Інформаційні технології

УДК 004.7

Бойко Павло Васильович

студент

Національного технічного університету України

«Київський політехнічний інститут імені Ігоря Сікорського»

Бойко Павел Васильевич

студент

Национального технического университета Украины

«Киевский политехнический институт имени Игоря Сикорского»

Boiko Pavlo

Student of the

National Technical University of Ukraine

"Igor Sikorsky Kyiv Polytechnic Institute"

WEB-ДОДАТОК ДЛЯ СПРОЩЕННЯ СТВОРЕННЯ ФАЙЛІВ КОНФІГУРАЦІЇ DOCKER-COMPOSE ТА KUBERNETES
WEB-ПРИЛОЖЕНИЕ ДЛЯ УПРОЩЕНИЯ СОЗДАНИЯ ФАЙЛОВ КОНФИГУРИРОВАНИЯ DOCKER-COMPOSE И KUBERNETES
WEB-APPLICATION TO MAKE IT EASIER TO CREATE CONFIGURATION FILE FOR DOCKER-COMPOSE AND KUBERNETES

Анотація. *У дослідженні здійснено аналіз docker-compose та Kubernetes як інструментів для впровадження CI/CD(Continious Integration - безперервна інтеграція/Continous Delivery - безперервний випуск). Також запропонований варіант спрощення роботи з цими інструментами, з прикладами*

Ключові слова: *CI, CD, безперервна інтеграція, безперервний випуск, docker-compose, Kubernetes.*

Аннотация. *В исследовании проведен анализ docker-compose и Kubernetes как инструментов для интеграции CI/CD(Continious Integration - беспрерывная интеграция/Continous Delivery - беспрерывный выпуск). Также предложен вариант упрощения работы с этими инструментами, с примерами*

Ключевые слова: *CI, CD, беспрерывная интеграция, беспрерывный выпуск, docker-compose, Kubernetes.*

Summary. *The study analyzed docker-compose and Kubernetes as tools for implementing CI / CD (Continious Integration/Continuous Delivery). Also offered is an option to simplify work with these tools, with examples*

Key words: *CI, CD, Continious Integration, Continuous Delivery, docker-compose, Kubernetes.*

Постановка проблеми. Інструменти docker-compose та Kubernetes набирають все більшої популярності для впровадження Continious Integration та Continuous Delivery. Так як створення файлів конфігурації - це значна частина роботи, треба вирішити проблема витрат часу та людських ресурсів.

Аналіз останніх досліджень і публікацій. Щодо створення програми для генерації файлів конфігурацій публікації та дослідження відсутні.

Метою даної роботи є попереднє ознайомлення с програмою, яку зможуть використовувати, як невеликі, так і великі компанії.

Виклад основного матеріалу. Спеціалісти у сфері розробки програмного забезпечення (ПЗ) намагаються уникати великої кількості ручної роботи, такої як тестування, підготовка до розповсюдження

продукту, розповсюдження продукту, інформування розробників, менеджерів та інженерів з забезпечення якості про наявність нової версії додатку. Для досягнення даної мети використовується підхід до розробки, в основі якого лежить виконання регулярного автоматичного збирання проекту з використанням раніше створених сценаріїв. Це дозволяє запобігати утворенню великої кількості помилок на різних етапах розробки програмного засобу та автоматизувати все те, що раніше виконувалося вручну. Даний процес називається неперервною інтеграцією (англ. Continuous Integration, CI) та застосовується для контролювання ризиків, зменшення часових, грошових та інших витрат у процесі розробки ПЗ. Однією з ключових переваг застосування такого підходу є можливість перевірки вихідного коду продукту за допомогою тестів після кожної фіксації змін, що дозволяє швидко виявляти помилки на будь-якому кроці розробки та зменшити витрати на їх усунення.

Також розглядаючи дане питання, нас цікавить Неперервна інтеграція (англ. Continious Delivery) – це підхід до розробки програмного забезпечення при якому продукт може бути випущений у виробництво у будь-який момент часу. Для його впровадження використовується спеціальне програмне забезпечення, базовим принципом роботи якого є моніторинг змін у репозиторії і запуск відповідних сценаріїв і задач.

