



# データ・サーバの コンテナ化

土屋 健

### ● ローカルPC上に作ったシステムの課題

第2部と第3部では、ローカルPC上でシステムを開発してきました。

開発に使用したオープンソース・ソフトウェア、開発言語は、どの環境でも動作するものを選んだので、クラウド、ラズベリー・パイ、Linux、Windowsなど、どこでも動作するものになっています。

それでも、環境に合わせてシステムのインストールを行い、環境を整えるには手間がかかります。また、システム起動に必要なソフトウェア(ミドルウェアやライブラリ)に関して、バージョン依存するケースも少なくありません。特に商用では、特定のバージョンの組み合わせしかサポートしないことも多いです。

### ● 仮想化サーバ上でシステムを稼働させるケースが増えた

そんな課題を解決するために、仮想化を導入する方法があります。最近ではコンピュータ性能が向上し、仮想化によるオーバーヘッドを気にしなくてよくなったので、仮想化サーバ上でシステムを稼働させるケースが多いです。本章ではデータ・サーバを例に、コンテナ化する方法を解説します。コンテナがどのようなものか伝えられればと思います。

## サーバをコンテナ化する理由

### ● 利点

システムの運用/管理の利便性からコンテナを利用したシステム構成が一般的になってきています。コンテナとは、環境分離を実現する仮想化の一種です。仮想化する大きな理由は次の通りです。

#### ▶ コンピュータ利用を効率化し管理を省力化する

コンピュータの高性能化によって1人が全てのリソースを使い果たすことが少なくなっています。サーバについては複数人でシェアすることで無駄なリソースやコストを削減できます。また、仮想化したアプリケーションは動作環境と共に1つのまとまり(イメージ)として管理されるので、そのイメージを持って行

くことで、別の場所でも簡単に動作させられます。このような特徴は、ハードウェア故障時に短時間で環境復旧できることや、システム複製が簡単にできることにつながります。

#### ▶ セキュリティを確保する

サーバを共有する場合、共有するユーザ同士が互いに干渉しないようにします。仮想化ではそういった個々の環境を分離させます。

### ● 仮想化ではコンテナが主流

仮想化方式としてコンテナが選ばれるのは、仮想化の効率が良いからです。コンテナではカーネルは共有し、カーネルの上に隔離されたアプリケーション実行環境を用意します。カーネルは変更できませんが、ベースとなるOSとは異なるライブラリやアプリケーションをコンテナ環境に閉じ込めて実行することが可能です。

コンテナ型は、仮想マシンを用意する方式ではなく(図1)、あくまで隔離されたOS/アプリケーション実行環境を用意するので、仮想化によるオーバーヘッドがなく、

- 他のコンテナ環境との隔離によるセキュリティ向上
- ホストOSとは異なるアプリケーション実行環境を用意

できます。リソースの使用効率が非常に高いため、現在のサーバ仮想化方式の主流と言えます。ただし、分離の度合いが低いため、カーネルや他のアプリケーションの脆弱性の影響を受けるのがデメリットです。このデメリットについても認識されており、対応済みのソフトウェアも存在します。コンテナとホスト型仮想化のハイブリッドの構成が有力で、多くの局面で利用されています。

### ● システムをコンテナ化する利点

次にシステムをコンテナ化する利点を整理します。

- 仮想化のオーバーヘッドが小さく性能が低いPC環境でもアプリケーションを複数立ち上げる程度の負荷で仮想環境を起動できる