

## White Paper

# 企業のデータセンターをモダンな環境へ移行する： ハイブリッド・クラウド基盤としての Kubernetes

Sponsored by: IBM

Gary Chen

AI Gillen

September 2019

## IDC の見解

---

ITに携わる者は、数年ごとにまったく新たな技術が登場するという変化の激しい世界に身を置いている。また、このような技術は企業のコンピューター環境に大きな影響を与えることになる。ごく最近では、IT業界は、長期に渡る理想的なシナリオとしてクラウド・ベースの導入モデルの概念を採用した。しかし、サーバー・アーキテクチャー、オペレーティング・システム（OS）、および仮想化やプラットフォーム・サービスを組み合わせて構成される従来のオンプレミス・コンピューティング環境から、クラウド中心型の環境へ、直接アプリケーションやデータを移行することは、単純でないばかりか多くの困難を伴っている。

多くの組織において、オンプレミス・コンピューティング・モデルからパブリック・クラウド中心型プラットフォームへの移行を行うための実際的な方法は、ハイブリッド・クラウド・コンピューティング・モデルを活用することである。それによって企業はオンプレミス・リソースとパブリック・クラウド・リソースの両方に対して、シームレスなアクセスが可能になる。この方法は新旧のコンピューティング・アーキテクチャー間の違いを埋める効果があり、企業が主要なワークロードをパブリック・クラウドに完全に移行する際、一気に進めるのではなく、時間をかけて少しずつ段階を追って進められるようにするものである。幸いなことにIT業界は、オンプレミスとオフプレミスの両方に共通な運用特性に対応できるハイブリッド・クラウドを、企業が構築できるようにする技術の実現に向けて、すでに踏み出している。

ハイブリッド・クラウドは、高い可用性を持つマルチクラウド・プラットフォームを通じて実現することになる。IT業界では、普遍的な抽象化レイヤーの基盤としてLinux、コンテナおよびKubernetesを用いることで、この方向に向かっている。AWS（Amazon Web Services）、グーグル、IBM（IBM Cloud）、マイクロソフト、Pivotal、レッドハット、ヴァイエムウェアは、このようなプラットフォームを開発する代表的なベンダーであり、企業が、さまざまなロケーションに渡って、さまざまなサーバーやクラウドを使用することができる環境を整備している。

IBMが現在提供中の次世代のLinuxONE IIIシステムは、パフォーマンスの改善や、オンボード暗号化機能のスループットを高める新たな圧縮サービス、および広範囲なソフトウェア統合を通して、ハイブリッド・クラウド環境におけるプライベート・クラウド領域を強化するための機能が拡張されている。Red Hat OpenShiftは、LinuxONEシステムと組み合わせることで、幅広く活用できるソフトウェア・リソースとなるであろう。具体的には、IBMの新たなCloud Paks（開発者や導入に向けた一連のサービス）と連動することで、最新の開発ツール、ミドルウェア、AI（Artificial Intelligence：人工知能）データサービスおよび管理リソースが提供されることになる。

## 概況

---

ITの簡素化に向けた業界のすべての取り組みそのものが、実は「次々と付加機能を後付けしている」ことになり、逆に簡素化の波自体が新たな複雑性をもたらしているともいえる。これはクラ

ウド・コンピューティングへの移行においても当てはまる。現在、IDCはクラウド・コンピューティング自体が、以下のような点において密接に関係する技術の集合体であるとみている。