Docker набуває чималої популярності у всьому світі за його узгодженість у різних середовищах. Завжди існують незначні відмінності між середовищами розробки та випуску, якщо у вас немає власного приватного середовища сховищ з жорсткими перевірками на місці. Ці відмінності можуть бути пов'язані з різними версіями пакетів або залежностями. Тим не менш, Docker може вирішити цей розрив, забезпечивши послідовне середовище від розробки до випуску. Контейнери Docker налаштовані на підтримку всіх конфігурацій та залежностей всередині. У результаті ви можете використовувати той же

контейнер від розробки до виробництва, переконавшись у відсутності розбіжностей або втручання вручну.

З контейнерами Docker ви також можете гарантувати, що розробникам не потрібна однакова виробнича середа. Замість цього вони можуть використовувати власну систему для запуску контейнерів Docker на VirtualBox. Перевага Docker полягає в тому, що ви можете запустити такий самий контейнер на екземплярах Amazon EC2. Якщо вам потрібно виконати оновлення протягом циклу випуску продукту, ви можете легко внести необхідні зміни в контейнери Docker, протестувати їх та застосувати ті самі зміни для існуючих контейнерів. Така гнучкість є ключовою перевагою використання Docker. Так само, як стандартні процеси розгортання та інтеграції, Docker дозволяє створювати, тестувати та випускати images (шаблони для запуску контейнерів), які можуть бути розгорнуті на декількох серверах. Навіть якщо доступний новий патч безпеки, процес залишається незмінним. Ви можете застосувати цей патч, протестувати його та віддати для релізу.

Так як для роботи з docker-compose або kubernetes потрібно створювати файл конфігурації, часом об'ємний та з великою вірогідністю помилок. Основна задача цього файлу - це дати зрозуміти даним системам які контейнери і з якою конфігурацією запускати.

На основі популярності даних технологій та відсутності додатків зі схожим функціоналом, виникла ідея створити такий.

Початкова версія, в якій реалізовано створення тільки файлів конфігурації для docker-compose виглядає наступним чином:

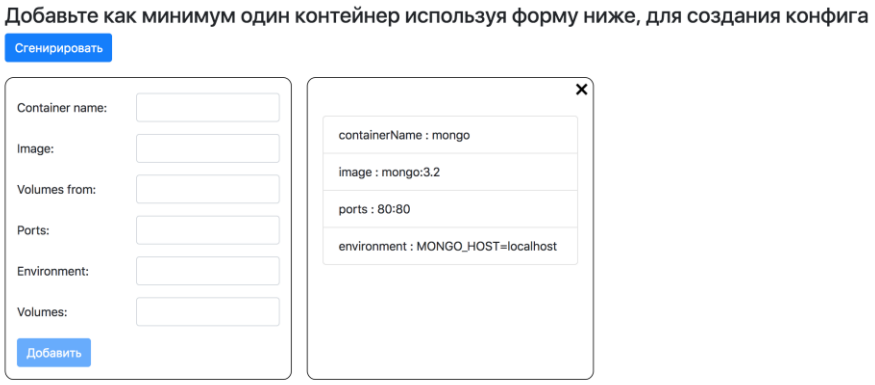

Рис. 1. Приклад роботи програми

Коли всі необхідні контейнери будуть додані, натискаємо "Сгенерировать" і отримуємо готовий файл для запуску нашої системи. Також в планах є додати підтримку Kubernetes, додаткових полів, та можливість запуску.

Висновки з даного дослідження. Docker-compose та Kebernetes є потужними інструментами для впровадження Continious Integration та Continuous Delivery (Безперервна інтеграція та Безперервний випуск). Виходячи з цього має вийте гарний продукт для полегшення та прискорення роботи з даними інструментами.

Література

1. Неперервна інтеграція [Електронний ресурс] — Режим доступу : https://uk.wikipedia.org/wiki/Неперервна_інтеграція.
2. Duvall P., Matyas S., Glover A. Continuous Integration. Improving Software Quality and Reducing Risk, 2007. — 17-164
3. Humble J., Farley D. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, 2011. — 86-112 .
4. Docker, офіційна сторінка [Електронний ресурс] — Режим доступу: https://www.docker.com

5. CI/CD за допомогою docker та gitlabci [Електронний ресурс] — Режим доступу : https://medium.com/@Empanado/simple-continuous-deployment-with-docker-compose-docker-machine-and-gitlab-ci-9047765322e1
6. 4 причини почати використовувати Kubernetis [Електронний ресурс] — Режим доступу: https://www.infoworld.com/article/3173266/containers/4-reasons-you-should-use-kubernetes.html