Контейнер технологиите срещу виртуалните машини. Docker и Kubernetes

Иван Белев
УНСС, София, България, i.belev@unwe.bg

Резюме

Докладът представя общи характеристики на контейнер технологиите и прави съпоставка с технологиите за виртуализация - виртуални машини. Очертани са основни аспекти на приложение на контейнерите, както и основните платформи за създаване и управление на контейнери и контейнер клъстери – Docker и Kubernetes.

Keywords: контейнер технологии, виртуални машини, Docker, Kubernetes.

Abstract

This report describes the main characteristics of container technology and compares them to virtualization technology and virtual machines. The key aspects for the application of containers are outlined as well as the most popular platforms for creation and orchestration of containers and container clusters – Docker and Kubernetes.

Keywords: container technology, virtual machines, containers, virtualization, Docker, Kubernetes.

ВЪВЕДЕНИЕ

В днешно време в света на информационните технологии голямо влияние оказват технологиите, свързани с термина виртуализация. Виртуализацията в компютърен аспект най-общо казано означава да се създаде виртуална версия на даден компютърен компонент, който по начало е реален (физически). Някои от компонентите, които могат да се виртуализират са хардуерни платформи, компоненти за съхранение на данни, мрежови компоненти и други.

Възникването и усъвършенстването на технологиите за виртуализация води до навлизането на информационните технологии в нов етап от развитието си. В последните години наред с технологиите за виртуализация се развива и друга сходна технология – технологията за „контейнеризация“ на компютърни приложения (т.нар. контейнери).

Въпреки, че контейнер технологията възниква преди повече от тридесет години, през последните петнадесет години тя се развива със силни темпове, а в последните няколко години навлиза в по-широка употреба. Контейнер технологиите и технологиите за виртуализация, специално виртуализацията на компютърни платформи (т.нар. виртуални машини), имат доста общи характеристики, както и някои съществени разлики, които се разглеждат от настоящото изследване.

1. КОНТЕЙНЕР ТЕХНОЛОГИИ – ОБЩИ ХАРАКТЕРИСТИКИ И ИСТОРИЧЕСКО РАЗВИТИЕ

Контейнерите в аспекта на информационните технологии могат да бъдат обяснени най-лесно чрез използването на пример извън информационните технологии, а именно – контейнерите за пренасяне на товари. Години преди да се създаден единен стандарт за пренос на товари чрез контейнери, превозвачите са изпитвали огромни затруднения да транспортират товари от всякакъв вид, размер и форма. Въвеждането на

стандарт, който регламентира формата и размера на контейнерите, както и други техни характеристики, дава възможност да се създадат останалите елементи, които да осигурят процеса по транспорт на товари чрез контейнери – морски кораби за превоз на контейнери, кранове за товарене и разтоварване, камиони, които превозват контейнери и др.

Подобно на физическите контейнери, контейнерите в аспекта на информационните технологии всъщност са механизъм за окомплектоване на програмния код на дадено компютърно приложение или група от приложения, както и други компоненти, от които зависи изпълнението на този код, в т.нар. контейнер, който може след това да бъде изпълнен на различни компютърни среди с минимални специфични изисквания към средите. Целта на „контейнеризацията“ на компютърни приложения е да се изолира само приложението и минимално необходимите компоненти за работата му в цялостен завършен компонент (контейнер). Всичко излишно не се включва и това прави контейнера много малък и бърз за изпълнение. Малкият размер на контейнерите позволява на една платформа (в общия случай работна станция или сървър) да се изпълняват голям брой приложения в контейнери, които споделят общи хардуерни ресурси, операционна система и други необходими за изпълнението елементи.

Следвайки примера с физическите контейнери, корабът превозвач на контейнери се явява сървър (или работна станция), която изпълнява голям брой приложения, пакетирани в контейнери. Всички контейнер-приложения зависят от някои общи характеристики на сървъра, за да бъдат изпълнявани, но безпроблемно могат да бъдат прехвърлени на друг сървър с други характеристики.

Най-важните характеристики на контейнерите могат да бъдат обобщени както следва:

- Контейнерите са самостоятелни елементи, включващи всичко необходимо за изпълнение на даденото софтуерно приложение;
- Контейнерите са своеобразна форма на виртуализация, но на ниво операционна система;
- Контейнерите включват всички необходими компоненти – библиотеки, конфигурационни файлове, бинарни файлове;
- Контейнерите дават възможност за осигуряване на платформена и инфраструктурна независимост на приложенията.

