

コンテナネットワークの基礎知識【第1回】:

いまさら聞けない「コンテナ」と「仮想マシン」の根本的な違いとは？

<https://techtarget.itmedia.co.jp/it/news/2005/18/news04.html>

「コンテナ」はアプリケーション開発に限らず、インフラ運用の技術としても関心が高まりつつあります。コンテナを理解するには、仮想マシンとの違いを整理することが重要です。何が違うのでしょうか。

2020年05月22日 05時00分 更新

[奈良昌紀 細谷典弘 千葉 豪, ネットワンシステムズ]

関連キーワード

[Docker](#) | [ネットワーク](#)

複数のクラウドやオンプレミスなど、異なる環境で迅速にアプリケーションを実行するための技術として、「コンテナ」の利用が広がっています。コンテナ管理ツールの「Docker」に加え、「Kubernetes」のようなコンテナオーケストレーションツールが利用できるようになり、開発者に限らずITインフラ運用者の間でもコンテナへの注目度が高まっています。本連載はコンテナやコンテナネットワークの基礎知識など、コンテナをITインフラ運用に用いる場合に知っておくべきポイントを解説します。

併せて読みたいお薦め記事

ネットワーク技術のトレンド

- [「5G」「Wi-Fi 6」だけではない 知らないともまずいネットワーク5大トレンド](#)
- [「NetOps」とは何か？ DevOps時代のネットワーク運用方法](#)

コンテナ関連記事

- [VMwareは「コンテナ」市場で存在感を示すために何をしているのか](#)
- [進化するコンテナ技術 2019年にさらに人気が高まる理由とは](#)
- [コンテナ管理ツールは「Docker」だけではない 主要ツールを紹介](#)

コンテナは仮想化技術の一つであり、単一のOSで複数のアプリケーションをそれぞれ隔離された状態で実行可能にします。コンテナを利用せず通常のOSで複数のアプリケーションを実行すると、そのOSが稼働するサーバ、そのサーバが利用するストレージ、ネットワークなどのリソースはアプリケーション間で共有されます。これに対し、コンテナでは各コンテナが個別に割り当てられたストレージやネットワークを持ち、他のコンテナから隔離された状態でアプリケーションを実行できます。

コンテナと仮想マシン、3つの違い

コンテナの役割は、隔離されたアプリケーション実行環境を用意することです。その点で、広く普及している仮想マシン（VM）と類似しています。ただしVMはハイパーバイザーなどの仮想化ソフトウェアによってハードウェアを抽象化し、単一のハードウェアで複数のOSが起動することを実現する技術であり、仕組みはコンテナと異なります（図1）。まずはVMと比較することで、コンテナの特徴を考えてみましょう。

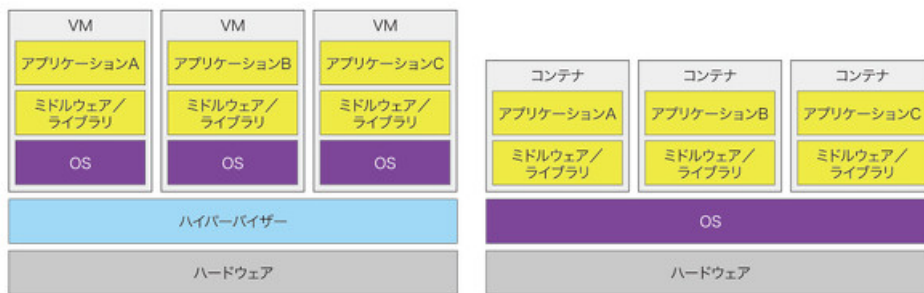


図1 VM（ハイパーバイザー型、左）とコンテナの構造《クリックで拡大》

違い1. イメージサイズと可搬性

VMはVMごとにOS（ゲストOS）を必要とします。そのためVMの実行に必要なデータをまとめたイメージファイル（以下、VMイメージ）には、アプリケーション本体とその実行に必要なライブラリ、ゲストOSが含まれます。一方コンテナはOSを利用して起動します。アプリケーションの実行に必要なミドルウェアやライブラリをパッケージ化したイメージファイル（以下、コンテナイメージ）はアプリケーション本体とその実行に必要なライブラリなどで構成され、VMとは異なりゲストOSは含まないため、VMイメージと比べるとファイルサイズが小さく可搬性に優れています。

違い2. 起動にかかる時間

VMはゲストOSを起動してからアプリケーション本体やミドルウェアを起動します。これに対してコンテナは起動済みのOSでアプリケーション本体やミドルウェアを実行するだけなので、アプリケーションの起動にかかる時間はVMよりも大幅に短縮します。

違い3. 隔離レベル

VMはハイパーバイザーなどのサーバ仮想化ソフトウェアによってOSを含めて隔離されるため、隔離の程度はコンテナより高くなります。VMは単一のハイパーバイザーで複数種類のゲストOSを起動させ、各ゲストOSのアプリケーションを実行することが可能です。これに対してコンテナは単一のOSで個々のコンテナに対して論理的にリソースを割り当てます。OS部分を共有する必要があるため、異なるOS向けのアプリケーションをコンテナとして起動することはできません。例えば「Linux」で稼働するコンテナで「Windows」用アプリケーションを起動することはできません。

アプリケーションの一貫した実行環境

こうした特徴を持つコンテナは、アプリケーションの実行環境として開発者の間で広く利用されています。コンテナイメージを再利用すればアプリケーションの実行環境を簡単に再現することが可能だからです。必要なミドルウェアやライブラリをひとまとめにすることで、開発、テスト、ステージング（検証）、本番といったアプリケーション開発の各フェーズで一貫したアプリケーションの実行環境を用意できます（図2）。オンプレミスかクラウドか、物理マシンかVMかといった環境の違いにとらわれることなくアプリケーションを実行できることもコンテナの利点です。

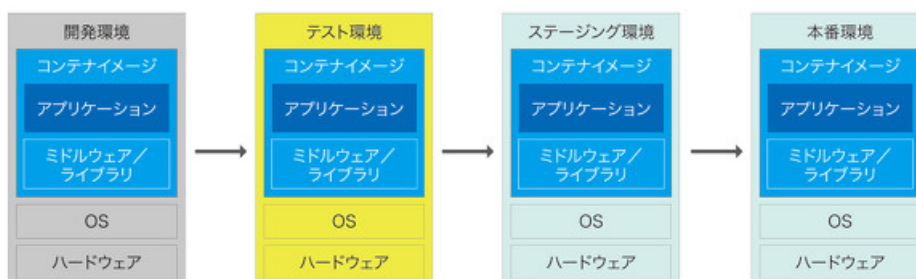


図2 開発におけるコンテナの活用例《クリックで拡大》

本番環境にもコンテナの導入が一般的になれば、Kubernetesのようなコンテナオーケストレーションツールを組み合わせることでインフラを抽象化と運用の高度化が可能になります。

□

次回はDockerのコンテナ利用における役割と、コンテナ運用時に重要なネットワークの基礎知識を解説します。

執筆者紹介

奈良昌紀(なら・まさのり) ネットワンシステムズ ビジネス開発本部 第1応用技術部

通信事業者のデータセンターにおいてネットワークやサーバの運用を経験後、ネットワンシステムズに入社。帯域制御やWAN高速化製品、仮想化関連製品を担当後、主にクラウドや仮想インフラの管理、自動化、ネットワーク仮想化の分野に注力している。

細谷典弘(ほそや・のりひろ) ネットワンシステムズ ビジネス開発本部第3応用技術部

データセンターネットワークの他、マルチクラウド向けのハードウェアやソフトウェアの最先端技術に関する調査・検証、技術支援などを担当。注目分野は「Kubernetes」。放送システムのIP化に向けた技術調査・検証も担当している。

千葉 豪(ちば・ごう) ネットワンシステムズ ビジネス開発本部 第1応用技術部

IaaS (Infrastructure as a Service) をはじめとしたクラウド基盤技術および管理製品を担当。コンテナ技術を中心とした開発・解析基盤の構築から運用、コンテナに関連した自動化技術や監視製品の技術検証などに注力している。

Copyright © ITmedia, Inc. All Rights Reserved.

