

◆「隔離」「仮想化」「サーバ」という言葉を使用して「コンテナ」の概要を説明してください。

コンテナは、アプリとファイルシステムが隔離されたプロセスである。1つの物理マシンの中で動いているにもかかわらず、コンテナ内のプロセスだけが別マシンで動いているような状態になる。動作させたいアプリをコンテナ上に載せることで、仮想化されたサーバーを簡単に作成できます。

◆「仮想マシン」と「コンテナ」の違いとそれぞれのメリット・デメリットを説明してください
「仮想マシン」

物理マシンにホスト OS・仮想化ソフトウェアがあり、その上で仮想マシンが動作する。仮想マシンにはゲスト OS が含まれ、マシン内のゲスト OS を使用する。

- ・メリット 複数 OS を同一物理マシンに構築することができる
- ・デメリット オーバーヘッドが大きい

「コンテナ」

物理マシンにホスト OS・コンテナ実行環境(Docker 等)があり、そのうえでコンテナが動作する。コンテナにはミドルウェアまでしか含まれず、カーネルは物理マシンのホスト OS のものを利用する。

- ・メリット オーバーヘッドが小さく、とても軽量
- ・デメリット ホスト OS のカーネルしか利用できないため、別々の OS を動作させることができない

◆コンテナイメージに関して

「可搬性」、「レジストリ」という言葉を使用して概要を説明してください。

コンテナは、コンテナイメージ(テンプレートのようなもの)から作られる。イメージは、インターネット上のレジストリに保管されている。基本的にはこのイメージをそのまま使う、もしくはカスタマイズして使うことになる。このイメージを配布することで、別の環境(OS やマシン)でも再現することができ、これを可搬性が高いという。

◆Docker Hub に関して、「プル(Pull)」、「ビルド(build)」、「プッシュ(push)」に関して、それぞれの概要を説明してください。

・プル(pull)

Docker Hub からイメージを取得すること

・ビルド(build)

Dockerfile を使用してイメージを構築すること

・プッシュ(push)

プルに対して、レジストリにイメージを登録すること(他の環境で使いたい場合や、他のユーザーに配布したい場合など)

◆コンテナの4つのライフサイクルに関して4つの状態をそれぞれあげて概要を説明してください。

- 作成 -> イメージからコンテナを作成した状態。あくまで作成しただけなので、アプリは動作せず、コンテナ内へのアクセスも不可能。
- 実行 -> コンテナを動作させた状態。
- 停止 -> 実行していたコンテナを停止させた状態。再度実行することも可能。
- 削除 -> コンテナを削除した状態。再度実行するには、再作成が必要。