- **オンプレミスのプライベート・クラウド**：これは、従来のハードウェアおよびソフトウェアを組み合わせたものであり、従来型のエンタープライズ・サーバーとして使用できるが、管理の観点から、パブリック・クラウド環境に極めて近いエクスペリエンスをもたらすことが可能である。理想的には、オンプレミスのプライベート・クラウドシステムがあたかもパブリック・クラウド・リソースのような環境を提供するようになることである。つまり、企業がハイブリッド・クラウド導入シナリオへ移行するための有効な学習曲線はない、言い換えると経験は役立たないということである。
- **オフプレミスのパブリック・クラウド**：これは、ハイパースケールのプロバイダーかあるいは小規模なクラウド・サービス・ベンダーによって実現される、サードパーティー提供のクラウド環境である。Alibaba Cloud、AWS、Google、IBM Cloud、Microsoft Azure などがある。また、パブリック・クラウドサービスはソフトウェアのみのソリューションの中で提供され、複数のハイパースケール・クラウド・プロバイダー環境において展開されている。Pivotal Cloud Foundry（および Cloud Foundry Foundation のオープンソース・ソフトウェア・ソリューション）、Red Hat OpenShift、SUSE Cloud や VMware Cloud などがある。
- **ハイブリッド・クラウド**：これは、基本的に異なる技術というよりはユースケースである。ハイブリッド・クラウド・サービスは複数のクラウドをまたいでの運用を可能にするものである。そこではオンプレミスのプライベート・クラウドあるいはパブリック・クラウドを含んでいるか、または共通のコントロール・プレーンによって管理される共通のインフラとして複数のパブリック・クラウド（マルチクラウドと呼ぶことが多い）を使用している。多くのパブリック・クラウド・ベンダーは、ハイブリッド・クラウド戦略の下で動いており、通常はそれほど多様なプラットフォームを有していない。逆に、Pivotal、レッドハット、ヴェイムウェア、SUSEなどが提供するソフトウェア・ソリューションには、ハイブリッド・クラウドやマルチプラットフォーム・ハイブリッド・クラウドの導入シナリオに対応しているものもある。

ハイブリッド・クラウド環境のオンプレミス部分はパブリック・クラウド・サービスと同レベルの拡張性を持っていない可能性がある。また、開発者にとって関心の高いパブリック・クラウド・サービスの中には、ローカルでは利用できないものがある（ただし、レイテンシーの問題を別にすれば、リモートでの利用は可能）。一方で、プライベート・クラウドを組み込むハイブリッド・クラウド導入シナリオは、実現可能性の高い選択シナリオになりつつある。多くの企業にとって、強固なプライベート・クラウド・コンポーネントを有するハイブリッド・クラウドへの移行は好ましいシナリオとなっている。

顧客がこれまでに直面した懸念の一つとしてアプリケーションのポータビリティという課題がある。IT業界では、以下のような方法で、明確な目標達成に向けて進展しつつある。

- Python や JavaScript などのインタープリター型言語の使用が増加している。以前は、与えられたプラットフォームの性能を最大限に引き出すためにコンパイラ言語を使うことが一般的であった。現在は、コンピューティング・リソースが潤沢なため、インタープリター型言語を使用してもパフォーマンス上の懸念はない。最新の継続的インテグレーション／継続的デリバリー（CI/CD）パイプラインは使おうとするコンパイラ型言語に対し、さまざまなプラットフォームへの導入シナリオに応じてバイナリー・コードを作成するように設定できる。
- コンテナは、異なるインフラ、クラウドおよびシステム・アーキテクチャーにまたがって汎用的に使用される抽象化レイヤーである。新規および既存のアプリケーションは、マイクロ・サービスの集合体として設計されているか否かを問わず、コンテナとしてリパッケージ化されつつある。現在コンテナは、マルチアーキテクチャー・サポートにも対応できる。
- IT業界は、次世代アプリケーション用の基本的なコンテナ管理やオーケストレーション・レイヤーである Kubernetes へ急速にシフトしている。多くの移植可能な Kubernetes 環

境が、プライベート・クラウドおよびパブリック・クラウドの導入において利用可能となっている。重要なシステムやクラウドにおいて一貫性のある Kubernetes プラットフォームが利用できるならば、物理的に異なるロケーションに広がるリソースを持つハイブリッド・クラウド環境の構築が可能となる。

すべてのコンテナが新規に作成されたマイクロ・サービスベースのソフトウェアで構成されるわけではない。現実的には、さまざまな外部サービスを呼び出して利用する極めて多様なアプリケーションがコンテナに取り込まれるようになる。たとえば、あるコンテナにはホストを他のアーキテクチャーに切り変えたモノリシックなアプリケーションを含んでいるかもしれない。あるいは、別のコンテナにはステートフルで、高いセキュリティを必要とし、そのコンテナの元となった環境を反映した属性を有するような、部分的にリファクタリングされたアプリケーションが含まれるかもしれない。これらのアプリケーションは、その要件のすべてに対応できるシステムやインフラストラクチャー・ソフトウェアを必要とするであろう。

