УДК 681.12

В. О. Уманець

ВИСОКОДОСТУПНИЙ ТА ВІДМОВОСТІЙКИЙ WEB-ДОДАТОК У KUBERNETES

Вінницький національний технічний університет

Анотація

Розглянуто спосіб реалізації відмовостійкого та високодоступного WEB-додатку у Kubernetes на основі програмного забезпечення з відкритим кодом.

Ключові слова: високо доступна та відмовостійка система, Varnish, HAProxy, UCARP, Nginx, Apache, реплікація, MySQL, Open Source, Linux, Kubernetes, мікросервіси.

Abstract

The method of implementation of a fail tolerant and high availability web application in kubernetes, based on free open source software.

Keywords: high availability and fault tolerant web application, Varnish, HAProxy, UCARP, Nginx, Apache, replication MySQL, Open Source, Linux, Kubernetes, microservices.

Вступ

Висока доступність - це здатність системи уникати втрати сервісу, мінімізуючи час простою. Це виражається в термінах безперебійності роботи системи, як відсоток від загального часу роботи. У більшості випадків стратегія безперервності сервісу включатиме як високу доступність, так і відмовостійкість щоб гарантувати, що підтримуються найважливіші функції під час незначних невдач, а також у випадку лиха. У конкурентному середовищі, сервіс повинен бути постійно доступним і витримувати великі навантаження [1].

Постановка задачі

У сучасному світі, більшість IT компаній використовують все більше і більше віртуальних машин для розгортання їх продуктів. Це доволі просто, та не потребує великих витрат. Однак компанії не завжди доцільно підбирають параметри віртуальної машини для

розгортання серверу.

Kubernetes - на даний момент самий гнучкий і став дуже популярним, давши відповідь на найважливіше питання глобальних програм, здатних згодом залучити мільйони користувачів - як швидко розгортати, оновлювати і масштабувати модулі і сервіси моєї програми в хмарі, ефективно використовуючи всю доступну обчислювальну потужність? Контейнери чудово справляються із завданням універсальної підготовки та упаковки модулів програми та сервісів для виконання на будь-яких серверних версіях Linux. Запуск різнорідних додатків стає простим завданням. Залишається управляти розгортанням і масштабуванням. Неважливо, чи доступний спочатку лише один сервер, якщо ідея вдала, а реалізація хороша, привернется увага користувачів, і Kubernetes допоможе розгорнути, а потім масштабувати додаток для мільйонів, практично так само швидко, як якщо б їм користувалися тільки кілька людей.

Порівняння систем

У публікації минулого року розглянуто один з способів реалізації високодоступного та відмовостійкого WEB-додатку [1]. Дана система може бути ще кращою, перевівши WEB-додаток з віртуальних машин до контейнерів.

Все, що потрібно для запуску програми, міститься у віртуальній машині - віртуалізоване обладнання, ОС та будь-які необхідні бінарні файли та бібліотеки. Тому віртуальні машини мають власну інфраструктуру і є автономними. Кожна віртуальна машина повністю ізольована від хост-операційної системи. Крім того, для цього потрібна власна ОС, яка може відрізнятися від ОС хоста. У кожного є свої бінарні файли, бібліотеки та програми. На рис. 1 наведена реалізація віртуалізації.

Переваги: віртуальні машини зменшують витрати. Замість запуску програми на одному сервері віртуальна машина дозволяє використовувати один фізичний ресурс, щоб зробити роботу багатьох. Тому не доведеться купувати, підтримувати та зберігати численні стеки серверів. Оскільки є одна хост-машина, вона дозволяє ефективно керувати всіма віртуальними середовищами з централізованою потужністю гіпервізора. Ці системи повністю відокремлені одна від одної, тобто можна встановити кілька системних середовищ.

Найголовніше, що віртуальна машина ізольована від хост-операційної системи і є безпечним місцем для експериментів та розробки програм [2].