В исторически план контейнер технологиите имат дълга история, която започва през 1979 година. Тогава във версия седем на операционната система Unix се добавя услугата “chroot”. Тази услуга дава възможност за създаване на „под-среда“ в операционната система, като в тази под-среда могат да бъдат изпълнявани приложения и услуги, изолирано от останалата част от системата. Приложението на chroot е да се изпълняват тестови сценарии на производствените сървъри и системи. Това дава началото на идеята за контейнери.

Двадесет години по-късно, през 2000 година се създават компонентите “Jails” в Unix-базираната операционна система FreeBSD. FreeBSD Jails компонентите добавят към Unix chroot допълнителни компоненти, които също са изолирани, като например потребители, файлове, мрежови слой. Това позволява всяка „клетка“ да има и собствен IP адрес, осигурявайки логическа изолация на мрежовия слой.

Още четири години по-късно в операционната система Solaris OS се появяват компоненти, които се наричат Solaris контейнери, които позволяват създаването на т.нар. зони, в които се изолират приложения и други компоненти.

През 2008 година в операционната система Linux се появява LXC – Linux Containers, което служи за основа при разработването на Docker контейнерите през

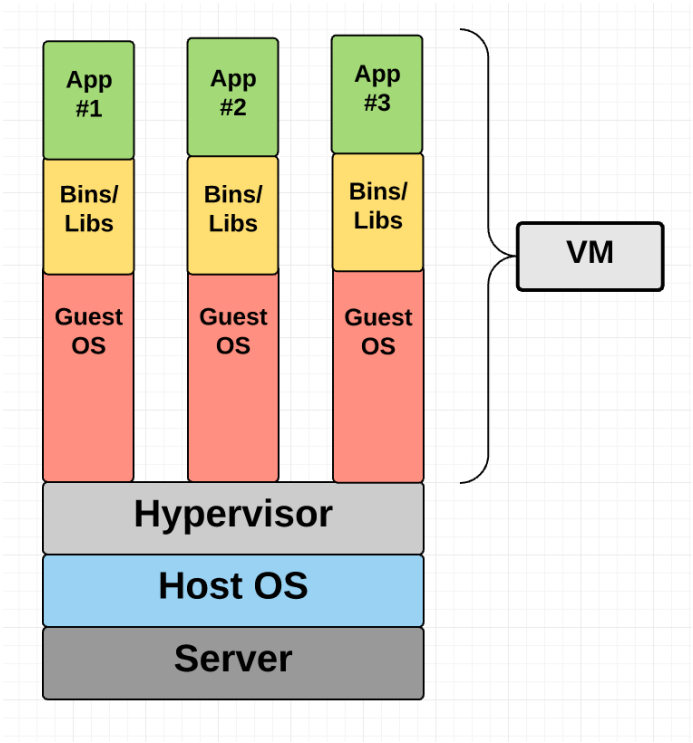
2013. След първоначалните варианти на Docker контейнерите от компанията Docker заменят LXC със собствено разработена библиотека за контейнери наречена „libcontainer“.

С появата на Docker контейнерите и все по-честото им използване се налага технологията за контейнерите да бъде стандартизирана, за да може контейнерите да са платформено и инфраструктурно независими. Поради тази причина през 2015 се създава Open Container Initiative, с което се създават единни стандарти за контейнер формати.

Създаването на стандартите, засягащи контейнер форматите, дава възможност контейнер технологиите да надраснат Unix-базираните операционни системи и да достигнат и до потребителите на операционна система Windows. Това става през 2016 година с Windows 10 и Wondows Server 2016, при които е възможно да се пускат Docker контейнери.

2. КОНТЕЙНЕР ТЕХНОЛОГИИТЕ И ТРАДИЦИОННАТА ВИРТУАЛИЗАЦИЯ

Контейнерите се приемат като своеобразна форма на виртуализация, която в много аспекти се доближава до традиционната виртуализация. Общите характеристики са много, но двете технологии имат и някои ключови разлики. На Фигура 1 е изобразена принципна схема на компонентите при виртуализация чрез виртуални машини:



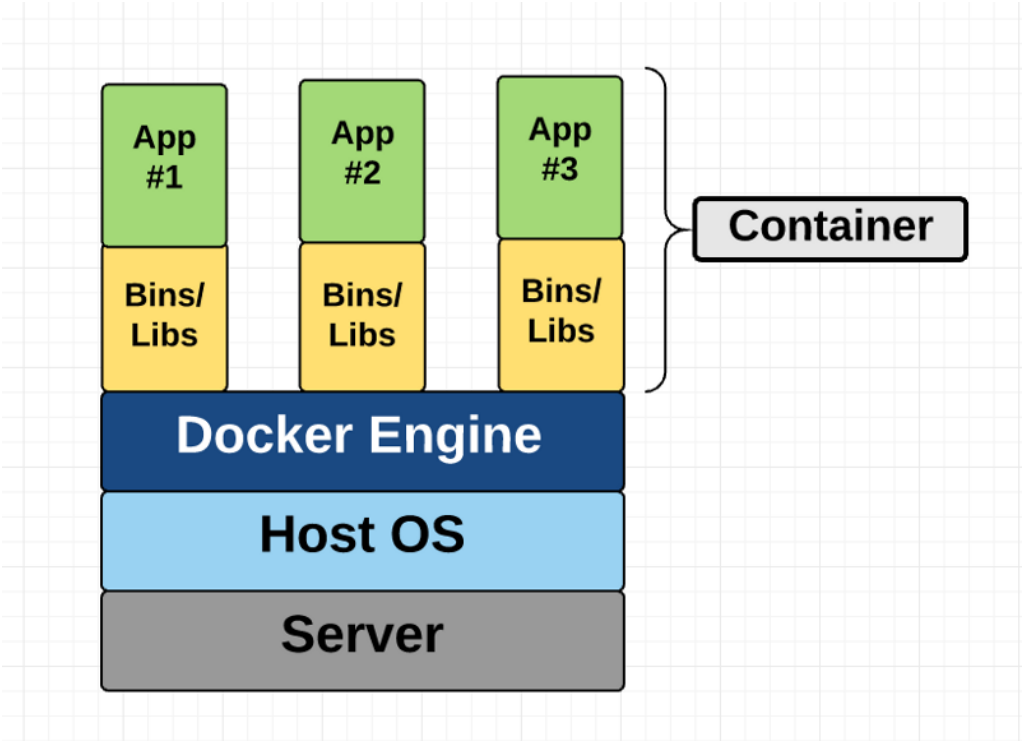
Фиг.1. Компоненти на традиционна виртуализация [3]

При виртуализация чрез виртуални машини на един физически сървър, който има своя операционна система се поставя т.нар. хипервайзор (Hypervisor), който осъществява създаването и управлението на отделните виртуални машини и разпределението на ресурси. В зависимост от ресурсите, на сървъра могат да се създадат множество различни виртуални машини (VM). Всяка една от тях си има собствена операционна система (Guest OS), която може да бъде различна и върху която са поставени необходимите библиотеки, бинарни файлове и разбира се, самите

приложения. Ключовият момент е наличието на операционна система и прилежащите и елементи във всяка отделна виртуална машина.

Ако следваме примера за контейнерите за превоз на товари, сървърът се явява товарен кораб който плува в океана, но него вместо да има контейнери, има малки водни басейни, в които плуват други по-малки товарни кораби, на които са разположени контейнерите с товарите.

На Фигура 2 по-долу е показана принципна схема на компонентите при Docker контейнери:



Фиг. 2. Компоненти при Docker контейнери [3]

При Docker контейнерите хипервайзора е заменен с Docker софтуерен компонент за управление и изпълнение на контейнери (Docker Engine). В зависимост от ресурсите на дадения сървър могат да се създадат и пуснат много на брой контейнери. Основната разлика е, че при контейнерите се премахва нуждата за дублиране на операционни системи. Всеки контейнер включва приложението и необходимите му библиотеки и бинарни файлове (Bins/Libs). За разлика от виртуалните машини, контейнерите използват операционната система на сървъра. Виртуализацията се случва на ниво операционна система. Все пак не е изключено в контейнера да бъде включена операционна система, ако това е необходимо за работата на дадено приложение.

Споделянето на операционната система на сървъра от всички контейнери води до няколко важни предимства на контейнер технологиите пред традиционната виртуализация:

- Контейнерите използват много по-малко ресурси;
- Контейнерите са малки по размер;
- Контейнерите се инициализират много бързо;
- Използването на контейнери отваря възможност за осъществяване на подход, базиран на т.нар. “microservices” (микро услуги);
- Контейнер инфраструктурата е лесно скалируема при нужда от увеличаване или намаляване на физическите ресурси;

- Контейнерите са лесно преносими към друга платформа и са платформено независими;
- Технологията на контейнерите е базирана на отворен код и регулирана от отворени стандарти;

В следващата част от научния доклад са разгледани два от най-разпространените приложения, които са свързани с контейнер технологиите.