## コンテナと Kubernetes

コンテナ化された Linux アプリケーションの概念は新しいものではなく、もともと Linux のオペレーティング・システムの一部として Linux コンテナ (LXC) 技術を基に実現されていた。しかし、Docker が出現してコンテナにアプリケーションを強制的かつ容易にパッケージ化する手法が確立されるまで、LXC の採用は非常に限定的なものであった。

Docker コンテナと OCI (Open Container Initiative) コンテナは、アプリケーションとそれに付帯するすべてを単一のポータブル・コンテナのイメージへとパッケージ化する。これらのイメージは、その後集中型コンテナ・レジストリーで共有され、他の開発者によって反復修正が行われるか、本番環境へ投入される。開発者が現存するイメージの上に容易に構築できるよう、コンテナ・イメージもイメージ内にレイヤーという概念を導入している。コンテナを実行するには、それぞれのコンテナを互いに分離させ、各コンテナがあたかも自分専用の OS を持つかのように、コンテナ自体のサンドボックスで実行する。開発者が使いやすいツールや API によって、コンテナを容易に構築、共有および実行できるようになったため、ツールや API はすぐに普及した。

複雑に絡み合ってきたコンテナ化されたサービスを運用する方法が必要とされる中、アプリケーションをコンテナ化することは単なる第一歩にすぎない。グーグルは自社の「Borg」コンテナオーケストレーション技術のコンセプトや知識を利用して、IBM やレッドハットなどの他社から積極的な参加を促すオープンソース・ソフトウェア Kubernetes プロジェクトを企画した。

Kubernetes プロジェクトの運営は、このプロジェクトを本当にオープンなものとし、IT 業界で利用できるよう新たに結成された Cloud Native Computing Foundation (CNCF) へと引き継がれた。業界はその後 Kubernetes を次世代の推奨インフラストラクチャー、あるいはコンテナ化されたアプリケーション用の導入やオーケストレーションのプラットフォームとして採用した。

コンテナはポータビリティがあり、ユニバーサルなアプリケーション・パッケージの標準として大きな牽引力を持つようになっている。この理由の一つは、コンテナが OCI の一部として標準化されていることにある。OCI は、コンテナをどのように実行するか (ランタイム)、あるいはコンテナ・イメージをどのようにフォーマットするかについて定めている。標準化はまた、さまざまなコンテナ・プラットフォームやクラウド・サービス間でコンテナに一貫性があり、相互運用が可能で、高いポータビリティがあることを意味する。標準化はさらに、Kubernetes によってスタックが拡張されオーケストレーションや管理を実現する。

Kubernetes は OCI のような成文化された標準ではなく、非常に多くの業界が参加し、単独で運営されるオープンソース・プロジェクトである。そのため、コンテナが標準フォーマットを持つだけでなく、大量の管理スタックが実行環境全体で一貫したものとなっている。Istio サービス・メッシュやサーバーレスのための Knative のようなコンポーネントは、さらにいっそう一般的なスタックを提供し、牽引力を持つようになり、今後いたるところに存在する可能性がある。しかし、標準化がコンテナの導入を後押しした要因の一つにはなっているが、そもそもコンテナが盛

り上がりを見せた第一の理由は、開発者に迅速な開発ライフ・サイクルおよび DevOps による展開をもたらしたからである。

開発者にとって、コンテナは新たなクラウド・ネイティブなマイクロ・サービスを効果的にカプセル化し、CI/CD を用いて自動化されたソフトウェアのビルド・パイプラインへと移行していくのに適している。開発者が使いやすい API は、複雑なソフトウェアを用いた作業を迅速化、簡便化し、それによって開発者のワークフローが改善される。コンテナはまた、多くの付随品が用意されているため、より多くの自動テスト・システムの有効化や環境管理向上の手助けをすることでコードの質を高めることができる。最終的には、コンテナはソフトウェアの開発スピードや変更の展開を速め、開発者の生産性を高めることができる。