Недоліки: віртуальні машини можуть забирати багато системних ресурсів хост-машини. Запуск одного додатка на віртуальному сервері означає запуск копії операційної системи, а також віртуальної копії всього обладнання, необхідного для роботи системи. Це швидко збільшує використання оперативної пам'яті та процесора. Процес переміщення програми, що працює на віртуальній машині, також може бути складним, оскільки він завжди додається до операційної системи. Отже, потрібно буде перемістити додаток, а також ОС з ним. Також при створенні віртуальної машини гіпервізор виділяє апаратні ресурси, присвячені віртуальній машині.

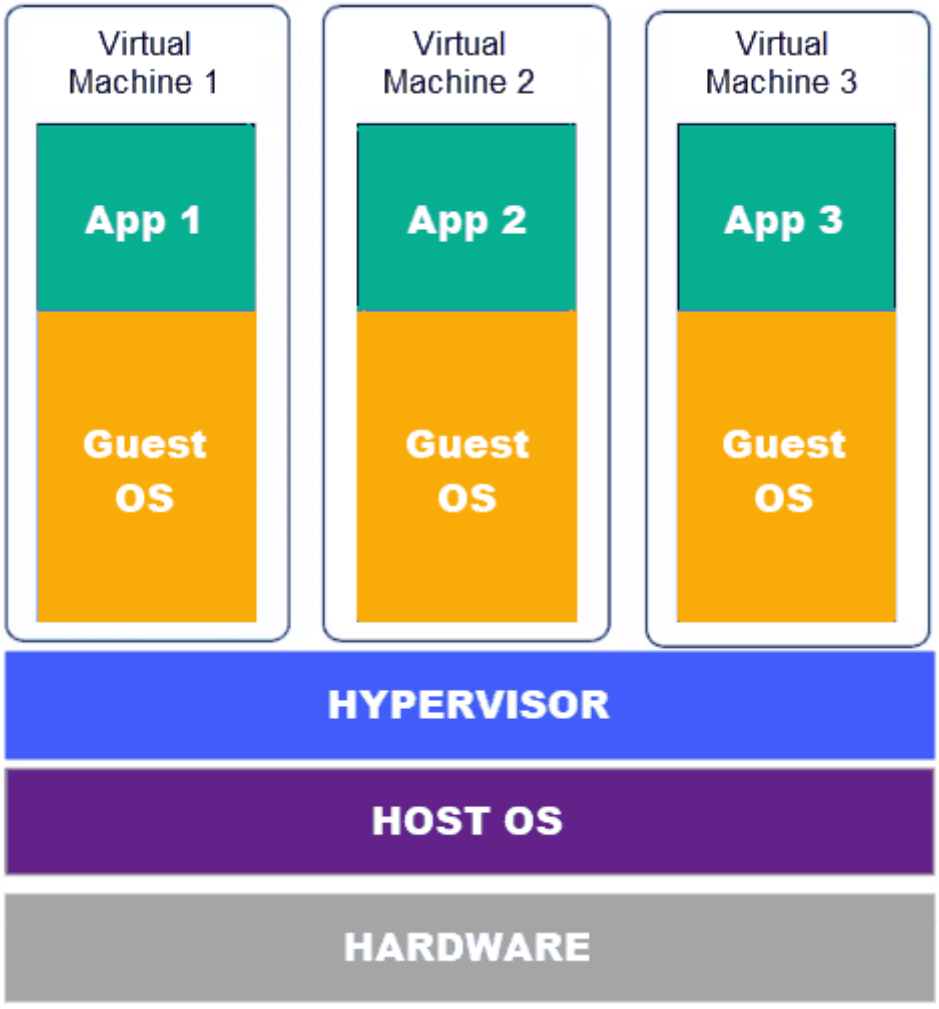

рис. 1 - Віртуальна машина

Контейнер - це середовище, в якому працює програма, яка не залежить від операційної системи. Він ізолює додаток від хоста шляхом віртуалізації. Це дозволяє користувачам створювати кілька робочих навантажень в одному екземплярі ОС. Ядро операційної системи хоста обслуговує потреби запуску різних функцій програми, розділених на контейнери. Кожен контейнер виконує окремі завдання. Це не може завдати шкоди хост-машині, а також не конфліктувати з іншими програмами, що працюють в окремих контейнерах.

