

ОСОБЛИВОСТІ РОЗГОРТАННЯ МІКРОСЕРВІСІВ В KUBERNETES КЛАСТЕРІ

Пироженко В. С.

Науковий керівник – к.т.н., доц. Кудрявцева М. С.

Харківський національний університет радіоелектроніки, каф. ІУС
м. Харків, Україна

E-mail vadym.pyrozhenko@nure.ua

This work is devoted to the study methods and practices of continuous integration and continuous delivery of applications to the Kubernetes cluster. Also, the current work investigated methods that collect microservices and applications in packages for the next installation with different parameters and in several environments.

Більшість компаній, починаючи від дрібних стартапів і до великих корпорацій, під час своєї діяльності користуються специфічним та вузькоспеціалізованим програмним забезпеченням. У зв'язку з цим, часто в організаціях присутні розробники, які відповідають за підтримку роботи програм, їх модернізацію та вдосконалення. Таким чином, з'являється необхідність впровадження безперервної інтеграції (Continuous Integration, CI) і безперервного постачання (Continuous Delivery, CD). CI/CD – це одна з DevOps-практик, яка також відноситься і до agile-практик.

Безперервна інтеграція – це методологія розробки та набір практик, за яких код вносяться невеликі зміни з частими фіксаціями (commit). Безперервне постачання починається там, де закінчується безперервна інтеграція. Це дозволяє автоматизувати розгортання додатків у різні оточення.

Актуальність теми дослідження обумовлена тим, що для проведення кожного випробування роботи програми розробник повинен вручну проводити рутинні дії зі збирання, тестування та розгортання. Все це займає багато часу, потребує уважності людини та певної кваліфікації.

Для вирішення таких задач необхідно дослідити наявні практики, методи та інструменти, що дозволяють вирішити проблему неперервної інтеграції та безперервного постачання коду для додатків, що являють собою набір сервісів.

Об'єктом дослідження є поточний процес реалізації безперервної інтеграції та безперервного постачання коду. Предметом дослідження є практики автоматизації та їх застосування в CI/CD.

Головною метою роботи є дослідження додатків та практик, що дозволяють реалізацію автоматизації безперервної інтеграції та безперервного постачання великої кількості сервісів в кластер.

Аналіз CI/CD-інструментів показав, що Jenkins має велику кількість корисних плагінів та велику кількість гарних відгуків спільноти. Крім

CI/CD-інструментів, ще проаналізований менеджер пакетів helm, що дозволяє збирати сервіси в архів, який досить просто можна інсталиувати в кластер та helmfile для організації доставлення додатку в безліч Kubernetes (k8s) оточень [1].

Галузь застосування CI/CD практики – розробка програмного забезпечення. Для того, щоб успішно реалізувати збирання сервісів в архів, який називається helm chart, необхідно ще дослідити один із підходів, що називається Umbrella Chart. Такий підхід об'єднує в собі чарти з сервісами. Для описання релізів та оточення досліджено helmfile.

Umbrella Chart – це метод встановлення кількох компонентів як одного. Вони дозволяють налаштувати досить складні конфігурації, як кластер- стек Elastic шляхом встановлення лише одного звичайного чарту [2]. Усі компоненти, які містить Umbrella Chart, також є helm чартами – це фактично робить її чартами чартів. Ці залежності визначаються в окремому файлі у спеціальному розділі, що називається залежностями [2].

Крім створення чарту над іншими чартами, досліджено helmfile, що являється обгорткою для helm, яка дозволяє в одному місці описувати безліч helm релізів, параметризувати їх чарти для кількох оточень, а також задавати порядок їх постачання на k8s кластер [3].

Завдяки такому методу, збирається пакет, що містить в собі helm чарти сервісів, їх версії та параметри яких описані в Umbrella. Релізи та оточення дозволяє описувати helmfile, який можна інсталиувати у вигляді пакета. При вирішенні задачі створення пакета найбільшою перешкодою є складність такого рішення, що потребує значних зусиль у реалізації CI/CD процесу. Результатом виконання даної роботи є дослідження особливостей використання методу, створення пакетів, що можуть бути інстальовані в оточення автоматично, з різними параметрами в різні оточення.

Практичним результатом виконання даної роботи є побудова CI/CD, що збирає велику кількість сервісів в пакет.

Сфера розробки програмного забезпечення неймовірно зросла за останнє десятиліття завдяки зростанню кількості додатків та їх складності. Проте, підтримка сервісів стає все складнішою і потребує значних зусиль інженерів та часу і потребує постійної автоматизації.

Список використаних джерел:

1. Djangostars.com (2022, Dec 26). Continuous Integration: CircleCI vs Travis CI vs Jenkins vs Alternatives. <https://djangostars.com/blog/continuous-integration-circleci-vs-travisci-vs-jenkins/>
2. Mikemybytes.com (2020, Nov 25). Isolation issues with Helm umbrella charts. <https://mikemybytes.com/2020/11/25/isolation-issues-with-helm-umbrella-charts/>
3. Lyz-code.github.io (no date). The Blue Book: Helmfile. <https://lyz-code.github.io/blue-book/devops/helmfile/>