コンテナと Kubernetes は、大規模な Web スケールのアプリケーションを実行するために必要な拡張性のある自動化された最新的手段を運用者に提供する。Kubernetes は、めまぐるしく変化するアプリケーションを大規模かつ確実に実行するという点で、最大の Web 企業の知識と経験値を多く盛り込んでいる。ブルー/グリーン・アップデートや新しいアプリケーション機能に対する A/B テスト、複数の自動スケーリング・オプションのような展開パターンが確立された。さらに、コンテナ・スタイルの導入は、イミュータブル・インフラストラクチャーを利用して設定上の悩みの種を解決できる。これは、コンテナの状態がイメージの形で定義され、実行中は決して変化しないことを意味する。変更が必要な場合、古いコンテナのインスタンスは削除され、実行中のイメージにパッチを適用したり設定変更を加える代わりに、新しいイメージが開始される。コンテナのリポジトリも、コンテナ・イメージの集中化やバージョンの維持を助ける。軽量で応答性に優れたコンテナの性質を、最新の管理プレーンと組み合わせることで最新のアプリケーションの効率的な導入や管理を IT が実現できるようにする。

企業がハイブリッド・クラウドやマルチクラウドに移行する際、コンテナは異なる環境全体でポータビリティや一貫性において重要な役割を果たす。上記の通り、IT 業界が OCI のフォーマットや Kubernetes のコントロール・プレーンを幅広く導入することで、コンテナのプラットフォームをその核心において本質的に似たようなものに変化させる。CNCF は Kubernetes の適合試験や認証を行っているが、このためすべての Kubernetes が核となる機能について同じ振る舞いをしなくてはならない。これは、企業があらゆる Kubernetes 製品の大部分を使用でき、一定の互換性が期待できることを意味する。

開発者は、配布されるものや背後にあるクラウド・サービスやインフラストラクチャーが何であろうと、希望通りの API やツールを用いてコンテナや Kubernetes を大規模に利用できる。これによって、オンプレミスや異なるパブリック・クラウド全体において一貫した開発環境を用意できるようになる。さらに、基盤となっているハードウェアの種類にかかわらずコンテナのインターフェースは変化しないため、コンテナは異なるシステム・アーキテクチャー間の違いの抽象化を支援できる。また、開発者はシステムや OS に対するアプリケーションを開発するために、システムや OS のすべてを知る必要はない。Kubernetes を管理する運用者においては、運用者がインストールや統合をしなければならないシステム特有の知識や開発ツールは存在するが、Kubernetes や Kubernetes によって管理されるアプリケーションの運用は、あらゆるディストリビューションにおいてほぼ一貫性が維持できる。

## マルチアーキテクチャー・コンテナとハイブリッド・クラウド

コンテナによって、開発者は基盤となるシステムの違いを考慮する必要がなくなる。しかし、IBM LinuxONE のようなエンタープライズ・プラットフォームは、コンテナの運用やアプリケーションにおいてさらなる利点ををもたらす。そしてコンテナの開発者は、その利点を活用するに当たって多くの新しいスキルを学ぶ必要はない。

LinuxONE には、安全で柔軟なコンピューティング、メモリー、I/O プロビジョニングを提供する z/VM あるいは KVM のソフトウェア・ハイパーバイザーと連動するファームウェア・ベースのハイパーバイザーが内蔵されている。現在は、多くのコンテナが、ベアメタル・オーケストレーシ

オン環境よりも仮想マシン（VM）上で動いている。これらの技術は重複しているように見えるが、その多くは大きく異なるレベルで作用している。ハイパーバイザーはハードウェアを仮想化してパーティションを分割し、コンテナはOSを仮想化する。

VM上でのコンテナの実行は以下に示す複数の利点をもたらす。

- VM バウンダリーはコンテナのバウンダリーよりも強固なため、分離および分割のための追加レイヤーを設定することが可能となる。
- 現在の大容量のハードウェア・システムをより適切な規模で資源を利用できるサイズに分割できる（これによって、一つのOSカーネル上に非常に多くのコンテナを搭載するといった状況を回避できるため、セキュリティや信頼性を改善できる）。
- 異なるOS、あるいは同じOSの異なるバージョンやパッチ・レベルでさえ自由に混在させることができる。

