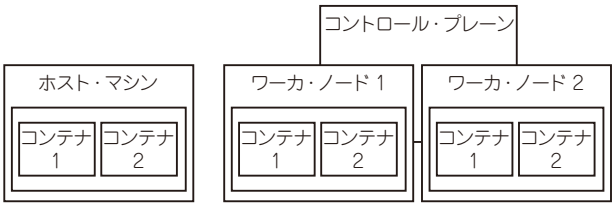
第4部

第2章 複数コンテナを手軽に管理！ローカル環境での実践で理解を深める

コンテナ運用基盤 Kubernetes入門

久保 仁詩



(a) Dockerは単一ホストでの実行環境　(b) Kubernetesは複数ホストの統合管理

図1　DockerとKubernetesの関係

表1　Kubernetesが提供する主な機能

機　能	内　容	効　果
スケーリング制御	• コンテナ数の自動調整 • リソース配分の最適化	負荷変動への自動対応
可用性管理	• 障害の自動検知 • ワークロードの移行	システムの安定稼働
デプロイ管理	• 段階的なバージョン更新 • 安全なロールバック	無停止でのアップデート
ネットワーク制御	• サービス間通信の自動化 • 経路の動的更新	接続性の維持
構成管理	• 設定の一元管理 • 環境別の切り替え	運用効率の向上

● 数十から数百のコンテナを管理する時代

最近，企業のシステム開発ではコンテナ技術の普及に伴い，アプリケーションを小さな機能単位に分けて開発・運用するマイクロサービス・アーキテクチャが広がりつつあります．この開発スタイルでは，数十～数百のコンテナを効率的に管理する必要があります．多くのコンテナを通常の運用方法で管理するのは大変です．

そこで運用基盤としてKubernetes（クバネティス）（略称k8s[注1]）への注目が高まっています．ここでは，Kubernetesの基本概念から実践的な活用方法までをローカル環境での実験を通じて解説します．

Kubernetesとは…複数ホストにまたがるコンテナの統合管理

Kubernetesは，コンテナ化されたアプリケーションのデプロイ（配置），スケーリング（規模に合わせた拡張），管理を自動化するためのオープンソース・プラットフォームです．もともとGoogleによって開発され，現在はCloud Native Computing Foundation（CNCF）によって管理されています．

● DockerとKubernetesの関係性

DockerとKubernetesは，それぞれ異なる役割を持つ補完的な関係にあります．これらは共にコンテナ・ベースのアプリケーション実行環境ですが，その目的と機能は大きく異なります．

Dockerは**図1**(a)のように単一のホスト環境での運用管理に特化しており，アプリケーションのコンテナ化と実行を担当します．開発環境から本番環境まで一貫した実行環境を提供し，環境の違いによる問題を解決します．また，Docker Composeを使用することで，単一ホスト上での複数コンテナの連携も可能です．

一方，Kubernetesは**図1**(b)のように複数のホストにまたがるコンテナの運用管理を担います．コントロール・プレーンと呼ばれる管理サーバから，複数のワーカ・ノード上で動作するコンテナを一元的に制御します．これにより，コンテナの配置やスケーリング，障害時の自動復旧といった高度な運用管理を自動化できます．

● コンテナ・オーケストレーションの重要性

数十～数百のコンテナを効率的に管理するために登場したのが，コンテナ・オーケストレーションです．Kubernetesは，このような複雑な運用作業を自動化する機能を提供します（**表1**）．

Kubernetesアーキテクチャと構成要素

表2はKubernetesの主な構成要素です．これらが

注1：Kubernetesのスペルの最初のKと最後のsの間が8文字であることからk8sと呼ばれています．