Працюючи всередині контейнера, можна створити шаблон потрібного середовища. Контейнер по суті виконує знімок системи в певний час, забезпечуючи послідовність у поведінці програми. Контейнер розділяє ядро хоста, щоб запустити всі окремі програми в контейнері. Єдині елементи, необхідні кожному контейнеру, - це бінарні файли, бібліотеки та інші компоненти. На рис. 2 наведена реалізація контейнеризації.

Переваги: контейнери можуть бути розміром до 10 Мб. Це робить контейнери надзвичайно легкими та швидкими для запуску, на відміну від розгортання віртуальних машин, де потрібно розгортати всю операційну систему. Через їх розмір можна швидко масштабувати та виходити з контейнерів, а також додавати однакові контейнери. Крім того, контейнери відмінно підходять для впровадження безперервної інтеграції та безперервної розгортання (CI / CD). Вони сприяють спільній розробці, поширюючи та об’єднуючи зображення серед розробників.

Недоліки: контейнер використовує ядро хост-операційної системи та має залежність від операційної системи. Тому контейнери можуть відрізнятися від базової ОС залежностями, але не за типом. Ядро хоста обмежує використання інших операційних систем. Контейнери все ще не пропонують тієї самої безпеки та стабільності, що можуть віртуальні машини. Оскільки вони діляться ядром хоста, вони не можуть бути настільки ізольованими, як віртуальна машина. Отже, контейнери ізолюються на рівні процесу, і один контейнер може впливати на інші, порушуючи стабільність ядра. Більше того, як тільки контейнер виконує своє завдання, він закривається, видаляючи всі дані всередині нього. Якщо потрібно, щоб дані залишалися на хост-сервері, потрібно зберегти їх за допомогою томів даних. Для цього потрібна ручна конфігурація та забезпечення хосту.

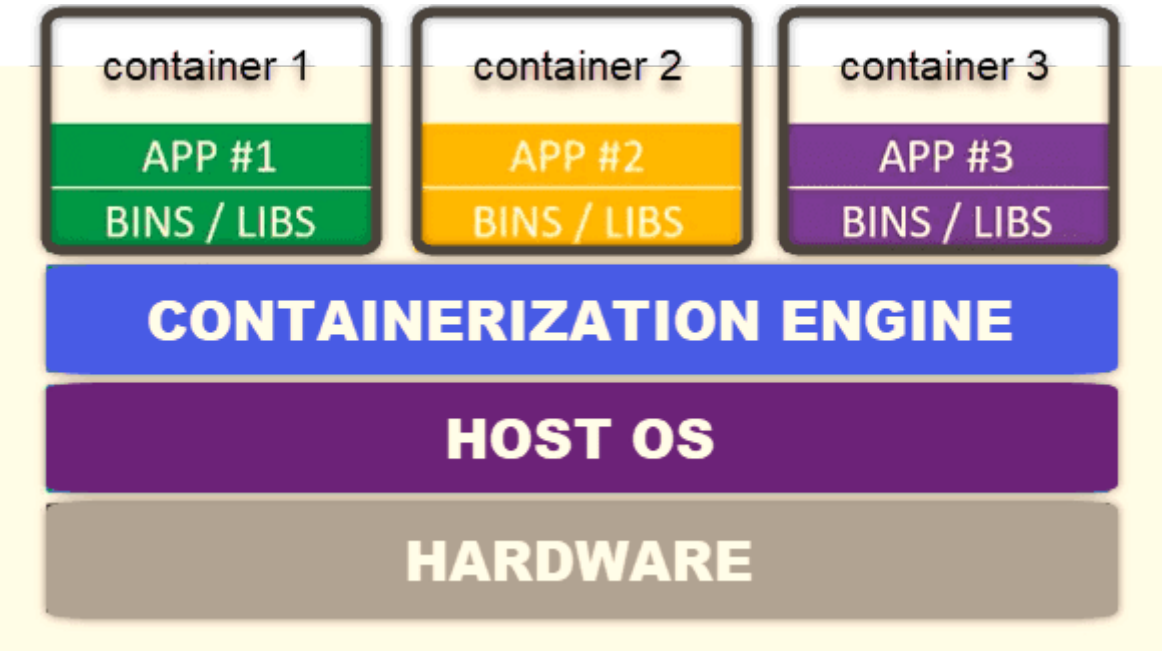

рис. 2 - Контейнер

Результати

Переведення системи з віртуальних машин на контейнери, які керуютьсся за допомогою Kubernetes, значно покращує відмовостійкість системи, так як система розділена на мікросервіси, які знаходяться на різних подах. З'являється можливість швидкого та легкого масштабування для забезпечення високодоступності.