IBM LinuxONE 特有のファームウェア・ハイパーバイザーは Kubernetes のスケールをサポートできる。IBM LinuxONE はハイパーバイザーを用いて、システムを停止することなくスケール・アップおよびスケール・アウトの両方を実施できる。たとえば、これによって、コンテナのためのCPUとメモリー容量の適正配備を支援する Kubernetes の垂直ポッド自動スケール機能をサポートすることができる。

企業がマイクロサービス・アーキテクチャーへと移行する場合、現在コンテナ化されているものの多くは従来のレガシー・アプリケーションである。それらはリファクタリングができたりできなかったりする。既存アプリケーションをコンテナ化することには利点があり、企業はコンテナ化を幅広く適用し、コンテナの約半分をレガシー・アプリケーションに割り当てている。

LinuxONE とそのハイパーバイザーが持つ柔軟性と信頼性により、このプラットフォームは、大規模でモノリシックなコンテナのホストとしても適したものとなっている。

IBM の LinuxONE システムの最新版は、IBM Cloud Paks と呼ばれる使いやすいフォーマットで、開発者と管理者に新たなソフトウェア・ソリューションを提供する。コンテナで提供される企業向け IBM Cloud Paks は、オープンソース・ミドルウェアと共に開発ツールやデータ、AI サービスを提供している。また、IBM Cloud Paks は Red Hat OpenShift Cloud Platform 上で作動する。

IBM はまた、IBM Cloud Hyper Protect Services を提供している。これは機密性の高いデータを扱うアプリケーションを開発者が容易に開発できるようにするため、IBM Secure Service Container 技術上に構築されたパブリック・クラウド・サービスのポートフォリオである。以下のようなものがある。

- **IBM Cloud Hyper Protect Crypto Services** : ユーザーが制御する Hardware Security Module を用いて行うクラウド・データの暗号化において、ユーザー自身が所有する鍵を保持できるようにするもの
- **IBM Cloud Hyper Protect DBaaS** : パブリック・クラウドにおいてデータの機密性を完全なものにするため、セキュリティが高く、使いやすい企業向けクラウド・データベース環境を提供
- **IBM Cloud Hyper Protect Virtual Servers** : 機密性の高いワークロード運用のための（ユーザー管理による）仮想サーバーへの完全な権限付与

コンテナは、イミュータブル・インフラストラクチャーや集中型イメージ・リポジトリのような機能を通じて多くのセキュリティ上の利点をもたらしているが、スタック内の新たなレイヤーとして現在も対応が必要であり、セキュリティは採用者にとって大きな課題となっている。

LinuxONE はコンテナで利用できる多くのセキュリティ機能を用意している。LinuxONE プラットフォームの設計は、ハードウェアからソフトウェアまで全体のセキュリティ強化を目的としている。IBM の暗号化コプロセッサ・アダプターは、HSM が提供する FIPS 140-2 Level 4 コンプライアンスという最も高い Linux セキュリティ認証を獲得している。また、LinuxONE は、IBM Secure Service Container という機能を提供している。これは、コンテナ化されたアプリケーション

用のセキュアなアプリケーション実行環境をもたらす高セキュリティ論理パーティションの一つである。

- 独立メモリーを持つファームウェアによる耐改ざん性があり高信頼度のブート・シーケンス
- アドミニストレーターのアクセスを制限することで、特権的（アクセス）資格の誤用を防ぎ、使用中のデータを保護
- 保存中および送信中のすべてのデータに対し透過的に、自動的かつオーバーヘッドを生じさせない暗号化を実施
- 高信頼かつ瞬時のスケールアップ機能
- 既存のデータやアプリケーションを対象とするコロケーション

