

IoT時代のモヤモヤを自宅で解消

メカニズム丸見え! ラズパイ AI サーバを作る

第22回 軽くて今風の仮想化 Docker をラズパイに入れる

土屋 健

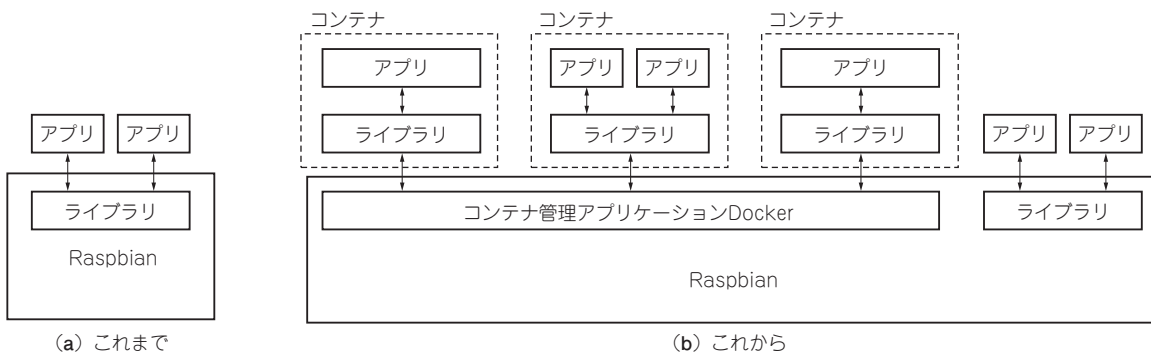


図1 仮想環境導入により、それぞれのアプリの実行環境を分離できるためセキュリティが向上する

今回から基盤要素として、サーバの仮想化を行います。軽くて使いやすい定番 Docker によるコンテナ方式を選びました。これまでの仮想化していない環境では図1(a)に示す通り本体の Raspbian 上でアプリケーションが動いていました。Docker によるコンテナ方式の仮想環境を使う場合、図1(b)に示す通りアプリケーションは Raspbian 上に作られたコンテナの中で動きます。

仮想化のメリット

クラウド・サーバは、1台のサーバ上で多くのサーバを動かしています。大抵の利用者はハードウェアを100%使うような使い方はしないことから、コストとサービス品質のバランスをとっています。設置面積や消費電力を削減できます。

このときサーバのハードウェアを効率良く利用したり、ハードウェア故障時の復旧を効率的に行ったりするために、仮想化の手法を取り入れています。

また、セキュリティ確保のために環境を分離するという側面もあります。

サーバの主な仮想化方式

初期のころは、準仮想化や完全仮想化といった仮想化技術を利用したシステムが多かったのですが、最近ではコンテナ環境を提供することも多いです。それぞれの仮想化方式には一長一短があり、利用者は用途に合わせて選択できるようになっています。

● その1：完全仮想化

完全仮想化は、通常の PC 環境をソフトウェア的に用意します。OS のインストールや OS 上へのアプリケーションのインストールなどが可能です。ホスト OS 型やハイパバイザ型があり、いずれの場合も通常の PC 向け OS をゲスト OS としてインストールできます。

▶ ホスト OS 型

ホスト OS 型は、Windows や Linux, macOS といった環境上に仮想化ソフトウェアをインストールし、その上にゲスト OS 環境を用意する方式です。ホスト OS 上を使いながら、仮想環境のゲスト OS も使用できます。

▶ ハイパバイザ型

ハイパバイザ型は、ハードウェアの管理や仮想環境の構築だけに特化した専用 OS (ハイパバイザ) をインストールして、その上にゲスト OS 環境を用意する方