Kubernetes може працювати на більшості кластерів обчислювальних ресурсів (серверів). Хмарні провайдери пропонують готові кластери довільних потужностей і розмірів для розгортання мікросервісів і додатків.

Все що потрібно від розробників мікросервісів або додатків - підготувати точки доступу через мережеві порти, і упакувати додаток з усіма залежностями в образ, на основі якого будуть запускатися контейнери.

Керуючим системам Kubernetes необхідно передати образ з мікросервісом і вказати бажаний стан цього мікросервіса - в звичайних випадках його ім'я, кількість примірників, необхідність доступу до нього з Інтернету.

Одним рядком можна вказати Kubernetes рівень масштабування мікросервіса, простою кількістю примірників, або логічно і просто налаштованим автоматичним масштабуванням.

Одним з основних переваг Kubernetes є те, що це відкритий проект, до всіх деталей якого є повний доступ. Розгорнути і налаштувати додаток можна з легкістю, скориставшись

публічними провайдерами хмари, такими як Google, Azure або AWS, і більш того, ніщо не заважає в будь-який момент налаштувати Kubernetes на своєму власному кластері, що знаходиться під вашим повним контролем, наприклад, якщо мова зайде про особливо важливих дані або їх географічні обмеження.

Kubernetes забезпечить максимізацію обчислювальних потужностей і мінімальні зусилля в порівнянні з ручним або напівавтоматичним створенням окремих віртуальних машин і розгортанням додатків на них [5].

Висновки

У даній публікації вказані переваги використання мікросервісної системи (контейнерів) та недоліки використання віртуальних машин. Продуктивність WEB-додатку залежить від: програмного забезпечення, яке на ньому використовується; апаратного забезпечення; правильного налаштування відмовостійкості та балансування навантаження.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Уманець В.О. ВИСОКОДОСТУПНИЙ ТА ВІДМОВОСТІЙКИЙ ВЕБ-ДОДАТОК [Електронний ресурс] / Е.Паламарчук В.Уманець — Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2019/paper/view/6995
2. Різниця між контейнерами та віртуальними машинами [Електронний ресурс] — Режим доступу: https://phoenixnap.com/kb/containers-vs-vms
3. Як віртуальні машини та контейнери порівнюють та протиставляють [Електронний ресурс] — Режим доступу: https://www.rcrwireless.com/20170721/network-infrastructure/20170721network-infrastructurehow-virtual-machines-and-containers-compare-and-contrast-tag27-tag99
4. Віртуальні машини та контейнери для мікросервісної архітектури [Електронний ресурс] — Режим доступу: https://dzone.com/articles/vms-vs-containers-for-microservices
5. Kubernetes: перші кроки [Електронний ресурс] — Режим доступу: https://ipsoftware.ru/posts-cloud/k8s-quickstart

Уманець Владислав Олександрович — студент групи 1АКІТ-19м, факультет комп'ютерних систем та автоматики, Вінницький національний технічний університет, Вінниця, e-mail: umanets.vladyslav@gmail.com

Науковий керівник: ***Паламарчук Євген Анатолійович*** — кандидат технічних наук, доцент кафедри автоматики та інформаційно-вимірювальної техніки, Вінницький національний технічний університет, м. Вінниця

Umanets Vladyslav A. — Department of Computer Systems and Automatic, Vinnytsia National Technical University, Vinnytsia, e-mail : umanets.vladyslav@gmail.com

Supervisor: ***Palamarchuk Yevhen A.*** — PhD, Docent of Automatics and Informatics and Measurement Techniques Department, Vinnytsia National Technical University, Vinnytsia, email : p@vntu.edu.ua