## アプリケーションのポータビリティ

これまでアプリケーションは、設定・調整・最適化のために、選択されたミドルウェアやデータ管理ソフトウェア製品ごとに、特定の OS、場合によっては特定のサーバー上の特定の OS と紐付けられていた。x86 サーバーでの仮想化が普及したことで、ある仮想サーバーから別の仮想サーバーに、OS やアプリケーションと関連する付帯ソフトウェアのスタック全体を簡単に移植できるようになった。これは、現在のコンピューター環境での移行のために重要なステップではあるものの、最終的にアプリケーションを特定の OS インスタンスから分離する必要があった。

うまく分離させるには、アプリケーションとその付帯ソフトウェアを、基盤となる OS の特定のインスタンスからまとめて取り除く必要がある。こうすることで、ワークロードごとに VM と OS を一つずつ割り当てる展開モデルと、それに関連するオーバーヘッドを減らすメリットもある。

インタープリター型言語が使用されることによって、ますますプラットフォームへ依存性は薄れている。一般的に、JavaScript、Perl、Python などの言語が選択される。Java などのバイトコード・コンパイラー型言語もある。インタープリター型言語はジャスト・インタイムを基本として解釈実行するというメリットがある。そのため、バイトコードの順序付けを気にせずに、プログラムが実行されるプラットフォームによって解釈実行される。バイトコードの順序付けは、LinuxONE 系製品と x86 ソリューションとの違いの一つである。

比較すると、C、C++、Go や Haskell など広く利用されているコンパイラー型言語は、実行の前にあらかじめコンパイルされる。これは、コンパイラーが、アプリケーションを展開する先のプラットフォーム用としてバイナリー・コードを作成することを意味する。つまり、x86 の Linux プラットフォーム用にコンパイルされたコードは、IBM LinuxONE システム上の Linux では実行できない。たとえそれが OpenShift Container Platform などのソフトウェア・クラウド・プラットフォーム上であったとしても実行することはできない。

この問題に対応するため、最新の継続的インテグレーション／継続的デリバリー（CI/CD）システムでは、一般的なソースコードからコンパイルした複数のバイナリー・コード群を管理でき、適切なプラットフォームに正しいバイナリー・コードを展開できるようになっている。Linux のクロス・コンパイラーによって、x86 の Linux システム上で Z Linux バイナリー・コードをコンパイルできるようになっている。また、OCI コンテナのイメージは、複数のイメージを使い分ける代わりに、単一のイメージの中に複数のシステム・タイプに有効なバイナリー・コードを含めることができるため、マルチアーキテクチャーに対応することが可能である。

## 将来の展望

---

### 業界の方向性

情報技術の分野では、画期的な新しい技術に関して、短期間で熱狂的に受け入れられ盛り上がるが、急速に冷めてしまう傾向にある。しかし現実には、どのようなテクノロジーであろうと何十

年にも及ぶサポートという問題がついてまわる。そのためユーザー企業は、ポータビリティ、柔軟性、サポートビリティについて、将来を見据え最も適切に選択できる技術を採用することに目を向けなければならない。

業界では、コンテナの管理プレーンとして Kubernetes を採用してきたため、これがイノベーションや統合に不可欠なものとなり、複数の利点をもたらしている。まず Kubernetes 自体が、その単独のプラットフォームに注目している大規模なコミュニティから様々な利点を受けている。次に、それ自体は Kubernetes の一部ではないものの、Kubernetes と統合し最適化するプロジェクトのイノベーションが拡大していることである。

たとえば、Istio サービス・メッシュや、サーバーレス・コンピューティングに向けた Knative は直接 Kubernetes 上に構築される。これらのプロジェクトは、大規模なコミュニティの発展に努めており、幅広く利用可能な汎用コンテナ・プラットフォームを生成することで、Kubernetes を展開する上での重要な要素となる。さまざまな Kubernetes ディストリビューションやクラウド・サービス全体に共通するコンポーネントをより多く獲得することは、コンポーネントが次第にハイブリッド・クラウドやマルチクラウドをベースとするものにシフトするのに伴い、ユーザーにとって大きなメリットとなる。

