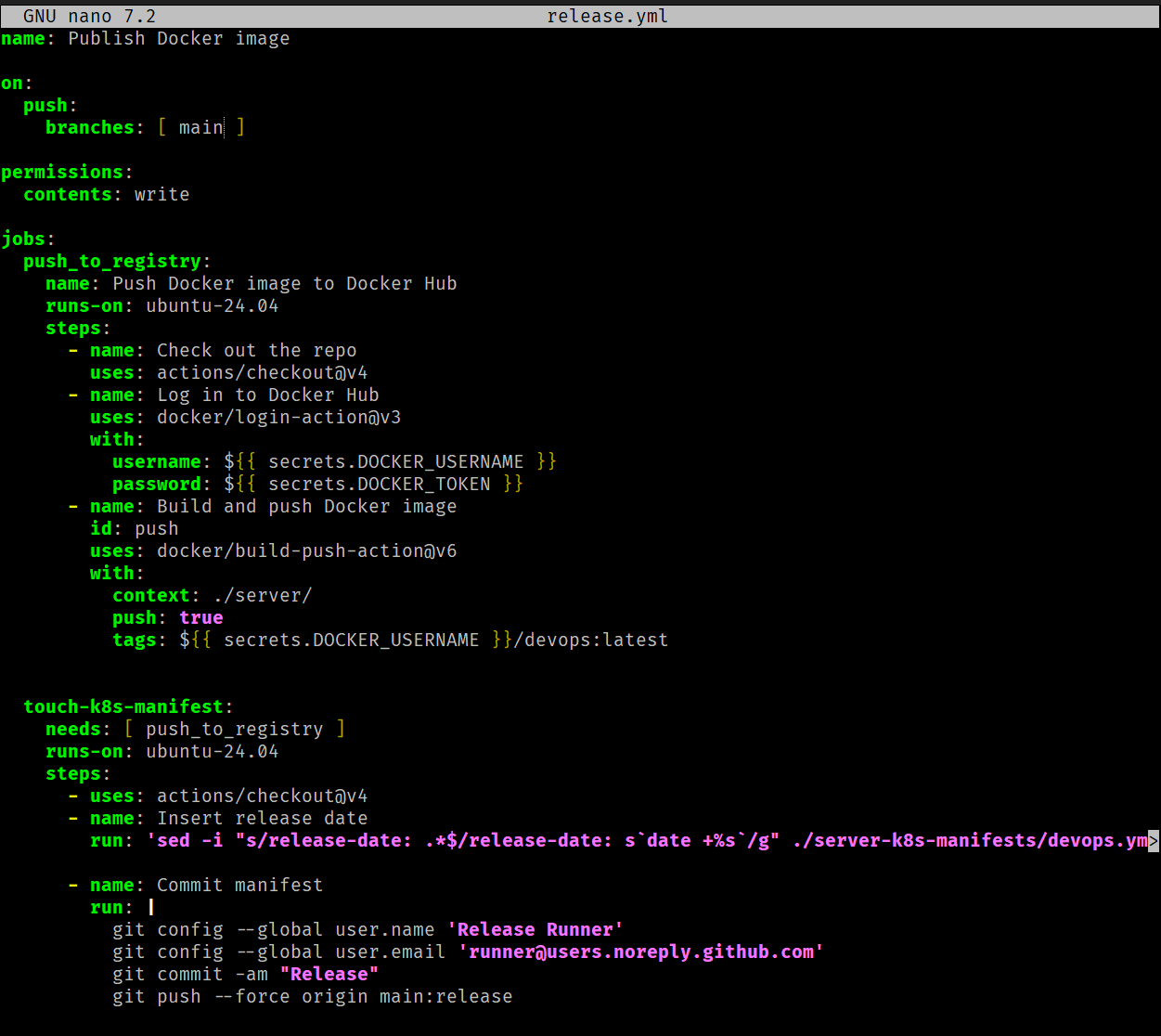
Создаем access токен:



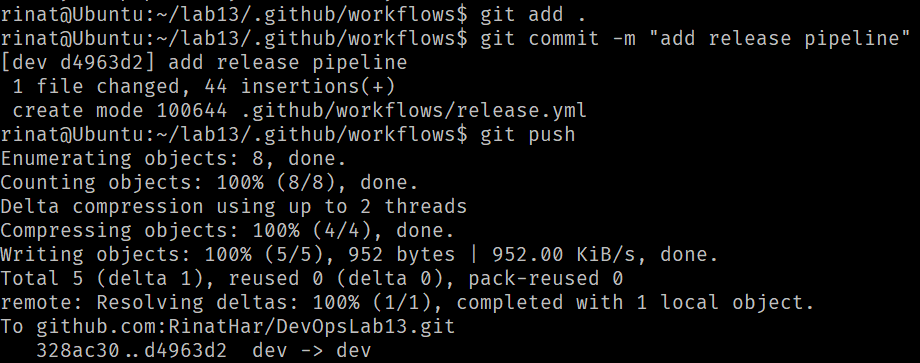
Создаем два секрета в github:



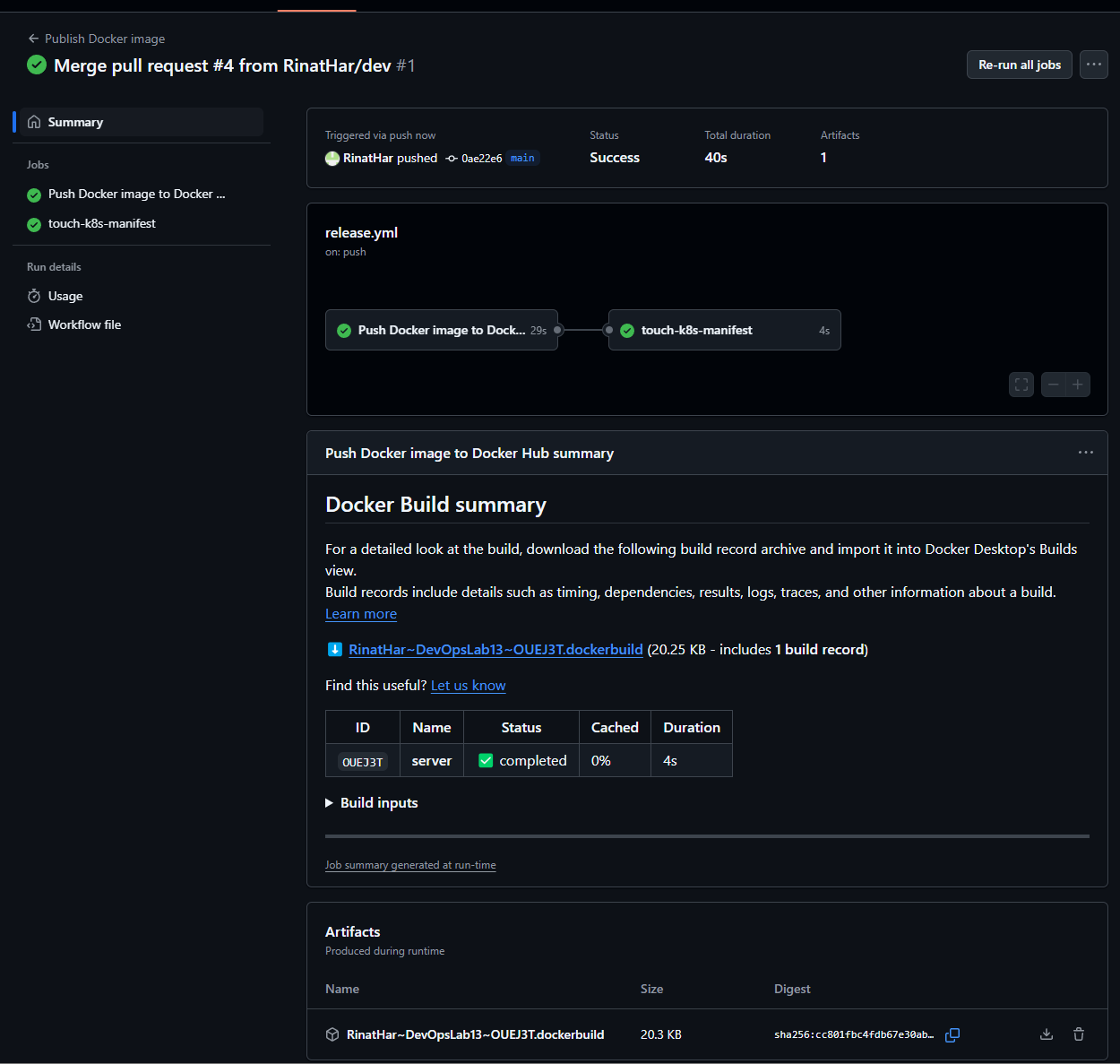


Создаем пайплайн для сборки и публикации образа в Docker Hub .github/workflows/release.yml:  