3. DOCKER И KUBERNETES

В изложението многократно бе споменато решението Docker. Към датата на създаване на настоящото научно изследване (Септември 2019) Docker е най-използваното решение за управление на контейнери. Docker предоставя цялостна платформа за изпълнение на контейнери под Linux и Windows, както и в облачна среда. Docker предоставя и хранилище на предварително създадени образи (“Images”) за различни приложения, които са готови за използване под формата на контейнери. Това улеснява работата на потребителите, които искат бързо да пуснат дадено приложение под формата на контейнер.

Някои от конкурентите на Docker са:

- CoreOS rkt (познат като Rocket);
- Apache Mesos;
- Red Hat Open Shift Container Platform;
- Rancher.

От друга страна голяма популярност има и платформата Kubernetes. Kubernetes не е платформа, която се конкурира директно с Docker, а е по скоро следващата стъпка в еволюцията на контейнер технологиите и стандартите – платформа, която надгражда Docker. Терминът, с който се характеризира Kubernetes е „Container Orchestration“ – оркестрация на контейнери. Kubernetes позволява да се автоматизира работата по създаване, стартиране и спиране на контейнери. Също така чрез Kubernetes се осъществява автоматизиране на балансирането на една среда от приложения. Контейнерите работят в режим на „групиране“ (Cluster), което позволява непрекъсваемост на работата на приложенията, разпределение на натоварването чрез автоматично увеличаване или намаляване на броя на работещи контейнери. Контейнер среда, изградена в Kubernetes се определя като само лекуваща се (self-healing environment) – при възникване на проблем средата го коригира автоматично, базирайки се на предефинирани сценарии и правила.

Началото на Kubernetes е през 2003 г. под името Borg System, разработено като вътрешна система в Google. През 2013 г. Borg System прераства в Omega cluster management system by Google. На следващата година Google предоставят платформата на общността като отворен код под името Kubernetes, а още една година по-късно през 2015 г. е публикувана версия 1.0 на Kubernetes. В разработката на платформата и стандартите за работата и със своята подкрепа се включват всички технологични гиганти като Microsoft, IBM, Google, Oracle и т.н. Към момента на настоящото изследване (Септември 2019) актуалната версия на Kubernetes е 1.16.

ЗАКЛЮЧЕНИЕ

Контейнер технологиите в сферата на информационните технологии възникват в края на седемдесетте години на двадесети век, но превръщането им в „гореща“ технология се случва след 2013 г., провокирано от развитието и налагането на компютърната виртуализация. Сходствата между двете технологии не са малко, но някои важни разлики в принципа на работа водят до съвсем различни насоки в

използването на двете технологии. Появата на Docker и Kubernetes водят до обособяването на стандарти в областта на контейнерите и до възникването на термина – оркестрация на контейнери.

БИБЛИОГРАФИЯ

1. LOWE, S. (2017) Containers for Dummies, HPE and Docker Special Edition, John Wiley & Sons, Inc.
2. LARDINOIS, F. (2016) WTF is a container? [Online] Available from: https://techcrunch.com/2016/10/16/wtf-is-a-container/ [Accessed 20/09/2019].
3. KASIREDDY, P. (2016) A Beginner-Friendly Introduction to Containers, VMs and Docker [Online] Available from: https://www.freecodecamp.org/news/a-beginner-friendly-introduction-to-containers-vms-and-docker-79a9e3e119b/ [Accessed 20/09/2019].
4. RUBENS, P. (2017) What are containers and why do you need them? [Online] Available from: https://www.cio.com/article/2924995/what-are-containers-and-why-do-you-need-them.html [Accessed 20/09/2019].
5. YEGULALP, S. (2014) 4 reasons why Docker's libcontainer is a big deal? [Online] Available from: https://www.infoworld.com/article/2607966/4-reasons-why-docker-s-libcontainer-is-a-big-deal.html [Accessed 20/09/2019].
6. FreeBSD Documentation, [Online] Available from: https://www.freebsd.org [Accessed 20/09/2019].
7. Kubernetes Documentation, [Online] Available from: https://kubernetes.io/docs/home/ [Accessed 20/09/2019].
8. PAPP, A. (2018) The History of Kubernetes on a Timeline, [Online] Available from: https://blog.risingstack.com/the-history-of-kubernetes/ [Accessed 20/09/2019].
9. Docker Documentation, [Online] Available from: https://www.docker.com/ [Accessed 20/09/2019].