IBM LinuxONE は以下のようなさまざまな方法でこのコミュニティに直接参加している。

- **IBM LinuxONE における Red Hat OpenShift Container Platform** : IBM がレッドハットのポートフォリオに求めた真に価値のあるものの一つは、OpenShift Kubernetes Platform であった。この技術はハイブリッド・クラウドの基盤として使用できる上、Amazon Web Services、Google Cloud Platform および Microsoft Azure など複数のハイパースケール・クラウド環境に展開でき、Istio や Knative に対応している。OpenShift への対応は Red Hat Enterprise Linux と連携して進めており、何年も前から LinuxONE サーバー上で利用可能となっている。
- **IBM LinuxONE における IBM Cloud Paks** : 2019 年 8 月上旬に、IBM は OpenShift 用の「Cloud Paks」のコンセプトについて発表した。これらのコンテナ化された製品には、データや AI (Cloud Pak for Data)、アプリケーション・モダナイゼーションとクラウド・ネイティブ・アプリケーションの開発支援 (Cloud Pak for Applications)、エンタープライズ・アプリケーションの統合 (Cloud Pak for Integration)、ビジネス・プロセスと意思決定の自動化 (Cloud Pak for Automation)、マルチクラウド管理用に特別に設計された製品 (Cloud Pak for Multicloud Management) を支援するための各製品が含まれる。
- **IBM LinuxONE における IBM Cloud Private** : IBM の既存の IBM Cloud Private 技術は、レッドハットの買収によって IBM のポートフォリオに新たに加わった技術と併せて、引き続き IBM からの支援や投資を受ける。
- **Linux ディストリビューション** : Canonical Ubuntu、Red Hat Enterprise Linux、SUSE Linux Enterprise Server などは、LinuxONE で利用できる。また、いくつかのオープンソース・コミュニティ・ディストリビューションに関しては、やや重要性の低いワークロードでの利用を考えるユーザーにとって便利である。Alpine や Red Hat Enterprise Linux CoreOS などの軽量カーネルによるディストリビューションも、コンテナに対応済みである。

## 課題と機会

- **課題** : 基盤となるシステムが何であろうと、コンテナは、開発者に対し、多くのものを抽象化し、同じ API を用いて同じように作業できる環境をもたらす。しかし、基盤となるシステムは、いまだにコンテナの実行方法やスケーリングおよびセキュリティの実現方法を大きく左右している。システム・ベンダーにとっての現在の課題は、インフラストラクチャーを徐々にコモディティ（必需品）として捉え始めている顧客に対し、その相違点と利点を伝えていくことである。
- **機会** : コンテナは IBM LinuxONE を新たな顧客（クラウド・ネイティブな開発者）に対し公開する。これらの開発者は、コンテナを有するインフラストラクチャーでの開発や導入

と同じ方法で、LinuxONE に対し開発を行い、導入できる。これによって、プラットフォームについて特に開発者をトレーニングすることなく現在のワークフローに LinuxONE の開発を組み入れ、コンテナ導入に参加できる。しかし、IBM Z の運用者がこのコンテナ・プラットフォームの導入や統合の任務を任されるようになるには学習が必要となるであろう。

- **課題**：多くの開発者は x86 環境での開発にはさまざまな経験を持っているが、それ以外のアーキテクチャーについてはほとんど経験がない。
- **機会**：インタープリター型言語によってポータビリティが高まっており、コンパイラー型言語は次第に、複数コンパイル・シナリオ、クロス・コンパイラー、マルチアーキテクチャー・コンテナに対応できるようになってきている。現在、GPU、ASIC、AI 用に最適化されたカスタム・プロセッサ、ARM プロセッサおよびエッジにおけるラズベリーパイ (Raspberry Pi) などの技術を含む、さまざまな形式の非 x86 プロセッサが使用されており、これは開発者がヘテロジニアス導入環境をサポートできるようなツールを採用していることを意味する。ここでは Kubernetes は共通の基準である。
- **課題**：IBM のメインフレームは「レガシー」と受け取られている。
- **機会**：IBM の LinuxONE システムは、一般的なプラットフォームに比べはるかに拡張性や信頼性、セキュリティ機能が高く、大規模なマルチクラウド導入の場合、その価値が発揮される。また、LinuxONE において、すでに使用可能な数多くのオープンソース・ツールやソフトウェアのエコシステムが存在し発展をとげている。開発者はコンテナによって基盤システムから大きく離れ、特別なスキルが必要なくなっているが、さまざまな要件に応じて、コンテナで利用可能なアプリケーションのタイプを把握するには、システムの機能を意識することがまだ重要である。サイト信頼性エンジニアリング (SRE) の役割を担うシステム・ベンダーや顧客の課題は、IBM LinuxONE の機能に合致するコンテナを LinuxONE が獲得できるよう、コンテナ・プラットフォームを通じて基盤システムの価値や違いをどのように浮かび上がらせるかという点にある。
- **課題**：オフプレミス・ソリューションへの移行を試みている顧客は、自身のデータセンターに大規模なシステムの導入を望まないであろう。
- **機会**：IBM は、IBM Cloud 上で実行される LinuxONE によって強化された IBM Cloud Hyper Protect Services を提供している。しかし、IBM は、IBM LinuxONE リソースを他のパブリック・クラウド環境で使用可能かどうかについてはまだ発表していない。IBM がコンサンプション (消費) ベースの価格モデルやマルチクラウド活用を提供するならば、顧客はこのプラットフォームを使用する機会が増える可能性がある。また、IBM は新たな LinuxONE III システムに業界標準の 19 インチフレームを採用しており、これによって標準的なデータセンターにおいて容易にシステムを設置できる。

## 結論

IT エグゼクティブは加速するイノベーションの時代に直面している。そうした中で、高まる需要に対応するため、自社独自のアプリケーションのニーズに対応し、ハイブリッド・クラウド環境におけるスケーリング、セキュリティ、さらにそれと同じくらい重要なポータビリティを提供するハイブリッド・クラウドおよびマルチクラウドのインフラストラクチャーの明確な計画を立てる必要がある。インフラストラクチャーやデプロイメント・ソフトウェア・レイヤーを標準化することは、企業が、ユーザー・エクスペリエンスの最適化、複数のユーザー・デバイスへの対応、アプリケーション自体の機能性/完全性など、自社のアプリケーションについて差別化を図ることにさらに多くのリソースを割けるようになることを意味する (また、そうすべきである)。言うまでもなく、その差別化には信頼性、スケーラビリティ、セキュリティおよび運用コストなどの中核となる属性も含まれるとみている。

IBM は革新的なプラットフォームを業界にもたらしてきた。このプラットフォームは、多数のソフトウェアや展開の経験、および開発者が現在必要とし要望する属性を提供する。これらのサービスは基盤プラットフォームから離れて抽象化されたものがほとんどであり、これは特に Red Hat



OpenShift Container Platform や IBM のプラットフォーム上でのセキュアでスケーラブルなコンテナ・デプロイメント・サービスによって実現される。IBM は、IBM LinuxONE を IT 投資に向けた行動や成果を期待している潜在的顧客に幅広く公開している。

## IDC 社 概要

International Data Corporation (IDC) は、IT および通信分野に関する調査・分析、アドバイザリーサービス、イベントを提供するグローバル企業です。50年にわたり、IDCは、世界中の企業経営者、IT 専門家、機関投資家に、テクノロジー導入や経営戦略策定などの意思決定を行う上で不可欠な、客観的な情報やコンサルティングを提供してきました。

現在、110 か国以上を対象として、1,100 人を超えるアナリストが、世界規模、地域別、国別での市場動向の調査・分析および市場予測を行っています。

IDC は世界をリードするテクノロジーメディア（出版）、調査会社、イベントを擁する IDG（インターナショナル・データ・グループ）の系列会社です。

## IDC Japan

IDC Japan（株） 〒 102-0073 東京都千代田区九段北 1-13-5

81.3.3556.4760

Twitter: @IDC

idc-community.com

www.idc.com

---

### Copyright Notice

External Publication of IDC Information and Data – Any IDC information that is to be used in advertising, press releases, or promotional materials requires prior written approval from the appropriate IDC Vice President or Country Manager. A draft of the proposed document should accompany any such request. IDC reserves the right to deny approval of external usage for any reason.

Copyright 2019 IDC. Reproduction without written permission is completely forbidden.