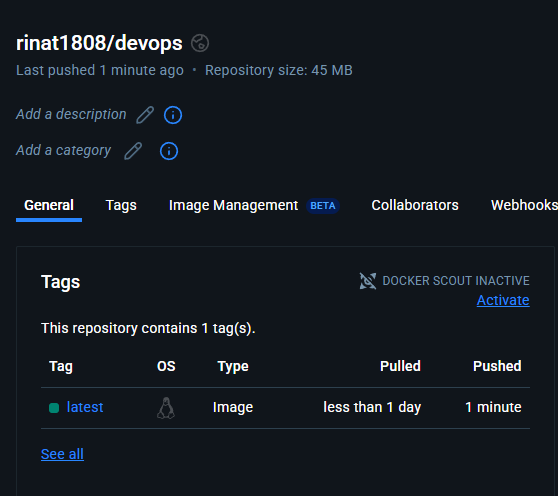
Заливаем в гит:



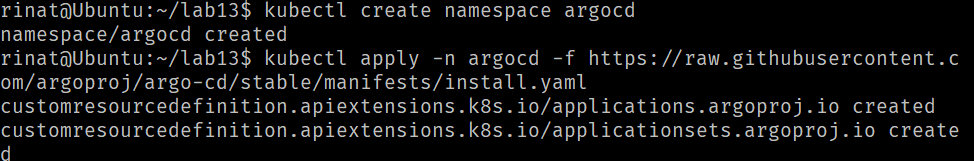
Смотрим отработку pipeline-а:

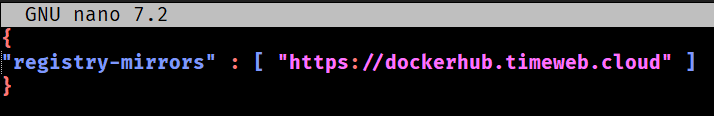


Видим, что образ успешно положен:

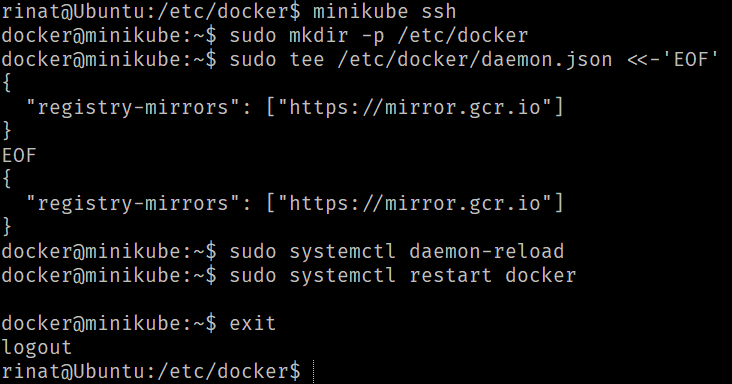


Устанавливаем ArgoCD:





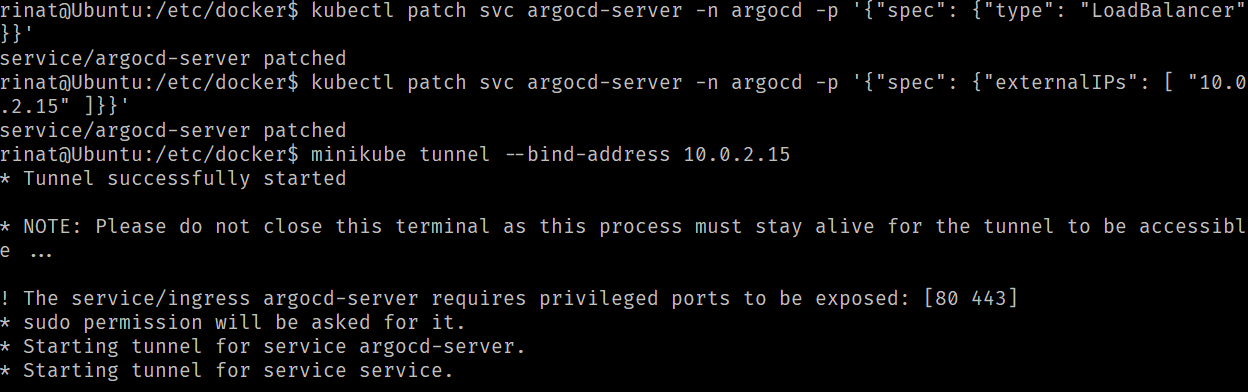




Получаем стартовый админский пароль:

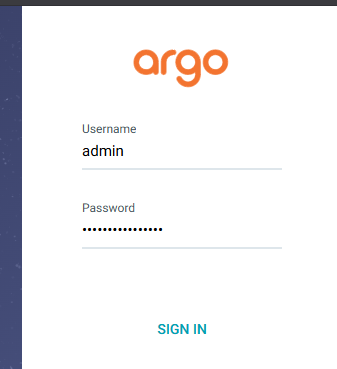


Пробрасываем api-сервис ArgoCD наружу:

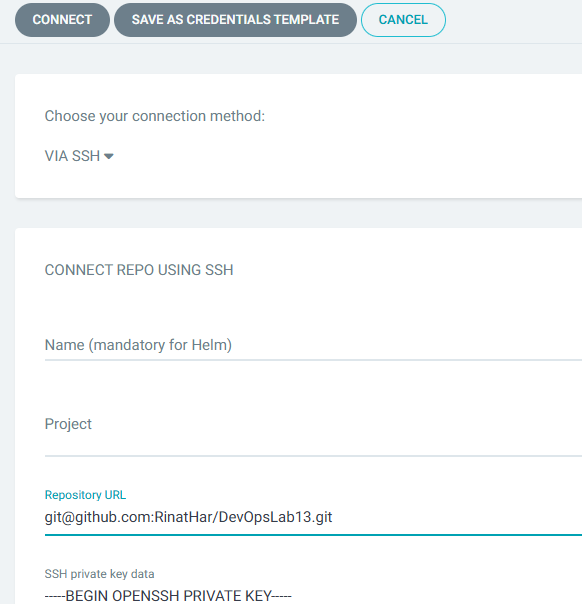


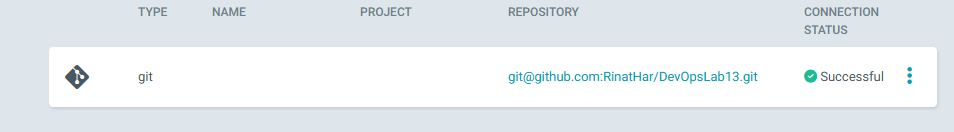


Входим, используя имя admin и пароль с предыдущего шага:

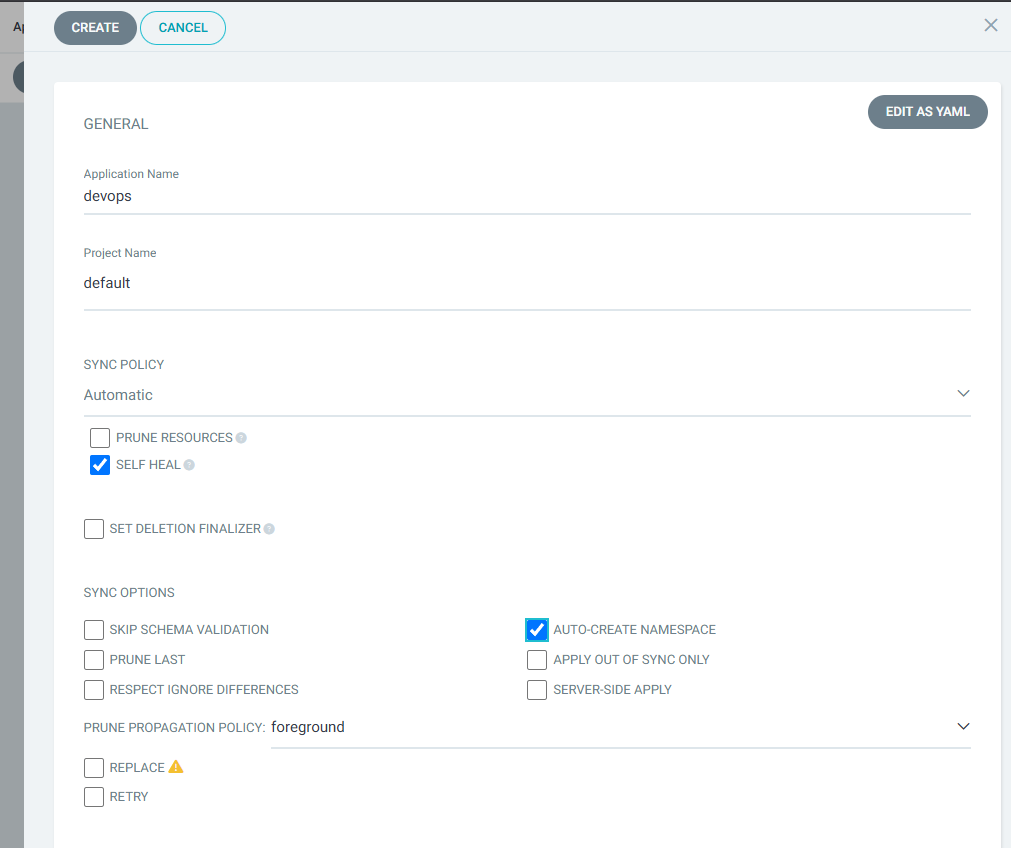


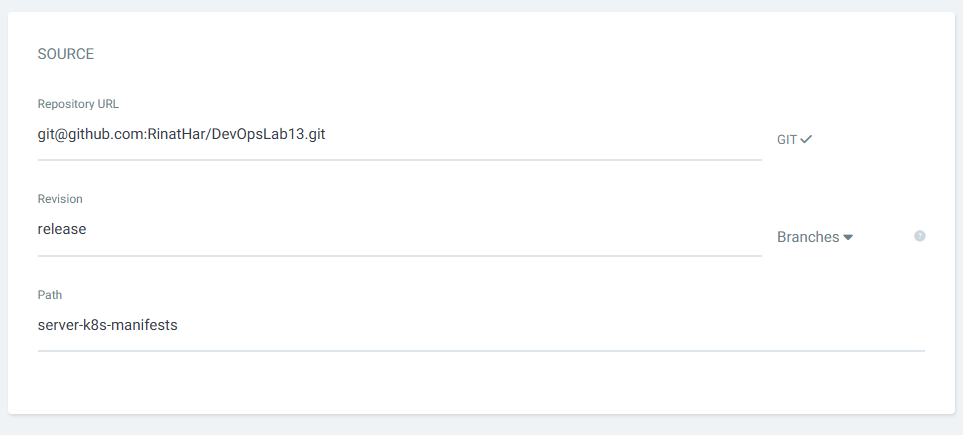
Подключаем к ArgoCD наш GitHub репо с использованием приватного ключа:

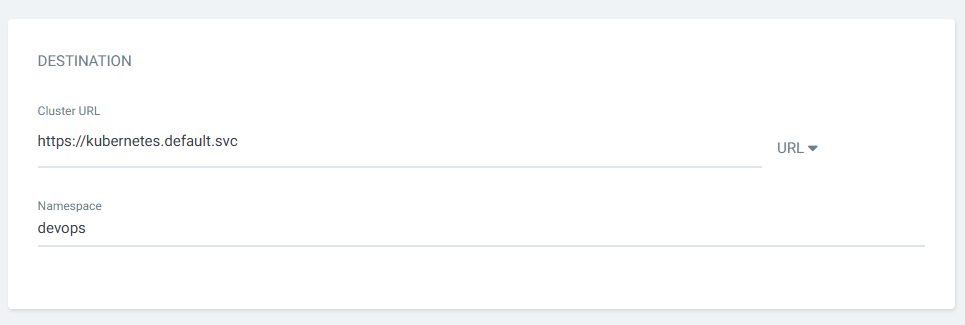




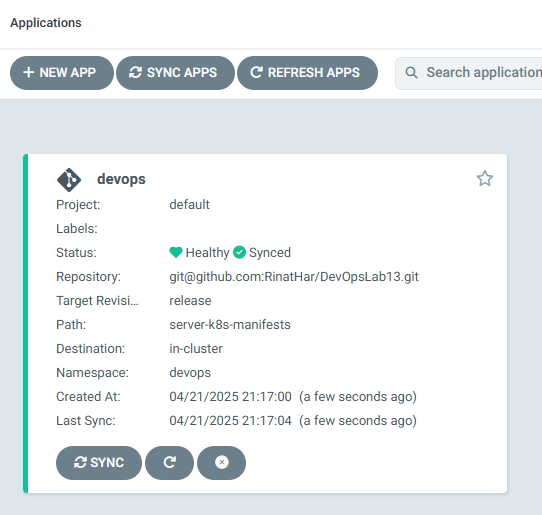
Подключаем манифесты приложения для контроля состояния:

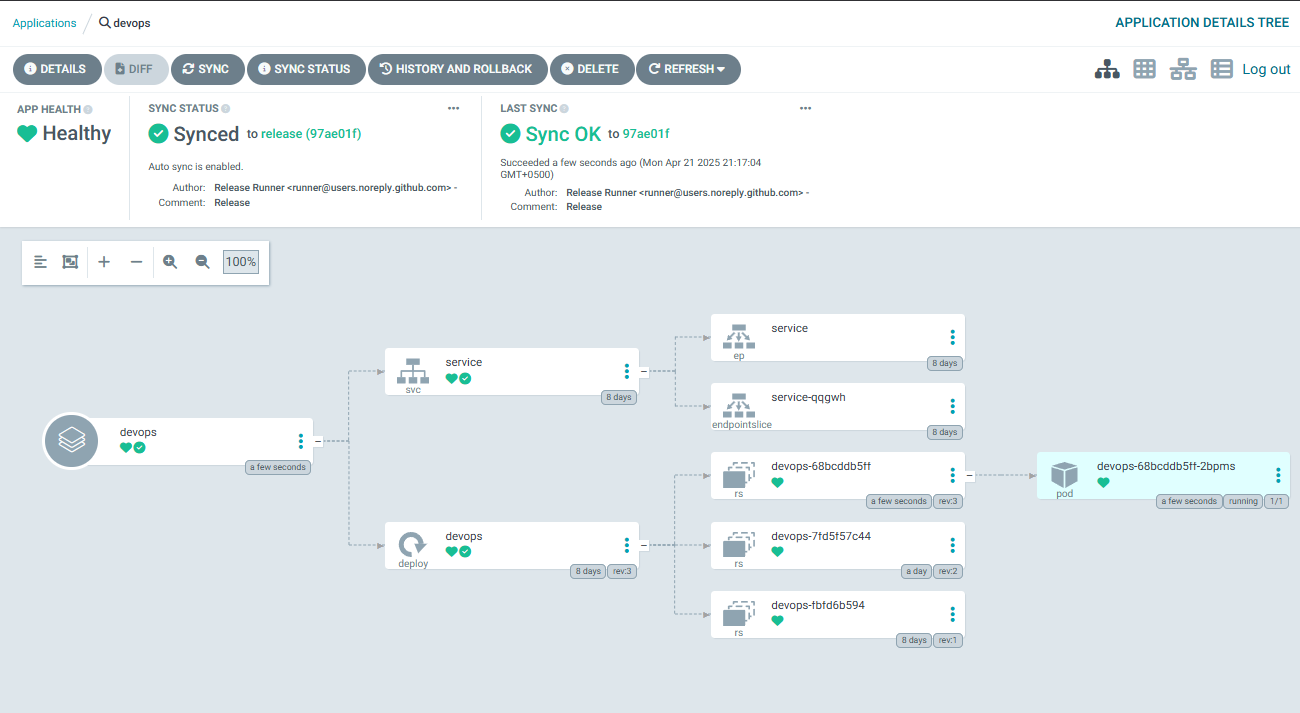






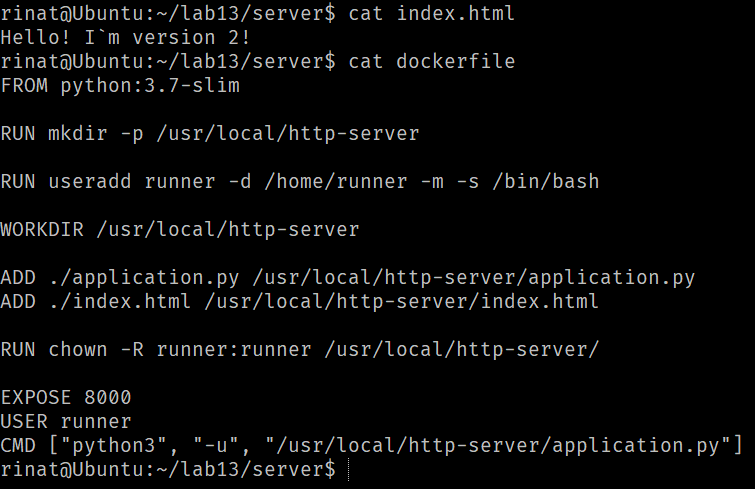
Всё работает успешно:



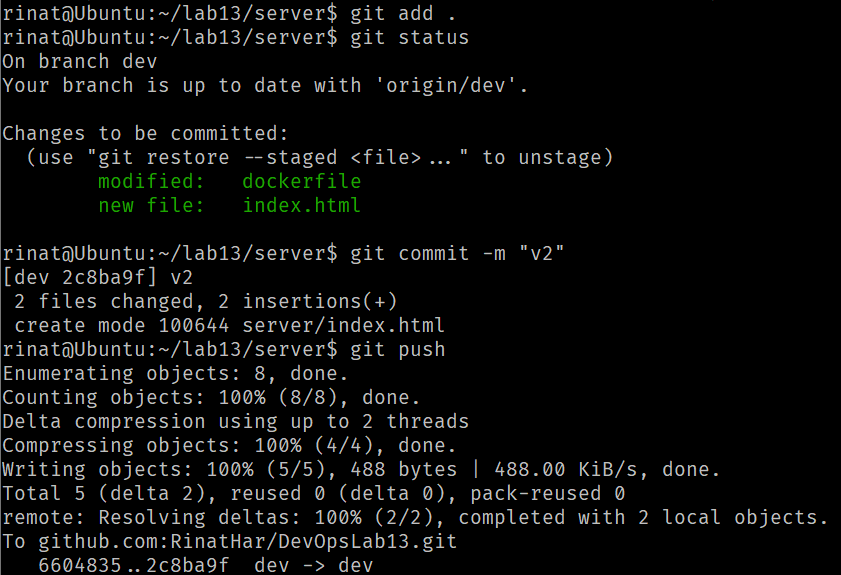


Проверка

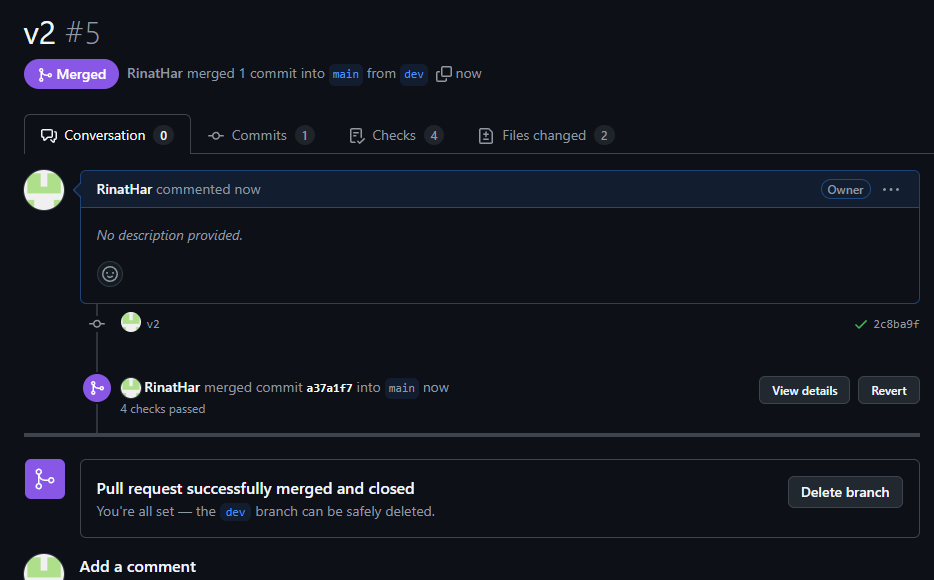
Добавляем и изменяем файлы:

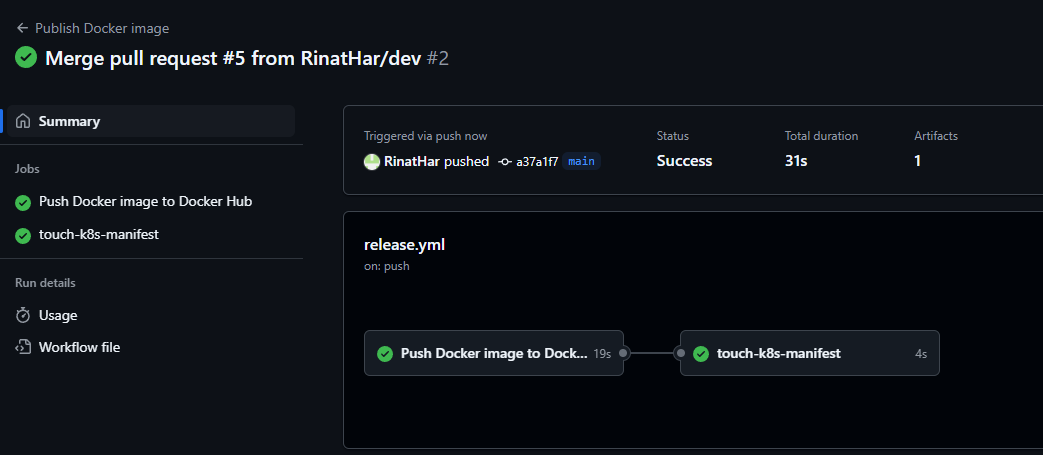


Отправляем в ветку dev:

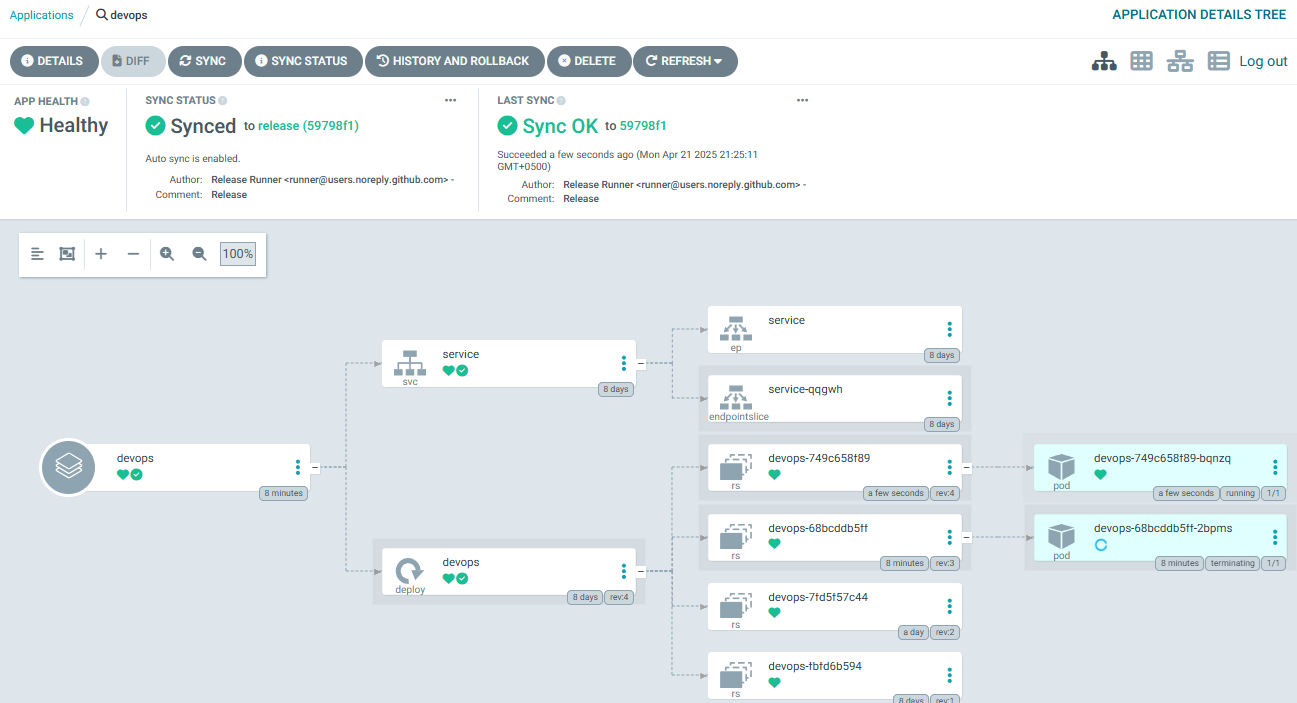


Создаем pull request и принимаем мердж:





ArgoCD синхронизирует всё:



Всё изменилось:

