Engineer

Побудувала пайплайни CI/CD на GitHub Actions

Побудувала пайплайни CI/CD на GitHub Actions з distroless-образом продакшн-фронтенду та просуванням релізів між кількома середовищами.

2025

один пайплайн, три середовища, distroless-образ

Ситуація. Випуск нових версій не повинен залежати від того, чи хтось пам'ятає всі кроки, а те, що зрештою працює в продакшні, не повинно бути товстим контейнером загального призначення з оболонкою та пакетним менеджером, якими він ніколи не скористається, — це просто поверхня атаки, яка існує без жодної причини.

Завдання. Зробити шлях від коміту до запуску в Azure автоматичним, а продакшн-образи — настільки малими й закритими, наскільки дозволяє кожне навантаження.

Дія. Автоматизація працює на GitHub Actions. Окремі робочі процеси відповідають за перевірку якості коду, тести й розгортання в кожне середовище, а поруч працюють CodeQL, dependency review та крок формування SBOM, тож ніщо не потрапляє в середовище, не пройшовши спершу перевірок. Образи збираються в кілька стадій, а базовий образ для кожної частини обрано з огляду на її потреби, а не одним рішенням для всіх: фронтенд виходить на distroless-образі (gcr.io/distroless/cc-debian13 — без оболонки, без пакетного менеджера), Go API — на легкому Alpine, а образ бази даних — на postgres-slim. Збірка просувається середовищами за визначеним маршрутом, а не вручну.

Результат. Релізи перестали бути обережним ручним ритуалом і стали рутинною, нудною подією, а саме цього від релізів і хочеться. Фронтенд у продакшні працює майже на мінімумі того, що йому взагалі можна дати,

перевірки ловлять проблеми до того, як ті потраплять у продакшн, а «розгортання» — це те, що роблять робочі процеси, а не те, через що комусь доводиться нервувати.

Застосуйте цей досвід у своєму проєкті

Використайте «Побудувала пайплайни CI/CD на GitHub Actions» як відправну точку для схожого завдання у вашій організації. Послуги, що стоять за цією роботою, доступні й для вашого проєкту. Перевірено, випущено і з можливістю відкату, тож зміни доходять до клієнтів у вівторок удень, а не пізно в п'ятницю.

Переглянути наші послуги Зв'язатися з нами →

Дізнайтеся, що привело до результату

Прочитайте про ситуацію, рішення та результат цього прикладу. Ми можемо обговорити, що відповідає вашим умовам, а що потребує іншого підходу. Побудувала пайплайни CI/CD на GitHub Actions — один пайплайн, три середовища, distroless-образ

Переглянути наші послуги Зв'язатися з нами →

Потрібен повноцінний CI/CD-пайплайн?

Обговоріть свій проєкт з інженером, відповідальним за виконання. Обсяг, основу розрахунку вартості та передачу результатів узгоджуємо письмово до початку робіт.

Роль у цій роботі: DevOps-інженер. Тут задіяні DevOps, Хмарні технології, Контейнери (Docker/Kubernetes), Автоматизація та CI/CD, Надійність і резервне копіювання, Безпека та Тестування та QA. І, до речі, ми добре вміємо DevOps та автоматизація CI/CD та Контейнеризація та оркестрація.

Зв'язатися з нами Ознайомтеся з нашими послугами

Одна частина **довшого переліку** — і він увесь на цьому сайті.

Кожен запис написано однаково: ситуація, завдання, що ми зробили і що змінилося.

Переглянути весь перелік

Визначте потрібне покращення

Опишіть шлях розгортання, експлуатаційні ризики та чергову команду. Ми можемо визначити зміну з перевірками, шляхом відновлення та чіткою відповідальністю.

Дивитися, скільки це коштує Зв'язатися з нами →

Потрібен повноцінний CI/CD-пайплайн?

Запитайте про рішення в роботі «Побудувала пайплайни CI/CD на GitHub Actions». Ви говоритимете з людьми, які можуть пояснити роботу й оцінити ваш проєкт.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/portfolio/built-github-actions-ci-cd-pipelines/