

АНАЛІЗ СУЧАСНИХ ТЕХНОЛОГІЙ ДЛЯ КЕРУВАННЯ КОНТЕЙНЕРИЗОВАНИМИ ДОДАТКАМИ З БАЛАНСУВАННЯМ ТРАФІКУ У ХМАРНОМУ СЕРЕДОВИЩІ

Вінницький національний технічний університет

Анотація

У роботі розглянуто сучасні інструменти для контейнеризації та керування програмами, такі як Kubernetes, Docker, що призначені для автоматизації процесу розгортання, масштабування та управління ресурсами. Показано актуальність балансування навантаження між контейнерами та розподілу навантаження в хмарному середовищі. Також наведено аналіз можливостей провідних постачальників хмарних послуг.

Ключові слова: Docker, Kubernetes, load balancer, cloud.

Abstract

The work presents modern tools for containerization and application management, such as Kubernetes, Docker, which are designed to automate the process of deployment, scaling, and resource management. The relevance of load balancing between containers and load distribution in the cloud environment is shown. A analysis of the capabilities of the leading cloud service providers is also provided.

Keywords: Docker, Kubernetes, load balancer, cloud.

Вступ

В індустрії програмного забезпечення дуже важливою темою є розгортання та керування контейнерами. Використання контейнеризації додатків дозволяє ефективно та швидко розгортати, масштабувати та керувати додатками на локальних серверах та в хмарних середовищах. Це сприяє більш ефективному розвитку та просуванню проектів, оскільки значно зменшує час та зусилля, необхідні для їх розгортання та керування. Однак, розгортання та керування контейнерами створює нові проблеми та виклики, такі як складність автоматизації процесів розгортання, масштабування та керування ресурсами, а також необхідність ефективного керування трафіком та навантаженням. Тому, для розробників та адміністраторів, які працюють з контейнерами, вивчення цих проблем та засобів їх вирішення є актуальним.

Результати досліджень

Docker і Kubernetes є інструментами, які використовуються для контейнеризації та керування програмами. Docker функціонує як платформа для створення, розгортання та керування контейнерами, тоді як Kubernetes є автоматизованим інструментом керування контейнерами. Kubernetes можна використовувати в поєднанні з Docker для створення та розгортання контейнерів у кластері Kubernetes [1, 2].

Використовуючи Docker, можна створювати та керувати контейнерами, кожен з яких містить власні залежності та середовище, що дозволяє додаткам функціонувати в ізольованому віртуальному середовищі [3]. Контейнери Docker розгортаються та керуються Kubernetes. Kubernetes забезпечує автоматичне розгортання, масштабування та керування контейнерами Docker у кластері Kubernetes. Кожен контейнер можна розгорнути на окремому вузлі в кластері, що забезпечує більшу стабільність і доступність для програм. Крім того, Kubernetes може забезпечити доступність і балансування навантаження на кластері. Для цього він може створити кілька копій кожного контейнера та розподілити робоче навантаження між ними. При зростанні навантаження, Kubernetes може автоматично масштабувати контейнери. Він також забезпечує конфігурування та

керування версіями для контейнерів.

На початку розвитку комп'ютерних систем та мереж балансування навантаження досягалося шляхом ручного розподілу навантажень між декількома серверами. Однак, зі збільшенням кількості серверів ставало дедалі важче керувати розподілом навантажень вручну, що призвело до розробки перших програмних балансувальників навантаження, призначених для автоматичного розподілу трафіку між кількома серверами.

Сучасні балансувальники навантаження є багатофункціональними, з можливостями розвантаження SSL, формування трафіку та оптимізації для конкретної програми тощо. Крім того, розвиток хмарних обчислень привів до розробки хмарних балансувальників навантаження, які призначені для розподілу трафіку між серверами, розташованими в різних центрах обробки даних, що дозволяє досягти високого рівня масштабованості, надійності та доступності [4].

Головними постачальниками послуг на ринку хмарних сервісів є такі компанії, як Amazon Web Services, Google Cloud Platform та Microsoft Azure [5]. У кожній з цих компаній є свої унікальні особливості та переваги, що відповідають потребам розробників та компаній у різних галузях.

- Amazon Web Services (AWS) славиться своїми широкими можливостями в області обчислень, зберігання та організації баз даних, які надають комплексні рішення, що охоплюють широкий спектр потреб, дозволяючи розробникам та компаніям ефективно використовувати обчислювальні ресурси та зберігати дані;
- Google Cloud Platform (GCP) відома своїми інноваційними послугами у сфері штучного інтелекту та аналітики даних, які надають інструменти, що допомагають виявляти та аналізувати цінну інформацію з великих обсягів даних, а також розробляти та впроваджувати інтелектуальні рішення;
- Microsoft Azure пропонує гібридні рішення, що поєднуються з існуючою інфраструктурою та інструментами Microsoft, тобто це означає, що компанії можуть поєднувати хмарні та локальні ресурси для досягнення оптимальної ефективності та гнучкості.

Кожен з цих провайдерів має свої унікальні переваги, які дозволяють розробникам та компаніям вибирати рішення, що найкраще відповідає їхнім потребам у конкретній галузі або поставленому завданні.

Висновки

Використання контейнеризації при розгортанні вебдодатку дає можливість швидкого та ефективного масштабування, забезпечує високу надійність та гнучкість при перевантаженні.

Балансування та розподіл навантаження дозволяє досягти високого рівня продуктивності, надійності та доступності для програмних продуктів, наприклад, використовуючи Docker у поєднанні з Kubernetes, можна досягти більшої стійкості, доступності та автоматизації керування контейнерами в кластері.

Кожен хмарний провайдер має свої особливості, які роблять їх привабливими для різних клієнтів. Вибір певного провайдера залежить від конкретних вимог і цілей кожної організації, але як результат – ці провідні постачальники хмарних послуг гарантують надійність, масштабованість та інноваційні рішення у сфері хмарних сервісів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Who Made Kubernetes And Why Is It Popular? [Електронний ресурс]. URL: <https://www.opsramp.com/guides/why-kubernetes/who-made-kubernetes>
2. The evolution of containers: Docker, Kubernetes and the future. [Електронний ресурс]. URL: <https://www.techtarget.com/searchitoperations/feature/Dive-into-the-decades-long-history-of-container-technology>
3. Docker overview. [Електронний ресурс]. URL: <https://docs.docker.com/get-started/overview>
4. What Is Load Balancing & How Does It Work? [Електронний ресурс]. URL: https://www.cdnetworks.com/knowledge-center/what_is_load_balancing
5. AWS Vs Azure Vs GCP – The Cloud Platform of Your Choice? [Електронний ресурс]. URL: <https://www.veritis.com/blog/aws-vs-azure-vs-gcp-the-cloud-platform-of-your-choice/>

Канарський Богдан Петрович – студент групи 1КІ-19Б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця.

Войцеховська Олена Валеріївна – кандидат технічних наук, доцент кафедри обчислювальної техніки, Вінницький національний технічний університет, Вінниця.

Мартинюк Тетяна Борисівна – доктор технічних наук, професор кафедри обчислювальної техніки, Вінницький національний технічний університет, Вінниця.

Kanarskyi Bohdan P. — student of group 1KI-19B, Faculty of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia.

Voytsekhovska Olena V. — PhD, Assistant Professor of the Chair of Computer Techniques, Vinnytsia National Technical University, Vinnytsia.

Martyniuk Tetiana B. — Dr. Sc. (Eng), Professor of the Chair of Computer Techniques, Vinnytsia National Technical University, Vinnytsia.