

記事

[Toshihiko Minamoto](#) · 2021年6月3日 4m read

コンテナ - コンテナとは？

コンテナ

[InterSystems IRIS Data Platformの公開](#)により、Dockerコンテナでも製品を提供しています。コンテナとは一体何でしょうか。

基本的なコンテナの定義は、プロセスのサンドボックスの定義です。

コンテナは、たとえば実行できるという点において、仮想マシン（VM）に似た部分を持つソフトウェア定義パッケージです。

コンテナは、完全なOSエミュレーションを使わずに分離することができるため、VMよりもはるかに軽量です。

本質的に、コンテナは、どのようにアプリケーションをシステムから別のシステムに確実に移動し、それが動作することを保証できるのかという問題に対する答えと言えます。アプリケーションのすべての依存関係をコンテナにカプセル化し、プロセス分離領域を作成することにより、アプリケーションソリューションがプラットフォーム間で移動した場合でも動作するという高い保証を得ることができます。

プロセスは、オペレーティングシステムによって実行が可能です。これらのプロセスはアドレス領域、ネームスペース、cgroupなどを共有し、通常、OSの環境全体にアクセスすることができます。それらのスケジューリングと管理は、OSが行います。すべて良いことではありますが、特定のプロセスやいくつかのプロセスを分離して、特定のタスクや演算、またはサービスを実行したい場合はどうでしょうか。

手短かに言えば、プロセスを分離できる機能こそ、コンテナが提供している機能なのです。

したがって、コンテナをプロセスのサンドボックスとして定義することができます。

では、サンドボックスとは何でしょうか。サンドボックスは、コンテナがプロセスを持つ分離レベルです。この機能は、ネットワークインターフェース、マウントポイント、インタープロセス通信（IPC）、ユニバーサルタイムシェアリング（UTS）といったシステムのほかの重要な部分もサンドボックス化することのできるネームスペースと呼ばれるLinuxのカーネル機能により実装されています（<https://en.wikipedia.org/wiki/Linuxnamespaces>）。

コンテナまたはサンドボックスは、制御グループまたはcgroupと呼ばれる別のカーネル機能によっても管理または制御することができます（<https://en.wikipedia.org/wiki/Cgroups>）。コンテナがほかのコンテナやホストとリソースを共有する上で、ほかのネイバーに害を与えないように、コンテナにルールが適用されます。

コンテナとVMの違いを理解するには、VMを家に、そしてコンテナをマンションに例えることができます。



VMは、一軒家のように、自己完結型で独立しています。
それぞれの家には、配管、暖房、電気といった固有のインフラストラクチャが備わっており、最低要件も与えられています（最低、寝室は1室、屋根は1つなど）。

一方のコンテナは、共有のインフラストラクチャを使用するように作られているため、マンションとして捉えることができます。マンションの建物は、配管、暖房、電機システム、正面玄関、エレベーターなどを共有していますが、これと同様に、コンテナも、Linuxカーネルを通じてホストが提供しているリソースを使用します。また、マンションの各居住戸のサイズや形には様々なものがあることも考えましょう。

コンテナには完全なOSはなく、/binの実行可能ファイルや/etcの構成ファイルと定義ファイルといった、最小限必要なLinux OSしか備わっていないため、サイズが非常に小さくなります。そのため、その場所を移動したり、スピンアップする場合に、一秒きっかりで敏速に行うことができます。コンテナを構築した瞬間から、ソフトウェアファクトリーのプロビジョニングパイプラインを通じ、本番環境での最終実行に至るまで、アジリティを得られることになります。ちなみに、コンテナはCI/CDマイクロサービスアーキテクチャのコンテキストにピッタリと収まりますが、これはまた別の機会にお話ししましょう。

コンテナ内のプロセスは、コンテナのライフサイクルと密接な関係があり、
コンテナを起動

すれば、アプリのすべてのサービスも起動して実行し（たとえば、あるWebサーバーコンテナのポート80と、InterSystems IRISコンテナのポート57772と1972）、コンテナを停止すれば、すべてのプロセスも停止します。

この記事では、コンテナのランタイムの基本的な概念について説明しました。コンテナは、プロセスをホストやその他のコンテナから分離するサンドボックスです。

コンテナを理解するには、その[イメージ](#)に関する別の側面もあります。
これについては、第2部の記事で説明することにします。

Published on InterSystems Developer Community (<https://community.intersystems.com>)

ソースURL:

<https://jp.community.intersystems.com/post/%E3%82%B3%E3%83%B3%E3%83%86%E3%83%8A-%E3%82%B3%E3%83%B3%E3%83%86%E3%83%8A%E3%81%A8%E3%81%AF%E3%81%BC%E3%81%9F>